



**Solicitation Information
November 10, 2021**

RFP #7664822

TITLE: CLOUD ERP SERVICES FOR THE UNIVERSITY OF RHODE ISLAND

Submission Deadline: December 8, 2021 2:00 PM (Eastern Time)

**PRE-BID/ PROPOSAL CONFERENCE: NO
MANDATORY:**

If YES, any Vendor who intends to submit a bid proposal in response to this solicitation must have its designated representative attend the mandatory Pre-Bid/ Proposal Conference. The representative must register at the Pre-Bid/ Proposal Conference and disclose the identity of the vendor whom he/she represents. A vendor's failure to attend and register at the mandatory Pre-Bid/ Proposal Conference shall result in disqualification of the vendor's bid proposals as non-responsive to the solicitation.

DATE:

LOCATION:

Questions concerning this solicitation must be received by the Division of Purchases at doa.purquestions15@purchasing.ri.gov no later than **December 22, 2021 2:00 PM (EST)**. Questions should be submitted in a *Microsoft Word attachment*. Please reference **RFP #7664822** on all correspondence. Questions received, if any, will be posted on the Division of Purchases' website as an addendum to this solicitation. It is the responsibility of all interested parties to download this information.

BID SURETY BOND REQUIRED: NO

PAYMENT AND PERFORMANCE BOND REQUIRED: NO

Robert DeAngelis, Senior Buyer

Note to Applicants:

1. Vendors must register in RIVIP at the Division of Purchases' website at <https://www.purchasing.ri.gov/RIVIP/VendorRegistration.aspx>.
2. Proposals received without a completed RIVIP Vendor Certification Cover Form attached may result in disqualification.

THIS PAGE IS NOT A RIVIP VENDOR CERTIFICATION COVER FORM

COVID-19 EMERGENCY PROTOCOL FOR BID OPENINGS

Vendors and the public are advised that due to Covid-19 emergency social distancing requirements bid openings at the Division of Purchases shall be conducted via live streaming on the ZOOM website/application. Vendors and the public shall not be permitted to enter the Division of Purchases to attend bid openings. Vendors and the public who attend bid openings via live streaming shall be required to identify themselves and a record of all such attendees shall be maintained by the Division of Purchases. Vendor bid proposals shall be opened and read aloud at the date and time listed herein. The results of bid solicitations requiring a public copy for public works projects shall be posted on the Division of Purchases website as soon as possible after the bid opening. For RFP solicitations only vendor names shall be read aloud at the opening.

Vendors and the public are further advised that visitor access to the Powers Building at One Capitol Hill, Providence, RI requires pre-screening at the entrance to the building. In accordance with the Governor's Executive Order(s) and Department of Health emergency regulations all visitors to the Powers Building must wear a cloth mask which covers the nose and mouth. Vendors delivering bid proposals to the Division of Purchases should allow sufficient time for the pre-screening process. The Division of Purchases assumes no responsibility for delays caused by the screening process or any other reason. Vendors are solely responsible for on time delivery of bid proposals. The Division of Purchases shall not accept late bids for any reason.

BID OPENING ZOOM INFORMATION

Division of Purchases is inviting you to a scheduled Zoom meeting for the bid opening.

Topic: 7664822

Time: Dec 8, 2021 02:00 PM Eastern Time (US and Canada)

Join Zoom Meeting

<https://us02web.zoom.us/j/85797441172?pwd=ZkpqMkVWcDNodXI4VXBSbjlYeTVXQT09>

Meeting ID: 857 9744 1172

Passcode: 210813

One tap mobile

+16465588656,,85797441172#,,,,*210813# US (New York)

+13017158592,,85797441172#,,,,*210813# US (Washington DC)

Dial by your location

+1 646 558 8656 US (New York)

+1 301 715 8592 US (Washington DC)

+1 312 626 6799 US (Chicago)

+1 669 900 9128 US (San Jose)

+1 253 215 8782 US (Tacoma)

+1 346 248 7799 US (Houston)

833 548 0276 US Toll-free

833 548 0282 US Toll-free

877 853 5247 US Toll-free

888 788 0099 US Toll-free

Meeting ID: 857 9744 1172

Passcode: 210813

Find your local number: <https://us02web.zoom.us/j/kuG418982>

Table of Contents

SECTION 1: INTRODUCTION	5
Instructions and Notifications to Vendors	5
SECTION 2: BACKGROUND	10
SECTION 3: SCOPE OF WORK AND REQUIREMENTS	11
SECTION 4: PROPOSAL	16
A. Technical Proposal	16
B. Cost Proposal	16
C. ISBE Proposal	16
SECTION 5: EVALUATION AND SELECTION	17
SECTION 6: QUESTIONS	19
SECTION 7: PROPOSAL CONTENTS	19
SECTION 8: PROPOSAL SUBMISSION	21
SECTION 9: CONCLUDING STATEMENTS	21
APPENDIX A. PROPOSER ISBE RESPONSIBILITIES AND MBE, WBE, AND/OR DISABILITY BUSINESS ENTERPRISE PARTICIPATION FORM	22
APPENDIX B. WORKLOAD REPOSITORY REPORT	25

SECTION 1: INTRODUCTION

The Rhode Island Department of Administration/Division of Purchases, on behalf of the University of Rhode Island (“URI”), is soliciting proposals from qualified firms to provide a cloud infrastructure to host our PeopleSoft system and provide managed services for both the infrastructure and application stack of PeopleSoft, in accordance with the terms of this Request for Proposals (“RFP”) and the State’s General Conditions of Purchase, which may be obtained at the Division of Purchases’ website at www.ridop.ri.gov.

The initial contract period will begin approximately March 1, 2022 for one year. Contracts may be renewed for up to four additional 12-month periods based on vendor performance and the availability of funds.

This is a Request for Proposals, not a Request for Quotes. Responses will be evaluated on the basis of the relative merits of the proposal, in addition to cost; there will be no public opening and reading of responses received by the Division of Purchases pursuant to this solicitation, other than to name those vendors who have submitted proposals.

Instructions and Notifications to Vendors

1. Potential vendors are advised to review all sections of this RFP carefully and to follow instructions completely, as failure to make a complete submission as described elsewhere herein may result in rejection of the proposal.
2. Alternative approaches and/or methodologies to accomplish the desired or intended results of this RFP are solicited. However, proposals which depart from or materially alter the terms, requirements, or scope of work defined by this RFP may be rejected as being non-responsive.
3. All costs associated with developing or submitting a proposal in response to this RFP or for providing oral or written clarification of its content, shall be borne by the vendor. The State assumes no responsibility for these costs even if the RFP is cancelled or continued.
4. Proposals are considered to be irrevocable for a period of not less than 180 days following the opening date, and may not be withdrawn, except with the express written permission of the State Purchasing Agent.
5. All pricing submitted will be considered to be firm and fixed unless otherwise indicated in the proposal.
6. It is intended that an award pursuant to this RFP will be made to a prime vendor, or prime vendors in the various categories, who will assume responsibility for all aspects of the work. Subcontracts are permitted, provided that their use is clearly indicated in the vendor’s proposal and the subcontractor(s) to be used is identified in the proposal.
7. The purchase of goods and/or services under an award made pursuant to this RFP will be contingent on the availability of appropriated funds.
8. Vendors are advised that all materials submitted to the Division of Purchases for consideration in response to this RFP may be considered to be public records as defined in

R. I. Gen. Laws § 38-2-1, *et seq.* and may be released for inspection upon request once an award has been made.

Any information submitted in response to this RFP that a vendor believes are trade secrets or commercial or financial information which is of a privileged or confidential nature should be clearly marked as such. The vendor should provide a brief explanation as to why each portion of information that is marked should be withheld from public disclosure. Vendors are advised that the Division of Purchases may release records marked confidential by a vendor upon a public records request if the State determines the marked information does not fall within the category of trade secrets or commercial or financial information which is of a privileged or confidential nature.

9. Interested parties are instructed to peruse the Division of Purchases website on a regular basis, as additional information relating to this solicitation may be released in the form of an addendum to this RFP.
10. By submission of proposals in response to this RFP vendors agree to comply with R. I. General Laws § 28-5.1-10 which mandates that vendors/subcontractors doing business with the State of Rhode Island exercise the same commitment to equal opportunity as prevails under Federal contracts controlled by Federal Executive Orders 11246, 11625 and 11375.

Vendors are required to ensure that they, and any subcontractors awarded a subcontract under this RFP, undertake or continue programs to ensure that minority group members, women, and persons with disabilities are afforded equal employment opportunities without discrimination on the basis of race, color, religion, sex, sexual orientation, gender identity or expression, age, national origin, or disability.

Vendors and subcontractors who do more than \$10,000 in government business in one year are prohibited from engaging in employment discrimination on the basis of race, color, religion, sex, sexual orientation, gender identity or expression, age, national origin, or disability, and are required to submit an “Affirmative Action Policy Statement.”

Vendors with 50 or more employees and \$50,000 or more in government contracts must prepare a written “Affirmative Action Plan” prior to issuance of a purchase order.

- a. For these purposes, equal opportunity shall apply in the areas of recruitment, employment, job assignment, promotion, upgrading, demotion, transfer, layoff, termination, and rates of pay or other forms of compensation.
- b. Vendors further agree, where applicable, to complete the “Contract Compliance Report”(<http://odeo.ri.gov/documents/odeo-eeo-contract-compliancereport.pdf>), as well as the “Certificate of Compliance” (<http://odeo.ri.gov/documents/odeo-eeo-certificate-of-compliance.pdf>), and submit both documents, along with their Affirmative Action Plan or an Affirmative Action Policy Statement, prior to issuance of a purchase order.

For further information, contact the Rhode Island Equal Employment Opportunity Office via e-mail at odeo.eoo@doa.ri.gov.

11. In accordance with R. I. Gen. Laws § 7-1.2-1401 no foreign corporation has the right to transact business in Rhode Island until it has procured a certificate of authority so to do

from the Secretary of State. This is a requirement only of the successful vendor(s). For further information, contact the Secretary of State at (401-222-3040).

12. In accordance with R. I. Gen. Laws §§ 37-14.1-1 and 37-2.2-1 it is the policy of the State to support the fullest possible participation of firms owned and controlled by minorities (MBEs) and women (WBEs) and to support the fullest possible participation of small disadvantaged businesses owned and controlled by persons with disabilities (Disability Business Enterprises a/k/a “DisBE”)(collectively, MBEs, WBEs, and DisBEs are referred to herein as ISBEs) in the performance of State procurements and projects. As part of the evaluation process, vendors will be scored and receive points based upon their proposed ISBE utilization rate in accordance with 150-RICR-90-10-1, “Regulations Governing Participation by Small Business Enterprises in State Purchases of Goods and Services and Public Works Projects”. As a condition of contract award vendors shall agree to meet or exceed their proposed ISBE utilization rate and that the rate shall apply to the total contract price, inclusive of all modifications and amendments. Vendors shall submit their ISBE participation rate on the enclosed form entitled “MBE, WBE and/or DisBE Plan Form”, which shall be submitted in a separate, sealed envelope as part of the proposal. ISBE participation credit will only be granted for ISBEs that are duly certified as MBEs or WBEs by the State of Rhode Island, Department of Administration, Office of Diversity, Equity and Opportunity or firms certified as DisBEs by the Governor’s Commission on Disabilities. The current directory of firms certified as MBEs or WBEs may be accessed at <http://odeo.ri.gov/offices/mbeco/mbe-wbe.php> . Information regarding DisBEs may be accessed at www.gcd.ri.gov . For further information, visit the Office of Diversity, Equity & Opportunity’s website, at <http://odeo.ri.gov/> and see R.I. Gen. Laws Ch. 37-14.1, R.I. Gen. Laws Ch. 37-2.2, and 150-RICR-90-10-1. The Office of Diversity, Equity & Opportunity may be contacted at, (401) 574-8670 or via email Dorinda.Keene@doa.ri.gov

13. In the RIVIP Vendor Certification Cover Form, Section 4, Question 11, bidders shall certify agreement to the State’s contract terms. However, in accordance with Section 220-RICR-30-00-13.3(C)(3) of the General Conditions, the Vendor may submit in their bid or proposal, qualified or conditional offers which impose limitations of the Vendor’s liability or modify the requirements of the solicitation, offers for alternate specifications, or offers which are made subject to different terms and conditions, including form contracts, other than those specified by the State.” However, qualified or conditional offers “may be, at the sole discretion of the State Purchasing Agent:
 - a. Rejected as being non-responsive; or,
 - b. Set aside in favor of the requirements set forth in the solicitation (with the consent of the Vendor); or,
 - c. Accepted, if the State Purchasing Agent determines in writing that such acceptance is in the best interest of the State.”

By submitting a conditional or qualified offer, the Vendor bears the risk of their bid or proposal being considered non-responsive. In the event the State receives a conditional or qualified offer, the State reserves the right to adjust evaluation points in an RFP procurement, conduct a best and final offer process offering the same terms to all vendors, and/or reject a qualified/conditional proposal as being non-responsive at any time during the review process. The Vendor should not assume that any further negotiation will occur upon selection.

14. **Insurance Requirements** – In accordance with this solicitation, or as outlined in Section 13.19 of the General Conditions of Purchase, found at <https://rules.sos.ri.gov/regulations/part/220-30-00-13> and General Conditions - Addendum

A found at <https://www.ridop.ri.gov/documents/general-conditions-addendum-a.pdf>, the following insurance coverage shall be required of the awarded vendor(s):

General Requirements:

- 14a) Liability - combined single limit of \$1,000,000 per occurrence, \$1,000,000 general aggregate and \$1,000,000 products/completed operations aggregate.
- 14b) Workers' compensation - \$100,000 each accident, \$100,000 disease or policy limit and \$100,000 each employee.
- 14c) Automobile liability - \$1,000,000 each occurrence combined single limit.
- 14d) Crime - \$500,000 per occurrence or 50% of contract amount, whichever is greater.

Professional Services:

- 14e) Professional liability ("errors and omissions") - \$2,000,000 per occurrence, \$2,000,000 annual aggregate.
- 14f) Environmental/Pollution Liability when past, present or future hazard is possible - \$1,000,000 per occurrence and \$2,000,000 aggregate.
- 14g) Working with Children, Elderly or Disabled Persons – Physical Abuse and Molestation Liability Insurance - \$1 Million per occurrence.

Information Technology and/or Cyber/Privacy:

- 14h) Technology Errors and Omissions - Combined single limit per occurrence shall not be less than \$5,000,000. Annual aggregate limit shall not be less than \$5,000,000.
- 14i) Information Technology Cyber/Privacy – minimum limits of \$5,000,000 per occurrence and \$5,000,000 annual aggregate. If Contract Party provides:
 - a) key back-office services Contract Party shall have a minimum limit of \$10,000,000 per occurrence and \$10,000,000 annual aggregate.
 - b) if Contract Party has access to Protected Health Information as defined in HIPAA and its implementing regulations, Personal Information as defined in R.I. Gen. Laws § 11-49.3-1, et seq., or as otherwise defined in the Contract (together Confidential Information”), Contract Party shall have as a minimum the per occurrence, per annual aggregate, the total rounded product of projected number of persons data multiplied by \$25 per person breach response expense per occurrence; but no less than \$5,000,000 per occurrence, per annual aggregate; or,

- c) if the Contract Party provides or has access to mission critical services, network architecture and/or the totality of confidential data \$20,000,000 per occurrence and in the annual aggregate.

Other:

Specify insurance type and minimum coverage required, e.g. builder's risk insurance, vessel operation (marine or aircraft):

- 14j) Other - Specify insurance type and minimum coverage required

15. **Complete a separate Higher Education Cloud Vendor Assessment Tool:** The Higher Education Cloud Vendor Assessment Tool (HECVAT) may be accessed electronically at <https://security.uri.edu/forms/sig/> and will need to be completed by each vendor. The "HECVAT" is intended to simplify and speed up the process of gathering the information to assess the controls used by your organization to protect the University's data, comply with the terms of the Agreement and to provide an operationally stable, protected and recoverable service. Your printed completed copy of the HECVAT, provided with your RFP response, will be reviewed and approved for compliance by the Associate Director of Information Security prior to the Technical Review. HECVATs not approved by the URI Director of Information Security will not proceed to the Technical Review.

SECTION 2: BACKGROUND

The University of Rhode Island (URI) is the state's only public land grant research institution. Founded in 1892, the University's main campus is in Kingston, Rhode Island with 3 additional satellite campuses in the state.

URI purchased and implemented PeopleSoft Human Capital Management ("HCM"), Financials and Student systems in 1998, and has continuously operated these systems in our data centers since that time.

As is common with many higher-education People Soft systems, the three major products have significant customizations, bolt-on packages and other modifications implemented over the years of operation. In PeopleSoft HCM there are approximately 1,440 modifications, 27,877 modifications in Campus and 11,348 modifications in Financials.

In early 2019, IT services began a project to move these systems to the public cloud. While partially successful, with several development instances currently running on the Oracle Cloud Infrastructure, all production instances of the products remain in our data centers.

In addition to slow progress implementing the three products into the cloud, URI is finding it increasingly challenging to maintain current PeopleSoft Update Machine ("PUM"), Oracle and other updates. Our current major version of all three products is 9.2, running on Linux servers against Oracle 12c. A full description of PUM levels, PeopleTools versions and other details of our current environment is provided in Section 4 titled PeopleSoft Environment Overview. More detailed information will be provided during the bidding process.

URI is aware that all three PeopleSoft systems have been identified as end-of-life in 2030. We are in the initial stages (pre-RFP) of charting our future course for these key administrative components and anticipate that the timeframe for full implementation at URI will be at least five years. Success in this RFP will allow URI staff to devote time and effort to the new implementation and is key to making progress in this vital task.

SECTION 3: SCOPE OF WORK AND REQUIREMENTS

General Scope of Work

URI is seeking a partner who will both help complete the transition project described in Section 2 and subsequently manage these services for the University. We are seeking a full managed services proposal for all three systems such that the selected vendor performs all regular operational tasks including all operating system maintenance, database maintenance and application maintenance tasks. We expect that any offeror will maintain 24x7x365 monitoring of these environments and will be able to scale up and scale down the size of the environments in response to the academic calendar (peak registration, peak application, grade reporting, etc.).

The partner must work within URI's Oracle Cloud Infrastructure (OCI) cloud tenancy. All proposals will be evaluated with consideration of the estimated total cost, to URI, excluding cloud infrastructure costs. Cost consideration includes pricing, URI human resource requirements and perceived risks to success for the university. Potential partners are asked to include in their proposal a breakdown of the anticipated cloud resources and infrastructure needed to implement their proposal.

In simplest terms, URI is seeking a partner who can make our PeopleSoft systems operate as if they were SaaS systems. A successful partnership will minimize or eliminate our need to perform system administration, database administration and application administration tasks for these three platforms.

Specific Activities / Tasks

Migration tasks

- Transition HCM, Financials and Campus application stacks to the cloud.
- Provide a detailed plan to re-architect the platform of each environment, we are not looking for a "lift and shift" proposal.
- Work with the staff of URI to maintain a stable PeopleSoft system during transition.

System Improvement

- The vendor shall work with the URI staff to improve the design and organization of the infrastructure, and technical processes of PeopleSoft and all its supporting systems. URI's current cloud attempt was somewhat of a "lift and shift" model, leaving the current implementation of PeopleSoft needing to be improved, possibly revamped while continuing to use OCI. For this proposal URI is **not** looking for an end result that is a simple "*lift-and-shift*" but more of a "*lift and improve*" of these systems into the cloud. The vendor should pivot from URI's existing implementation into one that follows best practices for PeopleSoft, as well as the cloud, such as: high-availability, scalability, quick recoverability, and good standard operating procedures, etc. This may be implemented as part of the migration, or separately after migration.

Ongoing Service Operation tasks

- Manage the cloud hosting platform, including all operating systems, host provisioning and scaling operations.
- Manage comprehensive security around the cloud infrastructure and application stacks.
- Provide the ability to scale servers and other infrastructure both automatically and manually to meet varying application usage/traffic. Note that URI sees dramatically high rates of usage based on the business cycle of the community that each system supports.

For example, the student system will see a very large spike in usage the week before a term as well as the first week or two; the Financial system will see an increase in usage and processing with fiscal year end processing. Unexpected events can also create sudden increases in usage.

- Provide cloud and platform support within specific SLAs.
- Implement monitoring, and reporting on system metrics, alerts, performance degradation, and suspicious activity
- Manage and implement all PeopleSoft upgrades on the cloud platform including Operating System, Database and Application upgrades.
- Work collaboratively with both URI staff and Peoplesoft to troubleshoot and resolve product issues manifested in the cloud environment(s), including any database issues presented by our product integrations.
- Support production, development, and test environments that are isolated from other client environments and each other and development and test environments that can be simply, quickly and easily refreshed from production.
- Work to minimize risk to university operations during updates, patches, and major upgrades
- Provide a full contingent of administration services to keep the PeopleSoft environment fully operational, including support to agency’s use of third-party PeopleSoft integration technologies.
- Provide Disaster Recovery services for all managed URI environments with a required Recovery Time Objective (RTO) of 24 hours and Recovery Point Object (RPO) of within 24 hours previous.

Technical Elements

Address the following elements in the RFP, specifically each of the following elements:

1. Staff Qualifications – Table 1

1.1	Staff who are capable of providing PeopleSoft redesign architecture for the PeopleSoft stack
1.2	Staff with experience in cloud infrastructure, Linux, Windows, Oracle and all other parts of the PeopleSoft application stack.
1.3	Staff with experience in transforming PeopleSoft architecture to the cloud, while taking advantage of new processes for automation opportunities.
1.4	Staff who are experienced with PeopleSoft Test Framework, Fluid, Push Technology, Integration Broker, Integrations, PUM management, and Change Assistant.
1.5	Staff who are experts in cloud managed services for PeopleSoft.

2. Capability, Capacity, and Qualifications of the Offeror – Table 2

2.1	Provide a brief company biography with an organizational structure and core competencies.
2.2	How many consecutive years has your company been providing the services defined in this RFP?
2.3	Is your company the subject of pending or current litigation? If so, describe all.
2.4	Provide merger/acquisition history.
2.5	Describe key differentiators from other vendors in the managed services marketplace.
2.6	Provide a description of the ownership of your company.
2.7	Provide your current customer retention percentage rate. Provide the annualized rate for the current and past 3 years.
2.8	Provide your most recent ratings for each of the agencies listed below:

	A.M. Best Standard & Poor's Fitch Moody's
2.9	What fiduciary responsibility does your organization assume for this project?
2.10	Who is your compliance officer/consultant or legal counsel?
2.11	Provide at least three customer references, including names, addresses, phone number, email addresses, dates of service and types of service provided for similar sized clients with similar scope of work. We require at least one higher-education reference for each of the three PeopleSoft systems. Additional higher-education references are preferred but not required.

3. Work Plan – Table 3

Please describe how your company plans to address the following technical areas in your work plan. **In all cases, cite any exceptions, special considerations, or limitations.**

3.1	Re-platforming / Migration
3.1.1	<ul style="list-style-type: none"> ● Cloud Platform. Provide a listing of the OCI technical resources and infrastructure that the vendors anticipate is needed to implement their proposal.
3.1.2	<ul style="list-style-type: none"> ● Integration with University and third-party systems and data sources. Describe your approach to planning, implementation, verification, and maintenance for the smooth transition of existing integrations, and the addition of any new integrations in the future. See attachments to the RFP for more details.
3.2	Operations and Monitoring
3.2.1	<ul style="list-style-type: none"> ● Development instances, refresh cycles, and other related operations. Describe your overall philosophy, implementation approach, and give examples of workflows.
3.2.2	<ul style="list-style-type: none"> ● Systems monitoring, metrics, alerts -- performance and security -- application, database, infrastructure. Identify and describe any software your company uses to monitor any special considerations relating to periodic and irregular events, such as deployments, upgrades, patches, and fixes.
3.2.3	<ul style="list-style-type: none"> ● Access to all logs for development, processing, and troubleshooting. Describe what access your organization will provide to any logs and similar information.
3.3	Reporting & Assessments. Describe your plans for what routine reporting and meetings will be implemented to ensure that relevant standards are being met, such as in systems operations and performance, public end-user task metrics, IT end-user task metrics, and any other areas that your organization deems appropriate.
3.4	Problem detection, diagnosis, and resolution. Describe your organization's approach to this, including workflow, escalation, remediation, and incident reporting. Include any relevant software or automated frameworks.
3.5	Patches, Fixes, Upgrades For each of the below areas, describe your organization's approach for all relevant processes, such as initial implementation (where applicable), refresh cycles and/or schedule, configuration auditing, maintenance, documentation, and knowledge transfer to client. <ul style="list-style-type: none"> ● Please be specific within each of the following items in 3.5, defining the timelines from when Oracle/PeopleSoft releases a patch or fix and when it will be available in URI's environment. ● Outline your organization's process for determining mutually agreeable days and times for applying patches and fixes from test to live production. ● Please provide examples from your higher education references of such practices.
3.5.1	<ul style="list-style-type: none"> ● PeopleSoft Upgrades for Tools, Weblogic, Tuxedo, Java, and all relevant software
3.5.2	<ul style="list-style-type: none"> ● Full PeopleSoft PUMs, patches, and new functionality

3.5.3	<ul style="list-style-type: none"> • Oracle Databases
3.5.4	<ul style="list-style-type: none"> • Server/VM/machine Operating Systems
3.5.5	<ul style="list-style-type: none"> • PeopleSoft Test Framework (new to URI)
3.6	Development
3.6.1	<ul style="list-style-type: none"> • Describe planned change management processes and procedures in all components of our systems.
3.6.2	<ul style="list-style-type: none"> • Describe your organization's policy to allow backend updates to the database, production and development.
3.6.3	<ul style="list-style-type: none"> • Describe the management, organization, and implementation of PeopleSoft customizations
3.6.4	<ul style="list-style-type: none"> • Describe how you would make continual improvements to the systems, such as in terms of efficiency, performance, automation, and security.
3.7	Disaster Recovery
3.7.1	<ul style="list-style-type: none"> • Describe your Disaster Recovery services for all managed URI environments. URI requires a minimum Recovery Time Objective (RTO) of 24 hours and Recovery Point Object (RPO) of within 24 hours previous.

4. Approach/Methodology – Table 4

4.1	Provide a narrative description of the proposed project team and its organizational structure, including resumes of key personnel on the project team.
4.2	<p>The Vendor must provide a comprehensive implementation plan which includes the recommended project methodology and a minimum of the following: project tasks, milestones, critical path, task start and end dates, appropriate resources to accomplish each task, budget expenditures, deliverables, constraints, and assumptions. The plan must include Planning, Design, Development, System Testing, Implementation, Training, and Documentation.</p> <ul style="list-style-type: none"> • This plan must include the plans and goals for the “lift and improve” services URI is requesting: what are the improvements, why are they recommended, and what is the intended outcome? How will this impact the specifics of the implementation to go live and on an ongoing basis?
4.3	<p>The successful vendor shall meet University's project team during an agreed upon time frame for discovery and planning. During this initial phase following the award of the contract. During these meetings, a “Detailed Project Plan” shall be developed, reviewed, and revised. The University shall provide project direction and shall maintain final approval of decisions relating to the overall design / configuration, development, implementation, testing, and acceptance of the software.</p> <ul style="list-style-type: none"> • Please provide details of your organization's plans for this phase.
4.4	Vendor Project Manager should have IT project management and system development, implementation experience in managing projects that are similar in scope and successfully implemented. Project managers must have supervisory skills and be able to work in a collaborative team environment and mentor clients.
4.5	Successful Vendor shall participate in monthly / weekly project status and performance review meetings to ensure measurable progress is being achieved and the Department's standards are followed. The activities of the vendor project team shall be directed, coordinated and communicated to ensure that the project progresses per project work plan and is completed on schedule.
4.6	Successful vendor shall provide detailed, written weekly status reports as appropriate at the stage of the project to the University project team. The status report shall include project status, description of the tasks and goals, work in progress, work accomplishments, date and percentage of work completed, major problems and their resolutions or alternative recommendations, critical issues and their possible solutions, work planned for next week. Project reports and dashboards will be provided through the vendors own platform.

PeopleSoft Environment Overview
Please see attachments to the RFP for more details.

General Information
Pillars of PeopleSoft Currently Running is Human Capital Management (HCM) - Financials (FS, SCM) – Campus Solutions (CS)
The following modules being used for each application pillar(s) <ul style="list-style-type: none"> ● FS -AR AP Purchasing – Travel & Expense ● CS – All except Prospecting ● HCM – Basic HR and Payroll
Languages used in the application pillar(s) to be hosted are: <ul style="list-style-type: none"> ● PL/SQL being called by app/engines ● App / Engines ● Cobol ● SQR
Our database is Unicode.
There is no requirement for multi-currency or multi-language.
Major release of the PeopleSoft application for each pillar is 9.2
Current version of PeopleTools being used for each pillar is FS 8.55.15, CS 8.56.12, HCM 8.58.08
We are back leveled on PUMS and would like to jump to the latest version. Our current PUM level is: <ul style="list-style-type: none"> ● HCM – PUM 24 ● CS – PUM 17 ● FS – PUM 23
FS PeopleTools is currently being updated from Tools 8.55 to 8.58.
We use Modo Labs instead of PeopleSoft Interaction Hub/Portal
In HCM & FS we have approximately 3,130 users in campus we have 20,000 users.
We have approximately 30 2-tier developer users.
We have a total of three non-production environments for each pillar. Fix Dev QA
We do have recurring refreshes of our non-production database environments, but on an ad hoc rotation. It revolves around regulatory updates and production issues.
Production database size for each pillar. <ul style="list-style-type: none"> ● FS – 553181.25 MB ● CS - 505680 MB ● HCM – 154743 MB
We have one Linux process scheduler for each system CS, HCM and FS with a Windows nVision Process Scheduler for FS.
We store credit card data.
We have a system containing the Informatica software agent, running along-side our PeopleSoft systems.
Users access in a combination of ways by using the public Internet and site to site connectivity.
Currently use PeopleSoft authentication. We hopefully plan to use LDAP authentication. We do not currently use two-factor authentication.
The major issue we have with our environment is keeping up with regular maintenance, PeopleSoft maintenance and Oracle Performance Tuning.
We use Enterprise Search (ES) with HCM and Campus.
We use Fluid in Campus and HCM

SECTION 4: PROPOSAL

A. Technical Proposal

Narrative and format: The proposal should address specifically each of the following elements:

1. **Staff Qualifications** – Provide staff resumes/CV and describe qualifications and experience of key staff who will be involved in this project, including their experience in providing the services of the offering including answers to Table 1 in the Technical Elements section above.
2. **Capability, Capacity, and Qualifications of the Vendor** - Please provide a detailed description of the Vendor’s experience as a Managed Services Provider and answers to Table 2 in the Technical Elements section above.
 - a. List a minimum of three (3) relevant client references as described in item 2.11 in Table 2 in the Technical Elements section above.
3. **Work Plan** - Please describe in detail, the framework within which requested as described in Table 3 in the Technical Elements section above. Please provide specific answers on how these services will be performed, including timelines and required URI resources.
4. **Approach/Methodology** – Define the methodology to be used for meeting the requirements of the Work Plan and answers to Table 4 in the Technical Elements section above.

B. Cost Proposal

Provide a proposal cost proposal to include the following:

1. Initial one-time charge of re-platforming/migration out of our on-premises infrastructure.
2. One-time charge for the improvement and enhancement of the PeopleSoft and supporting systems.
3. Ongoing, regular annual costs for managed services.
4. Optional services available from offeror outside of those specified in the RFP which offeror feels will enhance the services solicited for University of Rhode Island.

C. ISBE Proposal

See Appendix A for information and the MBE, WBE and/or Disability Business Enterprise Participation Plan form(s). Vendors are required to complete, sign and submit these form(s) with their overall proposal in a sealed envelope. Please complete separate form(s) for each MBE, WBE and/or Disability Business Enterprise subcontractor to be utilized on the solicitation.

SECTION 5: EVALUATION AND SELECTION

Proposals shall be reviewed by a technical evaluation committee (“TEC”) comprised of staff from State agencies. The TEC first shall consider technical proposals.

Technical proposals must receive a minimum of 60 (85.7%) out of a maximum of 70 points to advance to the cost evaluation phase. Technical proposals scoring less than 60 points shall not have the accompanying cost or ISBE participation proposals opened or evaluated; such proposals shall not receive further consideration.

Technical proposals scoring 60 points or higher shall have the cost proposals evaluated and assigned up to a maximum of 30 points bringing the total potential evaluation score to 100 points. As total possible evaluation points are determined, vendor ISBE proposals shall be evaluated and assigned up to 6 bonus points for ISBE participation.

The Division of Purchases reserves the right to select the vendor(s) or firm(s) (“vendor”) that it deems to be most qualified to provide the goods and/or services as specified herein; and, conversely, reserves the right to cancel the solicitation in its entirety in its sole discretion.

Proposals shall be reviewed and scored based upon the following criteria:

Criteria	Possible Points
Staff Qualifications	10 Points
Capability, Capacity, and Qualifications of the Vendor	15 Points
Work Plan	25 Points
Approach/Methodology	20 Points
Total Possible Technical Points	70 Points
Cost proposal*	30 Points
Total Possible Evaluation Points	100 Points
ISBE Participation**	6 Bonus Points
Total Possible Points	106 Points

*Cost Proposal Evaluation:

The vendor with the lowest cost proposal shall receive one hundred percent (100%) of the available points for cost. All other vendors shall be awarded cost points based upon the following formula:

$$(\text{lowest cost proposal} / \text{vendor's cost proposal}) \times \text{available points}$$

For example: If the vendor with the lowest cost proposal (Vendor A) bids \$65,000 and Vendor B bids \$100,000 for monthly costs and service fees and the total points available are thirty (30), Vendor B's cost points are calculated as follows:

$$\$65,000 / \$100,000 \times 30 = 19.5$$

**ISBE Participation Evaluation:

A. Calculation of ISBE Participation Rate

1. ISBE Participation Rate for Non-ISBE Vendors. The ISBE participation rate for non-ISBE vendors shall be expressed as a percentage and shall be calculated by dividing the amount of non-ISBE vendor's total contract price that will be subcontracted to ISBEs by the non-ISBE vendor's total contract price. For example, if the non-ISBE's total contract price is \$100,000.00 and it subcontracts a total of \$12,000.00 to ISBEs, the non-ISBE's ISBE participation rate would be 12%.
2. ISBE Participation Rate for ISBE Vendors. The ISBE participation rate for ISBE vendors shall be expressed as a percentage and shall be calculated by dividing the amount of the ISBE vendor's total contract price that will be subcontracted to ISBEs and the amount that will be self-performed by the ISBE vendor by the ISBE vendor's total contract price. For example, if the ISBE vendor's total contract price is \$100,000.00 and it subcontracts a total of \$12,000.00 to ISBEs and will perform a total of \$8,000.00 of the work itself, the ISBE vendor's ISBE participation rate would be 20%.

B. Points for ISBE Participation Rate:

The vendor with the highest ISBE participation rate shall receive the maximum ISBE participation points. All other vendors shall receive ISBE participation points by applying the following formula:

$$\begin{aligned} & (\text{Vendor's ISBE participation rate} \div \text{Highest ISBE participation rate} \\ & \quad \times \text{Maximum ISBE participation points}) \end{aligned}$$

For example, assuming the weight given by the RFP to ISBE participation is 6 points, if Vendor A has the highest ISBE participation rate at 20% and Vendor B's ISBE participation rate is 12%, Vendor A will receive the maximum 6 points and Vendor B will receive $(12\% \div 20\%) \times 6$ which equals 3.6 points.

General Evaluation:

Points shall be assigned based on the vendor's clear demonstration of the ability to provide the requested goods and/or services. Vendors may be required to submit additional written information or be asked to make an oral presentation before the TEC to clarify statements made in the proposal.

SECTION 6: QUESTIONS

Questions concerning this solicitation must be e-mailed to the Division of Purchases at doa.purquestions15@purchasing.ri.gov no later than the date and time indicated on page one of this solicitation. No other contact with State parties is permitted. Please reference **RFP #7664822** on all correspondence. Questions should be submitted in writing in a Microsoft Word attachment in a narrative format with no tables. Answers to questions received, if any, shall be posted on the Division of Purchases' website as an addendum to this solicitation. It is the responsibility of all interested parties to monitor the Division of Purchases website for any procurement related postings such as addenda. If technical assistance is required, call the Help Desk at (401) 574-8100.

SECTION 7: PROPOSAL CONTENTS

A. Proposals shall include the following:

1. One completed and signed RIVIP Vendor Certification Cover Form (included in the original copy only) downloaded from the Division of Purchases website at www.ridop.ri.gov. *Do not include any copies in the Technical or Cost proposals.*
2. Two (2) completed original and copy versions, signed and sealed Appendix A. MBE, WBE, and/or Disability Business Enterprise Participation Plan. Please complete separate forms for each MBE, WBE or Disability Business Enterprise subcontractor/vendor to be utilized on the solicitation. *Do not include any copies in the Technical or Cost proposals.*
3. Technical Proposal - describing the qualifications and background of the applicant and experience with and for similar projects, and all information described earlier in this solicitation. The technical proposal is limited to twelve (12) pages (this excludes any appendices and as appropriate, resumes of key staff that will provide services covered by this request).
 - a. One (1) Electronic copy on a CD-R, marked "Technical Proposal - Original".
 - b. One (1) printed paper copy, marked "Technical Proposal -Original" and signed.
 - c. Four (4) printed paper copies
4. Higher Education Cloud Vendor Assessment Tool (HECVAT)
 - a. One (1) Electronic copy on a CD-R, marked "HECVAT"
 - b. One (1) printed paper copy
5. Cost Proposal - A separate, signed and sealed cost proposal reflecting the hourly rate, or other fee structure, proposed to complete all of the requirements of this project.
 - a. One (1) Electronic copy on a CD-R, marked "Cost Proposal -Original".
 - b. One (1) printed paper copy, marked "Cost Proposal -Original" and signed.
 - c. Four (4) printed paper copies

B. Formatting of proposal response contents should consist of the following:

1. Formatting of CD-Rs – Separate CD-Rs are required for the technical proposal and cost proposal. All CD-Rs submitted must be labeled with:
 - a. Vendor's name
 - b. RFP #

- c. RFP Title
- d. Proposal type (e.g., technical proposal or cost proposal)
- e. If file sizes require more than one CD-R, multiple CD-Rs are acceptable.

Each CD-R must include the above labeling and additional labeling of how many CD-Rs should be accounted for (e.g., 3 CD-Rs are submitted for a technical proposal and each CD-R should have additional label of '1 of 3' on first CD-R, '2 of 3' on second CD-R, '3 of 3' on third CD-R).

Vendors are responsible for testing their CD-Rs before submission as the Division of Purchase's inability to open or read a CD-R may be grounds for rejection of a Vendor's proposal. All files should be readable and readily accessible on the CD-Rs submitted with no instructions to download files from any external resource(s). If a file is partial, corrupt or unreadable, the Division of Purchases may consider it "non-responsive". USB Drives or any other electronic media shall not be accepted. Please note that CD-Rs submitted, shall not be returned.

- 2. Formatting of written documents and printed copies:
 - a. For clarity, the technical proposal shall be typed. These documents shall be single-spaced with 1" margins on white 8.5"x 11" paper using a font of 12-point Calibri or 12-point Times New Roman.
 - b. All pages on the technical proposal are to be sequentially numbered in the footer, starting with number 1 on the first page of the narrative (this does not include the cover page or table of contents) through to the end, including all forms and attachments. The Vendor's name should appear on every page, including attachments. Each attachment should be referenced appropriately within the proposal section and the attachment title should reference the proposal section it is applicable to.
 - c. If the solicitation includes a proposal template for vendor use, it shall be typed using the formatting provided in the template.
 - d. Printed copies are to be only bound with removable binder clips.

SECTION 8: PROPOSAL SUBMISSION

Interested vendors must submit proposals to provide the goods and/or services covered by this RFP on or before the date and time listed on the cover page of this solicitation. Responses received after this date and time, as registered by the official time clock in the reception area of the Division of Purchases, shall not be accepted.

Proposals should be mailed or hand-delivered in a sealed envelope marked **RFP #7664822** to:

RI Dept. of Administration
Division of Purchases, 2nd floor
One Capitol Hill
Providence, RI 02908-5855

NOTE: Proposals received after the above-referenced due date and time shall not be accepted. Proposals misdirected to other State locations or those not presented to the Division of Purchases by the scheduled due date and time shall be determined to be late and shall not be accepted. Proposals faxed, or emailed, to the Division of Purchases shall not be accepted. The official time clock is in the reception area of the Division of Purchases.

SECTION 9: CONCLUDING STATEMENTS

Notwithstanding the above, the Division of Purchases reserves the right to award on the basis of cost alone, to accept or reject any or all proposals, and to award in the State's best interest.

Proposals found to be technically or substantially non-responsive at any point in the evaluation process will be rejected and not considered further.

If a Vendor is selected for an award, no work is to commence until a purchase order is issued by the Division of Purchases.

The State's General Conditions of Purchase shall be the contractual terms and conditions between the parties upon issuance of a Purchase Order by the Division of Purchases. The State's General Conditions of Purchase can be found at <https://rules.sos.ri.gov/regulations/part/220-30-00-13> and addenda can be found at <https://ridop.ri.gov/rules-regulations/>.

APPENDIX A. PROPOSER ISBE RESPONSIBILITIES AND MBE, WBE, AND/OR DISABILITY BUSINESS ENTERPRISE PARTICIPATION FORM

A. Proposer's ISBE Responsibilities (from 150-RICR-90-10-1.7.E)

1. Proposal of ISBE Participation Rate. Unless otherwise indicated in the RFP, a Proposer must submit its proposed ISBE Participation Rate in a sealed envelope or via sealed electronic submission at the time it submits its proposed total contract price. The Proposer shall be responsible for completing and submitting all standard forms adopted pursuant to 105-RICR-90-10-1.9 and submitting all substantiating documentation as reasonably requested by either the Using Agency's MBE/WBE Coordinator, Division, ODEO, or Governor's Commission on Disabilities including but not limited to the names and contact information of all proposed subcontractors and the dollar amounts that correspond with each proposed subcontract.
2. Failure to Submit ISBE Participation Rate. Any Proposer that fails to submit a proposed ISBE Participation Rate or any requested substantiating documentation in a timely manner shall receive zero (0) ISBE participation points.
3. Execution of Proposed ISBE Participation Rate. Proposers shall be evaluated and scored based on the amounts and rates submitted in their proposals. If awarded the contract, Proposers shall be required to achieve their proposed ISBE Participation Rates. During the life of the contract, the Proposer shall be responsible for submitting all substantiating documentation as reasonably requested by the Using Agency's MBE/WBE Coordinator, Division, ODEO, or Governor's Commission on Disabilities including but not limited to copies of purchase orders, subcontracts, and cancelled checks.
4. Change Orders. If during the life of the contract, a change order is issued by the Division, the Proposer shall notify the ODEO of the change as soon as reasonably possible. Proposers are required to achieve their proposed ISBE Participation Rates on any change order amounts.
5. Notice of Change to Proposed ISBE Participation Rate. If during the life of the contract, the Proposer becomes aware that it will be unable to achieve its proposed ISBE Participation Rate, it must notify the Division and ODEO as soon as reasonably possible. The Division, in consultation with ODEO and Governor's Commission on Disabilities, and the Proposer may agree to a modified ISBE Participation Rate provided that the change in circumstances was beyond the control of the Proposer or the direct result of an unanticipated reduction in the overall total project cost.

B. MBE, WBE, AND/OR Disability Business Enterprise Participation Plan Form:

Attached is the MBE, WBE, and/or Disability Business Enterprise Participation Plan form. Vendors are required to complete, sign and submit with their overall proposal in a sealed envelope. Please complete separate forms for each MBE, WBE and/or Disability Business Enterprise subcontractor/supplier to be utilized on the solicitation.

MBE, WBE, and/or DISABILITY BUSINESS ENTERPRISE PARTICIPATION PLAN

Vendor's Name:

Vendor's Address:

Point of Contact:

Telephone:

Email:

Solicitation No.:

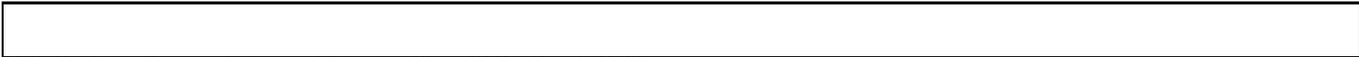
Project Name:

This form is intended to capture commitments between the prime contractor/vendor and MBE/WBE and/or Disability Business Enterprise subcontractors and suppliers, including a description of the work to be performed and the percentage of the work as submitted to the prime contractor/vendor. Please note that all MBE/WBE subcontractors/suppliers must be certified by the Office of Diversity, Equity and Opportunity MBE Compliance Office and all Disability Business Enterprises must be certified by the Governor's Commission on Disabilities at time of bid, and that MBE/WBE and Disability Business Enterprise subcontractors must self-perform 100% of the work or subcontract to another RI certified MBE in order to receive participation credit. Vendors may count 60% of expenditures for materials and supplies obtained from an MBE certified as a regular dealer/supplier, and 100% of such expenditures obtained from an MBE certified as a manufacturer. This form must be completed in its entirety and submitted at time of bid. **Please complete separate forms for each MBE/WBE or Disability Business Enterprise subcontractor/supplier to be utilized on the solicitation.**

Name of Subcontractor/Supplier:					
Type of RI Certification:	<input type="checkbox"/> MBE <input type="checkbox"/> WBE <input type="checkbox"/> Disability Business Enterprise				
Address:					
Point of Contact:					
Telephone:					
Email:					
Detailed Description of Work To Be Performed by Subcontractor or Materials to be Supplied by Supplier:					
Total Contract Value (\$):		Subcontract Value (\$):		ISBE Participation Rate (%):	
Anticipated Date of Performance:					

I certify under penalty of perjury that the forgoing statements are true and correct.

Prime Contractor/Vendor Signature		Title	Date
Subcontractor/Supplier Signature		Title	Date



WORKLOAD REPOSITORY REPORT (RAC)

Database Summary

Id	Database		Snapshot Ids		Number of Instances		Number of Hosts		Report Total (minutes)		
	Name	RAC	Block Size	Begin	End	In Report	Total	In Report	Total	DB time	Elapsed time
2413530610	SAHRPROD	YES	8192	61360	61361	3	3	3	3	89.62	60.12

Database Instances Included In Report

- Listed in order of instance number, I#

I#	Instance	Host	Startup	Begin Snap Time	End Snap Time	Release	Elapsed Time(min)	DB time(min)	Up Time(hrs)	Avg Active Sessions	Platform
1	sahrprod1	dbprd1.rac.uri.edu	23-Feb-21 07:26	23-Mar-21 10:00	23-Mar-21 11:00	12.1.0.2.0	60.13	30.76	675.56	0.51	Linux x86 64-bit
2	sahrprod2	dbprd2.rac.uri.edu	23-Feb-21 08:14	23-Mar-21 10:00	23-Mar-21 11:00	12.1.0.2.0	60.13	14.47	674.76	0.24	Linux x86 64-bit
3	sahrprod3	dbprd3.rac.uri.edu	23-Feb-21 08:57	23-Mar-21 10:00	23-Mar-21 11:00	12.1.0.2.0	60.13	44.39	674.05	0.74	Linux x86 64-bit

Report Summary

Top ADDM Findings by Average Active Sessions

Finding Name	Avg active sessions of the task	Percent active sessions of finding	Task Name	Begin Snap Time	End Snap Time
Top SQL Statements	1.49	41.39	ADDM:2413530610_61361	23-Mar-21 10:00	23-Mar-21 11:00
"User I/O" wait Class	1.49	28.12	ADDM:2413530610_61361	23-Mar-21 10:00	23-Mar-21 11:00
Top Segments by "User I/O" and "Cluster"	1.49	18.62	ADDM:2413530610_61361	23-Mar-21 10:00	23-Mar-21 11:00
Hard Parse	1.49	6.08	ADDM:2413530610_61361	23-Mar-21 10:00	23-Mar-21 11:00
Global Cache Messaging	1.49	5.46	ADDM:2413530610_61361	23-Mar-21 10:00	23-Mar-21 11:00

Cache Sizes

- All values are in Megabytes
- Listed in order of instance number, I#
- End values displayed only if different from Begin values

I#	Memory Target		Sga Target		DB Cache		Shared Pool		Large Pool		Java Pool		Streams Pool		PGA Target	
	Begin	End	Begin	End	Begin	End	Begin	End	Begin	End	Begin	End	Begin	End	Begin	End
1	12,800	8,320	5,760	2,336	128	64									4,480	27.48
2	12,800	8,320	5,664	2,432	64	64							64		4,480	27.48
3	12,800	8,320	5,888	2,272	64	64									4,480	27.48
Avg	12,800	8,320	5,771	2,347	85	64							64		4,480	27.48
Min	12,800	8,320	5,664	2,272	64	64							64		4,480	27.48
Max	12,800	8,320	5,888	2,432	128	64							64		4,480	27.48

Main Report

- [Report Summary](#)
- [OS Statistics](#)
- [Time Model Statistics](#)
- [Wait Events Statistics](#)
- [Global Activity Load Profile](#)
- [Global Cache and Enqueue Statistics Summary](#)
- [Global CR Served Stats](#)

- [Global CURRENT Served Stats](#)
- [Global Cache Transfer Stats](#)
- [Interconnect Stats](#)
- [SQL Statistics](#)
- [Global Activity Statistics](#)
- [I/O Statistics](#)
- [Library Cache Statistics](#)
- [Memory Statistics](#)
- [Supplemental Information](#)
- [Active Session History\(ASH\) Report](#)
- [ADDM Reports](#)

[Back to Top](#)

OS Statistics By Instance

- Listed in order of instance number, I#
- End values are displayed only if different from begin values

I#	Num CPUs	CPU Cores	CPU Sckts	Load Begin	Load End	% Busy	% Usr	% Sys	% WIO	% Idl	Busy Time (s)	Idle Time (s)	Total Time (s)	Memory (M)	End CPUs	End Cores	End Sckts	End Memory (M)
1	8	8	2	3.19	4.47	38.86	30.07	7.75	2.87	61.14	11,200.61	17,623.78	28,824.39	258,430.20				
2	8	8	2	2.42	1.18	35.40	26.77	7.98	3.04	64.60	10,133.11	18,490.49	28,623.60	258,430.20				
3	8	8	2	2.09	2.33	27.71	18.34	8.39	5.05	72.29	7,919.36	20,662.12	28,581.48	258,430.20				
Sum											29,253.08	56,776.39	86,029.47					

[Back to Top](#)

Time Model Statistics

- [Time Model](#)
- [Time Model - % of DB time](#)

[Back to Top](#)

Time Model

I#	DB time (s)	DB CPU (s)	SQL Exec Ela (s)	Parse Ela (s)	Hard Parse Ela (s)	PL/SQL Ela (s)	Java Ela (s)	bg time (s)	bg CPU (s)
1	1,845.60	1,526.18	1,628.07	94.19	77.04	19.52	0.00	227.98	209.84
2	868.44	696.38	618.79	137.06	120.61	0.37	0.00	165.87	172.31
3	2,663.45	1,493.61	2,315.23	151.94	129.18	5.14	0.92	225.72	222.62
Sum	5,377.49	3,716.17	4,562.09	383.19	326.83	25.02	0.92	619.57	604.77
Avg	1,792.50	1,238.72	1,520.70	127.73	108.94	8.34	0.31	206.52	201.59
Std	898.68	469.96	853.30	29.98	27.96	9.97	0.53	35.23	26.15

[Back to Time Model Statistics](#)

[Back to Top](#)

Time Model - % of DB time

- % Total [DB time/bg time] - instance [DB time/bg time] as a percentage of the cluster-wide total [DB time/bg time]

I#	% Total DB time	DB CPU %DB time	SQL Exec Ela %DB time	Parse Ela %DB time	Hard Parse %DB time	PL/SQL Ela %DB time	Java Ela %DB time	% Total bg time	bg CPU %bg time
1	34.32	82.69	88.21	5.10	4.17	0.00	0.00	36.80	92.04
2	16.15	80.19	71.25	15.78	13.89	0.00	0.00	26.77	103.88
3	49.53	56.08	86.93	5.70	4.85	0.00	0.03	36.43	98.63
Avg	33.33	72.99	82.13	8.86	7.64	0.00	0.01	33.33	98.18

[Back to Time Model Statistics](#)

[Back to Top](#)

Wait Events Statistics

- [Foreground Wait Classes - % of Total DB time](#)
- [Foreground Wait Classes](#)
- [Foreground Wait Classes - % of DB time](#)
- [Top Timed Events](#)
- [Top Timed Foreground Events](#)

- [Top Timed Background Events](#)

[Back to Top](#)

Foreground Wait Classes - % of Total DB time

- Cluster-wide totals of wait class foreground wait times as a percentage of the cluster-wide DB time

User I/O Sys I/O Other Applic Commit Network Concurcy Config Cluster DB CPU
 28.12 0.10 1.63 0.65 0.64 2.53 0.27 0.06 5.47 69.11

[Back to Wait Events Statistics](#)

[Back to Top](#)

Foreground Wait Classes

#	User I/O(s)	Sys I/O(s)	Other(s)	Applic (s)	Commit (s)	Network (s)	Concurcy (s)	Config (s)	Cluster (s)	DB CPU (s)	DB time
1	141.21	1.85	39.53	18.38	7.58	107.80	8.35	0.20	77.41	1,526.18	1,845.60
2	125.55	1.43	16.39	8.72	11.81	9.23	2.76	0.18	82.04	696.38	868.44
3	1,245.51	1.89	31.74	7.81	14.92	19.18	3.21	2.89	134.60	1,493.61	2,663.45
Sum	1,512.26	5.17	87.66	34.90	34.32	136.21	14.33	3.28	294.05	3,716.17	5,377.49
Avg	504.09	1.72	29.22	11.63	11.44	45.40	4.78	1.09	98.02	1,238.72	1,792.50
Std	642.14	0.26	11.77	5.86	3.68	54.26	3.11	1.56	31.77	469.96	898.68

[Back to Wait Events Statistics](#)

[Back to Top](#)

Foreground Wait Classes - % of DB time

- % of Total DB time - instance DB time as a percentage of the cluster-wide total DB time

#	User I/O	Sys I/O	Other	Applic	Commit	Network	Concurcy	Config	Cluster	DB CPU	% Total DB time
1	7.65	0.10	2.14	1.00	0.41	5.84	0.45	0.01	4.19	82.69	34.32
2	14.46	0.16	1.89	1.00	1.36	1.06	0.32	0.02	9.45	80.19	16.15
3	46.76	0.07	1.19	0.29	0.56	0.72	0.12	0.11	5.05	56.08	49.53
Avg	22.96	0.11	1.74	0.76	0.78	2.54	0.30	0.05	6.23	72.99	

[Back to Wait Events Statistics](#)

[Back to Top](#)

Top Timed Events

- Instance '**' - cluster wide summary
- '**' Waits, %Timeouts, Wait Time Total(s) : Cluster-wide total for the wait event
- '**' 'Wait Time Avg (ms)' : Cluster-wide average computed as (Wait Time Total / Event Waits) in ms
- '**' Summary 'Avg Wait Time (ms)' : Per-instance 'Wait Time Avg (ms)' used to compute the following statistics
- '**' [Avg/Min/Max/Std Dev] : average/minimum/maximum/standard deviation of per-instance 'Wait Time Avg(ms)'
- '**' Cnt : count of instances with wait times for the event

#	Class	Wait Event	Event		Wait Time			Summary Avg Wait Time (ms)					
			Waits	%Timeouts	Total(s)	Avg(ms)	%DB time	Avg	Min	Max	Std Dev	Cnt	
*		DB CPU			3,716.17			69.11					3
*	User I/O	db file sequential read	4,948,843	0.00	1,258.96	0.25	23.41	0.49	0.23	0.66	0.23	3	
*	User I/O	read by other session	680,235	0.00	128.25	0.19	2.38	0.70	0.19	1.53	0.73	3	
*	Network	SQL*Net message from dblink	419,366	0.00	101.84	0.24	1.89	0.59	0.24	0.95	0.35	3	
*	Cluster	gc current block 2-way	328,295	0.00	101.52	0.31	1.89	0.31	0.30	0.31	0.01	3	
*	User I/O	direct path read	90,466	0.00	68.08	0.75	1.27	0.75	0.67	0.86	0.09	3	
*	System I/O	log file parallel write	93,117	0.00	64.51	0.69	1.20	0.73	0.61	0.80	0.11	3	
*	Cluster	gc current block 3-way	114,198	0.00	51.64	0.45	0.96	0.45	0.45	0.45	0.00	3	
*	System I/O	control file sequential read	69,314	0.00	39.31	0.57	0.73	0.57	0.50	0.69	0.11	3	
*	User I/O	db file parallel read	29,840	0.00	35.14	1.18	0.65	1.57	0.98	2.71	0.99	3	
1		DB CPU			1,526.18			82.69					
1	Network	SQL*Net message from dblink	418,808	0.00	101.51	0.24	5.50						
1	User I/O	db file sequential read	170,422	0.00	99.21	0.58	5.38						
1	System I/O	log file parallel write	50,485	0.00	30.75	0.61	1.67						
1	Cluster	gc current block 2-way	96,570	0.00	30.40	0.31	1.65						
1	User I/O	direct path read	31,415	0.00	26.86	0.86	1.46						
1	Concurrency	library cache pin	98,931	0.00	22.76	0.23	1.23						
1	Other	PX Deq: reap credit	4,670,906	99.99	20.56	0.00	1.11						
1	Application	enq: TX - row lock contention	13	0.00	18.15	1396.00	0.98						

1 Cluster	gc current block 3-way	31,017	0.00	14.06	0.45	0.76
2	DB CPU			696.38		80.19
2 User I/O	db file sequential read	137,241	0.00	90.97	0.66	10.47
2 Cluster	gc current block 2-way	80,476	0.00	24.22	0.30	2.79
2 User I/O	direct path read	25,985	0.00	17.53	0.67	2.02
2 System I/O	control file sequential read	21,340	0.00	14.82	0.69	1.71
2 System I/O	log file parallel write	18,339	0.00	14.40	0.79	1.66
2 Cluster	gc current block 3-way	28,728	0.00	12.86	0.45	1.48
2 Cluster	gc current grant busy	34,829	0.00	11.95	0.34	1.38
2 Commit	log file sync	11,596	0.00	11.83	1.02	1.36
2 User I/O	db file parallel read	9,648	0.00	9.73	1.01	1.12
3	DB CPU			1,493.61		56.08
3 User I/O	db file sequential read	4,641,180	0.00	1,068.79	0.23	40.13
3 User I/O	read by other session	680,191	0.00	128.18	0.19	4.81
3 Cluster	gc current block 2-way	151,249	0.00	46.90	0.31	1.76
3 Cluster	gc current block 3-way	54,453	0.00	24.71	0.45	0.93
3 User I/O	direct path read	33,066	0.00	23.69	0.72	0.89
3 System I/O	log file parallel write	24,293	0.00	19.37	0.80	0.73
3 Cluster	gc cr multi block request	28,567	0.00	16.77	0.59	0.63
3 User I/O	db file parallel read	16,936	0.00	16.60	0.98	0.62
3 Network	SQL*Net more data to client	182,954	0.00	15.70	0.09	0.59

[Back to Wait Events Statistics](#)

[Back to Top](#)

Top Timed Foreground Events

- Foreground Activity is captured in release 11g and above
- For prior releases, Foreground Activity is computed as difference between Total Activity and Background Activity
- Instance '**' - cluster wide summary
- '**' Waits, %Timeouts, Wait Time Total(s) : Cluster-wide total for the wait event
- '**' 'Wait Time Avg (ms)' : Cluster-wide average computed as (Wait Time Total / Event Waits) in ms
- '**' Summary 'Avg Wait Time (ms)' : Per-instance 'Wait Time Avg (ms)' used to compute the following statistics
- '**' [Avg/Min/Max/Std Dev] : average/minimum/maximum/standard deviation of per-instance 'Wait Time Avg(ms)'
- '**' Cnt : count of instances with wait times for the event

#	Class	Wait Event	Event		Wait Time			Summary Avg Wait Time (ms)					
			Waits	%Timeouts	Total(s)	Avg(ms)	%DB time	Avg	Min	Max	Std Dev	Cnt	
*		DB CPU			3,716.17		69.11						3
* User I/O		db file sequential read	4,937,813	0.00	1,253.45	0.25	23.31	0.49	0.23	0.67	0.23	3	
* User I/O		read by other session	680,234	0.00	128.25	0.19	2.38	0.70	0.19	1.53	0.73	3	
* Network		SQL*Net message from dblink	419,366	0.00	101.84	0.24	1.89	0.59	0.24	0.95	0.35	3	
* Cluster		gc current block 2-way	324,287	0.00	100.29	0.31	1.87	0.31	0.30	0.32	0.01	3	
* User I/O		direct path read	90,394	0.00	68.00	0.75	1.26	0.75	0.67	0.85	0.09	3	
* Cluster		gc current block 3-way	112,535	0.00	50.94	0.45	0.95	0.45	0.45	0.45	0.00	3	
* User I/O		db file parallel read	29,566	0.00	34.58	1.17	0.64	1.57	0.96	2.73	1.01	3	
* Commit		log file sync	35,616	0.00	34.32	0.96	0.64	0.98	0.90	1.02	0.07	3	
* Application		enq: TX - row lock contention	36	0.00	34.30	952.70	0.64	944.96	586.79	1396.00	412.52	3	
1		DB CPU			1,526.18		82.69						
1 Network		SQL*Net message from dblink	418,808	0.00	101.51	0.24	5.50						
1 User I/O		db file sequential read	166,864	0.00	97.56	0.58	5.29						
1 Cluster		gc current block 2-way	95,755	0.00	30.16	0.32	1.63						
1 User I/O		direct path read	31,387	0.00	26.83	0.85	1.45						
1 Other		PX Deg: reap credit	4,541,484	99.99	19.79	0.00	1.07						
1 Application		enq: TX - row lock contention	13	0.00	18.15	1396.00	0.98						
1 Cluster		gc current block 3-way	30,506	0.00	13.84	0.45	0.75						
1 Cluster		gc cr grant 2-way	54,104	0.00	11.10	0.21	0.60						
1 User I/O		db file parallel read	3,220	0.00	8.79	2.73	0.48						
2		DB CPU			696.38		80.19						
2 User I/O		db file sequential read	134,521	0.00	89.80	0.67	10.34						
2 Cluster		gc current block 2-way	79,356	0.00	23.90	0.30	2.75						
2 User I/O		direct path read	25,952	0.00	17.48	0.67	2.01						
2 Cluster		gc current block 3-way	28,132	0.00	12.64	0.45	1.46						
2 Cluster		gc current grant busy	34,645	0.00	11.88	0.34	1.37						
2 Commit		log file sync	11,565	0.00	11.81	1.02	1.36						
2 User I/O		db file parallel read	9,645	0.00	9.73	1.01	1.12						
2 Cluster		gc cr multi block request	18,072	0.00	9.50	0.53	1.09						
2 Cluster		gc cr grant 2-way	39,713	0.00	8.77	0.22	1.01						

3	DB CPU		1,493.61	56.08		
3 User I/O	db file sequential read	4,636,428	0.00 1,066.09	0.23	40.03	
3 User I/O	read by other session	680,190	0.00 128.18	0.19	4.81	
3 Cluster	gc current block 2-way	149,176	0.00 46.23	0.31	1.74	
3 Cluster	gc current block 3-way	53,897	0.00 24.46	0.45	0.92	
3 User I/O	direct path read	33,055	0.00 23.68	0.72	0.89	
3 User I/O	db file parallel read	16,701	0.00 16.07	0.96	0.60	
3 Network	SQL*Net more data to client	182,954	0.00 15.70	0.09	0.59	
3 Commit	log file sync	16,563	0.00 14.92	0.90	0.56	
3 Cluster	gc cr multi block request	27,109	0.00 14.11	0.52	0.53	

[Back to Wait Events Statistics](#)
[Back to Top](#)

Top Timed Background Events

- % of bg time: % of background elapsed time
- Instance '*' - cluster wide summary
- '**' Waits, %Timeouts, Wait Time Total(s) : Cluster-wide total for the wait event
- '**' 'Wait Time Avg (ms)' : Cluster-wide average computed as (Wait Time Total / Event Waits) in ms
- '**' Summary 'Avg Wait Time (ms)' : Per-instance 'Wait Time Avg (ms)' used to compute the following statistics
- '**' [Avg/Min/Max/Std Dev] : average/minimum/maximum/standard deviation of per-instance 'Wait Time Avg(ms)'
- '**' Cnt : count of instances with wait times for the event

#	Class	Wait Event	Event		Wait Time			Summary Avg Wait Time (ms)					
			Waits	%Timeouts	Total(s)	Avg(ms)	%DB time	Avg	Min	Max	Std Dev	Cnt	
*		background cpu time			604.77		97.61						3
*	System I/O	log file parallel write	93,123	0.00	64.52	0.69	10.41	0.73	0.61	0.80	0.11	3	
*	System I/O	control file sequential read	54,154	0.00	34.14	0.63	5.51	0.63	0.56	0.78	0.12	3	
*	Concurrency	library cache pin	90,346	0.00	20.26	0.22	3.27	0.26	0.22	0.31	0.05	3	
*	Other	oracle thread bootstrap	566	0.00	13.91	24.58	2.25	24.47	22.30	27.29	2.56	3	
*	System I/O	log file sequential read	2,167	0.00	8.17	3.77	1.32	3.84	3.36	4.39	0.52	3	
*	System I/O	db file parallel write	26,046	0.00	7.64	0.29	1.23	0.29	0.24	0.32	0.04	3	
*	Other	DFS lock handle	13,820	99.94	6.22	0.45	1.00	0.38	0.30	0.46	0.08	3	
*	Other	reliable message	4,842	0.00	6.17	1.27	1.00	1.31	0.94	1.50	0.32	3	
*	Other	target log write size	7,013	0.54	6.04	0.86	0.98	0.86	0.79	0.97	0.10	3	
1		background cpu time			209.84		92.04						
1	System I/O	log file parallel write	50,485	0.00	30.75	0.61	13.49						
1	Concurrency	library cache pin	89,033	0.00	19.91	0.22	8.73						
1	System I/O	control file sequential read	18,934	0.00	10.63	0.56	4.66						
1	Other	oracle thread bootstrap	203	0.00	5.54	27.29	2.43						
1	System I/O	log file sequential read	607	0.00	2.28	3.76	1.00						
1	Other	PX Deq: Slave Join Frag	9,094	0.00	2.18	0.24	0.96						
1	Other	target log write size	2,595	0.66	2.04	0.79	0.90						
1	System I/O	db file parallel write	6,031	0.00	1.89	0.31	0.83						
1	Other	reliable message	1,911	0.00	1.79	0.94	0.79						
2		background cpu time			172.31		103.88						
2	System I/O	log file parallel write	18,343	0.00	14.40	0.78	8.68						
2	System I/O	control file sequential read	17,270	0.00	13.39	0.78	8.07						
2	Other	oracle thread bootstrap	182	0.00	4.33	23.82	2.61						
2	System I/O	db file parallel write	11,869	0.00	3.80	0.32	2.29						
2	System I/O	log file sequential read	629	0.00	2.76	4.39	1.67						
2	Other	reliable message	1,524	0.00	2.29	1.50	1.38						
2	System I/O	control file parallel write	2,525	0.00	1.71	0.68	1.03						
2	Application	enq: RO - fast object reuse	4,406	0.00	1.69	0.38	1.02						
2	Other	target log write size	1,904	0.16	1.55	0.81	0.94						
3		background cpu time			222.62		98.63						
3	System I/O	log file parallel write	24,295	0.00	19.37	0.80	8.58						
3	System I/O	control file sequential read	17,950	0.00	10.13	0.56	4.49						
3	Other	DFS lock handle	12,951	99.96	5.93	0.46	2.63						
3	Other	oracle thread bootstrap	181	0.00	4.04	22.30	1.79						
3	System I/O	log file sequential read	931	0.00	3.13	3.36	1.39						
3	User I/O	db file sequential read	4,761	0.00	2.71	0.57	1.20						
3	Cluster	gc cr multi block request	1,458	0.00	2.66	1.82	1.18						
3	Other	target log write size	2,514	0.72	2.45	0.97	1.08						
3	Other	reliable message	1,407	0.00	2.09	1.48	0.92						

[Back to Wait Events Statistics](#)
[Back to Top](#)

Global Activity Load Profile

- [System Statistics](#)
- [System Statistics - Per Second](#)
- [System Statistics - Per Transaction](#)
- [SysStat and Global Messaging - RAC](#)
- [SysStat and Global Messaging \(per Sec\)- RAC](#)
- [SysStat and Global Messaging \(per Tx\)- RAC](#)

[Back to Top](#)

System Statistics

#	Logical Reads	Physical Reads	Physical Writes	Redo Size (k)	Block Changes	User Calls	Execs	Parses	Logons	Txns
1	166,901,597	1,883,173	129,729	855,388	4,995,343	3,651,360	6,233,617	526,009	10,043	44,472
2	45,785,358	715,924	113,635	1,042,478	5,695,560	3,826,582	2,763,266	572,871	3,507	11,324
3	110,542,822	6,235,457	147,615	1,396,290	8,576,586	6,547,142	5,370,124	837,929	3,417	16,236
Sum	323,229,777	8,834,554	390,979	3,294,157	19,267,489	14,025,084	14,367,007	1,936,809	16,967	72,032
Avg	107,743,259	2,944,851	130,326	1,098,052	6,422,496	4,675,028	4,789,002	645,603	5,656	24,011
Std	60,606,633	2,908,897	16,998	274,700	1,898,066	1,623,664	1,806,685	168,199	3,800	17,889

[Back to Global Activity Load Profile](#)
[Back to Top](#)

System Statistics - Per Second

#	Logical Reads/s	Physical Reads/s	Physical Writes/s	Redo Size (k)/s	Block Changes/s	User Calls/s	Execs/s	Parses/s	Logons/s	Txns/s
1	46,266.16	522.03	35.96	237.12	1,384.74	1,012.18	1,728.00	145.81	2.78	12.33
2	12,691.87	198.46	31.50	288.98	1,578.83	1,060.74	765.99	158.80	0.97	3.14
3	30,643.01	1,728.50	40.92	387.06	2,377.47	1,814.90	1,488.62	232.28	0.95	4.50
Sum	89,601.04	2,448.98	108.38	913.16	5,341.04	3,887.82	3,982.61	536.89	4.70	19.97
Avg	29,867.01	816.33	36.13	304.39	1,780.35	1,295.94	1,327.54	178.96	1.57	6.66
Std	16,800.59	806.36	4.71	76.15	526.15	450.09	500.83	46.63	1.05	4.96

[Back to Global Activity Load Profile](#)
[Back to Top](#)

System Statistics - Per Transaction

#	Logical Reads/tx	Physical Reads/tx	Physical Writes/tx	Redo Size (k)/tx	Block Changes/tx	User Calls/tx	Execs/tx	Parses/tx	Logons/tx
1	3,752.96	42.35	2.92	19.23	112.33	82.10	140.17	11.83	0.23
2	4,043.21	63.22	10.03	92.06	502.96	337.92	244.02	50.59	0.31
3	6,808.50	384.05	9.09	86.00	528.25	403.25	330.75	51.61	0.21
Avg	4,868.22	163.21	7.35	65.76	381.18	274.42	238.31	38.01	0.25

[Back to Global Activity Load Profile](#)
[Back to Top](#)

SysStat and Global Messaging - RAC

#	Blocks Received		Blocks Served		CPU (seconds)		Messages						GC Blks	GC CR
	GC Current	GC CR	GC Current	GC CR	GC	IPC	GCS Received	GES Received	GCS Sent	GES Sent	Sent Direct	Sent Indir	Lost	Failure
1	155,823	26,363	187,541	22,747	0	0	1,216,033	235,875	1,014,635	326,481	485,762	472,217	0	0
2	162,962	90,476	257,014	100,797	0	0	1,100,157	224,471	977,588	169,742	304,834	429,376	0	0
3	313,405	100,878	187,677	94,191	0	0	1,049,861	240,612	1,366,352	212,086	449,142	632,496	0	0
Sum	632,190	217,717	632,232	217,735	0	0	3,366,051	700,958	3,358,575	708,309	1,239,738	1,534,089	0	0
Avg	210,730	72,572	210,744	72,578	0	0	1,122,017	233,653	1,119,525	236,103	413,246	511,363	0	0
Std	88,991	40,355	40,071	43,281	0	0	85,215	8,297	214,560	81,083	95,656	107,069	0	0

[Back to Global Activity Load Profile](#)
[Back to Top](#)

SysStat and Global Messaging (per Sec)- RAC

#	Blocks Received Per Second		Blocks Served per Second		CPU seconds per Second		Messages Per Second					GC Blks	GC CR	
	GC Current	GC CR	GC Current	GC CR	GC	IPC	GCS Received	GES Received	GCS Sent	GES Sent	Sent Direct	Sent Indir	Lost/s	Fail/s
1	43.20	7.31	51.99	6.31	0.00	0.00	337.09	65.39	281.26	90.50	134.66	130.90	0	0
2	45.17	25.08	71.25	27.94	0.00	0.00	304.97	62.22	270.99	47.05	84.50	119.02	0	0
3	86.88	27.96	52.02	26.11	0.00	0.00	291.03	66.70	378.76	58.79	124.50	175.33	0	0
Sum	175.25	60.35	175.26	60.36	0.00	0.00	933.09	194.31	931.01	196.35	343.66	425.26	0	0
Avg	58.42	20.12	58.42	20.12	0.00	0.00	311.03	64.77	310.34	65.45	114.55	141.75	0	0
Std	24.67	11.19	11.11	12.00	0.00	0.00	23.62	2.30	59.48	22.48	26.52	29.68	0	0

[Back to Global Activity Load Profile](#)
[Back to Top](#)

SysStat and Global Messaging (per Tx)- RAC

#	Blocks Received per Tx		Blocks Served		CPU (seconds) per Tx		Messages per Transaction					GC Blks	GC CR	
	GC Current	GC CR	GC Current	GC CR	GC	IPC	GCS Received	GES Received	GCS Sent	GES Sent	Sent Direct	Sent Indir	Lost/Tx	Fail/Tx
1	3.50	0.59	4.22	0.51	0.00	0.00	27.34	5.30	22.82	7.34	10.92	10.62	0.00	0.00
2	14.39	7.99	22.70	8.90	0.00	0.00	97.15	19.82	86.33	14.99	26.92	37.92	0.00	0.00
3	19.30	6.21	11.56	5.80	0.00	0.00	64.66	14.82	84.16	13.06	27.66	38.96	0.00	0.00
Avg	12.40	4.93	12.82	5.07	0.00	0.00	63.05	13.32	64.43	11.80	21.84	29.16	0.00	0.00

[Back to Global Activity Load Profile](#)
[Back to Top](#)

Global Cache and Enqueue Statistics Summary

- [Global Cache Efficiency Percentages](#)
- [Global Cache and Enqueue Workload Characteristics](#)
- [Global Cache and Enqueue Messaging Statistics](#)

[Back to Top](#)

Global Cache Efficiency Percentages

Buffer Access			
#	Local %	Remote %	Disk %
1	99.74	0.11	0.15
2	98.94	0.56	0.50
3	95.16	0.38	4.46

[Back to Global Cache and Enqueue Statistics Summary](#)
[Back to Top](#)

Global Cache and Enqueue Workload Characteristics

#	GE Get Time (ms)	Receive Time (ms)	CR Blocks				Log Flush CR Srvd %	Receive Time (ms)	Pin Time (ms)	CU Blocks			Log Flush CU Srvd %
			Build Time (ms)	Send Time (ms)	Flush Time (ms)	Send Time (ms)				Flush Time (ms)	Flush Time (ms)		
1	0.03	0.74	0.00	0.00	0.94	1.36	0.47	0.00	0.00	1.26	0.05		
2	0.09	0.70	0.00	0.00	0.91	1.05	0.53	0.00	0.00	1.64	0.05		
3	0.05	0.58	0.00	0.00	1.16	1.70	0.73	0.00	0.00	1.60	0.05		

[Back to Global Cache and Enqueue Statistics Summary](#)
[Back to Top](#)

Global Cache and Enqueue Messaging Statistics

#	Queue Time (ms)		Process Time		% Messages Sent		
	Sent on ksxp	Received	GCS msgs	GES msgs	Direct	Indirect	Flow Ctrl
1	2.46	0.39	0.01	0.01	50.18	48.78	1.04
2	1.44	0.32	0.02	0.01	41.03	57.79	1.18
3	3.22	0.26	0.01	0.01	40.77	57.42	1.81

[Back to Global Cache and Enqueue Statistics Summary](#)
[Back to Top](#)

CR Blocks Served Statistics

#	Block Requests					Results				Fairness		FreeGC		Flushes			Light Total			
	CR	CU	Data	Undo	TX	Current	Private	Zero	Dsk	Rd	Fail	Down	Conv	Clears	Elms	Total	Queued	QFull	MaxTm	Works
1	21,597	1,255	21,597	15	361	22,649	8	94	101	0	10,608	403	0	309	0	0	0	0	9	0
2	99,848	1,175	99,843	117	139	99,829	21	901	261	0	51,638	594	0	1,061	0	0	0	0	135	0
3	93,153	1,032	93,153	7	351	92,884	11	1,280	3	0	44,888	458	0	1,603	0	0	0	0	50	0
Sum	214,598	3,462	214,593	139	851	215,362	40	2,275	365	0	107,134	1,455	0	2,973	0	0	0	0	194	0
Avg	71,533	1,154	71,531	46	284	71,787	13	758	122	0	35,711	485	0	991	0	0	0	0	65	0
Std	43,375	113	43,373	61	125	42,696	7	606	130	0	22,001	98	0	650	0	0	0	0	64	0

[Back to Top](#)

Current Blocks Served Statistics

#	Pins	%					Flushes	%					Writes	%				
		<1ms	<10ms	<100ms	<1s	<10s		<1ms	<10ms	<100ms	<1s	<10s		<1ms	<10ms	<100ms	<1s	<10s
1	3	100.00	0.00	0.00	0.00	0.00	103	65.05	34.95	0.00	0.00	0.00	13,405	30.95	55.77	12.26	0.10	0.93
2	7	85.71	14.29	0.00	0.00	0.00	116	51.72	47.41	0.86	0.00	0.00	20,037	32.69	45.63	21.37	0.00	0.31
3	3	100.00	0.00	0.00	0.00	0.00	94	38.30	61.70	0.00	0.00	0.00	24,845	22.41	41.02	36.57	0.00	0.00
Sum	13						313						58,287					
Avg	4						104						19,429					
Std	2						11						5,744					

[Back to Top](#)

Global Cache Transfer Stats

- [Global Cache Transfer Stats](#)
- [Global Cache Transfer \(Immediate\)](#)

[Back to Top](#)

Global Cache Transfer Stats

- Immediate (Immed) - Block Transfer NOT impacted by Remote Processing Delays
- Busy (Busy) - Block Transfer impacted by Remote Contention
- Congested (Cngst) - Block Transfer impacted by Remote System Load
- All - average time of All blocks (Immed, Busy, Cngst) in ms
- Ordered by instance_number, CR + Current Blocks Received desc

Instance#	Block	CR Blocks					Current Blocks					CR Avg Time (ms)			Current Avg Time (ms)					
		Dest	Src	Class	Received	%Immed	%Busy	%Cngst	Received	%Immed	%Busy	%Cngst	All	Immed	Busy	Cngst	All	Immed	Busy	Cngst
1	3 data block				12,040	93.16	6.75	0.08	87,006	99.84	0.15	0.01	1	1.12	1.53	148.82	0	0.45	0.73	2.82
1	2 data block				13,547	97.39	2.61	0.00	63,885	99.56	0.40	0.03	0	0.36	1.31		1	0.53	0.58	3.49
1	2 others				370	100.00	0.00	0.00	2,505	99.52	0.48	0.00	0	0.35			0	0.35	0.93	
1	3 others				281	98.93	1.07	0.00	2,190	98.49	1.51	0.00	0	0.41	3.17		0	0.33	1.42	
1	3 undo header				82	64.63	35.37	0.00	137	98.54	1.46	0.00	1	0.23	1.10		0	0.34	3.39	
1	2 undo header				36	77.78	22.22	0.00	108	99.07	0.93	0.00	0	0.29	0.84		0	0.25	1.87	
1	2 undo block				2	100.00	0.00	0.00	0				0	0.25						
1	3 undo block				2	100.00	0.00	0.00	0				0	0.36						
2	3 data block				81,128	98.87	0.90	0.22	95,063	99.47	0.39	0.14	1	0.74	1.86	1.80	1	0.62	0.53	3.48
2	1 data block				7,986	97.88	2.12	0.00	61,999	99.62	0.37	0.01	0	0.35	1.30		0	0.43	0.60	2.02
2	3 others				396	98.48	1.52	0.00	3,182	98.74	1.26	0.00	0	0.34	2.77		0	0.27	1.39	
2	1 others				388	98.97	1.03	0.00	2,383	99.45	0.55	0.00	0	0.35	1.55		0	0.26	1.18	
2	1 undo header				298	91.61	8.39	0.00	154	99.35	0.65	0.00	0	0.26	1.26		0	0.25	1.17	
2	3 undo header				273	93.41	6.59	0.00	165	94.55	5.45	0.00	0	0.25	1.42		0	0.30	1.45	
2	1 undo block				15	100.00	0.00	0.00	0				0	0.26						
2	3 undo block				5	100.00	0.00	0.00	0				0	0.25						
3	2 data block				86,317	98.91	0.79	0.30	186,124	99.69	0.21	0.11	1	0.61	1.30	2.30	1	0.85	0.93	4.61
3	1 data block				13,613	99.19	0.81	0.00	119,855	99.80	0.15	0.05	0	0.40	1.20		1	0.58	0.71	3.50
3	2 others				400	98.25	1.75	0.00	4,158	99.30	0.70	0.00	0	0.30	2.19		0	0.27	1.73	
3	1 others				393	99.24	0.76	0.00	3,020	99.04	0.96	0.00	0	0.36	2.59		0	0.31	1.75	
3	1 undo header				75	94.67	5.33	0.00	150	96.67	3.33	0.00	0	0.24	1.57		0	0.35	1.55	

[Back to Global Cache Transfer Stats](#)
[Back to Top](#)

Global Cache Transfer (Immediate)

- Ordered by instance_number, CR + Current Blocks Received desc

Instance#	Block	Lost Blocks	Immediate CR Blocks			Immediate Current Blocks			Immed CR Avg Time (ms)			Immed Current Avg Time (ms)					
			Dest	Src	Class	Count	Time(ms)	Received	%2-hop	%3-hop	Received	%2-hop	%3-hop	Total	2-hop	3-hop	Total
1	3 data block	0				11,217	60.21	39.79	86,863	84.50	15.50	1.12	0.45	2.13	0.45	0.44	0.49
1	2 data block	0				13,194	77.93	22.07	63,606	66.67	33.33	0.36	0.33	0.45	0.53	0.44	0.70
1	2 others	0				370	55.95	44.05	2,493	68.35	31.65	0.35	0.30	0.41	0.35	0.31	0.42
1	3 others	0				278	51.44	48.56	2,157	52.85	47.15	0.41	0.28	0.56	0.33	0.27	0.40
1	3 undo header	0				53	100.00	0.00	135	100.00	0.00	0.23	0.23		0.34	0.34	
1	2 undo header	0				28	100.00	0.00	107	100.00	0.00	0.29	0.29		0.25	0.25	
1	2 undo block	0				2	100.00	0.00	0			0.25	0.25				
1	3 undo block	0				2	100.00	0.00	0			0.36	0.36				
2	3 data block	0				80,214	59.16	40.84	94,560	70.49	29.51	0.74	0.66	0.85	0.62	0.57	0.76
2	1 data block	0				7,817	67.81	32.19	61,763	73.64	26.36	0.35	0.32	0.42	0.43	0.40	0.50
2	3 others	0				390	79.23	20.77	3,142	83.58	16.42	0.34	0.32	0.43	0.27	0.24	0.43
2	1 others	0				384	58.85	41.15	2,370	66.33	33.67	0.35	0.37	0.31	0.26	0.23	0.33
2	1 undo header	0				273	100.00	0.00	153	99.35	0.65	0.26	0.26		0.25	0.25	0.28
2	3 undo header	0				255	100.00	0.00	156	100.00	0.00	0.25	0.25		0.30	0.30	
2	1 undo block	0				15	100.00	0.00	0			0.26	0.26				
2	3 undo block	0				5	100.00	0.00	0			0.25	0.25				
3	2 data block	0				85,378	59.40	40.60	185,542	64.44	35.56	0.61	0.53	0.73	0.85	0.69	1.14
3	1 data block	0				13,503	58.74	41.26	119,612	79.14	20.86	0.40	0.36	0.44	0.58	0.53	0.77
3	2 others	0				393	82.95	17.05	4,129	80.62	19.38	0.30	0.29	0.35	0.27	0.24	0.38
3	1 others	0				390	55.90	44.10	2,991	55.20	44.80	0.36	0.23	0.54	0.31	0.23	0.40
3	1 undo header	0				71	100.00	0.00	145	100.00	0.00	0.24	0.24		0.35	0.35	
3	2 undo header	0				89	100.00	0.00	120	100.00	0.00	0.25	0.25		0.33	0.33	
3	1 undo block	0				0			0								
3	2 undo block	0				0			0								

[Back to Global Cache Transfer Stats](#)
[Back to Top](#)

Interconnect Stats

- [Cluster Interconnect](#)
- [Ping Statistics](#)
- [Interconnect Client Statistics](#)
- [Interconnect Client Statistics \(per Second\)](#)
- [Interconnect Device Statistics](#)
- [Interconnect Device Statistics \(per Second\)](#)
- [Dynamic Remastering Statistics - per Remaster Ops](#)

[Back to Top](#)

Cluster Interconnect

#	NAME	IP Address	Pub Source	IP Address	Pub Source
1	bond1:1	169.254.78.178	N		
2	bond1:1	169.254.53.146	N		
3	bond1:1	169.254.232.32	N		

[Back to Interconnect Stats](#)
[Back to Top](#)

Ping Statistics

- Latency of the roundtrip of a message from Src Instance to the Target instances

Instance#	500 bytes					8 Kbytes				
	Src	Target	Ping Count	Ping Time(s)	Avg Time(ms)	Std Dev	Ping Count	Ping Time(s)	Avg Time(ms)	Std Dev
1	1		286	0.04	0.14	0.03	286	0.04	0.13	0.03
1	2		286	0.06	0.21	0.05	286	0.09	0.32	0.09
1	3		286	0.07	0.23	0.06	286	0.10	0.35	0.24
2	1		286	0.08	0.28	0.08	286	0.09	0.32	0.06
2	2		286	0.04	0.15	0.04	286	0.04	0.14	0.03
2	3		286	0.07	0.23	0.03	286	0.08	0.29	0.07
3	1		286	0.08	0.27	0.06	286	0.11	0.40	0.24
3	2		286	0.07	0.23	0.08	286	0.10	0.35	0.08
3	3		286	0.04	0.13	0.03	286	0.03	0.12	0.03

[Back to Interconnect Stats](#)

[Back to Top](#)

Interconnect Client Statistics

#	Sent (MB)						Received (MB)					
	Total	Cache	IPQ	DLM	PNG	Misc	Total	Cache	IPQ	DLM	PNG	Misc
1	4,581.78	1,390.58	2,750.45	345.85	7.20	87.71	3,157.10	1,223.56	1,323.27	395.68	7.20	207.38
2	2,937.03	1,789.90	710.17	317.00	7.20	112.76	3,089.96	1,113.20	1,534.35	368.04	7.20	67.16
3	2,739.86	1,335.04	828.64	445.52	7.20	123.47	4,011.67	2,178.57	1,431.68	344.71	7.20	49.51
Sum	10,258.67	4,515.52	4,289.26	1,108.36	21.59	323.93	10,258.73	4,515.34	4,289.30	1,108.43	21.59	324.06
Avg	3,419.56	1,505.17	1,429.75	369.45	7.20	107.98	3,419.58	1,505.11	1,429.77	369.48	7.20	108.02
Std	1,011.33	248.14	1,145.29	67.43	0.00	18.36	513.87	585.84	105.55	25.52	0.00	86.50

[Back to Interconnect Stats](#)

[Back to Top](#)

Interconnect Client Statistics (per Second)

#	Sent (MB/s)						Received (MB/s)					
	Total	Cache	IPQ	DLM	PNG	Misc	Total	Cache	IPQ	DLM	PNG	Misc
1	1.27	0.39	0.76	0.10	0.00	0.02	0.88	0.34	0.37	0.11	0.00	0.06
2	0.81	0.50	0.20	0.09	0.00	0.03	0.86	0.31	0.43	0.10	0.00	0.02
3	0.76	0.37	0.23	0.12	0.00	0.03	1.11	0.60	0.40	0.10	0.00	0.01
Sum	2.84	1.25	1.19	0.31	0.01	0.09	2.84	1.25	1.19	0.31	0.01	0.09
Avg	0.95	0.42	0.40	0.10	0.00	0.03	0.95	0.42	0.40	0.10	0.00	0.03
Std	0.28	0.07	0.32	0.02	0.00	0.01	0.14	0.16	0.03	0.01	0.00	0.02

[Back to Interconnect Stats](#)

[Back to Top](#)

Interconnect Device Statistics

- Data is retrieved from underlying Operating system and may overflow on some 32-bit OSs
- null means begin value > end value

#	Interface		Sent					Received							
	NAME	IP Address	IP Mask	MBytes	Packets	Errors	Packets Dropped	Buffer Ovrnun	Carrier Lost	MBytes	Packets	Errors	Packets Dropped	Buffer Ovrnun	Fram Error
1	bond1	10.2.2.104	255.255.255.0	15,746.42	17,561,359	0	0	0	0	10,932.87	13,734,822	0	0	0	0
1	bond1:1	169.254.78.178	255.255.0.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
1	em1	10.2.4.45	255.255.255.0	0.00	7	0	0	0	0	0.05	319	0	0	0	0
1	em2	10.2.5.45	255.255.255.0	0.00	1	0	0	0	0	0.05	313	0	0	0	0
1	lo	127.0.0.1	255.0.0.0	1,878.38	10,157,542	0	0	0	0	1,878.38	10,157,542	0	0	0	0
1	p2p1	131.128.3.104	255.255.255.0	3,496.94	6,736,092	0	0	0	0	1,932.11	6,221,268	0	0	0	0
1	p2p1:1	131.128.3.124	255.255.255.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
1	p2p1:3	131.128.3.97	255.255.255.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
1	p3p1	10.2.6.45	255.255.255.0	0.00	7	0	0	0	0	0.04	134	0	0	0	0
1	p3p2	10.2.7.45	255.255.255.0	0.00	7	0	0	0	0	0.04	134	0	0	0	0
2	bond1	10.2.2.105	255.255.255.0	13,240.28	15,039,457	0	0	0	0	10,371.15	12,790,485	0	0	0	0
2	bond1:1	169.254.53.146	255.255.0.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
2	em1	10.2.4.46	255.255.255.0	0.00	9	0	0	0	0	0.01	199	0	0	0	0

2 em2	10.2.5.46	255.255.255.0	0.00	9	0	0	0	0	0.01	199	0	0	0
2 lo	127.0.0.1	255.0.0.0	502.59	1,686,500	0	0	0	0	502.59	1,686,500	0	0	0
2 p2p1	131.128.3.105	255.255.255.0	3,468.97	6,775,165	0	0	0	0	1,934.32	6,168,372	0	0	0
2 p2p1:1	131.128.3.125	255.255.255.0	0.00	0	0	0	0	0	0.00	0	0	0	0
2 p2p1:2	131.128.3.98	255.255.255.0	0.00	0	0	0	0	0	0.00	0	0	0	0
2 p3p1	10.2.6.46	255.255.255.0	0.00	10	0	0	0	0	0.00	13	0	0	0
2 p3p2	10.2.7.46	255.255.255.0	0.00	10	0	0	0	0	0.00	13	0	0	0
3 bond1	10.2.2.106	255.255.255.0	11,629.29	14,216,731	0	0	0	0	19,251.93	20,253,759	0	0	0
3 bond1:1	169.254.232.32	255.255.0.0	0.00	0	0	0	0	0	0.00	0	0	0	0
3 em1	10.2.4.47	255.255.255.0	0.00	7	0	0	0	0	0.05	319	0	0	0
3 em2	10.2.5.47	255.255.255.0	0.00	7	0	0	0	0	0.05	319	0	0	0
3 lo	127.0.0.1	255.0.0.0	858.63	3,824,676	0	0	0	0	858.63	3,824,676	0	0	0
3 p2p1	131.128.3.106	255.255.255.0	4,629.54	8,640,495	0	0	0	0	2,346.02	8,111,227	0	0	0
3 p2p1:1	131.128.3.126	255.255.255.0	0.00	0	0	0	0	0	0.00	0	0	0	0
3 p2p1:2	131.128.3.96	255.255.255.0	0.00	0	0	0	0	0	0.00	0	0	0	0
3 p3p1	10.2.6.47	255.255.255.0	0.00	8	0	0	0	0	0.04	133	0	0	0
3 p3p2	10.2.7.47	255.255.255.0	0.00	8	0	0	0	0	0.04	133	0	0	0
Sum			55,451.04	84,638,107	0	0	0	0	50,008.41	82,950,879	0	0	0
Avg			1,848.37	2,821,270	0	0	0	0	1,666.95	2,765,029	0	0	0
Std			4,178.00	5,201,124	0	0	0	0	4,282.52	5,240,058	0	0	0

[Back to Interconnect Stats](#)
[Back to Top](#)

Interconnect Device Statistics (per Second)

- Data is retrieved from underlying Operating system and may overflow on some 32-bit OSs
- Blank means begin value > end value

#	Interface				Sent				Received						
	NAME	IP Address	IP Mask	MBytes	Packets	Errors	Packets Dropped	Buffer Overrun	Carrier Lost	MBytes	Packets	Errors	Packets Dropped	Buffer Overrun	Frame Errors
1	bond1	10.2.2.104	255.255.255.0	4.37	4,868	0	0	0	0	3.03	3,807	0	0	0	0
1	bond1:1	169.254.78.178	255.255.0.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
1	em1	10.2.4.45	255.255.255.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
1	em2	10.2.5.45	255.255.255.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
1	lo	127.0.0.1	255.0.0.0	0.52	2,816	0	0	0	0	0.52	2,816	0	0	0	0
1	p2p1	131.128.3.104	255.255.255.0	0.97	1,867	0	0	0	0	0.54	1,725	0	0	0	0
1	p2p1:1	131.128.3.124	255.255.255.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
1	p2p1:3	131.128.3.97	255.255.255.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
1	p3p1	10.2.6.45	255.255.255.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
1	p3p2	10.2.7.45	255.255.255.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
2	bond1	10.2.2.105	255.255.255.0	3.67	4,169	0	0	0	0	2.87	3,546	0	0	0	0
2	bond1:1	169.254.53.146	255.255.0.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
2	em1	10.2.4.46	255.255.255.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
2	em2	10.2.5.46	255.255.255.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
2	lo	127.0.0.1	255.0.0.0	0.14	468	0	0	0	0	0.14	468	0	0	0	0
2	p2p1	131.128.3.105	255.255.255.0	0.96	1,878	0	0	0	0	0.54	1,710	0	0	0	0
2	p2p1:1	131.128.3.125	255.255.255.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
2	p2p1:2	131.128.3.98	255.255.255.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
2	p3p1	10.2.6.46	255.255.255.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
2	p3p2	10.2.7.46	255.255.255.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
3	bond1	10.2.2.106	255.255.255.0	3.22	3,941	0	0	0	0	5.34	5,614	0	0	0	0
3	bond1:1	169.254.232.32	255.255.0.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
3	em1	10.2.4.47	255.255.255.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
3	em2	10.2.5.47	255.255.255.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
3	lo	127.0.0.1	255.0.0.0	0.24	1,060	0	0	0	0	0.24	1,060	0	0	0	0
3	p2p1	131.128.3.106	255.255.255.0	1.28	2,395	0	0	0	0	0.65	2,248	0	0	0	0
3	p2p1:1	131.128.3.126	255.255.255.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
3	p2p1:2	131.128.3.96	255.255.255.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
3	p3p1	10.2.6.47	255.255.255.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
3	p3p2	10.2.7.47	255.255.255.0	0.00	0	0	0	0	0	0.00	0	0	0	0	0
Sum			15.37	23,462	0	0	0	0	13.86	22,994	0	0	0	0	0
Avg			0.51	782	0	0	0	0	0.46	766	0	0	0	0	0
Std			1.16	1,442	0	0	0	0	1.19	1,453	0	0	0	0	0

[Back to Interconnect Stats](#)
[Back to Top](#)

Dynamic Remastering Statistics - per Remaster Ops

- Affinity/Read-mostly Obj- objects remastered due to affinity or being Read-Mostly at begin/end snap

#	Remaster Ops	Remastered Objects	Counts per Ops			Resources Cleaned	Time per Ops (seconds)						Affinity/Read Mostly Obj	
			Replay Locks Received	Replay Locks Sent	Replay Locks		Remaster	Quiesce	Freeze	Cleanup	Replay	Fixwrite	Sync	Begin
1	2	1.00	0.00	3.00	0.00	0.68	0.00	0.09	0.11	0.09	0.10	0.29	62	62
2	2	1.00	0.00	7,499.00	0.00	0.68	0.00	0.11	0.13	0.13	0.07	0.24	92	90
3	2	1.00	7,502.00	0.00	0.00	0.68	0.00	0.09	0.11	0.09	0.12	0.27	114	116
Avg	2	1.00	2,500.67	2,500.67	0.00	0.68	0.00	0.10	0.12	0.10	0.10	0.27	89	89
Max			7,502.00	7,499.00	0.00	0.68	0.00	0.11	0.13	0.13	0.12	0.29	114	116

[Back to Interconnect Stats](#)
[Back to Top](#)

SQL Statistics

- [SQL ordered by Elapsed Time \(Global\)](#)
- [SQL ordered by CPU Time \(Global\)](#)
- [SQL ordered by User I/O Time \(Global\)](#)
- [SQL ordered by Gets \(Global\)](#)
- [SQL ordered by Reads \(Global\)](#)
- [SQL ordered by UnOptimized Read Requests \(Global\)](#)
- [SQL ordered by Executions \(Global\)](#)
- [SQL ordered by Cluster Wait Time \(Global\)](#)

[Back to Top](#)

SQL ordered by Elapsed Time (Global)

- Captured SQL account for 61.5% of Total DB Time (s): 5,377
- Captured PL/SQL account for 10.9% of Total DB Time (s): 5,377

SQL Id	Total								Per Execution						
	Elapsed (s)	CPU (s)	IOWait (s)	Gets	Reads	Rows	Cluster (s)	Execs	Elapsed (s)	CPU (s)	IOWait (s)	Gets	Reads	Rows	Clu (s)
3nkwn9sq77s4	1,286.01	380.90	1,046.87	29,979,923	4,476,877	0	10.41	26	49.46	14.65	40.26	1,153,073.96	172,187.58	0.00	
768qjmf9pu5k	429.33	413.10	5.41	99,710,393	26,858	1	0.98	1	429.33	413.10	5.41	99,710,393.00	26,858.00	1.00	
07uac1c295aum	268.01	262.06	0.00	65,820,900	0	29,060	0.00	29,060	0.01	0.01	0.00	2,265.00	0.00	1.00	
9xuu98ug5bun5	267.61	278.08	0.13	9,634,935	155	57	0.45	1,643	0.16	0.17	0.00	5,864.23	0.09	0.03	
72uzt9d37ux2z	148.66	138.00	3.41	6,649,162	6,806	2,772	3.56	22	6.76	6.27	0.16	302,234.64	309.36	126.00	
dffkcnqfystw	88.27	72.51	1.78	171,033	4,578	408	0.24	408	0.22	0.18	0.00	419.20	11.22	1.00	
f7zfpahf79u4g	87.11	29.72	4.86	14,991,540	9,236	807,469	10.71	807,469	0.00	0.00	0.00	18.57	0.01	1.00	
8722qujwb0rfq	79.19	84.80	0.85	632,821	4,090	135	0.02	72	1.10	1.18	0.01	8,789.18	56.81	1.88	
gbb4d7q07pvm3	78.03	58.70	10.25	10,490,991	29,292	1	6.00	1	78.03	58.70	10.25	10,490,991.00	29,292.00	1.00	
fhf8upax5cxsz	59.55	41.43	2.17	256,536	4,747	516	0.85	516	0.12	0.08	0.00	497.16	9.20	1.00	

[Back to SQL Statistics](#)
[Back to Top](#)

SQL ordered by CPU Time (Global)

- Captured SQL account for 49.3% of Total CPU Time (s): 3,716
- Captured PL/SQL account for 14.2% of Total CPU Time (s): 3,716

SQL Id	Total								Per Execution						
	CPU (s)	Elapsed (s)	IOWait (s)	Gets	Reads	Rows	Cluster (s)	Execs	CPU (s)	Elapsed (s)	IOWait (s)	Gets	Reads	Rows	Clust (s)

768qjmf9pu5k	413.10	429.33	5.41	99,710,393	26,858	1	0.98	1	1413.10	429.33	5.41	99,710,393.00	26,858.00	1.00	0.00
3nkwt9sq77s4	380.90	1,286.01	1,046.87	29,979,923	4,476,877	0	10.41	26	14.65	49.46	40.26	1,153,073.96	172,187.58	0.00	0.00
9xuu98ug5bun5	278.08	267.61	0.13	9,634,935	155	57	0.45	1,643	0.17	0.16	0.00	5,864.23	0.09	0.03	0.00
07uac1c295aum	262.06	268.01	0.00	65,820,900	0	29,060	0.00	29,060	0.01	0.01	0.00	2,265.00	0.00	1.00	0.00
72uzt9d37ux2z	138.00	148.66	3.41	6,649,162	6,806	2,772	3.56	22	6.27	6.76	0.16	302,234.64	309.36	126.00	0.00
8722qujw0rfq	84.80	79.19	0.85	632,821	4,090	135	0.02	72	1.18	1.10	0.01	8,789.18	56.81	1.88	0.00
dffkcnqfystw	72.51	88.27	1.78	171,033	4,578	408	0.24	408	0.18	0.22	0.00	419.20	11.22	1.00	0.00
gbb4d7q07pvm3	58.70	78.03	10.25	10,490,991	29,292	1	6.00	1	58.70	78.03	10.25	10,490,991.00	29,292.00	1.00	6.00
41pxak5vzhfpw	48.36	56.07	0.01	2,177	28	0	0.04	0							
fyvbpjf4k8cjp	44.73	48.68	0.02	12,065,714	9	6	0.03	5,327	0.01	0.01	0.00	2,265.01	0.00	0.00	0.00

[Back to SQL Statistics](#)
[Back to Top](#)

SQL ordered by User I/O Time (Global)

- Captured SQL account for 83.0% of Total User I/O Wait Time (s): 1,522
- Captured PL/SQL account for 1.5% of Total User I/O Wait Time (s): 1,522

SQL Id	Total							Per Execution							
	IOWait (s)	Elapsed (s)	CPU (s)	Gets	Reads	Rows	Cluster (s)	Execs	IOWait (s)	Elapsed (s)	CPU (s)	Gets	Reads	Rows	Cluster (s)
3nkwt9sq77s4	1,046.87	1,286.01	380.90	29,979,923	4,476,877	0	10.41	26	40.26	49.46	14.65	1,153,073.96	172,187.58	0.00	0.00
gbb4d7q07pvm3	10.25	78.03	58.70	10,490,991	29,292	1	6.00	1	10.25	78.03	58.70	10,490,991.00	29,292.00	1.00	6.00
768qjmf9pu5k	5.41	429.33	413.10	99,710,393	26,858	1	0.98	1	5.41	429.33	413.10	99,710,393.00	26,858.00	1.00	0.00
72uzt9d37ux2z	3.41	148.66	138.00	6,649,162	6,806	2,772	3.56	22	0.16	6.76	6.27	302,234.64	309.36	126.00	0.00
dffkcnqfystw	1.78	88.27	72.51	171,033	4,578	408	0.24	408	0.00	0.22	0.18	419.20	11.22	1.00	0.00
8722qujw0rfq	0.85	79.19	84.80	632,821	4,090	135	0.02	72	0.01	1.10	1.18	8,789.18	56.81	1.88	0.00
9xuu98ug5bun5	0.13	267.61	278.08	9,634,935	155	57	0.45	1,643	0.00	0.16	0.17	5,864.23	0.09	0.03	0.00
fyvbpjf4k8cjp	0.02	48.68	44.73	12,065,714	9	6	0.03	5,327	0.00	0.01	0.01	2,265.01	0.00	0.00	0.00
41pxak5vzhfpw	0.01	56.07	48.36	2,177	28	0	0.04	0							
07uac1c295aum	0.00	268.01	262.06	65,820,900	0	29,060	0.00	29,060	0.00	0.01	0.01	2,265.00	0.00	1.00	0.00

[Back to SQL Statistics](#)
[Back to Top](#)

SQL ordered by Gets (Global)

- Total Buffer Gets: 323,229,777
- Captured SQL account for 72.0% of Total

SQL Id	Total							Per Execution					
	Gets	Reads	Elapsed (s)	CPU (s)	IOWait (s)	Rows	Cluster (s)	Execs	Gets	Reads	Elapsed (s)	CPU (s)	IOWait (s)

768qjmf9pu5k	99,710,393	26,858	429.33	413.10	5.41	1	0.98	1	99,710,393.00	26,858.00	429.33	413.10	5.41	1.00
07uac1c295aum	65,820,900	0	268.01	262.06	0.00	29,060	0.00	29,060	2,265.00	0.00	0.01	0.01	0.00	1.00
3nkwn9sq77s4	29,979,923	4,476,877	1,286.01	380.90	1,046.87	0	10.41	26	1,153,073.96	172,187.58	49.46	14.65	40.26	0.00
f7zfpahf79u4q	14,991,540	9,236	87.11	29.72	4.86	807,469	10.71	807,469	18.57	0.01	0.00	0.00	0.00	1.00
fyvbjfj4k8cjp	12,065,714	9	48.68	44.73	0.02	6	0.03	5,327	2,265.01	0.00	0.01	0.01	0.00	0.00
gbb4d7q07pvm3	10,490,991	29,292	78.03	58.70	10.25	1	6.00	1	10,490,991.00	29,292.00	78.03	58.70	10.25	1.00
ggaxpdz7wvv02	9,882,106	1,278	23.20	21.83	0.61	24,411	0.04	1	9,882,106.00	1,278.00	23.20	21.83	0.61	24,411.00
9xuu98ug5bun5	9,634,935	155	267.61	278.08	0.13	57	0.45	1,643	5,864.23	0.09	0.16	0.17	0.00	0.00
664ab0677vad5	7,483,162	7,859	36.16	27.05	3.67	99,225	3.25	6,152	1,216.38	1.28	0.01	0.00	0.00	16.10
5xb53jnnx2r98	6,809,975	5	11.70	10.94	0.00	732,595	0.00	1	6,809,975.00	5.00	11.70	10.94	0.00	732,595.00

[Back to SQL Statistics](#)
[Back to Top](#)

SQL ordered by Reads (Global)

- Total Disk Reads: 8,834,554
- Captured SQL account for 83.8% of Total

SQL Id	Reads	Gets	Total			Rows	Cluster (s)	Execs	Per Execution			Rows	%	
			Elapsed (s)	CPU (s)	IOWait (s)				Elapsed (s)	CPU (s)	IOWait (s)			
3nkwn9sq77s4	4,476,877	29,979,923	1,286.01	380.90	1,046.87	0	10.41	26	172,187.58	1,153,073.96	49.46	14.65	40.26	0.00
15pgaraszc3ps	1,466,355	1,568,420	38.83	23.28	12.67	62	0.34	44	33,326.25	35,645.91	0.88	0.53	0.29	1.41
4ztz048yfq32s	791,320	791,342	10.50	2.83	7.41	2	0.00	2	395,660.00	395,671.00	5.25	1.41	3.71	1.00
b9qxyft335u80	176,135	352,720	36.25	8.45	25.02	62,302	0.25	26,100	6.75	13.51	0.00	0.00	0.00	2.39
2u1d27gq1326c	99,390	875,109	13.29	11.30	1.28	108	0.67	1	99,390.00	875,109.00	13.29	11.30	1.28	108.00
4u72kk4b9fvdz	45,966	1,528,210	29.97	8.20	21.34	2	2.36	1	45,966.00	1,528,210.00	29.97	8.20	21.34	2.00
6g4vwbj336c8x	33,944	85,403	21.60	5.95	13.35	39,036	3.91	2	16,972.00	42,701.50	10.80	2.97	6.68	19,518.00
390ummz838758	30,308	62,265	15.57	2.35	8.04	825,259	0.00	1	30,308.00	62,265.00	15.57	2.35	8.04	825,259.00
8v0ctgmbmxzh1	30,239	2,473,134	19.17	13.39	4.94	0	0.83	1	30,239.00	2,473,134.00	19.17	13.39	4.94	0.00
gbb4d7q07pvm3	29,292	10,490,991	78.03	58.70	10.25	1	6.00	1	29,292.00	10,490,991.00	78.03	58.70	10.25	1.00

[Back to SQL Statistics](#)
[Back to Top](#)

SQL ordered by UnOptimized Read Requests (Global)

- Total Physical Read Requests: 5,404,708
- Captured SQL account for 90.8% of Total
- Total UnOptimized Read Requests: 5,404,708
- Captured SQL account for 90.8% of Total
- Total Optimized Read Requests: 1
- Captured SQL account for 0.0% of Total

SQL Id	UnOptimized Requests	Reads	Total			Rows	Cluster (s)	Execs	Per Execution			Rows	%	
			Elapsed (s)	CPU (s)	IOWait (s)				UnOptimized Requests	Reads	Elapsed (s)			
3nkwn9sq77s4	4,432,077	4,432,077	4,476,877	1,286.01	380.90	1,046.87	0	10.41	26	170,464.50	170,464.50	172,187.58	49.46	14
b9qxyft335u80	176,131	176,131	176,135	36.25	8.45	25.02	62,302	0.25	26,100	6.75	6.75	6.75	0.00	0
4u72kk4b9fvdz	45,966	45,966	45,966	29.97	8.20	21.34	2	2.36	1	45,966.00	45,966.00	45,966.00	29.97	8

6g4vwbj836c8x	33,931	33,931	33,944	21.60	5.95	13.35	39,036	3.91	2	16,965.50	16,965.50	16,972.00	10.80	2
gbb4d7q07pvm3	25,870	25,870	29,292	78.03	58.70	10.25	1	6.00	1	25,870.00	25,870.00	29,292.00	78.03	58
c83vn8rw2dstg	23,784	23,784	23,784	42.42	20.86	11.83	797,808	11.07	346	68.74	68.74	68.74	0.12	0
a712uyahq6fvf	19,226	19,226	29,207	20.71	2.84	13.28	46,219	2.56	42,941	0.45	0.45	0.68	0.00	0
dag0yvdynd3ky1	17,865	17,865	18,518	24.69	9.00	10.25	3,871	6.99	78	229.04	229.04	237.41	0.32	0
dw9t03ynrgw23	15,979	15,979	15,979	23.48	14.99	6.33	1	3.08	1	15,979.00	15,979.00	15,979.00	23.48	14
44zs47ktvznb9	13,468	13,468	13,468	12.31	2.13	9.30	2,609	1.02	18,166	0.74	0.74	0.74	0.00	0

[Back to SQL Statistics](#)
[Back to Top](#)

SQL ordered by Executions (Global)

- Total Executions: 14,367,007
- Captured SQL account for 44.9% of Total

SQL Id	Execs	Total							Per Execution							Perce		
		Elapsed (s)	CPU (s)	IOWait (s)	Gets	Reads	Rows	Cluster (s)	Elapsed (s)	CPU (s)	IOWait (s)	Gets	Reads	Rows	Cluster (s)	Execs	DB time	DB CPU
f7zfpahf79u4q	807,469	87.11	29.72	4.86	14,991,540	9,236	807,469	10.71	0.00	0.00	0.00	18.57	0.01	1.00	0.00	5.62	1.62	0.80
b08jjbpg4rs1n	732,595	9.37	9.32	0.00	1,559,207	0	28,333	0.00	0.00	0.00	0.00	2.13	0.00	0.04	0.00	5.10	0.17	0.25
93na5cg9gjc6p	623,770	7.66	7.46	0.00	1,272,821	0	24,222	0.00	0.00	0.00	0.00	2.04	0.00	0.04	0.00	4.34	0.14	0.20
3akwpg5buy85m	519,852	6.39	6.34	0.00	1,126,926	0	27,398	0.00	0.00	0.00	0.00	2.17	0.00	0.05	0.00	3.62	0.12	0.17
c93sup0snbjym	402,365	8.34	1.64	0.02	805,983	48	136	0.03	0.00	0.00	0.00	2.00	0.00	0.00	0.00	2.80	0.16	0.04
7ah2cs0jwhwbn	319,677	4.14	4.09	0.00	716,709	0	27,205	0.00	0.00	0.00	0.00	2.24	0.00	0.09	0.00	2.23	0.08	0.11
an88hf8k322jw	316,651	4.28	4.15	0.00	715,621	0	27,007	0.00	0.00	0.00	0.00	2.26	0.00	0.09	0.00	2.20	0.08	0.11
bdntyxtax2smg	261,913	11.87	1.58	0.00	527,499	0	261,914	3.23	0.00	0.00	0.00	2.01	0.00	1.00	0.00	1.82	0.22	0.04
8r9tyhzf6t4wb	233,647	11.60	3.90	0.01	1,011,012	16	233,647	0.01	0.00	0.00	0.00	4.33	0.00	1.00	0.00	1.63	0.22	0.10
d2uzkd87gy505	228,171	11.41	4.56	0.01	2,249,434	13	227,785	0.06	0.00	0.00	0.00	9.86	0.00	1.00	0.00	1.59	0.21	0.12

[Back to SQL Statistics](#)
[Back to Top](#)

SQL ordered by Cluster Wait Time (Global)

- Total Cluster Wait Time (s): 301
- Captured SQL account for 41.2% of Total

SQL Id	Cluster (s)	Elapsed (s)	CPU (s)	IOWait (s)	Total				Per Execution						
					Gets	Reads	Rows	Execs	Cluster (s)	Elapsed (s)	CPU (s)	IOWait (s)	Gets	Reads	R
2kxhn3uqvgc2f	16.31	32.64	16.42	0.00	3,169,843	2	698,863	74	0.22	0.44	0.22	0.00	42,835.72	0.03	9.4
c83vn8rw2dstg	11.07	42.42	20.86	11.83	1,216,152	23,784	797,808	346	0.03	0.12	0.06	0.03	3,514.89	68.74	2.1
f7zfpahf79u4q	10.71	87.11	29.72	4.86	14,991,540	9,236	807,469	807,469	0.00	0.00	0.00	0.00	18.57	0.01	

3nkwtn9sq77s4	10.41	1,286.01	380.90	1,046.87	29,979,923	4,476,877	0	26	0.40	49.46	14.65	40.26	1,153,073.96	172,187.58
6hvumstf84xhq	10.03	15.04	3.19	0.93	224,903	1,326,574,782	11,512	0.00	0.00	0.00	0.00		19.54	0.12
dag0yvdyndyn3ky1	6.99	24.69	9.00	10.25	4,699,673	18,518	3,871	78	0.09	0.32	0.12	0.13	60,252.22	237.41
gbb4d7q07pvm3	6.00	78.03	58.70	10.25	10,490,991	29,292	1	1	6.00	78.03	58.70	10.25	10,490,991.00	29,292.00
fu76ktu3rbuuf	4.80	9.05	2.98	2.58	87,968	6,447	192	19,260	0.00	0.00	0.00	0.00	4.57	0.33
avpp3w6rzbjr2	4.46	13.45	8.17	1.71	1,030,346	10,514	0	37	0.12	0.36	0.22	0.05	27,847.19	284.16
6g4vwbj836c8x	3.91	21.60	5.95	13.35	85,403	33,944	39,036	2	1.96	10.80	2.97	6.68	42,701.50	16,972.00

[Back to SQL Statistics](#)
[Back to Top](#)

Global Activity Statistics

- [System Statistics \(Global\)](#)
- [Global Messaging Statistics \(Global\)](#)
- [System Statistics \(Absolute Values\)](#)

[Back to Top](#)

System Statistics (Global)

- per Second Average - average of per-instance per Second rates
- per Second Std Dev - standard deviation of per-instance per Second rates
- per Second Min - minimum of per-instance per Second rates
- per Second Max - maximum of per-instance per Second rates

Statistic	Total	per Second	per Trans	Average	per Second		
					Std Dev	Min	Max
Batched IO (bound) vector count	4,990,499	1,383.39	69.28	461.13	784.05	5.68	1,366.47
Batched IO (full) vector count	79	0.02	0.00	0.01	0.00	0.01	0.01
Batched IO (space) vector count	1	0.00	0.00	0.00		0.00	0.00
Batched IO block miss count	15,087,627	4,182.36	209.46	1,394.12	2,372.79	18.43	4,133.97
Batched IO buffer defrag count	10,202	2.83	0.14	0.94	0.82	0.03	1.62
Batched IO double miss count	175,296	48.59	2.43	16.20	26.01	1.14	46.23
Batched IO same unit count	9,971,375	2,764.11	138.43	921.37	1,576.59	7.65	2,741.86
Batched IO single block count	4,933,364	1,367.55	68.49	455.85	776.62	3.76	1,352.61
Batched IO slow jump count	318	0.09	0.00	0.03	0.03	0.01	0.06
Batched IO vector block count	241,183	66.86	3.35	22.29	20.05	8.45	45.28
Batched IO vector read count	73,014	20.24	1.01	6.75	8.11	1.22	16.06
Block Cleanout Optim referenced	2,807	0.78	0.04	0.26	0.03	0.22	0.29
CCursor + sql area evicted	40,415	11.20	0.56	3.73	1.16	2.45	4.71
CPU used by this session	309,710	85.85	4.30	28.62	12.57	14.49	38.56
CPU used when call started	274,867	76.19	3.82	25.40	12.54	11.31	35.33
CR blocks created	5,286	1.47	0.07	0.49	0.41	0.14	0.94
Cached Commit SCN referenced	39,486	10.95	0.55	3.65	3.55	0.07	7.17
Clusterwide global transactions	18,774	5.20	0.26	1.73	3.00	0.00	5.20
Commit SCN cached	2,052	0.57	0.03	0.19	0.07	0.11	0.24
DB time	1,024,088	283.88	14.22	94.63	29.58	60.48	112.34
DBWR checkpoint buffers written	259,397	71.91	3.60	23.97	2.67	20.89	25.64
DBWR checkpoints	6,552	1.82	0.09	0.61	0.00	0.60	0.61
DBWR fusion writes	58,291	16.16	0.81	5.39	1.59	3.72	6.89
DBWR object drop buffers written	1,435	0.40	0.02	0.13	0.02	0.12	0.15
DBWR parallel query checkpoint buffers written	211	0.06	0.00	0.02	0.01	0.01	0.03
DBWR thread checkpoint buffers written	170,147	47.17	2.36	15.72	2.51	14.20	18.62
DBWR transaction table writes	632	0.18	0.01	0.06	0.00	0.06	0.06
DBWR undo block writes	192,460	53.35	2.67	17.78	3.92	14.91	22.24
DFO trees parallelized	4,238	1.17	0.06	0.39	0.12	0.31	0.53
DX/BB enqueue lock background get time	736	0.20	0.01	0.20		0.20	0.20
DX/BB enqueue lock background	37,548	10.41	0.52	3.47	6.00	0.00	10.40

commit batch/immediate performed	79	0.02	0.00	0.01	0.00	0.00	0.01
commit batch/immediate requested	79	0.02	0.00	0.01	0.00	0.00	0.01
commit cleanout failures: block lost	174	0.05	0.00	0.02	0.00	0.01	0.02
commit cleanout failures: buffer being written	1	0.00	0.00	0.00		0.00	0.00
commit cleanout failures: callback failure	3,553	0.98	0.05	0.33	0.26	0.15	0.62
commit cleanout failures: cannot pin	10	0.00	0.00	0.00	0.00	0.00	0.00
commit cleanouts	537,347	148.96	7.46	49.65	2.09	47.30	51.29
commit cleanouts successfully completed	533,609	147.92	7.41	49.31	2.06	47.07	51.12
commit immediate performed	71	0.02	0.00	0.01	0.00	0.00	0.01
commit immediate requested	71	0.02	0.00	0.01	0.00	0.00	0.01
commit txn count during cleanout	25,518	7.07	0.35	2.36	1.25	1.20	3.68
concurrency wait time	3,668	1.02	0.05	0.34	0.40	0.10	0.80
consistent changes	231,835	64.27	3.22	21.42	8.97	11.95	29.79
consistent gets	297,168,942	82,376.85	4,125.51	27,458.95	16,952.06	10,436.85	44,340.09
consistent gets direct	3,465,187	960.57	48.11	320.19	164.35	134.71	447.72
consistent gets examination	58,216,135	16,137.82	808.20	5,379.27	2,000.85	3,111.68	6,896.39
consistent gets examination (fastpath)	58,048,158	16,091.25	805.87	5,363.75	1,998.02	3,100.84	6,884.42
consistent gets from cache	293,703,755	81,416.28	4,077.41	27,138.76	16,795.27	10,302.14	43,892.38
consistent gets pin	235,487,622	65,278.46	3,269.21	21,759.49	14,913.97	7,190.46	36,995.98
consistent gets pin (fastpath)	225,642,205	62,549.26	3,132.53	20,849.75	14,687.45	6,911.99	36,186.74
cursor authentications	11,004	3.05	0.15	1.02	0.67	0.33	1.66
cursor reload failures	39	0.01	0.00	0.00	0.00	0.00	0.01
data blocks consistent reads - undo records applied	28,248	7.83	0.39	2.61	2.46	0.94	5.43
db block changes	19,267,489	5,341.04	267.49	1,780.35	526.15	1,384.74	2,377.47
db block gets	26,060,919	7,224.21	361.80	2,408.07	574.05	1,926.06	3,043.13
db block gets direct	2,854	0.79	0.04	0.26	0.11	0.20	0.39
db block gets from cache	26,058,065	7,223.42	361.76	2,407.81	574.13	1,925.67	3,042.93
db block gets from cache (fastpath)	15,115,360	4,190.05	209.84	1,396.68	310.90	1,103.99	1,723.05
deferred (CURRENT) block cleanout applications	213,423	59.16	2.96	19.72	8.13	13.75	28.98
dirty buffers inspected	31,417	8.71	0.44	2.90	4.92	0.02	8.59
enqueue conversions	56,001	15.52	0.78	5.17	0.59	4.69	5.83
enqueue releases	533,997	148.03	7.41	49.34	10.56	40.21	60.90
enqueue requests	539,341	149.51	7.49	49.84	10.52	40.54	61.26
enqueue timeouts	5,308	1.47	0.07	0.49	0.26	0.32	0.79
enqueue waits	43,757	12.13	0.61	4.04	0.76	3.39	4.87
exchange deadlocks	13	0.00	0.00	0.00	0.00	0.00	0.00
execute count	14,367,007	3,982.61	199.45	1,327.54	500.83	765.99	1,728.00
failed probes on index block reclamation	33	0.01	0.00	0.00	0.00	0.00	0.00
fastpath consistent get quota limit	3,238,247	897.66	44.96	299.22	221.80	130.10	550.35
file io service time	1,936,091	536.69	26.88	178.90	38.04	135.08	203.52
file io wait time	2,684,062,282	744,036.05	37,262.08	248,012.02	153,020.12	88,668.21	393,807.45
free buffer inspected	6,483,837	1,797.35	90.01	599.12	783.66	128.01	1,503.75
free buffer requested	6,536,137	1,811.85	90.74	603.95	784.56	136.46	1,509.73
gc blocks compressed	429,089	118.95	5.96	39.65	17.99	19.52	54.16
gc cleanout applied	19	0.01	0.00	0.00	0.00	0.00	0.00
gc cleanout saved	152	0.04	0.00	0.01	0.00	0.01	0.02
gc cr block flush time	312	0.09	0.00	0.03	0.02	0.01	0.05
gc cr block receive time	14,109	3.91	0.20	1.30	0.66	0.54	1.75
gc cr blocks flushed	2,971	0.82	0.04	0.27	0.18	0.08	0.44
gc cr blocks received	217,717	60.35	3.02	20.12	11.19	7.31	27.96
gc cr blocks served	217,735	60.36	3.02	20.12	12.00	6.31	27.94
gc current block flush time	47	0.01	0.00	0.00	0.00	0.00	0.01
gc current block receive time	38,725	10.73	0.54	3.58	2.40	2.01	6.34
gc current blocks flushed	362	0.10	0.01	0.03	0.01	0.03	0.04
gc current blocks pinned	30	0.01	0.00	0.00	0.00	0.00	0.00
gc current blocks received	632,190	175.25	8.78	58.42	24.67	43.20	86.88
gc current blocks served	632,232	175.26	8.78	58.42	11.11	51.99	71.25
gc force cr read cr	392	0.11	0.01	0.04	0.05	0.00	0.10
gc force cr read current	15	0.00	0.00	0.00	0.00	0.00	0.00
gc kbytes saved	2,176,157	603.24	30.21	201.08	113.80	71.62	285.36
gc kbytes sent	4,623,564	1,281.67	64.19	427.22	70.52	378.82	508.13

gc local grants	5,220,485	1,447.14	72.47	482.38	740.05	53.34	1,336.91
gc read time waited	5	0.00	0.00	0.00	0.00	0.00	0.00
gc read waits	11	0.00	0.00	0.00	0.00	0.00	0.00
gc reader bypass grants	10,145	2.81	0.14	0.94	0.14	0.78	1.03
gc reader bypass waits	152	0.04	0.00	0.01	0.00	0.01	0.02
gc remote grants	419,807	116.37	5.83	38.79	10.00	32.02	50.27
gcs messages sent	3,358,575	931.01	46.63	310.34	59.48	270.99	378.76
ges messages sent	708,309	196.35	9.83	65.45	22.48	47.05	90.50
global enqueue get time	13,520	3.75	0.19	1.25	0.70	0.67	2.04
global enqueue gets async	29,763	8.25	0.41	2.75	0.52	2.42	3.35
global enqueue gets sync	3,237,588	897.48	44.95	299.16	286.80	75.17	622.40
global enqueue releases	2,312,093	640.92	32.10	213.64	158.66	69.42	383.60
heap block compress	177,072	49.09	2.46	16.36	22.18	3.27	41.98
hot buffers moved to head of LRU	1,736,851	481.46	24.11	160.49	144.99	51.46	325.03
immediate (CR) block cleanout applications	5,370	1.49	0.07	0.50	0.27	0.18	0.69
immediate (CURRENT) block cleanout applications	227,688	63.12	3.16	21.04	8.79	11.05	27.60
in call idle wait time	70,132,741	19,441.14	973.63	6,480.38	252.71	6,216.37	6,720.01
index crx upgrade (positioned)	6,819	1.89	0.09	0.63	0.68	0.20	1.41
index crx upgrade (prefetch)	14	0.00	0.00	0.00	0.00	0.00	0.00
index fast full scans (full)	125,700	34.84	1.75	11.61	1.54	10.04	13.11
index fast full scans (rowid ranges)	8,628	2.39	0.12	0.80	1.34	0.00	2.34
index fetch by key	23,161,634	6,420.52	321.55	2,140.17	730.32	1,303.78	2,651.65
index scans kdiixs1	63,776,782	17,679.27	885.40	5,893.09	3,253.33	2,967.61	9,396.75
java call heap collected bytes	270,416	74.96	3.75	74.96		74.96	74.96
java call heap collected count	3,887	1.08	0.05	1.08		1.08	1.08
java call heap gc count	1	0.00	0.00	0.00		0.00	0.00
java call heap live object count	762	0.21	0.01	0.21		0.21	0.21
java call heap live object count max	762	0.21	0.01	0.21		0.21	0.21
java call heap live size	44,224	12.26	0.61	12.26		12.26	12.26
java call heap live size max	44,224	12.26	0.61	12.26		12.26	12.26
java call heap object count	3,317	0.92	0.05	0.92		0.92	0.92
java call heap object count max	1,875	0.52	0.03	0.52		0.52	0.52
java call heap total size	2,097,152	581.34	29.11	581.34		581.34	581.34
java call heap total size max	2,097,152	581.34	29.11	581.34		581.34	581.34
java call heap used size	211,632	58.67	2.94	58.67		58.67	58.67
java call heap used size max	206,296	57.19	2.86	57.19		57.19	57.19
leaf node 90-10 splits	401	0.11	0.01	0.04	0.05	0.01	0.10
leaf node splits	7,674	2.13	0.11	0.71	0.23	0.49	0.95
lob reads	110,890	30.74	1.54	10.25	2.22	8.06	12.50
lob writes	47,664	13.21	0.66	4.40	1.65	2.53	5.63
lob writes unaligned	47,449	13.15	0.66	4.38	1.65	2.50	5.61
logical read bytes from cache	2,619,489,058,816	726,136,511.38	36,365,629.98	242,045,503.79	136,332,662.16	102,866,613.74	375,341,480.62
logons cumulative	16,967	4.70	0.24	1.57	1.05	0.95	2.78
messages received	302,795	83.94	4.20	27.98	12.47	17.69	41.84
messages sent	302,795	83.94	4.20	27.98	12.47	17.69	41.84
min active SCN optimization applied on CR	4,234	1.17	0.06	0.39	0.29	0.07	0.62
no buffer to keep pinned count	3,623	1.00	0.05	0.33	0.29	0.05	0.63
no work - consistent read gets	230,752,703	63,965.92	3,203.47	21,321.97	14,514.33	6,991.25	36,013.19
non-idle wait count	36,775,422	10,194.34	510.54	3,398.11	1,567.42	1,592.55	4,409.32
non-idle wait time	235,159	65.19	3.26	21.73	18.25	8.83	42.61
opened cursors cumulative	6,056,684	1,678.95	84.08	559.65	536.33	165.87	1,170.49
parse count (describe)	25	0.01	0.00	0.00	0.00	0.00	0.01
parse count (failures)	1,822	0.51	0.03	0.17	0.01	0.16	0.17
parse count (hard)	42,559	11.80	0.59	3.93	1.21	2.56	4.86
parse count (total)	1,936,809	536.89	26.89	178.96	46.63	145.81	232.28
parse time cpu	20,539	5.69	0.29	1.90	0.64	1.15	2.28
parse time elapsed	26,315	7.29	0.37	2.43	0.74	1.58	2.91
physical read IO requests	5,404,708	1,498.21	75.03	499.40	732.80	70.98	1,345.55
physical read bytes	72,372,666,368	20,062,070.24	1,004,729.38	6,687,356.75	6,605,703.87	1,625,758.16	14,159,865.10
physical read total IO requests	5,480,453	1,519.21	76.08	506.40	733.22	77.46	1,353.03
physical read total bytes	77,472,933,888	21,475,889.42	1,075,534.96	7,158,629.81	6,685,172.92	2,064,547.28	14,728,546.67
physical read total multi block requests	32,119	8.90	0.45	2.97	1.35	1.42	3.86
physical reads	8,834,554	2,448.98	122.65	816.33	806.36	198.46	1,728.50

physical reads cache	5,339,188	1,480.05	74.12	493.35	741.78	62.86	1,349.88
physical reads cache prefetch	311,670	86.40	4.33	28.80	17.16	17.63	48.56
physical reads direct	3,495,366	968.93	48.53	322.98	166.68	135.59	454.72
physical reads direct (lob)	217,876	60.40	3.02	20.13	2.65	17.75	22.99
physical reads direct temporary tablespace	22,168	6.15	0.31	6.15		6.15	6.15
physical write IO requests	177,942	49.33	2.47	16.44	2.54	14.45	19.30
physical write bytes	3,202,899,968	887,859.84	44,464.96	295,953.28	38,600.34	258,048.38	335,213.36
physical write total IO requests	390,181	108.16	5.42	36.05	9.31	27.03	45.63
physical write total bytes	14,097,189,888	3,907,810.12	195,707.32	1,302,603.37	271,539.45	1,097,466.29	1,610,531.98
physical write total multi block requests	22,214	6.16	0.31	2.05	0.35	1.83	2.45
physical writes	390,979	108.38	5.43	36.13	4.71	31.50	40.92
physical writes direct	27,590	7.65	0.38	2.55	3.70	0.33	6.82
physical writes direct (lob)	1,742	0.48	0.02	0.16	0.10	0.09	0.28
physical writes direct temporary tablespace	23,249	6.44	0.32	2.15	3.56	0.09	6.25
physical writes from cache	363,389	100.73	5.04	33.58	6.14	29.14	40.59
physical writes non checkpoint	357,499	99.10	4.96	33.03	6.19	26.76	39.14
pinned buffers inspected	767	0.21	0.01	0.07	0.09	0.02	0.18
pinned cursors current	8	0.00	0.00	0.00		0.00	0.00
prefetch clients - default	1	0.00	0.00	0.00		0.00	0.00
prefetched blocks aged out before use	161	0.04	0.00	0.01	0.02	0.00	0.04
process last non-idle time	10,828	3.00	0.15	1.00	0.00	1.00	1.00
queries parallelized	4,238	1.17	0.06	0.39	0.12	0.31	0.53
recursive aborts on index block reclamation	2	0.00	0.00	0.00		0.00	0.00
recursive calls	6,109,769	1,693.66	84.82	564.55	671.06	98.63	1,333.71
recursive cpu usage	121,795	33.76	1.69	11.25	10.71	4.14	23.57
redo KB read	3,530,596	978.70	49.01	326.23	81.85	263.18	418.74
redo blocks checksummed by FG (exclusive)	913,077	253.11	12.68	84.37	29.57	52.76	111.36
redo blocks written	6,865,534	1,903.16	95.31	634.39	155.40	498.72	803.93
redo blocks written (group 0)	6,855,842	1,900.47	95.18	633.49	155.37	497.61	802.89
redo blocks written (group 1)	5,384	1.49	0.07	0.50	0.26	0.20	0.66
redo buffer allocation retries	364	0.10	0.01	0.03	0.06	0.00	0.10
redo entries	9,749,070	2,702.49	135.34	900.83	262.13	700.28	1,197.43
redo log space requests	402	0.11	0.01	0.04	0.06	0.00	0.11
redo log space wait time	280	0.08	0.00	0.03	0.04	0.00	0.08
redo ordering marks	111	0.03	0.00	0.03		0.03	0.03
redo size	3,373,216,348	935,071.97	46,829.41	311,690.66	77,975.40	242,809.78	396,347.95
redo size for direct writes	14,588,596	4,044.04	202.53	1,348.01	834.16	802.86	2,308.28
redo subscn max counts	161,075	44.65	2.24	14.88	6.90	6.94	19.38
redo synch long waits	756	0.21	0.01	0.07	0.08	0.01	0.16
redo synch time	3,319	0.92	0.05	0.31	0.10	0.20	0.40
redo synch time (usec)	34,570,838	9,583.20	479.94	3,194.40	1,024.28	2,122.30	4,163.01
redo synch time overhead (usec)	618,859,028	171,551.54	8,591.45	57,183.85	96,518.12	772.37	168,630.54
redo synch time overhead count (2ms)	37,173	10.30	0.52	3.43	1.26	2.25	4.75
redo synch time overhead count (8ms)	607	0.17	0.01	0.06	0.05	0.03	0.11
redo synch time overhead count (32ms)	19	0.01	0.00	0.00	0.00	0.00	0.00
redo synch time overhead count (128ms)	9	0.00	0.00	0.00	0.00	0.00	0.00
redo synch time overhead count (inf)	5	0.00	0.00	0.00	0.00	0.00	0.00
redo synch writes	37,830	10.49	0.53	3.50	1.26	2.28	4.79
redo wastage	27,517,692	7,628.05	382.02	2,542.68	1,472.84	1,553.82	4,235.40
redo write broadcast ack count	2,085	0.58	0.03	0.19	0.07	0.13	0.27
redo write broadcast ack time	4,803,449	1,331.54	66.68	443.85	112.35	314.24	513.55
redo write finish time	71,893,435	19,929.24	998.08	6,643.08	2,562.61	4,470.99	9,469.38
redo write gather time	6,345,423	1,758.98	88.09	586.33	213.11	409.38	822.89
redo write info find	37,819	10.48	0.53	3.49	1.26	2.28	4.79
redo write info find fail	6	0.00	0.00	0.00	0.00	0.00	0.00
redo write schedule time	6,525,655	1,808.95	90.59	602.98	217.16	421.56	843.59
redo write size count (4KB)	73,215	20.30	1.02	6.77	5.38	3.52	12.97
redo write size count (8KB)	2,672	0.74	0.04	0.25	0.07	0.18	0.31

redo write size count (16KB)	2,882	0.80	0.04	0.27	0.05	0.22	0.31
redo write size count (32KB)	3,719	1.03	0.05	0.34	0.31	0.13	0.70
redo write size count (128KB)	6,815	1.89	0.09	0.63	0.53	0.15	1.19
redo write size count (256KB)	706	0.20	0.01	0.07	0.02	0.05	0.08
redo write size count (512KB)	661	0.18	0.01	0.06	0.02	0.05	0.08
redo write size count (1024KB)	868	0.24	0.01	0.08	0.02	0.07	0.10
redo write size count (inf)	1,607	0.45	0.02	0.15	0.02	0.13	0.17
redo write time	7,196	1.99	0.10	0.66	0.26	0.45	0.95
redo write time (usec)	71,963,791	19,948.74	999.05	6,649.58	2,564.63	4,475.70	9,478.07
redo write total time	73,484,747	20,370.36	1,020.17	6,790.12	2,630.57	4,549.94	9,686.74
redo write worker delay (usec)	2,659,204	737.15	36.92	245.72	79.92	181.13	335.09
redo writes	93,145	25.82	1.29	8.61	4.74	5.09	14.00
redo writes (group 0)	93,096	25.81	1.29	8.60	4.74	5.08	13.99
redo writes (group 1)	20	0.01	0.00	0.00	0.00	0.00	0.00
redo writes adaptive all	93,145	25.82	1.29	8.61	4.74	5.09	14.00
redo writes adaptive worker	93,116	25.81	1.29	8.60	4.74	5.08	13.99
rollback changes - undo records applied	659	0.18	0.01	0.06	0.09	0.01	0.16
rollbacks only - consistent read gets	1,982	0.55	0.03	0.18	0.17	0.06	0.38
root node splits	1	0.00	0.00	0.00	0.00	0.00	0.00
rows fetched via callback	8,918,544	2,472.26	123.81	824.09	475.72	499.71	1,370.19
session cursor cache hits	563,611	156.24	7.82	52.08	24.41	34.40	79.93
session logical reads	323,229,777	89,601.04	4,487.31	29,867.01	16,800.59	12,691.87	46,266.16
shared hash latch upgrades - no wait	8,641,921	2,395.58	119.97	798.53	1,067.77	121.92	2,029.46
shared hash latch upgrades - wait	9,070	2.51	0.13	0.84	1.45	0.00	2.51
sorts (disk)	2	0.00	0.00	0.00	0.00	0.00	0.00
sorts (memory)	2,510,935	696.04	34.86	232.01	85.46	139.30	307.64
sorts (rows)	174,209,759	48,291.85	2,418.51	16,097.28	5,090.34	12,036.78	21,807.99
sql area evicted	41,292	11.45	0.57	3.82	1.26	2.45	4.94
sql area purged	1,834	0.51	0.03	0.17	0.01	0.16	0.18
summed dirty queue length	126,109	34.96	1.75	11.65	19.81	0.14	34.53
switch current to new buffer	84,565	23.44	1.17	7.81	5.80	1.19	11.97
table fetch by rowid	294,078,276	81,519.95	4,082.61	27,173.32	35,038.99	5,401.69	67,592.63
table fetch continued row	339,794	94.19	4.72	31.40	2.54	28.76	33.81
table scan blocks gotten	116,843,547	32,389.71	1,622.11	10,796.57	12,206.32	2,516.67	24,814.61
table scan disk non-IMC rows gotten	10,844,960,432	3,006,288.18	150,557.54	1,002,096.06	1,457,312.71	120,922.92	2,684,218.28
table scan rows gotten	10,891,554,343	3,019,204.25	151,204.39	1,006,401.42	1,456,967.31	124,413.94	2,688,091.46
table scans (direct read)	69	0.02	0.00	0.01	0.00	0.00	0.01
table scans (long tables)	2,507	0.69	0.03	0.23	0.38	0.01	0.67
table scans (rowid ranges)	2,448	0.68	0.03	0.34	0.47	0.01	0.67
table scans (short tables)	981,589	272.10	13.63	90.70	97.54	26.10	202.90
temp space allocated (bytes)	8,388,608	2,325.36	116.46	1,162.68	411.06	872.02	1,453.35
total cf enq hold time	6,590	1.83	0.09	0.61	0.05	0.55	0.65
total number of cf enq holders	1,305	0.36	0.02	0.12	0.01	0.11	0.13
total number of times SMON posted	862	0.24	0.01	0.08	0.12	0.01	0.22
transaction rollbacks	77	0.02	0.00	0.01	0.00	0.00	0.01
transaction tables consistent reads - undo records applied	542	0.15	0.01	0.15	0.00	0.15	0.15
undo change vector size	1,421,110,584	393,938.78	19,728.88	131,312.93	35,565.24	102,675.70	171,123.08
user I/O wait time	152,158	42.18	2.11	14.06	17.86	3.53	34.68
user calls	14,025,084	3,887.82	194.71	1,295.94	450.09	1,012.18	1,814.90
user commits	71,836	19.91	1.00	6.64	4.95	3.13	12.30
user logons cumulative	2,032	0.56	0.03	0.19	0.01	0.18	0.20
user logouts cumulative	2,025	0.56	0.03	0.19	0.01	0.18	0.20
user rollbacks	196	0.05	0.00	0.02	0.01	0.01	0.03
workarea executions - optimal	1,671,161	463.25	23.20	154.42	87.58	65.68	240.78
write clones created in background	35	0.01	0.00	0.00	0.00	0.00	0.01
write clones created in foreground	84	0.02	0.00	0.01	0.00	0.01	0.01

[Back to Global Activity Statistics](#)

[Back to Top](#)

Global Messaging Statistics (Global)

- per Second Average - average of per-instance per Second rates
- per Second Std Dev - standard deviation of per-instance per Second rates
- per Second Min - minimum of per-instance per Second rates
- per Second Max - maximum of per-instance per Second rates

Statistic	Total	per Second					
		per Second	per Trans	Average	Std Dev	Min	Max
acks for commit broadcast(actual)	173,200	48.01	2.40	16.00	4.67	10.70	19.54
acks for commit broadcast(logical)	176,639	48.97	2.45	16.32	4.73	10.94	19.83
broadcast msgs on commit(actual)	176,673	48.97	2.45	16.32	9.52	9.31	27.16
broadcast msgs on commit(logical)	185,970	51.55	2.58	17.18	9.48	10.14	27.96
broadcast msgs on commit(wasted)	5,322	1.48	0.07	0.49	0.10	0.39	0.58
flow control messages received	56	0.02	0.00	0.01	0.00	0.00	0.01
flow control messages sent	56	0.02	0.00	0.01	0.01	0.00	0.01
gcs assume cvt	21	0.01	0.00	0.00	0.00	0.00	0.00
gcs assume no cvt	438,834	121.65	6.09	40.55	6.29	34.11	46.69
gcs ast xid	6	0.00	0.00	0.00	0.00	0.00	0.00
gcs blocked converts	193,060	53.52	2.68	17.84	0.53	17.47	18.45
gcs blocked cr converts	217,690	60.34	3.02	20.11	4.03	15.78	23.75
gcs compatible bastas	3,750	1.04	0.05	0.35	0.06	0.28	0.40
gcs compatible cr bastas (global)	78,523	21.77	1.09	7.26	1.14	5.98	8.18
gcs compatible cr bastas (local)	444,318	123.17	6.17	41.06	2.12	38.62	42.49
gcs dbwr flush pi msgs	132,887	36.84	1.84	12.28	2.60	10.32	15.23
gcs dbwr write request msgs	44,199	12.25	0.61	4.08	1.53	3.17	5.85
gcs force cr block only	20	0.01	0.00	0.00	0.00	0.00	0.00
gcs force cr grant	15	0.00	0.00	0.00	0.00	0.00	0.00
gcs immediate (compatible) converts	28,489	7.90	0.40	2.63	1.06	1.99	3.85
gcs immediate (null) converts	70,431	19.52	0.98	6.51	0.83	5.55	7.00
gcs immediate cr (compatible) converts	97,194	26.94	1.35	8.98	13.58	1.09	24.66
gcs immediate cr (null) converts	345,560	95.79	4.80	31.93	2.25	30.10	34.44
gcs indirect ast	274,593	76.12	3.81	25.37	5.25	20.68	31.04
gcs indirect bidless ast	6,532	1.81	0.09	0.60	0.05	0.55	0.64
gcs indirect fg ast	274,429	76.07	3.81	25.36	5.24	20.68	31.02
gcs lms flush pi msgs	17,907	4.96	0.25	1.65	1.08	0.75	2.85
gcs lms write request msgs	31,857	8.83	0.44	2.94	0.83	1.98	3.43
gcs msgs process time(ms)	54,810	15.19	0.76	5.06	0.62	4.51	5.73
gcs msgs received	3,366,051	933.09	46.73	311.03	23.62	291.03	337.09
gcs new served by master	4,012	1.11	0.06	0.37	0.03	0.34	0.41
gcs out-of-order msgs	207	0.06	0.00	0.02	0.01	0.01	0.03
gcs pings refused	28,492	7.90	0.40	2.63	1.14	1.82	3.93
gcs queued converts	29	0.01	0.00	0.00	0.00	0.00	0.00
gcs reader bypass N->Xw ping local	687	0.19	0.01	0.06	0.03	0.03	0.08
gcs reader bypass N->Xw ping remote	437	0.12	0.01	0.04	0.03	0.01	0.08
gcs reader bypass grant X on assume	911	0.25	0.01	0.08	0.05	0.03	0.13
gcs reader bypass grant fg ast	6,913	1.92	0.10	0.64	0.11	0.51	0.73
gcs reader bypass grant immediate	9,547	2.65	0.13	0.88	0.05	0.84	0.93
gcs refuse xid	6	0.00	0.00	0.00	0.00	0.00	0.00
gcs regular cr	223,837	62.05	3.11	20.68	17.61	10.41	41.01
gcs retry convert request	420	0.12	0.01	0.04	0.00	0.04	0.04
gcs side channel msgs actual	71,463	19.81	0.99	6.60	1.40	5.74	8.22
gcs side channel msgs logical	1,130,711	313.44	15.70	104.48	36.63	73.18	144.77
gcs stale cr	372	0.10	0.01	0.03	0.01	0.03	0.05
gcs undo cr	145	0.04	0.00	0.01	0.02	0.00	0.03
gcs write notification msgs	2,646	0.73	0.04	0.24	0.20	0.07	0.46
gcs writes refused	2,583	0.72	0.04	0.24	0.01	0.23	0.25
ges msgs process time(ms)	5,444	1.51	0.08	0.50	0.03	0.47	0.53
ges msgs received	700,958	194.31	9.73	64.77	2.30	62.22	66.70
global posts queue time	1	0.00	0.00	0.00		0.00	0.00
global posts queued	2	0.00	0.00	0.00		0.00	0.00
global posts requested	2	0.00	0.00	0.00		0.00	0.00
global posts sent	2	0.00	0.00	0.00		0.00	0.00
implicit batch messages received	137,918	38.23	1.91	12.74	3.07	9.79	15.92
implicit batch messages sent	137,911	38.23	1.91	12.74	5.03	9.60	18.55
messages flow controlled	38,793	10.75	0.54	3.58	1.70	2.43	5.54
messages queue sent actual	1,113,984	308.80	15.47	102.93	29.42	70.02	126.70
messages queue sent logical	585,257	162.24	8.12	54.08	23.39	38.98	81.02
messages received actual	2,771,940	768.40	38.48	256.13	26.83	238.30	286.99
messages received logical	4,067,009	1,127.40	56.46	375.80	23.59	357.73	402.48
messages sent directly	1,239,738	343.66	17.21	114.55	26.52	84.50	134.66
messages sent indirectly	1,534,089	425.26	21.30	141.75	29.68	119.02	175.33
messages sent not implicit batched	814	0.23	0.01	0.08	0.01	0.07	0.09

messages sent pbatched	2,249,204	623.49	31.23	207.83	25.75	189.05	237.19
msgs received queue time (ms)	70,425	19.52	0.98	6.51	2.04	4.84	8.78
msgs received queued	4,067,035	1,127.40	56.46	375.80	23.58	357.73	402.48
msgs sent queue time (ms)	3,338,209	925.37	46.34	308.46	182.47	188.84	518.48
msgs sent queue time on ksxp (ms)	884,098	245.08	12.27	81.69	14.58	70.67	98.22
msgs sent queued	1,372,295	380.41	19.05	126.80	36.39	88.58	161.03
msgs sent queued on ksxp	2,771,840	768.37	38.48	256.12	37.09	221.40	295.20
number of directly freed master res	136,703	37.89	1.90	12.63	12.02	5.67	26.51
process batch messages received	1,022,679	283.49	14.20	94.50	25.55	78.62	123.97
process batch messages sent	1,212,524	336.12	16.83	112.04	16.70	95.81	129.16

[Back to Global Activity Statistics](#)
[Back to Top](#)

System Statistics (Absolute Values)

Sessions		Open Cursors		Session Cached Cursors		
#	Begin	End	Begin	End	Begin	End
1	124	134	1,540	1,617	5,485,841	5,511,577
2	125	127	1,612	1,666	4,858,245	4,876,254
3	120	127	1,886	1,774	4,380,031	4,398,722
Sum	369	388	5,038	5,057	14,724,117	14,786,553
Avg	123	129	1,679	1,686	4,908,039	4,928,851
Std	3	4	183	80	554,584	558,289

[Back to Global Activity Statistics](#)
[Back to Top](#)

I/O Statistics

- [IOStat by Function \(per Second\)](#)
- [IOStat by File Type \(per Second\)](#)
- [Segment Statistics \(Global\)](#)

[Back to Top](#)

IOStat by Function (per Second)

- Total Reads includes all Functions: Buffer Cache, Direct Reads, ARCH, Data Pump, Others, RMAN, Recovery, Streams/AQ and XDB
- Total Writes includes all Functions: DBWR, Direct Writes, LGWR, ARCH, Data Pump, Others, RMAN, Recovery, Streams/AQ and XDB

#	Reads MB/sec			Writes MB/sec				Reads requests/sec				Writes requests/sec						
	Total	Buffer	Cache Direct	Reads	Total	DBWR	Direct	Writes	LGWR	Total	Buffer	Cache	Direct	Reads	Total	DBWR	Direct	Writes
1	4.47		0.54	3.55	1.05	0.23	0.05	0.49	88.75	56.47			24.94	45.61	14.68		0.56	28.48
2	1.97		0.52	1.06	1.14	0.24	0.00	0.59	77.44	51.43			19.59	27.00	13.96		0.20	10.94
3	14.04		10.56	2.96	1.54	0.32	0.00	0.79	1,352.34	1,318.56			26.47	35.51	18.98		0.10	14.50
Sum	20.48		11.62	7.57	3.73	0.79	0.05	1.86	1,518.54	1,426.46			70.99	108.13	47.62		0.86	53.92
Avg	6.83		3.87	2.52	1.24	0.26	0.02	0.62	506.18	475.49			23.66	36.04	15.87		0.29	17.97

[Back to I/O Statistics](#)
[Back to Top](#)

IOStat by File Type (per Second)

- Total Reads includes all Filetypes: Data File, Temp File, Archive Log, Backups, Control File, Data Pump Dump File, Flashback Log, Log File, Other, etc
- Total Writes includes all Filetypes: Data File, Temp File, Log File, Archive Log, Backup, Control File, Data Pump Dump File, Flashback Log, Log File, Other, etc

#	Reads MB/sec				Writes MB/sec				Reads requests/sec				Writes requests/sec					
	Total	Data	File	Temp	Total	Data	File	Temp	File	Log	File	Total	Data	File	Temp	File	Log	File
1	4.47	4.03		0.05	1.05	0.23	0.05	0.49	88.75	80.35		1.35	45.61		15.09		0.49	28.39
2	1.97	1.55		0.00	1.14	0.24	0.00	0.59	77.45	70.61		0.40	27.00		14.23		0.22	10.86
3	14.02	13.49		0.00	1.52	0.32	0.00	0.79	1,352.09	1,344.36		0.37	35.48		19.09		0.21	14.39
Sum	20.45	19.07		0.06	3.70	0.79	0.05	1.86	1,518.29	1,495.32		2.13	108.10		48.41		0.91	53.64
Avg	6.82	6.36		0.02	1.23	0.26	0.02	0.62	506.10	498.44		0.71	36.03		16.14		0.30	17.88

[Back to I/O Statistics](#)
[Back to Top](#)

Segment Statistics (Global)

- % Total shows % of statistic for each segment compared to the global cluster-wide total (logical reads, physical reads, gc [cr/cu] blocks [recv/serv])
- % Capture shows % of statistic for each segment compared to the total captured by AWR for all segments during the snapshot interval
- Captured Segments account for 79.1% of Total Logical Reads: 323,229,777
- Captured Segments account for 72.2% of Total Physical Reads: 8,834,554
- Captured Segments account for 87.1% of Total Physical Read Requests: 5,404,708
- Captured Segments account for 87.1% of Total UnOptimized Read Requests: 5,404,708
- Captured Segments account for 0.0% of Total Optimized Read Requests: 1
- Captured Segments account for 46.5% of Total Direct Physical Reads: 3,495,366
- Captured Segments account for 27.5% of Total Physical Writes: 390,979
- Captured Segments account for 48.5% of Total Physical Write Requests: 177,942
- Captured Segments account for 5.6% of Total Direct Physical Writes: 27,590
- Captured Segments account for 67.7% of Total Table Scans: 128,207
- Captured Segments account for 84.9% of Total CR Blocks Received: 217,717
- Captured Segments account for 61.7% of Total Current Blocks Received: 632,190
- Captured Segments account for 84.9% of Total CR Blocks Served: 217,735
- Captured Segments account for 61.7% of Total Current Blocks Served: 632,232
- When ** MISSING ** occurs, some of the object attributes may not be available

Statistic	Owner	Tablespace Name	Object Name	Subobject Name	Obj. Type	Obj#	Dataobj#	Value	%Total	%Capture
buffer busy waits	SYSADM	AAAPP	PS_SAA_ADB_CRSEAVL		TABLE	95981	95981	116		90.63
buffer busy waits	SYSADM	PTLOCK	PSLOCK		TABLE	73454	73454	3		2.34
buffer busy waits	SYS	SYSTEM	SEG\$		TABLE	14	8	2		1.56
buffer busy waits	SYSADM	PTLOCK	PSVERSION		TABLE	74453	74453	2		1.56
buffer busy waits	SYS	SYSTEM	AUD\$		TABLE	384	1356279	1		0.78
buffer busy waits	SYSADM	PSINDEX	PS_PSIBLOGHDR		INDEX	141377	1324769	1		0.78
buffer busy waits	SYSADM	PSINDEX	PS_SAA_ADB_CRSEAVL		INDEX	814591	814591	1		0.78
buffer busy waits	SYSADM	SAAPP	PS_SAA_ADB_RESULTS		TABLE	95986	95986	1		0.78
buffer busy waits	SYSADM	URIWORK	PS_STDNT_ENRL_URI0		INDEX	769178	1971222	1		0.78
db block changes	SYSADM	AAAPP	PS_SAA_ADB_CRSEAVL		TABLE	95981	95981	1,726,080		33.13
db block changes	SYSADM	PSINDEX	PS0SAA_ADB_CRSEAVL		INDEX	814592	814592	908,128		17.43
db block changes	SYSADM	PSINDEX	PS_DYNROLE_TMP		INDEX	135844	135844	861,680		16.54
db block changes	SYSADM	PSINDEX	PS1SAA_ADB_CRSEAVL		INDEX	814593	814593	859,264		16.49
db block changes	SYSADM	PSINDEX	PS2SAA_ADB_CRSEAVL		INDEX	814594	814594	854,288		16.40
gc buffer busy	SYSADM	SAAPP	PS_ITEM_SF		TABLE	91349	91349	17,565		82.31
gc buffer busy	SYSADM	PSINDEX	PSGITEM_SF		INDEX	161808	2016180	3,667		17.18
gc buffer busy	SYSADM	PTLOCK	PSVERSION		TABLE	74453	74453	52		0.24
gc buffer busy	SYSADM	AAAPP	PS_SAA_ADB_CRSEAVL		TABLE	95981	95981	30		0.14
gc buffer busy	SYSADM	PTTBL	PSOPRDEFN		TABLE	809756	809756	27		0.13
gc cr blocks received	SYSADM	PSINDEX	PS_DYNROLE_TMP		INDEX	135844	135844	145,501	66.83	86.61
gc cr blocks received	SYSADM	AAAPP	PS_SAA_ADB_CRSEAVL		TABLE	95981	95981	6,228	2.86	3.71
gc cr	SYSADM	PSINDEX	PS_SAA_ADB_CRSEAVL		INDEX	814591	814591	6,212	2.85	3.70

blocks received	gc cr blocks received	SYSADM URI_PS_TDE PS_ISIR_00_1_EC	TABLE	91292	729462	5,323	2.44	3.17
gc cr blocks received	gc cr blocks received	SYSADM SRAPP PS_SSR_REGFORM	TABLE	100948	100948	4,737	2.18	2.82
gc cr blocks served	gc cr blocks served	SYSADM PSINDEX PS_DYNROLE_TMP	INDEX	135844	135844	145,501	66.82	86.61
gc cr blocks served	gc cr blocks served	SYSADM AAAPP PS_SAA_ADB_CRSEAVL	TABLE	95981	95981	6,228	2.86	3.71
gc cr blocks served	gc cr blocks served	SYSADM PSINDEX PS_SAA_ADB_CRSEAVL	INDEX	814591	814591	6,212	2.85	3.70
gc cr blocks served	gc cr blocks served	SYSADM URI_PS_TDE PS_ISIR_00_1_EC	TABLE	91292	729462	5,323	2.44	3.17
gc cr blocks served	gc cr blocks served	SYSADM SRAPP PS_SSR_REGFORM	TABLE	100948	100948	4,737	2.18	2.82
gc cu blocks received	gc cu blocks received	SYSADM PSINDEX PS_DYNROLE_TMP	INDEX	135844	135844	73,474	11.62	45.08
gc cu blocks received	gc cu blocks received	SYSADM SAAPP PS_ITEM_SF	TABLE	91349	91349	34,526	5.46	21.18
gc cu blocks received	gc cu blocks received	SYSADM PTTBL PSOPRDEFN	TABLE	809756	809756	24,721	3.91	15.17
gc cu blocks received	gc cu blocks received	SYSADM AAAPP PS_SAA_ADB_CRSEAVL	TABLE	95981	95981	17,876	2.83	10.97
gc cu blocks received	gc cu blocks received	SYSADM PSINDEX PS_PERS_DATA_EFFDT	INDEX	842104	842104	12,386	1.96	7.60
gc cu blocks served	gc cu blocks served	SYSADM PSINDEX PS_DYNROLE_TMP	INDEX	135844	135844	73,474	11.62	45.08
gc cu blocks served	gc cu blocks served	SYSADM SAAPP PS_ITEM_SF	TABLE	91349	91349	34,526	5.46	21.18
gc cu blocks served	gc cu blocks served	SYSADM PTTBL PSOPRDEFN	TABLE	809756	809756	24,721	3.91	15.17
gc cu blocks served	gc cu blocks served	SYSADM AAAPP PS_SAA_ADB_CRSEAVL	TABLE	95981	95981	17,876	2.83	10.97
gc cu blocks served	gc cu blocks served	SYSADM PSINDEX PS_PERS_DATA_EFFDT	INDEX	842104	842104	12,386	1.96	7.60
logical reads	logical reads	SYSADM URI_PS_TDE PS_U_CAM011_RESULT	TABLE	1698857	1698857	78,236,912	24.20	61.98
logical reads	logical reads	SYSADM PSINDEX PSGITEM_SF	INDEX	161808	2016180	20,120,080	6.22	15.94
logical reads	logical reads	SYSADM SAAPP PS_ITEM_SF	TABLE	91349	91349	10,252,848	3.17	8.12
logical reads	logical reads	SYSADM PSINDEX PS_ACAD_PROG	INDEX	815142	815142	9,046,832	2.80	7.17
logical reads	logical reads	SYSADM PSINDEX PS_FERPA_OVERRIDE	INDEX	139808	139808	8,579,376	2.65	6.80
physical reads	physical reads	SYSADM SAAPP PS_ITEM_SF	TABLE	91349	91349	4,249,843	48.10	69.30
physical reads	physical reads	SYSADM SRAPP PS_GRADE_ROSTER	TABLE	88408	88408	833,180	9.43	13.59
physical reads	physical reads	SYS SYSTEM AUD\$	TABLE	384	1356279	791,325	8.96	12.90
physical reads	physical reads	SYSADM PSINDEX PSGITEM_SF	INDEX	161808	2016180	240,259	2.72	3.92
physical reads	physical reads	SYSADM PSINDEX PS_PSRULEUSER	INDEX	135815	1525548	17,702	0.20	0.29
physical reads	physical reads	SYSADM SRAPP PS_GRADE_ROSTER	TABLE	88408	88408	833,150		51.26

direct										
physical reads	SYS	SYSTEM	AUD\$	TABLE	384	1356279	791,320			48.69
direct										
physical reads	SYSADM	PSIMAGE2	SYS_LOB0001433542C00007\$\$	LOB	1433543	1433543	397			0.02
direct										
physical reads	SYSADM	PSIMAGE2	SYS_LOB0001435984C00008\$\$	LOB	1435985	1435985	295			0.02
direct										
physical reads	SYSADM	PSIMAGE2	SYS_LOB0001433602C00003\$\$	LOB	1433603	1433603	28			0.00
direct										
physical read requests	SYSADM	SAAPP	PS_ITEM_SF	TABLE	91349	91349	4,214,052	77.97		93.61
physical read requests	SYSADM	PSINDEX	PSGITEM_SF	INDEX	161808	2016180	240,259	4.45		5.34
physical read requests	SYSADM	PSINDEX	PS_PSRULEUSER	INDEX	135815	1525548	17,702	0.33		0.39
physical read requests	SYSADM	URI_PS_TDE	PS_ISIR_00_1_EC	TABLE	91292	729462	15,497	0.29		0.34
physical read requests	SYSADM	SAAPP	PS_ITEM_DUE_SF	TABLE	91340	91340	14,251	0.26		0.32
physical writes	SYSADM	AAAPP	PS_SAA_ADB_CRSEAVL	TABLE	95981	95981	17,417			29.47
physical writes	SYSADM	PSINDEX	PS_DYNROLE_TMP	INDEX	135844	135844	15,745			26.64
physical writes	SYSADM	PSINDEX	PS0SAA_ADB_CRSEAVL	INDEX	814592	814592	10,531			17.82
physical writes	SYSADM	URI_PS_TDE	PS_ISIR_00_1_EC	TABLE	91292	729462	7,773			13.15
physical writes	SYSADM	PSINDEX	PS1SAA_ADB_CRSEAVL	INDEX	814593	814593	7,641			12.93
physical writes direct	SYSADM	PSIMAGE2	SYS_LOB0001435984C00008\$\$	LOB	1435985	1435985	547			35.85
physical writes direct	SYSADM	PSIMAGE2	SYS_LOB0001433542C00007\$\$	LOB	1433543	1433543	465			30.47
physical writes direct	SYS	SYSAUX	SYS_LOB0000886039C00009\$\$	SYS_LOB_P75396	LOB PARTITION	2016016	2016016	390		25.56
physical writes direct	SYSADM	PSIMAGE2	SYS_LOB0001433602C00003\$\$	LOB	1433603	1433603	84			5.50
physical writes direct	SYS	SYSAUX	SYS_LOB0000006339C00038\$\$	LOB	6340	6340	40			2.62
physical write requests	SYSADM	AAAPP	PS_SAA_ADB_CRSEAVL	TABLE	95981	95981	14,033	7.89		27.42
physical write requests	SYSADM	PSINDEX	PS_DYNROLE_TMP	INDEX	135844	135844	13,570	7.63		26.52
physical write requests	SYSADM	PSINDEX	PS0SAA_ADB_CRSEAVL	INDEX	814592	814592	9,832	5.53		19.21
physical write requests	SYSADM	PSINDEX	PS1SAA_ADB_CRSEAVL	INDEX	814593	814593	7,061	3.97		13.80
physical write requests	SYSADM	PSINDEX	PS2SAA_ADB_CRSEAVL	INDEX	814594	814594	6,682	3.76		13.06
row lock waits	SYSADM	PTLOCK	PSLOCK	TABLE	73454	73454	31			53.45
row lock waits	SYSADM	PSINDEX	PS2SAA_RPT_ERROR	INDEX	814625	814625	3			5.17
row lock waits	** MISSING	SYSAUX	** MISSING: 2015973/2015973	** MISSING **	UNDEFINED	2015973	2015973	2		3.45

row lock waits	SYSADM PSINDEX	PS0SAA_ADB_COURSES	INDEX	814588	814588	2	3.45
row lock waits	SYSADM PSINDEX	PS0SAA_ADB_CRSEAVL	INDEX	814592	814592	2	3.45
row lock waits	SYSADM PSINDEX	PS1ENRL_REQ_DETAIL	INDEX	135632	1971232	2	3.45
row lock waits	SYSADM PSINDEX	PS1SAA_ADB_RESULTS	INDEX	814606	814606	2	3.45
row lock waits	SYSADM PSINDEX	PS1SAA_RPT_ERROR	INDEX	814624	814624	2	3.45
row lock waits	SYSADM PSINDEX	PS1SA_REQUEST_HDR	INDEX	814577	814577	2	3.45
row lock waits	SYSADM PSINDEX	PS2SAA_ADB_RESULTS	INDEX	814607	814607	2	3.45
row lock waits	SYSADM PSINDEX	PSBSTDNT_ENRL	INDEX	135660	1971211	2	3.45
row lock waits	SYSADM PSINDEX	PS_PSACCESSLOG	INDEX	826934	826934	2	3.45
row lock waits	SYSADM URIWORK	PS_SA_REQUEST_HDR_URI1	INDEX	710663	824850	2	3.45
row lock waits	SYSADM PSINDEX	PS0SAA_ADB_WHATIF	INDEX	814613	814613	1	1.72
row lock waits	SYSADM SAAPP	PS_RUN_CNTL_SF	TABLE	95722	95722	1	1.72
table scans	SYSADM PSINDEX	PS_PSMACRO	INDEX	809718	809718	48,492	55.94
table scans	SYSADM PSINDEX	PS3FACILITY_TBL	INDEX	813985	813985	20,163	23.26
table scans	SYSADM PSINDEX	PSAPSRROLECLASS	INDEX	1436294	1436294	15,173	17.50
table scans	SYSADM CCAPP	PS_EMAIL_ADDRESSES	TABLE	77836	77836	2,407	2.78
table scans	SYSADM PSINDEX	PS_DYNROLE_TMP	INDEX	135844	135844	456	0.53

[Back to I/O Statistics](#)
[Back to Top](#)

Library Cache Activity

- Summarized for all instances in the report
- "Pct Misses" should be very low

Namespace	Get Requests	Pct Miss	Pin Requests	Pct Miss	Reloads	Invali-dations	GES Lock Requests	GES Pin Requests	GES Pin Releases	GES Inval Requests	GES Invali-dations
ACCOUNT_STATUS	8,132	0.05	0		0	0	8,132	0	0	0	0
BODY	3,863	0.18	20,082	0.11	16	0	0	18,945	18,945	0	0
CLUSTER	87	2.30	87	2.30	0	0	87	87	87	0	0
DBLINK	775,180	0.00	0		0	0	775,180	0	0	0	0
DIRECTORY	32	12.50	64	12.50	0	0	0	60	60	0	0
EDITION	4,399	0.00	4,399	0.00	0	0	4,399	4,399	4,399	0	0
HINTSET OBJECT	28	14.29	28	28.57	0	0	28	28	28	0	0
INDEX	133	2.26	135	17.04	20	0	135	135	135	0	0
LOCATION	92	3.26	92	3.26	0	0	0	92	92	0	0
OBJECT ID	1	100.00	0		0	0	0	0	0	0	0
QUEUE	852	0.00	5,696	0.11	2	0	426	5,695	5,695	0	0
RULESET	0		1	100.00	1	0	0	1	1	0	0
SCHEMA	3,078	0.03	0		0	0	2,072	0	0	0	0
SQL AREA	1,196,404	6.28	14,683,136	0.84	4,515	2,287	0	0	0	0	0
SQL AREA BUILD	42,299	94.97	0		0	0	0	0	0	0	0
SQL AREA STATS	42,022	92.58	42,022	92.58	0	0	0	0	0	0	0
SUBSCRIPTION	0		23	100.00	23	0	0	23	23	0	0
TABLE/PROCEDURE	675,168	0.12	1,111,809	0.65	3,883	0	268,642	847,999	847,999	0	0
TRANSFORMATION	360	0.00	360	0.00	0	0	360	360	360	0	0
TRIGGER	41	14.63	41	17.07	1	0	0	41	41	0	0
USER AGENT	2	50.00	1	100.00	0	0	1	1	1	0	0

[Back to Top](#)

Memory Statistics

- [PGA Aggregate Target Statistics](#)
- [Process Memory Summary](#)

[Back to Top](#)

PGA Aggregate Target Statistics

- all stats are reported in MegaBytes

#	PGA Aggr Target		Auto PGA Target		PGA Mem Alloc		Auto Workareas		Manual Workarea		Global Mem Bound	
	Begin	End	Begin	End	Begin	End	Begin	End	Begin	End	Begin	End
1	4,480.00	4,480.00	3,063.57	3,052.36	1,325.22	1,402.83	0.00	0.76	0.00	0.00	448.00	448.00
2	4,480.00	4,480.00	3,462.72	3,461.48	939.53	960.82	0.00	0.00	0.00	0.00	448.00	448.00
3	4,480.00	4,480.00	3,326.65	3,333.59	1,054.10	1,049.34	0.77	0.77	0.00	0.00	448.00	448.00

[Back to Memory Statistics](#)

[Back to Top](#)

Process Memory Summary

- Max Alloc is Maximum PGA allocation size at snapshot time
- Hist Alloc is the Historical Maximum Allocation for still-connected processes
- Num Procs or Allocs: For Begin/End snapshot lines, it is the number of processes For Category lines, it is the number of allocations
- Allocation sizes are displayed in MegaBytes
- ordered by instance, Allocated Total (End) desc

#	Category	Allocated		Used		Avg Alloc		Std Dev		Max Alloc		Hist Max		Num Procs		Num Allocs	
		Begin	End	Begin	End	Begin	End	Begin	End	Begin	End	Begin	End	Begin	End	Begin	End
1	Other	787.05	812.62			4.71	4.59	8.40	8.22	59.63	59.63	59.63	59.63	167	177	167	177
1	PL/SQL	382.17	382.21	374.55	374.52	2.43	2.34	30.27	29.71	379.27	379.29	379.84	379.84	157	163	124	130
1	Freeable	132.13	185.00	0.00	0.00	1.21	1.61	4.01	6.18	40.69	52.38			109	115	109	115
1	JAVA	16.21	16.21	9.65	9.65	8.11	8.11	2.19	2.19	9.66	9.66	13.08	13.08	2	2	2	2
1	SQL	5.84	6.98	0.78	1.71	0.05	0.06	0.08	0.12	0.39	1.02	317.48	333.78	115	120	73	76
2	Other	790.06	802.70			4.79	4.84	8.33	8.33	59.63	59.63	101.33	101.46	165	166	165	166
2	Freeable	137.44	139.56	0.00	0.00	1.27	1.26	3.70	3.66	31.56	31.56			108	111	108	111
2	SQL	12.35	20.01	7.66	15.04	0.11	0.17	0.55	0.95	5.79	8.63	317.49	317.82	115	115	70	72
2	PL/SQL	2.30	3.59	1.17	2.38	0.01	0.02	0.06	0.12	0.46	1.31	0.97	1.61	159	160	126	127
3	Other	779.17	769.86			4.99	4.75	8.58	8.42	59.63	59.63	136.63	136.63	156	162	156	162
3	Freeable	127.31	127.13	0.00	0.00	1.21	1.21	3.97	3.97	39.19	39.19			105	105	105	105
3	PL/SQL	87.72	87.70	47.18	47.18	0.57	0.55	5.10	5.01	59.13	59.10	59.42	59.42	154	160	121	127
3	JAVA	51.36	49.68	44.75	43.07	10.27	9.94	2.97	2.94	14.69	14.69	18.64	18.64	5	5	5	5
3	SQL	8.58	15.04	3.12	9.68	0.08	0.13	0.18	0.74	1.35	7.79	317.58	317.90	108	113	65	69

[Back to Memory Statistics](#)

[Back to Top](#)

Supplemental Information

- [init.ora Parameters](#)
- [init.ora Multi-valued Parameters](#)
- [Complete List of SQL Text](#)

[Back to Top](#)

init.ora Parameters

- '*' indicates same value across all instances

Parameter Name	#	Begin value	End value (if different)
_fix_control	* 14033181:0		
_gby_hash_aggregation_enabled	* FALSE		
_unnest_subquery	* FALSE		
audit_file_dest	* /oracle/db/admin/sahrprod/adump		
audit_trail	* DB		
cluster_database	* TRUE		
compatible	* 11.2.0.0.0		
control_files	* +DATA/sahrprod/controlfile/control01.ctl, +FRA1/sahrprod/controlfile/control02.ctl		
db_block_size	* 8192		

```

db_create_file_dest      * +DATA
db_create_online_log_dest_1 * +REDO1
db_create_online_log_dest_2 * +REDO2
db_domain                * world
db_files                 * 500
db_name                  * sahrprod
db_recovery_file_dest    * +FRA1
db_recovery_file_dest_size * 157286400000
diagnostic_dest          * /oracle/db
dispatchers              * (PROTOCOL=TCP) (SERVICE=sahrprodXDB)
event                    *
instance_number          1 1
instance_number          2 2
instance_number          3 3
job_queue_processes      * 500
local_listener            1 (ADDRESS=(PROTOCOL=TCP)(HOST=131.128.3.124)(PORT=1521))
local_listener            2 (ADDRESS=(PROTOCOL=TCP)(HOST=131.128.3.125)(PORT=1521))
local_listener            3 (ADDRESS=(PROTOCOL=TCP)(HOST=131.128.3.126)(PORT=1521))
log_archive_format        * %t_%s_%r.dbf
memory_max_target        * 13421772800
memory_target             * 13421772800
open_cursors              * 300
optimizer_adaptive_features * FALSE
processes                 * 300
remote_listener           * dbprd:1521
remote_login_passwordfile * EXCLUSIVE
sessions                  * 1072
thread                    1 1
thread                    2 2
thread                    3 3
undo_retention            * 1800
undo_tablespace           1 UNDO1
undo_tablespace           2 UNDO2
undo_tablespace           3 UNDO3

```

[Back to Supplemental Information](#)
[Back to Top](#)

init.ora Multi-valued Parameters

- This section only displays parameters that have more one value
- '*' indicates same value across all instances
- '(NULL)' indicates that the parameter value was not set
- A blank value in the End Snapshot indicates the value is the same as the Begin Snapshot

Parameter Name	#	Begin value	End value (if different)
control_files	*	+DATA/sahrprod/controlfile/control01.ctl	
control_files	*	+FRA1/sahrprod/controlfile/control02.ctl	

[Back to Supplemental Information](#)
[Back to Top](#)

Complete List of SQL Text

SQL Id	SQL Text
07uac1c295aum	SELECT COUNT(*) FROM PS_U_CAM011_RESULT WHERE EMPLID = :B1 AND USER_PSWD > '' SELECT EMPLID, ACAD_CAREER, INSTITUTION, STRM, CLASS_NBR, GRD_RSTR_TYPE_SEQ, GRADE_ROSTER_TYPE, BLIND_GRADING_ID, LAST_NAME_SRCH, FIRST_NAME_SRCH, CRSE_GRADE_INPUT, TSCRPT_NOTE_ID, TSCRPT_NOTE_EXISTS, DESCR FROM PS_SSR_MIDTERM_GRD WHERE EXISTS (SELECT 'X' FROM PS_STDNT_ENRL_SSV1 WHERE EMPLID=:1 AND ACAD_CAREER=:2 AND INSTITUTION=:3 AND STRM=:4 AND STDNT_ENRL_STATUS='E' AND PS_SSR_MIDTERM_GRD.EMPLID = PS_STDNT_ENRL_SSV1.EMPLID AND PS_SSR_MIDTERM_GRD.ACAD_CAREER = PS_STDNT_ENRL_SSV1.ACAD_CAREER AND PS_SSR_MIDTERM_GRD.INSTITUTION = PS_STDNT_ENRL_SSV1.INSTITUTION AND PS_SSR_MIDTERM_GRD.STRM = PS_STDNT_ENRL_SSV1.STRM AND PS_SSR_MIDTERM_GRD.CLASS_NBR = PS_STDNT_ENRL_SSV1.CLASS_NBR) ORDER BY EMPLID, ACAD_CAREER, INSTITUTION, STRM DESC, CLASS_NBR, GRD_RSTR_TYPE_SEQ
15pqqaraszc3ps	delete from ps_dynrole_tmp WHERE ROLENAME = :1
2kxhn3uqvgc2f	SELECT DISTINCT TO_CHAR(CAST((A.CREATED_DTTM) AS TIMESTAMP), 'YYYY-MM-DD-HH24.MI.SS.FF'), A.EMPLID, D.FIRST_NAME, D.LAST_NAME, B.ACAD_PROG, C.ACAD_PLAN, A.NAME, E.ADMIT_TERM, F.ADMIT_TYPE FROM PS_U_PCON A, ((PS_ACAD_PROG B LEFT OUTER JOIN (PS_ADM_APPL_PROG E INNER JOIN PS_ADM_MAINT_SCTY E1 ON (E.EMPLID = E1.EMPLID AND E.ACAD_CAREER = E1.ACAD_CAREER AND E.STDNT_CAR_NBR = E1.STDNT_CAR_NBR AND

E.ADM_APPL_NBR = E1.ADM_APPL_NBR AND E.APPL_PROG_NBR = E1.APPL_PROG_NBR AND E1.OPRCLASS = :1 AND E1.OPRID = :2)) ON B.EMPLID = E.EMPLID AND B.ACAD_CAREER = E.ACAD_CAREER AND (E.EFFDT = (SELECT MAX(E_ED.EFFDT) FROM PS_ADM_APPL_PROG E_ED WHERE E.EMPLID = E_ED.EMPLID AND E.ACAD_CAREER = E_ED.ACAD_CAREER AND E.STDNT_CAR_NBR = E_ED.STDNT_CAR_NBR AND E.ADM_APPL_NBR = E_ED.ADM_APPL_NBR AND E.APPL_PROG_NBR = E_ED.APPL_PROG_NBR) AND E.EFFSEQ = (SELECT MAX(E_ES.EFFSEQ) FROM PS_ADM_APPL_PROG E_ES WHERE E.EMPLID = E_ES.EMPLID AND E.ACAD_CAREER = E_ES.ACAD_CAREER AND E.STDNT_CAR_NBR = E_ES.STDNT_CAR_NBR AND E.ADM_APPL_NBR = E_ES.ADM_APPL_NBR AND E.APPL_PROG_NBR = E_ES.APPL_PROG_NBR AND E.EFFDT = E_ES.EFFDT) OR E.EFFDT IS NULL) AND E.PROG_ACTION = 'MATR') LEFT OUTER JOIN (PS_ADM_APPL_DATA F INNER JOIN PS_ADM_APPL_SCTY F1 ON (F.EMPLID = F1.EMPLID AND F.ACAD_CAREER = F1.ACAD_CAREER AND F.ADM_APPL_NBR = F1.ADM_APPL_NBR AND F1.OPRCLASS = :3 AND F1.OPRID = :4)) ON E.EMPLID = F.EMPLID AND E.ACAD_CAREER = F.ACAD_CAREER AND E.ADM_APPL_NBR = F.ADM_APPL_NBR), PS_ACAD_PLAN C, PS_SCC_NAMES_QVW D WHERE ((A.U_DOC_TYPE_ID IN ('321Z16H_0000HFGHC00003K', '321Z16G_0000H4LPJ00001K') AND A.CREATED_DTTM > TRUNC(TO_DATE(:5, 'YYYY-MM-DD')) AND B.EMPLID = A.EMPLID AND B.EFFDT = (SELECT MAX(B_ED.EFFDT) FROM PS_ACAD_PROG B_ED WHERE B.EMPLID = B_ED.EMPLID AND B.ACAD_CAREER = B_ED.ACAD_CAREER AND B.STDNT_CAR_NBR = B_ED.STDNT_CAR_NBR AND B_ED.EFFDT <= SYSDATE) AND B.EFFSEQ = (SELECT MAX(B_ES.EFFSEQ) FROM PS_ACAD_PROG B_ES WHERE B.EMPLID = B_ES.EMPLID AND B.ACAD_CAREER = B_ES.ACAD_CAREER AND B.STDNT_CAR_NBR = B_ES.STDNT_CAR_NBR AND B.EFFDT = B_ES.EFFDT) AND B.PROG_STATUS = 'AC' AND B.EMPLID = C.EMPLID AND B.ACAD_CAREER = C.ACAD_CAREER AND B.STDNT_CAR_NBR = C.STDNT_CAR_NBR AND B.EFFSEQ = C.EFFSEQ AND C.EFFDT = (SELECT MAX(C_ED.EFFDT) FROM PS_ACAD_PLAN C_ED WHERE C.EMPLID = C_ED.EMPLID AND C.ACAD_CAREER = C_ED.ACAD_CAREER AND C.STDNT_CAR_NBR = C_ED.STDNT_CAR_NBR AND C_ED.EFFDT <= SYSDATE) AND C.EFFSEQ = (SELECT MAX(C_ES.EFFSEQ) FROM PS_ACAD_PLAN C_ES WHERE C.EMPLID = C_ES.EMPLID AND C.ACAD_CAREER = C_ES.ACAD_CAREER AND C.STDNT_CAR_NBR = C_ES.STDNT_CAR_NBR AND C.EFFDT = C_ES.EFFDT) AND D.EMPLID = A.EMPLID AND D.EFFDT = (SELECT MAX(D_ED.EFFDT) FROM PS_SCC_NAMES_QVW D_ED WHERE D.EMPLID = D_ED.EMPLID AND D.NAME_TYPE = D_ED.NAME_TYPE AND D_ED.EFFDT <= SYSDATE) AND D.NAME_TYPE = 'PRI')) ORDER BY 1, 4, 3, 2

SELECT /*+ OPAQUE_TRANSFORM */ "EMPLID", "EMPL_RCD", "EFFDT", "EFFSEQ", "PER_ORG", "DEPTID", "JOBCODE", "POSITION_NBR", "SUPERVISOR_ID", "HR_STATUS", "APPT_TYPE", "MAIN_APPT_NUM_JPN", "POSITION_OVERRIDE", "POSN_CHANGE_RECORD", "EMPL_STATUS", "ACTION", "ACTION_DT", "ACTION_REASON", "LOCATION", "TAX_LOCATION_CD", "JOB_ENTRY_DT", "DEPT_ENTRY_DT", "POSITION_ENTRY_DT", "SHIFT", "REG_TEMP", "FULL_PART_TIME", "COMPANY", "PAYGROUP", "BAS_GROUP_ID", "ELIG_CONFIG1", "ELIG_CONFIG2", "ELIG_CONFIG3", "ELIG_CONFIG4", "ELIG_CONFIG5", "ELIG_CONFIG6", "ELIG_CONFIG7", "ELIG_CONFIG8", "ELIG_CONFIG9", "BEN_STATUS", "BAS_ACTION", "COBRA_ACTION", "EMPL_TYPE", "HOLIDAY_SCHEDULE", "STD_HOURS", "STD_HRS_FREQUENCY", "OFFICER_CD", "EMPL_CLASS", "SAL_ADMIN_PLAN", "GRADE", "GRADE_ENTRY_DT", "STEP", "STEP_ENTRY_DT", "GL_PAY_TYPE", "ACCT_CD", "EARNS_DIST_TYPE", "COMP_FREQUENCY", "COMPRATE", "CHANGE_AMT", "CHANGE_PCT", "ANNUAL_RT", "MONTHLY_RT", "DAILY_RT", "HOURLY_RT", "ANNUAL_BENEF_BASE_RT", "SHIFT_RT", "SHIFT_FACTOR", "CURRENCY_CD", "BUSINESS_UNIT", "SETID_DEPT", "SETID_JOBCODE", "SETID_LOCATION", "SETID_SALARY", "SETID_EMPL_CLASS", "REG_REGION", "DIRECTLY_TIPPED", "FLSA_STATUS", "EEO_CLASS", "FUNCTION_CD", "TARIFF_GER", "TARIFF_AREA_GER", "PERFORM_GROUP_GER", "LABOR_TYPE_GER", "SPK_COMM_ID_GER", "HOURLY_RT_FRA", "ACCDNT_CD_FRA", "VALUE_1_FRA", "VALUE_2_FRA", "VALUE_3_FRA", "VALUE_4_FRA", "VALUE_5_FRA", "CTG_RATE", "PAID_HOURS", "PAID_FTE", "PAID_HRS_FREQUENCY", "UNION_FULL_PART", "UNION_POS", "MATRICULA_NBR", "SOC_SEC_RISK_CODE", "UNION_FEE_AMOUNT", "UNION_FEE_START_DT", "UNION_FEE_END_DT", "EXEMPT_JOB_LBR", "EXEMPT_HOURS_MONTH", "WRKS_CNCL_FUNCTION", "INTERCTR_WRKS_CNCL", "CURRENCY_CD1", "PAY_UNION_FEE", "UNION_CD", "BARG_UNIT", "UNION_SENIORITY_DT", "ENTRY_DATE", "LABOR_AGREEMENT", "EMPL_CTG", "EMPL_CTG_L1", "EMPL_CTG_L2", "SETID_LBR_AGRMNT", "WPP_STOP_FLAG", "LABOR_FACILITY_ID", "LBR_FAC_ENTRY_DT", "LAYOFF_EXEMPT_FLAG", "LAYOFF_EXEMPT_RSN", "GP_PAYGROUP", "GP_DFLT_ELIG_GRP", "GP_ELIG_GRP", "GP_DFLT_CURRTYP", "CUR_RT_TYPE", "GP_DFLT_EXRTDT", "GP_ASOF_DT_EXG_RT", "ADDS_TO_FTE_ACTUAL", "CLASS_INDC", "ENCUMB_OVERRIDE", "FICA_STATUS_EE", "FTE", "PRORATE_CNT_AMT", "PAY_SYSTEM_FLG", "BORDER_WALKER", "LUMP_SUM_PAY", "CONTRACT_NUM", "JOB_INDICATOR", "WRKS_CNCL_ROLE_CHE", "BENEFIT_SYSTEM", "WORK_DAY_HOURS", "REPTS_TO", "FORCE_PUBLISH", "JOB_DATA_SRC_CD", "ESTABID", "SUPV_LVL_ID", "SETID_SUPV_LVL", "ABSENCE_SYSTEM_CD", "POI_TYPE", "HIRE_DT", "LAST_HIRE_DT", "TERMINATION_DT", "ASGN_START_DT", "LST_ASGN_START_DT", "ASGN_END_DT", "LDW_OVR", "LAST_DATE_WORKED", "EXPECTED_RETURN_DT", "EXPECTED_END_DATE", "AUTO_END_FLG", "LASTUPDDTTM", "LASTUPDOPRID" FROM "SYSADM"."PS_JOB"

UPDATE PS_U_SIF_SR_LDAP SHG SET SHG.U_HOME_EMAIL_ADDR = :B2 WHERE SHG.EMPLID = :B1
 SELECT ITEM_NBR, ITEM_AMT, APPLIED_AMT, CONTRACT_AMT, CONTRACT_NUM FROM PS_ITEM_SF WHERE BUSINESS_UNIT = :1 AND COMMON_ID = :2 AND SA_ID_TYPE = :3 AND ACCOUNT_NBR = :4 AND ITEM_TYPE = :5 AND REF1_DESCR = :6 AND PAYMENT_ID_NBR = :7 AND REFUND_NBR = :8 AND ACAD_CAREER = :9 AND STDNT_CAR_NBR = :10 AND SESSION_CODE = :11 AND ADM_APPL_NBR = :12 AND ACCOUNT_TERM = :13 AND FEE_CD = :14 AND COURSE_LIST = :15 AND CRSE_ID = :16 AND CONTRACT_EEMPLID = :17 AND WAIVER_CODE = :18 AND TRANS_FEE_CD = :19 AND ITEM_TERM = :20 AND BILLING_CAREER = :21 AND LATE_FEE_CODE = :22 AND CLASS_NBR = :23 AND ORIGNL_CURRENCY_CD = :24 AND ITEM_NBR_SOURCE = :25 AND TAX_AUTHORITY_CD = :26 AND SEL_GROUP = :27 AND SF_DEPOSIT_ID = :28 AND SSF_INSTMNT_ID = :29

SELECT DISTINCT A.STRM, B.DESCR, A.SESSION_CODE, C.XLATLONGNAME FROM PS_SSR_CS_INDX_BND A, PS_TERM_VAL_TBL B, PSXLATITEM C, PS_U_CUR_TERM_VW D WHERE (B.STRM = A.STRM AND C.EFFDT = (SELECT MAX(C_ED.EFFDT) FROM PSXLATITEM C_ED WHERE C.FIELDNAME = C_ED.FIELDNAME AND C.FIELDVALUE = C_ED.FIELDVALUE AND C_ED.EFFDT <= SYSDATE) AND C.FIELDNAME = 'SESSION_CODE' AND C.FIELDVALUE = A.SESSION_CODE AND A.STRM < DECODE(SUBSTR(D.STRM, 4, 1), '1', D.STRM, '0', D.STRM-1)) ORDER BY 1 DESC

SELECT DISTINCT IC.OWNING_SCHOOL_CD FROM PS_ISIR_CONTROL IC, PS_ISIR_STUDENT IR WHERE IC.INSTITUTION = :1 AND IC.ORIG_SSN = :2 AND IC.NAME_CD = :3 AND IR.SSN = :4 AND IR.BIRTHDATE = :5

SELECT DISTINCT A.EMPLID, F.NAME, B.ITEM_TYPE, B.DESCRSHORT, A.OFFER_AMOUNT, A.ACCEPT_AMOUNT, A.DISBURSED_AMOUNT, C.LN_ACTION_CD, TO_CHAR(CAST((C.LN_ACTION_DT) AS TIMESTAMP), 'YYYY-MM-DD-HH24.MI.SS.FF'), C.LN_DISBACTN_NET, E.DESCR, B.FA_SOURCE FROM PS_ITEM_TYPE_FA B, PS_STDNT_AWARDS A, PS_STDNT_FA_TERM D, PS_PERS_DATA_SA_VW F, PS_LOAN_DISB_ACTN C, PS_LN_DISBCD_VW E WHERE (B.EFFDT = (SELECT MAX(B_ED.EFFDT) FROM PS_ITEM_TYPE_FA B_ED WHERE B.SETID = B_ED.SETID AND B.ITEM_TYPE = B_ED.ITEM_TYPE AND B.AID_YEAR = B_ED.AID_YEAR AND B_ED.EFFDT <= SYSDATE) AND B.SETID = A.INSTITUTION AND B.ITEM_TYPE = A.ITEM_TYPE AND B.AID_YEAR = A.AID_YEAR AND B.AID_YEAR = :1 AND B.SETID = :2 AND A.EMPLID = D.EMPLID AND A.INSTITUTION = D.INSTITUTION AND A.AID_YEAR = D.AID_YEAR AND A.ACAD_CAREER = D.ACAD_CAREER AND D.EFFDT = (SELECT MAX(D_ED.EFFDT) FROM PS_STDNT_FA_TERM D_ED WHERE D.EMPLID = D_ED.EMPLID AND D.INSTITUTION = D_ED.INSTITUTION AND D.STRM = D_ED.STRM AND D_ED.EFFDT <= SYSDATE) AND D.EFFSEQ = (SELECT MAX(D_ES.EFFSEQ) FROM PS_STDNT_FA_TERM D_ES WHERE D.EMPLID = D_ES.EMPLID AND D.INSTITUTION =

D.ES.INSTITUTION AND D.STRM = D.ES.STRM AND D.EFFDT = D.ES.EFFDT) AND A.OFFER_AMOUNT > 0 AND A.EMPLID = F.EMPLID AND A.EMPLID = C.EMPLID AND A.INSTITUTION = C.INSTITUTION AND A.AID_YEAR = C.AID_YEAR AND A.ITEM_TYPE = C.ITEM_TYPE AND A.ACAD_CAREER = C.ACAD_CAREER AND C.AID_YEAR = E.AID_YEAR AND E.LN_ACTION_TYPE = C.LN_ACTION_TYPE AND E.LN_ACTION_CD = C.LN_ACTION_CD AND E.LOAN_PROGRAM = 'A' AND C.LN_ACTION_CD = 'D' AND NOT EXISTS (SELECT G.EMPLID FROM PS_LOAN_DISB_ACTN G WHERE G.EMPLID = C.EMPLID AND G.INSTITUTION = 'URIPS' AND G.AID_YEAR = C.AID_YEAR AND G.ACAD_CAREER = C.ACAD_CAREER AND G.LOAN_TYPE = C.LOAN_TYPE AND G.LN_APPL_SEQ = C.LN_APPL_SEQ AND G.DISBURSEMENT_ID = C.DISBURSEMENT_ID AND G.LN_DISB_SEQ_NBR = C.LN_DISB_SEQ_NBR + 1 AND G.LN_ACTION_CD IN ('PSDS', 'PSAD', 'PSCR', 'PSFH')) AND B.FA_SOURCE = 'P')

4ztz048yfq32s
SELECT TO_CHAR(current_timestamp AT TIME ZONE 'GMT', 'YYYY-MM-DD HH24:MI:SS TZD') AS curr_timestamp, COUNT(username) AS failed_count, TO_CHAR(MIN(timestamp), 'yyyy-mm-dd hh24:mi:ss') AS first_occur_time, TO_CHAR(MAX(timestamp), 'yyyy-mm-dd hh24:mi:ss') AS last_occur_time FROM sys.dba_audit_session WHERE returncode != 0 AND timestamp >= current_timestamp - TO_DSINTERVAL('0 0:30:00')

5xb53jnnx2r98
SELECT AD.EMPLID, AD.ADDRESS1, AD.CITY, AD.STATE, AD.POSTAL FROM PS_ADDRESSES AD WHERE AD.ADDRESS_TYPE = 'HOME' AND AD.EFF_STATUS = 'A' AND AD.EFFDT = (SELECT MAX(AD1.EFFDT) FROM PS_ADDRESSES AD1 WHERE AD.EMPLID = AD1.EMPLID AND AD1.EFF_STATUS = 'A' AND AD1.ADDRESS_TYPE = AD.ADDRESS_TYPE) AND NOT EXISTS (SELECT INFO.EMPLID FROM PS_FERPA_OVERRIDE INFO WHERE AD.EMPLID = INFO.EMPLID AND INFO.RECNAME = 'ADDR_FERPA_VW' AND INFO.FIELDNAME = 'HOME')

664ab0677vad5
SELECT SE.STRM, SE.EMPLID, SE.STDNT_ENRL_STATUS, SE.ENRL_STATUS_REASON FROM PS_STDNT_ENRL SE WHERE SE.STRM = :B3 AND SE.CLASS_NBR = :B2 AND SE.SESSION_CODE = :B1 AND SE.LAST_ENRL_DT_STMP = (SELECT MAX(B.LAST_ENRL_DT_STMP) FROM PS_STDNT_ENRL B WHERE SE.EMPLID = B.EMPLID AND SE.STRM = B.STRM AND SE.CLASS_NBR = B.CLASS_NBR AND SE.SESSION_CODE = B.SESSION_CODE) AND SE.LAST_ENRL_TM_STMP = (SELECT MAX(U.LAST_ENRL_TM_STMP) FROM PS_STDNT_ENRL U WHERE SE.EMPLID = U.EMPLID AND SE.STRM = U.STRM AND SE.CLASS_NBR = U.CLASS_NBR AND SE.SESSION_CODE = U.SESSION_CODE AND U.LAST_ENRL_DT_STMP = SE.LAST_ENRL_DT_STMP) AND SE.EMPLID IN (SELECT X.EMPLID FROM PS_U_HDI001_LDAP_D2L X WHERE X.EMPLID = SE.EMPLID AND X.U_STUDENT_FLG = 'Y')

6g4vwbj836c8x
SELECT HEAD.ECTRANSID, HEAD.ECQUEUEINSTANCE, HEAD.ECTRANSINOUTSW, HEAD.IWD_RECORD_ID, HEAD.IWD_RECORD_TYPE, DTL2.IWD_ETI_DEST_CODE, HEAD.IWD_CREATE_DATE, HEAD.IWD_CREATE_TIME, HEAD.IWD_AWARD_YEAR, HEAD.IWD_BATCH_NBR, HEAD.CPS_SCHOOL_CODE, HEAD.IWD_TRANS_SRC_CD, HEAD.IWD_TRANS_SITE_CD, HEAD.ISIR_LOAD_DTTM, DTL.ISIR_SEQ_NO, DTL.ISIR_LOAD_STATUS, DTL.ISIR_LOAD_ACTION, DTL.ADMIT_LVL, DTL.ORIG_SSN, DTL.NAME_CD, DTL.IWD_TRANS_NBR, DTL.EMPLID, DTL.CPS_SCHOOL_CODE, DTL.IWD_BATCH_YR, DTL.LAST_NAME_SSN_CHNG, DTL.SSN, DTL.IWD_STD_LAST_NAME, DTL.IWD_STD_FIRST_NM02, DTL.IWD_STU_MI, DTL.IWD_PERM_ADDR02, DTL.IWD_CITY, DTL.IWD_STATE, DTL.IWD_ZIP, DTL.BIRTHDATE, DTL.IWD_PERM_PHONE, DTL.IWD_HAVE_DL, DTL.DRIVERS_LICENSE_NO, DTL.IWD_DL_STATE, DTL.CITIZENSHIP_STATUS, DTL.IWD_STU_ALIEN_REG, DTL.MARITAL_STAT, DTL.IWD_MAR_STAT_DT, DTL.ENROLL_CURNT_SUM, DTL.ENROLL_FALL, DTL.ENROLL_WINTER, DTL.ENROLL_SPRING, DTL.ENROLL_NEXT_SUM, DTL.FATHER_GRADE_LVL, DTL.MOTHER_GRADE_LVL, DTL.IWD_STATE_RES, DTL.IWD_RES_PRIOR_94, DTL.IWD_LEGAL_RES_DT, DTL.IWD_MALE, DTL.SEL_SERV_REGISTER, DTL.DEGREE_CERTIF, DTL.IWD_GRADE_LVL, DTL.HS_GED_RCVD, DTL.FIRST_BACH_DEGREE, DTL.INTERESTED_IN_SL, DTL.INTERESTED_IN_WS, DTL.NUM_MONTH_VET_BEN, DTL.IWD_MTHLY_VET_BEN, DTL.TAX_FORM_STU, DTL.TAX_FORM_FILED, DTL.IWD_ELIG_1040A_EZ, DTL.IWD_STU_AGI, DTL.IWD_STU_TAX_PAID, DTL.IWD_STU_NBR_EXEMP, DTL.IWD_STU_EIC, DTL.IWD_STU_WORK_INC, DTL.IWD_SPS_WORK_INC, DTL.IWD_STU_WK_A, DTL.IWD_STU_WK_B, DTL.IWD_STU_CASH, DTL.IWD_STU_INVEST, DTL.IWD_STU_BUS_WORTH, DTL.IWD_STU_FARM_WORTH, DTL.IWD_BORN_PRIOR_76, DTL.GRADUATE_STUDENT, DTL.MARRIED, DTL.ORPHAN, DTL.VETERAN, DTL.DEPENDENTS, DTL.IWD_STU_FAM_MEM, DTL.IWD_STU_MEM_COLL, DTL.TAX_FORM_PAR, DTL.TAX_FORM_FILED_PAR, DTL.IWD_ELIG_1040A_PAR, DTL.IWD_PAR_AGI, DTL.IWD_PAR_TAX_PAID, DTL.IWD_PAR_NBR_EXEMP, DTL.IWD_PAR_EIC, DTL.IWD_FATH_WORK_INC, DTL.IWD_PAR_BUS_WORTH, DTL.IWD_PAR_FARM_WORTH, DTL.MARITAL_STAT_PAR, DTL.IWD_PAR_FAM_MEM, DTL.IWD_PAR_MEM_COLL, DTL.IWD_PAR_STATE_RES, DTL.IWD_PAR_RES_PRI_94, DTL.IWD_PAR_RES_DT, DTL.AGE_OLDER_PARENT, DTL.SCHOOL_CHOICE_1, DTL.HOUSING_CODE_1, DTL.SCHOOL_CHOICE_2, DTL.HOUSING_CODE_2, DTL.SCHOOL_CHOICE_3, DTL.HOUSING_CODE_3, DTL.SCHOOL_CHOICE_4, DTL.HOUSING_CODE_4, DTL.SCHOOL_CHOICE_5, DTL.HOUSING_CODE_5, DTL.SCHOOL_CHOICE_6, DTL.HOUSING_CODE_6, DTL.DT_APP_COMPLETED, DTL.IWD_APP_SIGNED_BY, DTL.IWD_PREP_SSN, DTL.IWD_PREP_EIN, DTL.IWD_PREP_SIGNATURE, DTL.DEPENDENCY_OVERRIDE, DTL.FAA_INSTITUT_NUM, DTL.FAA_SIGNATURE, DTL.ADJ_EFC_CALC_REQ, DTL.IWD_DRN, DTL.TRANS_RECEIPT_DT, DTL.IWD_SERIAL_NO, DTL.BATCH_NUMBER, DTL2.DEPNDNCY_STAT, DTL2.TRANS_PROCESS_DT, DTL.ID_OVERRIDE, DTL.ALT_EFFDT, DTL.IWD_RES_PRIOR, DTL.DRUG_OFFENSE_CONV, DTL.IWD_BORN_PRIOR, DTL.FATHER_SSN, DTL.FATHER_LAST_NAME, DTL.MOTHER_SSN, DTL.MOTHER_LAST_NAME, DTL.IWD_PAR_RES_PRI, DTL.IWD_INPUT_REC_TYPE, HEAD.CPS_RECORD_TYPE, DTL.IWD_STU_WK_C, DTL.CHILDREN, DTL2.DUPLICATE_REQUEST, DTL2.IWD_DUP_SSN_CD, DTL.IWD_PAR_MAR_STAT_DT, DTL2.IWD_PRIMARY_EFC, DTL.FATHER_1ST_NM_INIT, DTL.FATHER_DOB, DTL.MOTHER_1ST_NM_INIT, DTL.MOTHER_DOB, DTL.ISIR_STU_ENRL_STAT, DTL.IWD_PAR_EMAIL, DTL2.ISIR_CPS_PUSHED, DTL2.IWD_TRANS_SOURCE, DTL2.ISIR_ETI, DTL2.MULTI_INSTITUTION1, DTL2.MULTI_INSTITUTION2, DTL2.MULTI_INSTITUTION3, DTL2.MULTI_INSTITUTION4, DTL2.MULTI_INSTITUTION5, DTL2.MULTI_INSTITUTION6, DTL2.IWD_SOURCE_CORR, DTL.SFA_INTRSTD_IN_WSL, DTL.SFA_ACTIVE_DUTY, DTL.SFA_PAR_SSI_INCOME, DTL.SFA_PAR_FOOD_STAMP, DTL.SFA_PAR_LUNCH_BEN, DTL.SFA_PAR_TANF_BEN, DTL.SFA_PAR_WIC_BEN, DTL.SFA_SSI_INCOME, DTL.SFA_FOOD_STAMPS, DTL.SFA_SCHL_LUNCH_PRG, DTL.SFA_TANF_BENEFITS, DTL.SFA_WIC_BENEFITS, DTL.SFA_STDNT_GENDER, DTL.SFA_HS_DIP_EQUIV, DTL.SFA_SCHL_CHOICE_7, DTL.SFA_SCHL_CHOICE_8, DTL.SFA_SCHL_CHOICE_9, DTL.SFA_SCHL_CHOICE_10, DTL.SFA_HOUSING_CODE7, DTL.SFA_HOUSING_CODE8, DTL.SFA_HOUSING_CODE9, DTL.SFA_HOUSING_CODE10, DTL2.SFA_REJ_OVR_MR_STU, DTL2.SFA_REJ_OVR_NO_TAX, DTL2.SFA_IW_SIGNREJ_EFC, DTL2.IWD_CORRECTIONS, DTL2.IWD_HIGHLIGHT, DTL2.IWD_DATA_VERIFY, DTL2.SFA_MULTI_INST7, DTL2.SFA_MULTI_INST8, DTL2.SFA_MULTI_INST9, DTL2.SFA_MULTI_INST10, DTL.SFA_TCH_COURSE_WRK, DTL.SFA_EMANCIPT_MINOR, DTL.SFA_LEGAL_GUARDIAN, DTL.SFA_YOUTH_LIASON, DTL.SFA_YOUTH_HUD, DTL.SFA_RISK_HOMELESS, DTL.SFA_STU_DISLOC_WRK, DTL.SFA_STU_EDU_CREDIT, DTL.SFA_STU_SUP_PAID, DTL.SFA_STU_NEED_EMPL, DTL.SFA_STU_GRANT_AID, DTL.SFA_STU_COMBAT_PAY, DTL.SFA_STU_PEN_PAY, DTL.SFA_STU_IRA_PAY, DTL.SFA_STU_SUP_RECV, DTL.SFA_STU_INT_INCOME, DTL.SFA_STU_IRA_DIST, DTL.SFA_STU_UNTAX_PEN, DTL.SFA_STU_MIL_ALLOW, DTL.SFA_STU_VET_ED_BEN, DTL.SFA_STU_UNTAX_INC, DTL.SFA_STU_NREP_MONEY, DTL.SFA_PAR_DISLOC_WRK, DTL.SFA_PAR_EDU_CREDIT, DTL.SFA_PAR_SUP_PAID, DTL.SFA_PAR_NEED_EMPL, DTL.SFA_PAR_GRANT_AID, DTL.SFA_PAR_COMBAT_PAY, DTL.SFA_PAR_PEN_PAY, DTL.SFA_PAR_IRA_PAY, DTL.SFA_PAR_SUP_RECV, DTL.SFA_PAR_INT_INCOME, DTL.SFA_PAR_IRA_DIST,

```

DTL.SFA_PAR_UNTAX_PEN , DTL.SFA_PAR_MIL_ALLOW , DTL.SFA_PAR_VET_ED_BEN , DTL.SFA_PAR_UNTAX_INC ,
DTL.SFA_RECV_VET_BEN , DTL.SFA_VET_BEN_TYPE , DTL2.SFA_SPL_CIRCUM_FLG , DTL.SFA_STU_COOP_EARN ,
DTL.SFA_PAR_COOP_EARN , DTL.SFA_HIGH_SCHL_NAME , DTL.SFA_HIGH_SCHL_CITY , DTL.SFA_HIGH_SCHL_STAT ,
DTL.SFA_HIGH_SCHL_CODE , DTL2.SFA_HIGH_SCHL_FLAG , DTL.SFA_STU_ASSET_THRS , DTL.SFA_PAR_ASSET_THRS ,
DTL2.SFA_REJ_OVR_MAR_ST , DTL2.SFA_STU_ASM_ASSET , DTL2.SFA_PAR_ASM_ASSET , DTL.SFA_STU_TAX_RET ,
DTL.SFA_PAR_TAX_RET , DTL2.SFA_STU_IRS_DISP , DTL2.SFA_PAR_IRS_DISP , DTL2.SFA_STU_IRS_TAXRET ,
DTL2.SFA_STU_IRS_EXEMPT , DTL2.SFA_STU_IRS_EDCRDT , DTL2.SFA_STU_IRS_IRAPAY , DTL2.SFA_STU_IRS_INTRST ,
DTL2.SFA_STU_IRS_IRADIS , DTL2.SFA_STU_IRS_UNTAX , DTL2.SFA_STU_IRS_STATUS , DTL2.SFA_PAR_IRS_TAXRET ,
DTL2.SFA_PAR_IRS_EXEMPT , DTL2.SFA_PAR_IRS_EDCRDT , DTL2.SFA_PAR_IRS_IRAPAY , DTL2.SFA_PAR_IRS_INTRST ,
DTL2.SFA_PAR_IRS_IRADIS , DTL2.SFA_PAR_IRS_UNTAX , DTL2.SFA_PAR_IRS_STATUS , DTL2.SFA_STU_IRS_SCHED ,
DTL2.SFA_PAR_IRS_SCHED FROM PS_ECQUEUE QUEUE, PS_ISIR_00_HEAD_EC HEAD, PS_ISIR_00_1_EC DTL,
PS_ISIR_00_2_EC DTL WHERE QUEUE.ECTRANSID = :1 AND QUEUE.ECTRANSINOUTSW = 'I' AND
QUEUE.ECQUEUESTATUS = 'L' AND QUEUE.ECBUSDOCID = (SELECT MAX(X.ECBUSDOCID) FROM PS_ECQUEUE X WHERE
X.ECTRANSID = QUEUE.ECTRANSID AND X.ECTRANSINOUTSW = QUEUE.ECTRANSINOUTSW AND X.ECQUEUEINSTANCE =
QUEUE.ECQUEUEINSTANCE) AND QUEUE.ECTRANSID = HEAD.ECTRANSID AND QUEUE.ECQUEUEINSTANCE =
HEAD.ECQUEUEINSTANCE AND QUEUE.ECTRANSINOUTSW = HEAD.ECTRANSINOUTSW AND QUEUE.ECTRANSID =
DTL.ECTRANSID AND QUEUE.ECQUEUEINSTANCE = DTL.ECQUEUEINSTANCE AND QUEUE.ECTRANSINOUTSW =
DTL.ECTRANSINOUTSW AND ((DTL.ISIR_LOAD_STATUS = 'U' AND DTL.ISIR_LOAD_ACTION <> 'I') OR
(DTL.PROCESS_INSTANCE <> :2 AND DTL.ISIR_LOAD_ACTION IN ('A', 'R')) AND QUEUE.ECTRANSID = DTL2.ECTRANSID AND
QUEUE.ECQUEUEINSTANCE = DTL2.ECQUEUEINSTANCE AND QUEUE.ECTRANSINOUTSW = DTL2.ECTRANSINOUTSW AND
DTL.ISIR_SEQ_NO = DTL2.ISIR_SEQ_NO AND HEAD.IWD_DEST_NBR = :3 AND HEAD.IWD_AWARD_YEAR = :4 AND (('Y' = :5
AND DTL2.IWD_TRANS_SOURCE IN ('1A', '2A', '2B', '2R', '2E', '2F', '2G', '2H', '2T', '3A', '3E', '3G', '3R', '4A', '4B', '4J', '4K', '4R', '5S',
'7R', '7T', '7G', '7H', '4U', '4V', '6A', '6B', '6E', '6F', '8A', '8E', '8G', '8R')) OR ('Y' = :6 AND DTL2.IWD_TRANS_SOURCE IN ('1C', '2C',
'3C', '3V', '4C', '6C', '2Q', '4Q', '8C')) OR ('Y' = :7 AND DTL2.IWD_TRANS_SOURCE IN ('5C', '5M', '5N', '5D', '5L', '5P', '5W', '5X', '5Y'))
ORDER BY HEAD.ECQUEUEINSTANCE, DTL.ISIR_SEQ_NO

```

```

6hvmstf84xhq insert into PS_DYNROLE_TMP (OPRID, ROLENAME) values (:1, :2)
72uzt9d37ux2z SELECT OPRCLASS, BUSINESS_UNIT, EEMPLID, ACCOUNT_BALANCE, NID_COUNTRY, NATIONAL_ID_TYPE,
NID_DESCRSHORT, NATIONAL_ID, NATIONAL_ID_MSK, NAME, CAMPUS_ID, LAST_NAME_SRCH, FIRST_NAME_SRCH FROM
PS_PEOPLE_SRCH2_SF WHERE OPRCLASS=:1 ORDER BY OPRCLASS, BUSINESS_UNIT, EEMPLID
768qjmf9pu5k DECLARE BEGIN U_SIF004(3125720); END;
7ah2cs0jwhwbw UPDATE PS_U_SIF_SR_LDAP SHG SET SHG.EMAIL_ADDR = :B2 WHERE SHG.EEMPLID = :B1
8722qujwb0rfq SELECT EEMPLID, E_ADDR_TYPE, EMAIL_ADDR, PEF_EMAIL_FLAG, DISPLAY_ORDER FROM PS_U_SAD_EMAIL_VW WHERE
EEMPLID=:1 ORDER BY EEMPLID
8r9tyhzhf6t4wb INSERT INTO PS_SEARCH_CNTL_TBX ( PROCESSINSTANCE , SAD_EEMPLID , FIELDNAME , FIELDTYPE , SA_CHARACTER ,
SA_NUMBER , SA_DECIMAL ) VALUES (:1 , :2 , :3 , 0 , :4 , 0 , 0 )
8v0ctgmbmxzh1 call U_HHR072_BUILDTABLE('LDAP', '', 'U_HHR072')
93na5cg9gjc6p UPDATE PS_U_SIF_SR_LDAP SHG SET SHG.U_HOME_PHONE = :B2 WHERE SHG.EEMPLID = :B1
SELECT DISTINCT D.OPRID , A.PTAI_LIST_ID , A.PTAI_LABEL , A.PORTAL_NAME , A.PTAI_PARENT_TMPL ,
NVL(A.PTAI_DUE_DATE, ITML.SCC_TM_DUE_DATE) SCC_TM_DUE_DATE , A.LAST_UPDATE_DTTM , ((
NVL(A.PTAI_DUE_DATE, ITML.SCC_TM_DUE_DATE)) - (TO_DATE(TO_CHAR(SYSDATE, 'YYYY-MM-DD'), 'YYYY-MM-DD')) FROM
PS_PTAI_LIST A , PS_SCC_TM_CONFIG B , PSOPRDEFN D , PS_SCC_TM_PTAIITEM ITML , PS_SCC_TM_PTAIITEM ITMF
WHERE A.PTAI_PARENT_TMPL = B.SCC_TM_TASK_ID AND A.PTAI_ACTIVE_FLG = 'Y' AND A.PTAI_IS_TMPL='N' AND
A.PTAI_LIST_STATUS = 'IP' AND A.PTAI_LIST_ID = ITML.SCC_TM_LIST_ID AND ITML.SCC_TM_PTAI_SEQ = ( SELECT
MAX(ITM1.SCC_TM_PTAI_SEQ) FROM PS_SCC_TM_PTAIITEM ITM1 WHERE A.PTAI_LIST_ID = ITM1.SCC_TM_LIST_ID ) AND
A.PTAI_LIST_ID = ITMF.SCC_TM_LIST_ID AND ITMF.SCC_TM_PTAI_SEQ = ( SELECT MIN(ITM2.SCC_TM_PTAI_SEQ) FROM
PS_SCC_TM_PTAIITEM ITM2 WHERE A.PTAI_LIST_ID = ITM2.SCC_TM_LIST_ID ) AND ( EXISTS ( SELECT 'X' FROM
PS_SCC_TM_PTAIPRV F WHERE F.SCC_TM_LIST_ID = A.PTAI_LIST_ID AND EXISTS ( SELECT 'X' FROM P_SROLEUSER X
WHERE X.ROLEUSER = D.OPRID AND ((F.SCC_TM_PTAIMBRTYPE = 'USER' AND F.SCC_TM_PTAIMBRNAME = X.ROLEUSER)
OR (F.SCC_TM_PTAIMBRTYPE = 'ROLE' AND F.SCC_TM_PTAIMBRNAME = X.ROLENAME)))) AND NVL(A.PTAI_START_DT,
TO_DATE(TO_CHAR(SYSDATE, 'YYYY-MM-DD'), 'YYYY-MM-DD')) <= TO_DATE(TO_CHAR(SYSDATE, 'YYYY-MM-DD'), 'YYYY-MM-
DD') AND NVL(ITMF.SCC_TM_START_DATE, TO_DATE(TO_CHAR(SYSDATE, 'YYYY-MM-DD'), 'YYYY-MM-DD')) <=
TO_DATE(TO_CHAR(SYSDATE, 'YYYY-MM-DD'), 'YYYY-MM-DD') AND A.PTAI_AGRENDERTYPE IN ('NONS', 'NOSP', 'SNNO') AND
D.OPRID = :1 AND EXISTS ( SELECT 'X' FROM PS_SCC_TM_PTAICNTX CNTX WHERE CNTX.SCC_TM_LIST_ID
=A.PTAI_LIST_ID AND CNTX.SCC_TM_CTX_KEY='Y' AND CNTX.FIELDNAME='EEMPLID' AND
UPPER(CNTX.SCC_TM_KEY_VALUE) = :2) ORDER BY SCC_TM_DUE_DATE ASC , A.PTAI_LABEL ASC
SELECT VERSION, PROGRUNLOC, NAMECOUNT, PROGLEN, PROGTXT, LICENSE_CODE, TO_CHAR(CAST((LASTUPDDTTM)
AS TIMESTAMP), 'YYYY-MM-DD-HH24.MI.SS.FF'), LASTUPDOPRID, PROGFLAGS, PROGEXTENDS, PTOOLSREL FROM
PSPCMPROG WHERE OBJECTID1 = :1 AND OBJECTVALUE1 = :2 AND OBJECTID2 = :3 AND OBJECTVALUE2 = :4 AND
OBJECTID3 = :5 AND OBJECTVALUE3 = :6 AND OBJECTID4 = :7 AND OBJECTVALUE4 = :8 AND OBJECTID5 = :9 AND
OBJECTVALUE5 = :10 AND OBJECTID6 = :11 AND OBJECTVALUE6 = :12 AND OBJECTID7 = :13 AND OBJECTVALUE7 = :14
ORDER BY PROGSEQ
a712uyahq6fvf UPDATE PS_U_SIF_SR_LDAP SHG SET SHG.OPRID = :B2 WHERE SHG.EEMPLID = :B1
an88hf8k322jw update psoprdfn set version = :1 where oprid in (select roleuser from psroleuser where rolename = :2 and dynamic_sw = 'Y') and oprid
not in (select oprid from ps_dynrole_tmp WHERE ROLENAME = :3)
avpp3w6rzbjr2 UPDATE PS_U_SIF_SR_LDAP SHG SET SHG.U_HOME_ADDR1 = :B5 , SHG.U_HOME_CITY = :B4 , SHG.U_HOME_STATE = :B3 ,
SHG.U_HOME_POSTAL = :B2 WHERE SHG.EEMPLID = :B1
b08jbbpg4rs1n SELECT ALTCONTNUM, SEQNUM, CONTDATA FROM PSCONTENT WHERE CONTNAME = :1 AND CONTTYPE = :2 ORDER BY
ALTCONTNUM, SEQNUM
b9qxyft335u80 SELECT VERSION FROM PSVERSION WHERE OBJECTTYPENAME = 'SYS'
bdntyxtax2smq DELETE FROM PS_SAA_ADB_CRSEAVL WHERE EEMPLID = :1 AND ANALYSIS_DB_SEQ = :2
c83vn8rw2dstg UPDATE PS_SSR_CLS_TBL_ES SET CRSE_ID=:1, CRSE_OFFER_NBR=:2, STRM=:3, SESSION_CODE=:4, CLASS_SECTION=:5,
INSTITUTION=:6, ACAD_CAREER=:7, CLASS_NBR=:8, SSR_COMPONENT=:9, ASSOCIATED_CLASS=:10,
c93sup0snbjym LAST_UPDATE_DTTM=TO_TIMESTAMP(:11, 'YYYY-MM-DD-HH24.MI.SS.FF'), AUDIT_ACTN=:12 WHERE CRSE_ID=:13 AND
CRSE_OFFER_NBR=:14 AND STRM=:15 AND SESSION_CODE=:16 AND CLASS_SECTION=:17
d2uzkd87gy505 SELECT A.CRSE_ID , A.EFFDT , A.EQUIV_CRSE_ID FROM PS_CRSE_CATALOG A WHERE A.CRSE_ID = :1 AND A.EFFDT =
(SELECT MAX(B.EFFDT) FROM PS_CRSE_CATALOG B WHERE B.CRSE_ID = A.CRSE_ID AND B.EFF_STATUS = 'A' AND
B.EFFDT <= :2)

```

```

dag0yvdynd3ky1 DELETE FROM PS_DYNROLE_TMP WHERE ROLENAME = :1 AND OPRID IN (SELECT ROLEUSER FROM PSROLEUSER
WHERE ROLENAME = :2 AND DYNAMIC_SW <>'Y')
dffkcnqfystw WITH MONITOR_DATA AS (SELECT INST_ID, KEY, NVL2(PX_QCSID, NULL, STATUS) STATUS, FIRST_REFRESH_TIME,
LAST_REFRESH_TIME, REFRESH_COUNT, PROCESS_NAME, SID, SQL_ID, SQL_EXEC_START, SQL_EXEC_ID, DBOP_NAME,
DBOP_EXEC_ID, SQL_PLAN_HASH_VALUE, SQL_FULL_PLAN_HASH_VALUE, SESSION_SERIAL#, SQL_TEXT,
IS_FULL_SQLTEXT, PX_SERVER#, PX_SERVER_GROUP, PX_SERVER_SET, PX_QCINST_ID, PX_QCSID, CASE WHEN
ELAPSED_TIME < (CPU_TIME+ APPLICATION_WAIT_TIME+ CONCURRENCY_WAIT_TIME+ CLUSTER_WAIT_TIME+
USER_IO_WAIT_TIME+ QUEUING_TIME) THEN (CPU_TIME+ APPLICATION_WAIT_TIME+ CONCURRENCY_WAIT_TIME+
CLUSTER_WAIT_TIME+ USER_IO_WAIT_TIME+ QUEUING_TIME) ELSE ELAPSED_TIME END ELAPSED_TIME, QUEUING_TIME,
CPU_TIME, APPLICATION_WAIT_TIME, CONCURRENCY_WAIT_TIME, CLUSTER_WAIT_TIME, USER_IO_WAIT_TIME, CASE
WHEN ELAPSED_TIME < (CPU_TIME+ APPLICATION_WAIT_TIME+ CONCURRENCY_WAIT_TIME+ CLUSTER_WAIT_TIME+
USER_IO_WAIT_TIME+ QUEUING_TIME) THEN 0 ELSE ELAPSED_TIME - (CPU_TIME+ APPLICATION_WAIT_TIME+ C
ONCURRENCY_WAIT_TIME+ CLUSTER_WAIT_TIME+ USER_IO_WAIT_TIME+ QUEUING_TIME) END OTHER_WAIT_TIME,
PLSQL_EXEC_TIME, JAVA_EXEC_TIME, FETCHES, BUFFER_GETS, IO_INTERCONNECT_BYTES IO_INTER_BYTES,
PHYSICAL_READ_REQUESTS READ_REQS, PHYSICAL_READ_BYTES READ_BYTES, PHYSICAL_WRITE_REQUESTS
WRITE_REQS, PHYSICAL_WRITE_BYTES WRITE_BYTES, NVL(PHYSICAL_READ_BYTES, 0) + NVL(PHYSICAL_WRITE_BYTES,
0) IO_BYTES, NVL(PHYSICAL_READ_REQUESTS, 0) + NVL(PHYSICAL_WRITE_REQUESTS, 0) IO_REQS, USER# USERID,
USERNAME, MODULE, ACTION, SERVICE_NAME, CLIENT_IDENTIFIER, CLIENT_INFO, PROGRAM, PLSQL_OBJECT_ID
PL_OID, PLSQL_SUBPROGRAM_ID PL_PROGID, PLSQL_ENTRY_OBJECT_ID PL_ENTRY_OID,
PLSQL_ENTRY_SUBPROGRAM_ID PL_ENTRY_PROGID, PX_MAXDOP MAX_DOP, PX_IS_CROSS_INSTANCE,
PX_MAXDOP_INSTANCES MAX_DOP_INSTANCES, PX_SERVERS_REQUESTED SERVERS_REQUESTED,
PX_SERVERS_ALLOCATED SERVERS_ALLOCATED, ERROR_NUMBER, ERROR_FACILITY, ERROR_MESSAGE,
NVL2(OTHER_XML, 'Y', NULL) HAS_OTHER_XML, NVL2(BINDS_XML, 'Y', NULL) HAS_BINDS_XML,
NVL2(RM_CONSUMER_GROUP, NVL2(RM_LAST_ACTION_TIME, TO_CHAR(RM_LAST_ACTION_TIME, 'B12'), '00/00/0000
00:00:01') || XMLELEMENT("rminfo", XMLATTRIBUTES(RM_LAST_ACTION AS "rmlastact", RM_LAST_ACTION_REASON AS
"rmlastreason", TO_CHAR(RM_LAST_ACTION_TIME, 'B12') AS "rmlasttime", RM_CONSUMER_GROUP AS
"rmcg"))).GETSTRINGVAL(), '00/00/0000 00:00:00') RM_INFO, CON_NAME, CON_ID FROM GV$SQL_MONITOR MO1 WHERE
MO1.INST_ID BETWEEN :B11 AND :B10 AND MO1.SQL_ID = NVL(:B9, MO1.SQL_ID) AND MO1.SQL_EXEC_START = NVL(:B8,
MO1.SQL_EXEC_START) AND MO1.SQL_EXEC_ID = NVL(:B7, MO1.SQL_EXEC_ID) OR (MO1.DBOP_NAME = NVL(:B6,
MO1.DBOP_NAME) AND MO1.DBOP_EXEC_ID = NVL(:B5, MO1.DBOP_EXEC_ID)) AND ((:B4 = 1 AND MO1.PX_QCSID IS NULL)
OR (MO1.PX_SERVER_GROUP = NVL(:B3, MO1.PX_SERVER_GROUP) AND MO1.PX_SERVER_SET = NVL(:B2,
MO1.PX_SERVER_SET) AND MO1.PX_SERVER# = NVL(:B1, MO1.PX_SERVER#))), MONITOR_AGG AS (SELECT MAX
PX_QCSID, MAX_KEY, MAX_INST_ID, MAX_SESSION_ID, MAX_SESSION_SERIAL, MAX_PX_DOP, MAX_PX_DOP_INSTANCES,
MAX_PX_IS_CROSS_INSTANCE, SUM_SERVERS_REQUESTED, SUM_SERVERS_ALLOCATED, DIST_INST_COUNT,
DIST_PX_GROUP_COUNT, DIST_PX_SET_COUNT, MAX_PLAN_HASH_VALUE, MAX_FULL_PLAN_HASH_VALUE, MAX_USERID,
MAX_PROGRAM, MAX_USERNAME, MAX_MODULE, MAX_ACTION, MAX_SERVICE_NAME, MAX_CLIENT_ID,
MAX_CLIENT_INFO, MAX_ERROR_NUMBER, MAX_ERROR_FACILITY, MAX_ERROR_MESSAGE, QC_HAS_OTHER_XML,
QC_HAS_BINDS_XML, MAX_PL_OID, MAX_PL_PROGID, MAX_PL_ENTRY_OID, MAX_PL_ENTRY_PROGID, MAX_SQL_ID,
MAX_SQL_EXEC_START, MAX_SQL_EXEC_ID, MAX_LAST_REFRESH_TIME, MAX_DBOP_NAME, MAX_DBOP_EXEC_ID, CASE
WHEN MAX_PL_OID IS NOT NULL THEN NVL((SELECT P.OWNER || '.' || P.OBJECT_NAME || DECODE(P.PROCEDURE_NAME,
NULL, '', '.' || P.PROCEDURE_NAME) FROM DBA_PROCEDURES P WHERE P.OBJECT_ID = MAX_PL_OID AND
P.SUBPROGRAM_ID = MAX_PL_PROGID AND ROWNUM = 1), 'Unavailable') END MAX_PL_NAME, CASE WHEN MAX_PL_EN
TRY_OID IS NOT NULL THEN NVL((SELECT P.OWNER || '.' || P.OBJECT_NAME || DECODE(P.PROCEDURE_NAME, NULL, '', '.' ||
P.PROCEDURE_NAME) FROM DBA_PROCEDURES P WHERE P.OBJECT_ID = MAX_PL_ENTRY_OID AND P.SUBPROGRAM_ID
= MAX_PL_ENTRY_PROGID AND ROWNUM = 1), 'Unavailable') END MAX_PL_ENTRY_NAME, MAX_STATUS,
SUM_REFRESH_COUNT, MIN_FIRST_REFRESH_TIME, IS_FULL_TEXT, SQLMON_TEXT, SUM_ELAPSED_TIME,
MAX_ELAPSED_TIME, MAX_QUEUING_TIME, SUM_CPU_TIME, SUM_USER_IO_WAIT_TIME, SUM_APPLICATION_WAIT_TIME,
SUM_CONCURRENCY_WAIT_TIME, SUM_CLUSTER_WAIT_TIME, SUM_PLSQL_EXEC_TIME, SUM_JAVA_EXEC_TIME,
SUM_OTHER_WAIT_TIME, SUM_FETCHES, SUM_BUFFER_GETS, SUM_READ_REQS, SUM_READ_BYTES,
SUM_WRITE_REQS, SUM_WRITE_BYTES, SUM_IO_BYTES, SUM_IO_INTER_BYTES, DECODE(:B14, 1, 'db_name',
DB.DB_UNIQUE_NAME) DB_UNIQUE_NAME, DECODE(:B14, 1, 'platform_name', DB.PLATFORM_NAME) PLATFORM_NAME,
DECODE(:B14, 1, 'host_name', INST.HOST_NAME) HOST_NAME, AGG_RM_INFO MAX_RM_INFO, MAX_CON_NAME, DEC
ODE(MAX_CON_ID, 0, NULL, MAX_CON_ID) MAX_CON_ID FROM (SELECT MAX(PX_QCSID) MAX_PX_QCSID, MAX(CASE
WHEN PX_QCSID IS NULL THEN KEY ELSE NULL END) MAX_KEY, MAX(CASE WHEN PX_QCSID IS NULL THEN INST_ID ELSE
NULL END) MAX_INST_ID, MAX(CASE WHEN PX_QCSID IS NULL THEN SID ELSE NULL END) MAX_SESSION_ID, MAX(CASE
WHEN PX_QCSID IS NULL THEN SESSION_SERIAL# ELSE NULL END) MAX_SESSION_SERIAL, MAX(MAX_DOP)
MAX_PX_DOP, MAX(MAX_DOP_INSTANCES) MAX_PX_DOP_INSTANCES, MAX(PX_IS_CROSS_INSTANCE)
MAX_PX_IS_CROSS_INSTANCE, SUM(SERVERS_REQUESTED) SUM_SERVERS_REQUESTED, SUM(SERVERS_ALLOCATED)
SUM_SERVERS_ALLOCATED, COUNT(DISTINCT INST_ID) DIST_INST_COUNT, COUNT(DISTINCT PX_SERVER_GROUP)
DIST_PX_GROUP_COUNT, COUNT(DISTINCT PX_SERVER_SET) DIST_PX_SET_COUNT, MAX(SQL_PLAN_HASH_VALUE)
MAX_PLAN_HASH_VALUE, MAX(SQL_FULL_PLAN_HASH_VALUE) MAX_FULL_PLAN_HASH_VALUE, MAX(USERID)
MAX_USERID, MAX(PROGRAM) MAX_PROGRAM, MAX(USERNAME) MAX_USERNAME, MAX(MODULE) MAX_MODULE,
MAX(ACTION) MAX_ACTION, MAX(SERVICE_NAME) MAX_SERVICE_NAME, MAX(CLIENT_IDENTIFIER) MAX_CLIENT_ID,
MAX(CLIENT_INFO) MAX_CLIENT_INFO, MAX(ERROR_NUMBER) MAX_ERROR_NUMBER, MAX(ERROR_FACILITY)
MAX_ERROR_FACILITY, MAX(ERROR_MESSAGE) MAX_ERROR_MESSAGE, MAX(NVL2(PX_QCSID, HAS_OTHER_XML, NULL))
QC_HAS_OTHER_XML, MAX(HAS_BINDS_XML) QC_HAS_BINDS_XML, MAX(PL_OID) MAX_PL_OID, MAX(PL_PROGID)
MAX_PL_PROGID, MAX(PL_ENTRY_OID) MAX_PL_ENTRY_OID, MAX(PL_ENTRY_PROGID) MAX_PL_ENTRY_PROGID,
MAX(SQL_ID) MAX_SQL_ID, MAX(SQL_EXEC_START) MAX_SQL_EXEC_START, MAX(SQL_EXEC_ID) MAX_SQL_EXEC_ID,
MAX(LAST_REFRESH_TIME) MAX_LAST_REFRESH_TIME, MAX(STATUS) MAX_STATUS, SUM(REFRESH_COUNT)
SUM_REFRESH_COUNT, MIN(FIRST_REFRESH_TIME) MIN_FIRST_REFRESH_TIME, MAX(DBOP_NAME) MAX_DBOP_NAME,
MAX(DBOP_EXEC_ID) MAX_DBOP_EXEC_ID, CASE WHEN :B13 = 0 THEN NULL ELSE MAX(SQL_TEXT) END SQLMON_TEXT,
MAX((IS_FULL_SQLTEXT) IS_FULL_TEXT, SUM(ELAPSED_TIME) SUM_ELAPSED_TIME, MAX(ELAPSED_TIME)
MAX_ELAPSED_TIME, MAX(QUEUING_TIME) MAX_QUEUING_TIME, SUM(CPU_TIME) SUM_CPU_TIME,
SUM(USER_IO_WAIT_TIME) SUM_USER_IO_WAIT_TIME, SUM(APPLICATION_WAIT_TIME) SUM_APPLICATION_WAIT_TIME,
SUM(CONCURRENCY_WAIT_TIME) SUM_CONCURRENCY_WAIT_TIME, SUM(CLUSTER_WAIT_TIME)
SUM_CLUSTER_WAIT_TIME, SUM(PLSQL_EXEC_TIME) SUM_PLSQL_EXEC_TIME, SUM(JAVA_EXEC_TIME)
SUM_JAVA_EXEC_TIME, SUM(OTHER_WAIT_TIME) SUM_OTHER_WAIT_TIME, SUM(FETCHES) SUM_FETCHES,
SUM(BUFFER_GETS) SUM_BUFFER_GETS, SUM(READ_REQS) SUM_READ_REQS, SUM(READ_BYTES) SUM_READ_BYTES,
SUM(WRITE_REQS) SUM_WRITE_REQS, SUM(WRITE_BYTES) SUM_WRITE_BYTES, NVL(SUM(READ_BYTES), 0) +
NVL(SUM(WRITE_BYTES), 0) SUM_IO_BYTES, SUM(IO_INTER_BYTES) SUM_IO_INTER_BYTES, MAX(RM_INFO)
AGG_RM_INFO, MAX(CON_NAME) MAX_CON_NAME, MAX(CON_ID) MAX_CON_ID FROM MONITOR_DATA) MD, (SELECT

```

```

HOST_NAME FROM V$INSTANCE) INST, (SELECT DB_UNIQUE_NAME, PLATFORM_NAME FROM V$DATABASE) DB),
ASH_DATA AS (SELECT AD0.INST_ID, AD0.SESSION_ID, AD0.PLAN_LINE_ID, AD0.PLSQL_OBJECT_ID,
AD0.PLSQL_SUBPROGRAM_ID, AD0.ACTIVITY_BUCKET_NUM, AD0.PLAN_ACTIVITY_BUCKET_NUM, AD0.SQL,
AD0.TOP_LEVEL_SQL_ID, AD0.DBOP_NAME, AD0.IS_MONITORED_SQL, AD0.IS_PX_SLAVE, AD0.BUCKET_ACTIVITY_START,
AD0.ACTIVITY_START, AD0.BUCKET_ACTIVITY_END, AD0.ACTIVITY_END, AD0.ACTIVITY_COUNT, AD0.ACTIVITY_TYPE,
AD0.OTHER_SQL_ACTIVITY_TYPE, AD0.EVENT_NAME, AD0.IMQ_COUNT, AD0.WAIT_COUNT, AD0.CPU_COUNT,
AD0.OTHER_SQL_COUNT, AD0.PX_SERVER_SET, AD0.PX_DFO_ACTIVITY_COUNT, AD0.DFO_MOST_ACTIVE_IID,
AD0.DFO_MOST_ACTIVE_SID, (CASE WHEN AD0.DFO_MOST_ACTIVE_IID = AD0.INST_ID AND AD0.DFO_MOST_ACTIVE_SID
= AD0.SESSION_ID AND (((AD0.PX_DFO_ACTIVITY_COUNT / AD0.DFO_MOST_ACTIVE_COUNT) >= AD0.PX_DOP * 1.05) OR
((AD0.PX_DFO_ACTIVITY_COUNT / AD0.DFO_MOST_ACTIVE_COUNT) <= AD0.PX_DOP * 0.95)) AND (AD0.DFO_DURATION *
100) > :B32 THEN AD0.ACTIVITY_COUNT ELSE NULL END) DFO_MOST_ACTIVE_COUNT, AD0.BUCKET_DISTINCT_SAMPLES,
AD0.SQL_BUCKET_DISTINCT_SAMPLES, CASE WHEN AD0.PX_SERVER_GROUP IS NULL AND AD0.IS_PX_SLAVE = 1 THEN 1
ELSE AD0.PX_SERVER_GROUP END PX_SERVER_GROUP, AD0.PX_STEP_ID, AD0.PX_DFO_PAIR_TYPE,
AD0.PX_DFO_PAIR_ID, AD0.PX_STEP_ARG, AD0.PX_DOP, CASE WHEN AD0.PX_DOP IS NOT NULL AND AD0.PX_DOP <>
AD0.PX_MIN_DOP AND AD0.PX_MIN_DOP != 0 THEN PX_MIN_DOP ELSE NULL END PX_MIN_DOP FROM (SELECT /*+
use_hash(ash) leading(ash) */ ASH.INST_ID, ASH.IS_PX_SLAVE, ASH.SESSION_ID, ASH.PLAN_LINE_ID,
ASH.PLSQL_OBJECT_ID, ASH.PLSQL_SUBPROGRAM_ID, ASH.ACTIVITY_BUCKET_NUM,
ASH.PLAN_ACTIVITY_BUCKET_NUM, ASH.SQL, ASH.TOP_LEVEL_SQL_ID, ASH.DBOP_NAME, ASH.IS_MONITORED_SQL,
ASH.BUCKET_ACTIVITY_START, ASH.ACTIVITY_START, ASH.BUCKET_ACTIVITY_END, ASH.ACTIVITY_END,
ASH.ACTIVITY_COUNT, ASH.ACTIVITY_TYPE, ASH.OTHER_SQL_ACTIVITY_TYPE, ASH.EVENT_NAME, ASH.IMQ_COUNT,
ASH.WAIT_COUNT, ASH.CPU_COUNT, ASH.OTHER_SQL_COUNT, MO.PX_SERVER_SET, ASH.PX_DFO_ACTIVITY_COUNT,
TRUNC(ASH.MOST_ACTIVE_IN_DFO / 1000000000) DFO_MOST_ACTIVE_COUNT, MOD(TRUNC(ASH.MOST_ACTIVE_IN_DFO /
1000000), 10000) DFO_MOST_ACTIVE_IID, MOD(ASH.MOST_ACTIVE_IN_DFO, 1000000) DFO_MOST_ACTIVE_SID,
ASH.DFO_DURATION, ASH.BUCKET_DISTINCT_SAMPLES, ASH.SQL_BUCKET_DISTINCT_SAMPLES,
MO.PX_SERVER_GROUP, ASH.PX_STEP_ID, ASH.PX_DFO_PAIR_TYPE, ASH.PX_DFO_PAIR_ID, ASH.PX_STEP_ARG,
ASH.PX_DOP, ASH.PX_MIN_DOP FROM (SELECT /*+ no_merge */ MD.INST_ID, MD.SID, MD.SESSION_SERIAL#,
MD.PX_SERVER_SET, MD.PX_SERVER_GROUP FROM MONITOR_DATA MD WHERE MD.SID IS NOT NULL OR :B31 = 1) MO,
(SELECT /*+ no_merge */ GVTF.INST_ID, GVTF.GLOBAL_SAMPLE_ID, GVTF.IS_PX_SLAVE, GVTF.SESSION_ID,
GVTF.PLAN_LINE_ID, GVTF.PLSQL_OBJECT_ID, GVTF.PLSQL_SUBPROGRAM_ID, GVTF.ACTIVITY_BUCKET_NUM,
GVTF.PLAN_ACTIVITY_BUCKET_NUM, GVTF.SQL, GVTF.TOP_LEVEL_SQL_ID, GVTF.DBOP_NAME,
GVTF.IS_MONITORED_SQL, GVTF.BUCKET_ACTIVITY_START, GVTF.ACTIVITY_START, GVTF.BUCKET_ACTIVITY_END,
GVTF.ACTIVITY_END, GVTF.ACTIVITY_COUNT, GVTF.ACTIVITY_TYPE, GVTF.OTHER_SQL_ACTIVITY_TYPE,
GVTF.EVENT_NAME, GVTF.IMQ_COUNT, GVTF.WAIT_COUNT, GVTF.CPU_COUNT, GVTF.OTHER_SQL_COUNT,
MAX(GVTF.PER_SERVER_DFO_COUNT * 1000000000 + GVTF.INST_ID * 1000000 + GVTF.SESSION_ID) OVER(PARTITION BY
GVTF.PX_DFO_PAIR_TYPE, GVTF.PX_DFO_PAIR_ID) MOST_ACTIVE_IN_DFO, SUM(GVTF.ACTIVITY_COUNT)
OVER(PARTITION BY GVTF.PX_DFO_PAIR_TYPE, GVTF.PX_DFO_PAIR_ID) PX_DFO_ACTIVITY_COUNT,
GVTF.DFO_DURATION, GVTF.PX_STEP_ID, GVTF.PX_DFO_PAIR_TYPE, GVTF.PX_DFO_PAIR_ID, GVTF.PX_STEP_ARG,
GVTF.PX_DOP, GVTF.PX_MIN_DOP, COUNT(DISTINCT GVTF.GLOBAL_SAMPLE_ID) OVER(PARTITION BY
GVTF.ACTIVITY_BUCKET_NUM) BUCKET_DISTINCT_SAMPLES, COUNT(DISTINCT GVTF.GLOBAL_SAMPLE_ID)
OVER(PARTITION BY GVTF.SQL, GVTF.ACTIVITY_BUCKET_NUM) SQL_BUCKET_DISTINCT_SAMPLES FROM
TABLE(GV$(CURSOR( (SELECT USERENV('INSTANCE') INST_ID, ASH2.GLOBAL_SAMPLE_ID, CASE WHEN :B30 = 1 AND :B27
> 1 THEN BUCKET_NUM ELSE NULL END ACTIVITY_BUCKET_NUM, CASE WHEN :B29 = 1 AND :B27 > 1 THEN BUCKET_NUM
ELSE NULL END PLAN_ACTIVITY_BUCKET_NUM, ASH2.SQL, ASH2.TOP_LEVEL_SQL_ID, ASH2.DBOP_NAME,
ASH2.IS_MONITORED_SQL, ASH2.PLAN_LINE_ID, ASH2.PLSQL_OBJECT_ID, ASH2.PLSQL_SUBPROGRAM_ID,
ASH2.ACTIVITY_TYPE, ASH2.OTHER_SQL_ACTIVITY_TYPE, ASH2.EVENT_NAME, ASH2.IS_PX_SLAVE, ASH2.SESSION_ID,
ASH2.PX_STEP_ID, ASH2.PX_DFO_PAIR_TYPE, ASH2.PX_DFO_PAIR_ID, ASH2.PX_STEP_ARG, CASE WHEN
ASH2.PX_DFO_PAIR_ID IS NOT NULL THEN DECODE(ASH2.PX_DOP, 0, :B28, ASH2.PX_DOP) ELSE NULL END PX_DOP,
ASH2.PX_MIN_DOP, :B20 + NUMTODSINTERVAL(:B26 * (ASH2.BUCKET_NUM-1), 'SECOND') BUCKET_ACTIVITY_START, :B20 +
NUMTODSINTERVAL(:B26 * ASH2.BUCKET_NUM - 1, 'SECOND') BUCKET_ACTIVITY_END, ASH2.ACTIVITY_START,
ASH2.ACTIVITY_END, ASH2.ACTIVITY_COUNT, ASH2.IMQ_COUNT, ASH2.WAIT_COUNT, ASH2.CPU_COUNT,
ASH2.OTHER_SQL_COUNT, SUM(ASH2.ACTIVITY_COUNT) OVER(PARTITION BY ASH2.PX_DFO_PAIR_ID,
ASH2.PX_DFO_PAIR_TYPE, DECODE(ASH2.PX_DFO_PAIR_ID, NULL, NULL, ASH2.SESSION_ID)) PER_SERVER_DFO_COUNT,
CEIL((MAX(ASH2.MAX_SAMPLE_DATE) OVER(PARTITION BY ASH2.PX_DFO_PAIR_ID, ASH2.PX_DFO_PAIR_TYPE) -
MIN(ASH2.MIN_SAMPLE_DATE) OVER(PARTITION BY ASH2.PX_DFO_PAIR_ID, ASH2.PX_DFO_PAIR_TYPE)) * 3600 * 24)
DFO_DURATION FROM (SELECT ASH1.BUCKET_NUM, ASH1.GLOBAL_SAMPLE_ID, ASH1.PLAN_LINE_ID,
ASH1.PLSQL_OBJECT_ID, ASH1.PLSQL_SUBPROGRAM_ID, ASH1.ACTIVITY_TYPE, ASH1.OTHER_SQL_ACTIVITY_TYPE,
ASH1.EVENT_NAME, ASH1.SESSION_ID, ASH1.PX_STEP_ID, ASH1.PX_STEP_ARG, MAX(ASH1.SQL) SQL,
MAX(ASH1.IS_MONITORED_SQL) IS_MONITORED_SQL, MAX(ASH1.PX_DFO_PAIR_TYPE) PX_DFO_PAIR_TYPE,
MAX(ASH1.PX_DFO_PAIR_ID) PX_DFO_PAIR_ID, MIN(SAMPLE_DATE) MIN_SAMPLE_DATE, MAX(SAMPLE_DATE)
MAX_SAMPLE_DATE, MAX(ASH1.IS_PX_SLAVE) IS_PX_SLAVE, MAX(ASH1.PX_DOP) PX_DOP, MIN(ASH1.PX_DOP)
PX_MIN_DOP, MIN(ASH1.SAMPLE_DATE) ACTIVITY_START, MAX(ASH1.SAMPLE_DATE) ACTIVITY_END, COUNT(ASH1.SQL)
ACTIVITY_COUNT, COUNT(CASE WHEN ASH1.ACTIVITY_TYPE = 'Cpu' AND ASH1.EVENT_NAME = 'in memory' THEN 1 ELSE
NULL END) IMQ_COUNT, COUNT(CASE WHEN ASH1.ACTIVITY_TYPE != 'Other SQL Execution' AND ASH1.ACTIVITY_TYPE !=
'Non SQL' AND ASH1.ACTIVITY_TYPE != 'Cp u' THEN 1 ELSE NULL END) WAIT_COUNT, COUNT(CASE WHEN
ASH1.ACTIVITY_TYPE = 'Cpu' AND ASH1.EVENT_NAME IS NULL THEN 1 ELSE NULL END) CPU_COUNT, COUNT(CASE WHEN
ASH1.ACTIVITY_TYPE = 'Other SQL Execution' AND ASH1.ACTIVITY_TYPE != 'Non SQL' THEN 1 ELSE NULL END)
OTHER_SQL_COUNT, MAX(ASH1.TOP_LEVEL_SQL_ID) TOP_LEVEL_SQL_ID, MAX(ASH1.DBOP_NAME) DBOP_NAME FROM (
SELECT (CASE WHEN :B27 > 1 THEN (TRUNC(DELTA_TIME_SECONDS/ :B26) + 1) ELSE 1 END) BUCKET_NUM, ASH00.SQL,
ASH00.SAMPLE_DATE, NVL2(DUP.C2, TRUNC(DELTA_TIME_SECONDS/ (:B25)) + 1, NULL) GLOBAL_SAMPLE_ID,
NVL2(DUP.C2, NULL, ASH00.IS_MONITORED_SQL) IS_MONITORED_SQL, NVL2(DUP.C2, NULL, ASH00.IN_INMEMORY_QUERY)
IN_INMEMORY_QUERY, NVL2(DUP.C2, NULL, ASH00.WAIT_CLASS) WAIT_CLASS, NVL2(DUP.C2, NULL,
ASH00.ACTIVITY_TYPE) ACTIVITY_TYPE, NVL2(DUP.C2, NULL, ASH00.OTHER_SQL_ACTIVITY_TYPE)
OTHER_SQL_ACTIVITY_TYPE, NVL2(DUP.C2, NULL, ASH00.EVENT_NAME) EVENT_NAME, NVL2(DUP.C2, NULL,
ASH00.TOP_LEVEL_SQL_ID) TOP_LEVEL_SQL_ID, NVL2(DUP.C2, NULL, ASH00.DBOP_NAME) DBOP_NAME, NVL2(DUP.C2,
NULL, ASH00.IS_PX_SLAVE) IS_PX_SLAVE, NVL2(DUP.C2, NULL, ASH00.SESSION_ID) SESSION_ID, NVL2(DUP.C2, NULL,
ASH00.PLSQL_OBJECT_ID) PLSQL_OBJECT_ID, NVL2(DUP.C2, NULL, ASH00.PLSQL_SUBPROGRAM_ID)
PLSQL_SUBPROGRAM_ID, NVL2(DUP.C2, NULL, ASH00.PLAN_LINE_ID) PLAN_LINE_ID, NVL2(DUP.C2, NULL,
ASH00.PX_STEP_ID) PX_STEP_ID, NVL2(DUP.C2, NULL, ASH00.PX_STEP_ARG) PX_STEP_ARG, NVL2(DUP.C2, NULL,
ASH00.PX_DFO_PAIR_ID) PX_DFO_PAIR_ID, NVL2(DUP.C2, NULL, ASH00.PX_DFO_PAIR_TYPE) PX_DFO_PAIR_TYPE,
NVL2(DUP.C2, NULL, ASH00.PX_DOP) PX_DOP FROM (SELECT 1 C1, NULL C2 FROM V$TIMER UNION ALL SELECT 1 C1, 1 C2
FROM V$TIMER) DUP, (SELECT /*+ no_merge */ 1 C1, ASH000.SAMPLE_DATE, ASH000.IS_MONITORED_SQL,

```

```

((EXTRACT(SECOND FROM(DELTA_TIME)) + EXTRACT(MINUTE FROM(DELTA_TIME)) * 60 + EXTRACT(HOUR
FROM(DELTA_TIME)) * 3600 + EXTRACT(DAY FROM(DELTA_TIME)) * 86400)) DELTA_TIME_SECONDS,
ASH000.IN_INMEMORY_QUERY, ASH000.WAIT_CLASS, DECODE(ASH000.IS_MONITORED_SQL, 1, NVL(ASH000.WAIT_CLASS,
'Cpu'), DECODE(SQL_ID, NULL, 'Non SQL', 'Other SQL Execution')) ACTIVITY_TYPE, NVL(ASH000.WAIT_CLASS, 'Cpu')
OTHER_SQL_ACTIVITY_TYPE, DECODE(:B24, 1, CASE WHEN ASH000.IS_MONITORED_SQL = 1 THEN NVL(ASH000.EVENT,
DECODE(ASH000.IN_INMEMORY_QUERY, 'Y', 'in memory', ASH000.EVENT)) WHEN ASH000.SQL_ID IS NOT NULL THEN 'sql_id: '
|| ASH000.SQL_ID WHEN ASH000.CALL_NAME IS NOT NULL THEN 'call: ' || ASH000.CALL_NAME ELSE 'anonymous: ' ||
ASH000.EVENT END, NULL) EVENT_NAME, CASE WHEN ASH000.IS_MONITORED_SQL = 1 AND (NVL(ASH000.SQL_ID, :B9) =
:B9 OR NVL(ASH000.DBOP_NAME, :B6) = :B6) THEN 'this' WHEN ASH000.IS_PX_SLAVE = 1 AND ASH000.SQL_ID IS NOT NULL
AND ASH000.TOP_LEVEL_SQL_ID != ASH000.SQL_ID THEN ASH000.TOP_LEVEL_SQL_ID WHEN ASH000.SQL_ID IS NOT NULL
THEN ASH000.SQL_ID ELSE NVL(CALL_NAME, 'anonymous') END SQL, CASE WHEN ASH000.IS_PX_SLAVE = 0 AND
(ASH000.SQL_ID IS NULL OR ASH000.TOP_LEVEL_SQL_ID != ASH000.SQL_ID) THEN ASH000.TOP_LEVEL_SQL_ID END
TOP_LEVEL_SQL_ID, ASH000.DBOP_NAME, ASH000.IS_PX_SLAVE, CASE WHEN ASH000.IS_PX_SLAVE = 1 AND
ASH000.IS_MONITORED_SQL != 1 THEN 65536 ELSE ASH000.SESSION_ID END SESSION_ID,
DECODE(ASH000.IS_MONITORED_SQL, 1, ASH000.PLSQL_OBJECT_ID, NULL) PLSQL_OBJECT_ID,
DECODE(ASH000.IS_MONITORED_SQL, 1, ASH000.PLSQL_SUBPROGRAM_ID, NULL) PLSQL_SUBPROGRAM_ID,
DECODE(ASH000.IS_MONITORED_SQL, 1, ASH000.SQL_PLAN_LINE_ID, NULL) PLAN_LINE_ID,
DECODE(ASH000.IS_MONITORED_SQL, 1, ASH000.PX_STEP_ID, NULL) PX_STEP_ID, CASE WHEN ASH000.IS_PX_SLAVE = 1
AND ASH000.IS_MONITORED_SQL = 1 AND ASH000.PX_STEP_ID IN (1, 2, 3) THEN ASH000.PX_STEP_ARG ELSE NULL END
PX_DFO_PAIR_ID, CASE WHEN ASH000.IS_PX_SLAVE = 0 OR ASH000.IS_MONITORED_SQL != 1 THEN NULL WHEN
ASH000.PX_STEP_ID = 1 THEN 1 WHEN ASH000.PX_STEP_ID IN (2, 3) THEN 0 ELSE NULL END PX_DFO_PAIR_TYPE,
DECODE(ASH000.IS_MONITORED_SQL, 1, ASH000.PX_STEP_ARG, NULL) PX_STEP_ARG, DECODE(ASH000.IS_
MONITORED_SQL, 1, ASH000.PX_DOP, NULL) PX_DOP FROM (SELECT ASH0.*, CASE WHEN ASH0.IS_TARGET_SQL = 1 OR
(ASH0.IS_PX_SLAVE = 1 AND ((NVL(ASH0.TOP_LEVEL_SQL_ID, ASH0.SQL_ID) IS NOT NULL AND
NVL(ASH0.TOP_LEVEL_SQL_ID, ASH0.SQL_ID) = :B9) OR (SQL_ID IS NULL AND :B23 = 'Y'))) THEN 1 ELSE 0 END
IS_MONITORED_SQL FROM (SELECT (CASE WHEN (ASH.SQL_ID = :B9 AND ASH.SQL_EXEC_ID = :B7 AND
ASH.SQL_EXEC_START = :B8) THEN 1 ELSE 0 END) IS_TARGET_SQL, ASH.SQL_ID, ASH.SQL_PLAN_LINE_ID,
ASH.PLSQL_OBJECT_ID, ASH.PLSQL_SUBPROGRAM_ID, ASH.TOP_LEVEL_SQL_ID, DECODE(ASH.SQL_ID, NULL,
ASH.TOP_LEVEL_CALL_NAME, NULL) CALL_NAME, ASH.EVENT, ASH.IN_INMEMORY_QUERY, ASH.WAIT_CLASS,
ASH.SQL_EXEC_ID, ASH.SQL_EXEC_START, ASH.DBOP_NAME, ASH.DBOP_EXEC_ID, ASH.SESSION_ID,
ASH.SESSION_SERIAL#, CASE WHEN QC_INSTANCE_ID IS NOT NULL AND (ASH.SESSION_ID != ASH.QC_SESSION_ID OR
ASH.SESSION_SERIAL# != ASH.QC_SESSION_SERIAL# OR USERENV('instance') != ASH.QC_INSTANCE_ID) THEN 1 ELSE 0
END IS_PX_SLAVE, SAMPLE_TIME - CAST(:B20 AS TIMESTAMP) DELTA_TIME, CAST(FROM_TZ(ASH.SAMPLE_TIME,
DBTIMEZONE) AS DATE) SAMPLE_DATE, TRUNC(MOD(PX_FLAGS/65536, 32)) PX_STEP_ID, MOD(PX_FLAGS, 65536)
PX_STEP_ARG, TRUNC(PX_FLAGS/2097152) PX_DOP FROM V$ACTIVE_SESSION_HISTORY ASH WHERE ((ASH.SESSION_ID
= :B19 AND ASH.SESSION_SERIAL# = :B18 AND USERENV('INSTANCE') = :B17) OR (ASH.QC_SESSION_ID IS NOT NULL AND
ASH.QC_SESSION_ID = :B19 AND ASH.QC_SESSION_SERIAL# = :B18 AND ASH.QC_INSTANCE_ID = :B17)) AND
SAMPLE_TIME BETWEEN :B16 AND :B15) ASH0 WHERE (ASH0.SAMPLE_DATE BETWEEN :B20 + 1/24/3600 AND :B22 -
1/24/3600 OR (ASH0.SQL_ID = :B9 AND ASH0.SQL_EXEC_START = :B8 AND ASH0.SQL_EXEC_ID = :B7) OR
(ASH0.DBOP_NAME = :B6 AND ASH0.DBOP_EXEC_ID = :B5)) AND (:B21 IS NULL OR ASH0.SQL_PLAN_LINE_ID = :B21) AND
(ASH0.IS_PX_SLAVE = 0 OR ASH0.SQL_ID IS NOT NULL)) ASH000) ASH000 WHERE ASH0.C1 = DUP.C1) ASH1 WHERE
ASH1.BUCKET_NUM > 0 AND ASH1.BUCKET_NUM <= :B27 GROUP BY USERENV('INSTANCE'), ASH1.GLOBAL_SAMPLE_ID,
ASH1.BUCKET_NUM, ASH1.SESSION_ID, ASH1.PLAN_LINE_ID, ASH1.PLSQL_OBJECT_ID, ASH1.PLSQL_SUBPROGRAM_ID,
ASH1.ACTIVITY_TYPE, ASH1.EVENT_NAME, ASH1.OTHER_SQL_ACTIVITY_TYPE, ASH1.PX_STEP_ID, ASH1.PX_STEP_ARG)
ASH2)))) GVTF WHERE GVTF.INST_ID BETWEEN :B11 AND :B10) ASH WHERE ASH.GLOBAL_SAMPLE_ID IS NULL AND
ASH.SESSION_ID = MO.SID(+) AND ASH.INST_ID = MO.INST_ID(+) AD0), RESPONSE_TIME_DATA AS (SELECT
ADH.BUCKET_NUM, ADH.SQL_ROWNUM, ADH.SQL, ADH.TOP_LEVEL_SQL_ID, ADH.DBOP_NAME, ADH.PX_ID, DECODE(
ADH.PX_STEP_ID, NULL, 0, NULL, 1, 'PX Server(s) - Executing Parent DFO', 2, 'PX Server(s) - Executing Child DFO', 3, 'PX
Server(s) - Sampling Child DFO', 4, 'PX Server(s) - Joining Group', 5, 'QC - Scheduling Child DFO', 6, 'QC - Scheduling Parent DFO', 7,
'QC - Initializing Objects', 8, 'QC - Flushing Objects', 9, 'QC - Allocating Slaves', 10, 'QC - Initializing Granules', 11, 'PX Server(s) -
Parsing Cursor', 12, 'PX Server(s) - Executing Cursor', 13, 'PX Server(s) - Preparing Transaction', 14, 'PX Server(s) - Joining
Transaction', 15, 'PX Server(s) - Load Commit', 16, 'PX Server(s) - Aborting Transaction', 17, 'QC - Executing Child DFO', 18, 'QC -
Executing Parent DFO', 'PX Step - ' || PX_STEP_ID) PX_STEP_ID, ADH.PX_STEP_ARG, ADH.PX_DFO_PAIR_ID, ADH.PX_DOP,
ADH.PX_MIN_DOP, ADH.DFO_MOST_ACTIVE_ID, ADH.DFO_MOST_ACTIVE_SID, ADH.DFO_MOST_ACTIVE_COUNT,
ADH.ACTIVITY_START, ADH.ACTIVITY_END, ADH.ACTIVITY_TYPE, ADH.OTHER_SQL_ACTIVITY_TYPE, ADH.EVENT_NAME,
ADH.PLAN_LINE_ID, ADH.PLSQL_OBJECT_ID, ADH.PLSQL_SUBPROGRAM_ID, CASE WHEN PLSQL_ROWNUM = 1 AND
ADH.PLSQL_OBJECT_ID IS NOT NULL THEN NVL((SELECT P.OWNER || '.' || P.OBJECT_NAME ||
DECODE(P.PROCEDURE_NAME, NULL, '', '' || P.PROCEDURE_NAME) FROM DBA_PROCEDURES P WHERE P.OBJECT_ID =
ADH.PLSQL_OBJECT_ID AND P.SUBPROGRAM_ID = ADH.PLSQL_SUBPROGRAM_ID), 'Unavailable') ELSE NULL END
PLSQL_NAME, ADH.ACTIVITY_COUNT, ADH.BUCKET_ACTIVE_SECONDS, ADH.BUCKET_IDLE_SECONDS, (CASE WHEN
ADH.IS_MONITORED_SQL = 0 THEN ADH.ACTIVE_SECONDS WHEN ADH.PX_DFO_PAIR_ID IS NOT NULL AND
ADH.DFO_PAIR_CPU_HEIGHT >= :B34 THEN DECODE(ADH.ACTIVITY_TYPE, 'Cpu', (ADH.DFO_PAIR_ACTIVITY_HEIGHT /
ADH.DFO_PAIR_CPU_HEIGHT) * ADH.DFO_PAIR_TOTAL_SECONDS, 0) WHEN ADH.PX_DFO_PAIR_ID IS NOT NULL AND
ADH.PX_DOP > :B34 THEN (ADH.DFO_PAIR_TOTAL_SECONDS * DECODE(ADH.ACTIVITY_TYPE, 'Cpu',
ADH.DFO_PAIR_ACTIVITY_HEIGHT, (ADH.DFO_PAIR_ACTIVITY_HEIGHT * (:B34 - ADH.DFO_PAIR_CPU_HEIGHT)) /
(ADH.DFO_PAIR_TOTAL_HEIGHT_ADJ - ADH.DFO_PAIR_CPU_HEIGHT))) / :B34 WHEN ADH.PX_DFO_PAIR_ID IS NOT NULL
THEN (ADH.DFO_PAIR_TOTAL_SECONDS * ADH.DFO_PAIR_ACTIVITY_HEIGHT) / ADH.PX_DOP ELSE ADH.ACTIVE_SECONDS
END) RESP_TIME, (CASE WHEN ADH.PX_DFO_PAIR_ID IS NOT NULL AND ADH.DFO_PAIR_IDLE_HEIGHT > 0 AND
ADH.PX_DOP > :B34 THEN (ADH.DFO_PAIR_TOTAL_SECONDS * (ADH.DFO_PAIR_IDLE_HEIGHT * (:B34 -
ADH.DFO_PAIR_CPU_HEIGHT)) / (ADH.DFO_PAIR_TOTAL_HEIGHT_ADJ - ADH.DFO_PAIR_CPU_HEIGHT)) / :B34 WHEN
ADH.PX_DFO_PAIR_ID IS NOT NULL AND ADH.DFO_PAIR_IDLE_HEIGHT > 0 THEN (ADH.DFO_PAIR_TOTAL_SECONDS *
ADH.DFO_PAIR_IDLE_HEIGHT) / ADH.PX_DOP ELSE 0 END) DFO_PAIR_IDLE_RESP_TIME,
ADH.DFO_PAIR_ACTIVITY_HEIGHT, ADH.DFO_PAIR_CPU_HEIGHT, ADH.DFO_PAIR_IDLE_HEIGHT,
ADH.DFO_PAIR_TOTAL_HEIGHT, ADH.DFO_PAIR_CPU_ACTIVITY, ADH.DFO_PAIR_TOTAL_SECONDS FROM (SELECT ADH_1.*,
(ROW_NUMBER() OVER(PARTITION BY ADH_1.PLSQL_OBJECT_ID, ADH_1.PLSQL_SUBPROGRAM_ID ORDER BY
ADH_1.BUCKET_NUM, ADH_1.ACTIVITY_TYPE, ADH_1.EVENT_NAME)) PLSQL_ROWNUM, (ROW_NUMBER() OVER(PARTITION
BY ADH_1.SQL ORDER BY ADH_1.BUCKET_NUM, ADH_1.ACTIVITY_TYPE, ADH_1.EVENT_NAME)) SQL_ROWNUM, (CASE
WHEN ADH_1.PX_DFO_PAIR_ID IS NOT NULL AND (ADH_1.DFO_PAIR_TOTAL_HEIGHT < ADH_1.PX_DOP) AND
(ADH_1.DFO_PAIR_CPU_HEIGHT < :B34) THEN ADH_1.PX_DOP - ADH_1.DFO_PAIR_TOTAL_HEIGHT ELSE 0 END)
DFO_PAIR_IDLE_HEIGHT, (CASE WHEN ADH_1.PX_DFO_PAIR_ID IS NOT NULL AND (ADH_1.DFO_PAIR_TOTAL_HEIGHT <
ADH_1.PX_DOP) AND (ADH_1.DFO_PAIR_CPU_HEIGHT < :B34) THEN ADH_1.PX_DOP ELSE

```

```

ADH_1.DFO_PAIR_TOTAL_HEIGHT END) DFO_PAIR_TOTAL_HEIGHT_ADJ FROM (SELECT ADH_0.*, (CASE WHEN
ADH_0.DFO_PAIR_TOTAL_SECONDS > 0 THEN (ADH_0.DFO_PAIR_TOTAL_ACTIVITY * :B25 ) /
ADH_0.DFO_PAIR_TOTAL_SECONDS ELSE 0 END) DFO_PAIR_TOTAL_HEIGHT, (CASE WHEN
ADH_0.DFO_PAIR_TOTAL_SECONDS > 0 THEN (ADH_0.DFO_PAIR_CPU_ACTIVITY * :B25 ) /
ADH_0.DFO_PAIR_TOTAL_SECONDS ELSE 0 END) DFO_PAIR_CPU_HEIGHT, (CASE WHEN ADH_0.PX_DFO_PAIR_ID IS NOT
NULL AND ADH_0.DFO_PAIR_TOTAL_SECONDS > 0 THEN (ADH_0.ACTIVITY_COUNT * :B25 ) /
ADH_0.DFO_PAIR_TOTAL_SECONDS ELSE 0 END) DFO_PAIR_ACTIVITY_HEIGHT FROM (SELECT AD3.*,
(SUM(DECODE(AD3.PX_DFO_PAIR_ID, NULL, 0, AD3.ACTIVE_SECONDS)) OVER(PARTITION BY
DECODE(AD3.PX_DFO_PAIR_ID, NULL, NULL, AD3.BUCKET_NUM), DECODE(AD3.PX_DFO_PAIR_ID, NULL, NULL,
AD3.PX_STEP_ARG))) DFO_PAIR_TOTAL_SECONDS FROM (SELECT AD2.*, CASE WHEN AD2.IS_MONITORED_SQL = 0 THEN
SQL_BUCKET_DISTINCT_SAMPLES * :B25 WHEN AD2.PX_ID IS NULL THE N AD2.ACTIVITY_COUNT * :B25 WHEN
AD2.BUCKET_PARALLEL_MON_ACTIVITY > 0 THEN (AD2.ACTIVITY_COUNT * AD2.BUCKET_PARALLEL_MON_SECONDS) /
AD2.BUCKET_PARALLEL_MON_ACTIVITY ELSE 0 END ACTIVE_SECONDS, CASE WHEN AD2.BUCKET_INTERVAL >
BUCKET_ACTIVE_SECONDS THEN AD2.BUCKET_INTERVAL - BUCKET_ACTIVE_SECONDS ELSE 0 END
BUCKET_IDLE_SECONDS FROM (SELECT AD1.*, (AD1.BUCKET_SERIAL_MON_ACTIVITY * :B25 )
BUCKET_SERIAL_MON_SECONDS, (AD1.BUCKET_TOTAL_MON_ACTIVITY - AD1.BUCKET_SERIAL_MON_ACTIVITY)
BUCKET_PARALLEL_MON_ACTIVITY, (AD1.BUCKET_ACTIVE_SECONDS - (AD1.BUCKET_OTHER_ACTIVITY +
AD1.BUCKET_SERIAL_MON_ACTIVITY) * :B25 ) BUCKET_PARALLEL_MON_SECONDS, (AD1.BUCKET_OTHER_ACTIVITY * :B25
) BUCKET_OTHER_SECONDS, DECODE(AD1.PX_DFO_PAIR_ID, NULL, NULL, SUM(AD1.ACTIVITY_COUNT) OVER(PARTITION
BY DECODE(AD1.PX_DFO_PAIR_ID, NULL, NULL, AD1.BUCKET_NUM), AD1.PX_DFO_PAIR_ID)) DFO_PAIR_TOTAL_ACTIVITY,
DECODE(AD1.PX_DFO_PAIR_ID, NULL, NULL, SUM(DECODE(AD1.ACTIVITY_TYPE, 'Cpu', AD1.ACTIVITY_COUNT, 0))
OVER(PARTITION BY DECODE(AD1.PX_DFO_PAIR_ID, NULL, NULL, AD1.BUCKET_NUM), AD1.PX_DFO_PAIR_ID))
DFO_PAIR_CPU_ACTIVITY FROM (SELECT AD01.*, NVL((SUM(DECODE(AD01.IS_MONITORED_SQL, 1,
AD01.ACTIVITY_COUNT, NULL)) OVER(PARTITION BY AD01.BUCKET_NUM)), 0) BUCKET_TOTAL_MON_ACTIVITY,
(NVL(SUM(CASE WHEN AD01.IS_MONITORED_SQL = 1 AND AD01.PX_ID IS NULL THEN AD01.ACTIVITY_COUNT ELSE NULL
END) OVER(PARTITION BY AD01.BUCKET_NUM), 0)) BUCKET_SERIAL_MON_ACTIVITY, (NVL((SUM(DECODE(AD01.IS_
MONITORED_SQL, 0, AD01.SQL_BUCKET_DISTINCT_SAMPLES, NULL)) OVER(PARTITION BY AD01.BUCKET_NUM)), 0))
BUCKET_OTHER_ACTIVITY, (NVL(AD01.BUCKET_DISTINCT_SAMPLES, 0) * :B25 ) BUCKET_ACTIVE_SECONDS,
DECODE(AD01.BUCKET_NUM, :B27 , MOD(:B32 , :B26 ) , :B26 ) BUCKET_INTERVAL FROM (SELECT
AD0.ACTIVITY_BUCKET_NUM BUCKET_NUM, AD0.PX_ID, AD0.ACTIVITY_TYPE, AD0.OTHER_SQL_ACTIVITY_TYPE,
AD0.EVENT_NAME, AD0.PLAN_LINE_ID, AD0.PX_STEP_ID, AD0.PX_STEP_ARG, AD0.PLSQL_OBJECT_ID,
AD0.PLSQL_SUBPROGRAM_ID, SUM(AD0.ACTIVITY_COUNT) ACTIVITY_COUNT, MIN(AD0.ACTIVITY_START)
ACTIVITY_START, MAX(AD0.ACTIVITY_END) ACTIVITY_END, MAX(AD0.IS_MONITORED_SQL) IS_MONITORED_SQL,
MAX(AD0.SQL) SQL, MAX(AD0.TOP_LEVEL_SQL_ID) TOP_LEVEL_SQL_ID, MAX(AD0.DBOP_NAME) DBOP_NAME,
MAX(DECODE(AD0.DFO_MOST_ACTIVE_COUNT, NULL, NULL, AD0.DFO_MOST_ACTIVE_ID)) DFO_MOST_ACTIVE_ID,
MAX(DECODE(AD0.DFO_MOST_ACTIVE_COUNT, NULL, NULL, AD0.DFO_MOST_ACTIVE_SID)) DFO_MOST_ACTIVE_SID,
SUM(AD0.DFO_MOST_ACTIVE_COUNT) DFO_MOST_ACTIVE_COUNT, MAX(PX_DFO_PAIR_TYPE) PX_DFO_PAIR_TYPE,
MAX(PX_DFO_PAIR_ID) PX_DFO_PAIR_ID, MAX(AD0.PX_DOP) PX_DOP, MIN(AD0.PX_MIN_DOP) PX_MIN_DOP,
MAX(AD0.BUCKET_DISTINCT_SAMPLES) BUCKET_DISTINCT_SAMPLES, MAX(AD0.SQL_BUCKET_DISTINCT_SAMPLES)
SQL_BUCKET_DISTINCT_SAMPLES FROM (SELECT AD00.*, (CASE WHEN AD00.IS_MONITORED_SQL = 1 AND
(AD00.SESSION_ID != :B19 OR AD00.INST_ID != :B17 ) THEN AD00.PX_SERVER_GROUP END) PX_ID FROM ASH_DATA AD00
WHERE :B33 = 1) AD0 GROUP BY AD0.ACTIVITY_BUCKET_NUM, AD0.PX_ID, AD0.ACTIVITY_TYPE, AD0.EVENT_NAME,
AD0.OTHER_SQL_ACTIVITY_TYPE, AD0.PLAN_LINE_ID, AD0.PLSQL_OBJECT_ID, AD0.PLSQL_SUBPROGRAM_ID,
AD0.PX_STEP_ID, AD0.PX_STEP_ARG ) AD01) AD1) AD2) AD3) ADH_0) ADH_1) ADH) SELECT /*+ no_monitor
no_xml_query_rewrite */ XMLELEMENT("sql_monitor_report", XMLATTRIBUTES(:B59 AS "version", TO_CHAR(:B58 , :B12 ) AS
"sysdate"), XMLELEMENT("report_parameters", NULL, CASE WHEN :B57 IS NOT NULL THEN XMLFOREST(:B57 AS
"dbop_name", :B56 AS "dbop_exec_id") ELSE XMLFOREST(:B55 AS "sql_id", :B54 AS "sql_exec_id") END, XMLFOREST(:B53 AS
"session_id", :B52 AS "session_serial", TO_CHAR(:B51 , :B12 ) AS "sql_exec_start", :B27 AS "bucket_count", TO_CHAR(:B50 , :B12 )
AS "interval_start", TO_CHAR(:B22 , :B12 ) AS "interval_end", DECODE(:B49 , 'Y', :B48 , NULL) AS "auto_refresh", :B47 AS
"base_path"), CASE WHEN :B23 = 'Y' AND :B46 IS NOT NULL AND NOT (:B4 = 1 AND :B3 IS NULL AND :B2 IS NULL AND :B1 IS
NULL) THEN XMLELEMENT("parallel_filter", NULL, XMLFOREST( DECODE(:B4 , 1, 'yes', 'no') AS "qc", :B3 AS "server_group", :B2
AS "server_set", :B1 AS "server_number")) ELSE NULL END), CASE WHEN :B14 = 1 THEN XMLELEMENT(
"derived_parameters_testing", NULL, XMLFOREST(:B11 AS "instance_id_low", :B10 AS "instance_id_high", :B26 AS
"bucket_interval_sec", :B32 AS "interval_second", :B27 AS "bucket_calc_count", :B45 AS "bucket_calc_max_count", :B13 AS
"sel_sqltext")) ELSE NULL END, (SELECT XMLCONCAT( CASE WHEN :B46 IS NULL AND :B60 IS NULL THEN XMLELEMENT(
"target", XMLATTRIBUTES(:B17 AS "instance_id", :B19 AS "session_id", :B18 AS "session_serial", NVL2(:B6 , NULL, :B9 ) AS "sql_id",
NVL2(:B6 , NULL, TO_CHAR(:B8 , :B12 )) AS "sql_exec_start", NVL2(:B6 , NULL, :B7 ) AS "sql_exec_id", NVL2(:B6 , :B6 , NULL) AS
"dbop_name", NVL2(:B6 , :B5 , NULL) AS "dbop_exec_id", NVL2(:B6 , TO_CHAR(:B8 , :B12 ) , NULL) AS "dbop_exec_start", NVL2(:B6
, NULL, MAX_PLAN_HASH_VALUE) AS "sql_plan_hash", NVL2(:B6 , NULL, MAX_FULL_PLAN_HASH_VALUE) AS
"sql_full_plan_hash", MAGG.DB_UNIQUE_NAME AS "db_unique_name", MAGG.PLATFORM_NAME AS "db_platform_name",
MAGG.HOST_NAME AS "report_host_name"), NVL2(SUM_SERVERS_REQUESTED, XMLELEMENT("servers_requested", NULL,
SUM_SERVERS_REQUESTED), NULL), NVL2(SUM_SERVERS_ALLOCATED, XMLELEMENT("servers_allocated", NULL,
SUM_SERVERS_ALLOCATED), NULL), NVL2(MAX_USERID, XMLELEMENT("user_id", NULL, MAX_USERID), NULL),
NVL2(MAX_USERNAME, XMLELEMENT("user", NULL, MAX_USERNAME), NULL), NVL2(MAX_CON_ID, XMLELEMENT("con_id",
NULL, MAX_CON_ID), NULL), NVL2(MAX_CON_NAME, XMLELEMENT("con_name", NULL, MAX_CON_NAME), NULL),
NVL2(MAX_PROGRAM, XMLELEMENT("program", NULL, MAX_PROGRAM), NULL), NVL2(MAX_MODULE, XMLELEMENT(
"module", NULL, MAX_MODULE), NULL), NVL2(MAX_ACTION, XMLELEMENT("action", NULL, MAX_ACTION), NULL),
NVL2(MAX_SERVICE_NAME, XMLELEMENT("service", NULL, MAX_SERVICE_NAME), NULL), NVL2(MAX_CLIENT_ID,
XMLELEMENT("client_id", NULL, MAX_CLIENT_ID), NULL), NVL2(MAX_CLIENT_INFO, XMLELEMENT("client_info", NULL,
MAX_CLIENT_INFO), NULL), NVL2(MAX_PL_ENTRY_OID, XMLELEMENT("plsql_entry_object_id", NULL, MAX_PL_ENTRY_OID),
NULL), NVL2(MAX_PL_ENTRY_PROGID, XMLELEMENT("plsql_entry_subprogram_id", NULL, MAX_PL_ENTRY_PROGID), NULL),
NVL2(MAX_PL_ENTRY_NAME, XMLELEMENT("plsql_entry_name", NULL, MAX_PL_ENTRY_NAME), NULL), NVL2(MAX_PL_OID,
XMLELEMENT("plsql_object_id", NULL, MAX_PL_OID), NULL), NVL2(MAX_PL_PROGID, XMLELEMENT("plsql_subprogram_id",
NULL, MAX_PL_PROGID), NULL), NVL2(MAX_PL_NAME, XMLELEMENT("plsql_name", NULL, MAX_PL_NAME), NULL), CASE
WHEN (:B13 = 0 OR :B6 IS NOT NULL) THEN NULL ELSE XMLELEMENT("sql_fulltext", XMLATTRIBUTES(NVL2(SQL_VTEXT, 'Y',
IS_FULL_TEXT) AS "is_full"), NVL2(SQL_VTEXT, SQL_VTEXT, SQLMON_TEXT)) END, XMLELEMENT("status", NULL,
MAX_STATUS), XMLELEMENT("refresh_count", NULL, SUM_REFRESH_COUNT), XMLELEMENT("first_refresh_time", NULL,
TO_CHAR(MIN_FIRST_REFRESH_TIME, :B12 )), XMLELEMENT("last_refresh_time", NULL, TO_CHAR(:B58 , :B12 )),
XMLELEMENT("duration", NULL, GREATEST(:B65 , LEAST(MAX_ELAPSED_TIME/1000000, 1)),
CEIL(MAX_QUEUEING_TIME/1000000))), DECODE(MAX_RM_INFO, '00/00/0000 00 :00:00', NULL,
XMLTYPE(SUBSTR(MAX_RM_INFO, 20))), CASE WHEN (:B63 = 'Y') THEN XMLELEMENT("adaptive_plan", XMLATTRIBUTES(:B64

```

```

AS "is_final"), :B63 ) ELSE NULL END, NVL((SELECT XMLFOREST( XMLAGG( XMLELEMENT( "param", XMLATTRIBUTES(E.NAME
AS "name"), DECODE(:B14 , 1, 'XXXX', E.VALUE)) ORDER BY E.NAME) AS "optimizer_env") SQL_ENV FROM
GV$SQL_OPTIMIZER_ENV E WHERE E.INST_ID = :B17 AND E.SQL_ID = :B9 AND E.CHILD_ADDRESS = HEXTORAW(:B66 ) AND
(DECODE(:B14 , 1, 'YES', E.ISDEFAULT) = 'NO' OR E.ID IN (2, 12, 24, 35, 36, 37, 246, 256, 257, 274, 275, 289, 290))), (SELECT
XMLELEMENT( "optimizer_env", XMLATTRIBUTES('sys' AS "type"), XMLAGG( XMLELEMENT( "param", XMLATTRIBUTES(E.NAME
AS "name"), DECODE(:B14 , 1, 'XXXX', E.VALUE)) ORDER BY E.NAME)) SYS_ENV FROM V$SYS_OPTIMIZER_ENV E WHERE
(DECODE(:B14 , 1, 'YES', E.ISDEFAULT) = 'NO' OR E.ID IN (2, 12, 24, 35, 36, 37, 24 6, 256, 257, 274, 275, 289, 290)))) ) END,
XMLELEMENT( "stats", XMLATTRIBUTES('monitor' AS "type"), DECODE(NVL(SUM_ELAPSED_TIME, 0), 0, NULL, XMLELEMENT(
"stat", XMLATTRIBUTES('elapsed_time' AS "name"), SUM_ELAPSED_TIME)), DECODE(NVL(MAX_QUEUEING_TIME, 0), 0, NULL,
XMLELEMENT( "stat", XMLATTRIBUTES('queueing_time' AS "name"), MAX_QUEUEING_TIME)), DECODE(NVL(SUM_CPU_TIME, 0),
0, NULL, XMLELEMENT( "stat", XMLATTRIBUTES('cpu_time' AS "name"), SUM_CPU_TIME)),
DECODE(NVL(SUM_USER_IO_WAIT_TIME, 0), 0, NULL, XMLELEMENT( "stat", XMLATTRIBUTES('user_io_wait_time' AS "name"),
SUM_USER_IO_WAIT_TIME)), DECODE(NVL(SUM_APPLICATION_WAIT_TIME, 0), 0, NULL, XMLELEMENT( "stat",
XMLATTRIBUTES('application_wait_time' AS "name"), SUM_APPLICATION_WAIT_TIME)), DECODE(NVL(SUM_CONCURRENCY_
WAIT_TIME, 0), 0, NULL, XMLELEMENT( "stat", XMLATTRIBUTES('concurrency_wait_time' AS "name"),
SUM_CONCURRENCY_WAIT_TIME)), DECODE(NVL(SUM_CLUSTER_WAIT_TIME, 0), 0, NULL, XMLELEMENT( "stat",
XMLATTRIBUTES('cluster_wait_time' AS "name"), SUM_CLUSTER_WAIT_TIME)), DECODE(NVL(SUM_PLSQL_EXEC_TIME, 0), 0,
NULL, XMLELEMENT( "stat", XMLATTRIBUTES('plsql_exec_time' AS "name"), SUM_PLSQL_EXEC_TIME)),
DECODE(NVL(SUM_JAVA_EXEC_TIME, 0), 0, NULL, XMLELEMENT( "stat", XMLATTRIBUTES('java_exec_time' AS "name"),
SUM_JAVA_EXEC_TIME)), DECODE(NVL(SUM_OTHER_WAIT_TIME, 0), 0, NULL, XMLELEMENT( "stat",
XMLATTRIBUTES('other_wait_time' AS "name"), SUM_OTHER_WAIT_TIME)), DECODE(NVL(SUM_FETCHES, 0), 0, NULL,
XMLELEMENT( "stat", XMLATTRIBUTES('user_fetch_count' AS "name"), SUM_FETCHES)), DECODE(NVL(SUM_BUFFER_GETS,
0), 0, NULL, XMLELEMENT( "stat", XMLATTRIBUTES('buffer_gets' AS "name"), SUM_BUFFER_GETS)),
DECODE(NVL(SUM_READ_REQS, 0), 0, NULL, XMLELEMENT( "stat", XMLATTRIBUTES('disk_reads' AS "name"),
SUM_READ_REQS)), DECODE(NVL(SUM_WRITE_REQS, 0), 0, NULL, XMLELEMENT( "stat", XMLATTRIBUTES('direct_writes' AS
"name"), SUM_WRITE_REQS)), DECODE(NVL(SUM_READ_REQS, 0), 0, NULL, XMLELEMENT( "stat",
XMLATTRIBUTES('read_reqs' AS "name"), SUM_READ_REQS)), DECODE(NVL(SUM_READ_BYTES, 0), 0, NULL, XMLELEMENT(
"stat", XMLATTRIBUTES('read_bytes' AS "name"), SUM_READ_BYTES)), DECODE(NVL(SUM_WRITE_REQS, 0), 0, NULL,
XMLELEMENT( "stat", XMLATTRIBUTES('write_reqs' AS "name"), SUM_WRITE_REQS)), DECODE(NVL(SUM_WRITE_BYTES, 0), 0,
NULL, XMLELEMENT( "stat", XMLATTRIBUTES('write_bytes' AS "name"), SUM_WRITE_BYTES)), CASE WHEN
SUM_IO_INTER_BYTES IS NULL OR SUM_IO_BYTES = 0 OR SUM_IO_INTER_BYTES = SUM_IO_BYTES THEN NULL ELSE
XMLELEMENT( "stat", XMLATTRIBUTES('cell_offload_efficiency' AS "name"), ROUND(SUM_IO_BYTES /
DECODE(SUM_IO_INTER_BYTES, 0, 1, SUM_IO_INTER_BYTES, 2))) END, CASE WHEN :B33 = 1 THEN (SELECT CASE WHEN
SUM(ACTIVITY_COUNT) > 0 THEN XMLELEMENT( "activity_sampled", XMLAGG( XMLELEMENT( "activity", XMLATTRIBUTES(
AD1.ACTIVITY_TYPE AS "class", AD1.EVENT_NAME AS "event"), AD1.ACTIVITY_COUNT) ORDER BY AD1.ACTIVITY_TYPE,
AD1.EVENT_NAME)) ELSE NULL END FROM (SELECT AD0.ACTIVITY_TYPE, AD0.EVENT_NAME, SUM(AD0.ACTIVITY_COUNT)
ACTIVITY_COUNT FROM ASH_DATA AD0 GROUP BY AD0.ACTIVITY_TYPE, AD0.EVENT_NAME) AD1) ELSE NULL END,
NVL2(MAX_ERROR_NUMBER, XMLELEMENT( "error", XMLATTRIBUTES(MAX_ERROR_NUMBER AS "number",
MAX_ERROR_FACILITY AS "facility"), MAX_ERROR_MESSAGE), NULL), CASE WHEN :B62 = 1 AND
MAGG.QC_HAS_BINDS_XML = 'Y' THEN (SELECT XMLTYPE(BINDS_XML) FROM GV$SQL_MONITOR MON WHERE
MON.INST_ID = :B17 AND MON.KEY = MAGG.MAX_KEY AND MON.SID = MAGG.MAX_SESSION_ID AND MON.SQL_ID = :B9
AND MON.SQL_EXEC_START = :B8 AND MON.SQL_EXEC_ID = :B7 AND ROWNUM = 1) ELSE NULL END, CASE WHEN :B61 = 1
AND MAGG.QC_HAS_OTHER_XML = 'Y' THEN (SELECT XMLTYPE(OTHER_XML) FROM GV$SQL_MONITOR MON WHERE
MON.INST_ID = MAGG.MAX_INST_ID AND MON.KEY = MAGG.MAX_KEY AND MON.SID = MAGG.MAX_SESSION_ID AND
MON.SQL_ID = :B9 AND MON.SQL_EXEC_START = :B8 AND MON.SQL_EXEC_ID = :B7 AND ROWNUM = 1) ELSE NULL END)
FROM (SELECT V.*, CASE WHEN :B13 = 2 AND IS_FULL_TEXT = 'N' AND :B46 IS NULL AND :B60 IS NULL THEN (SELECT
SQL_FULLTEXT FROM GV$SQL SQ WHERE SQ.INST_ID BETWEEN :B11 AND :B10 AND SQ.SQL_ID = :B9 AND ROWNUM = 1)
ELSE NULL END SQL_VTEXT FROM MONITOR_AGG V) MAGG, CASE WHEN :B44 = 1 THEN (SELECT CASE WHEN
AT.ACTIVITY_COUNT > 0 THEN XMLELEMENT( "activity_sampled", XMLATTRIBUTES( :B41 AS "ash_missing_seconds",
TO_CHAR(AT.ACTIVITY_START, :B12 ) AS "first_sample_time", TO_CHAR(AT.ACTIVITY_END, :B12 ) AS "last_sample_time",
ROUND((AT.ACTIVITY_END - AT.ACTIVITY_START) * 3600 * 24) + 1 AS "duration", AT.ACTIVITY_COUNT AS "count",
AT.IMQ_COUNT AS "imq_count", AT.WAIT_COUNT AS "wait_count", AT.CPU_COUNT AS "cpu_count",
DECODE(AT.OTHER_SQL_COUNT, 0, NULL, AT.OTHER_SQL_COUNT) AS "other_sql_count", :B40 AS "cpu_cores", :B39 AS
"hyperthread"), AT.ACTIVITY_TOTAL, AH.GLOB_ACTIVITY_HISTO) WHEN :B41 IS NOT NULL THEN XMLELEMENT(
"activity_sampled", XMLATTRIBUTES( DECODE( :B41 , -1, 'all', TO_CHAR( :B41 )) AS "ash_missing_seconds")) ELSE NULL END
FROM (SELECT MIN(AD1.ACTIVITY_START) ACTIVITY_START, MAX(AD1.ACTIVITY_END) ACTIVITY_END,
SUM(AD1.ACTIVITY_COUNT) ACTIVITY_COUNT, SUM(AD1.IMQ_COUNT) IMQ_COUNT, SUM(AD1.WAIT_COUNT) WAIT_COUNT,
SUM(AD1.CPU_COUNT) CPU_COUNT, SUM(AD1.OTHER_SQL_COUNT) OTHER_SQL_COUNT,
SUBSTR(MAX(LPAD(AD1.ACTIVITY_COUNT, 10)|| AD1.ACTIVITY_TYPE), 11) MOST_ACTIVE, XMLAGG( XMLELEMENT( "activity",
XMLATTRIBUTES( AD1.ACTIVITY_TYPE AS "class", AD1.EVENT_NAME AS "event"), AD1.ACTIVITY_COUNT) ORDER BY
AD1.ACTIVITY_TYPE, AD1.EVENT_NAME) ACTIVITY_TOTAL FROM (SELECT AD0.ACTIVITY_TYPE, AD0.EVENT_NAME,
MIN(AD0.ACTIVITY_START) ACTIVITY_START, MAX(AD0.ACTIVITY_END) ACTIVITY_END, SUM(AD0.ACTIVITY_COUNT)
ACTIVITY_COUNT, SUM(AD0.IMQ_COUNT) IMQ_COUNT, SUM(AD0.WAIT_COUNT) WAIT_COUNT, SUM(AD0.CPU_COUNT)
CPU_COUNT, SUM(AD0.OTHER_SQL_COUNT) OTHER_SQL_COUNT FROM ASH_DATA AD0 GROUP BY AD0.ACTIVITY_TYPE,
AD0.EVENT_NAME) AD1) AT, (SELECT CASE WHEN :B30 = 1 AND :B27 > 1 THEN XMLELEMENT( "activity_histogram",
XMLATTRIBUTES( :B26 AS "bucket_interval", :B27 AS "bucket_count", TO_CHAR( :B20 , :B12 ) AS "start_time", TO_CHAR( :B22 ,
:B12 ) AS "end_time", ROUND(( :B22 - :B20 ) * 3600 * 24) + 1 AS "duration"), XMLAGG( XMLELEMENT( "bucket", XMLATTRIBUTES(
AD2.BUCKET_NUM AS "number"), ACTIVITY_BUCKET) ORDER BY AD2.BUCKET_NUM)) ELSE NULL END
GLOB_ACTIVITY_HISTO FROM (SELECT AD1.BUCKET_NUM, SUM(ACTIVITY_COUNT) ACTIVITY_COUNT, SUM(IMQ_COUNT)
IMQ_COUNT, SUM(WAIT_COUNT) WAIT_COUNT, SUM(CPU_COUNT) CPU_COUNT, SUM(OTHER_SQL_COUNT)
OTHER_SQL_COUNT, MIN(AD1.ACTIVITY_START) ACTIVITY_START, MAX(AD1.ACTIVITY_END) ACTIVITY_END,
MIN(AD1.BUCKET_ACTIVITY_START) BUCKET_ACTIVITY_START, MAX(AD1.BUCKET_ACTIVITY_END)
BUCKET_ACTIVITY_END, SUBSTR(MAX(LPAD(AD1.ACTIVITY_COUNT, 10)|| AD1.ACTIVITY_TYPE), 11) MOST_ACTIVE,
XMLAGG( XMLELEMENT( "activity", XMLATTRIBUTES( AD1.ACTIVITY_TYPE AS "class", AD1.EVENT_NAME AS "event"),
AD1.ACTIVITY_COUNT) ORDER BY AD1.ACTIVITY_TYPE, AD1.EVENT_NAME) ACTIVITY_BUCKET FROM (SELECT
AD0.ACTIVITY_BUCKET_NUM BUCKET_NUM, AD0.ACTIVITY_TYPE, AD0.EVENT_NAME, MIN(AD0.ACTIVITY_START)
ACTIVITY_START, MAX(AD0.ACTIVITY_END) ACTIVITY_END, SUM(AD0.ACTIVITY_COUNT) ACTIVITY_COUNT,
SUM(AD0.IMQ_COUNT) IMQ_COUNT, SUM(AD0.WAIT_COUNT) WAIT_COUNT, SUM(AD0.CPU_COUNT) CPU_COUNT,
SUM(AD0.OTHER_SQL_COUNT) OTHER_SQL_COUNT, MIN(AD0.BUCKET_ACTIVITY_START) BUCKET_ACTIVITY_START,
MAX(AD0.BUCKET_ACTIVITY_END) BUCKET_ACTIVITY_END FROM ASH_DATA AD0 GROUP BY
AD0.ACTIVITY_BUCKET_NUM, AD0.ACTIVITY_TYPE, AD0.EVENT_NAME) AD1 GROUP BY AD1.BUCKET_NUM) AD2) AH) ELSE

```

```

NULL END, CASE WHEN :B33 = 1 THEN (SELECT CASE WHEN AH.ACTIVITY_COUNT > 0 THEN XMLELEMENT( "activity_detail",
XMLATTRIBUTES( TO_CHAR( :B20 , :B12 ) AS "start_time", TO_CHAR( :B22 , :B12 ) AS "end_time", :B41 AS
"ash_missing_seconds", TO_CHAR(AH.ACTIVITY_START, :B12 ) AS "first_sample_time", TO_CHAR(AH.ACTIVITY_END, :B12 ) AS
"last_sample_time", ROUND((AH.ACTIVITY_END - AH.ACTIVITY_START) * 3600 * 24) + 1 AS "duration", :B25 AS "sample_interval",
:B26 AS "bucket_interval", :B27 AS "bucket_count", ROUND((:B22 - :B20) * 3600 * 24) + 1 AS "bucket_duration", :B40 AS "cpu_cores",
:B34 AS "total_cpu_cores", :B39 AS "hyperthread"), AH.GLOB_ACTIVITY_HISTO) WHEN :B41 IS NOT NULL THEN XMLELEMENT(
"activity_detail", XMLATTRIBUTES( DECODE( :B41 , -1, 'all', TO_CHAR( :B41 )) AS "ash_missing_seconds")) ELSE NULL END
FROM (SELECT MIN(AD2.ACTIVITY_START) ACTIVITY_START, MAX(AD2.ACTIVITY_END) ACTIVITY_END,
SUM(AD2.ACTIVITY_COUNT) ACTIVITY_COUNT, XMLAGG( XMLELEMENT( "bucket", XMLATTRIBUTES( AD2.BUCKET_NUM AS
"number"), ACTIVITY_BUCKET_XML) ORDER BY AD2.BUCKET_NUM) GLOB_ACTIVITY_HISTO FROM (SELECT
AD1.BUCKET_NUM, MIN(AD1.ACTIVITY_START) ACTIVITY_START, MAX(AD1.ACTIVITY_END) ACTIVITY_END,
SUM(AD1.ACTIVITY_COUNT) ACTIVITY_COUNT, MAX(AD1.BUCKET_IDLE_SECONDS) BUCKET_IDLE_SECONDS, XMLAGG(
XMLCONCAT( CASE WHEN AD1.DFO_PAIR_IDLE_RESP_TIME != 0 AND DFO_PAIR_ROWNUM = 1 THEN XMLELEMENT (
"activity", XMLATTRIBUTES( 'Parallel Skew' AS "class", AD1.PX_STEP_ARG AS "line", AD1.PX_ID AS "px",
ROUND(AD1.DFO_PAIR_IDLE_RESP_TIME, 2) AS "rt"), 0) ELSE NULL END, XMLELEMENT( "activity", XMLATTRIBUTES(
NVL(AD1.OTHER_SQL, AD1.RPI) AS "sql", AD1.NON_SQL AS "non_sql", AD1.CLASS AS "class", AD1.OTHER_SQL_CLASS AS
"other_sql_class", AD1.EVENT AS "event", AD1.PLAN_LINE_ID AS "line", NVL2(AD1.PLSQL_OBJECT_ID,
AD1.PLSQL_OBJECT_ID||':'||AD1.PLSQL_SUBPROGRAM_ID, NULL) AS "plsql_id", AD1.PLSQL_NAME AS "plsql_name", CASE
WHEN AD1.SQL_ROWNUM = 1 THEN AD1.TOP_LEVEL_SQL_ID END AS "top_sql_id", CASE WHEN AD1.DBOP_NAME IS NOT
NULL THEN AD1.DBOP_NAME END AS "dbop_name", CASE WHEN AD1.DFO_MOST_ACTIVE_IID IS NOT NULL AND :B67 = 'Y'
THEN AD1.DFO_MOST_ACTIVE_IID END AS "skew_iid", DECODE(AD1.DFO_MOST_ACTIVE_COUNT, NULL, NULL,
AD1.DFO_MOST_ACTIVE_SID) AS "skew_sid", AD1.DFO_MOST_ACTIVE_COUNT AS "skew_count", DECODE(AD1.PX_DOP, :B28
, NULL, AD1.PX_DOP) AS "dop", DECODE(AD1.PX_DOP, AD1.PX_MIN_DOP, NULL, AD1.PX_MIN_DOP) AS "min_dop", AD1.PX_ID
AS "px", AD1.PX_STEP_ID AS "step", AD1.PX_STEP_ARG AS "arg", DECODE(AD1.ACTIVITY_COUNT, AD1.RESP_TIME, NULL,
ROUND(AD1.RESP_TIME, 2)) AS "rt"), AD1.ACTIVITY_COUNT) ORDER BY AD1.PX_STEP_ID, AD1.PX_STEP_ARG,
AD1.DFO_PAIR_ROWNUM) ACTIVITY_BUCKET_XML FROM (SELECT AD01.*, CASE WHEN AD01.ACTIVITY_TYPE != 'Other SQL
Execution' AND AD01.ACTIVITY_TYPE != 'Non SQL' THEN AD01.ACTIVITY_TYPE END CLASS, CASE WHEN
(AD01.ACTIVITY_TYPE = 'Other SQL Execution' OR AD01.ACTIVITY_TYPE = 'Non SQL') THEN
AD01.OTHER_SQL_ACTIVITY_TYPE END OTHER_SQL_CLASS, CASE WHEN AD01.ACTIVITY_TYPE != 'Other SQL Execution'
AND AD01.ACTIVITY_TYPE = 'Non SQL' THEN AD01.EVENT_NAME END EVENT, CASE WHEN AD01.SQL IN ('this', 'anonymous')
THEN NULL ELSE AD01.SQL END RPI, DECODE(AD01.ACTIVITY_TYPE, 'Other SQL Execution', SUBSTR(AD01.EVENT_NAME,
9), NULL) OTHER_SQL, DECODE(AD01.ACTIVITY_TYPE, 'Non SQL', AD01.EVENT_NAME, NULL) NON_SQL, ROW_NUMBER()
OVER(PARTITION BY AD01.BUCKET_NUM, AD01.PX_DFO_PAIR_ID ORDER BY AD01.ACTIVITY_TYPE, AD01.EVENT_NAME,
AD01.PLAN_LINE_ID) DFO_PAIR_ROWNUM FROM RESPONSE_TIME_DATA AD01) AD1 GROUP BY AD1.BUCKET_NUM) AD2)
AH) ELSE NULL END, CASE WHEN :B23 = 'Y' THEN (SELECT XMLELEMENT( "parallel_info", XMLATTRIBUTES( :B17 AS
"qc_instance_id", MAX_PX_QCSID AS "qc_session_id", MAX_PX_IS_CROSS_INSTANCE AS "is_cross_instance", MAX_PX_DOP AS
"dop", MAX_PX_DOP_INSTANCES AS "max_dop_instances", DIST_INST_COUNT AS "inst_count", DIST_PX_GROUP_COUNT AS
"server_group_count", DIST_PX_SET_COUNT AS "server_set_count"), CASE WHEN :B69 = 1 THEN PX_SESSIONS ELSE NULL
END, CASE WHEN :B67 = 'Y' THEN DECODE(:B68, 1, PX_INSTANCES, NULL) ELSE NULL END) FROM (SELECT
MAX_PX_QCSID, MAX_PX_DOP, MAX_PX_DOP_INSTANCES, MAX_PX_IS_CROSS_INSTANCE, SUM_SERVERS_REQUESTED,
SUM_SERVERS_ALLOCATED, DIST_INST_COUNT, DIST_PX_GROUP_COUNT, DIST_PX_SET_COUNT, (SELECT XMLELEMENT(
"sessions", XMLATTRIBUTES(MAX(PX_SESSION.ACTIVITY_COUNT) AS "max_activity_count", MAX(PX_SESSION.IMQ_COUNT)
AS "max_imq_count", MAX(PX_SESSION.CPU_COUNT) AS "max_cpu_count", MAX(PX_SESSION.WAIT_COUNT) AS
"max_wait_count", MAX(PX_SESSION.OTHER_SQL_COUNT) AS "max_other_sql_count", MAX(PX_SESSION.MAX_IO_REQS) AS
"max_io_reqs", MAX(PX_SESSION.MAX_IO_BYTES) AS "max_io_bytes", MAX(PX_SESSION.MAX_BUFFER_GETS) AS
"max_buffer_gets", MAX(PX_SESSION.MAX_ELAPSED_TIME) AS "max_elapsed_time"), XMLAGG(PX_SESSION.PX_SESSION_
XML ORDER BY PX_SERVER_GROUP NULLS FIRST, PX_SERVER_SET, PX_SERVER#)) FROM (SELECT PX_SERVER_GROUP,
PX_SERVER_SET, PX_SERVER#, MAX(PI.MAX_ELAPSED_TIME) MAX_ELAPSED_TIME, MAX(PI.MAX_IO_REQS)
MAX_IO_REQS, MAX(PI.MAX_IO_BYTES) MAX_IO_BYTES, MAX(PI.MAX_BUFFER_GETS) MAX_BUFFER_GETS,
SUM(PI.ACTIVITY_COUNT) ACTIVITY_COUNT, SUM(PI.IMQ_COUNT) IMQ_COUNT, SUM(PI.WAIT_COUNT) WAIT_COUNT,
SUM(PI.CPU_COUNT) CPU_COUNT, SUM(PI.OTHER_SQL_COUNT) OTHER_SQL_COUNT, XMLELEMENT( "session",
XMLATTRIBUTES( INST_ID AS "inst_id", PROCESS_NAME AS "process_name", SID AS "session_id", SESSION_SERIAL# AS
"session_serial", PX_SERVER_GROUP AS "server_group", PX_SERVER_SET AS "server_set", PX_SERVER# AS "server_num"),
XMLELEMENT( "stats", XMLATTRIBUTES( 'monitor' AS "type"), NVL2(MAX(ELAPSED_TIME), XMLELEMENT( "stat",
XMLATTRIBUTES('elapsed_time' AS "name"), MAX(ELAPSED_TIME)), NULL), NVL2(MAX(QUEUING_TIME), XMLELEMENT( "stat",
XMLATTRIBUTES('queuing_time' AS "name"), MAX(QUEUING_TIME)), NULL), NVL2(MAX(CPU_TIME), XMLELEMENT( "stat",
XMLATTRIBUTES('cpu_time' AS "name"), MAX(CPU_TIME)), NULL), NVL2(MAX(USER_IO_WAIT_TIME), XMLELEMENT( "stat",
XMLATTRIBUTES('user_io_wait_time' AS "name"), MAX(USER_IO_WAIT_TIME)), NULL), NVL2(MAX(APPLICATION_WAIT_TIME),
XMLELEMENT( "stat", XMLATTRIBUTES('application_wait_time' AS "name"), MAX(APPLICATION_WAIT_TIME)), NULL),
NVL2(MAX(CONCURRENCY_WAIT_TIME), XMLELEMENT( "stat", XMLATTRIBUTES('concurrency_wait_time' AS "name"),
MAX(CONCURRENCY_WAIT_TIME)), NULL), NVL2(MAX(CLUSTER_WAIT_TIME), XMLELEMENT( "stat",
XMLATTRIBUTES('cluster_wait_time' AS "name"), MAX(CLUSTER_WAIT_TIME)), NULL), NVL2(MAX(PLSQL_EXEC_TIME),
XMLELEMENT( "stat", XMLATTRIBUTES('plsql_exec_time' AS "name"), MAX(PLSQL_EXEC_TIME)), NULL), NVL2(MAX(JAVA
_EXEC_TIME), XMLELEMENT( "stat", XMLATTRIBUTES('java_exec_time' AS "name"), MAX(JAVA_EXEC_TIME)), NULL),
NVL2(MAX(OTHER_WAIT_TIME), XMLELEMENT( "stat", XMLATTRIBUTES('other_wait_time' AS "name"),
MAX(OTHER_WAIT_TIME)), NULL), NVL2(MAX(FETCHES), XMLELEMENT( "stat", XMLATTRIBUTES('user_fetch_count' AS
"name"), MAX(FETCHES)), NULL), NVL2(MAX(BUFFER_GETS), XMLELEMENT( "stat", XMLATTRIBUTES('buffer_gets' AS "name"),
MAX(BUFFER_GETS)), NULL), NVL2(MAX(READ_REQS), XMLELEMENT( "stat", XMLATTRIBUTES('disk_reads' AS "name"),
MAX(READ_REQS)), NULL), NVL2(MAX(WRITE_REQS), XMLELEMENT( "stat", XMLATTRIBUTES('direct_writes' AS "name"),
MAX(WRITE_REQS)), NULL), NVL2(MAX(READ_REQS), XMLELEMENT( "stat", XMLATTRIBUTES('read_reqs' AS "name"),
MAX(READ_REQS)), NULL), NVL2(MAX(READ_BYTES), XMLELEMENT( "stat", XMLATTRIBUTES('read_bytes' AS "name"),
MAX(READ_BYTES)), NULL), NVL2(MAX(WRITE_REQS), XMLELEMENT( "stat", XMLATTRIBUTES('write_reqs' AS "name"),
MAX(WRITE_REQS)), NULL), NVL2(MAX(WRITE_BYTES), XMLELEMENT( "stat", XMLATTRIBUTES('write_bytes' AS "name"),
MAX(WRITE_BYTES)), NULL), CASE WHEN MAX(IO_INTER_BYTES) IS NULL OR NVL(MAX(IO_BYTES), 0) = 0 OR
MAX(IO_INTER_BYTES) = MAX(IO_BYTES) THEN NULL ELSE XMLELEMENT( "stat", XMLATTRIBUTES('cell_offload_efficiency' AS
"name"), ROUND(MAX(IO_BYTES) / DECODE(MAX(IO_INTER_BYTES), 0, 1, MAX(IO_INTER_BYTES)), 2) END), CASE WHEN
SUM(PI.ACTIVITY_COUNT) > 0 AND :B33 = 1 THEN XMLELEMENT( "activity_sampled", XMLATTRIBUTES(
TO_CHAR(MIN(PI.ACTIVITY_START), :B12 ) AS "first_sample_time", TO_CHAR(MAX(PI.ACTIVITY_END), :B12 ) AS
"last_sample_time", ROUND((MAX(PI.ACTIVITY_END) - MIN(PI.ACTIVITY_START)) * 3600 * 24) + 1 AS "duration", SU
M(PI.ACTIVITY_COUNT) AS "count", SUM(PI.IMQ_COUNT) AS "imq_count", SUM(PI.CPU_COUNT) AS "cpu_count",
SUM(PI.WAIT_COUNT) AS "wait_count", SUM(PI.OTHER_SQL_COUNT) AS "other_sql_count", :B40 AS "cpu_cores", :B39 AS

```

```

"hyperthread"), XMLAGG( NVL2(ACTIVITY_TYPE, XMLELEMENT("activity", XMLATTRIBUTES( PI.ACTIVITY_TYPE AS "class",
PI.EVENT_NAME AS "event"), ACTIVITY_COUNT), NULL) ORDER BY PI.ACTIVITY_TYPE, PI.EVENT_NAME)) ELSE NULL END,
CASE WHEN :B61 = 1 AND PI.HAS_OTHER_XML = 'Y' THEN (SELECT XMLTYPE(OTHER_XML) FROM GV$SQL_MONITOR MON
WHERE MON.INST_ID = PI.INST_ID AND MON.KEY = PI.KEY AND MON.SID = PI.SID AND MON.SQL_ID = :B9 AND
MON.SQL_EXEC_START = :B8 AND MON.SQL_EXEC_ID = :B7 AND ROWNUM = 1) ELSE NULL END) PX_SESSION_XML FROM
(SELECT MO.HAS_OTHER_XML, MO.KEY, MO.INST_ID, DECODE(MO.PROCESS_NAME, 'ora', 'PX Coordinator',
MO.PROCESS_NAME) PROCESS_NAME, MO.SID, MO.SESSION_SERIAL#, MO.PX_SERVER_GROUP, MO.PX_SERVER_SET,
MO.PX_SERVER#, ASH0.ACTIVITY_TYPE, ASH0.EVENT_NAME, MAX(MO.IO_REQS) MAX_IO_REQS, MAX(MO.IO_BYTES)
MAX_IO_BYTES, MAX(MO.BUFFER_GETS) MAX_BUFFER_GETS, MAX(MO.ELAPSED_TIME) MAX_ELAPSED_TIME,
SUM(DECODE(ASH0.ACTIVITY_TYPE, NULL, NULL, ASH0.ACTIVITY_COUNT)) ACTIVITY_COUNT,
SUM(DECODE(ASH0.ACTIVITY_TYPE, NULL, NULL, ASH0.IMQ_COUNT)) IMQ_COUNT, SUM(DECODE(ASH0.ACTIVITY_TYPE,
NULL, NULL, ASH0.WAIT_COUNT)) WAIT_COUNT, SUM(DECODE(ASH0.ACTIVITY_TYPE, NULL, NULL, ASH0.CPU_COUNT))
CPU_COUNT, SUM(DECODE(ASH0.ACTIVITY_TYPE, NULL, NULL, ASH0.OTHER_SQL_COUNT)) OTHER_SQL_COUNT,
MIN(ASH0.ACTIVITY_START) ACTIVITY_START, MAX(ASH0.ACTIVITY_END) ACTIVITY_END,
MAX(DECODE(MO.ELAPSED_TIME, 0, NULL, MO.ELAPSED_TIME)) ELAPSED_TIME, MAX(DECODE(MO.QUEUING_TIME, 0,
NULL, MO.QUEUING_TIME)) QUEUING_TIME, MAX(DECODE(MO.CPU_TIME, 0, NULL, CPU_TIME)) CPU_TIME,
MAX(DECODE(MO.FETCHES, 0, NULL, FETCHES)) FETCHES, MAX(DECODE(MO.BUFFER_GETS, 0, NULL,
MO.BUFFER_GETS)) BUFFER_GETS, MAX(DECODE(MO.IO_INTER_BYTES, 0, NULL, MO.IO_INTER_BYTES))
IO_INTER_BYTES, MAX(DECODE(MO.READ_REQS, 0, NULL, MO.READ_REQS)) READ_REQS,
MAX(DECODE(MO.READ_BYTES, 0, NULL, MO.READ_BYTES)) READ_BYTES, MAX(DECODE(MO.WRITE_REQS, 0, NULL,
MO.WRITE_REQS)) WRITE_REQS, MAX(DECODE(MO.WRITE_BYTES, 0, NULL, MO.WRITE_BYTES)) WRITE_BYTES,
MAX(DECODE(MO.IO_BYTES, 0, NULL, MO.IO_BYTES)) IO_BYTES, MAX(DECODE(MO.APPLICATION_WAIT_TIME, 0, NULL,
MO.APPLICATION_WAIT_TIME)) APPLICATION_WAIT_TIME, MAX(DECODE(MO.CONCURRENCY_WAIT_TIME, 0, NULL,
MO.CONCURRENCY_WAIT_TIME)) CONCURRENCY_WAIT_TIME, MAX(DECODE(MO.CLUSTER_WAIT_TIME, 0, NULL,
MO.CLUSTER_WAIT_TIME)) CLUSTER_WAIT_TIME, MAX(DECODE(MO.USER_IO_WAIT_TIME, 0, NULL,
MO.USER_IO_WAIT_TIME)) USER_IO_WAIT_TIME, MAX(DECODE(PLSQL_EXEC_TIME, 0, NULL, PLSQL_EXEC_TIME))
PLSQL_EXEC_TIME, MAX(DECODE(MO.JAVA_EXEC_TIME, 0, NULL, MO.JAVA_EXEC_TIME)) JAVA_EXEC_TIME, MAX(DECO
DE(MO.OTHER_WAIT_TIME, 0, NULL, MO.OTHER_WAIT_TIME)) OTHER_WAIT_TIME FROM MONITOR_DATA MO, (SELECT
ASH1.INST_ID, ASH1.SESSION_ID, ASH1.ACTIVITY_TYPE, ASH1.EVENT_NAME, SUM(ASH1.ACTIVITY_COUNT)
ACTIVITY_COUNT, SUM(ASH1.IMQ_COUNT) IMQ_COUNT, SUM(ASH1.WAIT_COUNT) WAIT_COUNT, SUM(ASH1.CPU_COUNT)
CPU_COUNT, SUM(ASH1.OTHER_SQL_COUNT) OTHER_SQL_COUNT, MIN(ASH1.ACTIVITY_START) ACTIVITY_START,
MAX(ASH1.ACTIVITY_END) ACTIVITY_END FROM ASH_DATA ASH1 GROUP BY ASH1.INST_ID, ASH1.SESSION_ID,
ASH1.ACTIVITY_TYPE, ASH1.EVENT_NAME) ASH0 WHERE MO.INST_ID = ASH0.INST_ID(+) AND MO.SID =
ASH0.SESSION_ID(+) AND (:B69 = 1 OR :B68 = 1) GROUP BY MO.INST_ID, MO.KEY, MO.HAS_OTHER_XML,
MO.PROCESS_NAME, MO.SID, MO.SESSION_SERIAL#, MO.PX_SERVER_GROUP, MO.PX_SERVER_SET, MO.PX_SERVER#,
ASH0.ACTIVITY_TYPE, ASH0.EVENT_NAME) PI WHERE (:B69 = 1) GROUP BY PI.INST_ID, PI.KEY, PI.HAS_OTHER_XML, PI.SID,
PI.PROCESS_NAME, PI.SESSION_SERIAL#, PI.PX_SERVER_GROUP, PI.PX_SERVER_SET, PI.PX_SERVER#) P X_SESSION)
PX_SESSIONS, (SELECT XMLELEMENT("instances", XMLATTRIBUTES( MAX(PX_INSTANCE.ACTIVITY_COUNT) AS
"max_activity_count", MAX(PX_INSTANCE.IMQ_COUNT) AS "max_imq_count", MAX(PX_INSTANCE.CPU_COUNT) AS
"max_cpu_count", MAX(PX_INSTANCE.WAIT_COUNT) AS "max_wait_count", MAX(PX_INSTANCE.OTHER_SQL_COUNT) AS
"max_other_sql_count", MAX(PX_INSTANCE.ELAPSED_TIME) AS "max_elapsed_time", MAX(PX_INSTANCE.BUFFER_GETS) AS
"max_buffer_gets", MAX(PX_INSTANCE.IO_REQS) AS "max_io_reqs", MAX(PX_INSTANCE.IO_BYTES) AS "max_io_bytes"),
XMLAGG(PX_INSTANCE.PX_INSTANCES_XML ORDER BY INST_ID)) FROM (SELECT PI.INST_ID, MAX(PI.ELAPSED_TIME)
ELAPSED_TIME, MAX(PI.IO_REQS) IO_REQS, MAX(PI.IO_BYTES) IO_BYTES, MAX(PI.BUFFER_GETS) BUFFER_GETS,
SUM(PI.ACTIVITY_COUNT) ACTIVITY_COUNT, SUM(PI.IMQ_COUNT) IMQ_COUNT, SUM(PI.WAIT_COUNT) WAIT_COUNT,
SUM(PI.CPU_COUNT) CPU_COUNT, SUM(PI.OTHER_SQL_COUNT) OTHER_SQL_COUNT, XMLELEMENT("instance",
XMLATTRIBUTES( INST_ID AS "inst_id"), XMLELEMENT("stats", XMLATTRIBUTES('monitor' AS "type"),
NVL2(MAX(ELAPSED_TIME), XMLELEMENT("stat", XMLATTRIBUTES('elapsed_time' AS "name"), MAX(ELAPSED_TIME)), NULL),
NVL2(MAX(QUEUING_TIME), XMLELEMENT("stat", XMLATTRIBUTES('queuing_time' AS "name"), MAX(QUEUING_TIME)), NULL),
NVL2(MAX(CPU_TIME), XMLELEMENT("stat", XMLATTRIBUTES('cpu_time' AS "name"), MAX(CPU_TIME)), NULL),
NVL2(MAX(USER_IO_WAIT_TIME), XMLELEMENT("stat", XMLATTRIBUTES('user_io_wait_time' AS "name"),
MAX(USER_IO_WAIT_TIME)), NULL), NVL2(MAX(APPLICATION_WAIT_TIME), XMLELEMENT("stat",
XMLATTRIBUTES('application_wait_time' AS "name"), MAX(APPLICATION_WAIT_TIME)), NULL),
NVL2(MAX(CONCURRENCY_WAIT_TIME), XMLELEMENT("stat", XMLATTRIBUTES('concurrency_wait_time' AS "name"),
MAX(CONCURRENCY_WAIT_TIME)), NULL), NVL2(MAX(CLUSTER_WAIT_TIME), XMLELEMENT("stat",
XMLATTRIBUTES('cluster_wait_time' AS "name"), MAX(CLUSTER_WAIT_TIME)), NULL), NVL2(MAX(PLSQL_EXEC_TIME),
XMLELEMENT("stat", XMLATTRIBUTES('plsql_exec_time' AS "name"), MAX(PLSQL_EXEC_TIME)), NULL),
NVL2(MAX(JAVA_EXEC_TIME), XMLELEMENT("stat", XMLATTRIBUTES('java_exec_time' AS "name"), MAX(JAVA_EXEC_TIME)),
NULL), NVL2(MAX(OTHER_WAIT_TIME), XMLELEMENT("stat", XMLATTRIBUTES('other_wait_time' AS "name"),
MAX(OTHER_WAIT_TIME)), NULL), NVL2(MAX(FETCHES), XMLELEMENT("stat", XMLATTRIBUTES('user_fetch_count' AS
"name"), MAX(FETCHES)), NULL), NVL2(MAX(BUFFER_GETS), XMLELEMENT("stat", XMLATTRIBUTES('buffer_gets' AS "name"),
MAX(BUFFER_GETS)), NULL), NVL2(MAX(READ_REQS), XMLELEMENT("stat", XMLATTRIBUTES('disk_reads' AS "name"),
MAX(READ_REQS)), NULL), NVL2(MAX(WRITE_REQS), XMLELEMENT("stat", XMLATTRIBUTES('direct_writes' AS "name"),
MAX(WRITE_REQS)), NULL), NVL2(MAX(READ_REQS), XMLELEMENT("stat", XMLATTRIBUTES('read_reqs' AS "name"),
MAX(READ_REQS)), NULL), NVL2(MAX(READ_BYTES), XMLELEMENT("stat", XMLATTRIBUTES('read_bytes' AS "name"),
MAX(READ_BYTES)), NULL), NVL2(MAX(WRITE_REQS), XMLELEMENT("stat", XMLATTRIBUTES('write_reqs' AS "name"),
MAX(WRITE_REQS)), NULL), NVL2(MAX(WRITE_BYTES), XMLELEMENT("stat", XMLATTRIBUTES('write_bytes' AS "name"),
MAX(WRITE_BYTES)), NULL), CASE WHEN MAX(IO_INTER_BYTES) IS NULL OR NVL(MAX(IO_BYTES), 0) = 0 OR
MAX(IO_INTER_BYTES) = MAX(IO_BYTES) THEN NULL ELSE XMLELEMENT("stat", XMLATTRIBUTES('cell_offload_efficiency' AS
"name"), ROUND(MAX(IO_BYTES)/DECODE(MAX(IO_INTER_BYTES), 0, 1, MAX(IO_INTER_BYTES)), 2)) END), CASE WHEN
:B33 = 1 AND SUM(PI.ACTIVITY_COUNT) > 0 THEN XMLELEMENT("activity_sampled", XMLATTRIBUTES(
TO_CHAR(MIN(PI.ACTIVITY_START), :B12) AS "start_time", TO_CHAR(MAX(PI.ACTIVITY_END), :B12) AS "end_time",
ROUND((MAX(PI.ACTIVITY_END) - MIN(PI.ACTIVITY_START)) * 3600 * 24) + 1 AS "duration", SUM(PI.ACTIVITY_COUNT) AS
"count", SUM(PI.IMQ_COUNT) AS "imq_count", SUM(PI.WAIT_COUNT) AS "wait_count", SUM(PI.CPU_COUNT) AS "cpu_count",
SUM(PI.OTHER_SQL_COUNT) AS "other_sql_count", :B40 AS "cpu_cores", :B39 AS "hyperthread"), XMLAGG(
NVL2(ACTIVITY_TYPE, XMLELEMENT("activity", XMLATTRIBUTES( PI.ACTIVITY_TYPE AS "class", PI.EVENT_NAME AS "event"),
ACTIVITY_COUNT), NULL) ORDER BY PI.ACTIVITY_TYPE, PI.EVENT_NAME) ELSE NULL END) PX_INSTANCES_XML FROM
(SELECT MO.INST_ID, ASH.ACTIVITY_TYPE, ASH.EVENT_NAME, ASH.ACTIVITY_COUNT, ASH.IMQ_COUNT,
ASH.WAIT_COUNT, ASH.CPU_COUNT, ASH.OTHER_SQL_COUNT, ASH.ACTIVITY_START, ASH.ACTIVITY_END,
MO.ELAPSED_TIME, MO.QUEUING_TIME, MO.CPU_TIME, MO.APPLICATION_WAIT_TIME, MO.CONCURRENCY_WAIT_TIME,
MO.CLUSTER_WAIT_TIME, MO.USER_IO_WAIT_TIME, MO.PLSQL_EXEC_TIME, MO.JAVA_EXEC_TIME,

```

```

MO.OTHER_WAIT_TIME, MO.FETCHES, MO.BUFFER_GETS, MO.IO_INTER_BYTES, MO.IO_BYTES, MO.READ_REQS,
MO.READ_BYTES, MO.WRITE_REQS, MO.WRITE_BYTES, MO.IO_REQS FROM (SELECT MO0.INST_ID,
SUM(MO0.ELAPSED_TIME) ELAPSED_TIME, SUM(MO0.QUEUING_TIME) QUEUING_TIME, SUM(MO0.CPU_TIME) CPU_TIME,
SUM(MO0.FETCHES) FETCHES, SUM(MO0.BUFFER_GETS) BUFFER_GETS, SUM(MO0.IO_INTER_BYTES) IO_INTER_BYTES,
SUM(MO0.IO_BYTES) IO_BYTES, SUM(MO0.READ_REQS) READ_REQS, SUM(MO0.READ_BYTES) READ_BYTES,
SUM(MO0.WRITE_REQS) WRITE_REQS, SUM(MO0.WRITE_BYTES) WRITE_BYTES, SUM(MO0.IO_REQS) IO_REQS,
SUM(MO0.APPLICATION_WAIT_TIME) APPLICATION_WAIT_TIME, SUM(MO0.CONCURRENCY_WAIT_TIME)
CONCURRENCY_WAIT_TIME, SUM(MO0.CLUSTER_WAIT_TIME) CLUSTER_WAIT_TIME, SUM(MO0.USER_IO_WAIT_TIME)
USER_IO_WAIT_TIME, SUM(MO0.PLSQL_EXEC_TIME) PLSQL_EXEC_TIME, SUM(MO0.JAVA_EXEC_TIME) JAVA_EXEC_TIME,
SUM(MO0.OTHER_WAIT_TIME) OTHER_WAIT_TIME FROM MONITOR_DATA MO0 GROUP BY MO0.INST_ID) MO, (SELECT
ASH0.INST_ID, ASH0.ACTIVITY_TYPE, ASH0.EVENT_NAME, SUM(ASH0.ACTIVITY_COUNT) ACTIVITY_COUNT,
SUM(ASH0.IMQ_COUNT) IMQ_COUNT, SUM(ASH0.WAIT_COUNT) WAIT_COUNT, SUM(ASH0.CPU_COUNT) CPU_COUNT,
SUM(ASH0.OTHER_SQL_COUNT) OTHER_SQL_COUNT, MIN(ASH0.ACTIVITY_START) ACTIVITY_START,
MAX(ASH0.ACTIVITY_END) ACTIVITY_END FROM ASH_DATA ASH0 GROUP BY ASH0.INST_ID, ASH0.ACTIVITY_TYPE,
ASH0.EVENT_NAME) ASH, MONITOR_AGG WHERE MO.INST_ID = ASH.INST_ID(+) AND MONITOR_AGG.DIST_INST_COUNT >
0 ) PI GROUP BY PI.INST_ID)PX_INSTANCE) PX_INSTANCES FROM MONITOR_AGG)) ELSE NULL END, XPLAN_XML, CASE
WHEN :B43 = 1 THEN (SELECT XMLELEMENT( "plan_monitor", XMLATTRIBUTES(MAX(PLI.MAX_LINE_ACTIVITY_COUNT) AS
"max_activity_count ", MAX(PLI.OVERALL_MAX_IO_REQS) AS "max_io_reqs", MAX(PLI.OVERALL_MAX_IO_BYTES) AS
"max_io_bytes", MAX(PLI.MAX_LINE_IMQ_COUNT) AS "max_imq_count", MAX(PLI.MAX_LINE_CPU_COUNT) AS
"max_cpu_count", MAX(PLI.MAX_LINE_WAIT_COUNT) AS "max_wait_count", MAX(PLI.MAX_LINE_OTHER_SQL_COUNT) AS
"max_other_sql_count"), XMLAGG( XMLELEMENT( "operation", XMLATTRIBUTES( PLI.PLAN_LINE_ID AS "id", PLI.PARENT_ID AS
"parent_id", PLI.OPERATION AS "name", PLI.OPTIONS AS "options", PLI.DEPTH AS "depth", PLI.POSITION AS "position",
PLI.INACTIVE AS "skp", PLI.PX_TYPE AS "px_type"), NVL2(PLI.OBJECT_NAME, XMLELEMENT( "object",
XMLATTRIBUTES(PLI.OBJECT_TYPE AS "type"), XMLFOREST(PLI.OBJECT_OWNER AS "owner"),
XMLFOREST(PLI.OBJECT_NAME AS "name")), NULL), XMLFOREST(PLI.PARTITION_START AS "partition_start",
PLI.PARTITION_STOP AS "partition_stop"), CASE WHEN PLI.CARDINALITY IS NULL AND PLI.BYTES IS NULL AND PLI.COST IS
NULL AND PLI.TEMP_SPACE IS NULL AND PLI.TIME IS NULL THEN NULL ELSE XMLELEMENT( "optimizer", NULL,
NVL2(PLI.CARDINALITY, XMLFOREST(PLI.CARDINALITY AS "cardinality"), NULL), NVL2(PLI.BYTES, XMLFOREST(PLI.BYTES AS
"bytes"), NULL), NVL2(PLI.COST, XMLFOREST(PLI.COST AS "cost"), NULL), NVL2(PLI.CPU_COST, XMLFOREST(PLI.CPU_COST
AS "cpu_cost"), NULL), NVL2(PLI.IO_COST, XMLFOREST(PLI.IO_COST AS "io_cost"), NULL), NVL2(PLI.TEMP_SPACE,
XMLFOREST(PLI.TEMP_SPACE AS "temp"), NULL), NVL2(PLI.TIME, XMLFOREST(PLI.TIME AS "time"), NULL)) END,
XMLELEMENT( "stats", XMLATTRIBUTES('plan_monitor' AS "type"), NVL2(PLI.FIRST_MOVE_TIME, XMLELEMENT( "stat",
XMLATTRIBUTES('first_active' AS "name"), TO_CHAR(FIRST_MOVE_TIME, :B12 )), NULL), CASE WHEN PLI.FIRST_MOVE_TIME
!= PLI.FIRST_CHANGE_TIME THEN XMLELEMENT( "stat", XMLATTRIBUTES('first_row' AS "name"),
TO_CHAR(FIRST_CHANGE_TIME, :B12 )) ELSE NULL END, NVL2(PLI.LAST_MOVE_TIME, XMLELEMENT( "stat",
XMLATTRIBUTES('last_active' AS "name"), TO_CHAR(LAST_MOVE_TIME, :B12 )), NULL), CASE WHEN (PLI.FIRST_MOVE_TIME
IS NULL OR PLI.LAST_MOVE_TIME IS NULL) THEN NULL ELSE XMLELEMENT( "stat", XMLATTRIBUTES('duration' AS "name"),
ROUND((LAST_MOVE_TIME - FIRST_MOVE_TIME) * 3600 * 24)+1) END, CASE WHEN (PLI.OVERALL_LAST_MOVE_TIME IS
NULL OR PLI.LAST_MOVE_TIME IS NULL) THEN NULL ELSE XMLELEMENT( "stat", XMLATTRIBUTES('from_most_recent' AS
"name"), ROUND((PLI.OVERALL_LAST_MOVE_TIME - PLI.LAST_MOVE_TIME) * 3600 * 24) END, NVL2(PLI.LAST_MOVE_TIME,
XMLELEMENT( "stat", XMLATTRIBUTES( 'from_sql_exec_start' AS "name"), ROUND( (FIRST_MOVE_TIME - :B8 ) * 3600*24)), NU
LL), NVL2(PLI.PERCENT_COMPLETE, XMLELEMENT( "stat", XMLATTRIBUTES('percent_complete' AS "name"),
PLI.PERCENT_COMPLETE), NULL), NVL2(PLI.TIME_REMAINING, XMLELEMENT( "stat", XMLATTRIBUTES('time_left' AS "name"),
PLI.TIME_REMAINING), NULL), CASE WHEN PLI.HAS_EXECUTED = 1 THEN XMLELEMENT( "stat", XMLATTRIBUTES('starts' AS
"name"), PLI.STARTS) ELSE NULL END, CASE WHEN PLI.DOP > 0 AND PLI.HAS_EXECUTED = 1 THEN XMLELEMENT( "stat",
XMLATTRIBUTES( 'max_starts' AS "name", DECODE(:B67 , 'Y', MOD(TRUNC(MAX_STARTS/1000000), 10000), NULL) AS "iid",
MOD(MAX_STARTS, 1000000) AS "sid"), TRUNC(PLI.MAX_STARTS/10000000000)) ELSE NULL END, CASE WHEN PLI.DOP > 0
AND PLI.HAS_EXECUTED = 1 THEN XMLELEMENT( "stat", XMLATTRIBUTES('dop' AS "name"), PLI.DOP) ELSE NULL END, CASE
WHEN NEED_ROWS IS NOT NULL AND PLI.FIRST_MOVE_TIME IS NOT NULL THEN XMLELEMENT( "s stat",
XMLATTRIBUTES('cardinality' AS "name"), PLI.OUTPUT_ROWS) ELSE NULL END, CASE WHEN PLI.NEED_ROWS IS NOT NULL
AND PLI.DOP > 0 AND PLI.MAX_OUTPUT_ROWS IS NOT NULL AND (PLI.FIRST_MOVE_TIME IS NOT NULL) THEN
XMLELEMENT( "stat", XMLATTRIBUTES( 'max_card' AS "name", DECODE(:B67 , 'Y', MOD(TRUNC(MAX_OUTPUT_ROWS/
1000000), 10000), NULL) AS "iid", MOD(MAX_OUTPUT_ROWS, 1000000) AS "sid"), TRUNC(PLI.MAX_OUTPUT_ROWS/
10000000000)) ELSE NULL END, CASE WHEN PLI.MEM > 0 THEN XMLELEMENT( "stat", XMLATTRIBUTES('memory' AS "name"),
PLI.MEM) ELSE NULL END, CASE WHEN PLI.MAX_MEM > 0 THEN XMLELEMENT( "stat", XMLATTRIBUTES('max_memory' AS
"name"), PLI.MAX_MEM) ELSE NULL END, CASE WHEN PLI.DOP > 0 AND PLI.MIN_MAX_MEM IS NOT NULL AND PLI.MAX_MEM
> 0 THEN XMLELEMENT( "stat", XMLATTRIBUTES( 'min_max_mem' AS "name", DECODE(:B67 , 'Y',
MOD(TRUNC(MIN_MAX_MEM/10 00000), 10000), NULL) AS "iid", MOD(MIN_MAX_MEM, 1000000) AS "sid"),
TRUNC(PLI.MIN_MAX_MEM/10000000000)) ELSE NULL END, CASE WHEN PLI.TEMP > 0 THEN XMLELEMENT( "stat",
XMLATTRIBUTES('temp' AS "name"), PLI.TEMP) ELSE NULL END, CASE WHEN PLI.MAX_TEMP > 0 THEN XMLELEMENT( "stat",
XMLATTRIBUTES('max_temp' AS "name"), PLI.MAX_TEMP) ELSE NULL END, CASE WHEN PLI.MAX_TEMP > 0 THEN
XMLELEMENT( "stat", XMLATTRIBUTES('spill_count' AS "name"), PLI.SPILL_COUNT) ELSE NULL END, CASE WHEN PLI.DOP > 0
AND PLI.MAX_MAX_TEMP IS NOT NULL AND PLI.MAX_TEMP > 0 THEN XMLELEMENT( "stat", XMLATTRIBUTES(
'max_max_temp' AS "name", DECODE(:B67 , 'Y', MOD(TRUNC(MAX_MAX_TEMP/1000000), 10000), NULL) AS "iid",
MOD(MAX_MAX_TEMP, 1000000) AS "sid"), TRUNC(PLI.MAX_MAX_TEMP/10000000000)) ELSE NULL END, CASE WHEN
PLI.READ_REQS > 0 THEN XMLELEMENT( "stat", XMLATTRIBUTES('read_reqs' AS "name"), PLI.READ_REQS) ELSE NULL END,
CASE WHEN PLI.DOP > 0 AND PLI.MAX_READ_REQS IS NOT NULL AND PLI.READ_REQS > 0 THEN XMLELEMENT( "stat",
XMLATTRIBUTES( 'max_read_reqs' AS "name", DECODE(:B67 , 'Y', MOD(TRUNC(PLI.MAX_READ_REQS/1000000), 10000), NULL)
AS "iid", MOD(PLI.MAX_READ_REQS, 1000000) AS "sid"), TRUNC(PLI.MAX_READ_REQS/10000000000)) ELSE NULL END, CASE
WHEN PLI.READ_BYTES > 0 THEN XMLELEMENT( "stat", XMLATTRIBUTES('read_bytes' AS "name"), PLI.READ_BYTES) ELSE
NULL END, CASE WHEN PLI.DOP > 0 AND PLI.MAX_READ_BYTES IS NOT NULL AND PLI.READ_BYTES > 0 THEN
XMLELEMENT( "stat", XMLATTRIBUTES( 'max_read_bytes' AS "name", DECODE(:B67 , 'Y', MOD(TRUNC(PLI.MAX_READ_BYTES/
10000000), 10000), NULL) AS "iid", MOD(PLI.MAX_READ_BYTES, 1000000) AS "sid"), TRUNC(PLI.MAX_READ_BYTES/
10000000000)) ELSE NULL END, CASE WHEN PLI.WRITE_REQS > 0 THEN XMLELEMENT( "stat", XMLATTRIBUTES('write_reqs'
AS "name"), PLI.WRITE_REQS) ELSE NULL END, CASE WHEN PLI.DOP > 0 AND PLI.MAX_WRITE_REQS IS NOT NULL AND
PLI.WRITE_REQS > 0 THEN XMLELEMENT( "stat", XMLATTRIBUTES( 'max_write_reqs' AS "name", DECODE(:B67 , 'Y',
MOD(TRUNC(PLI.MAX_WRITE_REQS/1000000), 10000), NULL) AS "iid", MOD(PLI.MAX_WRITE_REQS, 1000000) AS "sid"),
TRUNC(PLI.MAX_WRITE_REQS/10000000000)) ELSE NULL END, CASE WHEN PLI.WRITE_BYTES > 0 THEN XMLELEMENT(
"stat", XMLATTRIBUTES('write_bytes' AS "name"), PLI.WRITE_BYTES) ELSE NULL END, CASE WHEN PLI.DOP > 0 AND
PLI.MAX_WRITE_BYTES IS NOT NULL AND PLI.WRITE_BYTES > 0 THEN XMLELEMENT( "stat", XMLATTRIBUTES(
'max_write_bytes' AS "name", DECODE(:B67 , 'Y', MOD(TRUNC(PLI.MAX_WRITE_BYTES/1000000), 10000), NULL) AS "iid",

```

```

MOD(PLI.MAX_WRITE_BYTES, 1000000) AS "sid"), TRUNC(PLI.MAX_WRITE_BYTES/10000000000) ELSE NULL END, CASE
WHEN PLI.IO_INTER_BYTES IS NOT NULL AND PLI.IO_BYTES > 0 AND PLI.IO_BYTES != PLI.IO_INTER_BYTES THEN
XMLELEMENT("stat", XMLATTRIBUTES('io_inter_bytes' AS "name"), PLI.IO_INTER_BYTES) ELSE NULL END, CASE WHEN
PLI.DOP > 0 AND PLI.MAX_IO_INTER_BYTES IS NOT NULL AND PLI.IO_BYTES > 0 AND PLI.IO_BYTES != PLI.IO_INTER_BYTES
THEN XMLELEMENT("stat", XMLATTRIBUTES('max_io_inter_bytes' AS "name", DECODE(:B67, 'Y',
MOD(TRUNC(MAX_IO_INTER_BYTES/1000000), 10000), NULL) AS "iid", MOD(MAX_IO_INTER_BYTES, 1000000) AS "sid"),
TRUNC(PLI.MAX_IO_INTER_BYTES/10000000000) ELSE NULL END, CASE WHEN PLI.IO_INTER_BYTES IS NOT NULL AND
PLI.IO_BYTES > 0 AND PLI.IO_BYTES != PLI.IO_INTER_BYTES THEN XMLELEMENT("stat", XMLATTRIBUTES('cell_oload_
efficiency' AS "name"), ROUND(PLI.IO_BYTES / DECODE(PLI.IO_INTER_BYTES, 0, 1, PLI.IO_INTER_BYTES), 2)) ELSE NULL
END), NVL2(STAT_GID, XMLELEMENT("rwsstats", XMLATTRIBUTES(PLI.STAT_GID AS "group_id"), DECODE(GID_ROWNUM, 1,
(SELECT XMLELEMENT("metadata", NULL, XMLAGG(XMLELEMENT("stat", XMLATTRIBUTES(ROWNUM AS "id", NAME AS
"name", DESCRIPTION AS "desc", TYPE AS "type", DECODE(FLAGS, 0, NULL, FLAGS) AS "flags"), NULL) ORDER BY ID)) FROM
V$SQL_MONITOR_STATNAME WHERE GROUP_ID = PLI.STAT_GID), NULL), NVL2(STAT1_VALUE, XMLELEMENT("stat",
XMLATTRIBUTES(1 AS "id"), STAT1_VALUE), NULL), NVL2(STAT2_VALUE, XMLELEMENT("stat", XMLATTRIBUTES(2 AS "id"),
STAT2_VALUE), NULL), NVL2(STAT3_VALUE, XMLELEMENT("stat", XMLATTRIBUTES(3 AS "id"), STAT3_VALUE), NULL),
NVL2(STAT4_VALUE, XMLELEMENT("stat", XMLATTRIBUTES(4 AS "id"), STAT4_VALUE), NULL), NVL2(STAT5_VALUE, XMLELE
MENT("stat", XMLATTRIBUTES(5 AS "id"), STAT5_VALUE), NULL), NVL2(STAT6_VALUE, XMLELEMENT("stat",
XMLATTRIBUTES(6 AS "id"), STAT6_VALUE), NULL), NVL2(STAT7_VALUE, XMLELEMENT("stat", XMLATTRIBUTES(7 AS "id"),
STAT7_VALUE), NULL), NVL2(STAT8_VALUE, XMLELEMENT("stat", XMLATTRIBUTES(8 AS "id"), STAT8_VALUE), NULL),
NVL2(STAT9_VALUE, XMLELEMENT("stat", XMLATTRIBUTES(9 AS "id"), STAT9_VALUE), NULL), NVL2(STAT10_VALUE,
XMLELEMENT("stat", XMLATTRIBUTES(10 AS "id"), STAT10_VALUE), NULL)), CASE WHEN PLI.LINE_ACTIVITY_COUNT >
0 AND :B33 = 1 THEN XMLELEMENT("activity_sampled", XMLATTRIBUTES( TO_CHAR(PLI.LINE_ACTIVITY_START, :B12 ) AS
"start_time", TO_CHAR(PLI.LINE_ACTIVITY_END, :B12 ) AS "end_time", ROUND((PLI.LINE_ACTIVITY_END -
PLI.LINE_ACTIVITY_START) * 3600*24) + 1 AS "duration", PLI.LINE_ACTIVITY_COUNT AS "count", PLI.LINE_IMQ_COUNT AS
"imq_count", PLI.LINE_WAIT_COUNT AS "wait_count", PLI.LINE_CPU_COUNT AS "cpu_count", PLI.LINE_OTHER_SQL_COUNT AS
"other_sql_count", :B40 AS "cpu_cores", :B39 AS "hyperthread"), PLI.ACTIVITY_TOTAL, PLI.PLAN_ACTIVITY_HISTO) ELSE NULL
END) ORDER BY PLI.PLAN_LINE_ID ) FROM (SELECT AT.MAX_LINE_ACTIVITY_COUNT, PM.OVERALL_MAX_IO_REQS,
PM.OVERALL_MAX_IO_BYTES, AT.MAX_LINE_IMQ_COUNT, AT.MAX_LINE_CPU_COUNT, AT.MAX_LINE_WAIT_COUNT,
AT.LINE_OTHER_SQL_COUNT, AT.MAX_LINE_OTHER_SQL_COUNT, PM.PLAN_LINE_ID, PM.PARENT_ID, PM.OPERATION,
PM.OPTIONS, PM.DEPTH, PM.POSITION, PM.INACTIVE, CASE WHEN PM.OPERATION = 'PX COORDINATOR' AND :B23 = 'Y'
AND (PM.PX_SERVER_SET IS NOT NULL OR AH.PX_SERVER_SET IS NOT NULL) THEN 'QC' WHEN PM.PX_SERVER_SET IS
NOT NULL THEN TO_CHAR(PM.PX_SERVER_SET) WHEN AH.PX_SERVER_SET IS NOT NULL THEN
TO_CHAR(AH.PX_SERVER_SET) WHEN (:B23 = 'N' OR (PM.LAST_CHANGE_TIME IS NULL AND AT.LINE_ACTIVITY_END IS
NULL)) THEN NULL ELSE 'QC' END PX_TYPE, PM.FIRST_CHANGE_TIME, AT.LINE_ACTIVITY_START, PM.LAST_CHANGE_TIME,
PM.OVERALL_LAST_CHANGE_TIME, AT.LINE_ACTIVITY_END, AT.OVERALL_LINE_ACTIVITY_END,
LEAST(NVL(AT.LINE_ACTIVITY_START, PM.FIRST_CHANGE_TIME), NVL(PM.FIRST_CHANGE_TIME,
AT.LINE_ACTIVITY_START)) FIRST_MOVE_TIME, GREATEST(NVL(AT.LINE_ACTIVITY_END, PM.LAST_CHANGE_TIME),
NVL(PM.LAST_CHANGE_TIME, AT.LINE_ACTIVITY_END)) LAST_MOVE_TIME, GREATEST(NVL(AT.OVERALL_LINE_
ACTIVITY_END, PM.OVERALL_LAST_CHANGE_TIME), NVL(PM.OVERALL_LAST_CHANGE_TIME,
AT.OVERALL_LINE_ACTIVITY_END)) OVERALL_LAST_MOVE_TIME, CASE WHEN PM.STARTS IS NOT NULL AND PM.STARTS >
0 THEN 1 ELSE 0 END HAS_EXECUTED, PM.OBJECT_NAME, PM.OBJECT_TYPE, PM.OBJECT_OWNER,
PM.PARTITION_START, PM.PARTITION_STOP, PM.CARDINALITY, PM.BYTES, PM.COST, PM.TEMP_SPACE, PM.TIME,
PM.CPU_COST, PM.IO_COST, LO.PERCENT_COMPLETE, LO.TIME_REMAINING, PM.STARTS, PM.DOP, PM.MAX_STARTS,
PM.OUTPUT_ROWS, PM.NEED_ROWS, PM.MAX_OUTPUT_ROWS, PM.MEM, PM.MAX_MEM, PM.MIN_MAX_MEM, PM.TEMP,
PM.MAX_TEMP, PM.SPILL_COUNT, PM.MAX_MAX_TEMP, PM.READ_REQS, PM.MAX_READ_REQS, PM.READ_BYTES,
PM.MAX_READ_BYTES, PM.WRITE_REQS, PM.MAX_WRITE_REQS, PM.WRITE_BYTES, PM.MAX_WRITE_BYTES,
PM.IO_INTER_BYTES, PM.IO_BYTES, PM.MAX_IO_INTER_BYTES, AT.LINE_ACTIVITY_COUNT, AT.LINE_IMQ_COUNT,
AT.LINE_WAIT_COUNT, AT.LINE_CPU_COUNT, AT.ACTIVITY_TOTAL, AH.PLAN_ACTIVITY_HISTO, PM.STAT_GID,
PM.GID_ROWNUM, PM.STAT1_VALUE, PM.STAT2_VALUE, PM.STAT3_VALUE, PM.STAT4_VALUE, PM.STAT5_VALUE,
PM.STAT6_VALUE, PM.STAT7_VALUE, PM.STAT8_VALUE, PM.STAT9_VALUE, PM.STAT10_VALUE FROM (SELECT AT0.*,
MAX(LINE_ACTIVITY_END) OVER() OVERALL_LINE_ACTIVITY_END, MAX(LINE_ACTIVITY_COUNT) OVER()
MAX_LINE_ACTIVITY_COUNT, MAX(LINE_IMQ_COUNT) OVER() MAX_LINE_IMQ_COUNT, MAX(LINE_CPU_COUNT) OVER()
MAX_LINE_CPU_COUNT, MAX(LINE_WAIT_COUNT) OVER() MAX_LINE_WAIT_COUNT, MAX(LINE_OTHER_SQL_COUNT)
OVER() MAX_LINE_OTHER_SQL_COUNT FROM (SELECT AD1.PLAN_LINE_ID, MIN(AD1.LINE_ACTIVITY_START)
LINE_ACTIVITY_START, MAX(AD1.LINE_ACTIVITY_END) LINE_ACTIVITY_END, SUM(AD1.ACTIVITY_COUNT)
LINE_ACTIVITY_COUNT, SUM(AD1.IMQ_COUNT) LINE_IMQ_COUNT, SUM(AD1.WAIT_COUNT) LINE_WAIT_COUNT,
SUM(AD1.OTHER_SQL_COUNT) LINE_OTHER_SQL_COUNT, SUM(AD1.CPU_COUNT) LINE_CPU_COUNT,
SUBSTR(MAX(LPAD(AD1.ACTIVITY_COUNT, 10)|| AD1.ACTIVITY_TYPE), 11) MOST_ACTIVE, XMLAGG(XMLELEMENT("activity",
XMLATTRIBUTES( AD1.ACTIVITY_TYPE AS "class", AD1.EVENT_NAME AS "event"), AD1.ACTIVITY_COUNT) ORDER BY
AD1.ACTIVITY_TYPE, AD1.EVENT_NAME) ACTIVITY_TOTAL FROM (SELECT AD0.PLAN_LINE_ID, AD0.ACTIVITY_TYPE,
AD0.EVENT_NAME, MIN(AD0.ACTIVITY_START) LINE_ACTIVITY_START, MAX(AD0.ACTIVITY_END) LINE_ACTIVITY_END,
SUM(AD0.ACTIVITY_COUNT) ACTIVITY_COUNT, SUM(AD0.IMQ_COUNT) IMQ_COUNT, SUM(AD0.WAIT_COUNT) WAIT_COUNT,
SUM(AD0.CPU_COUNT) CPU_COUNT, SUM(AD0.OTHER_SQL_COUNT) OTHER_SQL_COUNT,
MAX(AD0.DFO_MOST_ACTIVE_IID) DFO_MOST_ACTIVE_IID, MAX(AD0.DFO_MOST_ACTIVE_SID) DFO_MOST_ACTIVE_SID,
SUM(AD0.DFO_MOST_ACTIVE_COUNT) DFO_MOST_ACTIVE_COUNT FROM ASH DATA AD0 WHERE AD0.PLAN_LINE_ID IS
NOT NULL GROUP BY AD0.PLAN_LINE_ID, AD0.ACTIVITY_TYPE, AD0.EVENT_NAME) AD1 GROUP BY AD1.PLAN_LINE_ID)
AT0 AT, (SELECT AD2.PLAN_LINE_ID, MIN(AD2.PX_SERVER_SET) PX_SERVER_SET, MIN(AD2.LINE_ACTIVITY_START)
LINE_ACTIVITY_START, MAX(AD2.LINE_ACTIVITY_END) LINE_ACTIVITY_END, SUM(AD2.ACTIVITY_COUNT)
ACTIVITY_COUNT, SUM(AD2.IMQ_COUNT) IMQ_COUNT, SUM(AD2.WAIT_COUNT) WAIT_COUNT, SUM(AD2.CPU_COUNT)
CPU_COUNT, SUM(AD2.OTHER_SQL_COUNT) OTHER_SQL_COUNT, CASE WHEN :B29 = 1 AND SUM(AD2.ACTIVITY_COUNT)
> 0 THEN XMLELEMENT("activity_histogram", XMLATTRIBUTES(:B26 AS "bucket_interval", :B27 AS "bucket_count", TO_CHAR(
:B20, :B12 ) AS "start_time", TO_CHAR(:B22, :B12 ) AS "end_time", ROUND((:B22 - :B20) * 3600*24) + 1 AS "duration"),
XMLAGG(XMLELEMENT("bucket", XMLATTRIBUTES(AD2.BUCKET_NUM AS "number"), AD2.ACTIVITY_BUCKET) ORDER BY
AD2.BUCKET_NUM)) ELSE NULL END PLAN_ACTIVITY_HISTO FROM (SELECT AD1.PLAN_LINE_ID, AD1.BUCKET_NUM,
MIN(AD1.PX_SERVER_SET) PX_SERVER_SET, MIN(AD1.LINE_ACTIVITY_START) LINE_ACTIVITY_START,
MAX(AD1.LINE_ACTIVITY_END) LINE_ACTIVITY_END, MIN(AD1.BUCKET_ACTIVITY_START) BUCKET_ACTIVITY_START,
MAX(AD1.BUCKET_ACTIVITY_END) BUCKET_ACTIVITY_END, SUM(AD1.ACTIVITY_COUNT) ACTIVITY_COUNT,
SUM(AD1.IMQ_COUNT) IMQ_COUNT, SUM(AD1.WAIT_COUNT) WAIT_COUNT, SUM(AD1.CPU_COUNT) CPU_COUNT,
SUM(AD1.OTHER_SQL_COUNT) OTHER_SQL_COUNT, SUBSTR(MAX(LPAD(AD1.ACTIVITY_COUNT, 10)|| AD1.ACTIVITY_TYPE),
11) MOST_ACTIVE, XMLAGG(NVL2(AD1.ACTIVITY_TYPE, XMLELEMENT("activity", XMLATTRIBUTES( AD1.ACTIVITY_TYPE AS
"class", AD1.EVENT_NAME AS "event"), AD1.ACTIVITY_COUNT), NULL) ORDER BY AD1.ACTIVITY_TYPE, AD1.EVENT_NAME)

```

```

ACTIVITY_BUC KET FROM (SELECT AD0.PLAN_LINE_ID, AD0.PLAN_ACTIVITY_BUCKET_NUM BUCKET_NUM,
AD0.ACTIVITY_TYPE, AD0.EVENT_NAME, MIN(NVL2(AD0.ACTIVITY_START, AD0.PX_SERVER_SET, NULL)) PX_SERVER_SET,
MIN(AD0.ACTIVITY_START) LINE_ACTIVITY_START, MAX(AD0.ACTIVITY_END) LINE_ACTIVITY_END,
MIN(AD0.BUCKET_ACTIVITY_START) BUCKET_ACTIVITY_START, MAX(AD0.BUCKET_ACTIVITY_END)
BUCKET_ACTIVITY_END, SUM(AD0.IMQ_COUNT) IMQ_COUNT, SUM(AD0.CPU_COUNT) CPU_COUNT,
SUM(AD0.WAIT_COUNT) WAIT_COUNT, SUM(AD0.OTHER_SQL_COUNT) OTHER_SQL_COUNT, SUM(AD0.ACTIVITY_COUNT)
ACTIVITY_COUNT FROM ASH_DATA AD0 GROUP BY AD0.PLAN_LINE_ID, AD0.PLAN_ACTIVITY_BUCKET_NUM,
AD0.ACTIVITY_TYPE, AD0.EVENT_NAME) AD1 GROUP BY AD1.PLAN_LINE_ID, AD1.BUCKET_NUM) AD2 GROUP BY
AD2.PLAN_LINE_ID) AH, (SELECT LO.SQL_PLAN_LINE_ID PLAN_LINE_ID, DECODE(SUM(LO.TOTALWORK), 0, NULL,
ROUND(SUM(LO.SOFAR)*100/SUM(LO.TOTALWORK))) PERCENT_COMPLETE, MAX(LO.TIME_REMAINING) TIME_REMAINING
FROM GV$SESSION_LONGOPS LO, MONITOR_DATA MO WHERE (:B49 = 'Y' OR :B70 = 'DONE (ERROR)') AND LO.SQL_ID = :B9
AND LO.SQL_EXEC_START = :B8 AND LO.SQL_EXEC_ID = :B7 AND LO.INST_ID = MO.INST_ID AND LO.SID = MO.SID GROUP
BY LO.SQL_PLAN_LINE_ID) LO, (SELECT PM0.*, CASE WHEN PM0.STARTS IS NULL OR PM0.STARTS = 0 OR
PM0.OUTPUT_ROWS IS NULL THEN NULL ELSE 1 END NEED_ROWS, ROW_NUMBER() OVER(PARTITION BY PM0.STAT_GID
ORDER BY PM0.PLAN_LINE_ID ) GID_ROWNUM, MAX(LAST_CHANGE_TIME) OVER() OVERALL_LAST_CHANGE_TIME,
MAX(MAX_IO_REQS) OVER() OVERALL_MAX_IO_REQS, MAX(MAX_IO_BYTES) OVER() OVERALL_MAX_IO_BYTES FROM
(SELECT /*+ leading(mo) use_hash(plm) */ PLM.PLAN_LINE_ID PLAN_LINE_ID, PLM.PLAN_OPERATION OPERATION,
PLM.PLAN_OPTIONS OPTIONS, MAX(PLM.PLAN_PARENT_ID) PARENT_ID, MAX(PLM.PLAN_DEPTH) DEPTH,
MAX(PLM.PLAN_POSITION) POSITION, MAX(PLM.PLAN_OPERATION_INACTIVE) INACTIVE,
MAX(PLM.PLAN_OBJECT_OWNER) OBJECT_OWNER, MAX(PLM.PLAN_OBJECT_NAME) OBJECT_NAME,
MAX(PLM.PLAN_OBJECT_TYPE) OBJECT_TYPE, MAX(PLM.PLAN_COST) COST, MAX(PLM.PLAN_CARDINALITY) CARDINALI
TY, MAX(PLM.PLAN_BYTES) BYTES, MAX(PLM.PLAN_CPU_COST) CPU_COST, MAX(PLM.PLAN_IO_COST) IO_COST,
MAX(PLM.PLAN_TEMP_SPACE) TEMP_SPACE, MAX(PLM.PLAN_TIME) TIME, MAX(PLM.PLAN_PARTITION_START)
PARTITION_START, MAX(PLM.PLAN_PARTITION_STOP) PARTITION_STOP, MIN(PLM.FIRST_CHANGE_TIME)
FIRST_CHANGE_TIME, MAX(PLM.LAST_CHANGE_TIME) LAST_CHANGE_TIME, MIN(PLM.LAST_CHANGE_TIME)
MIN_LAST_CHANGE_TIME, MIN(NVL2(PLM.FIRST_CHANGE_TIME, MO.PX_SERVER_SET, NULL)) PX_SERVER_SET,
COUNT(CASE WHEN PLM.PAR IS NOT NULL AND PLM.STARTS IS NOT NULL AND PLM.STARTS > 0 AND PLM.PLAN_LINE_ID !=
0 AND PLM.PLAN_OPERATION != 'PX COORDINATOR' THEN 1 ELSE NULL END) DOP, SUM(PLM.STARTS) STARTS,
MAX(NVL2(PAR, PLM.STARTS * 10000000000 + PLM.ES, NULL)) MAX_STARTS, SUM(PLM.OUTPUT_ROWS) OUTPUT_ROWS,
MAX(NVL2(PAR, PLM.OUTPUT_ROWS * 10000000000 + PLM.ES, NULL)) MAX_OUTPUT_ROWS, SUM(PLM.WORKAREA_MEM)
MEM, SUM(PLM.WORKAREA_MAX_MEM) MAX_MEM, MIN(NVL2(PAR, PLM.WORKAREA_MAX_MEM * 10000000000 + PLM.ES,
NULL)) MIN_MAX_MEM, SUM(PLM.WORKAREA_TEMPSEG) TEMP, SUM(PLM.WORKAREA_MAX_TEMPSEG) MAX_TEMP,
MAX(NVL2(PAR, PLM.WORKAREA_MAX_TEMPSEG * 10000000000 + PLM.ES, NULL)) MAX_MAX_TEMP,
COUNT(PLM.WORKAREA_MAX_TEMPSEG) SPILL_COUNT, SUM(PLM.PHYSICAL_READ_REQUESTS) READ_REQS,
MAX(NVL2(PAR, PLM.PHYSICAL_READ_REQUESTS * 10000000000 + PLM.ES, NULL)) MAX_READ_REQS,
SUM(PLM.PHYSICAL_READ_BYTES) READ_BYTES, MAX(NVL2(PAR, PLM.PHYSICAL_READ_BYTES * 10000000000 + PLM.ES,
NULL)) MAX_READ_BYTES, SUM(PLM.PHYSICAL_WRITE_REQUESTS) WRITE_REQS, MAX(NVL2(PAR,
PLM.PHYSICAL_WRITE_REQUESTS * 10000000000 + PLM.ES, NULL)) MAX_WRITE_REQS,
SUM(PLM.PHYSICAL_WRITE_BYTES) WRITE_BYTES, MAX(NVL2(PAR, PLM.PHYSICAL_WRITE_BYTES * 10000000000 +
PLM.ES, NULL)) MAX_WRITE_BYTES, NVL(SUM(PLM.PHYSICAL_READ_BYTES), 0) + NVL(SUM(PLM.PHYSICAL_WRITE_
BYTES), 0) IO_BYTES, SUM(NVL(PLM.PHYSICAL_READ_REQUESTS, 0) + NVL(PLM.PHYSICAL_WRITE_REQUESTS, 0))
MAX_IO_REQS, SUM(NVL(PLM.PHYSICAL_READ_BYTES, 0) + NVL(PLM.PHYSICAL_WRITE_BYTES, 0)) MAX_IO_BYTES,
SUM(PLM.IO_INTERCONNECT_BYTES) IO_INTER_BYTES, MAX(NVL2(PAR, PLM.IO_INTERCONNECT_BYTES * 10000000000 +
PLM.ES, NULL)) MAX_IO_INTER_BYTES, MAX(OTHERSTAT_GROUP_ID) STAT_GID, NVL(DECODE(MAX(OTHERSTAT_1_TYPE),
3, MAX(OTHERSTAT_1_VALUE), 4, MAX(OTHERSTAT_1_VALUE), 6, MIN(OTHERSTAT_1_VALUE), 7,
MIN(OTHERSTAT_1_VALUE), NULL), SUM(DECODE(OTHERSTAT_1_TYPE, 1, OTHERSTAT_1_VALUE, 2, OTHERSTAT_1_VALUE,
NULL))) STAT1_VALUE, NVL(MOD((DECODE(MAX(OTHERSTAT_2_TYPE), 3, MAX(OTHERSTAT_2_VALUE), 5,
DECODE(MAX(OTHERSTAT_1_TYPE), 4, MAX(RMAX1 + OTHERSTAT_2_VALUE), 7, MIN(RMAX1 + OTHERSTAT_2_VALUE),
NULL), 6, MIN(OTHERSTAT_2_VALUE), NULL)), 1000000000000000), SUM(DECODE(OTHERSTAT_2_TYPE, 1,
OTHERSTAT_2_VALUE, 2, OTHERSTAT_2_VALUE, NULL))) STAT2_VALUE, NVL(MOD((DECODE(MAX(OTHERSTAT_3_TYPE), 3,
MAX(OTHERSTAT_3_VALUE), 5, DECODE(MAX(OTHERSTAT_1_TYPE), 4, MAX(RMAX1 + OTHERSTAT_3_VALUE), 7,
MIN(RMAX1 + OTHERSTAT_3_VALUE), NULL), 6, MIN(OTHERSTAT_3_VALUE), NULL)), 1000000000000000),
SUM(DECODE(OTHERSTAT_3_TYPE, 1, OTHERSTAT_3_VALUE, 2, OTHERSTAT_3_VALUE, NULL))) STAT3_VALUE,
NVL(MOD((DECODE(MAX(OTHERSTAT_4_TYPE), 3, MAX(OTHERSTAT_4_VALUE), 5, DECODE(MAX(OTHERSTAT_1_TYPE), 4,
MAX(RMAX1 + OTHERSTAT_4_VALUE), 7, MIN(RMAX1 + OTHERSTAT_4_VALUE), NULL), 6, MIN(OTHERSTAT_4_VALUE),
NULL)), 1000000000000000), SUM(DECODE(OTHERSTAT_4_TYPE, 1, OTHERSTAT_4_VALUE, 2, OTHERSTAT_4_VALUE,
NULL))) STAT4_VALUE, NVL(MOD((DECODE(MAX(OTHERSTAT_5_TYPE), 3, MAX(OTHERSTAT_5_VALUE), 5,
DECODE(MAX(OTHERSTAT_1_TYPE), 4, MAX(RMAX1 + OTHERSTAT_5_VALUE), 7, MIN(RMAX1 + OTHERSTAT_5_VALUE),
NULL), 6, MIN(OTHERSTAT_5_VALUE), NULL)), 1000000000000000), SUM(DECODE(OTHERSTAT_5_TYPE, 1,
OTHERSTAT_5_VALUE, 2, OTHERSTAT_5_VALUE, NULL))) STAT5_VALUE, NVL(MOD((DECODE(MAX(OTHERSTAT_6_TYPE), 3,
MAX(OTHERSTAT_6_VALUE), 5, DECODE(MAX(OTHERSTAT_1_TYPE), 4, MAX(RMAX1 + OTHERSTAT_6_VALUE), 7,
MIN(RMAX1 + OTHERSTAT_6_VALUE), NULL), 6, MIN(OTHERSTAT_6_VALUE), NULL)), 1000000000000000),
SUM(DECODE(OTHERSTAT_6_TYPE, 1, OTHERSTAT_6_VALUE, 2, OTHERSTAT_6_VALUE, NULL))) STAT6_VALUE,
NVL(MOD((DECODE(MAX(OTHERSTAT_7_TYPE), 3, MAX(OTHERSTAT_7_VALUE), 5, DECODE(MAX(OTHERSTAT_1_TYPE), 4,
MAX(RMAX1 + OTHERSTAT_7_VALUE), 7, MIN(RMAX1 + OTHERSTAT_7_VALUE), NULL), 6, MIN(OTHERSTAT_7_VALUE),
NULL)), 1000000000000000), SUM(DECODE(OTHERSTAT_7_TYPE, 1, OTHERSTAT_7_VALUE, 2, OTHERSTAT_7_VALUE,
NULL))) STAT7_VALUE, NVL(MOD((DECODE(MAX(OTHERSTAT_8_TYPE), 3, MAX(OTHERSTAT_8_VALUE), 5,
DECODE(MAX(OTHERSTAT_1_TYPE), 4, MAX(RMAX1 + OTHERSTAT_8_VALUE), 7, MIN(RMAX1 + OTHERSTAT_8_VALUE),
NULL), 6, MIN(OTHERSTAT_8_VALUE), NULL)), 1000000000000000), SUM(DECODE(OTHERSTAT_8_TYPE, 1,
OTHERSTAT_8_VALUE, 2, OTHERSTAT_8_VALUE, NULL))) STAT8_VALUE, NVL(MOD((DECODE(MAX(OTHERSTAT_9_TYPE), 3,
MAX(OTHERSTAT_9_VALUE), 5, DECODE(MAX(OTHERSTAT_1_TYPE), 4, MAX(RMAX1 + OTHERSTAT_9_VALUE), 7,
MIN(RMAX1 + OTHERSTAT_9_VALUE), NULL), 6, MIN(OTHERSTAT_9_VALUE), NULL)), 1000000000000000),
SUM(DECODE(OTHERSTAT_9_TYPE, 1, OTHERSTAT_9_VALUE, 2, OTHERSTAT_9_VALUE, NULL))) STAT9_VALUE,
NVL(MOD((DECODE(MAX(OTHERSTAT_10_TYPE), 3, MAX(OTHERSTAT_10_VALUE), 5, DECODE(MAX(OTHERSTAT_1_TYPE), 4,
MAX(RMAX1 + OTHERSTAT_10_VALUE), 7, MIN(RMAX1 + OTHERSTAT_10_VALUE), NULL), 6, MIN(OTHERSTAT_10_VALUE),
NULL)), 1000000000000000), SUM(DECODE(OTHERSTAT_10_TYPE, 1, OTHERSTAT_10_VALUE, 2, OTHERSTAT_10_VALUE,
NULL))) STAT10_VALUE FROM (SELECT A.*, OTHERSTAT_1_VALUE * 1000000000000000 RMAX1, A.INST_ID * 1000000 + A.SID
ES, DECODE(A.PROCESS_NAME, 'ora', NULL, 1) PAR FROM GV$$SQL_PLAN_MONITOR A) PLM, MONITOR_DATA MO WHERE
PLM.SQL_ID = :B9 AND PLM.SQL_EXEC_START = :B8 AND PLM.SQL_EXEC_ID = :B7 AND PLM.INST_ID = MO.INST_ID AND
PLM.INST_ID BETWEEN :B11 AND :B10 AND PLM.KEY = MO.KEY GROUP BY PLM.PLAN_LINE_ID, PLM.PLAN_OPERATION,
PLM.PLAN_OPTION S) PM0) PM WHERE AH.PLAN_LINE_ID(+) = PM.PLAN_LINE_ID AND AT.PLAN_LINE_ID(+) =

```

```

PM.PLAN_LINE_ID AND LO.PLAN_LINE_ID(+) = PM.PLAN_LINE_ID AND (:B21 IS NULL OR PM.PLAN_LINE_ID = :B21 )) PLI)
ELSE NULL END, CASE WHEN :B42 = 1 THEN DBMS_SQLTUNE.BUILD_STASH_XML( SESSION_ID=>:B19 ,
SESSION_SERIAL=>:B18 , SESSION_INST_ID=>:B17 , PX_MODE=>'yes', START_TIME=>:B20 , END_TIME=>:B22 ,
MISSING_SECONDS=> :B41 , INSTANCE_LOW_FILTER=>:B11 , INSTANCE_HIGH_FILTER=>:B10 ,
BUCKET_MAX_COUNT=>NULL, BUCKET_INTERVAL=>:B26 , REPORT_LEVEL=>'typical', CPU_CORES=>:B40 , IS_HYPER=>:B39
) ELSE NULL END), (CASE WHEN :B38 = 1 THEN (SELECT XMLELEMENT( "skewed_sessions", XMLATTRIBUTES(
DECODE(MIN(INST_ID), NULL, :B17 , MIN(INST_ID)) AS "min_iid", DECODE(MAX(INST_ID), NULL, :B17 , MAX(INST_ID)) AS
"max_iid"), XMLAGG( XMLELEMENT( "s", XMLATTRIBUTES( INST_ID AS "i", SESSION_ID) ORDER BY INST_ID, SESSION_ID))
FROM (SELECT SESS.* FROM (SELECT DECODE(AD.DFO_MOST_ACTIVE_IID, :B17 , NULL, AD.DFO_MOST_ACTIVE_IID)
INST_ID, AD.DFO_MOST_ACTIVE_SID SESSION_ID FROM ASH_DATA AD WHERE AD.DFO_MOST_ACTIVE_COUNT IS NOT
NULL GROUP BY AD.DFO_MOST_ACTIVE_IID, AD.DFO_MOST_ACTIVE_SID ORDER BY MAX(AD.DFO_MOST_ACTIVE_COUNT)
DESC) SESS WHERE ROWNUM <= 100)) ELSE NULL END), (CASE WHEN :B37 = 1 THEN (SELECT XMLELEMENT(
"report_repository_summary", CASE WHEN MA.MAX_DBOP_NAME IS NOT NULL THEN XMLELEMENT( "dbop",
XMLATTRIBUTES('F AS "detail", MA.MAX_DBOP_NAME AS "dbop_name", TO_CHAR(MA.MIN_FIRST_REFRESH_TIME, :B12 ) AS
"dbop_exec_start", MA.MAX_DBOP_EXEC_ID AS "dbop_exec_id"), XMLFOREST( MA.MAX_STATUS AS "status",
TO_CHAR(NVL(MA.MAX_SQL_EXEC_START, MA.MIN_FIRST_REFRESH_TIME), :B12 )AS "first_refresh_time",
TO_CHAR(MA.MAX_LAST_REFRESH_TIME, :B12 )AS "last_refresh_time", MA.SUM_REFRESH_COUNT AS "refresh_count",
MA.MAX_INST_ID AS "inst_id", MA.MAX_SESSION_ID AS "session_id", MA.MAX_SESSION_SERIAL AS "session_serial",
MA.MAX_USERID AS "user_id", MA.MAX_USERNAME AS "user", MA.MAX_CON_ID AS "con_id", MA.MAX_CON_NAME AS
"con_name", MA.MAX_MODULE AS "module", MA.MAX_ACTION AS "action", MA.MAX_SERVICE_NAME AS "service",
MA.MAX_CLIENT_ID AS "client_id", MA.MAX_CLIENT_INFO AS "client_info", MA.MAX_PROGRAM AS "program",
MA.MAX_PL_ENTRY_OID AS "plsql_entry_object_id", MA.MAX_PL_ENTRY_PROGID AS "plsql_entry_subprogram_id",
MA.MAX_PL_OID AS "plsql_object_id", MA.MAX_PL_PROGID AS "plsql_subprogram_id", MA.MAX_PX_IS_CROSS_INSTANCE AS
"is_cross_instance", MA.MAX_PX_DOP AS "dop", MA.MAX_PX_DOP_INSTANCES AS "instances"), CASE WHEN
MA.MAX_ERROR_NUMBER IS NULL THEN NULL ELSE XML ELEMENT("error", XMLATTRIBUTES(MA.MAX_ERROR_NUMBER
AS "number", MA.MAX_ERROR_FACILITY AS "facility"), MA.MAX_ERROR_MESSAGE) END, XMLELEMENT("stats",
XMLATTRIBUTES('monitor' AS "type"), XMLELEMENT("stat", XMLATTRIBUTES('duration' AS "name"),
GREATEST(ROUND((MA.MAX_LAST_REFRESH_TIME- NVL(MA.MAX_SQL_EXEC_START, MA.MIN_FIRST_REFRESH_TIME))*
86400), LEAST(MA.SUM_ELAPSED_TIME/1000000, 1), CEIL(MA.MAX_QUEUEING_TIME/1000000))),
DECODE(MA.SUM_ELAPSED_TIME, 0, NULL, XMLELEMENT("stat", XMLATTRIBUTES('elapsed_time' AS "name"),
MA.SUM_ELAPSED_TIME)), DECODE(MA.MAX_QUEUEING_TIME, 0, NULL, XMLELEMENT("stat",
XMLATTRIBUTES('queuing_time' AS "name"), MA.MAX_QUEUEING_TIME)), DECODE(MA.SUM_CPU_TIME, 0, NULL,
XMLELEMENT("stat", XMLATTRIBUTES('cpu_time' AS "name"), MA.SUM_CPU_TIME)), DECODE(MA.SUM_USER_IO_WAIT_TIME,
0, NULL, XMLELEMENT("stat", XMLATTRIBUTES('user_io_wait_time' AS "name"), MA.SUM_USER_IO_WAIT_TIME)),
DECODE(MA.SUM_APPLICATION_WAIT_TIME, 0, NULL, XMLELEMENT("stat", XMLATTRIBUTES('application_wait_time' AS
"name"), MA.SUM_APPLICATION_WAIT_TIME)), DECODE(MA.SUM_CONCURRENCY_WAIT_TIME, 0, NULL, XMLELEMENT(
"stat", XMLATTRIBUTES('concurrency_wait_time' AS "name"), MA.SUM_CONCURRENCY_WAIT_TIME)),
DECODE(MA.SUM_CLUSTER_WAIT_TIME, 0, NULL, XMLELEMENT("stat", XMLATTRIBUTES('cluster_wait_time' AS "name"),
MA.SUM_CLUSTER_WAIT_TIME)), DECODE(MA.SUM_PLSQL_EXEC_TIME, 0, NULL, XMLELEMENT("stat",
XMLATTRIBUTES('plsql_exec_time' AS "name"), MA.SUM_PLSQL_EXEC_TIME)), DECODE(MA.SUM_JAVA_EXEC_TIME, 0, NULL,
XMLELEMENT("stat", XMLATTRIBUTES('java_exec_time' AS "name"), MA.SUM_JAVA_EXEC_TIME)),
DECODE(MA.SUM_OTHER_WAIT_TIME, 0, NULL, XMLELEMENT("stat", XMLATTRIBUTES('other_wait_time' AS "name"),
MA.SUM_OTHER_WAIT_TIME)), DECODE(MA.SUM_FETCHES, 0, NULL, XMLELEMENT("stat", XMLATTRIBUTES('user_fetch_
count' AS "name"), MA.SUM_FETCHES)), DECODE(MA.SUM_BUFFER_GETS, 0, NULL, XMLELEMENT("stat",
XMLATTRIBUTES('buffer_gets' AS "name"), MA.SUM_BUFFER_GETS)), DECODE(MA.SUM_READ_REQS, 0, NULL,
XMLELEMENT("stat", XMLATTRIBUTES('read_reqs' AS "name"), MA.SUM_READ_REQS)), DECODE(MA.SUM_READ_BYTES, 0,
NULL, XMLELEMENT("stat", XMLATTRIBUTES('read_bytes' AS "name"), MA.SUM_READ_BYTES)),
DECODE(MA.SUM_WRITE_REQS, 0, NULL, XMLELEMENT("stat", XMLATTRIBUTES('write_reqs' AS "name"),
MA.SUM_WRITE_REQS)), DECODE(MA.SUM_WRITE_BYTES, 0, NULL, XMLELEMENT("stat", XMLATTRIBUTES('write_bytes' AS
"name"), MA.SUM_WRITE_BYTES)), CASE WHEN MA.SUM_IO_INTER_BYTES IS NULL OR MA.SUM_IO_INTER_BYTES =
MA.SUM_IO_BYTES OR MA.SUM_IO_BYTES = 0 THEN NULL ELSE XMLELEMENT("stat", XMLATTRIBUTES('cell_offload_
efficiency' AS "name"), ROUND(MA.SUM_IO_BYTES / DECODE(MA.SUM_IO_INTER_BYTES, 0, 1, MA.SUM_IO_INTER_BYTES),
2)) END) ) ELSE XMLELEMENT("sql", XMLATTRIBUTES( MA.MAX_SQL_ID AS "sql_id", TO_CHAR(MA.MAX_SQL_EXEC_START,
:B12 ) AS "sql_exec_start", MA.MAX_SQL_EXEC_ID AS "sql_exec_id"), XMLFOREST(MA.MAX_STATUS AS "status",
SUBSTR(MA.SQLMON_TEXT, 1, 100) AS "sql_text", TO_CHAR(MA.MIN_FIRST_REFRESH_TIME, :B12 ) AS "first_refresh_time",
TO_CHAR(MA.MAX_LAST_REFRESH_TIME, :B12 ) AS "last_refresh_time", MA.SUM_REFRESH_COUNT AS "refresh_count",
MA.MAX_INST_ID AS "inst_id", MA.MAX_SESSION_ID AS "session_id", MA.MAX_SESSION_SERIAL AS "session_serial",
MA.MAX_USERID AS "user_id", MA.MAX_USERNAME AS "user", MA.MAX_CON_ID AS "con_id", MA.MAX_CON_NAME AS
"con_name", MA.MAX_MODULE AS "module", MA.MAX_ACTION AS "action", MA.MAX_SERVICE_NAME AS "service",
MA.MAX_CLIENT_ID AS "client_id", MA.MAX_CLIENT_INFO AS "client_info", MA.MAX_PROGRAM AS "program",
MA.MAX_PLAN_HASH_VALUE AS "plan_hash", MA.MAX_PL_ENTRY_OID AS "plsql_entry_object_id",
MA.MAX_PL_ENTRY_PROGID AS "plsql_entry_subprogram_id", MA.MAX_PL_OID AS "plsql_object_id", MA.MAX_PL_PROGID AS
"plsql_subprogram_id", MA.MAX_PX_IS_CROSS_INSTANCE AS "is_cross_instance", MA.MAX_PX_DOP AS "dop",
MA.MAX_PX_DOP_INSTANCES AS "instances", MA.SUM_SERVERS_REQUESTED AS "px_servers_requested",
MA.SUM_SERVERS_ALLOCATED AS "px_servers_allocated"), XMLELEMENT("stats", XMLATTRIBUTES('monitor' AS "type"),
XMLELEMENT("stat", XMLATTRIBUTES('duration' AS "name"), GREATEST(ROUND((MA.MAX_LAST_REFRESH_TIME-
MA.MAX_SQL_EXEC_START)*86400), LEAST(MA.SUM_ELAPSED_TIME/1000000, 1), CEIL(MA.MAX_QUEUEING_TIME/1000000))),
DECODE(MA.SUM_ELAPSED_TIME, 0, NULL, XMLELEMENT("stat", XMLATTRIBUTES('elapsed_time' AS "name"),
MA.SUM_ELAPSED_TIME)), DECODE(MA.MAX_QUEUEING_TIME, 0, NULL, XMLELEMENT("stat",
XMLATTRIBUTES('queuing_time' AS "name"), MA.MAX_QUEUEING_TIME)), DECODE(MA.SUM_CPU_TIME, 0, NULL,
XMLELEMENT("stat", XMLATTRIBUTES('cpu_time' AS "name"), MA.SUM_CPU_TIME)), DECODE(MA.SUM_USER_IO_WAIT_TIME,
0, NULL, XMLELEMENT("stat", XMLATTRIBUTES('user_io_wait_time' AS "name"), MA.SUM_USER_IO_WAIT_TIME)),
DECODE(MA.SUM_APPLICATION_WAIT_TIME, 0, NULL, XMLELEMENT("stat", XMLATTRIBUTES('application_wait_time' AS
"name"), MA.SUM_APPLICATION_WAIT_TIME)), DECODE(MA.SUM_CONCURRENCY_WAIT_TIME, 0, NULL, XMLELEMENT(
"stat", XMLATTRIBUTES('concurrency_wait_time' AS "name"), MA.SUM_CONCURRENCY_WAIT_TIME)),
DECODE(MA.SUM_CLUSTER_WAIT_TIME, 0, NULL, XMLELEMENT("stat", XMLATTRIBUTES('cluster_wait_time' AS "name"),
MA.SUM_CLUSTER_WAIT_TIME)), DECODE(MA.SUM_PLSQL_EXEC_TIME, 0, NULL, XMLELEMENT("stat",
XMLATTRIBUTES('plsql_exec_time' AS "name"), MA.SUM_PLSQL_EXEC_TIME)), DECODE(MA.SUM_JAVA_EXEC_TIME, 0, NULL,
XMLELEMENT("stat", XMLATTRIBUTES('java_exec_time' AS "name"), MA.SUM_JAVA_EXEC_TIME)),
DECODE(MA.SUM_OTHER_WAIT_TIME, 0, NULL, XMLELEMENT("stat", XMLATTRIBUTES('other_wait_time' AS "name"),
MA.SUM_OTHER_WAIT_TIME)), DECODE(MA.SUM_FETCHES, 0, NULL, XMLELEMENT("stat", XMLATTRIBUTES('user_fetch_

```

```

count' AS "name"), MA.SUM_FETCHES)), DECODE(MA.SUM_BUFFER_GETS, 0, NULL, XMLELEMENT( "stat",
XMLATTRIBUTES('buffer_gets' AS "name"), MA.SUM_BUFFER_GETS)), DECODE(MA.SUM_READ_REQS, 0, NULL,
XMLELEMENT( "stat", XMLATTRIBUTES('read_reqs' AS "name"), MA.SUM_READ_REQS)), DECODE(MA.SUM_READ_BYTES, 0,
NULL, XMLELEMENT( "stat", XMLATTRIBUTES('read_bytes' AS "name"), MA.SUM_READ_BYTES)),
DECODE(MA.SUM_WRITE_REQS, 0, NULL, XMLELEMENT( "stat", XMLATTRIBUTES('write_reqs' AS "name"),
MA.SUM_WRITE_REQS)), DECODE(MA.SUM_WRITE_BYTES, 0, NULL, XMLELEMENT( "stat", XMLATTRIBUTES('write_bytes' AS
"name"), MA.SUM_WRITE_BYTES)), CASE WHEN MA.SUM_IO_INTER_BYTES IS NULL OR MA.SUM_IO_INTER_BYTES =
MA.SUM_IO_BYTES OR MA.SUM_IO_BYTES = 0 THEN NULL ELSE XMLELEMENT( "stat", XMLATTRIBUTES('cell_offload_
efficiency' AS "name"), ROUND(MA.SUM_IO_BYTES / DECODE(MA.SUM_IO_INTER_BYTES, 0, 1, MA.SUM_IO_INTER_BYTES),
2)) END) ) END) FROM MONITOR_AGG MA) ELSE NULL END) FROM (SELECT CASE WHEN V1.XPLAN_XML IS NULL OR
V1.XPLAN_XML EXISTS NODE('/error') > 0 THEN NULL ELSE V1.XPLAN_XML END XPLAN_XML FROM (SELECT CASE WHEN
:B36 = 1 THEN DBMS_XPLAN.BUILD_PLAN_XML(TABLE_NAME=>'gv$sql_plan', PLAN_TAG=>'plan', FILTER_PREDS=>:B35 ,
FORMAT=>'-PROJECTION +ALIAS +ADAPTIVE') ELSE NULL END XPLAN_XML FROM DUAL) V1) CONST_VIEW
SELECT COUNT ( EMPLID || NAME || SEX || BIRTHDATE_MSK || CAMPUS_ID || NATIONAL_ID || NATIONAL_ID_MSK ||
NID_COUNTRY || NID_DESCRSHORT || LAST_NAME_SRCH || FIRST_NAME_SRCH ) AS COUNT_KEYS FROM
PS_PEOPLE_SRCH A WHERE OPRCLASS=:1
INSERT INTO PS_SAA_ADB_CRSEAVL ( EMPLID , ANALYSIS_DB_SEQ , SAA_CAREER_RPT , SAA_ENTRY_SEQ ,
SAA_COURSE_SEQ , RPT_DATE , TSCRPT_TYPE , SAA_RPT_IDENTIFIER , COURSE_LIST , EFFDT , R_COURSE_SEQUENCE ,
RQRMNT_GROUP , RQ_GRP_LINE_NBR , REQUIREMENT , RQ_LINE_NBR , SUBJECT , CATALOG_NBR , CRSE_ID ,
CRS_TOPIC_ID , RQMNT_DESIGNTN , DESCR , SAA_WHERE_DETAIL ) VALUES ( :1 , :2 , :3 , :4 , :5 , :6 , :7 , :8 , :9 , :10 , :11 , :12 , :13 ,
:14 , :15 , :16 , :17 , :18 , :19 , :20 , :21 , NULL)
BEGIN sys.dbms_auto_report_internal.i_save_report (:rep_ref , :snap_id , :pr_class , :rep_id , :suc); END;
SELECT ITEM_TYPE , ITEM_EFFECTIVE_DT FROM PS_ITEM_SF K WHERE K.COMMON_ID = :B2 AND K.ACCOUNT_TERM IN
(:B1 ) AND K.ITEM_TYPE IN ('611100001000', '611100001100', '611100001200', '611100001300', '611100002600', '691000000500',
'900000000103', '900000000110', '900000000113', '900000000120', '900000000135')
SELECT a.user_pswd from ps_u_cam011_result a where a.emplid = :1 and a.user_pswd > ''
DECLARE BEGIN U_SAR295_P2('2211', '2212', '2215', '', '', 3125655); END;
SELECT PR.EMPLID, PR.ACAD_PROG, PL.ACAD_PLAN, PL2.DESCR FROM PS_ACAD_PROG PR, PS_ACAD_PLAN PL,
PS_ACAD_PLAN_TBL PL2 WHERE PR.EMPLID = PL.EMPLID AND ( PR.EFFDT = (SELECT MAX(PR2.EFFDT) FROM
PS_ACAD_PROG PR2 WHERE PR2.EMPLID = PR.EMPLID AND PR.ACAD_CAREER = PR2.ACAD_CAREER AND
PR.STDNT_CAR_NBR = PR2.STDNT_CAR_NBR AND PR2.EFFDT <= SYSDATE) OR ( PR.EFFDT = (SELECT MAX(PR2.EFFDT)
FROM PS_ACAD_PROG PR2 WHERE PR2.EMPLID = PR.EMPLID AND PR.ACAD_CAREER = PR2.ACAD_CAREER AND
PR.STDNT_CAR_NBR = PR2.STDNT_CAR_NBR ))) AND PR.EFFSEQ = (SELECT MAX(PR2.EFFSEQ) FROM PS_ACAD_PROG
PR2 WHERE PR.EMPLID = PR2.EMPLID AND PR.ACAD_CAREER = PR2.ACAD_CAREER AND PR.STDNT_CAR_NBR =
PR2.STDNT_CAR_NBR AND PR.EFFDT = PR2.EFFDT) AND (PR.PROG_STATUS = 'AC' OR PR.PROG_STATUS = 'LA') AND
PR.STDNT_CAR_NBR = (SELECT MIN(PR4.STDNT_CAR_NBR) FROM PS_ACAD_PROG PR4 WHERE PR.EMPLID =
PR4.EMPLID AND PR.ACAD_CAREER = PR4.ACAD_CAREER AND PR.PROG_STATUS = PR4.PROG_STATUS) AND
PR.ACAD_CAREER = PL.ACAD_CAREER AND PR.STDNT_CAR_NBR = PL.STDNT_CAR_NBR AND PR.EFFSEQ = PL.EFFSEQ
AND PR.EFFDT = PL.EFFDT AND PL.EFFDT = (SELECT MAX(PL3.EFFDT) FROM PS_ACAD_PLAN PL3 WHERE PL3.EMPLID =
PL.EMPLID AND PL3.EFFDT = PL.EFFDT) AND PL2.ACAD_PLAN = PL.ACAD_PLAN AND PL.PLAN_SEQUENCE = (SELECT
MIN(PL2.PLAN_SEQUENCE) FROM PS_ACAD_PLAN PL2 WHERE PL2.EMPLID = PR.EMPLID AND PL2.ACAD_CAREER =
PR.ACAD_CAREER AND PL2.STDNT_CAR_NBR = PR.STDNT_CAR_NBR AND PL2.EFFDT = PR.EFFDT AND PL2.EFFSEQ =
PR.EFFSEQ) AND PL2.EFFDT = (SELECT MAX(PL4.EFFDT) FROM PS_ACAD_PLAN_TBL PL4 WHERE PL2.ACAD_PLAN =
PL4.ACAD_PLAN) AND NOT EXISTS (SELECT INFO.EMPLID FROM PS_FERPA_OVERRIDE INFO WHERE PR.EMPLID =
INFO.EMPLID AND INFO.RECNAME = 'MAJOR_FERPA_VW')

```

[Back to Supplemental Information](#)
[Back to Top](#)

Active Session History(ASH) Report

- [Top SQL with Top Events](#)
- [Top SQL with Top Row Sources](#)
- [Top Sessions](#)
- [Top Blocking Sessions](#)
- [Top PL/SQL Procedures](#)
- [Top Events](#)
- [Top Event P1/P2/P3 Values](#)
- [Top DB Objects](#)
- [Activity Over Time](#)

[Back to Top](#)

Top SQL with Top Events

- Top SQL statements by DB Time along with the top events by DB Time for those SQLs.
- % Activity is the percentage of DB Time due to the SQL.
- % Event is the percentage of DB Time due to the event that the SQL is waiting on.
- % Row Source is the percentage of DB Time due to the row source for the SQL waiting on the event.
- Executions is the number of executions of the SQL that were sampled in ASH.

SQL ID	Plan Hash	Executions	% Activity	Event	% Event	Top Row Source	% Row Source	SQL Text
3nkwt9sq77s4	3996409867	20		db file 21.62 sequential read	15.71	TABLE ACCESS - BY INDEX ROWID BATCHED	14.70	SELECT ITEM_NBR , ITEM_AMT , A...
3nkwt9sq77s4	3996409867	20		21. CPU + 621621621621621621621621 Wait for	3.55	TABLE ACCESS - BY INDEX ROWID	3.21	

	7027027		V1-V3)			
155,18648	2.36 CPU + Wait for CPU	1.35 SYSADM	PSRUN@augusta....u (TNS V1-V3)	8/361 [2%]	8	
21,35201	1.69 CPU + Wait for CPU	1.18 SYSADM	PSAPPSRV@augus...u (TNS V1-V3)	7/361 [2%]	0	

[Back to Active Session History\(ASH\) Report](#)
[Back to Top](#)

Top Blocking Sessions

- Blocking session activity percentages are calculated with respect to waits on enqueues, latches and "buffer busy" only
- '% Activity' represents the load on the database caused by a particular blocking session
- '# Samples Active' shows the number of ASH samples in which the blocking session was found active.
- 'XIDs' shows the number of distinct transaction IDs sampled in ASH when the blocking session was found active.

Blocking Sid (Inst)	% Activity	Event Caused	% Event	User	Program	# Samples Active	XIDs
959, 4286(3)	1.35	read by other session	1.35	SYSADM	PSRUN@augusta....u (TNS V1-V3)	69/361 [19%]	10

[Back to Active Session History\(ASH\) Report](#)
[Back to Top](#)

Top PL/SQL Procedures

- 'PL/SQL entry subprogram' represents the application's top-level entry-point (procedure, function, trigger, package initialization or RPC call) into PL/SQL.
- 'PL/SQL current subprogram' is the PL/SQL subprogram being executed at the point of sampling . If the value is 'SQL', it represents the percentage of time spent executing SQL for the particular PL/SQL entry subprogram

PL/SQL Entry Subprogram	% Activity	PL/SQL Current Subprogram	% Current
SYSADM.U_SIF004	7.26	SQL	7.26
SYS.DBMS_AUTO_REPORT_INTERNAL.I_SAVE_REPORT	2.20	SQL	2.20
SYSADM.U_SAR295_P2	1.35	SQL	1.18
SYSADM.U_HRL002_001_MAIN	1.18	SQL	1.18

[Back to Active Session History\(ASH\) Report](#)
[Back to Top](#)

Top Events

- Top Events by DB Time
- % Activity is the percentage of DB Time due to the event

Event	Event Class	Session Type	% Activity	Avg Active Sessions
CPU + Wait for CPU	CPU	BACKGROUND	51.01	0.84
db file sequential read	User I/O	BACKGROUND	20.61	0.34
CPU + Wait for CPU	CPU	BACKGROUND	8.78	0.14
SQL*Net message from dblink	Network	BACKGROUND	2.20	0.04
read by other session	User I/O	BACKGROUND	2.19594594594594594594594594594595	0.04

[Back to Active Session History\(ASH\) Report](#)
[Back to Top](#)

Top Event P1/P2/P3 Values

- Top Events by DB Time and the top P1/P2/P3 values for those events.
- % Event is the percentage of DB Time due to the event
- % Activity is the percentage of DB Time due to the event with the given P1,P2,P3 Values.

Event	% Event	P1, P2, P3 Values	% Activity	Parameter 1	Parameter 2	Parameter 3
db file sequential read	20.61	"56","620325","1"	0.17	file#	block#	blocks
SQL*Net message from dblink	2.20	"675562835","1","0"	2.03	driver id	#bytes	NOT DEFINED
read by other session	2.20	"122","1169403","1"	0.17	file#	block#	class#
direct path read	1.52	"1","326912","128"	0.17	file number	first dba	block cnt

[Back to Active Session History\(ASH\) Report](#)
[Back to Top](#)

Top DB Objects

- Top DB Objects by DB Time with respect to Application, Cluster, User I/O, buffer busy waits and In-Memory DB events only.

- Tablespace name is not available for reports generated from the root PDB of a consolidated database.

Object ID	% Activity	Event	% Event	Object Name (Type)	Tablespace
91349		17.23 db file sequential read	14.86	SYSADM.PS_ITEM_SF (TABLE)	SAAPP
91349	17.22972972972972972972972972972973	read by other session	1.86	SYSADM.PS_ITEM_SF (TABLE)	SAAPP
161808		1.35 db file sequential read	1.01	SYSADM.PSGITEM_SF (INDEX)	PSINDEX

[Back to Active Session History\(ASH\) Report](#)

[Back to Top](#)

Activity Over Time

- Analysis period is divided into smaller time slots as indicated in the 'Slot Time (Duration)'.
- Top 3 events are reported in each of those slots
- 'Slot Count' shows the number of ASH samples in that slot
- 'Event Count' shows the number of ASH samples waiting for that event in that slot
- '% Event' is 'Event Count' over all ASH samples in the analysis period

Slot Time (Duration)	Slot Count	Event	Event Count	% Event
10:00:05 (-5 secs)	2	CPU + Wait for CPU	2	0.34
10:00:05 (4.9 min)	110	CPU + Wait for CPU	46	7.77
10:00:05 (4.9 min)	110	db file sequential read	30	5.07
10:00:05 (4.9 min)	110	SQL*Net message from dblink	13	2.20
10:05:00 (5.0 min)	70	CPU + Wait for CPU	36	6.08
10:05:00 (5.0 min)	70	db file sequential read	23	3.89
10:05:00 (5.0 min)	70	db file parallel read	2	0.34
10:10:00 (5.0 min)	31	CPU + Wait for CPU	16	2.70
10:10:00 (5.0 min)	31	db file sequential read	4	0.68
10:10:00 (5.0 min)	31	SQL*Net more data to client	2	0.34
10:15:00 (5.0 min)	83	CPU + Wait for CPU	38	6.42
10:15:00 (5.0 min)	83	db file sequential read	31	5.24
10:15:00 (5.0 min)	83	read by other session	8	1.35
10:20:00 (5.0 min)	71	CPU + Wait for CPU	35	5.91
10:20:00 (5.0 min)	71	db file sequential read	26	4.39
10:20:00 (5.0 min)	71	read by other session	5	0.84
10:25:00 (5.0 min)	36	CPU + Wait for CPU	27	4.56
10:25:00 (5.0 min)	36	db file sequential read	4	0.68
10:25:00 (5.0 min)	36	gc cr block 3-way	1	0.17
10:30:00 (5.0 min)	54	CPU + Wait for CPU	49	8.28
10:30:00 (5.0 min)	54	db file sequential read	2	0.34
10:30:00 (5.0 min)	54	direct path read	2	0.34
10:35:00 (5.0 min)	41	CPU + Wait for CPU	34	5.74
10:35:00 (5.0 min)	41	direct path read	2	0.34
10:35:00 (5.0 min)	41	db file sequential read	1	0.17
10:40:00 (5.0 min)	20	CPU + Wait for CPU	14	2.36
10:40:00 (5.0 min)	20	gc current block 2-way	2	0.34
10:40:00 (5.0 min)	20	control file sequential read	1	0.17
10:45:00 (5.0 min)	21	CPU + Wait for CPU	16	2.70
10:45:00 (5.0 min)	21	db file scattered read	1	0.17
10:45:00 (5.0 min)	21	db file sequential read	1	0.17
10:50:00 (5.0 min)	25	CPU + Wait for CPU	21	3.55
10:50:00 (5.0 min)	25	gc current block 2-way	2	0.34
10:50:00 (5.0 min)	25	direct path read	1	0.17
10:55:00 (5.0 min)	28	CPU + Wait for CPU	20	3.38
10:55:00 (5.0 min)	28	library cache pin	4	0.68
10:55:00 (5.0 min)	28	direct path read	1	0.17

[Back to Active Session History\(ASH\) Report](#)

[Back to Top](#)

ADDM Task ADDM:2413530610_61361

Error encountered in special section 99994 ORA-13605: The specified task or object does not exist for the current user.
Continuing to Next Section...

[Back to Top](#)

End of Report