An Oracle White Paper
March 2010

Oracle Real User Experience Insight Implementation Case Study

Monitoring Oracle's Global Single Instance

Executive Overview

Oracle's Global Single Instance (GSI) is the worldwide E-Business environment used internally to run Oracle's business, and externally by many of its customers. It combines disparate ERP systems, managing numerous applications with a single database. GSI usage statistics are staggering: its website averages 300,000 peak page requests per hour—3.5 *million* each business day. At any given moment, there are 15,000 concurrent users. The primary benefit the environment offers them is transparent data available 24/7 across the enterprise, resulting in a faster turnaround of all business processes. Proactive analysis of GSI end-user experience is critical to Oracle's success, and traditional monitoring tools do not always identify problems.

Introduction

Separate monitoring tools have been used for each layer of the GSI architecture—tools for the database (e.g. SQL tracing), others for the middle tier (e.g. Apache Logs), and still others for the network between Oracle and client (e.g.: EM ASLM tools). None provide a 360-degree view of application performance, availability and usage. Without holistic monitoring of user transactions, performance issues went undetected until they became critical or were reported by the users themselves.

To address this shortfall, Oracle deployed Real User Experience Insight (RUEI). RUEI uses groundbreaking, state-of-the-art network protocol analysis (NPA) to capture initial user requests and track them through all responses, collecting, processing and presenting details of every transaction. Once installed and running, RUEI instantly eradicated blind spots and provided GSI support analysts with a real-world, end-to-end vision of user experience. Performance issues were swiftly identified and neutralized, *before* they could become critical.

This technical white paper highlights the deployment of RUEI to monitor Oracle's Global Single Instance. It describes how RUEI works and discusses key features, hardware and database installation and configuration options, security concerns, and the implementation's pitfalls and successes. Customers will gain insight into the powerful monitoring solution RUEI provides, and learn how best to apply it to their own infrastructure.

Product Overview – How RUEI Works

RUEI provides end-to-end monitoring based on network protocol analysis, a process of decoding network protocol headers and trailers that is industry-standard, secure and unobtrusive. Each incoming page request is captured and matched with its outgoing response; response time and status are then stored in a repository for subsequent analysis and tracking.

Using a powerful OLAP engine, support analysts can quickly determine the most frequently accessed business functions, which pages are performing poorly, and how time is being spent within each component of the environment. In addition to standard reports, custom reports can be created to link pages together—for example, the pages required to submit a product order. The reports can then be utilized for capacity planning and alerts.

Key Features of RUEI

A number of significant elements differentiate RUEI from traditional monitoring tools:

- It monitors the entire technology stack, combining data from each layer of architecture into a holistic analysis of application performance, availability and usage.
- Unlike tools that *simulate* end-user experience from a datacenter perspective, such as Application Service Level Management (ASLM), RUEI provides real-world data directly from the end user. Analysts see the pages visited during specific sessions from the user's perspective, allowing for swift root-cause analysis of performance issues.
- It enables data to be viewed in a variety of ways: examples include seeing performance and volumes by either page or application, and choosing to view either internal or external data.
- It breaks data down into network and server latencies, allowing analysts to assess the role each component is playing in total response time.
- It allows analysts to drill down into sessions that are receiving errors and review their progression.
- It tracks historical performance of specific pages for trend analysis.
- It contains a rich set of customizable reports and alerts. Transactions can be defined for key business flows; key performance indicators (KPIs) can be created for functionality, along with alerts to warn of performance issues.
- KPIs can be incorporated into the Service Level Agreements (SLAs) that define the level of service required.
- The RUEI Accelerator for Oracle E-Business Suite correlates JSPs, Forms and other network objects with the appropriate applications (Receivables, iProcurement, iStore etc.), allowing KPIs and reports to be customized for specific LOB areas of interest.

Architectural Choices and Designs

In order to function—reassembling TCP/IP packets and pairing them with the corresponding HTTP requests—RUEI requires a connection to the networks that transmit and receive traffic between the client and the web servers. The type and placement of this network connection impacts the capabilities and performance of the completed installation.

Types of Network Connections

RUEI's mechanisms of attachment to the network are secure, unobtrusive and have no negative impact on network performance. There are two connection options: utilizing the span port of a network switch, or via a specialized tap device that replicates the network traffic in a read-only mode.

Network Connection Option #1 – Span Port

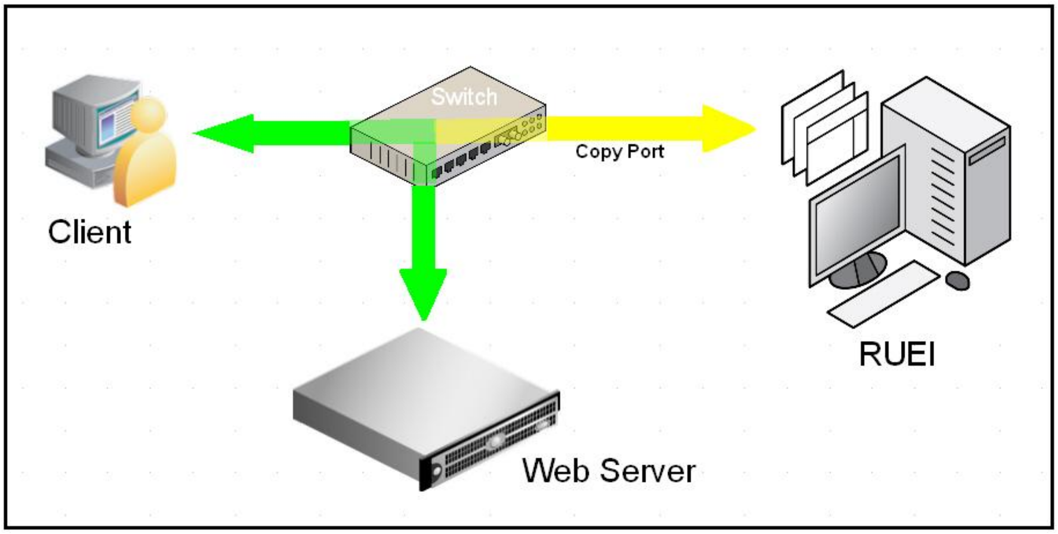
A network connection is made to the span (copy) port of a network switch.

There are two distinct advantages to this type of connection:

- It requires no additional hardware components between the switch and the RUEI collector.
- It requires only one connection between the collector and the network.

There are also disadvantages:

- Network switches have a limited number of copy ports, and they are commonly used for other diagnostics.
- Future network analysis/diagnostic devices may be compromised.
- Span ports can drop packets when traffic is heavy, causing errors to the monitored application.
- Use of the copy port can potentially alter the behavior of the network switch.



Network Connection Using a Span Port

Network Connection Option #2 – Tap Device

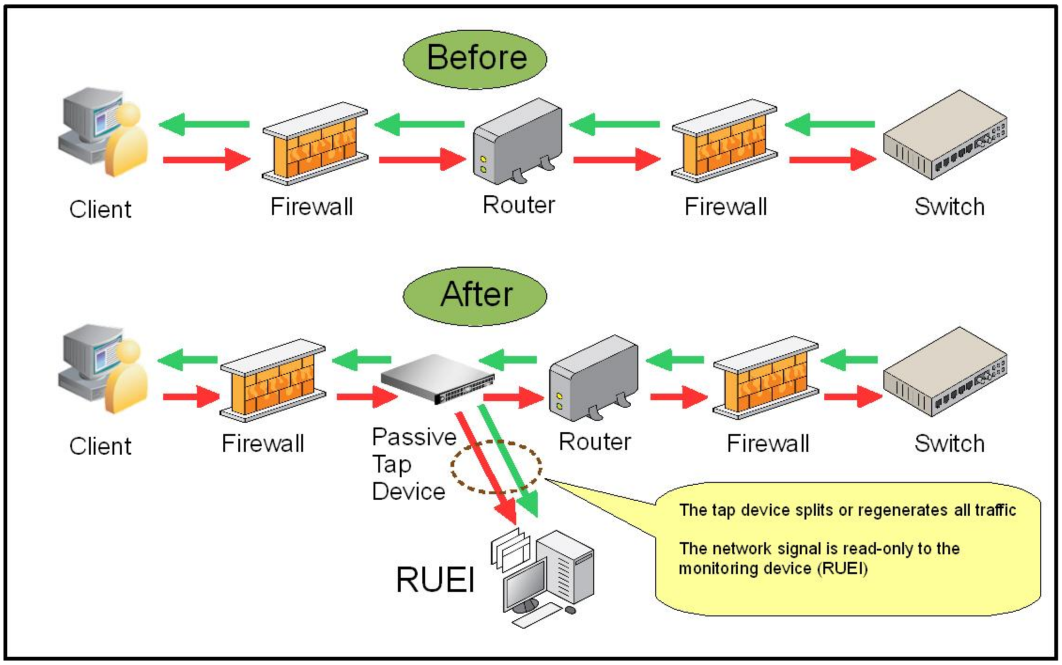
A tap device is placed between switches in a fully passive manner. The RUEI collector reads the traffic that the tap device reads and regenerates.

There are three central advantages to this type of connection:

- There is no chance of data loss (dropped packets).

- The device is read-only and has no IP address, making it secure.
- It does not change the behavior of the monitored network.

The key disadvantage of a tap connection is that it requires a minimum of two ports on the RUEI server.



Network Connection Using a Tap Device

During the GSI implementation, both types of connections were tested. The RUEI server was initially deployed to monitor several small test environments via a span connection, and that connection was subsequently replaced by a tap device. There was no discernable impact on the network with either configuration, but the monitored traffic was minimal during testing of the span connection. The scale of the implementation prompted the choice of a tap device for the end-state architecture of the production environment in order to avoid the risk of impacting network performance, and to keep additional span ports available for diagnostics.

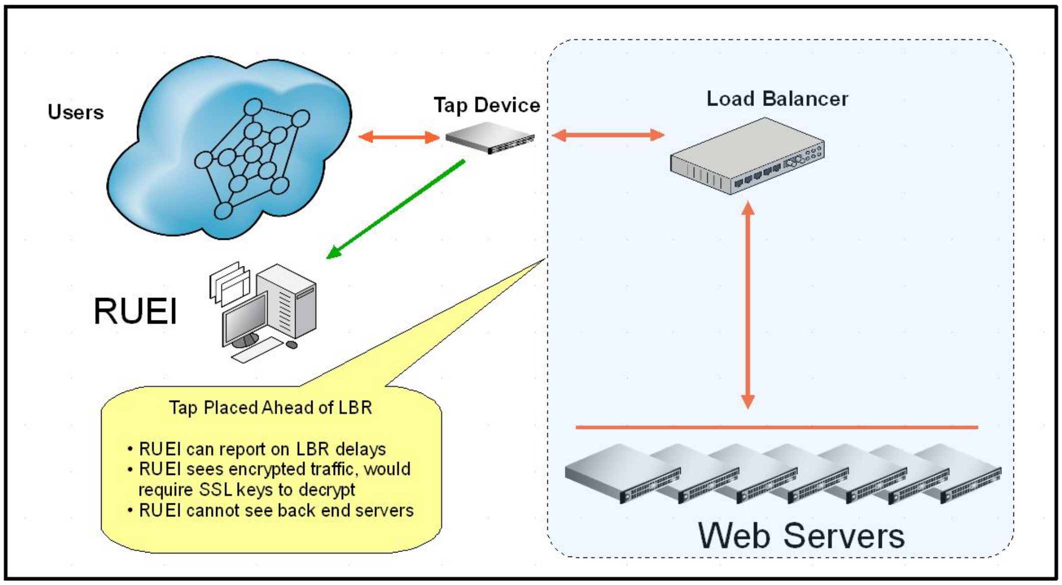
Placement of Network Connections

The tap device may be placed either in front of or behind a Load Balancing Router (LBR).

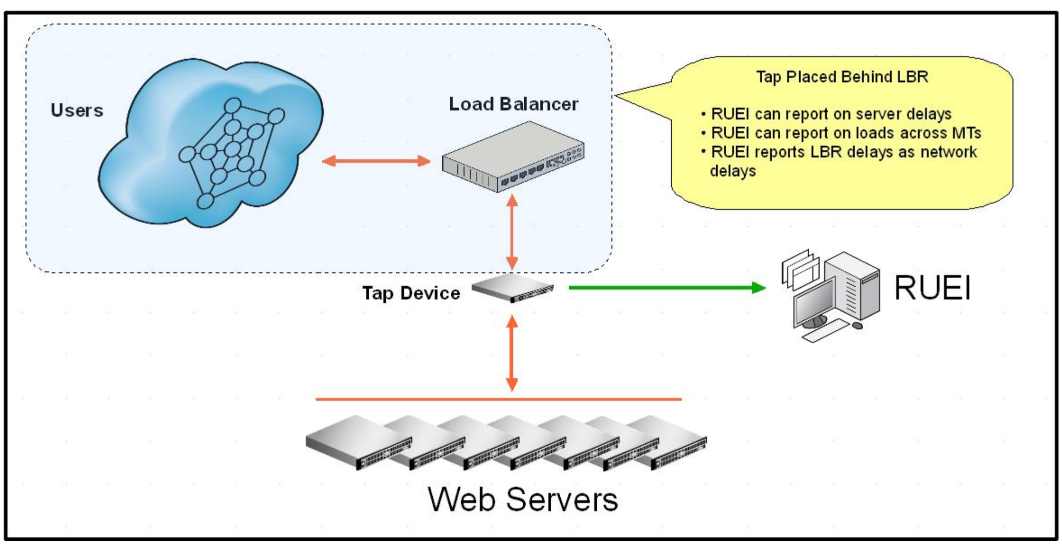
There are arguments for and against each option:

- Placing it in front of the LBR has the advantage of simplified IP filtering, but traffic in that location is generally encrypted, requiring RUEI to be supplied with decryption keys.
- Placing it behind the LBR has the advantages of enhanced server/load visibility, and added security; the data is decrypted, requiring no disclosure of encryption information. The disadvantage of this placement is that RUEI does not differentiate between network delays and processing time within the LBR—any delays between the user and RUEI are attributed to the network by default.

The choice of placement depends upon the complexity of the web and application tiers, and overall objectives. The GSI architecture includes dozens of middle tier servers configured into pools based upon application and function (Oracle Self Service, Forms, Concurrent Manager, etc.). Ensuring proper balancing across the middle tier was paramount. In order to enable RUEI to match specific servers with the requests they were handling and report the load of each server and pool, the tap device was placed *behind* the LBR. One hypothetical scenario calling for placement of the device in *front* is a situation in which LBR performance issues are of concern, and therefore the LBR needs to be monitored. RUEI enables the tracking of any response-time degradation caused by the load balancers; this was not relevant to the GSI implementation, because performance issues caused by LBRs within the environment are extremely rare.



Tap Placed Ahead of Load Balancer (LBR)



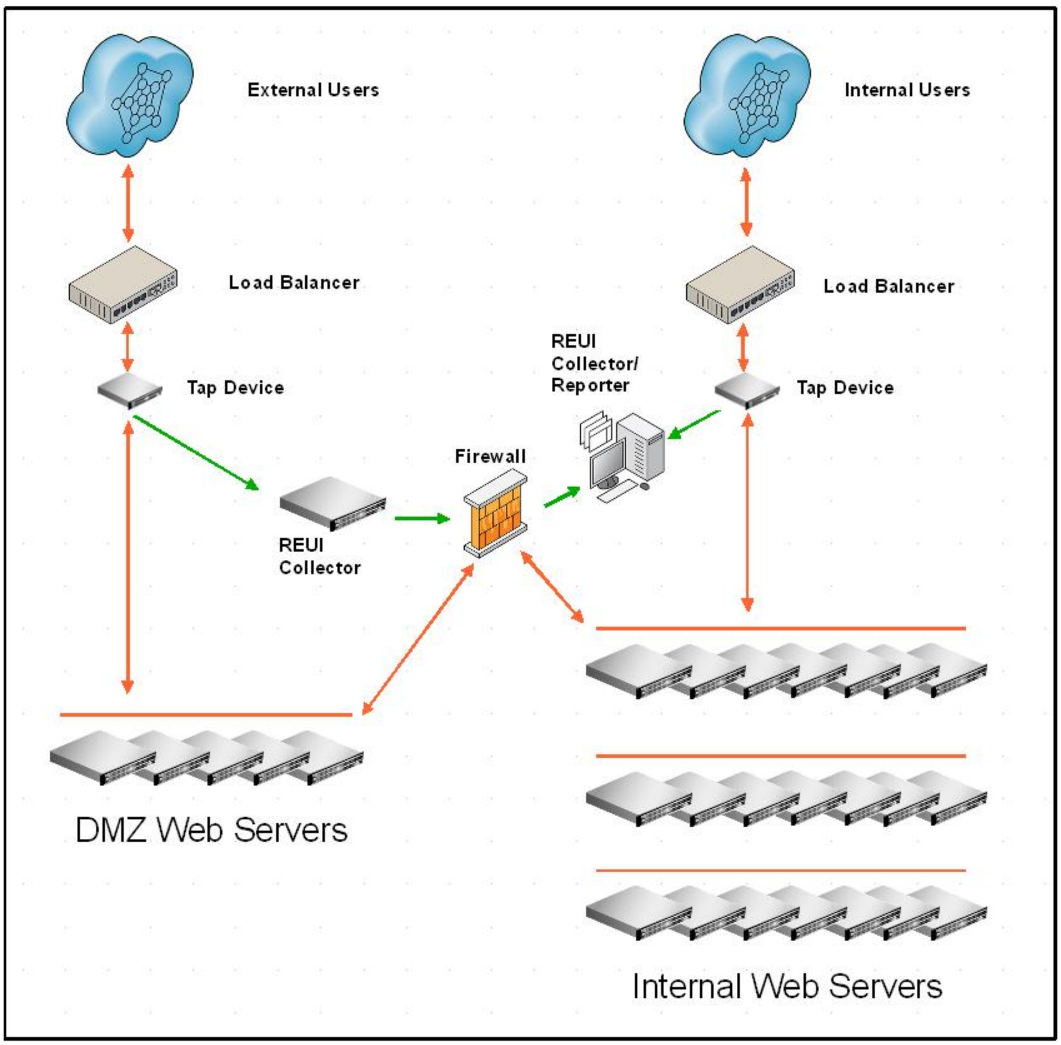
Tap Placed Behind Load Balancer (LBR)

Deployment Architecture for GSI

The *Real User Experience Insight Installation Guide* describes deployment options extensively. Each system contains Collector and Reporter components that are configured separately. A Collector gathers data from the networks it monitors and submits it to a Reporter system that contains the RUEI UI application server and repository database.

The Collector and Reporter can be deployed on separate servers. This is beneficial in high traffic or multiple network (e.g. DMZ and Internal) environments, as well as for failover to a disaster recovery site.

Because Oracle's GSI is used both internally, and externally by Oracle customers and partners, its architectural components straddle the firewall. It was equally important to monitor external traffic, therefore a split-server deployment consisting of a Collector residing in the DMZ and a Reporter residing in the intranet was chosen. This is a highly secure architecture, because the tap is a read-only device with no IP address. The Reporter, along with its UI and database, are protected behind the firewall, and the Collector and Reporter communicate through a secure connection.



RUEI and GSIAP Application Server Architecture

Hardware/Software Configuration

The installation guide details minimum hardware requirements. Standardized commodity middle tier servers were used for the RUEI Collector and Collector / Reporter for the GSI implementation. The servers are oversized from the perspective of CPU, memory and disk. Deployed components and disk capacity are highlighted below.

<table>
<tr><th>Item</th><th>Model / Version</th></tr>
<tr><td>Network Tap</td><td>(2) NetOptics SX Regenerative Tap IL Dual (Model RGN-SX5-IL2-D), special order for Oracle / RUEI</td></tr>
<tr><td>RUEI Servers</td><td>
<table>
<tr><td>Nodes:</td><td>DMZ Collector & Intranet Collector/Reporter</td></tr>
<tr><td>Model:</td><td>Dell PE2950 III</td></tr>
<tr><td>CPU:</td><td>2 x Quadcore</td></tr>
<tr><td>Memory:</td><td>32x GB 667 Mhz Dual ranked RAM</td></tr>
<tr><td>Ethernet Card:</td><td>2 x Intel PRO/1000 PF Quad Port NIC</td></tr>
<tr><td>Disk:</td><td>8x SAS: 146 GB 10K RPM
- 2 x OS Disks
- 6 disks configured in a RAID5 array (1030 GB usable)</td></tr>
<tr><td>OS:</td><td>OEL 5 (2.6.18-92.el5) 64 bit</td></tr>
<tr><td>RUEI:</td><td>6.0.1 with EBS Accelerator</td></tr>
<tr><td>DB:</td><td>Oracle 11.1.07 64 bit</td></tr>
</table>
</td></tr>
</table>

Database Considerations

RUEI uses an Oracle database running locally on the RUEI server as its data repository. A standard RUEI installation operates as a maintenance-free, standalone environment that can be termed an appliance. It manages available disk space automatically, recycling space and increasing performance by rolling daily data up to weekly data, weekly data to monthly data and monthly data to yearly data. Some details may be lost during the aggregations; retention periods can be configured to align with individual requirements.

The standard database installed by RUEI is not intended to be backed up. However, RUEI does provide features that export user configuration settings such as users, passwords, security roles, reports and KPIs.

Oracle's Applications IT group manages hundreds of databases. There was a large group of analysts accessing the RUEI environment during the GSI implementation, making it a highly fluid environment from the standpoint of user base, reports, KPIs, etc. Even at low risk, the loss of user configurations and/or performance data was unacceptable. To mitigate these risks, the RUEI repository database was aligned with existing standards to allow DBAs to support the environment with standard Oracle RMAN backup and recovery processes and procedures.

The IT group employs a general strategy for backing up the operating system, application, and database:

- The OS disk is mirrored, which protects the disk in case of local failure. In the event of a catastrophic loss of the entire machine, the disk can be reinstalled quickly via a standard Linux image.
- Non-database application disk is written to tape. This includes the RUEI application software, configuration files and data logs.
- The database is backed up via RMAN to disk, then tape, on a daily basis. Archive logs are written to tape when the archive log file system reaches 80% capacity.

Database Configuration

Changes were made to RUEI's out-of-the-box configuration to meet implementation requirements:

- Oracle RDBMS 11.1.0.7
- Database was configured to support RMAN backups and reduce the overall disk footprint:
  - Archive log mode
  - Configured for forced logging to support backups, because many of RUEI's frequent data logging operations are executed with no redo logging for performance purposes.
- Advanced Compression was enabled at the tablespace level to reduce the overall disk footprint. This feature has been added to the standard product, and can be configured via the User Interface in RUEI Version 6.
- AUTOEXTEND of the database files was disabled in order to control database growth and ensure maintenance of a large amount of historical performance data.

Database Memory Configuration

- SGA max size: 1200MB
- Shared pool reserved: 36MB
- PGA Aggregate Target: 240MB
- Log Buffer: 8MB

File System Configuration

RUEI has basic space requirements, but disk number and layout is customizable. GSI implementation specifics are highlighted below.

File System Usage	Size (GB)
RUEI Software & Logs	170
EM Agent	This is a shared file system mounted on all hosts monitored by EM
ORACLE_HOME	30
Oracle Archive Logs	60
Oracle Trace	20
Oracle datafiles	250
Flashback / RMAN	500

Security

RUEI allows HTML page content to be recorded, and the details of pages visited by any user session to be viewed. While this makes it a powerful diagnostic tool for root-cause analysis of operational issues, it also necessitates data protection.

RUEI's security capabilities are flexible and granular. It provides a mechanism to mask URL POST arguments, HTTP header content, cookie logging, and URL prefixes.

For a simple application, it may be adequate to specify the URL POST arguments that should be masked within RUEI; this approach can be thought of as "blacklisting". For a more complex application, all data values can be disabled, with only those that are not confidential or are required for ease of analysis specified. This may be thought of as "whitelisting".

Support analysts and members of the security team within Oracle Applications IT determined a security configuration that worked with the applications being monitored, the types of data being passed via the POST arguments, and the overall security requirements. Because the GSI contains over 37 billion rows of data and thousands of different URL arguments, the implementation involved a vast amount of potentially sensitive data, and therefore "whitelisting" was chosen. All page content was masked by default; selected arguments were subsequently unmasked, such as Application & Responsibility Ids and other non-confidential elements that are critical to analysis.

Implementation Pitfalls

Although the implementation went smoothly, there were minor stumbling blocks. Future implementations may or may not face similar issues, depending upon the standards of the IT organization and the hardware platform.

1. When RUEI was installed, a standard Oracle Enterprise Linux 5 (OEL5) image that had recently been created was used. This image was missing some of the packages that were required by RUEI, as documented in the installation guide. There was a delay as those discrepancies were resolved.

2. The database volumes were configured within a single RAID-5 array. RUEI's logging operations (inserting data, performing cube rollups, etc.) have been slightly impacted by the inherent performance hit of using RAID-5. This is exhibited within the database as relatively poor, but acceptable wait times on 'log file sync' (10 milliseconds). This was deemed acceptable, because only RUEI's *internal* operations are affected; user queries are *not* impacted by this disk configuration, and the disk-protection benefits of RAID-5 outweigh the performance issues.

Implementation Successes

The deployment of RUEI had an immediate positive impact, even before personnel were proficient in its use. It is being rolled out to a wide group of internal support analysts, and is expected to have a profound impact on Oracle's monitoring and management strategies. The following cases are just a few of the implementation experiences that highlight RUEI's diagnostic capabilities:

Case 1: External Customers

The Partner Ordering Portal is an external-facing application used by Oracle customers and partners to order products. Its product support team had been receiving slow-performance complaints from users, but the team was unable to duplicate the performance problems when conducting the same activities.

Traditional tools such as database SQL tracing and mining of Apache Logs were used with only minor success.

When RUEI was deployed, it became apparent that certain page requests were consistently slow when their point of origin was a specific location in Asia. Strangely, other application pages requested from the same location were not having the same trouble.

RUEI support analysts examined the pages requested by sessions from that location, and immediately noticed a correlation between Page Not Found (404) errors returned to the client with slow page deliveries. There was no problem with the relevant applications; the *location* had a slow connection to the internet. When coupled with a missing page element—in this case, an image file that was not being cached—each request for the troubled pages caused a request for the missing element from the client, and a non-fatal error was then returned to the client's browser stating that the requested file did not exist.

Normally, the page element would have been cached on the client machine but in this case, each page view caused an unnecessary, round trip attempt to retrieve the page element over a slow internet connection. RUEI was able not only to diagnose the slow connection, but to quickly drill down to the missing page element and find that the image file was in the wrong location on the application server; the problem, which had persisted for months, was fixed within hours.

Case 2: Complaints from an Executive

An Oracle executive using the Compensation Workbench application was experiencing intermittent performance issues. As he updated employee records, his sessions would occasionally hang when he saved a change, and then time out. In the majority of cases his updates went smoothly, but when sessions timed out he was forced to log back in and start over.

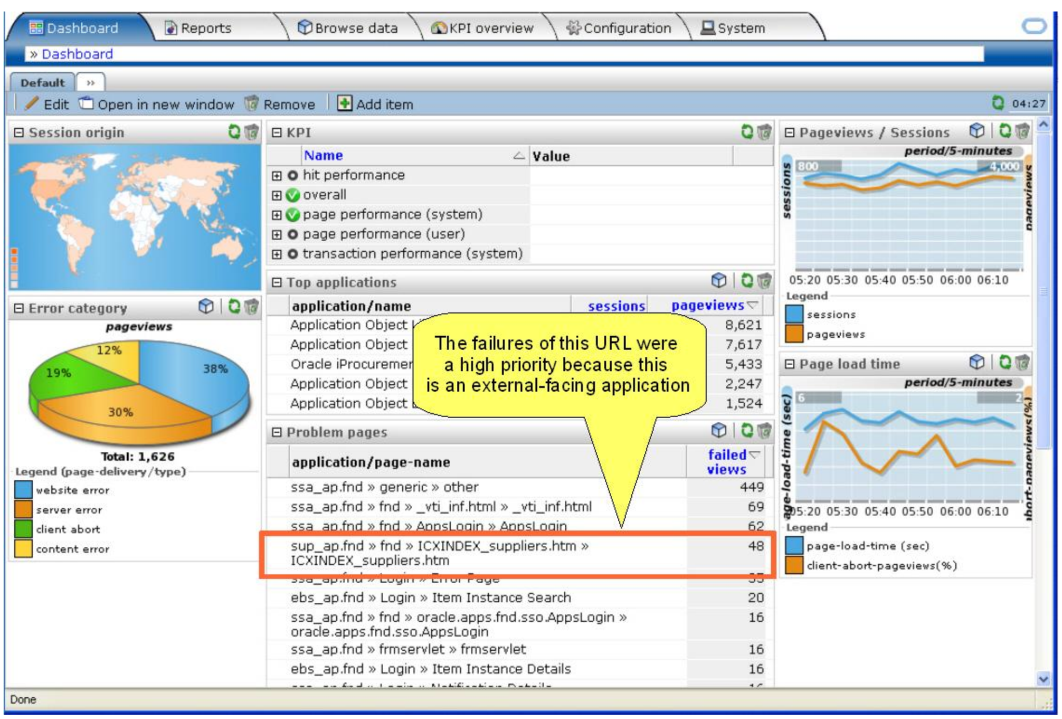
Analysts examined his sessions with RUEI's Session Diagnostics feature, and a distinct pattern was evident – 7 updates were fast, and the 8th led to failure. This seemed to rule out the database as the culprit, and the middle tier was examined. The behavior of the JVMs revealed that garbage collection intervals needed adjustment, and the issue was quickly resolved.

Without RUEI, root-cause analysis would have been arduous. Considerable time would have been spent unsuccessfully attempting to reproduce the issue either in a test environment, or by searching for nonexistent database issues in the production environment using AWR data from the time the issues occurred. RUEI allowed the analysts to see exactly what the executive had experienced; the pattern of 7 successful updates followed by a failure narrowed the focus to the most likely cause of the problem.

Case 3: Obsolete ASLM Transactions Causing High Error Rates

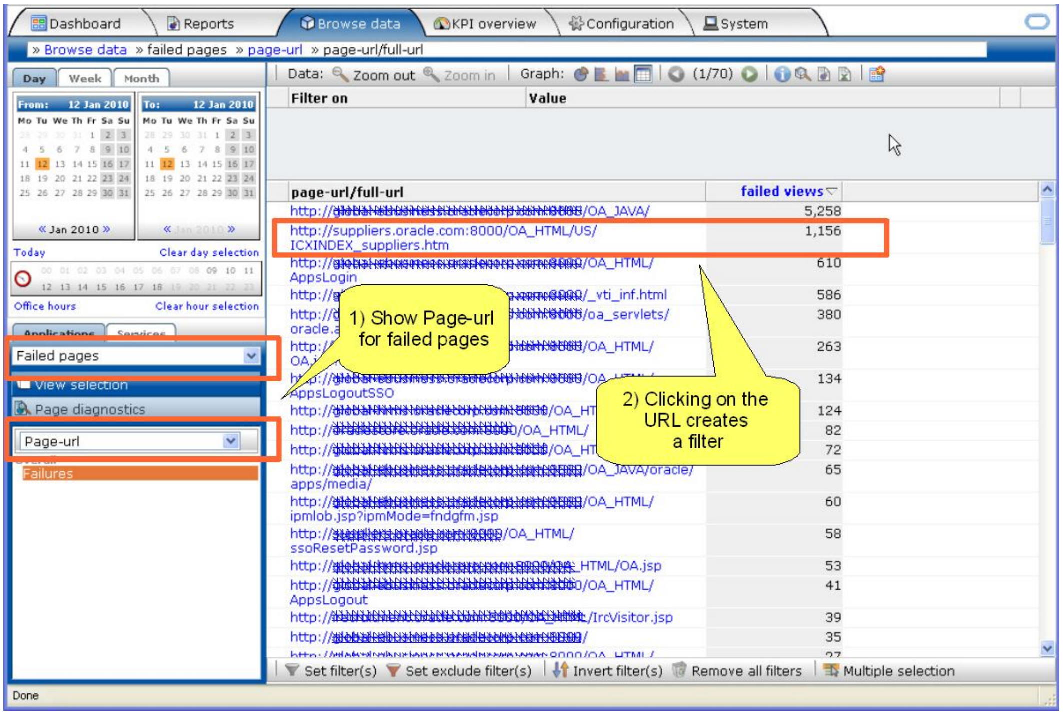
RUEI's Dashboard provides a high-level view of the overall health and welfare of the environment with areas for volume, performance, top applications and top errors.

A brief view of the dashboard for the GSI environment surprised analysts. It revealed that an external- facing Oracle website was experiencing significant errors, while the EM ASLM tools that had been set up to monitor its health continued to indicate that everything was fine.



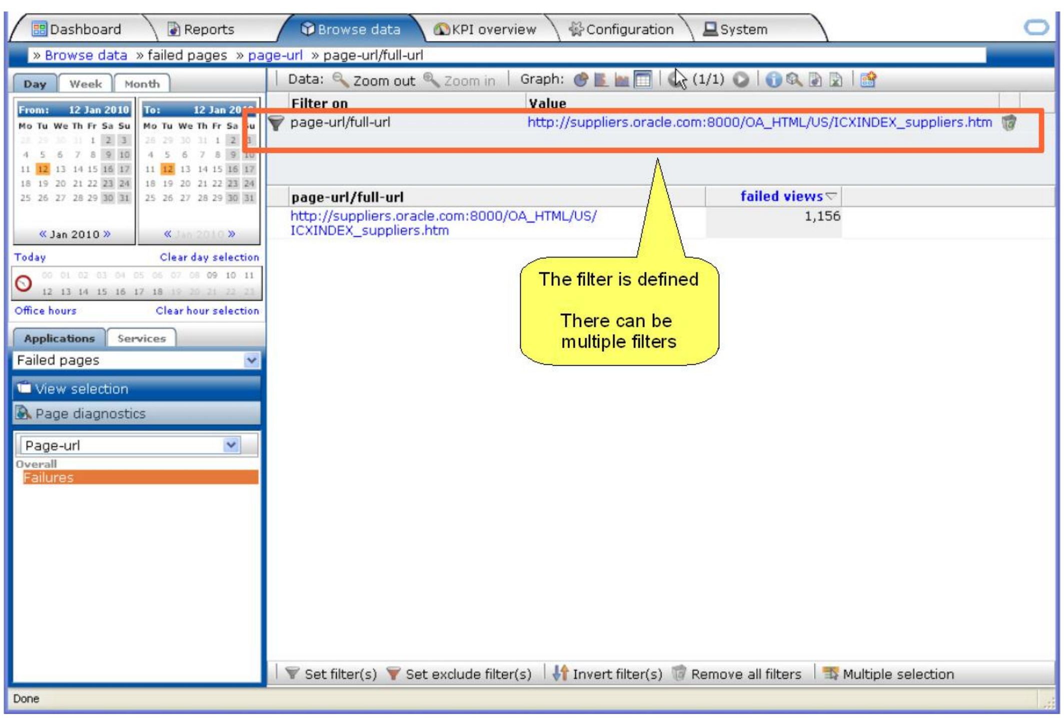
RUEI Dashboard / Top Failed Pages

The investigation began by utilizing RUEI to isolate the URL metrics. A filter was created by drilling down to the URL by navigating to the Browse Data tab, then showing the URLs under the category *Failed Pages* and subcategory *Page-URL*.



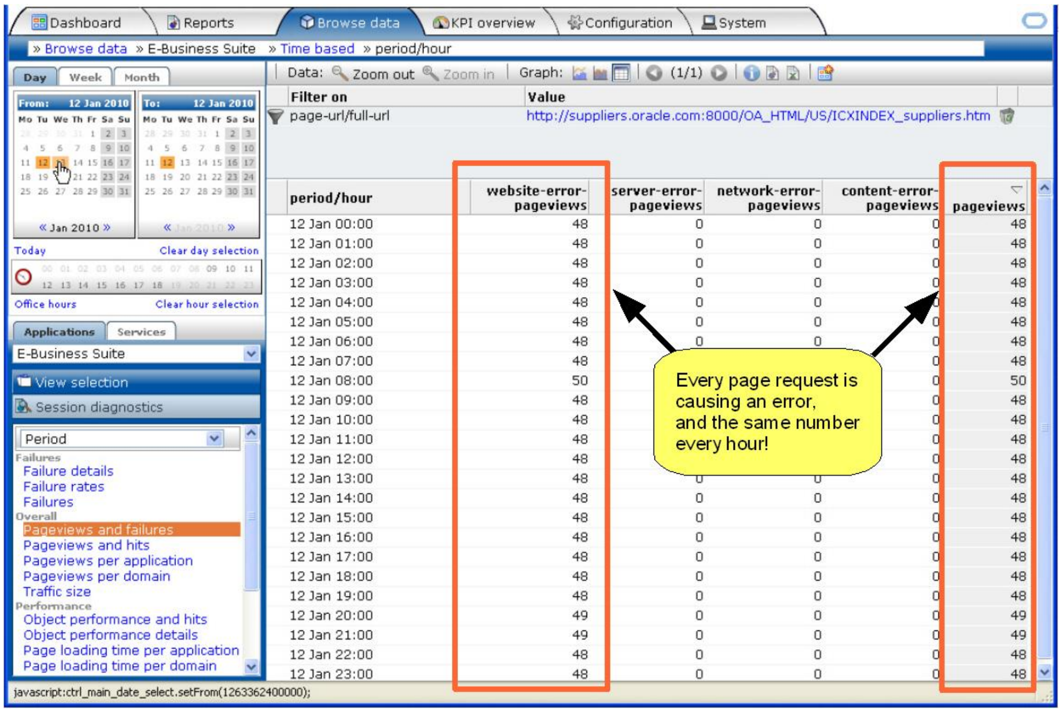
Creating a filter

The filter was created by double clicking on the URL.



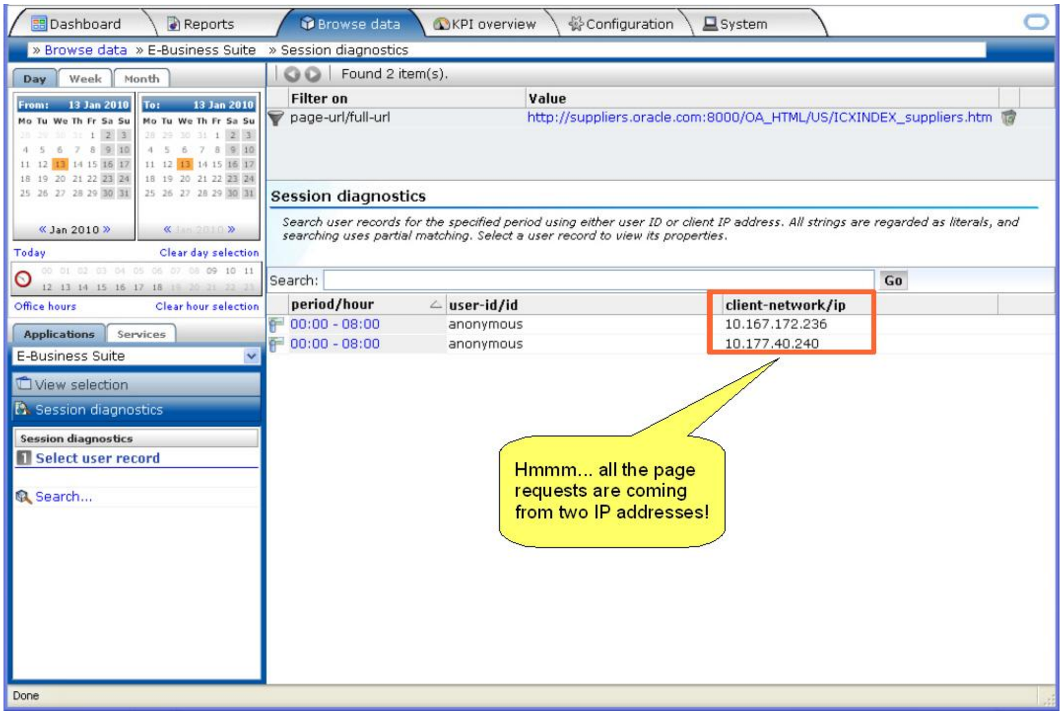
Defined Filters

With the filter in place, the URL was isolated and the metrics could be easily examined. In order to determine if the errors were consistent or intermittent, page views and failures were examined over a period of time. It became clear that each page view for the URL was failing with a website error, and there was a distinct pattern to the failures – 48 per hour. The plot was thickening!



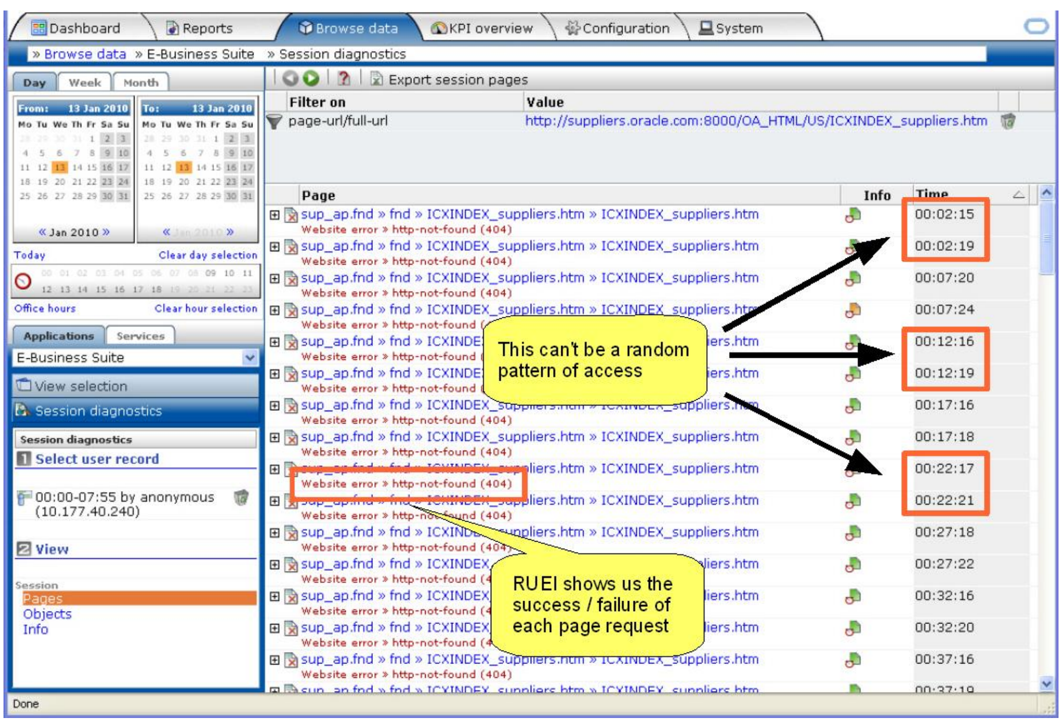
Period Pageviews and Hits

RUEI's Session Diagnostics feature was used to study individual sessions that had attempted to access the page. The results were intriguing; the sessions accessing the page originated from just two IP addresses.



Session IP Addresses

By simply clicking on one of the sessions, all of the relevant information was displayed – which pages were accessed at what time, and the responses from the web server. It was clear that the sessions were accessing the page, getting a http-not-found (404) error, trying it again five seconds later, and repeating the process five minutes later.



Session Pages by Time

The IP addresses were recognized as originating from EM ASLM beacons that had been set up to periodically check the health of the website from different geographic locations. The URL had been made obsolete; although there had been no complaints, anyone attempting to access the page was presented with a nicely formatted page directing them to the new page, with an underlying return code of http-not-found. The beacons continued to run and get positive page performance results, but were not checking the return code.

The group responsible for monitoring the environment had not been informed of the obsolescence of the original URL, and therefore had not updated the beacons. The beacons continued to run, providing false indications that the external site was operating properly, and consuming system resources on the web servers and beacon sites.

RUEI quickly revealed the problem, which would have been extremely difficult to do with other tools, allowing the ASLM beacons to be updated and to function as originally intended. It also highlighted a broader interdepartmental issue, prompting an improvement in communications.

Conclusion

Oracle's Global Single Instance sustains ROI by increasing efficiency. Effective use of its applications maintains revenue, productivity and satisfaction. Complete analysis of GSI end-user experience is critical, and traditional tools took narrow views. The deployment of Real User Experience Insight provided a powerfully proactive monitoring solution. Once RUEI was installed and running, analysts had a real-world view of user experience across the technology stack. They were able to drill down to specific sessions, examine data in a variety of ways, quickly resolve long-standing issues that had been frustrating users, and discover impending issues that had not yet had any impact. RUEI's comprehensive stare at applications is unrivaled, and helps maintain the high levels of performance and positive user experience that are fundamental to the health of Oracle and the clients it serves.

ORACLE®

White Paper Title
February 2010
Author: Dave Johnson
Contributing Author: Anne Grahn
Implementation Team: Mark Jamieson, Padmanabha Iyengar, Fun Cheung, Kevin Ng, Tracy Stollberg

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Oracle is committed to developing practices and products that help protect the environment