UMS Application

Quick Start Guide

Venkatraj Subramanian

CONTENTS

DOCUMENT CONTROL PAGE

Revision History

Version No	Revision Date	Nature of Change
1.0	19th October 2014	
1.1	27th November 2014	
1.2	10th January 2015	
1.3	05th March 2015	
1.4	06th June 2015	Updated refreshtree command
1.5	26th June 2015	New situation added for missing ep in business_hour UMS_BUSHOURMISS_CRI
1.6	18th August 2015	Updated new menu option in GUI named “Report Viewer” provides facility to view and download historical data in excel, pdf and other formats.

1. Purpose

Use this document to understand UMS application architecture and to follow the installation, configuration and administration procedures choosing Linux as platform.

2. Scope

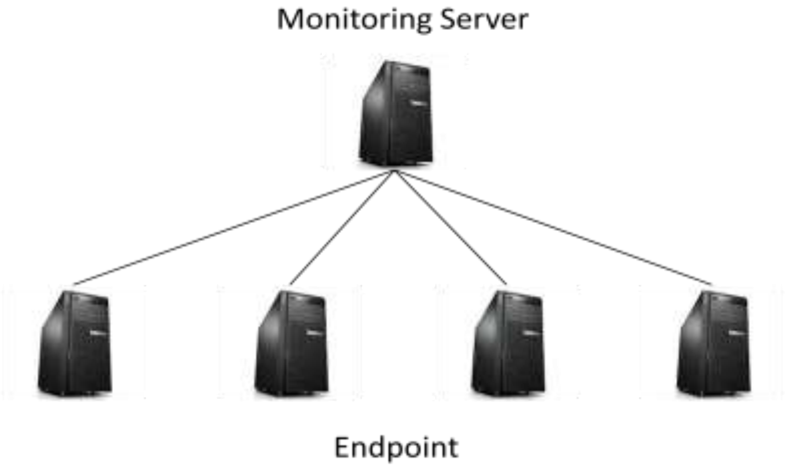
The main aim of UMS is to monitor the performance and availability of unix & linux servers. Proactively monitoring of system resources helps to detect potential problems, automatically generation of alerts and notification to the authorized user through E-mail and SMS. By identifying issues early, UMS system enables rapid fixes problems before any performance bottleneck is noticed.

3. Solution Infrastructure

Operating System	Products Installed
Red Hat Enterprise Linux Server release 6.3 64-bit	• java version "1.6.0_24" • Apache Tomcat/6.0.32 • Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production

4. Solution Architecture

The main purpose of UMS application is to monitor the performance and availability of unix and linux servers. Monitoring system is based on the Client - Server architecture.



5. Components of the monitoring architecture

This section describes the architecture of the UMS application. UMS application use a set of service components, they are mentioned below

1. Endpoint
2. Upload Engine
3. Core Engine
4. Summarization and Pruning component
5. GUI component

Endpoint:

Endpoint component is installed on the system where the resources need to be monitored. The installation is done by deploying monitoring scripts on the endpoints. These monitoring scripts collect data from devices and upload the data to Monitoring Server using SCP service.

Upload Engine:

The purpose of Upload engine to store the monitoring data received from the endpoint to the backend database.

Core Engine:

The purpose of the Core engine is to analyze the data in the database and raise notification to Netcool Server. The core engine is rule based engines which raise notifications on the basis of alert criteria defined in the rule file. The rule file contains the set of alert criteria properties files names; these properties files are also called situations which defines the alert criteria. Duplication of events is defined by a unique Identifier attribute and is defined core level by concatenating several other attributes to form an Identifier key. The Core engine then uses this unique field to compare events and match those with the same value automatically. This allows for similar events to be dropped.

Summarization and Pruning component:

Summarization is happening by running "/ums/install/summarisation.sh" script. The script runs every Monday and creates summarized historical data for last week. The summarization script creates summarized tables of type daily business hour data and non-bussines hour data summarization. For example, AIXMEM_DB (Summarized daily business hour data derived from AIXMEM table), AIXMEM_DNB (Summarized daily non-business hour data derived from AIXMEM table)

GUI Component:

GUI Component provides the core presentation layer for retrieval, manipulation, analysis, and pre-formatting of monitoring data retrieved from database. GUI component is deployed in Apache Tomcat.

6. Hardware and Software Configuration

Hardware Configuration	
Processor Type	Intel(R) Xeon(R) CPU E7- 4850 @ 2.00GHz
Number of Cores	4
Usable Memory	3 GB
Software Configuration	
Operating System	Red Hat Enterprise Linux Server release 6.3 64-bit
Java	java version "1.6.0_24"
Oracle Database	Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
Apache Tomcat	Apache Tomcat/6.0.32

7. Create Operating Systems user IDs and groups

Create the following Operating System User IDs and Groups on the server where the application and database is to be installed.

Groups	
oinstall	Any user supporting the Oracle database should be a member of this group.
IDs	
ID ums	This id will be used for all UMS installations and upgrades.
ID ora11	This id will be used for all oracle installations and upgrades.

8. File Systems & Directory Permissions

The following file systems should be created /mounted and required permissions granted as defined below.

FILE SYSTEM	**SIZE**	**Directory Permissions**	**Owner**
/ums	20GB	755	ums.ums
/oracle	18GB	755	ora11.oinstall
/umsdata	30GB	755	ora11.oinstall
/tmp	150MB Free Space		
/home	150MB Free Space		

9. Profile change

Below mentioned changes applies for application user.

```
export CORE_HOME=/ums/install/core
export UPLOAD_HOME=/ums/install/upload

export ORACLE_HOME=/oracle/ora11/app/ora11/product/11.2.0/dbhome_1
export ORACLE_SID=UMSDB
export ORACLE_UNQNAME=UMSDB

alias sqldev="/oracle/ora11/app/ora11/product/11.2.0/dbhome_1/sqldeveloper/sqldeveloper/bin/sqldeveloper &"
PATH=$PATH:$HOME/bin:$ORACLE_HOME/bin:/ums/install/UMSEIF:/usr/java/jdk1.6.0_24/bin;export PATH;
```

Any modification in $CORE_HOME variable needs update in $CORE_HOME/props/importprop.properties

10. Directory Structure of Core and upload engine

<table>
<tr><th>Core Engine</th><th>Upload Engine</th></tr>
<tr><td></td><td></td></tr>
<tr><td>$CORE_HOME</td><td>$UPLOAD_HOME</td></tr>
<tr><td>|_ rules</td><td>|_ bin</td></tr>
<tr><td>| |_ ums_engine.rules</td><td>| |_ Server_infobean.java</td></tr>
<tr><td>|_ appjar</td><td>| |_ Insert_records.java</td></tr>
<tr><td>| |_ ojdbc6.jar</td><td>| |_ Info_datetime.java</td></tr>
<tr><td>| |_ com.jcraft.jsch_0.1.31.jar</td><td>| |_ Database_connection.java</td></tr>
<tr><td>| |_ jconn3.jar</td><td>| |_ UMSUpload.java</td></tr>
<tr><td>|_ bin</td><td>| |_ UMSUpload.class</td></tr>
<tr><td>| |_ Notify_eif.java</td><td>| |_ UMSUpload$1.class</td></tr>
<tr><td>| |_ Info_datetime.java</td><td>| |_ Server_infobean.class</td></tr>
<tr><td>| |_ Database_connection.java</td><td>| |_ Insert_records.class</td></tr>
<tr><td>| |_ Core_engine.java</td><td>| |_ Info_datetime.class</td></tr>
<tr><td>| |_ Core_engine.class</td><td>| |_ Database_connection.class</td></tr>
<tr><td>| |_ Info_datetime.class</td><td>|_ appjar</td></tr>
<tr><td>| |_ Database_connection.class</td><td>| |_ ojdbc6.jar</td></tr>
<tr><td>|_ eif</td><td>| |_ com.jcraft.jsch_0.1.31.jar</td></tr>
<tr><td>| |_ umspost</td><td>|_ props</td></tr>
<tr><td>| |_ eif.conf</td><td>| |_ uploadprop.properties</td></tr>
<tr><td>|_ props</td><td>|_ run.sh</td></tr>
<tr><td>| |_ importprop.properties</td><td>|_ compile.sh</td></tr>
<tr><td>| |_ sybconfig.properties</td><td>|_ Readme</td></tr>
</table>

```
|_ situations
|   |_ AIX_....._CRITICAL.properties
|   |_...
|   |_...
|_ Readme
|_ run.sh
|_ compile.sh
|_ sitclear.sh
|_ runsitclear.sh
```

11. Understanding De-duplication

De-duplication of events is defined by a unique Identifier attribute and is defined core level by concatenating several other attributes to form an Identifier key. The Core engine then uses this unique field to compare events and match those with the same value automatically. This allows for similar events to be dropped. To understand the Identifier, refer the below code

```
BEGIN
IF sit_alert_type=”STD”
THEN
        fin_identifier = sit_alert_name + ":" + fin_hostname + ":" + sit_class_name + ":" +
        fin_sit_display_item3 + ":" + fin_sit_display_item2 + ":" + sit_display_item1 + ":" +
        sit_alert_type;
ELSE
        fin_identifier = sit_alert_name + ":" + fin_hostname + ":" + sit_class_name + ":" +
        rs_sit_query.getString("SERVER_DATE") + ":" + fin_sit_display_item3 + ":" +
        fin_sit_display_item2 + ":" + sit_display_item1 + ":" + sit_alert_type;
FI
END
```

Where:
fin_identifier: Denotes the Identifier.
sit_alert_name: Denotes the Situation alert name. Used for framing identifier and message attribute.
sitclassname: Alerts are grouped based on class name. Unique class name is mentioned. Used as Identifier attribute.
sit_alert_type: Standard or baseline situation. Possible options STD – Standard, BLLW – Base Line Last Week, BLLM – Base Line Last Month. Also used as Identifier attribute.
fin_hostname: Denotes the hostname of the machine.
sit_display_item1 (Static Display Item – SDI1): Used for framing identifier and message attribute. Static value is assigned; mostly alert criteria attribute is assigned as Static Display Item.
fin_sit_display_item2 (Dynamic Display Item – DDI2): Used for framing identifier and message attribute. Default value for Dynamic Display Item is “HOSTNAME”. If SDI1 doesn’t suffice the needs for creating Identifier then use DDI2.
fin_sit_display_item3 (Dynamic Display Item – DDI3): Used for framing identifier and message attribute. Default value for Dynamic Display Item is “HOSTNAME”. If SDI1 and DDI2 doesn’t suffice the needs for creating Identifier then use DDI3

12. Understanding Situation and Rule

There are two types of situations, Standard situation and Base Line situation. In Standard situation alert criteria are predefined static. For Base Line situation alert criteria is derived from standard deviation of historical data. For BLLW (Base Line Last Week) type situations alert criteria derived from the last seven days data and for BLLM (Base Line Last Month) type situations alert criteria derived from last month last week data. Default location of situations is $CORE_HOME/situations

Below mentioned all attributed needs to be used for standard or Base line alerts.

Alert Name (sitalertname): Denotes the Situation alert name. Used for framing identifier and message attribute.
Alert Type (sittype): Denotes Standard or baseline situation. Possible options STD – Standard, BLLW – Base Line Last Week, BLLM – Base Line Last Month. Also used as Identifier attribute.
Alert Nature (sitnature): Defines the nature of the situation.
Class Name (sitclassname): Alerts are grouped based on class name. Unique class name is mentioned. Used as Identifier attribute.
SQL Query (sitsql): SQL query defines the alert criteria.
Formula (sitformula): For Standard situation denotes the alert criteria attribute with threshold value and for base line situation denotes only the alert criteria attribute. Formula definition is used in framing the message attribute.
Static Display Item – SDI1 (sitdisplayitem1): Used for framing identifier and message attribute. Static value is assigned; mostly alert criteria attribute is assigned as Static Display Item.
Dynamic Display Item – DDI2 (sitdisplayitem2): Used for framing identifier and message attribute. Default value for Dynamic Display Item is “HOSTNAME”. If SDI1 doesn’t suffice the needs for creating Identifier then use DDI2.
Dynamic Display Item – DDI3 (sitdisplayitem3): Used for framing identifier and message attribute. Default value for Dynamic Display Item is “HOSTNAME”. If SDI1 and DDI2 don’t suffice the needs for creating Identifier then use DDI3
Additional Attribute list (sitatrlist): In addition to default attributes, the attributes defined in AAL also embedded in event. Attributes list should be comma separated with no spacing.
Alert Message (sitmsg): Denotes the alert message. If no custom message is required mark as DEFAULT
Alert Description (sitalertdesc): Defines the description of the alert.

The core engine is rule based engines which raise notifications on the basis of alert criteria (situations) defined in the rule file. Default location of rule file is $CORE_HOME/rules/ ums_engine.rules
Any modification in the situation needs to change the status of the open events to close in UMS and Netcool Omnibus to avoid false alert. $CORE_HOME/runsitclear.sh Script is used to handle this scenario events and the same is configured in cron. The nature of events is categorized as Sampled and Pure, the events generated by pure situations are closed when consecutive true samples are true.

13. Message attribute definition

To understand the message attribute, refer the below code

```
BEGIN
IF sitmsg="DEFAULT"
    IF sit_alert_type=”STD”
    THEN
        IF   sit_display_item2=HOSTNAME && sit_display_item3= HOSTNAME
        THEN
            exe_sit_msg = sit_alert_name + "[( " + sit_formula + " ) ON " + fin_hostname + "
            ( " + sit_display_item1 + "=" + rs_sit_query.getString(sit_display_item1) + " )]";
        FI

        IF  sit_display_item2!= HOSTNAME && sit_display_item3= HOSTNAME
        THEN
            exe_sit_msg = sit_alert_name + "[( " + sit_formula + " ) ON " + fin_hostname + "
            ( " + sit_display_item1 + "=" + rs_sit_query.getString(sit_display_item1) + " " +
            sit_display_item2 + "=" + rs_sit_query.getString(sit_display_item2) + " )]";
        FI

        IF  sit_display_item2!=HOSTNAME && sit_display_item3!=HOSTNAME
        THEN
            IF  sit_display_item1=VAL(sit_display_item3)
            THEN
                exe_sit_msg = sit_alert_name + "[( " + sit_formula + " ) ON " +
                fin_hostname + " ( "+ sit_display_item1 + "=" +
                rs_sit_query.getString(sit_display_item1) + "  " + sit_display_item2 + "="
                + rs_sit_query.getString(sit_display_item2) + " )]";
            ELSE
                exe_sit_msg = sit_alert_name + "[( " + sit_formula + " ) ON " +
                fin_hostname + " ( "+ sit_display_item1 + "=" + rs_sit_query.getString
                (sit_display_item1) + " " +sit_display_item2 + "=" +
                rs_sit_query.getString(sit_display_item2) + " " +
                sit_display_item3 + "=" + rs_sit_query.getString
                (sit_display_item3) + " )]";
            FI
    ELSE
        String max_bl = "MAX_BL_" + sit_formula;
        String min_bl = "MIN_BL_" + sit_formula;

        IF sit_display_item2=HOSTNAME
        THEN
            exe_sit_msg = sit_alert_name + "[( Baseline check not between " +
            rs_sit_query.getString(min_bl) + " and " + rs_sit_query.getString(max_bl) + " )
            ON " + fin_hostname + " ( " + sit_display_item1 + "=" +
            rs_sit_query.getString(sit_display_item1) + " )]";
        ELSE
            exe_sit_msg = sit_alert_name + "[( Baseline check not between " +
            rs_sit_query.getString(min_bl) + " and " + rs_sit_query.getString(max_bl) + " )
            ON " + fin_hostname + " ( " + sit_display_item1 + "=" +
            rs_sit_query.getString(sit_display_item1) + " " + sit_display_item2 + "=" +
            rs_sit_query.getString(sit_display_item2) + " )]";
    FI
ELSE
    sitmsg = rb.getString("sitmsg");
    Pattern pattern = Pattern.compile("%(.*?)%");
    Matcher matcher = pattern.matcher(sitmsg);
    String msg_per_tmp_value;
    WHILE (matcher.find())
        String msg_per_value = matcher.group(1);
        msg_per_tmp_value = sitmsg.replace("%" + msg_per_value + "%",
        rs_sit_query.getString(msg_per_value));
        sitmsg = msg_per_tmp_value;
    DONE
    exe_sit_msg = sitmsg;
FI
END
```

Where:
exe_sit_msg: Denotes the message attribute.
sit_alert_name: Denotes the Situation alert name. Used for framing identifier and message attribute.
sit_alert_type: Standard or baseline situation. Possible options STD – Standard, BLLW – Base Line Last Week, BLLM – Base Line Last Month. Also used as Identifier attribute.
sit_formula: For Standard situation denotes the alert criteria attribute with threshold value and for base line situation denotes only the alert criteria attribute. Formula definition is used in framing the message attribute.
fin_hostname: Denotes the hostname of the machine.
Static Display Item – SDI1 (sitdisplayitem1): Used for framing identifier and message attribute. Static value is assigned; mostly alert criteria attribute is assigned as Static Display Item.
Dynamic Display Item – DDI2 (sitdisplayitem2): Used for framing identifier and message attribute. Default value for Dynamic Display Item is “HOSTNAME”. If SDI1 doesn’t suffice the needs for creating Identifier then use DDI2.
Dynamic Display Item – DDI3 (sitdisplayitem3): Used for framing identifier and message attribute. Default value for Dynamic Display Item is “HOSTNAME”. If SDI1 and DDI2 don’t suffice the needs for creating Identifier then use DDI3
Alert Message (sitmsg): Denotes the alert message. If no custom message is required mark as DEFAULT
rs_sit_query.getString(sit_display_item1): Value derived for sit_display_item1.
rs_sit_query.getString(sit_display_item2): Value derived for sit_display_item2.
rs_sit_query.getString(sit_display_item3): Value derived for sit_display_item3.

rs_sit_query.getString(min_bl): Value derived for MIN_BL_" + sit_formula”

rs_sit_query.getString(max_bl): Value derived for MAX_BL_" + sit_formula”

14. Pre-defined Situations

14.1. AIX_PHY_MEM_CRI

AIX_PHY_MEM_CRI is a standard situation and fire when the “USED_PHYSICAL_MEMORY_ PERC” greater than or equal to threshold value. The default attribute of “sitdisplayitem2” is not changed.

```
sitalertname=AIX_PHY_MEM_CRI
sittype=STD
sitnature=SAMPLED
sitclassname=AIXMEM
sitsql=select HOSTNAME, SERVER_DATE_TIME, TOTAL_PHYSICAL_MEMORY_MB,
USED_PHYSICAL_MEMORY_MB, USED_PHYSICAL_MEMORY_PERC,
FREE_PHYSICAL_MEMORY_MB,  TOTAL_PAGING_SPACE_MB,
USED_PAGING_SPACE_MB, FREE_PAGING_SPACE_MB, DBWRITETIME from aixmem where
USED_PHYSICAL_MEMORY_PERC >= 95 and (hostname,server_date_time) in (select hostname,
max(server_date_time) from aixmem group by hostname)
sitformula=USED_PHYSICAL_MEMORY_PERC >= 95
sitdisplayitem1=USED_PHYSICAL_MEMORY_PERC
sitdisplayitem2=HOSTNAME
sitdisplayitem3=HOSTNAME
sitatrlist=SERVER_DATE_TIME,TOTAL_PHYSICAL_MEMORY_MB,USED_PHYSICAL_MEMO
RY_MB,USED_PHYSICAL_MEMORY_PERC
sitmsg=DEFAULT
sitalertdesc=Physical memory utilization has reached the threshold value
```

14.2. AIX_PAGING_CRI

AIX_PAGING_CRI is a standard situation and fire when the “USED_PAGING_SPACE_PERC” greater than or equal to threshold value. The default attribute of “sitdisplayitem2” is not changed.

```
sitalertname=AIX_PAGING_CRI
sittype=STD
sitnature=SAMPLED
sitclassname=AIXMEM
sitsql=select HOSTNAME, SERVER_DATE_TIME, TOTAL_PHYSICAL_MEMORY_MB,
USED_PHYSICAL_MEMORY_MB, USED_PHYSICAL_MEMORY_PERC, FREE_PHYSICAL_
MEMORY_MB,  TOTAL_PAGING_SPACE_MB, USED_PAGING_SPACE_MB,
USED_PAGING_SPACE_PERC, FREE_PAGING_SPACE_MB, DBWRITETIME from aixmem
where USED_PAGING_SPACE_PERC >= 5 and (hostname,server_date_time) in (select hostname,
max(server_date_time) from aixmem group by hostname)
sitformula=USED_PAGING_SPACE_PERC >= 5
```

```
sitdisplayitem1=USED_PAGING_SPACE_PERC
sitdisplayitem2=HOSTNAME
sitdisplayitem3=HOSTNAME
sitatrlist=SERVER_DATE_TIME,TOTAL_PHYSICAL_MEMORY_MB,USED_PHYSICAL_MEMORY_MB,USED_PHYSICAL_MEMORY_PERC
sitmsg=DEFAULT
sitalertdesc=Paging space utilization has reached the threshold value
```

14.3. AIX_FILESYSTEM_CRI

AIX_FILESYSTEM_CRI is a standard situation and fire when the “FILESYSTEM_USED_PERC” greater than or equal to threshold value. The default attribute of “sitdisplayitem2” is changed to FILESYSTEM as it is required to define unique identifier.

```
sitalertname=AIX_FILESYSTEM_CRI
sittype=STD
sitnature=SAMPLED
sitclassname=AIXFS
sitsql=select HOSTNAME, SERVER_DATE_TIME, FILESYSTEM, FILESYSTEM_TOTAL_MB, FILESYSTEM_USED_MB, FILESYSTEM_USED_PERC, FILESYSTEM_FREE_MB, FILESYSTEM_FREE_PERC, DBWRITETIME from aixfs where FILESYSTEM_USED_PERC >= 95 and (hostname,server_date_time) in (select hostname, max(server_date_time) from aixfs group by hostname)
sitformula=FILESYSTEM_USED_PERC >= 95
sitdisplayitem1=FILESYSTEM_USED_PERC
sitdisplayitem2=FILESYSTEM
sitdisplayitem3=HOSTNAME
sitatrlist=SERVER_DATE_TIME,FILESYSTEM,FILESYSTEM_TOTAL_MB,FILESYSTEM_USED_MB,FILESYSTEM_USED_PERC
sitmsg=DEFAULT
sitalertdesc=Filesystem utilization has reached the threshold value
```

14.4. AIX_CPU_CAPPED_CRI

AIX_CPU_CAPPED_CRI is a standard situation for NON-FHA servers and fire when the “CPU_USED_PERC” greater than or equal to threshold value. The default attribute of “sitdisplayitem2” is not changed.

```
sitalertname=AIX_CPU_CAPPED_CRI
sittype=STD
sitnature=SAMPLED
sitclassname=AIXCPU
sitsql=select a.HOSTNAME, a.SERVER_DATE_TIME, a.CPU_USED_PERC, a.CPU_IDLE_PERC, a.ACTIVE_CPU_COUNT, a.DBWRITETIME from aixcpu a, aixsystem b where a.CPU_USED_PERC >= 80 and (a.hostname, a.server_date_time) in (select hostname, max(server_date_time) from aixcpu group by hostname) and (b.hostname, b.server_date_time) in (select
hostname, max(server_date_time) from aixsystem group by hostname) and a.hostname = b.hostname
and b.aix_mode like 'Capped%'
sitformula=CPU_USED_PERC >= 80
sitdisplayitem1=CPU_USED_PERC
sitdisplayitem2=HOSTNAME
sitdisplayitem3=HOSTNAME
sitatrlist=SERVER_DATE_TIME,CPU_USED_PERC,CPU_IDLE_PERC,ACTIVE_CPU_COUNT
sitmsg=DEFAULT
sitalertdesc=CPU utilization has reached the threshold value
```

14.5. AIX_CPU_UNCAPPED_CRI

AIX_CPU_UNCAPPED_CRI is a standard situation for FHA servers and fire when the (ACTIVE_CPU_COUNT*100) / (ENTITLED_CAPACITY) greater than or equal to threshold value. The default attribute of “sitdisplayitem2” is not changed.

```
sitalertname=AIX_CPU_UNCAPPED_CRI
sittype=STD
sitnature=SAMPLED
sitclassname=AIXCPU
sitsql=select a.HOSTNAME, a.SERVER_DATE_TIME,
a.ACTIVE_CPU_COUNT*100/b.ENTITLED_CAPACITY as ECPU_USED_PERC ,
a.DBWRITETIME from aixcpu a, aixsystem b where
a.ACTIVE_CPU_COUNT*100/b.ENTITLED_CAPACITY >= 80 and (a.hostname,
a.server_date_time) in (select hostname, max(server_date_time) from aixcpu group by hostname) and
(b.hostname, b.server_date_time) in (select hostname, max(server_date_time) from aixsystem group by
hostname) and a.hostname = b.hostname and b.aix_mode like 'Uncapped%' and b.type like 'Shared%'
sitformula=ECPU_USED_PERC>=80
sitdisplayitem1=CPU_USED_PERC
sitdisplayitem2=HOSTNAME
sitdisplayitem3=HOSTNAME
sitatrlist=SERVER_DATE_TIME,CPU_USED_PERC,CPU_IDLE_PERC,ACTIVE_CPU_COUNT
sitmsg=DEFAULT
sitalertdesc=CPU utilization has reached the threshold value
```

14.6. AIX_DISKIO_CRI

AIX_DISKIO_CRI is a standard situation and fire when the “IO_PERCENTAGE” greater than or equal to threshold value. The default attribute of “sitdisplayitem2” is set to HDISK_NAME as it is required to define unique identifier.

```
sitalertname=AIX_DISKIO_CRI
sittype=STD
sitnature=SAMPLED
sitclassname= AIXIOSTAT
sitsql=select HOSTNAME, SERVER_DATE_TIME, HDISK_NAME, IO_PERCENTAGE,
IO_REQUEST_PER_SECOND, DBWRITETIME from aixiostat where IO_PERCENTAGE >= 2 and
(hostname,server_date_time) in (select hostname, max(server_date_time) from aixiostat group by
hostname)
sitformula=IO_PERCENTAGE >= 20
sitdisplayitem1=IO_PERCENTAGE
sitdisplayitem2=HDISK_NAME
sitdisplayitem3=HOSTNAME
sitatrlist=SERVER_DATE_TIME,IO_REQUEST_PER_SECOND
sitmsg=DEFAULT
sitalertdesc=Disk IO utilization has reached the threshold value
```

14.7. AIX_FDRL_CRI

AIX_FDRL_CRI is a standard situation and fire when failed root login occurs. The default attribute of "sitdisplayitem2" is set to LOGIN_USER and "sitdisplayitem3" is set to DIRECT_FAILED_LOGIN_DATE_TIME as it is required to define unique identifier. This situation requires closer of events using housing keeping rules.

```
sitalertname=AIX_FDRL_CRI
sittype=STD
sitnature=PURE
sitclassname= AIXFDRL
sitsql=select
HOSTNAME,SERVER_DATE_TIME,LOGIN_USER,PROTOCOL,DIRECT_FAILED_LOGIN_DAT
E_TIME,IP from AIXFDRL where to_char(DBWRITETIME, 'dd/mm/yy')= to_char(SYSDATE,
'dd/mm/yy') group by
hostname,SERVER_DATE_TIME,LOGIN_USER,PROTOCOL,DIRECT_FAILED_LOGIN_DATE_T
IME,IP
sitformula=FDRL
sitdisplayitem1=IP
sitdisplayitem2=LOGIN_USER
sitdisplayitem3=DIRECT_FAILED_LOGIN_DATE_TIME
sitatrlist=SERVER_DATE_TIME,PROTOCOL,DIRECT_FAILED_LOGIN_DATE_TIME,IP
sitmsg=DEFAULT
sitalertdesc=Failed Direct ROOT Login
```

14.8. AIX_FSURL_CRI

AIX_FSURL_CRI is a standard situation and fire when failed su root login occurs. The default attribute of "sitdisplayitem2" is set to TO_USER and "sitdisplayitem3" is set to SU_FAILED_LOGIN_DATE_TIME as it is required to define unique identifier. This situation requires closer of events using housing keeping rules.

```
sitalertname=UMS_AIX_FSURL_CRITICAL
sittype=STD
sitnature=PURE
```

```
sitclassname=AIXFSURL
sitsql=select DISTINCT a.hostname,a.SERVER_DATE_TIME, b.IP, a.TERMINAL, B.LOGIN_USER,
a.FROM_USER, a.TO_USER, a.SU_FAILED_LOGIN_DATE_TIME, b.FROM_DATE_TIME,
b.TO_DATE_TIME from aixfsurl a ,AIXLASTLOG b where a.SU_STATUS = '-' and
to_char(a.SERVER_DATE_TIME, 'dd/mm/yy')= to_char(SYSDATE, 'dd/mm/yy') and
a.SU_FAILED_LOGIN_DATE_TIME between b.FROM_DATE_TIME and b.TO_DATE_TIME and
a.TERMINAL=b.TERMINAL and a.HOSTNAME=b.HOSTNAME
sitformula=FSURL
sitdisplayitem1=IP
sitdisplayitem2=TO_USER
sitdisplayitem3=SU_FAILED_LOGIN_DATE_TIME
sitatrlist=hostname,SERVER_DATE_TIME,IP,TERMINAL,LOGIN_USER,FROM_USER,TO_USER,
SU_FAILED_LOGIN_DATE_TIME,FROM_DATE_TIME,TO_DATE_TIME
sitmsg=DEFAULT
sitalertdesc=SU Failed ROOT Login
```

14.9. AIX_ERRPT_CRI

AIX_ERRPT_CRI is a standard situation and fire when the hardware error is occurs. The default attribute of “sitdisplayitem2” is set to ERROR_IDENTIFIER as it is required to define unique identifier. This situation requires closer of events using housing keeping rules.

```
sitalertname=AIX_ERRPT_CRI
sittype=STD
sitnature=PURE
sitclassname= AIXERRPT
sitsql=select HOSTNAME, SERVER_DATE_TIME, ERROR_IDENTIFIER, ERROR_TIMESTAMP,
ERROR_TYPE, ERROR_CLASS, ERROR_RESOURCE_NAME, ERROR_DESCRIPTION,
ERROR_REPEAT_COUNT, DBWRITETIME from aixerrpt where ERROR_CLASS in ('H','U') and
(hostname,server_date_time) in (select hostname, max(server_date_time) from aixerrpt group by
hostname) and to_char(SERVER_DATE_TIME, 'dd/mm/yy') = to_char(SYSDATE, 'dd/mm/yy')
sitformula=ERROR_CLASS='H','U'
sitdisplayitem1=ERROR_RESOURCE_NAME
sitdisplayitem2=ERROR_IDENTIFIER
sitdisplayitem3=ERROR_TIMESTAMP
sitatrlist=SERVER_DATE_TIME,ERROR_IDENTIFIER,ERROR_TIMESTAMP,ERROR_TYPE,ERR
OR_CLASS,ERROR_RESOURCE_NAME
sitmsg=UMS_AIX_ERRPT_CRITICAL: %ERROR_IDENTIFIER% %ERROR_TIMESTAMP%
%ERROR_TYPE% %ERROR_CLASS% %ERROR_RESOURCE_NAME%
%ERROR_DESCRIPTION% ::Error repeated %ERROR_REPEAT_COUNT% Times
sitalertdesc=Hardware error has occurred
```

14.10. AIX_PHY_MEM_CRI_BLLW

AIX_PHY_MEM_CRI_BLLW is a baseline situation and fire when the threshold is reached. This situation requires closer of events using housing keeping rules.

```
sitalertname=AIX_PHY_MEM_CRI_BLLW
sittype=BLLW
sitnature=SAMPLED
sitclassname=AIXMEM
sitsql=select a.HOSTNAME, a.SERVER_DATE_TIME, to_char(a.SERVER_DATE_TIME,
'dd/mm/yy') as SERVER_DATE, a.TOTAL_PHYSICAL_MEMORY_MB,
a.USED_PHYSICAL_MEMORY_MB, a.DBWRITETIME, (SELECT
AVG(USED_PHYSICAL_MEMORY_MB)+ STDDEV(USED_PHYSICAL_MEMORY_MB) FROM
AIXMEM WHERE SERVER_DATE_TIME between trunc(SYSDATE, 'IW')-7 AND trunc(SYSDATE,
'IW') AND to_char(to_date(SERVER_DATE_TIME), 'DAY') NOT IN (SELECT
NON_BUSINESS_DAY FROM NON_BUSINESS_DAY WHERE HOSTNAME=a.HOSTNAME)
AND to_char( SERVER_DATE_TIME, 'hh24miss') >= (SELECT BUSINESS_HOUR_START FROM
BUSINESS_HOUR WHERE HOSTNAME=a.HOSTNAME) AND to_char( SERVER_DATE_TIME,
'hh24miss') <= (SELECT BUSINESS_HOUR_END FROM BUSINESS_HOUR WHERE
HOSTNAME=a.HOSTNAME) AND  HOSTNAME=a.HOSTNAME GROUP BY HOSTNAME) as
MAX_BL_USED_PHYSICAL_MEMORY_MB, (SELECT
AVG(USED_PHYSICAL_MEMORY_MB)- STDDEV(USED_PHYSICAL_MEMORY_MB) FROM
AIXMEM WHERE SERVER_DATE_TIME between trunc(SYSDATE, 'IW')-7 AND trunc(SYSDATE,
'IW') AND to_char(to_date(SERVER_DATE_TIME), 'DAY') NOT IN (SELECT
NON_BUSINESS_DAY FROM NON_BUSINESS_DAY WHERE HOSTNAME=a.HOSTNAME)
AND to_char( SERVER_DATE_TIME, 'hh24miss') >= (SELECT BUSINESS_HOUR_START FROM
BUSINESS_HOUR WHERE HOSTNAME=a.HOSTNAME) AND to_char( SERVER_DATE_TIME,
'hh24miss') <= (SELECT BUSINESS_HOUR_END FROM BUSINESS_HOUR WHERE
HOSTNAME=a.HOSTNAME) AND  HOSTNAME=a.HOSTNAME GROUP BY HOSTNAME) as
MIN_BL_USED_PHYSICAL_MEMORY_MB from aixmem a where
a.USED_PHYSICAL_MEMORY_MB >=(SELECT  AVG(USED_PHYSICAL_MEMORY_MB)+
STDDEV(USED_PHYSICAL_MEMORY_MB) FROM AIXMEM WHERE SERVER_DATE_TIME
between trunc(SYSDATE, 'IW')-7 AND trunc(SYSDATE, 'IW') AND
to_char(to_date(SERVER_DATE_TIME), 'DAY') NOT IN (SELECT NON_BUSINESS_DAY FROM
NON_BUSINESS_DAY WHERE HOSTNAME=a.HOSTNAME) AND to_char(
SERVER_DATE_TIME, 'hh24miss') >= (SELECT BUSINESS_HOUR_START FROM
BUSINESS_HOUR WHERE HOSTNAME=a.HOSTNAME) AND to_char( SERVER_DATE_TIME,
'hh24miss') <= (SELECT BUSINESS_HOUR_END FROM BUSINESS_HOUR WHERE
HOSTNAME=a.HOSTNAME) AND  HOSTNAME=a.HOSTNAME GROUP BY HOSTNAME) and
(a.hostname,a.server_date_time) in (select hostname, max(server_date_time) from aixmem group by
hostname)
sitformula=USED_PHYSICAL_MEMORY_MB
sitdisplayitem1=USED_PHYSICAL_MEMORY_MB
sitdisplayitem2=HOSTNAME
sitdisplayitem3=HOSTNAME
sitatrlist=SERVER_DATE_TIME,TOTAL_PHYSICAL_MEMORY_MB,USED_PHYSICAL_MEMO
RY_MB
sitmsg=DEFAULT
sitalertdesc=This is a Base Line alert for Used Physical memory based on last seven days utilization
```

14.1. UMS_BUSHOURMISS_CRI

UMS_BUSHOURMISS_CRI is a standard situation and fire for missing endpoints in business_hour table.

```
sitalertname=UMS_BUSHOURMISS_CRI
sittype=STD
sitnature=SAMPLED
sitclassname=UMSSYS
sitsql=select 'LIBRA' as "HOSTNAME", to_char(sysdate, 'yyyy-mm-dd hh24:mi:ss') as
"SERVER_DATE_TIME",  "BUSINESS_HOUR_COUNT" from (select count(distinct(hostname)) as
"BUSINESS_HOUR_COUNT" from aixsystem where hostname not in (select hostname from
business_hour)) where "BUSINESS_HOUR_COUNT" >= 0
sitformula=BUSINESS_HOUR_COUNT>=0
sitdisplayitem1=BUSINESS_HOUR_COUNT
sitdisplayitem2=HOSTNAME
sitdisplayitem3=HOSTNAME
sitatrlist=SERVER_DATE_TIME
sitmsg=DEFAULT
sitalertdesc=Alert for missed ep in business_hour table
```

15. Installation Procedure

15.1. Install Endpoint

1. Create a user “ums” in client machine

   Characteristics of ums user

User NAME	: ums
Primary GROUP	: staff
HOME directory	: /IBM/UMS
User INFORMATION	: Unix Monitoring System

2. Copy the scripts from ORION (XXX.XXX.XXX.XXX) server and create outputs and backup directory under /IBM/UMS

   ```
   $ su - ums
   ums's Password:
   $
   $ cd /IBM/UMS
   $ scp -r ums@XXX.XXX.XXX.XXX:/IBM/UMS/scripts .
   $ mkdir outputs
   $ mkdir backup
   ```

3. Cron configuration for ums and root user is mentioned below, the same needs to be configured.

   Cron configuration for ums user

   ```
   0,10,20,30,40,50 * * * *  sh /IBM/UMS/scripts/mainmonitoring.sh > /dev/null 2>&1
   0 7 * * * sh /IBM/UMS/scripts/mainsysdetail.sh > /dev/null 2>&1
   ```

   Cron configuration for root user

   ```
   0 7 * * * sh /IBM/UMS/scripts/mainsysdetailroot.sh > /dev/null 2>&1
   ```

15.2. Install Upload Engine Component

1. Create Upload engine home directory and add entry in .bash_profile as mentioned below

   ```
   $export UPLOAD_HOME=<home directory>
   ```

2. Create below mentioned directories under home directory props, appjar, bin

3. Copy the uploadprop.properties files under props
   Copy the dependent jar files under appjar
   Copy the java and class files under bin

4. Remove below lines from UMSUpload.java, Database_connection.java, Info_datatime.java, Notify_eif.java

   ```
   package ums_system;
   ```

5. Changes to be made in UMSUpload.java

   ```
   * DISABLE   //ResourceBundle rb = ResourceBundle.getBundle("ums_system/uploadprop");
   ENABLE    ResourceBundle rb = ResourceBundle.getBundle("uploadprop");
   ```

6. Changes to be made in importprop.properties

   ```
   DISABLE
    #path=E:\\ums\\source_upload
   #desti_path=E:\\ums\\destination_backup\\
   ENABLE
    path=/ums/source_dir
   desti_path=/ums/destination_dir/
   ```

7. Make the necessary changes in compile.sh and run.sh and run the application

15.3. Install Core Engine Component

1. Create Core engine home directory and add entry in .bash_profile as mentioned below

   ```
   $export CORE_HOME=<home directory>
   ```

2. Create below mentioned directories under home directory rules, eif, props, situations, appjar, bin

   ```
   $mkdir $CORE_HOME/rules
   ```

3. Copy the situation files UMS_*.properties under situations directory
   Copy the ums_engine.rules files under rules
   Copy the importprop.properties files under props
   Copy the umspost, eif.conf files from backup under eif
   Copy the dependent jar files under appjar
   Copy the java, class, Readme, compile.sh and run.sh files under bin

4. Remove below lines from Core_engine.java, Database_connection.java, Info_datatime.java, Notify_eif.java

   ```
   package ums_core_engine;
   ```

5. Changes to be made in Core_engine.java

   ```
   * DISABLE   //rb = ResourceBundle.getBundle("ums_core_engine/importprop");
   ```

```
ENABLE    rb = ResourceBundle.getBundle("importprop");
* DISABLE   //rb = ResourceBundle.getBundle("ums_core_engine/importprop");
ENABLE    rb = ResourceBundle.getBundle("ums_core_engine/" + sit_name_process);
```

6. Changes to be made in importprop.properties

```
 DISABLE
 #rule_file_name=E:\\ums\\UMS_CORE_ENGINE_v3\\src\\ums_core_engine\\ums_engine.rules
ENABLE
rule_file_name=<home>/rules/ums_engine.rules
```

7. Make the necessary changes in compile.sh and run.sh and run the application

15.4. Install GUI Component

1. Extract the apache-tomcat-6.0.32.tar compressed file under $UMS_INSTALL directory
2. Stop the tomcat service

```
$ cd $UMS_INSTALL/ apache-tomcat-6.0.32/bin
$ shutdown.sh
or
$ catalina.sh stop
```

3. Copy the war (GUI application) file under $UMS_INSTALL/ apache-tomcat-6.0.32/webapps/
4. Start the tomcat service

```
$ cd $UMS_INSTALL/ apache-tomcat-6.0.32/bin
$ startup.sh
or
$ catalina.sh start
```

16. Process Flow for Upload Engine

1. UMSUpload

1.1.main method

a. Check for database connection, if fail then exit else proceed.
b. Create Resource Bundle to read the source path and destination path defined in uploadprop.properties file, the source path defines the location of metafile which contains the list of data files to be uploaded and destination path defines the destination location.
c. For each meta file in source path
   CALL **read_Meta_File** and pass meta file name (meta_file_name), source path (source_path) and destination path (desti_path) derived from the uploadprop.properties file. (Conn. 1.2.a.)
d. Processed meta file is moved to a backup directory (Conn. 1.1.c.)
e. Program exits once all the meta files are processed in the source directory.

1.2.read_Meta_File method

a. Read the content of meta file and slit the pipe separated data in to meaning full attributes.

b. For each data filename
c. CALL **set_File_Content** and pass data file name (file_to_read), number of records in data file (meta_file_count), table name (table_name) and destination path (desti_path). (Conn. 1.3.a.)

1.3.set_File_Content method

a. Create object for Java Bean (serverInfoBeansList)
b. Read the content of data file and slit the pipe separated data in to meaning full attributes.
c. The derived attributes are stored in Java Bean using set property.
d. CALL **Insert_Into_Database** and pass Java Bean containing the stored attributes (serverInfoBeansList) and table name (table_name) (Conn. 2.1.a.)
e. Once all the data files defined in the metafile is processed.
f. CALL **move_file** and pass data file name (file_name) and destination path (desti_path). (Conn. 1.4.a.)

1.4.move_file method

a. Processed data file is moved to a backup directory. (Conn. 1.1.c.)

2. Insert_records

2.1.Insert_Into_Database

a. Based on the derived table name the attributes stored in Java Bean are inserted in the database using sql query executed by prepared statement. The sql queries are defined in the Insert_Into_Database method corresponding to the table name.
b. After successful insert (Conn. 1.2.b)

3. Info_datetime

a. Formatted date variable is assigned to umsdate

17. Process Flow for Core Engine

1. Core_engine

1.1. main method

Note: refer Core Engine table structure

a. Create Resource Bundle to read the rule file name and eif location defined in importprop.properties file.
b. Read rule file to process the situations one by one.
c. Read the situations one by one for processing.
d. Attributes defined in the situations are assigned to variables.
e. Alert criteria query (sitsql defined in the situation) defined in the situation run on the database using prepared statement. Events matching those alert criteria stored in UMSTEMP table with the unique identifier key created.
f. Read the next situation to be processed i.e. (Conn. 1.1.c) once all situations are processed (Conn. 1.1.g)
g. Update the UMS_EVENT_INT_STATUS='N' for all records in UMSEVENT table; this is done to identify the closed events.

h. Sending open alert and updating the duplicate alert using identifier

```
BEGIN
FOR EACH RECORD IN UMSTEMP
     CHECK UMS_IDENTIFIER
     EXIST IN
     UMSEVENT
     WHERE
     UMS_EVENT_ALERT_STATUS=OPEN

    IF TRUE
        UPDATE
            UMS_EVENT_INT_STATUS=Y,
            UMS_EVENT_ORIGIN_TIMESTAMP_UPD,
            DBWRITETIME
        IN UMSEVENT
        WHERE
        UMS_EVENT_IDENTIFIER = <ID derived from select query>
        UMS_EVENT_ALERT_STATUS='OPEN'

    ELSE FALSE
        INSERT
            …
            …
            …
            UMS_EVENT_ALERT_STATUS=OPEN
            UMS_EVENT_INT_STATUS=Y
            UMS_EVENT_EIF_STATUS=Y
        IN UMSEVENT
    FI

    SEND OPEN ALERT BY CALL Notify_eif METHOD

 FOREND
 BEGINEND
```

i. Sending closed alert

```
BEGIN
     //update is for closing pure events i.e. don’t sent closed alert for pure event
     UPDATE
        UMS_EVENT_ALERT_STATUS='CLOSED',
        UMS_EVENT_EIF_STATUS='Y'
     WHERE
     UMS_EVENT_NATURE='PURE',
     UMS_EVENT_ALERT_STATUS='OPEN',
        UMS_EVENT_INT_STATUS = 'N'

        FOR EACH RECORD IN UMSEVENT
        WHERE
        UMS_EVENT_INT_STATUS = 'N',
        UMS_EVENT_ALERT_STATUS = 'OPEN',
        UMS_EVENT_ALERT_TYPE = 'STD'

        UPDATE
                UMS_EVENT_ALERT_STATUS=’CLOSED’
        IN UMSEVENT

        SEND CLOSED ALERT BY CALL Notify_eif METHOD
       ENDFOR
ENDBEGIN
```

j. Close the open alerts for NON STD alerts at end of day
k. After processing data truncate UMSTEMP TABLE

1.2. Notify_eif method

a. Variables definition for session i.e. user, password and host
b. Establishes a session and exec channel with libra server
c. Execute the command “/ums/install/core/eif/umspost ….”
d. Close the channel and session

18. Summarization script

Summarization is happening by running “/ums/install/summarisation.sh” script. The script runs every Monday and creates summarized historical data for last week. The summarization script creates summarized tables of type daily business hour data and non-bussines hour data summarization. For example, AIXMEM_DB (Summarized daily business hour data derived from AIXMEM table), AIXMEM_DNB (Summarized daily non-business hour data derived from AIXMEM table). Currently summarization is configured for AIXFS, AIXMEM, AIXCPU, AIXLOAD, AIXIOSTAT tables.
Note: Before running “/ums/install/summarisation.sh” script, ensure new server entries are added in NON_BUSINESS_DAY and BUSINESS_HOUR tables.

/ums/install/summarisation.sh

```
########################################################################
# Script Name:          summarisation.sh
# Description:          Does the summarisation for detailed tables
# Frequency to run:     Every Monday
# Data to summarize:    Last Week's Data
########################################################################

export ORACLE_HOME=/oracle/ora11/product/11.2.0/dbhome_1
export ORACLE_SID=UMSDB
export ORACLE_BIN=${ORACLE_HOME}/bin
export PATH=$PATH:${ORACLE_BIN}

sqlplus /nolog << _EOF_
CONNECT umsadmin@umsdb_libra/XXXXX
set heading off;
@/ums/install/summarisation.sql
SPOOL OFF
EXIT;
_EOF_
exit 0
```

19. Core Engine table structure

UMS application table structure shown below, detailed table structure is available in ums_oracle_table_ver15_10jan2015.xls

UMSTEMP			
No	**COLUMN NAME**	**DATATYPE**	**DESCRIPTION**
1	UMS_IDENTIFIER	VARCHAR2 (200)	Unique identifier for each event
2	UMS_ORIGIN	VARCHAR2 (30)	Hostname
3	UMS_ORIGIN_TIMESTAMP	DATE	Monitoring Timestamp
4	UMS_ALERT_NAME	VARCHAR2 (32)	Name of the alert
5	UMS_CLASS_NAME	VARCHAR2 (32)	Alerts are grouped based on class name
6	UMS_DISPLAY_ITEM_01	VARCHAR2 (32)	Static display item
7	UMS_DISPLAY_ITEM_02	VARCHAR2 (32)	Dynamic display item, value derived from database
8	UMS_MSG	VARCHAR2 (4000)	Message
9	UMS_ATR_LIST	VARCHAR2 (4000)	List of additional attributes
10	UMS_ALERT_TYPE	VARCHAR2 (4)	STD / BLLW / BLLM
11	UMS_ALERT_DESC	VARCHAR2 (1000)	Description about alert
12	UMS_NATURE	VARCHAR2(32)	Denotes the nature of the situation
13	DBWRITETIME	DATE (SYSDATE)	Record insert time in database
UMSEVENT			
1	UMS_EVENT_IDENTIFIER	VARCHAR2 (200)	Unique identifier for each event
2	UMS_EVENT_ORIGIN	VARCHAR2 (32)	Hostname
3	UMS_EVENT_ORIGIN_TIMESTAMP	DATE	Monitoring Timestamp
4	UMS_EVENT_ORIGIN_TIMESTAMP_UPD	DATE – NULL (32)	Updated monitoring timestamp for closed event
5	UMS_EVENT_ALERT_NAME	VARCHAR2 (32)	Name of the alert
6	UMS_EVENT_CLASS_NAME	VARCHAR2 (32)	Alerts are grouped based on class name
7	UMS_EVENT_DISPLAY_ITEM_01	VARCHAR2 (32)	Static display item
8	UMS_EVENT_DISPLAY_ITEM_02	VARCHAR2 (32)	Dynamic display item, value derived from database
9	UMS_EVENT_MSG	VARCHAR2 (4000)	Message
10	UMS_EVENT_ATR_LIST	VARCHAR2 (4000)	List of additional attributes sent with the event
11	UMS_EVENT_ALERT_STATUS	VARCHAR2 (16)	OPEN / CLOSED Denotes status of the alert
12	UMS_EVENT_INT_STATUS	VARCHAR2 (4)	Y / N Is an internal status is used to identify the status of the event; N denotes old records in event table and Y denotes either new event or duplicate event.
13	UMS_EVENT_ALERT_TYPE	VARCHAR2 (4)	STD (Standard) / BLLW (Base Line Last Week) / BLLM (Base Line Last Month)
14	UMS_EVENT_ALERT_DESC	VARCHAR2 (1000)	Description about alert
15	UMS_EVENT_EIF_STATUS	VARCHAR2 (4)	Y / N EIF status denotes notification is sent to Netcool or not. Y denotes sent and N denotes not sent
16	UMS_EVENT_NATURE		Denotes the nature of the situation
17	DBWRITETIME	DATE(SYSDATE)	Record insert time in database

20. Managing UMS application

1. To start Upload Engine use below command and for errors refer /ums/install/logs/ums_upload_engine.log

   ```
   $ sh /ums/install/upload/run.sh >> /ums/install/logs/ums_upload_engine.log
   ```

2. To start Core Engine use below command and for errors refer /ums/ install/logs/ums_core_engine.log

   ```
    $ sh /ums/install/core/run.sh >> /ums/install/logs/ums_core_engine.log
   ```

3. To start Summarization process use below command and for errors refer /ums/install/logs/summarisation.log

   ```
   $ sh /ums/install/summarisation.sh >> /ums/install/logs/summarisation.log
   ```

   Cron configuration for ums user

   ```
   [ums@itmtest]$ crontab -l
   05 * * * * sh /ums/install/upload/run.sh >> /ums/install/logs/ums_upload_engine.log
   10 * * * * sh /ums/install/core/run.sh >> /ums/install/logs/ums_core_engine.log
   05 11 * * 1 sh /ums/install/summarisation.sh >> /ums/install/logs/summarisation.log
   ```

21. GUI Component

The GUI component is a browser-based user interface to view monitoring data in graphical presentation and deployed on top of tomcat server. The GUI component is developed under java technology. The GUI interface displays information about monitored systems in enterprise; on the left is the Navigator, which shows the arrangement of monitored systems and allows to access information collected by scripts deployed in monitored systems. On the right is a workspace. The workspace consists of tabular and graphical data views.

Below mentioned table represent the core Servlet, Jsp and JAR files for GUI component

Servlet	ServletDatabaseConnect	Defines the core presentation view which includes tables and graphs
Servlet	LoginServlet	For login authentication
Jsp	index.jsp	Login page
Jsp	tree.jsp	Defined the navigator view
Jsp	LoginSuccess.jsp	Page view containing Event list
JAR	jcommon-1.0.23.jar jfreechart-1.0.19.jar	For graph generation
JAR	ojdbc6.jar	For database connection

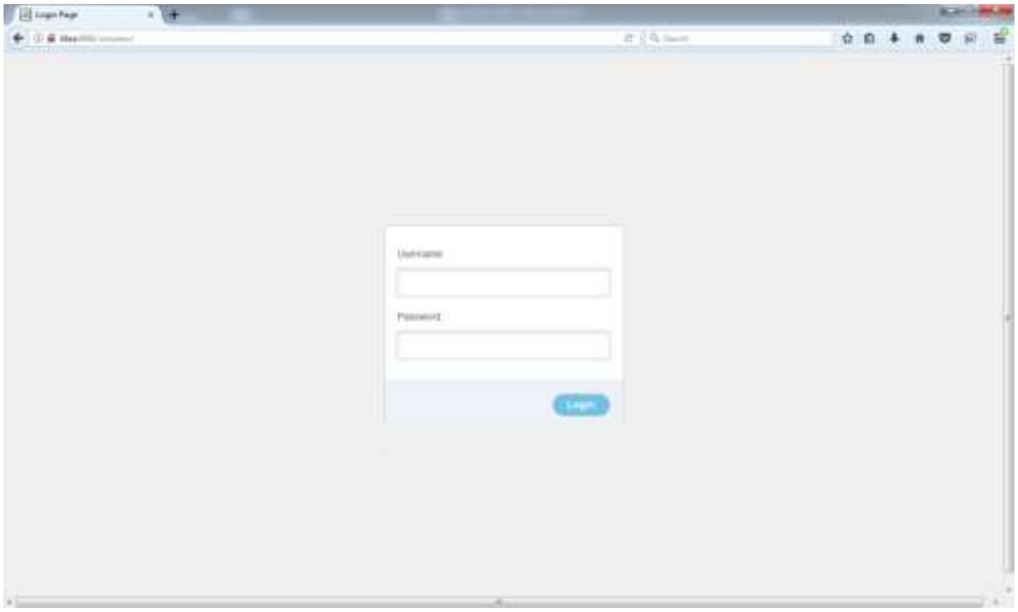
The directory structure for ums GUI component is mentioned below

```
.
ums
|_ aix_report.jsp
|_ aix_report_parm.jsp
|_ css
|   |_ bootstrap.css
|   |_ bootstrap.css.map
|   |_ bootstrap.min.css
|   |_ bootstrap-theme.css
|   |_ bootstrap-theme.css.map
|   |_ bootstrap-theme.min.css
|   |_ bootums.css
|   |_ datepicker.css
|   |_ down.css
|   |_ dtree.css
|   |_ jquery-ui.css
|   |_ layout.css
|_ fonts
|   |_ glyphicons-halflings-regular.eot
|   |_ glyphicons-halflings-regular.svg
|   |_ glyphicons-halflings-regular.ttf
|   |_ glyphicons-halflings-regular.woff
|_ head.jsp
|_ home.jsp
|_ images
|   |_ aixcpu_Araratstby.png
|   |_ aixcpu_atlas.png
|   |_ aixcpu_atmlivestby.png
|   |_ aixcpu_FinacleTD.png
|   |_ aixcpu_KARAKORAMSTBY.png
|   |_ aixcpu_mobapptest.png
|   |_ aixcpu_neptune.png
|   |_ aixcpu_p780voi2.png
|   |_ aixcpu_phobos.png
|   |_ aixcpu_SATPURA.png
|   |_ aixcpu_technetium.png
|   |_ aixcpu_vioserver1.png
|   |_ aixfs.png
|   |_ aixpm.png
|   |_ aixps.png
|   |_ custom_bl.png
|   |_ custom_b.png
|   |_ custom_br.png
|   |_ custom_l.png
|   |_ custom_r.png
|   |_ custom_tl.png
|   |_ custom_t.png
|   |_ custom_tr.png
|   |_ datepicker_bl.png
|   |_ datepicker_b.png
|   |_ datepicker_br.png
|   |_ datepicker_l.png
|   |_ datepicker_r.png
|   |_ datepicker_tl.png
|   |_ datepicker_t.png
|   |_ datepicker_tr.png
|   |_ field.png
|_ img
|   |_ aix.gif
|   |_ base.gif
|   |_ cd.gif
|   |_ empty.gif
|   |_ enterprise.gif
|   |_ folder.gif
|   |_ folderopen.gif
|   |_ globe.gif
|   |_ imgfolder.gif
|   |_ joinbottom.gif
|   |_ join.gif
|   |_ line.gif
|   |_ minusbottom.gif
|   |_ minus.gif
|   |_ musicfolder.gif
|   |_ node.gif
|   |_ nolines_minus.gif
|   |_ nolines_plus.gif
|   |_ page.gif
|   |_ plusbottom.gif
|   |_ plus.gif
|   |_ question.gif
|   |_ trash.gif
|   |_ work.gif
|   |_ workstation.gif
|_ index.jsp
|_ js
|   |_ bootstrap.js
|   |_ bootstrap.min.js
|   |_ carousel.js
|   |_ dtree.js
|   |_ eye.js
|   |_ jquery.js
|   |_ jquery-ui.js
|   |_ layout.js
|   |_ utils.js
|_ linux_report.jsp
|_ LoginSuccess.jsp
|_ META-INF
|   |_ context.xml
|   |_ MANIFEST.MF
|_ report_body.jsp
|_ report_head.jsp
|_ report_home.jsp
|_ slide.jsp
|_ tree.jsp
|_ WEB-INF
    |_ classes
    |   |_ Gen_excel.class
    |   |_ LoginServlet.class
    |   |_ PoiWriteExcelFile.class
    |   |_ ServletDatabaseConnect.class
```

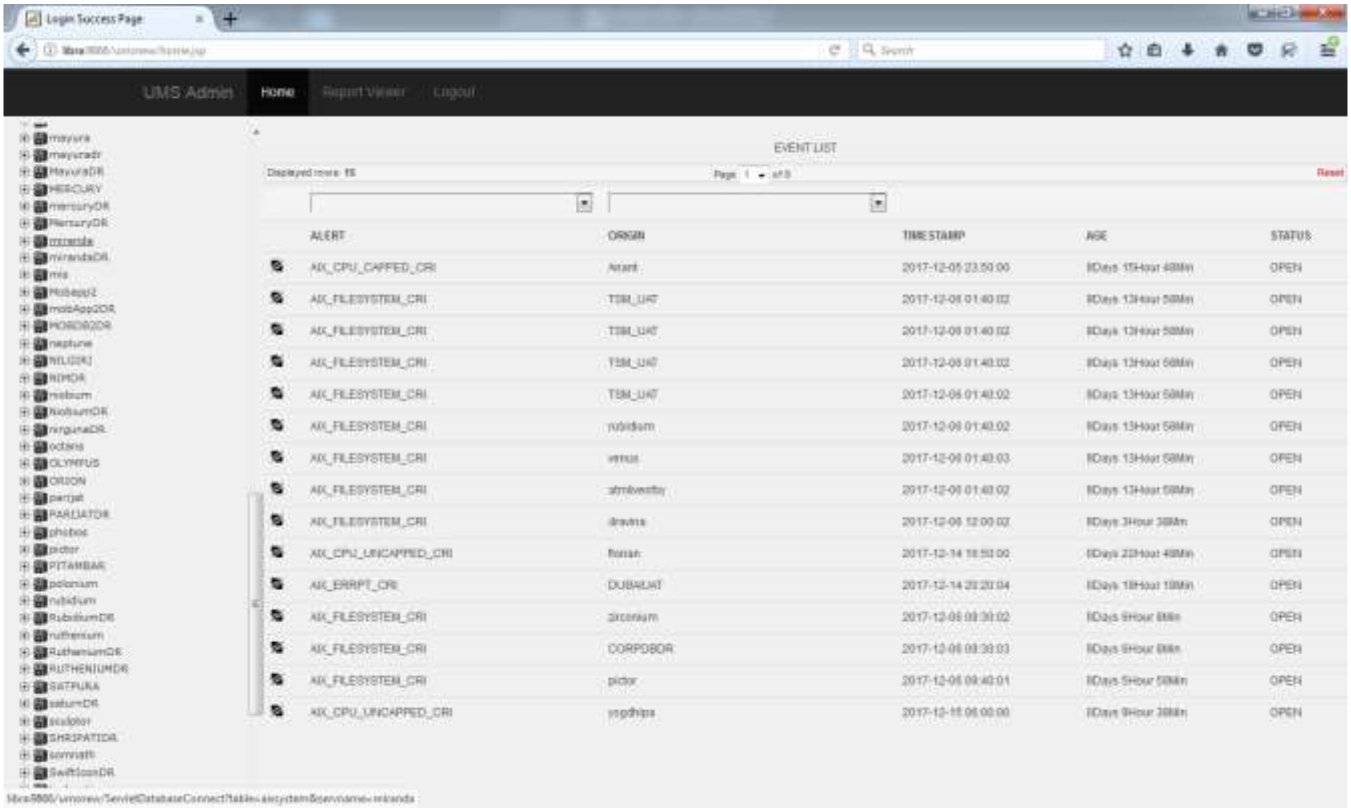
```
|_ lib
|   |_ activation.jar
|   |_ jcommon-1.0.23.jar
|   |_ jfreechart-1.0.19.jar
|   |_ ojdbc6.jar
|   |_ poi-3.2-FINAL-20081019.jar
|   |_ poi-3.5-FINAL.jar
|   |_ poi-contrib-3.2-FINAL-20081019.jar
|   |_ poi-scratchpad-3.2-FINAL-20081019.jar
|   |_ webservices-api.jar
|   |_ webservices-extra-api.jar
|   |_ webservices-extra.jar
|   |_ webservices-rt.jar
|   |_ webservices-tools.jar
|_ web.xml
```

22. GUI Administration

The GUI application is hosted in Tomcat and can be accessed using below link
http://xxx.xxx.xxx.xxx:xxxx/ums
The GUI is authenticated using username and password, once authentication is success redirected to Home page.



The home page contains Menu bar at top, Navigator bar at left and Workspace at right; Menu bar having Home and logout options and Navigator bar has the list of monitored servers grouped under a tree structure. By expanding the tree we get options to access the Processor, Memory, Filesystem, Disk-IO and Process; data available in Right side workspace. The home page workspace gives detailed list of open alerts derived from UMSEVENT table



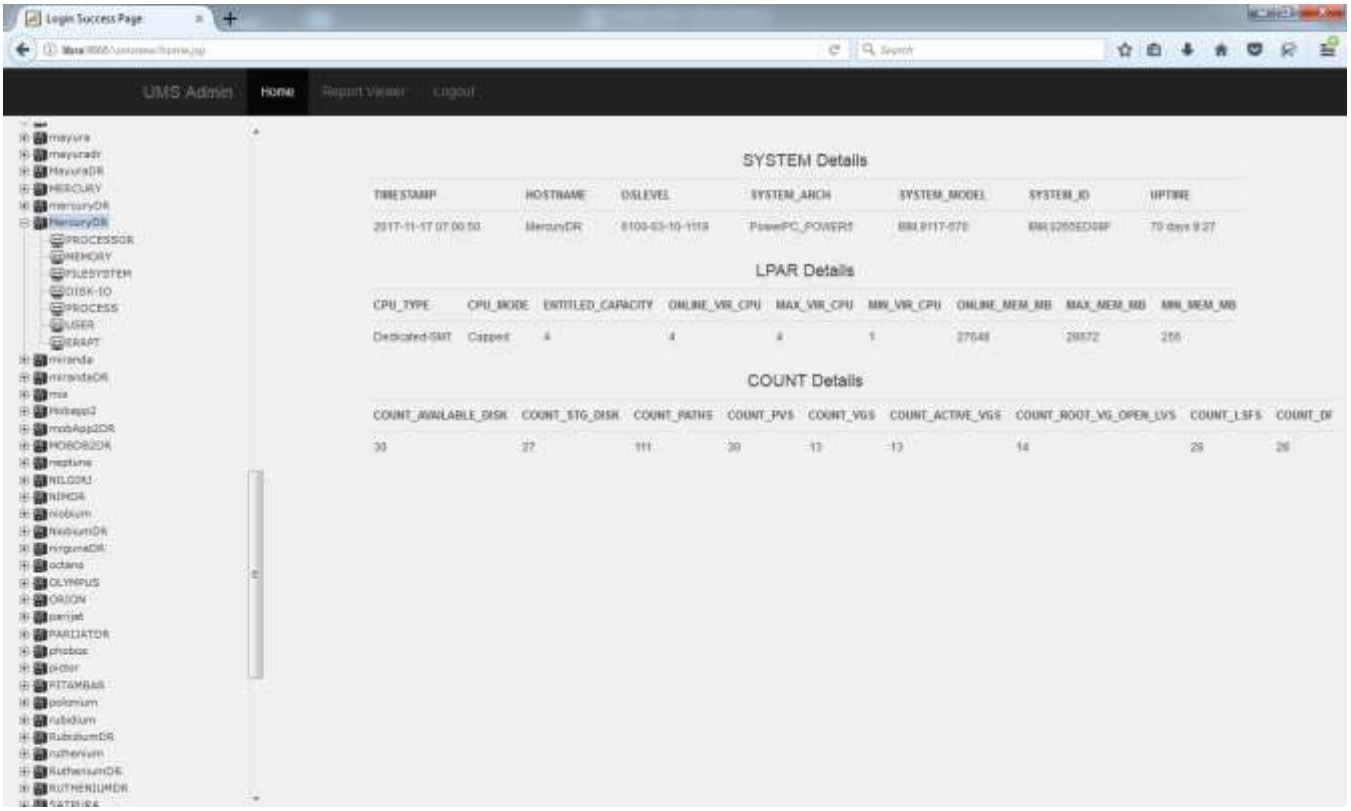
EVENT LIST

Displayed rows: 15 Page 1 of 8

	ALERT	ORIGIN	TIMESTAMP	AGE	STATUS
	AIX_CPU_CAPPED_CRI	Antant	2017-12-05 23:50:00	8Days 15Hour 48Min	OPEN
	AIX_FILESYSTEM_CRI	TSM_UAT	2017-12-06 01:40:02	8Days 13Hour 58Min	OPEN
	AIX_FILESYSTEM_CRI	TSM_UAT	2017-12-06 01:40:02	8Days 13Hour 58Min	OPEN
	AIX_FILESYSTEM_CRI	TSM_UAT	2017-12-06 01:40:02	8Days 13Hour 58Min	OPEN
	AIX_FILESYSTEM_CRI	TSM_UAT	2017-12-06 01:40:02	8Days 13Hour 58Min	OPEN
	AIX_FILESYSTEM_CRI	rubidium	2017-12-06 01:40:02	8Days 13Hour 58Min	OPEN
	AIX_FILESYSTEM_CRI	venus	2017-12-06 01:40:03	8Days 13Hour 58Min	OPEN
	AIX_FILESYSTEM_CRI	atmiventby	2017-12-06 01:40:02	8Days 13Hour 58Min	OPEN
	AIX_FILESYSTEM_CRI	dravina	2017-12-06 12:00:02	8Days 3Hour 38Min	OPEN
	AIX_CPU_UNCAPPED_CRI	florian	2017-12-14 18:50:00	8Days 23Hour 48Min	OPEN
	AIX_ERRPT_CRI	DUBAIUAT	2017-12-14 20:20:04	8Days 18Hour 18Min	OPEN
	AIX_FILESYSTEM_CRI	zirconium	2017-12-06 09:30:02	8Days 9Hour 8Min	OPEN
	AIX_FILESYSTEM_CRI	CORPDBDR	2017-12-06 09:30:03	8Days 9Hour 8Min	OPEN
	AIX_FILESYSTEM_CRI	pictor	2017-12-06 09:40:01	8Days 5Hour 58Min	OPEN
	AIX_CPU_UNCAPPED_CRI	yogdhipa	2017-12-15 06:00:00	8Days 9Hour 38Min	OPEN

Mira9886/umsnew/ServletDatabaseConnect?table=aixsystem&servname=miranda

By clicking the server Node we get the basic System, LPAR and VG details of server



SYSTEM Details

TIMESTAMP	HOSTNAME	OSLEVEL	SYSTEM_ARCH	SYSTEM_MODEL	SYSTEM_ID	UPTIME
2017-11-17 07:00:50	MercuryDR	6100-03-10-1119	PowerPC_POWER5	IBM,9117-570	IBM,0265ED38F	70 days 9:27

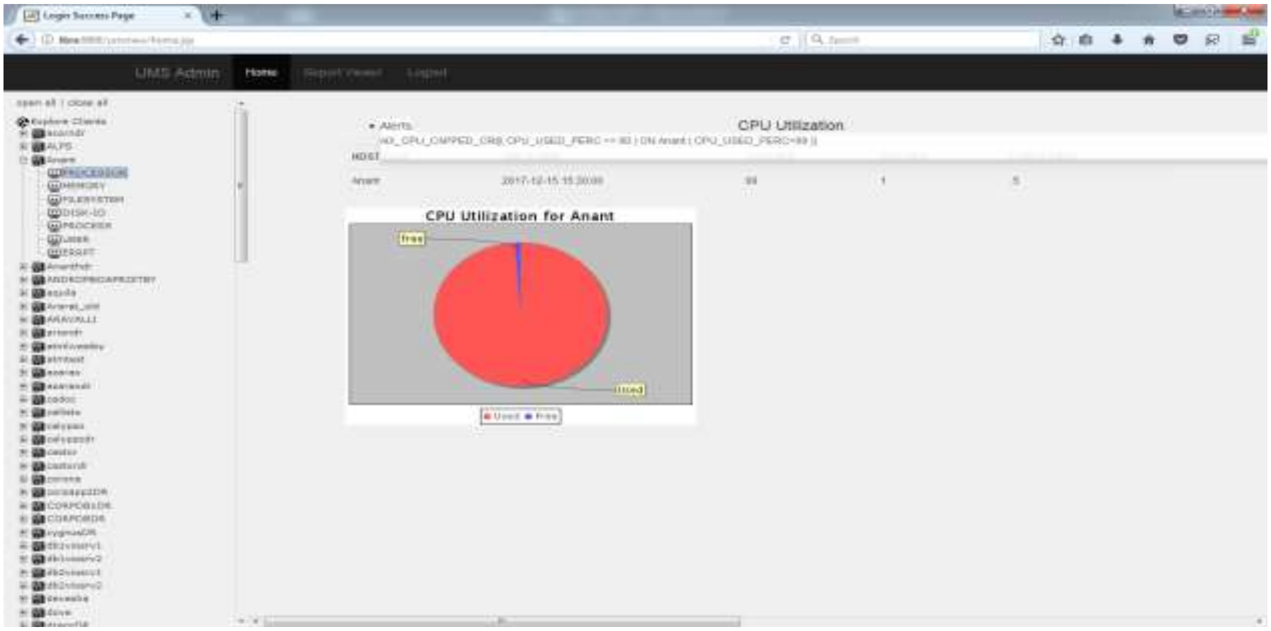
LPAR Details

CPU_TYPE	CPU_MODE	ENTITLED_CAPACITY	ONLINE_VIR_CPU	MAX_VIR_CPU	MIN_VIR_CPU	ONLINE_MEM_MB	MAX_MEM_MB	MIN_MEM_MB
Dedicated-SMT	Capped	4	4	4	1	27648	28672	256

COUNT Details

COUNT_AVAILABLE_DISK	COUNT_STG_DISK	COUNT_PATHS	COUNT_PVS	COUNT_VGS	COUNT_ACTIVE_VGS	COUNT_ROOT_VG_OPEN_LVS	COUNT_LSFS	COUNT_DF
30	27	111	30	13	13	14	28	28

By clicking the Processor child node we get the CPU utilization details

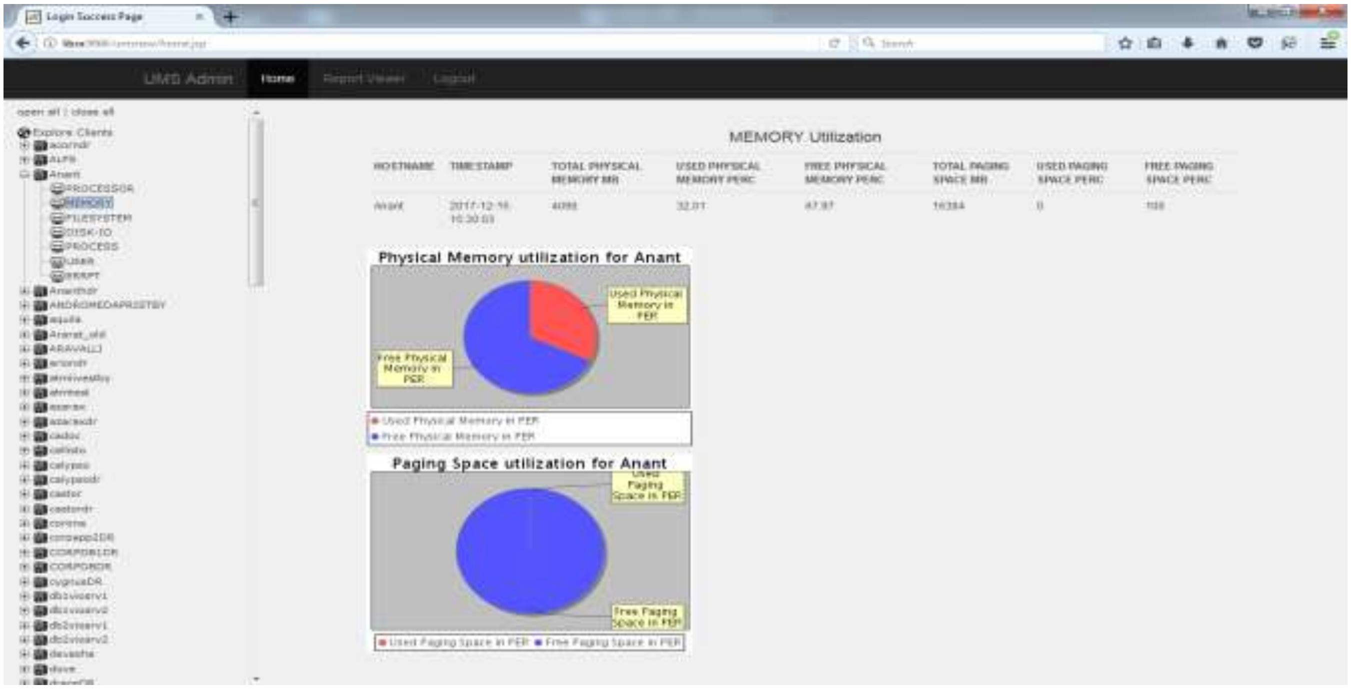


CPU Utilization

HO_CPU_CAPPED_CR8(CPU_USED_PERC >= 80) ON Anant (CPU_USED_PERC=99))

HOST				
Anant	2017-12-15 15:30:00	99	1	5

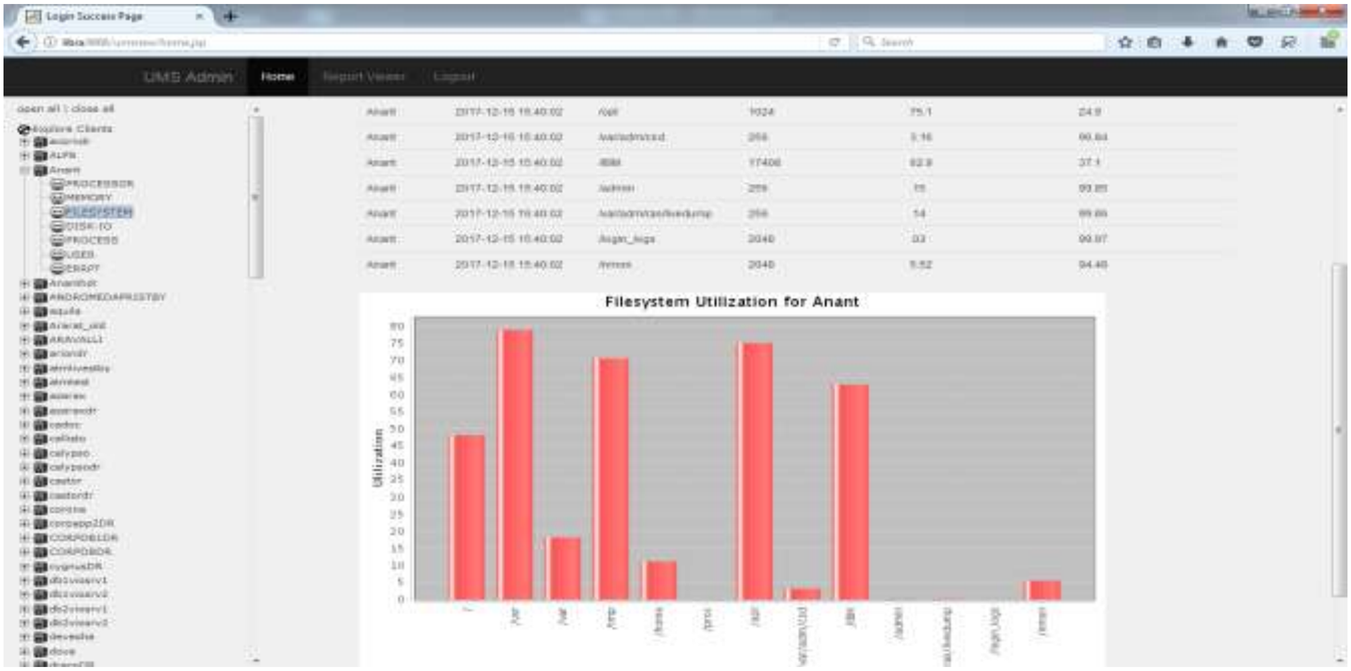
By clicking the Memory child node we get the physical and virtual memory utilization details. Alert specific to the selected server is also available for quick view; this is accessible by mouse over option on Alerts menu option available left side of workspace. If no alerts are available for selected server node, Alert menu will be disabled


Login Success Page
UMS Admin
Home
Report Viewer
Logout
open all | close all
Explore Clients
ALPS
Anant
PROCESSOR
MEMORY
FILESYSTEM
DISK-IO
PROCESS
USER
ERRPT
Physical Memory utilization for Anant
Used Physical Memory in PER
Free Physical Memory in PER
● Used Physical Memory in PER
● Free Physical Memory in PER
Paging Space utilization for Anant
Used Paging Space in PER
Free Paging Space in PER
● Used Paging Space in PER ● Free Paging Space in PER

MEMORY Utilization

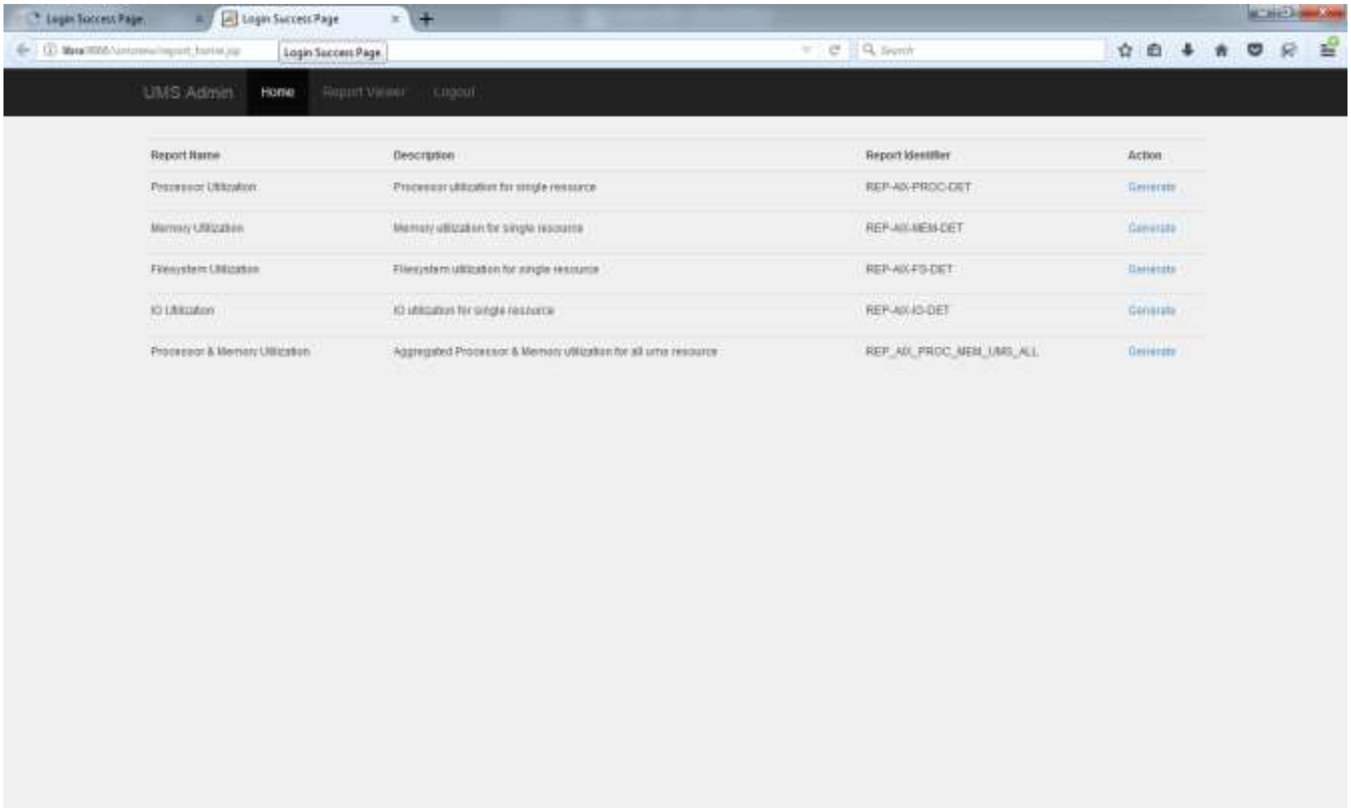
HOSTNAME	TIMESTAMP	TOTAL PHYSICAL MEMORY MB	USED PHYSICAL MEMORY PERC	FREE PHYSICAL MEMORY PERC	TOTAL PAGING SPACE MB	USED PAGING SPACE PERC	FREE PAGING SPACE PERC
Anant	2017-12-16 16:30:03	4096	32.03	67.97	16384	0	100

By clicking the Filesystem child node we get the Filesystem utilization details



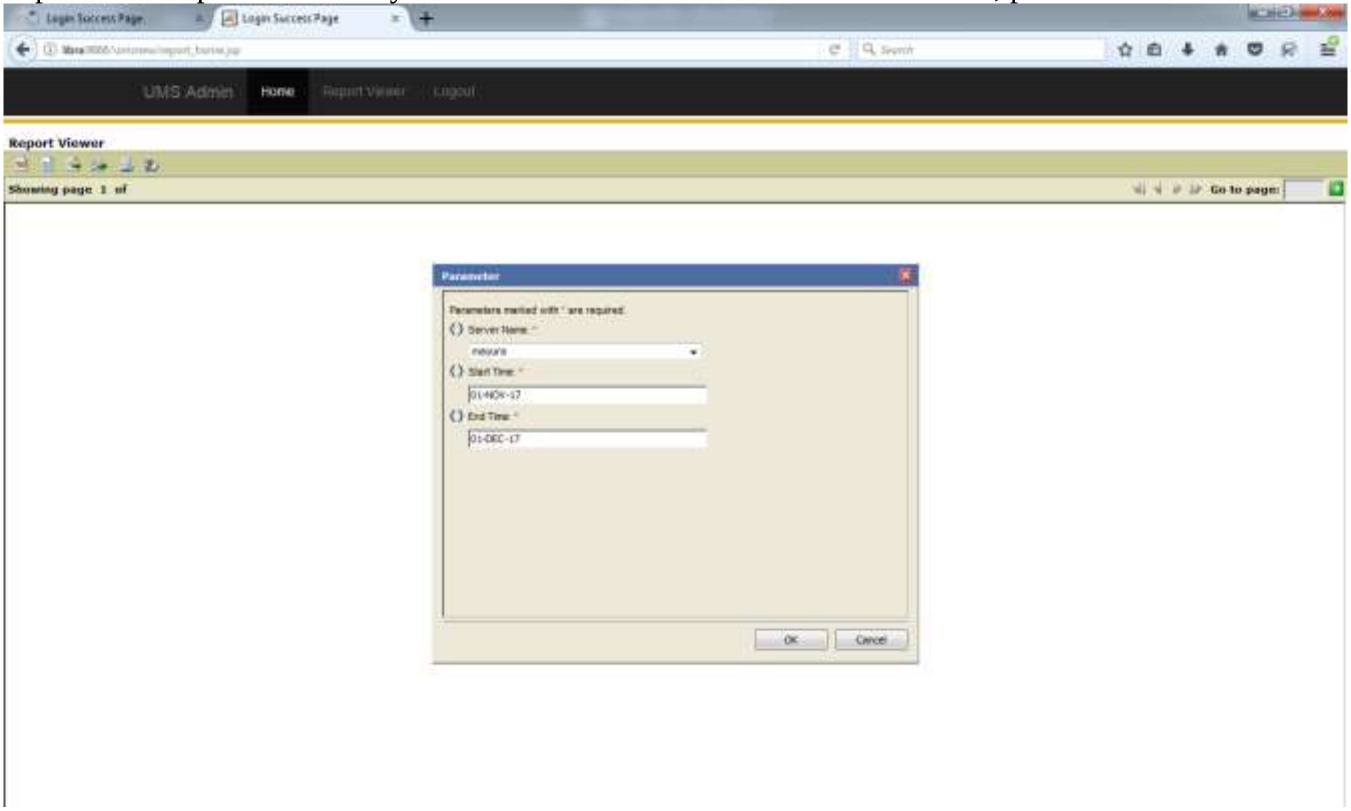
Anant	2017-12-15 15:40:02	/opt	1024	75.1	24.9
Anant	2017-12-15 15:40:02	/var/adm/ras	256	3.16	96.84
Anant	2017-12-15 15:40:02	/888	17408	62.9	37.1
Anant	2017-12-15 15:40:02	/admin	256	15	93.85
Anant	2017-12-15 15:40:02	/var/adm/ras/livedump	256	14	99.86
Anant	2017-12-15 15:40:02	/login_logs	2048	03	99.97
Anant	2017-12-15 15:40:02	/nmon	2048	5.52	94.48

By clicking the process child node we get the process utilization details & DISK-IO child node we get the IO utilization details for disk.



Report Name	Description	Report Identifier	Action
Processor Utilization	Processor utilization for single resource	REP-AIX-PROC-DET	Generate
Memory Utilization	Memory utilization for single resource	REP-AIX-MEM-DET	Generate
Filesystem Utilization	Filesystem utilization for single resource	REP-AIX-FS-DET	Generate
IO Utilization	IO utilization for single resource	REP-AIX-IO-DET	Generate
Processor & Memory Utilization	Aggregated Processor & Memory utilization for all ums resource	REP_AIX_PROC_MEM_UMS_ALL	Generate

Report Viewer provides facility to view and download historical data in excel, pdf and other formats.



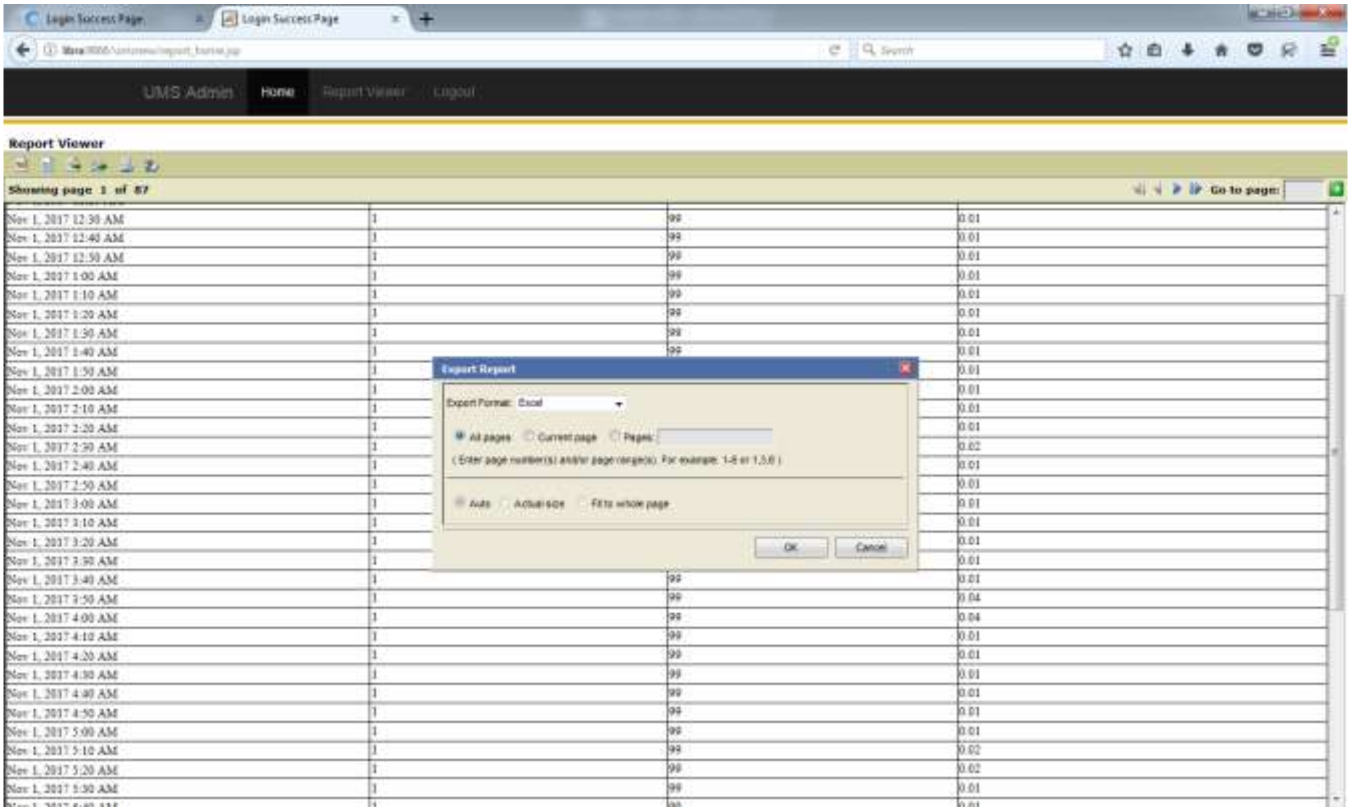
Parameters marked with * are required.

Server Name: * mesura

Start Time: * 01-NOV-17

End Time: * 01-DEC-17

Here is sample report prompt page for user inputs.



Showing page 1 of 87

Nov 1, 2017 12:30 AM	1	99	0.01
Nov 1, 2017 12:40 AM	1	99	0.01
Nov 1, 2017 12:50 AM	1	99	0.01
Nov 1, 2017 1:00 AM	1	99	0.01
Nov 1, 2017 1:10 AM	1	99	0.01
Nov 1, 2017 1:20 AM	1	99	0.01
Nov 1, 2017 1:30 AM	1	99	0.01
Nov 1, 2017 1:40 AM	1	99	0.01
Nov 1, 2017 1:50 AM	1		0.01
Nov 1, 2017 2:00 AM	1		0.01
Nov 1, 2017 2:10 AM	1		0.01
Nov 1, 2017 2:20 AM	1		0.01
Nov 1, 2017 2:30 AM	1		0.02
Nov 1, 2017 2:40 AM	1		0.01
Nov 1, 2017 2:50 AM	1		0.01
Nov 1, 2017 3:00 AM	1		0.01
Nov 1, 2017 3:10 AM	1		0.01
Nov 1, 2017 3:20 AM	1		0.01
Nov 1, 2017 3:30 AM	1		0.01
Nov 1, 2017 3:40 AM	1	99	0.01
Nov 1, 2017 3:50 AM	1	99	0.04
Nov 1, 2017 4:00 AM	1	99	0.04
Nov 1, 2017 4:10 AM	1	99	0.01
Nov 1, 2017 4:20 AM	1	99	0.01
Nov 1, 2017 4:30 AM	1	99	0.01
Nov 1, 2017 4:40 AM	1	99	0.01
Nov 1, 2017 4:50 AM	1	99	0.01
Nov 1, 2017 5:00 AM	1	99	0.01
Nov 1, 2017 5:10 AM	1	99	0.02
Nov 1, 2017 5:20 AM	1	99	0.02
Nov 1, 2017 5:30 AM	1	99	0.01

Export Report

Export Format: Excel

● All pages ○ Current page ○ Pages:

(Enter page number(s) and/or page range(s). For example: 1-8 or 1,3,6)

● Auto ○ Actual size ○ Fit to whole page

OK Cancel

Sample report with viewing and downloading facility.
For logout, click the Logout option in menu bar which redirects to login page.

23. Troubleshooting

1. For basic level troubleshooting of Upload Engine use the below command

```
$ grep java /ums/install/logs/ums_upload_engine.log -B 3
```

```
INFO 141120190517158 <read_Meta_File>--Name of Monitoring data File to be processed-->/ums/source_dir/AIXTOPCPU_niobium.txt_20_11_2014_19_00 INFO 141120190517670 <Insert_Into_Database> 20 record needs to be inserted in AIXTOPCPU Table Name INFO 141120190517670 <Insert_Into_Database> Query to be inserted-->INSERT INTO AIXTOPCPU(HOSTNAME, SERVER_DATE_TIME, OWNER, PID, PPID, CPU_PERC, COMMAND, DBWRITETIME) VALUES(?,to_date(?,'dd/mm/yyyy hh24:mi:ss'),?,?,?,?,?,SYSDATE) ERROR 141120190517667 <set_File_Content/Insert_Into_Database> UMS-CE-E005 java.sql.BatchUpdateException: ORA-01461: can bind a LONG value only for insert into a LONG column
```

From the above output we came to know the problem is with AIXTOPCPU_niobium.txt_20_11_2014_19_00 data file. Check the content format and data to resolve the issue.

2. For Advance level troubleshooting of Upload Engine view the log /ums/install/logs/ums_upload_engine.log. For troubleshooting Core Engine view the log /ums/install/logs/ums_core_engine.log
3. For troubleshooting summarization process view the log /ums/install/logs/summarisation.log
4. Core engine are also available in UMSEXCEPTION table. To query the table use the below command
   SQL> select * from UMSEXCEPTION order by dbwritetime DESC;

24. Manage cron and SCP service

Cron is a scheduler used to schedule the scripts at specific intervals. The users who need the cron facility; username need to be added in /etc/cron.allow file.

SSH is often used to login from one system to another without requiring passwords. It requires client machine to generate set of private/public pair. ssh-keygen command is used to generate that key pair.

-End-