COM Health check & Monitoring

# Need for a heath check and monitoring

Admin for COM while running in production or testing environment,

* Server is up, running and usable
* Server load – DB connections, worker threads, async threads, memory footprint, http sessions etc. Stats on the Type/ Number of requests to COM system,
* System problem identification and notification
* Identification of runtime errors/exception - COM doing lots of system operations like Payment retry, Shipment Rejection. This heath check should be able to identify get the unresolved problems to Admin notice
* Debugging data runtime errors/exceptions,

# System Availability

## Xserver

Post a simple logical date Request to Xserver using a poster dialog

* Set wait time for the request
* If request timeout, Raise an warning alert on xserver
* If error response, Raise an RED alert as system down
  + Further actions: Save the server log dump and start new instance of xserver
* If request doesn’t timeout, verify the response (current Datetime)
* \*If difference between logicalDate and systemDate is more than 30 min, raise RED alert (This would help for DST switch)

## DB

* Check if the DB server is up and Running
* Check if the Oracle Service and Listener are running in the Machine
  + If service or listener not running, start it automatically and log it in the monitoring system
  + If Server down notify the User.
* Create a local connection with DB as admin and close it.
  + If no errors in the above the connection should work fine
* Verify the connection pool from xserver and verify the peak & foot time of connection utilization.
  + Find a way to get validation of the connection pool from xserver.

## Weblogic

(When servers running Co-allocated mode, this is not needed)

* Weblogic doesn’t have any MBeans In-built to access the Memory details run GC.

# Memory Management/Monitoring

## Heap Memory available for all Servers

(Use JMX libs for getting this info)

* Make use of MBeans available with Java SE,
* We can implement this in-line with jConsole available for JVM along with Java SE.

## \*System space for DB

* This mostly performed by the Infrastructure support people.
* Use standard system libs to get the system space, additionally application or DB insert exception can be handled in heathcheck.
* Simple Code in Java for getting the Space information

(File file = **new** File("c:");

**long** totalSpace = file.getTotalSpace();

*//total disk space in bytes.*

**long** usableSpace = file.getUsableSpace();

*//unallocated / free disk space in bytes.*

**long** freeSpace = file.getFreeSpace();

*//unallocated / free disk space in bytes.*

## CPU Usage

* OperatingSystemMXBean can be used to get the CPU usage.
* Alert for Maximum CPU usage,
* Stats on CPU usage timeframe during the working hours and type of requests
  + This can be done using the periodical data collection from system and store in XML file.

# Application monitoring

* 1. Up time from last system maintenance

Showing the last maintenance time and Uptime since last Maintenance

* 1. Exceptions occurred in the system

Analyzing logs and get No of exceptions occurred and type of Exceptions in the system,

See if we can handle in XServer and push message.

* 1. Stats on the Type of requests and Timeframe
     1. CreateOrder Request (No Of request per Min or Sec/ Rejected or Error Request)
     2. Payment Request (No Of request per Min or Sec/ Error Request)
     3. Shipment Exceptions (Shipment Analyze)
  2. Average Response time for a Request (Search Customer/ Login CSR or SA)
  3. Average active users (Numbers and most active user) and timeframe.
  4. Verifying if any un resolved system actions like (Timers, Payment Errors, Manual Events)

# Thread Handling (Use JMX libs if required)

* Get Weblogic libs to get the Threads created and how many are active in the system,
* Need to verify in co-allocated mode
  + Threads in weblogic and threads in xServer
  + How the session managed between xserver and weblogic server
  + Find if any deadlocks

# \*Uncommitted Transactions

* Get the List of open connections
* Need to think about uncommitted transactions

# Exception/Error Logging

* This can be monitored/ push data from ABPP profiler (which is in discussion of implementing)
* Stats using the Log file,
  + Type of Exceptions /Errors, Time frame and its request type

# Garbage Collection

* This force GC can be done using JMX libs,
* Before Doing a force GC, get log dump to the separate shared location with timestamp
* Stats/ Graph for the memory usage when system is online.

## Some Useful links:

* System properties

<http://alvinalexander.com/blog/post/java/print-all-java-system-properties>

* CPU Usage and Platform controller

<http://stackoverflow.com/questions/47177/how-to-monitor-the-computers-cpu-memory-and-disk-usage-in-java>

* Plug-in for Weblogic from Spring Source for Mbean access

<http://static.springsource.org/spring/docs/1.2.x/api/org/springframework/jmx/support/WebLogicJndiMBeanServerFactoryBean.html>

* JMX configuration in weblogic server

<http://docs.oracle.com/cd/E12483_01/wlevs20/config_server/jmx.html>

* System Memory information using file object

<http://docs.oracle.com/javase/6/docs/api/java/io/File.html>

\* Need to re-visit.