

# OctoberTest - Software issues and performance

## Overview

- Overall we were pleased with the behavior of the tools
- Test was very useful for seeing how tools performed under semi-realistic load
  - Revealed some issues we need to address between now and launch

## Oracle crash

SCCS restarted database in ~30 minutes. Due to "out of shared pool space". Shared pool is used, among other things, to cache recent queries so that they can be reused. Non optimal queries in the DataQualityMonitoring application seem to have been responsible for this. The queries were fixed immediately.

We learned that we have to e-mail **db-admin** to report database outages.

- Learn to use GRIDControl, an insanely complicated tool to monitor Oracle. We can use it to find out what queries are heavier on the system and ways to optimize them.
- RAC system: Real Application Clusters. For load balancing, scalability and fail over.

## Web Server Crashes

There were a few (5?) crashes of the web server on which the DataQualityMonitoring application is deployed. All due to the server running out of memory.

The DataQualityMonitoring application needs to be finely tuned:

- provides (**so far**) 30,031 trending quantities, stored at **15 seconds** intervals
- 31,525 histograms

When multiple runs are selected, histograms are added and trending quantities are fetched from the database for the resulting time period.

The first crash happened 2.5 hours into the test on Tue morning. Applications were rebalanced (i.e. DataQualityMonitoring application was isolated) and the memory was increased to 1.5 Gb. Three crashes happened in the last shift in an hour (guess that shifters liked to see the trended data over the 16 orbits (at 15 seconds intervals?) ).

- Clustering : Tomcat supports clustering.
- Aggregate old data to bigger time intervals?
- Prevent users from getting more data points than pixels on the screen? (6 hours)

Overall [heavily used](#) by shifters.

## Digi Trending Ingestion Failure (2)

Due to a deadlock when digi ingestion overlapped with recon ingestion of the previous processing run.

This is not really a deadlock in the sense that Digi and Recon data is non overlapping. But they share the same tables.

- indexing might fix the problem
- catching exceptions and waiting some time will fix it for sure

## Trending problems

- **Counter Normal Evts** quantity jumping by a factor of two. This problem was due to a duplicate variable in Digi and Recon that should have not been ingested in the database. The jump happened after Recon ingestion.
  - This was the only duplicated variable
- **Periodic Trigger Rate** went from 800Hz to 400Hz. Is this reproducible?

We need experts to sit down and compare trending plots from the web to the ones produced with desktop tools opening root files (there are only 30K of them!).

## Pipeline

- Does not seem to like oracle crashing. Will try to make it more robust. Difficult to test without actually taking oracle down.
- SubStream rollback did not work as expected.
- Work is ongoing to speed up processing pipeline to decrease data turn-around time.

## NFS Problem

Half pipe for orbit 2 failed a few times due to NFS problems.

## Safari Compatibility

We might have to spend some time making sure that the javascript we use is compatible with Safari.

## Things we need to fix afterwards

In no particular order...

- Figure out how to read NFS files from Tomcat
- Stop using SLACDEV database
  - Rationalize use of Dev/prod etc, decide if we need other configuration options
- Remove duplication between data processing page and other apps
- Look into tomcat clustering
- Cross application trending, and Daily Report Application
- Improve data catalog interface especially for real data
  - L1 data products arranged by groups rather than folders
  - Look into WebDav/GUI for data catalog
- Integration between monitoring tools and Ops Log
  - Ability to easily copy plots to ops log and comment on them
  - Ability to find all recent comments on a plot
- Make Ops Log use same login system as everything else
- Get LAT Data Server tied into L1Proc
- Get portal working, at least for items like data processing page, grb summary etc
  - Generate RSS feeds from LogWatcher, Ops Log, JIRA, Confluence etc to display on portal page
- Ability to monitor all tomcat servers/applications from one page (and maybe restart them)
- Figure out why automatic generation of tomcat configuration on glast-win01,02 did not work
- .... and much more

What else should we have learned?