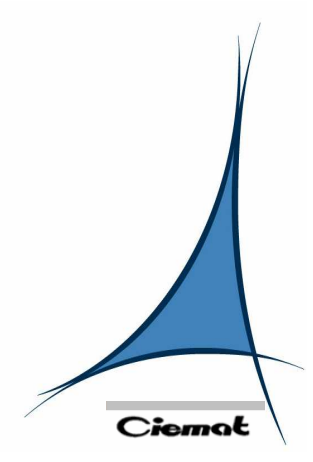
Report on WG1

Josep Flix (PIC/CIEMAT)
Simone Campana (CERN-IT-ES)

WLCG-TEG-OPS F2F meeting
[28th November2011]



WG1 - Overview

- Many contributions from Experiments (LHCb?) and Sites
- The contributions have been merged in a ***Google Doc***:
  - Read-only, available at:
    https://docs.google.com/a/pic.es/document/d/1o5uesrQz3AZMhzmve4wTGhc15QKionaSLE9Vvd5n5Jw/edit?pli=1
  - Linked from the WG1 sub-twiki:

    https://twiki.cern.ch/twiki/bin/view/LCG/WLCGTegOperationsWG1

- **Working on:**
  - Building of table template (“works well” & “needs improvement”), from the doc and comments from this talk.
  - Some cosmetics/polishing in the doc.
  - Proper inclusion of ALICE contribution to the text; “WLCG reporting of availability and reliability” discussion; inclusion of Jeff’s comments…

Technology and Tools (VOs p.o.v.)

- Some systems are based on a **common framework**, VO-customized, for example:
  - **ALICE**: in-house complete and homogeneous workload and data management monitoring. Utilization of ***Monalisa*** and of a common ***library*** to feed the monitoring system. Benefiting from local VOBOXes (SLAs) to collect site specific information, exposed via web + ***information feeds***.
  - **ATLAS** & **CMS**: homogeneus use of ***Dashboard*** common framework for job & tasks monitoring, accounting, and data management (ATLAS).
- Indeed, a variety of **ad-hoc monitoring systems**: **CMS PhEDEx**, **CMS Overview**, **CMS Site Readiness**, **ATLAS PanDA Monitoring**, etc... heavily used in Operations.
- All VOs adopted **SAM+Nagios** to to run experiment specific tests on remote sites and look for site availability.
- **Site Status Board** heavily used by ATLAS and CMS with similar goals.
- **HammerCloud** proved to be a very flexible system for functional and stress testing of production and analysis experiment specific workloads.

Technology and Tools (sites p.o.v.)

- Sites instrumented their internal monitoring to be able to **cope with fabric issues** (black holes WNs, crashing daemons, monitoring critical services, etc...):
  - **Nagios**, **Ganglia** and **Lemon** as the main tools used for fabric monitoring.
  - Rather different opinions of the various systems. In particular, **Nagios** has impossibility to monitor services rather than nodes, limitations in alarms masking, offers counter intuitive interface (especially for historical views) → The Nagios fork's **Icinga** can be adopted to cope with those limitations (for example, **PIC is deploying an instance already in Production**).
- Some sites have the VOs SAM Nagios alarms embedded on their local monitoring
- Normally, not all sites add VOs adhoc monitoring tools feeds to check for their status:
  - KIT uses **HappyFaces**
  - **CMS Site Readiness Widget**
  - …

Procedures and Policies Described

- WLCG Monitoring Coordination (absence of) ← Probably the **_main_ issue**
  - Many sparse initiatives, no standards → Monitoring "zoo"

- Sites and Experiments perspectives are still rather **distant**
  - Experiments do not understand what happens at sites (**SIRs** are fine to understand problems a posteriori, but sometimes is difficult to trace ongoing problems, except for sites with a strong contact with the experiment operations, which indeed tend to have less of those problems).
  - Generally, sites do not understand what the experiment is doing at the site.
  - **WLCG daily Operations calls** help to fill this gap, but not sufficient.

- Normally, sites have **procedures** to solve site incidents. **SAM availability policies** are generally adopted by both sites/VOs, as well as **critical services availability policies**.

- **Policies for Downtimes** declarations are as well reasonable and followed by sites.

Areas of Improvement & efficiency gains [1/2]

- **Monitoring** is generally **not treated "as a service"**:
  - Slow response, scalability issues, non-existing upgrading procedures, lack of QA checks,...
  - No programmatic APIs
  - No care to backward compatibility
  - No procedures/escalation/postmortem (*lack of procedures/policies*)

- Improve the current situation to deal with partial downtimes (CIC,GOCDB,OIM). This includes both **downtimes of a given service for a subset of VOs** and the **downtime of "part of a service"**
  - Additionally, **how to 'retire' a service from GOCDB needs to be improved** (currently, declaration of large downtimes for deprecated services)
  - **Experiment** computing systems **have also downtimes**, hardly exposed to sites. So the sites can hardly tell if a lack of activity is due to a lack of experiment activity, a problem in the experiment stack or a malfunctioning service at a site.

Areas of Improvement & efficiency gains [2/2]

- Experiments and sites find **SAM tests very convenient**, they strongly recommend it continue to be supported and they believe it would benefit of **improvements**:
  - Tests should be improved to be as closed as possible to reality. Ability to monitor and report the status of the site in terms of running an activity, as defined by the VOs (**CMS Site Readiness**).
  - VO SAM tests should need no interpretation, identifying unambiguously which service is failing so that the information can be properly injected in the local site monitoring. Reducing the number of fake positives is essential. Sometimes is difficult for VOs to breakdown failures modes and indicate origin of problems (the help of experts from both sides is still important).
  - Sampling at a high rate (several tests per hour) is essential to get quick feedback from sites… But it shouldn't impose sizeable load on the sites… and tests should run on busy sites (not sitting on the queue for long time).
  - VOs feel that the SAM Ops tests are not representing the actual availability of the sites, while most of the sites are using it to report to funding agencies, and those are the ones subject to WLCG MoU policies... But some sites might be applying "special setups" for tests… Some would like to move to VO-specific, but need clear understanding of errors seen (which are site problems?)…

Identify the largest use of operational effort

- Intervention of **experts** both in the experiment operations and at the site (experiment contact people) to solve incidences. Sites with a strong contact with the experiment operations and good service operators tend to have less problems, and they follow-up open issues much better. Obviously the cost in Ops manpower is not negligible.
- Lot of effort go in understanding and solving **Network problems**, mainly due to lack of efficient network monitoring tools.
- Efforts to **plan, develop, deploy and operate ad-hoc VO monitoring services** and building and **operating site monitoring/alarming tools + 24x7 support which is provided,** both at the sites (T1s) and in the experiments.
- Follow-up and resolving **false positives** (in particular with site downtimes monitoring).
- Follow-up and **QA for common monitoring tools** & baby-sitting upgrades.

Missing Areas [1/2]

- **Monitoring the Services**
  - Many (most services) do not come with a native monitoring
    - FTS comes with some monitoring, which is not at the level to be used in Ops. **FTSMonitor** provided by IN2P3-folks is much more suitable (ad-hoc monitoring tool).
  - Many services do not even expose enough info (logfiles) for fabric monitoring
  - Many sites/VOs build an ad/hoc service monitoring (more to the zoo…)
- Exposing **Monitoring/Accounting Information**
  - Information exist but it is not correctly exposed to experiments via Information System (InfoProviders over-written to cope with site internals, double-countings, ...).
  - VOs coping with incorrect values very often. System unreliable. Ad-hoc tools: **SiteDB**.
- Experiments have no access to some **"internal" site infos** (fairshare for example)
  - No reliable tape monitoring. No off-Grid monitoring.
  - Some VOs adapting ad-hoc tools such as **HappyFaces**, or finding other solutions...

Missing Areas [2/2]

- **Network monitoring**
  - Is simply not sufficient to efficiently Operate the complex network system we have.
  - A headache for operations. Underlying problems take long to be properly identified and solved.
  - Again, VOs developed their own ad-hoc monitoring tools (such as CMS parsing FTSMonitor instances at Tier-1s). Now a common tool in progress to help with this area (the *Dashboard Data Transfer Monitor*)... It might help, to check for performance degradations, but it might be not enough to look into details (JumboFrames problems, routes used, PerfSonar tests, ...)

- The **SSB** is tailored to give an experiment view of service and activities at the site. We are missing an equivalent system tailored for the sites.