

# Monitoring Overview: status, issues and outlook

Simone Campana  
Pepe Flix

- Technology and tools
  - Overview of Experiment, Site and Infrastructure monitoring
- Areas of Improvement and Potential Efficiency Gains
  - In order of impact
  - Proposals and discussions
- I'll mark with **(OP)** the Open Points worth discussion and agreement

- Alice: in-house complete monitoring
  - Workload Management, Data Management, Service Monitoring
  - Common Framework (MonAlisa) and Common Library for all use cases
  - Benefits from VOBOXes at all sites
- LHCb is very similar to Alice
- ATLAS and CMS: largely relying on Experiment Dashboards
  - Based on common framework
  - Several ad-hoc monitoring systems (CMS PheDex monitor, CMS Site Readiness, ATLAS Panda Monitors)
- (OP) could the Alice model be considered generally for all experiment monitoring

- All sites instrumented fabric monitoring
  - Crashing daemons, blackhole WNs etc ..
  - Popular tools: Nagios, Ganglia, Lemmon
- (OP) Should we aim at a common fabric monitoring system?
  - Not realistic from the site perspective
- (OP) Requirement on middleware providers
  - Avoid tight integration with any particular monitoring
  - Provide instead generic service probes which can be integrated in any framework
  - Apparently this is already a requirement for EMI provided services

- SAM is used by 4 experiments to monitor services at sites
  - Nagios probes (NGIs and Exp): launch tests and publish results in Messaging System
  - Results stored in Central DB as well as NGIs local DBs
  - ACE component: calculate availabilities
- SAM allows the definition of profiles (list of metrics)
  - Very useful to provide views to different communities
- SAM test results can be fetched from messaging and injected into local monitoring
  - Natively if the site uses Nagios
- Andrea' s presentation will deal in details with SAM measurements of availability and Usability

- HammerCloud is used by ATLAS and CMS (and LHCb?) for sites (stress) testing
  - Data Processing (Production and Analysis)
- The Site Status Board is used by ATLAS and CMS for site monitoring
  - Visualizes arbitrary metrics for a list of sites (highly configurable)
  - Filtering/Sorting + Visualization
  - Values are published by various providers and fetched by SSB through HTTP
  - Offers a programmatic interface to expose current and historical values
- Some experiments integrate the SSB with ad-hoc monitoring tools
  - For example the CMS Site Readiness



- Impact “rating” in parenthesis (to be discussed as well)
  - 0 (equals “negligible”) to 10 (equals “disaster”)
- (8) Monitoring coordination
- (7) Bridging sites and experiments perspectives
- (5) Network monitoring
- (5) Monitoring of Services
- (5) Monitoring as a Service
- (3) Exposing Monitoring Information

- No central monitoring coordination
  - Duplication of effort
  - Proliferation of monitoring tools, not necessarily covering all use cases
- (OP) Should we suggest the MB to create a semi-permanent working covering this role?
  - E.g. this was done inside CERN IT for internal monitoring
  - How broad should be the mandate?



- Sites and Experiment perspectives rather distant
  - Sites monitor services, experiments monitor services AND activities
  - Mapping an activity to a set of services is not straightforward
- Bridging today is done through “people”
- Technologies in the game:
  - SAM for service monitoring (4 VOs)
  - SSB for activity monitoring (for ATLAS and CMS)
  - Experiment specific tools (e.g. CMS Site Readiness)

- Availability/Reliability/Usability will be discussed in the next presentation
  - “Standard Tests” run by OPS vs “Standard Tests” run by VOs vs experiment-specific tests run by the VOs.
- How to have realistic tests?
  - Sampling at high rate vs DOS
  - Testing the real service and not a dedicated test node
  - Using experiment frameworks instead of Nagios probes could be a solution?
- SAM deployment
  - Nagios can scale horizontally, once the use case is given
    - Can we spell out what we need (number of tests, sampling rate)?
  - Does WLCG need a non centralized ACE for scalability?
    - General opinion is NO

- SAM granularity: today we can test service endpoints in GOCDDB/OIM
  - Do we need lower granularity (space tokens for example)?
  - Do we want to test services not in GOCDDB/OIM (e.g. squids)?
- SAM focuses on service endpoints, SSB focused at “activities” at sites
  - Should SAM allow to test the SITE for an ACTIVITY (e.g site X works for ATLAS analysis)
    - SAM comes with batteries included (ACE availability calculation, integration with site Nagios or in general site fabric monitoring)
    - At the same time, this is what other tools already do (Hammercloud + SSB for the example above)
  - Where is the boundary between SAM and SSB-like tools?
- How do we provide a site-oriented view of monitoring information?
  - The equivalent of SAM+SSB for experiments today
  - Do we need a SSB for sites? Should we revisit SiteView?
  - Or sites are happy with a view of Nagios tests (it probably depends on the outcome of the discussion above)?

- Availabilities base od SAM tests are today calculated by various means
  - ACE for MB monthly reports
  - SAM Dashboards for weekly (SCOD) reports
  - There should be a unique engine for availabilities. This will be ACE
    - Nothing to be discussed here, just informational
- SAM visualization: today we have different tools
  - MyWLCG: the native SAM visualization portal
    - Covers functionality of GridMap, GridView and SAM portal
  - SUM Dashboard: adaptation of the previous SAM Dashboard
  - Do we need both? Sinergies, overlaps, missing functionalities?

- Network problems today are very difficult to identify and to cure
  - We discuss “identify” here (monitoring)
- **(OP)** Perfsonar(PS) has proved to be very valuable
  - Latency + Throughput
  - Should we push for its deployment at every T1 and T2 (at least)? MHO is YES.
  - In this case we need (again) coordination
    - Someone needs to follow actively the deployment
    - Someone needs to decide on frequencies vs topology etc ..
  - How do we visualize? E.g. BNL today provides a dashboard, is it enough?
- The new FTS monitoring would be complementary
  - Profiling of transfer statistics



- Very few services come with native monitoring
  - Proliferation vs missing functionality
  - Both for fabric monitoring (probes) and activity monitoring (FTS for example)
- (OP) Requirements on the middleware
  - Provide generic service probes (already mentioned)
  - Improve logging to facilitate development of new probes (sites need to provide concrete examples)
- (OP) Do we need a general service monitoring (like “SLS for WLCG”)? Can this be MyWLCG?
  - Should this include experiment’s central services? See discussion above (SAM and services in GOCDB)



- Generally understood that monitoring should be a service.
- (OP) Should we formalize it for various areas?
  - Development:
    - Keep backward compatibility in APIs
  - Integration:
    - Provide a preproduction/test instance
  - Deployment:
    - Try to minimize impact of interventions
    - Provide an infrastructure properly sized
  - Operations:
    - Announce downtimes, Produce SIRS
- (OP) How do we treat experiment central services?
  - Should we publish their downtimes? Most think YES
  - Do we need an IT Status Board equivalent for experiment central services?

- Several question raised on exposing site internal monitoring
  1. History information of running/pending jobs
  2. Average HEPSEPCs per core
  3. Dynamic fair-share
  4. Tape systems
  5. Non-grid activities
- Controversial discussion
  - Some of those (e.g. 2.) are exposed via Information System, but numbers many times are not correct
  - Sites are not happy to grant direct access to core services e.g. batch system head nodes
  - Some of those (e.g. 3.) are difficult to provide/interpret
- (OP) Still... experiments believe they would be beneficial. How do we proceed?