



Analysis support towards STEP09

*Follow up of the analysis support session at the
WLCG Workshop – Praha March 2009*

Massimo Lamanna and Dan van der Ster





Message from WLCG Workshop

- Started an “inventory” of useful tools/strategies
 - Lots of similarities and complementary approaches
- Example of tools
 - Shift system (e.g. ATLAS)
 - Job robots (e.g. CMS)
 - Site commissioning and monitoring
 - Stress generators (e.g. ATLAS)
 - Detailed studies of site and code performances
- Analysis support essential for 2009/2010 data taking
 - Learn from each other (experiments and IT/GS)
 - Metrics to evaluate the user support



Conclusions

- In the framework of STEP09 IT/GS will start fostering
 - Further exchange across experiment and IT/GS in the area of Analysis Users Support
 - Complete the inventory of useful tools/strategies
 - Explore the possibility to extend the usage of tools (like robots and stress tests)
 - Devise sensible metrics to evaluate the user support

More material





STEP09

- Concurrent Real(istic) computing activities
 - Data distribution ($T_0 \rightarrow T_1 \Rightarrow T_2 \Rightarrow \dots$)
 - Reprocessing
 - Analysis (include user-related data management)
- Analysis:
 - Can a realistic number of users perform distributed analysis?
 - Metrics: #users, #jobs, data access patterns etc...
 - Can we support all users?
 - Metrics: ?



Analysis support

- Tools
 - Communication:
 - Mail, Twiki, HNews, eGroups, Savannah, Calendars, eLog, ...
 - Infrastructure status:
 - GocDB, BDII, experiment-specific configuration
 - Experiment dashboards
 - Ideal aggregation points
 - “Site” views exist (should be improved)
- Teams
 - Developers, advanced users (user2user), shifters
- Services
 - Experiment SAM
 - Robots (JobRobot, GangaRobot)
 - Stress Generators (HammerCloud)



Communication tools

- All essential, but address different areas
 - (Mail) - HNews – Egroups
 - Heavily used
 - Good for allow/encourage user2user support
 - Weak for more formal problem handling
 - Savannah (bug and task tracker)
 - Everything is tracked. Just for developers?
 - Elog (Log book functionality)
 - Appealing
 - Frequent request to extend it beyond the original scope
 - E.g. Threading comments, alter eLog entry...

WLCG status and config



- Information not in a single place
- Central part
 - BDII, GocDB, ...
- Experiments usually maintain their part
 - And aggregate info from the infrastructure area, from their own system. Both area are characterised by static and dynamic quantities

Teams



- Developers
- Advanced users (user2user)
- Shifters

Services (1/3)



- SAM (“Atomic tests”)
 - Essential to run experiments test
 - Factorise real activities, e.g. $CE(UP)$ or $CE(UP)*LFC(UP)$...
 - Effort started by CMS to display SAM test in a useful way for experiments experts to tackle problems
 - Generalised version from ATLAS
 - Keep and maintain it (VO is a “parameter”)



Services (2/3)

- Functional tests (inclusive tests)
 - (CMS, JobRobot), (ATLAS, LHCb, GangaRobot)
 - Essential that the tests are realistic (application and data)
- Components:
 - *cron” * “exp-specific config” * “task execution”
 - Similar to all SAM tests
 - More sophisticated on result extraction compared to std SAM
 - and more “fragile” operations due the complexity of the tasks and their analysis
 - Error detection and classification is essential
- Keep it and try to exploit commonality (similar framework, different applications)
 - Common operations possible (if procedures are in place)

Services (3/3)

- Distributed analysis stress tests
 - ATLAS has developed HammerCloud
 - Immediately usable by LHCb (Ganga)
 - CMS has a similar system in their plan (opportunity to share?)
 - ALICE is running similar test (with a lot of additional info)
- Main idea:
 - Book a test (a site, a group, the entire infrastructure)
 - Not necessarily centralised (a site can ask to check itself)
 - Emulate/complement users load
 - Measure the performance (various metrics)



More on DA stress tests

- Ultimately a DA stress test tells:
 - How many evt/s a site can digest (analysis) in real(istic) conditions
 - How much is the efficiency for users jobs in real conditions
 - Reveal bottlenecks in hw or configurations (network, storage, etc...)
 - ALICE produces the most detailed information (analysis of the running jobs – memory etc..)
 - All this is essential for commissioning (Experiment and WLCG)



More on DA stress tests

- Several use cases:
 - Moving to more and more realistic jobs
 - mixture of different type of analysis
 - Data base access
 - Performance measurements, DB services fine-tuning
 - Use as a tool to validate new software when a testbed cannot be used
 - E.g. Data access (via pilot) --> new version validated by coupling the new version to a set of functional tests. Essentially self contained. Normal activities essentially independent.