Post-TEG Working Groups

Ian Bird

GDB

13th June 2012

WG’s, teams, fora needed

- DM&S:
  - Federation
  - Benchmarking
  - Networking (as from MB)
- WLM
  - Definition of extensions (1 pre-GDB mtg to formalise reqs – Davide)
  - Information system
- Security
  - What exactly??
- Database
  - None specifically
- Operations
  - m/w sw process (sw lifecycle, deployment, testing, rollback, etc.)
  - Monitoring – overview/integration

- Teams:
  - Operations coordination team
- Fora:
  - Sharing experiences:
    - NoSQL/Hadoop etc
    - ???
  - Technology watch
  - ??? How?

Working groups v2

- WLCG Operations Coordination team - Maria
  - long term - operational and deployment issues, (glexec, SHA-2, de-WMS)
  - mandated to require sites to do xxx?
- Storage Interface wg – Wahid/Markus
  - defines all needed interfaces to storage systems; for data management, transfer, querying, monitoring, accounting, etc.
  - fixed term wg
- Data federation wg – Markus
  - This is more information sharing of work in progress?
- Benchmarking wg - Dirk
  - proposal for benchmarks + plan for follow up and measurements
- Networking - Michael Ernst
  - longer term network overview, etc -
- CE extensions - Davide
  - short term wg on
  - defining required extensions to CE, inc batch system support, pilot job support, etc
- Information system – Maria A.
  - define plan
  - follow implementation
- Monitoring wg - Ian
  - Overview of all monitoring activities- define monitoring plan - ensure coordination
- Middleware lifecycle process definition - Ian
  - short term working group - defines process
- Traceability - Romain
  - short term wg
- ID federation pilot - Romain
  - define pilot
  - follow implementation
- Risk mitigation plan - Romain
  - short term wg follow up on risk assessment
- Discussions:
  - Cloud
  - Cloud policy - discuss with HN, HEPiX, etc
  - Batch systems - experiences and etc (overlap HEPIX?)
  - NoQSL/Hadoop etc
  - Technology developments/technology watch

Benchmarking etc

- During the discussions of the Data and Storage Evolution Group several shortcomings in the area of collecting and reproducing realistic workloads for benchmark and optimization purposes have been identified:
  - 1) the real aggregate I/O access pattern against WLCG SEs is not easy to quantify or to reproduce
  - 2) sites, experiments and software providers use a variety of tools to address performance optimization and resource planning this including root scripts, HammerCloud, OS level I/O benchmarks
  - 3) the existing tools do not necessarily use a common approach to define the key metrics nor are benchmark codes and results centrally available from a managed repository.
  - 4) not all benchmarks can be scaled to run in multi-client mode to obtain the performance of a fully loaded server.
  - 5) in many cases the actual type of access (eg sparseness vs sequential, WN local, site local, WAN federated) is either not documented or not adaptable at the potentially changing access approaches of the experiments.
- We propose to setup a small (<5 people) working group to perform a “market survey”, documenting agreed key metrics, existing tools, pointing out areas where more coherence could be obtained. The document should describe a systematic approach for the different main use-cases for performance analysis using existing tools:
  - 1) optimization of existing or planned site installations with respect to an expected I/O workload (eg CPU vs Network vs RAM vs SSD vs Disk cost)
  - 2) optimization of experiment I/O layer wrt to local and federated data access
  - 3) optimization of SE implementations wrt to an expected I/O load
  - 4) determination of aggregate I/O patter of a real job population in order to obtain realistic parameters for 1-3) and in order to identify changes of the real I/O over time.
- The latter task should involve a survey of the existing monitoring information (from sites & experiments) wrt to key metrics, which would help to validate existing load generators against measured I/O load. It should also investigate the option of logging and replaying I/ O patterns in order to create easily deployable workload generators without dependency on experiment software frameworks. Expected duration 3 months with first GDB report after 1 month. Expected minimal contribution during the project 0.3 FTE per person.

WM: CE Extensions

- Why: help experiments getting the type of resources they need - and help sites satisfying their requests.
- How to proceed:
  1. Define scope of CE extensions
     - Multi-core support (start with this)
     - Streamed submission
     - I/O vs. CPU tagging
  2. Agree on implementation and testing plans
- Proposal: next pre-GDB day (July 10) dedicated to discuss details for multi-core support.
- Who: site and experiment representatives, CE developers, LRMS experts.

From the WM TEG Report

- Specify whole-node / multi-core requirements in the job description:
  - Request whole node or not
  - Request a fixed # of cores
  - Request a variable # of cores (e.g. min and/or max)
  - Request total memory (or per-core memory if # of cores is variable)
  - Consider support for multiple batch systems

Presented at the GDB, Feb 2012

Whole node / multi-core: proposed changes and workplan

- We need to define a new JDL attribute for memory and implement requests for a variable # of cores.
  - CREAM developers agreed to implement this (manpower estimate one week) – to be formalized with EMI.
  - The GRAM job manager should be easily modified (as long as the underlying LRMS supports the feature). Need to investigate with the Globus team; if Globus not interested, the OSG software team can write a patch.
- Testing, two phases:
  1. Experiments will require either whole nodes (without dedicated queues), or an exact number of cores using JDL/RSL. We will start with # cores = 4 to make things simpler.
  2. Experiments may also require a variable # of cores; the job will be able to utilize as many cores as are made available.
     - Need to define an environment variable telling the job how many cores / how much memory the job has been allocated; will build on the proposal made last year by the HEPiX-virt WG.
- Sites / experiments:
  - CMS will modify WMagent / glideinWMS to support multi-core scheduling.
  - Potentially all sites currently supporting whole node testing for CMS plus GRIF will join the test.
- Information system: will need to define a way for sites to flag the max # of cores they support and whether they support whole nodes, and/or generic multi-core requests.