Abstract

Automated distributed analysis tests are necessary to ensure smooth operations of the ATLAS grid resources. In this work we present the recent developments of the ATLAS GangaRobot, a tool designed to perform regular tests of all grid sites by running arbitrary user applications with varied configurations at predefined time intervals. Specifically the GangaRobot creates and submits several real ATLAS user applications to the various grid sites using the distributed analysis framework GANGA, a front end for easy grid job definition and management. Success or failure rates of these test jobs are individually monitored. Test definitions and results are stored in a database and made available to users and site administrators through a web interface, the ATLAS Site Status Board (SSB) and the Service Availability Monitor (SAM). The test results provide on the one hand a fast way to identify systematic or temporary site problems, and on the other hand allow for an effective distribution of the workload on the available resources.

ATLAS data analysis

Requirements:
- Large data volume: several Petabytes/year
- High demand of computing resources: >100k CPUs
- Data availability 24/7 at high bandwidth

The ATLAS computing model: Job goes to data

Typical users analysis:
- Athena analysis on Monte Carlo (MC) and collision data, using various types of input: AODs, DPDs, ESDs
- Tag-based analysis, which require direct access to data using a metadata index
- Calibration and alignment studies, requiring access to the conditions database
- Private MC production
- ROOT analysis.

Distributed analysis in ATLAS

Grid computing:
- Centralized MC production and data reconstruction
- Centralized data management by DDM/D22 Tools
- Decentralized distributed data analysis
- Grid infrastructure
- EGEE, NodGrid, Open Science Grid
- Clients for job submission:
  - pathera (Panda client), GANGA

HammerCloud/GangaRobot framework

Automated distributed analysis tools are necessary to ensure smooth operations of the ATLAS grid resources. In this work we present the recent developments of the ATLAS GangaRobot, a tool designed to perform regular tests of all grid sites by running arbitrary user applications with varied configurations at predefined time intervals. Specifically the GangaRobot creates and submits several real ATLAS user applications to the various grid sites using the distributed analysis framework GANGA, a front end for easy grid job definition and management. Success or failure rates of these test jobs are individually monitored. Test definitions and results are stored in a database and made available to users and site administrators through a web interface, the ATLAS Site Status Board (SSB) and the Service Availability Monitor (SAM). The test results provide on the one hand a fast way to identify systematic or temporary site problems, and on the other hand allow for an effective distribution of the workload on the available resources.

ATLAS data analysis

Requirements:
- Large data volume: several Petabytes/year
- High demand of computing resources: >100k CPUs
- Data availability 24/7 at high bandwidth

The ATLAS computing model: Job goes to data

Typical users analysis:
- Athena analysis on Monte Carlo (MC) and collision data, using various types of input: AODs, DPDs, ESDs
- Tag-based analysis, which require direct access to data using a metadata index
- Calibration and alignment studies, requiring access to the conditions database
- Private MC production
- ROOT analysis.

ATLAS data analysis

Requirements:
- Large data volume: several Petabytes/year
- High demand of computing resources: >100k CPUs
- Data availability 24/7 at high bandwidth

The ATLAS computing model: Job goes to data

Typical users analysis:
- Athena analysis on Monte Carlo (MC) and collision data, using various types of input: AODs, DPDs, ESDs
- Tag-based analysis, which require direct access to data using a metadata index
- Calibration and alignment studies, requiring access to the conditions database
- Private MC production
- ROOT analysis.

Results

The site efficiency on functional tests is defined as:

\[
\text{site efficiency} = \frac{\text{# successful + # failed + # running}}{\text{# total}}
\]

Automatic blacklisting for EGEE sites

GangaRobot results are used to automatically prevent user analysis jobs to be submitted to temporarily failing sites. If the efficiency drops below 80% in the last 24 hours, the site is blacklisted.

A summary of the efficiency per site is displayed on the HammerCloud/GangaRobot website.

GangaRobot results are exported to SAM and the ATLAS SSB and provide information to evaluate site ranking and availability as a function of time. In general, most EGEE sites have availabilities larger than 90%. Remarkably only a handful of sites have availabilities smaller than 80%. Problems at a site are normally of transient nature, and are solved in a couple of days. Note that scheduled site downtimes are also included in the plots.

Conclusion

The new HammerCloud/GangaRobot framework is running since May 2010. It provides an easy interface for the creation and submission of functional/stress tests. Functional test results are stored in a database, published on the HammerCloud website, and exported to further Grid information services such as SAM and the ATLAS SSB. The site efficiency on functional tests is used to blacklist sites temporarily failing the tests.