Experimental Evaluation of Job Provenance in ATLAS Environment

Atlas Job Processing

Production computations of the Atlas experiment (ProdSys [1], Fig. 1) consist of tasks. Each task applies a transformation on a dataset (collection of files). A task is split up into jobs computing the transformation on smaller number of files.

Fig. 1. Overview of the whole architecture of the Atlas production system.

Tasks are stored in Production Database (ProdDB), which reflects the tasks-job structure of computations (Fig. 2). They are managed by job execution and submitted into various environments by executions.

Fig. 2. Main structure of Atlas ProdDB

Custom job catalogue development takes considerable effort. Solutions like Atlas ProdDB are very purpose-specific.

Data Model

Atlas-specific JP jobs are shown in Tab. 1 (namespace prefix is omitted). Unlike in ProdB, the JP data model is flat, there is no hierarchy of entities, jobs are the only ones. Any detailed information on jobs (including the logical relationship w.r.t. tasks) is expressed in terms of JP job attributes (shown in Tab. 1). Despite of its clear limitations, we argue for the feasibility of the model:
- the hierarchy (task - job - jobset - jobexec) is still present in a logical view.
- the logical model can be substituted without changing JP concepts and implementation.
- the principal problem of the flat model, replication of task-level attributes with each job, can be avoided by storing tasks as “virtual jobs” only once.

| Tab. 1: Attributes specific for Atlas jobs. |
| taskid | Atlas internal task identifier |
| taskname | Human-readable task name |
| jobid | Job definition identifier |
| jobsetid | Job execution attempts identifier |
| inputDatasetname | Input dataset name |
| parset | Data partition number |

Service Setup

The experimental setup consists of the following services:
- Lexar and gLite WMS (egee-rc-01.mi.infn.it, Atlas production).
- Lexar is instrumented to record the specific JP attributes:
  - *values known at job submissions are stored in JDL.
  - *values known at job run-time or completion are recorded as LB user tags [2].

Dedicated L&B server (skurf68-1.cern.ch, EGEE JRA1 Preview Testbed)
- *setup specifically to avoid interference with other traffic
- *configured to export data into JP

JP Primary Storage (unisol.rsmui.cn, EGEE JRA1 Preview Testbed)

JP Index Server (skurf68-2.cern.ch, EGEE JRA1 Preview Testbed)
- *Besides all attributes in Table 1, this JP also retrieves system attributes listed in Tab. 2.
- *Retrieved jobs are restricted to Atlas jobs identified by existence of Atlas taskset attribute (submitted in August 2007). New deployment would include further restriction on tasks, datasets, etc.
- *We use just one JP instance for demonstration. However, JP design encourages setup of more JPs over the same JPPS, with different configurations (retrieved attributes, conditions on jobs, e.g., sets of tasks) according to different user needs.

| Tab. 2: Additional system attributes gathered by JP (namespace prefix is omitted) |
| user | Job owner |
| finalStatus | Final job status (Done, Aborted, Cancelled) |
| LRMSSDoneReason | Reason of the status |

Test Results

On-line measurements were done on Atlas production jobs during two weeks in August 2007. All the involved services sustained the load without stability problems or observable congestion. Stress tests (feeding JP at the maximal possible speed) achieved approx. 10x higher rate on the same set of jobs. Finally we repeated the stress test with a larger number of jobs extracted from ProdDB. The achieved higher rate was expected due to missing LB tag in this case. The main result is the rate sustainability for a realistic number of jobs.

| Tab. 3: On-line test |
| Number of jobs | 14,079 |
| Duration of test | 12 days |
| Job arrival rate | 1,110 jobs/day |

| Tab. 4: Stress tests arrival rate |
| Full jobs (14,079) | 147,000 jobs/day |
| Older jobs (560,000) | 663,000 jobs/day |

gLite Job Provenance Service

Job Provenance (JP) service keeps long-term track of execution of Grid jobs, as well as provides mechanisms for job annotations [3], and data mining capabilities.

Fig. 3. gLite job submission and data flow into JP

There are two classes of components (Fig. 3): JP Primary Storage (JPPS) and JP Index Server (JPS). JPPSs are installed in a few instances only (e.g., one for the whole Atlas experiment), keeping all the job data permanently in a storage efficient way.

On the contrary, JPSs provide indexed views on the data according to particular (and evolving) user needs. JPSs are created semi-dynamically and they are populated with a fraction of the primary data only.

Data enter JP via direct user tags (name = value pairs) or file uploads (the letter including whole job record from LB). All these data are queryable in the form of JP attributes = namespace name = value pairs assigned to jobs (files are processed by plugins to extract the attributes).

JP is a generic job catalogue service

User Interface

Graphical user interface of the application is shown below. Jobs are grouped in cells of a 2D array according to datasets they process and Computing Element where they ran. The arrangement is an example only, not being fixed, different views (e.g. tasks vs. week/month of execution) can be easily added. This specific application is not the final outcome of our work. It was developed for the JP capabilities demonstration (it needed only one-moth of the programming from scratch, same level of effort needed to develop different application oriented views).

| Each node corresponds to a processed dataset, each column to a Computing Element
| Job success and failure is shown by color
| Clicking on a cell selects it for detail display on the right side
| Switching to application-specific and user-annotation view is supported

Non-gLite jobs

All the described experiments were done with gLite jobs (i.e., Atlas LCG executor only). However, JP is designed to store data on arbitrary jobs, hence allowing easy transition to other grids. Details are given in [3].

Conclusions

- Job Provenance is mature service able to process production data with a large throughput margin
- Using JP, job catalogues can be built with moderate effort compared to custom solutions

References


JP deployment for Atlas experiment is supported by EGEE II project funded by European Commission (contract number INFSO-RI-031688).