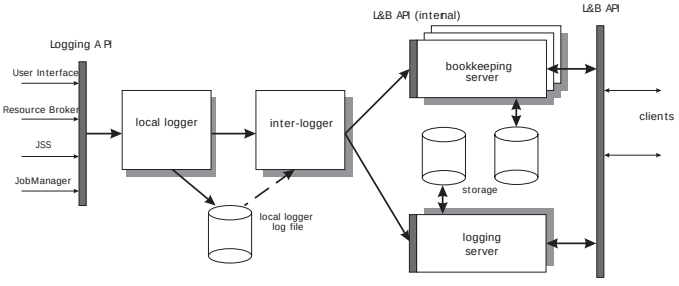
The Grid Job Monitoring Service

Luděk Matyska[1], Aleš Křenek, Miroslav Ruda, Michal Voců, Zdeněk Salvet, Jiří Sitera,
Jan Pospíšil, Daniel Kouřil
CESNET, z.s.p.o., Zikova 4, 162 00 Prague, Czech Republic
Firt_name.Last_name@cesnet.cz

Job tracking, i. e. monitoring its behavior from submission to completion, is becoming very complicated in the heterogeneous Grid environment. Use of advanced Grid services like Resource broker (Metascheduler) means that users lose direct control over where jobs are submitted. Security issues may prevent direct contact with computing element where job is running, further complicating the monitoring of its state. Additional complexity accompanies parallel and multipart jobs as well as job assemblies, where up to several thousand tasks may represent one logical job. It is practically impossible to monitor all the individual tasks manually. While monitoring in general is an on-going activity within the Grid community (see, e. g. [1]), there is no such activity specifically focused on job tracking and monitoring. Within the DataGrid project we are developing a *Logging and Bookkeeping Service* [2] focused on this task.

During its lifetime, the job may pass through several grid components — user interface, resource broker, submission service and the computing element itself. The L&B service collects important *events* in the job life (e. g. a matching resource found, job execution started, etc.), delivers them to a L&B server and stores them in a reliable way. In general an event indicates a change in job *status*. From the user's point of view the folowing states are recognised: SUBMITTED (job just entered the Grid), WAITING (resource discovery), READY (job transfer to the selected computing element), SCHEDULED (waiting in a local queue), RUNNING, and DONE (completed successfully) or ABORTED (terminated abnormally). CLEARED is reached whenever user retrieves all the output files produced by a job. Finally, the job may be also checkpointed for later restart (CHKPT). The described job states (in that order) form a finite state automaton of a basically linear structure. Backward transitions are possible as well, e. g. from READY to WAITING: a resource has been chosen based on data provided by the Grid information service, however, the resource refuses to accept the job on actual attempt to put it into local queue.

The L&B service architecture (in the figure) features two APIs, local logger subservice and the servers. Events generated by the Grid components enter the L&B system through the *logging (producer) API*. The API, implemented as a simple library, provides locally persistent non-blocking calls, regardless of the state of other L&B components.



A message format based on the ULM standard [3] and semantic rules are prescribed, and the event passed to the local logger subservice. This part, composed of *local-* and *interlogger* daemons, resides as close to the event source as possible, preferably at the same machine. Local persistence is implemented with a transaction log file written by the locallogger. The interlogger is responsible for actual transfer of messages to the L&B databases, typically over WAN. At this point the data stream is forked, all the detailed data required for job monitoring during its life are delivered to a *bookkeeping server* while certain important events are sent to a *logging server* for long time storage.

[1]Corresponding author, tel: +420541512310, e-mail: Ludek.Matyska@cesnet.cz

The L&B servers manage the persistent data storage. The event is the principal storage entity, job status is computed on demand (client request) by analyzing the events related to the job according to the defined job life cycle finite automaton. Clients query the L&B server with a *server (producer) API*. The core functions of the API return a list of jobs owned by the user, job status of a given job, and list of events related to the job.

Jobs are identified by a unique grid job identifier. It follows URL syntax `https://hostname:port/unique_string?...`. The `hostname` is a DNS name of the bookkeeping server machine, where the server listens on the specified `port`. The "unique string" part is generated upon job submission according to rules ensuring a reasonable probability of uniqueness (IP address of the user interface, submission timestamp and a random number are all included). The syntax allows to retrieve virtually any information on the job by direct access of the bookkeeping server, without the need to contact another grid service. In fact, the bookkeeping server speaks a subset of the HTTPS protocol, information on job status is obtained with its GET request. Currently, an address of the Resource Broker that handles the job is encoded in the job ID in a similar way (after the question mark).

In the hostile Grid/Internet environment security considerations are of utmost importance. Logging events with faked job ID might cause serious confusion in the L&B system. The job state information and associated events may be a source of valuable information which must be prevented to fall into improper hands. Therefore all the communication between L&B components is authenticated with either user or service certificate. The L&B server associates the job ID with a user when the first user-authenticated event of the job is logged. This association is checked on any further events, preventing thus to store new events without possession of a valid certificate. All the queries are also authenticated and currently only information about own jobs (i. e. jobs whose ID is associated with the submitted certificate) is provided.

The L&B system just described is a part of the resource management subsystem for DataGrid project currently used and evaluated on the testbed. Even the first experience leads to extensions and new functionality of the L&B system. The most important issues are notification, user attributes and dynamic and multiple jobs support. Using the *notification* system users will be able to register, via a GMA provided environment, for a specific event or job state change. The L&B server is beginning to provide a (near) real-time processing of events, keeping the actual job state in a cache. *User attributes* allow job annotations, to be used later as a part of a complex query like "Give me the actual state of all my jobs belonging to the experiment X".

Multiple and dynamic jobs, being composed from many (often independent) tasks running on different computing elements, are common practice in computational science. To keep track of the whole herd, the *group ID* is being introduced and the L&B APIs extended to support aggregate queries on the whole group. The aggregation goes through individual states and provides more compact view on the actual multiple job state.

This way the logging and bookkeeping service is becoming vital part of the Grid environment, providing a bridge between the general job submission and monitoring services.

References

[1] B. Tierney et al: *A Grid Monitoring Architecture*, Global Grid Forum Working Draft, `http://www-didc.lbl.gov/GGF-PERF/GMA-WG//papers/GWD-GP-16-2.pdf`

[2] D. Kouřil et al: *Logging and Bookkeeping Service for the DataGrid*, EU DataGrid project document, `http://server11.infn.it/workload-grid/docs/DataGrid-01-TEN-0109-1_0.pdf`

[3] D. Gunter, B. Tierney, B. Crowley, M. Holding, J. Lee: *NetLogger: A Toolkit for Distributed System Performance Analysis*, Proceedings of the IEEE Mascots 2000 Conference, August 2000