GRID OBSERVATORY
TECHNICAL DOCUMENTATION
VERSION 1.3

AUTHORS

Cécile Germain-Renaud

Alain Cady

Julien Nauroy

DATE **2011-03-11**

TECHNICAL DOCUMENTATION
Version 1.3

Date : 2011-03-11

The Grid Observatory is an open collaboration involving researchers from

- Laboratoire de Recherche en Informatique (LRI) and Laboratoire de l'Accélérateur Linéaire (LAL)
- Imperial College London
- Università degli studi del Piemonte Orientale Amedeo Avogadro

The Grid Observatory is supported by

- The European Grid Initiative (EGI)
- France-Grilles, CNRS Institut des Grilles and GDR ASR
- Paris-Sud University
- Regional Council of Ile de France and DigiteoLabs

TECHNICAL DOCUMENTATION
Version 1.3

Date : 2011-03-11

TABLE OF CONTENTS

TECHNICAL DOCUMENTATION
Version 1.3

Date : 2011-03-11

1. INTRODUCTION

1.1. PURPOSE

This document presents the traces provided by the Grid Observatory. The goals of the document are:

- to provide a basic introduction of the role and thus possible use of each trace for scientific research; this very introductory material is not a substitute for the documentation produced by the EGI project;
- to describe the format of the traces whenever possible.

This documentation will continue to evolve with the project.

1.2. DOCUMENT ORGANIZATION

The document contains a short introduction followed by a description for each trace. The next section describes the current sources of traces. The examples follow in the annexes.

1.3. REFERENCES

G.1	The gLite user Guide https://edms.cern.ch/file/722398/1.2/gLite-3-UserGuide.html
G.2	E. Laure et al, Programming the grid with gLite http://doc.cern.ch//archive/electronic/egee/tr/egee-tr-2006-001.pdf
G 3	EGEE Glossary http://egee-technical.web.cern.ch/egee-technical/documents/glossary.htm
IS.1	S. Andreozzi at al. The Glue schema specification Version 1.3 http://forge.cnaf.infn.it/plugins/scmsvn/viewcvs.php/*checkout*/v_1_3/spec/pdf/GLUESchema.pdf?rev=49&root=glueschema
IS.2	OSG Glue Working Group http://forge.ogf.org/sf/projects/glue-wg
IS.3	https://twiki.cern.ch/twiki//bin/view/EGEE/BDII
LB.1	The LB user guide http://egee.cesnet.cz/cvsweb/LB/LBUG.pdf
LB.2	Ludêk Matyska et al. Job tracking on a grid-the Logging and Bookkeeping and Job Provenance services. Technical Report 4/2007, CESNET, 2007. http://www.cesnet.cz/doc/techzpravy/2007/grid-job-tracking/
LB.3	Logging and Bookkeeping Administrator's Guide http://egee.cesnet.cz/cvsweb/LB/LBAG.pdf
BS.1	The PBS administration guide (Section 10.12.5)
BS.2	The parallel workload archive http://www.cs.huji.ac.il/labs/parallel/workload/l_lpc/pbs2swf/
RTM.1	The Real Time Monitor site http://gridportal.hep.ph.ic.ac.uk/rtm/

Table 1: Table of references

1.4. COMMENTS AND CONTACT

Comments and suggestions are welcome at contact@grid-observatory.org

2. OVERVIEW

The EU project EGI-inSPIRE operates a production infrastructure originating in the grid developed in the Datagrid and EGEE-I, -II and –III projects. For simplicity, this grid will be called the *EGI grid* in the following.

2.1. PRESENTATION

There are two categories of traces. The Real Time Monitor traces are a summary of the lifecycle of jobs. All jobs managed by the gLite middleware are reported. The other traces have been collected at the GRIF-LAL site of EGI. Depending on their nature, they are more or less exhaustive. See the summary documentation for more details. At this date, the portal provides only "raw" trace, i.e. an ASCII version of the logs as they are generated by the various gLite components. Defining the adequate organization and indexing of the datasets includes provenance issues at the operational level and ontology issues at the fundamental level, and is one the goals of the project. The database will be updated by extension, mostly weekly.

To download traces, go to the query page. Access to the documentation, including excerpts does not require registration. To access full data, you have to fill the questionnaire. If you plan to exploit these traces for publications, please have a look at the rules of usage.

This guide covers only the traces collected at GRIF/LAL.

2.2. OVERVIEW OF GLITE

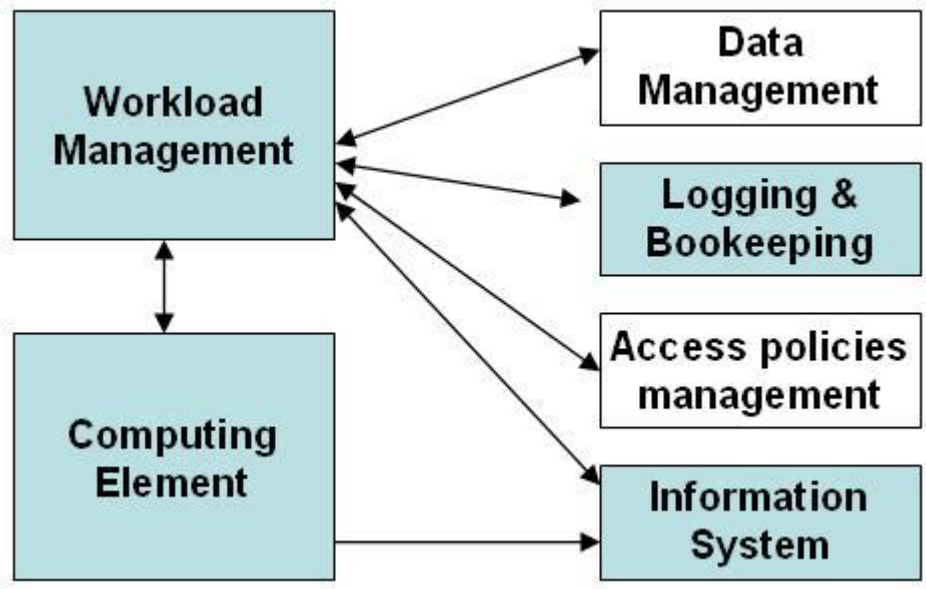

Figure 1. gLite architecture

EGI combines globally-distributed computational and storage resources into a single production infrastructure available to EGI users. Each participating site configures, runs, and maintains a batch

system containing its computational resources and makes those resources available to the grid via a gatekeeper.

The gLite middleware deployed on the EGI infrastructure integrates the sites' computing resources through the Workload Management System (WMS). The WMS is a set of middleware-level services responsible for the distribution and management of jobs. The site computational resources present a common interface to the WMS, the Computing Element (CE) service.

The core of the WMS is the Workload Manager (WM) which accepts jobs from users and dispatches them to computational resources based on the user requirements on one hand, and the characteristics (hardware, software, localization) and state of the resources on the other hand. The WM is implemented as a distributed set of resource brokers, with some tens of them currently installed; all the brokers get an approximately consistent view of the resource availability through the grid information system.

The Logging and Bookkeeping service (LB) tracks jobs in terms of events (important points of job life, e.g. submission, finding a matching CE, starting execution etc.) gathered from various WMS components as well as sites.

3. THE INFORMATION SYSTEM

3.1. DESCRIPTION

The Information System provides detailed information to the other services about the static and dynamic status of the grid infrastructure and services. The content of the information system is the Glue information model (Glue 1.3 [IS.1], and OSG Glue WG. [IS.2]), which is the current basis for interoperability between the EGEE grid and other grids.

The schema including tens of attributes, the reader is referred to [IS1] for a detailed description of them. To limit the size of the log files, the attributes related to services have been omitted. We plan to release integral logs in the future. See Appendix A for the list of available attributes.

The CE specification is one of the core parts of the Glue information model. The Computing Element (CE) provides the abstraction of a computing resource, typically a batch queue of a cluster but can also describe supercomputers or even single workstations. The schema includes: information related to the job behaviour, e.g. about the number of waiting or running jobs in this queue; information related to the policy regarding this queue, e.g. limits on the number of jobs; and finally estimates about the response time, i.e. the waiting time a job would experience if using this queue. While the first two categories of information are factual, it must be stressed that the third one is an estimate, provided under the responsibility of the site, not a measurement.

Remark: In the context of middleware development (e.g. CREAM CE, LCG CE), or in the description of the lifecycle of a job, the term *Computing Element* has a very different interpretation. It describes all the functions that are the interface between a site and the common services (typically the WMS): running jobs, staging the files required by the job, providing information about resource availability, and notifying the WMS of the job-related events.

3.2. SCOPE

The information system is conceptually unique, even if its implementation (the BDII, [IS.3]) is distributed. Thus the traces cover the whole EGEE infrastructure.

3.3. FORMAT

The Information System trace uses LDIF (LDAP Data Interchange Format), which is the standard ascii format for representing LDAP directory contents.

Each day, a base file (`.lis` suffix) is created, and the diff file (`.diff` suffix) with this original file is recorded each 15 minutes.

To recover the complete files, use the `patch` command.

For instance, with `GRIF-LAL:20080730-00h00:bdii.lis` and `GRIF-LAL:20080730-15h30:bdii.diff` as the original files

```
patch --output=GRIF-LAL:20080730-15h30:bdii.lis < GRIF-LAL:20080730-15h30:bdii.diff
```

produces the the `GRIF-LAL:20080730-15h30:bdii.lis`, which is the LDIF representation of the Information System at 2008-07-30, 15h30.

3.4. MORE INFORMATION

The operational reference about the Glue information model is [IS.1].

4. THE LOGGING AND BOOKKEEPING

4.1. DESCRIPTION

4.1.1. Overview

The Logging and Bookkeeping (LB) Service is the Grid service responsible to store and manage logging and bookkeeping information generated by the various components of the Workload Management System.

The Logging and Bookkeeping service logs most of the events in a jobs lifecycle, as provided by the various services of gLite. The service is job-centric: any event is assigned to a unique Grid job. Upon creation each job is assigned a unique, virtually non-recyclable job identifier (JobId) in an URL form, which is used in all logged events. The events are processed to give a higher level view on the job states (e.g. Submitted, Running and Done when the jobs starts and stops execution, or Transfer from a WMS component to another one), and records various attributes (e.g. submission file in JDL language, destination CE name, job exit code, etc.).

Logging and bookkeeping is based on a push model, whereby the various WMS components and the Computing Element actively send suitable messages to the Logging and Bookkeeping Service whenever certain events occur. This process is highly reliable, within the limits of an asynchronous implementation.

4.1.2. The lifecycle of a job

Figure 2 gives the transition diagram of the jobs states. Figure 3 shows the different services involved in the actual job management.

The interpretation of the states is as follows

- Submitted: A job is submitted through a User Interface. Information about the job properties and user requirements are expressed in JDL language
- Waiting: the WMS has been notified of the job
- Ready: a matching CE has been discovered
- Scheduled: the site (gatekeeper) has received the job
- Running: the job is executing
- Done: the job execution is complete
- Cleared: the output files (if any) associated to the job have been downloaded (this refers to the “output sandbox” scheme).

Along this path, a job may be voluntarily cancelled by the user, or may abort, because the middleware was not able to run it.

A job state transition is recorded in the LB database as one or two events generated by WMS components. A single event is sufficient when the transition is within a component. When the state transition means also moving the responsibility for the job management between two components two events are recorded, one generated by the component that passes the job on and one generated by the component that accepts the job.

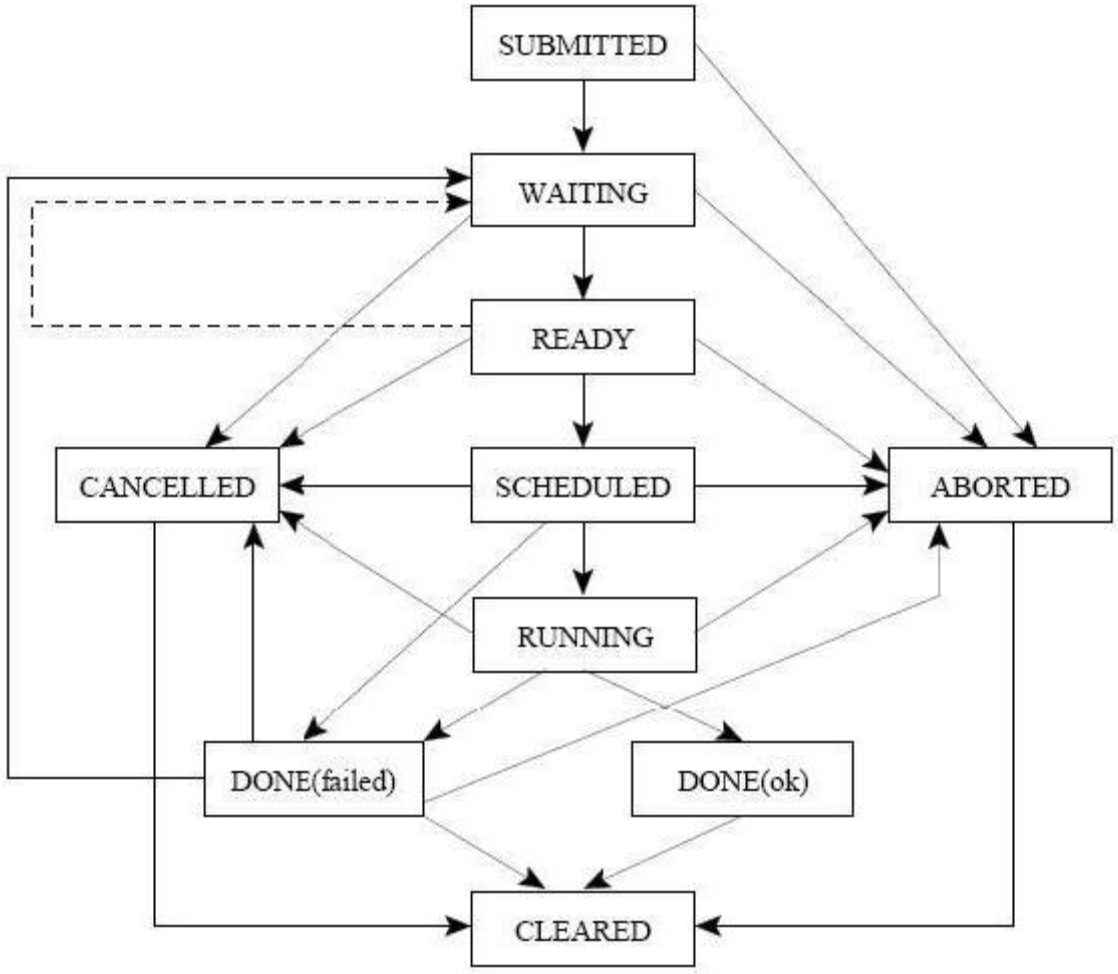

Figure 2. The lifecycle of a job

4.1.3. The LB Database

The L&B is implemented in gLite as a relational (SQL) database. The most important tables are *events*, *short_fields* and *long_fields*. The format used by the GO is NO LONGER (since January 2011) this database (see 4.3 for details); nonetheless, the database organization will be described, as it conveys the semantics of the job management. Appendix B provides some interpretation of codes and attributes.

4.1.3.1. Table *events*

This table has only one row per event.

- Jobid: the job unique identifier
- Event: the number of the event in a sequential ordering which is close to the physical time, but not necessarily identical. The event number is mainly an index, which allows retrieving information in the two other tables

- Code: internal code (see Annex 1)
- Prog: describes which service has generated this event
- Timestamp and userId are self-explanatory

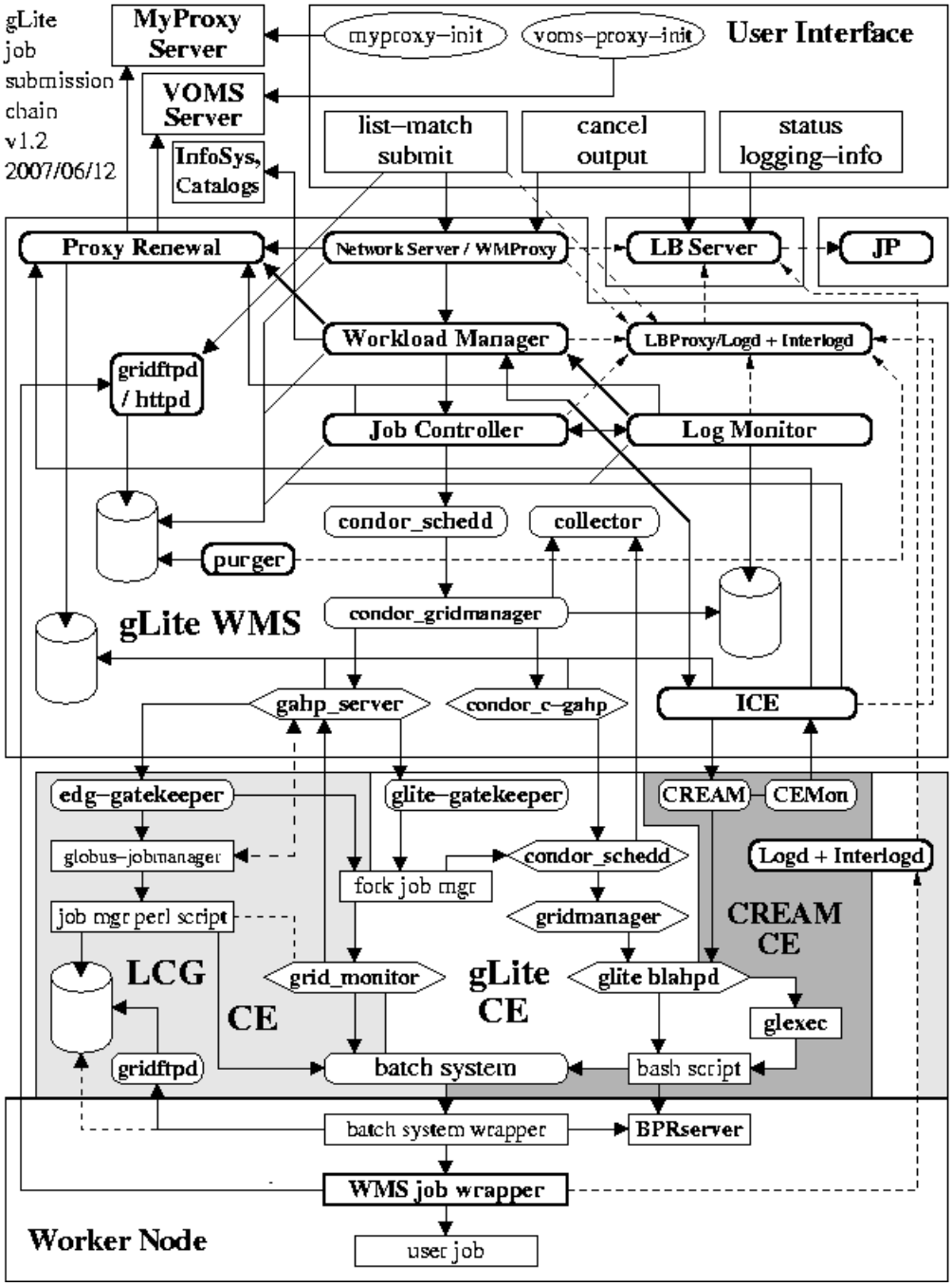

Figure 3. Job submission chain, by Maarten Litmaath

4.1.3.2. Table *short_fields*

This table logs the internals sub-components of an event. The *JobId* and *event* fields allow correlating the rows with those of the previous table. The next two fields are namely the gLite event attribute name and value. A close approximation of the attributes is given in Annex 1, which is the specification of the attributes names in the L&B API. The outcome of the job (failure or success) is very often clearly stated as the value of attribute REASON (e.g., for successful events, the value is "Job terminated successfully").

4.1.3.3. 3. Table *long_fields*

This table is the more complicated, mainly because information is represented as a blob. The general idea is as follows.

- The first row is the user description of the job, enriched with defaults values, expressed in JDL.
- The next rows translate and enrich this description in the dialect of the next services the job has to pass through.

4.1.4. Database purging

To prevent the database from ever increasing, an automatic purge system has been made part of the LB. Each job and its associated events should eventually be removed, but the grace period will depend on its state. For instance, at the LAL site, jobs that terminated successfully will be removed in about 48 hrs[1], aborted or cancelled jobs will be kept longer - 14 days - and jobs in any other state will be kept 60 days, according to the current configuration. The details can be found in the LB's Administrator's Guide [LB.3], section 3.2.5.

As the LB is normally regularly purged, the full SQL dump cannot be used *as is* to provide an accurate history of every job's lifecycle.

However, the purged jobs can be dumped into a text file for later use. This history of purged jobs can be considered reliable as it was build to restore the LB in a previous state.

Figure 4 illustrates the temporal repartition of events at a given time. The Y axis represents the number of events for any given day, while the X axis is the time period for which the dump contains events (in this example, august 14, 2010 through January 7, 2011). This figure shows that the events of the last 48h are all present, while only part of the data of the last 14 days is kept and nearly no information about older jobs is present.

[1] The actual purge policy is set to 48 hrs, but the purge daemon is run once every 24h, so any job may be kept a little more than 48hrs.

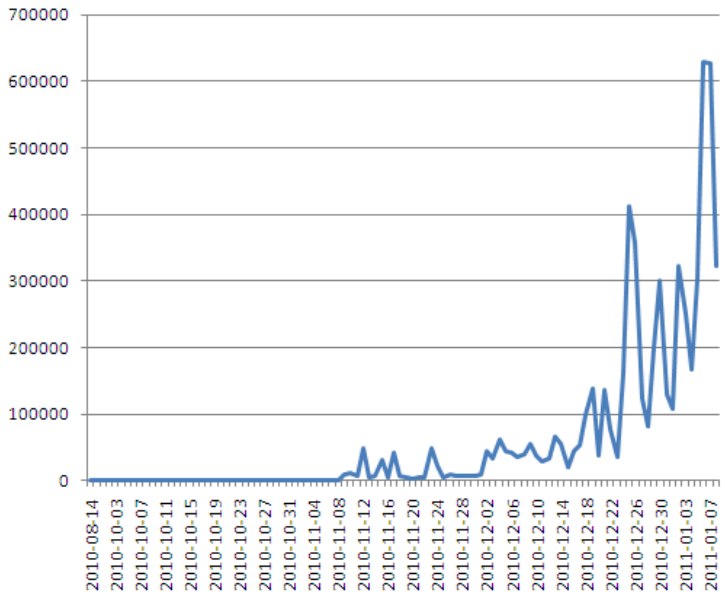

Figure 4. LB events history

4.2. SCOPE

Each LB service is associated to a particular broker; thus, each LB log covers only those jobs that were managed by this particular broker. Nevertheless, the scope of the broker is the full grid resources.

4.3. FORMAT

4.3.1. Evolutions

The LB trace we provide has evolved with modifications in the purge systems:

- For logs acquired before September 2009, we provide an SQL dump of the database (ASCII, resulting of `mysqldump`), which can easily be reloaded as a mySQL database. It is a complete copy of the LB server.
- The dumps between October 2009 and January 2011 have been impacted by a modification of the purge system and are incomplete[2].
- Since January 2011, daily trace files are acquired through the `glite-lb-dump` utility and provided as such. The file format is described below.

4.3.2. Daily trace files

In the daily trace file provided since January 2011, each record (text line) in the LB dump is an LB event; hence it maps 1:1 on the rows in the *events* table described in section 4.1.3.1. Columns of the *events* table appear in the dump records in a straightforward way. The data from *short_fields* and *long_fields* are encoded again in the dump in the format DG.<eventtype>.<name>="<value>". "name" and "value" come directly from the two tables.

[2] These faulty dumps are not directly downloadable from the website but can be made available on a per-request basis. Feel free to contact us in case you need them.

Meaning of the event fields is described in the comments with C structures definitions in /opt/glite/include/glite/lb/events.h when the glite-lb-common RPM is installed.

This format can therefore easily be translated into the L&B internal format if needed. We plan to propose a convert script that can rebuild the database from the traces in the future.

4.3.3. WARNING

The jobs listed are extracted by with `glite-lb-dump` (no relation with the previous `mysqldump` command), and are those that will later be selected by the purge system (see section 4.1.4). Thus, it is important to note that:

- The events date is correct, but this date might be days or even weeks in the past, in the case of erroneous jobs.
- The file at date *x* will show the successful jobs, but **NOT the failed jobs**, of the previous day. As a rule of thumb, a two weeks delay gives the complete picture: the jobs completed at date x are found in the files dated *x*+1… *x*+15.

4.3.4. Usage of the daily trace files

These files can be used in different ways:

- If you are running an LB database, the text files can be directly used with LB tools such as glite-lb-statistics or glite-jp-importer. Importing purge files in a mySQL database can be done with the glite-lb-load utility.
- Installing an LB server to load these dumps directly is an option, but requires good technical skills. Installation and configuration of such a server is detailed in the LB Administrator's Guide at the following URL : http://egee.cesnet.cz/cvsweb/LB/LBAG.pdf
- The simplest and more frequent solution is to parse directly the files.

4.4. MORE INFORMATION

Section 6 (workload management) of the gLite user guide [G.1] introduces job types and management from the user point of view.

The LB user guide [LB1] mainly describes the query API, but information on the semantics of the fields can be inferred.

The technical report [LB.2] gives an in-depth presentation of the LB.

5. THE BATCH SYSTEM TRACES

5.1. DESCRIPTION

The batch system traces report on the job events related to the batch system of a site. Scheduling events and memory consumption are recorded. The batch system is not part of the gLite middleware, and indeed various batch systems are used in the EGEE grid (PBS/MAUI, LFS, etc.). Thus, these traces do not provide any gLite job identifier. In many cases, the site's resources are shared between gLite-EGEE, non gLite-EGEE, and non EGEE jobs.

5.2. SCOPE

Each trace is local to a batch system.

5.3. FORMAT

The traces are provided as daily ascii files. Each tar archive holds one week of traces.

For now, only PBS logs are available. See Appendix C for the format.

5.4. MORE INFORMATION

The PBS log format is fully documented in [BS.1].

A comprehensive presentation, together with scripts for converting these logs into Standard Workload Format, is available, as a part of the parallel workload archive project [BS.2].

6. INTERNAL LOGS

6.1. DESCRIPTION

This section covers various services that take place in the job submission chain, as described in fig. 3. These traces log the internal details of the services activity. Their main usage is diagnosis.

Service	*Description*
Wmproxy_events	Simple job submission log: date, user, job identifier.
Jobcontroller_events	The jobcontroller handles the reliable forwarding of job submission and control (cancellation) requests to Condor-G.
CondorG	Condor-G logging : -job submission to the site -executing host -job's status change notification
Logmonitor_events	Services interactions within the WMS. The 'logmonitor' relays changes in the state of the job, as obtained from the Condor-G user log, to the rest of the WMS system.
Workload_manager_events	Summary of the matchmaking (CE selection), including the number of matching resources, the service time, and the outcome (selected CE).
Jobmap	Gatekeeper information, provides the translation between gLite information and local information.

6.2. SCOPE

Same as the LB.

6.3. FORMAT

The traces are provided as daily ascii files. Each tar archive holds one week of traces.
The log format is not documented in the gLite suite.

7. THE REAL TIME MONITOR DATA

7.1. OVERVIEW OF THE REAL TIME MONITOR

The RTM is being developed by Imperial College London, in the framework of the UK Particle Physics grid (GridPP) and the EGEE project. As the name says, the data collection scheme allows monitoring the Grid working in near real time. The RTM is able to show various information such as running and scheduled jobs, job transfers and detailed information on Resource Brokers and Computing Elements for each site, and to interact with the display. The RTM project also publishes processed information, such as graphs, and pdf summaries. For more information about the RTM project, visit http://gridportal.hep.ph.ic.ac.uk/rtm/

Besides real-time monitoring, the RTM data are a major source of information on the EGEE behavior. The RTM summarizes information that is available only from the Logging and Bookkeeping system. This information, together with the information system (BDII) archives published by the GO, should allow exploring many of the questions related to grid usages, as long as only jobs are considered.

7.2. DESCRIPTION

The trace registers 37 attributes for each job, which can be distinguished as Information (table 7.1), Timestamps (table 7.2), and Metrics (table 7.3). The metrics are derived from the timestamps.

7.2.1. Information

N	Name	Description
1	Jobid	The glite Job identifier
2	Type	A type *computed from the RTM data*
3	FINAL REASON	Job termination status
4	FINAL EXIT CODE	Job exit code
5	RB	Name of the Resource Broker
6	UI	Name of the User Interface
7	CE	Name of the Computing Element
8	WN	Name of the Worker Node
9	VO	Name of the Virtual Organization
10	DN	Name of the Identity
11	Requirements	The job requirements
12	Rank	The ranking formula
13	RegistrationTimeString	Registration date
14	TimeWrittenString	

Table 7.1: information attributes

Most of this information comes from the LB system. However, the type is not the value code as provided by the LB. A job is noted REGISTERED (resp RAN, DONE, CLEAR) if an event associated

with the end of the registration (resp scheduling-mapping, execution, finalization) process has been seen by the RTM.
The job requirements are the constraints on the job mapping. The ranking formula is the prescribed method to rank candidates CEs. Both can be specified by the user, or resort to default values. The gLite User Guide [G.1] gives the detailed syntax.

7.2.2. Timestamps

N	Name	Description
15	Userinterface_regjob_Epoch	Time at which the job was registered in epoch format
16	networkserver_accepted_Epoch	Time at which the job was accepted by the network server
17	workloadmanager_match_Epoch	Time at which the matchmaker did match the job to a resource
18	Jobcontroller_transfer_Epoch	Time at which the job controller transferred the job to the resource
19	logmonitor_accepted_Epoch	Time at which the job was accepted by the resource as seen from the logmonitor
20	logmonitor_running_Epoch	Same but for the running state
21	logmonitor_done_Epoch	Same but for the done state
22	lrms_running_Epoch	Time at which the lrms found the job to be running
23	lrms_done_Epoch	Time at which the lrms found the job to be done

Table 7.2: timestamps attributes

In general, the LRMS (local resource management system) values are more accurate than the logmonitor ones. However, not all CEs provide this service.

Both the information attributes and the timestamps attributes make sense only if the corresponding step in the job lifecycle has been reached. However, the records are fixed-length (37 attributes). The default values are used in place of the meaningless values.

7.2.3. Metrics

N	Name	Description	Formula
24	Registration Time	between a user submitting a job and the job being accepted by the RB (Submitted State time)	networkserver_accepted_Epoch - userinterface_regjob_Epoch;
25	Match Time	between a job being accepted and finding a CE to put it on (Waiting State time)	workloadmanager_match_Epoch - networkserver_accepted_Epoch;
26	Upto Scheduled Time	between a job being accepted and transferred to the CE (Waiting + Ready States times)	jobcontroller_transfer_Epoch - networkserver_accepted_Epoch;
27	Upto Scheduled Acceptance Time	same, but as reported by the logmonitor.	logmonitor_accepted_Epoch - networkserver_accepted_Epoch;
28	Computing Element Total Time	includes both the Scheduled and Running States times	logmonitor_done_Epoch - logmonitor_accepted_Epoch;
29	Computing	includes only the Scheduled Time	logmonitor_running_Epoch -

	Element Scheduled Time		logmonitor_accepted_Epoch;
30	Worker Node Time (LogMonitor)	exists both from the LogMonitor and the LRMS	logmonitor_done_Epoch - logmonitor_running_Epoch;
31	Worker Node Time (LRMS)		lrms_done_Epoch - lrms_running_Epoch
32	Total Time	between acceptance and completion on a site; does not include User Interface interaction (registration and finalization)	total_Time = logmonitor_done_Epoch - networkserver_accepted_Epoch
33	Efficiency	ratio between the actual running time and the total time in the system	efficiency = (float) logmonitor_wn_Time / (float) total_Time;
34	requirementsCount	Number of conjunctive clauses in the requirement expression	
35	rankCount		
36	resubmitcount	Number of resubmissions	
37	logmonitor_resubmission_wasted_wn_Time	Cumulative running time before failure	

Table 7 3: metrics attributes

7.2.4. Default values

jobID	"unknown"
type	"UNDEFINED"
final_reason	"-"
final_exit_code	"-"
TimeWritten	"unknown"
RegistrationTimeString	"unknown"
RB	"unknown"
VO	"unknown"
DN	"unknown"
Requirements	"unknown"
Rank	"unknown"
CE	"unknown"
UI	"unknown"
WN	"unknown"
userinterface_regjob_Epoch	0
networkserver_accepted_Epoch	0
workloadmanager_match_Epoch	0
jobcontroller_transfer_Epoch	0
logmonitor_accepted_Epoch	0
logmonitor_running_Epoch	0
logmonitor_done_Epoch	0
lrms_running_Epoch	0
lrms_done_Epoch	0
registration_Time	0

match_Time	0
upto_scheduled_transfer_Time	0
upto_scheduled_acceptance_Time	0
logmonitor_ce_total_Time	0
logmonitor_ce_scheduled_Time	0
logmonitor_wn_Time	0
lrms_wn_Time	0
total_Time	0
resubmitcount	0
logmonitor_resubmission_wasted_wn_Time	0
efficiency	-1
requirementsCount	0
rankCount	0

7.3. SCOPE

The trace covers all gLite jobs.

7.4. FORMAT

Each job record is line; the attributes inside a record are listed following the numbering of tables 7.1, 7.2, and 7.3, and are separated by tabulations.

The trace is provided as a weekly archive. Each file in the archive corresponds to the activity of one CE and one day.

APPENDIX A AVAILABLE INFORMATION SYSTEM ATTRIBUTES

GlueCEHostingCluster	GlueSEArchitecture
GlueCEImplementationName	GlueSEName
GlueCEInfoLRMSType	GlueSEUsedNearlineSize
GlueCEInfoDataDir	GlueSEType
GlueCEStateEstimatedResponseTime	GlueSEUniqueID
GlueCEStateTotalJobs	GlueSEStatus
GlueCEStateFreeJobSlots	GlueSETotalNearlineSize
GlueCEPolicyMaxObtainableCPUTime	GlueSEImplementationVersion
GlueCEPolicyMaxTotalJobs	GlueSETotalOnlineSize
GlueCEPolicyPriority	GlueSEImplementationName
GlueCEPolicyPreemption	
GlueCEName	GlueServiceWSDL
GlueCEInfoGatekeeperPort	GlueServiceSemantics
GlueCEInfoJobManager	GlueServiceUniqueID
GlueCEInfoDefaultSE	GlueServiceOwner
GlueCEStateRunningJobs	GlueSiteDescription
GlueCEStateWaitingJobs	GlueServiceStatusInfo
GlueCEStateFreeCPUs	GlueServiceVersion
GlueCEPolicyMaxRunningJobs	GlueServiceURL
GlueCEPolicyMaxWallClockTime	GlueServiceAccessControlRule
GlueCEPolicyAssignedJobSlots	GlueServiceName
GlueCEInfoHostName	GlueServiceType
GlueCEInfoApplicationDir	GlueServiceStatus
GlueCEInfoTotalCPUs	GlueServiceStartTime
GlueCEStateStatus	
GlueCEStateWorstResponseTime	GlueSiteOtherInfo
GlueCEPolicyMaxCPUTime	GlueSiteUniqueID
GlueCEPolicyMaxWaitingJobs	
GlueCEPolicyMaxObtainableWallClockTime	GlueClusterService
GlueCEPolicyMaxSlotsPerJob	GlueInformationServiceURL
GlueCEAccessControlBaseRule	
GlueCEUniqueID	

APPENDIX B: LB EVENTS CODES AND ATTRIBUTES

ABORT	12
ACCEPTED	2
CANCEL	11
CHKPT	18
CLEAR	13
CURDESCR	20
DEQUEUED	5
DONE	10
ENQUEUED	4
HELPERCALL	6
HELPERRETURN	7

LISTENER	19
MATCH	15
PENDING	16
PURGE	14
REFUSED	3
REGJOB	17
RESUBMISSION	9
RUNNING	8
TRANSFER	1
TYPE	?
UNDEF	0
USERTAG	21

ARRIVED	[common] Time the event was stored into the bookkeeping server database.
CLASSAD	[Chkpt] Application specific checkpoint value.
DEST_HOST	[Notification] Hostname the notification is sent to.
	[Transfer] Hostname of server that takes over control of the job.
DEST_ID	[Match] Identification of the queue on the CE that the job could be sent to.
DEST_INSTANCE	[Transfer] Service (instance) that takes over control of the job.
DEST_JOBID	[Transfer] Job id as assigned by the receiving software component.
DEST_PORT	[Notification] Port number the notification is sent to.
DESTINATION	[Transfer] Destination where the job is being transfered to.
EXIT_CODE	[Done] Exit code of the job's process.
FROM	[Accepted] The software component the job was received from.
	[Refused] The software component that tried to send the job.
FROM_HOST	[Accepted] Hostname of the component the job was received from.
	[Refused] Hostname of the component that tried to send the job.
FROM_INSTANCE	[Accepted] Instance of the component the job was received from.
	[Refused] Instance of the component that tried to send the job.
HOST	[common] Hostname of the machine where the event was generated.
JDL	[RegJob] Job description of the job being registered.
JOB	[EnQueued] Job description in the receiver's language.
	[Transfer] Job description in receiver's language.
JOBID	[common] Grid job id of the job the event belongs to.
JOBSTAT	[Notification] Status of the job (the notification content).
JOBTYPE	[RegJob] Type of the registered job (SIMPLE, DAG, PARTITIONABLE, PARTITIONED)
LOCAL_JOBID	[Accepted] New job id as assigned by the receiving component.
	[DeQueued] New job id as assigned by the retreiving component.
NAME	[UserTag] Arbitrary user tag name.
NODE	[Running] Worker node on which the job executable is being run.
NS	[RegJob] NetworkServer handling the newly registered job.

NSUBJOBS	[RegJob] Number of subjobs this job plans to spawn.
OWNER	[Notification] Identification of the job owner (certificate subject).
PARENT	[RegJob] Grid job id of the parent job registering this new one.
QUEUE	[DeQueued] Name of the queue the job was obtained from.
	[EnQueued] Queue into which the job has been stored for retrieval by another component.
REASON	[Abort] Reason why the job was aborted by the system.
	[Cancel] Details on the attempt to cancel the job, especially the reason of failure.
	[Clear] Reason why the output sandbox removed (USER, TIMEOUT or NOOUTPUT).
	[Done] Detailed description why the job was terminated.
	[EnQueued] Details on the attempt to enqueue the job, especially the reason of failure.
	[Pending] Description why the matching CE for the job was not found (yet).
	[Refused] Description of the reason why the job was refused.
	[Resubmission] Reason why the job will or will not be resubmitted.
	[Transfer] Detailed description of the transfer, especially reason of failure.
RESULT	[EnQueued] Result code of the attempt to queue the job (START, OK, REFUSED, FAIL).
	[Resubmission] Result code of the resubmission decision (WILLRESUB or WONTRESUB or SHALLOW).
	[Transfer] Result code of the transfer attempt (START, OK, REFUSED or FAIL).
SEQCODE	[common] Sequence code assigned to the event.
SOURCE	[common] Source (software component) which generated this event.
SRC_INSTANCE	[common] Instance of source component (e.g. service communication endpoint).
STATUS_CODE	[Cancel] Classification of the attempt to cancel the job (REQ, REFUSE, DONE, ABORT).
	[Done] Reason code for the termination of the job (OK, FAILED or CANCELLED).
TAG	[Chkpt] Application specific checkpoint tag.
	[Resubmission] Value of the attribute on which the decision to resubmit the job was based.
TIMESTAMP	[common] Time the event was generated.
UNIT	[ResourceUsage] units (sec, kB, etc.)
USER	[common] Identity (certificate subject) of the event sender.
VALUE	[UserTag] Arbitrary user tag value.
WN_SEQ	[ReallyRunning] sequence code on the worker node

APPENDIX C: PBS LOGS FORMAT

Each entry is terminated by a new line. The format of an entry is:

```
date time;record_type;id_string;message_text
```

- The `date time` field is a timestamp
- The `id_string` is the job identifier.
- The `message_text` format is blank separated keyword=value fields.
- The `record_type` is a single character indicating the type of record, which is associated with scheduling events. Table C1 describes the most frequent ones.

The content of `message_text` depends on the record type. Table C2 describes the most frequent attributes. Note that:

- all information is purely local, so names and identifier are not the LB or RTM ones;
- the presence of attributes in `message_text` depends on the event *e.g.* `end` or `resources_used` make sense only after the execution terminates, thus in an `E` type event;
- the value associated to resources_used.<resource> are accurate only up to the sampling frequency of the PBS logger. Thus, they might be grossly underestimated for short jobs.

Type	Description
Q	Job entered a queue
S	Job execution started
E	Job ended (terminated execution)
A	Job was aborted by the server
D	Job was deleted by request

Table C1: PBS logs attributes. All times are in seconds

Attribute	*Description*
`user`=username	The local user name under which the job will execute
`group`=groupname	The local group name under which the job will execute.
`jobname`=job_name	The name of the job.
`queue`=queue_name	The name of the queue in which the job resides.
`ctime`=time	Time when job was created (first submitted).
`qtime`=time	Time when job was queued
`etime`=time	Time when job became eligible to run;
`start`=time	Time when job execution started.
`end`=time	Time when job ended execution.
`exec_host`=host	Name of host on which the job is being executed.
`Resource_List.<resource>`=limit	List of the specified resource limits.
`Exit_status`=value	The exit status of the job.
`Resources_used.<resource>`=value	Aggregate specified resource usage of the job.

Table C2: PBS logs attributes. All times are in seconds