

## L&B Status and Plans

*Aleš Křenek*  
 CESNET

[www.eu-egee.org](http://www.eu-egee.org)



#7625	Support	ongoing	
#7626	Internal code and functionality cleanup	ongoing	transactional code finished
#7627	Revised documentation	done	neverending process
#7628	Megajob challenge #2	ongoing	testbed setup, original tests redone
#7629	release 2.0	ongoing	code ready, thoroughly tested, tags missing
#7630	Massive use of L&B notification in job monitoring tools (prototype)	done	presented at EGEE'08
#7631	Massive use of L&B notification in job monitoring tools (consolidation)	ongoing	use in Dashboard and RTM

#7632	Non-intrusive purger	done	
#7633	Adoption of common logging format	pending	
#7634	Advanced authorization	ongoing	done on input to L&B server
#7637	Consolidate and standardize network interfaces	ongoing	HTML and plain text interfaces; standards reviewed
#7638	Support native CREAM jobs	pending	Java producer API available
#7640	Follow changes in WMS and address emerging requirements	pending	no changes to follow so far

- patch #1727 gLite 3.1 LB for slc4/i386 platform
- patch #1803 First cumulative update of LB 3.1
  - grey area in release process revealed
  - #1803 certification delayed due to so-far-hidden bug in UI
  - #1841 (WMS update) provided in parallel
    - ▶ **Which L&B version to use for #1841 certification?**
- no other critical issues
- details at SA3 session

- changes in modules / RPMs
  - `org.glite.jobid` subsystem
    - ▶ job identifier manipulation
  - `org.glite.lbjp-common` subsystem
    - ▶ common libraries with JP
  - split `org.glite.security.gss` from `gsoap-plugin`
    - ▶ GSS and auxiliary Globus functions wrappers
  - added `lb.types`
    - ▶ common events and status definitions, previously in `org.glite.lb`
  - dropped `lb.client-interface`
    - ▶ interfaces in `lb.client` and `lb.common`
  - standalone `lb.state-machine`
    - ▶ state machine implementation in library
    - ▶ reused in JP and `lb.utils`
  - dropped `lb.proxy`
    - ▶ functionality merged into server

- merged L&B server and proxy
  - 95% of code was shared anyway
  - merged into single binary with 3 run-time modes:
    - ▶ server only – network sockets, SSL AuthN/Z
    - ▶ proxy only – UNIX sockets, local AuthZ
    - ▶ hybrid – both
  
- hybrid mode
  - single instance and database for WMS and users' queries
  - events processed (state computed) only once
  - less total disk and CPU load
  - no need for extra L&B machine in smaller installations
  - to be reflected in deployment

- internals cleanup
  - fully transactional code
  - fine grain DB locks to replace semaphores
- zombie jobs
  - keep minimal job record after purging
  - outside normal job tables – no slowdown
  - less than 100B per job, ie. 3.5 GB/year @ 100 kjobs/day
- extensions of notifications
  - relaxed “single job” requirement
  - conditions like “all users jobs”, “all jobs at a given WMS”
  - on-change condition
    - ▶ send notification only when specific attribute changes, to whatever value

- protocol compatibility
  - fully compatible with 3.1 clients and logger
  - thoroughly tested
- API compatibility
  - `glite/jobid` instead of `glite/wmsutils/jobid`
  - added parameters to notification API calls
- database schema
  - incompatible
    - ▶ support for server, proxy, and hybrid mode
    - ▶ raw events stored unparsed
  - migration scripts provided



- Etics client overhead
  - still too much for developer's daily work
  - fragile environment is easily corrupted, start from scratch required
  - yields waste of developers' time
- almost impossible to create standard source distribution
  - download single tarball
  - install a few dependencies
  - `./configure && make && make install`
- porting still difficult
  - SL5 x86\_64 was a nightmare
    - ▶ all problems coming from build environment and dependencies
    - ▶ virtually no changes in L&B code
  - `glite_3_1_0_dev` branch quickly evolving
  - `etics-checkout && etics-build` never worked for L&B on this platform

- caused by wrong code structuring and approach to Etics
- alternate approach tested
  - single configure script for whole L&B
  - `--enable-client`, `-server`, `-logger` to select components
  - `--with-*` to specify dependencies
  - generates top-level Makefile
- developers build or download dependencies once
- source distribution straightforward
- etics configurations call this configure consistently
- dependency information duplicated

- **M**essaging **S**ystem for **G**rids
  - unified framework for grid monitoring (GridView, Dashboard)
- L&B to MSG publisher
  - subscribe with L&B to receive notifications
  - maintain active notification handle
  - all jobs on a WMS, important state changes only
  - feed record of few job attributes into MSG
  - accurate job state available to MSG clients
- replace inefficient frequent L&B queries
- presented at EGEE'08
- further stress tests continue

- **Real Time Monitor**
  - monitoring and visualization of grid traffic
- current implementation of L&B collector
  - dedicated thread for each L&B – scalability limits
  - direct queries to L&B databases – will break with L&B 2.0
  - inaccurate job state computation
- replacement implementation (work in progress)
  - based on L&B notifications
  - handle more L&B's in single thread
  - insert arriving information into RTM internal database

- release LB 2.0, integrate with WMS
- rerun and extend “megajob” tests, address emerging bottlenecks
- discuss relevant standards, plan related changes in L&B
  - OGSA-BES, WS-DAI(R/X), WS-Notification
- MSG and RTM
  - run stress-tests, towards production deployment
- Authorization
  - implement fine-grained privileged access (required by MSG and RTM)