CHEP '06

Computing in High Energy Physics

13 – 17 February 2006, Tata Institute of Fundamental Research, Mumbai

BLAHP: A local batch system abstraction layer for global use

Primary Authors:

REBATTO, Davide[1] (david.rebatto@mi.infn.it) - PRELZ, Francesco[1] - FIORENTINO, Giuseppe[1] - MEZZADRI, Massimo[1] - MARTELLI, Enzo[1] - MOLINARI, Elisabetta[1]

Co-Authors:

DVORAK, F.[2] - KOURIL, D.[2] - KRENEK, A.[2] - MATYSKA, L.[2] - MULAC, M.[2] - POSPISIL, J.[2] - RUDA, M.[2] - SALVET, Z.[2] - SITERA, J.[2] - SKRABAL, J.[2] - VOCU, M.[2] - AVELLINO, G.[3] - BECO, S.[3] - CAVALLINI, A.[3] - MARASCHINI, A.[3] - PACINI, F.[3] PARRINI, A.[3] - SCARCELLA, C.[3] - SOTTILARO, M.[3] - TERRACINA, A.[3] - MONFORTE, S.[4] - PAPPALARDO, M.[4] - ANDREOZZI, S.[5] - CHECCHI, M.[5] - CIASCHINI, V.[5] - FERRARI, T.[5] - GIACOMINI, F.[5] - LOPS, R.[5] - RONCHIERI, E.[5] - VENTURI, V.[5] GUARISE, A.[7] - PATANIA, G.[7] - PIRO, R.[7] - WERBROUCK, A.[7] - ANDREETTO, P.[6] - BORGIA, A.[6] - DORIGO, A.[6] - GIANELLE, A.[6] - MARZOLLA, M.[6] - MORDACCHINI, M.[6] - SGARAVATTO, M.[6] - ZANGRANDO, L.[6]

[1]INFN Milano – [2]CESNET – [3]Datamat – [4]INFN Catania – [5]INFN CNAF – [6]INFN Padova – [7]INFN Torino

Overview

In current, widely deployed management schemes, intensive computing farms are locally managed by **batch systems** (e.g. *Platform LSF*, *PBS/Torque*, *BQS*, etc.). When approached from the outside, at the global – or *grid* – level, these local resource managers (LRMS) are seen as services providing at least a basic set of job operations, namely **submission**, **status retrieval**, **cancellation** and **security credential renewal**. The Batch-system Local ASCII Helper Protocol (**BLAHP**) was designed to offer a simple abstraction layer over the different *LRMS*, providing uniform access to the underlying computing resources. In order to preserve the simplicity and portability of the scheme and the robustness of the implementation, the functionality in the abstraction had to be carefully limited. The daemon, originally developed for the **EGEE gLite Condor[1]-based Computing Element**, is going to be used by Condor also outside the gLite framework. It is also a component of **CREAM[2]**, the Web Services oriented Computing Element for gLite.

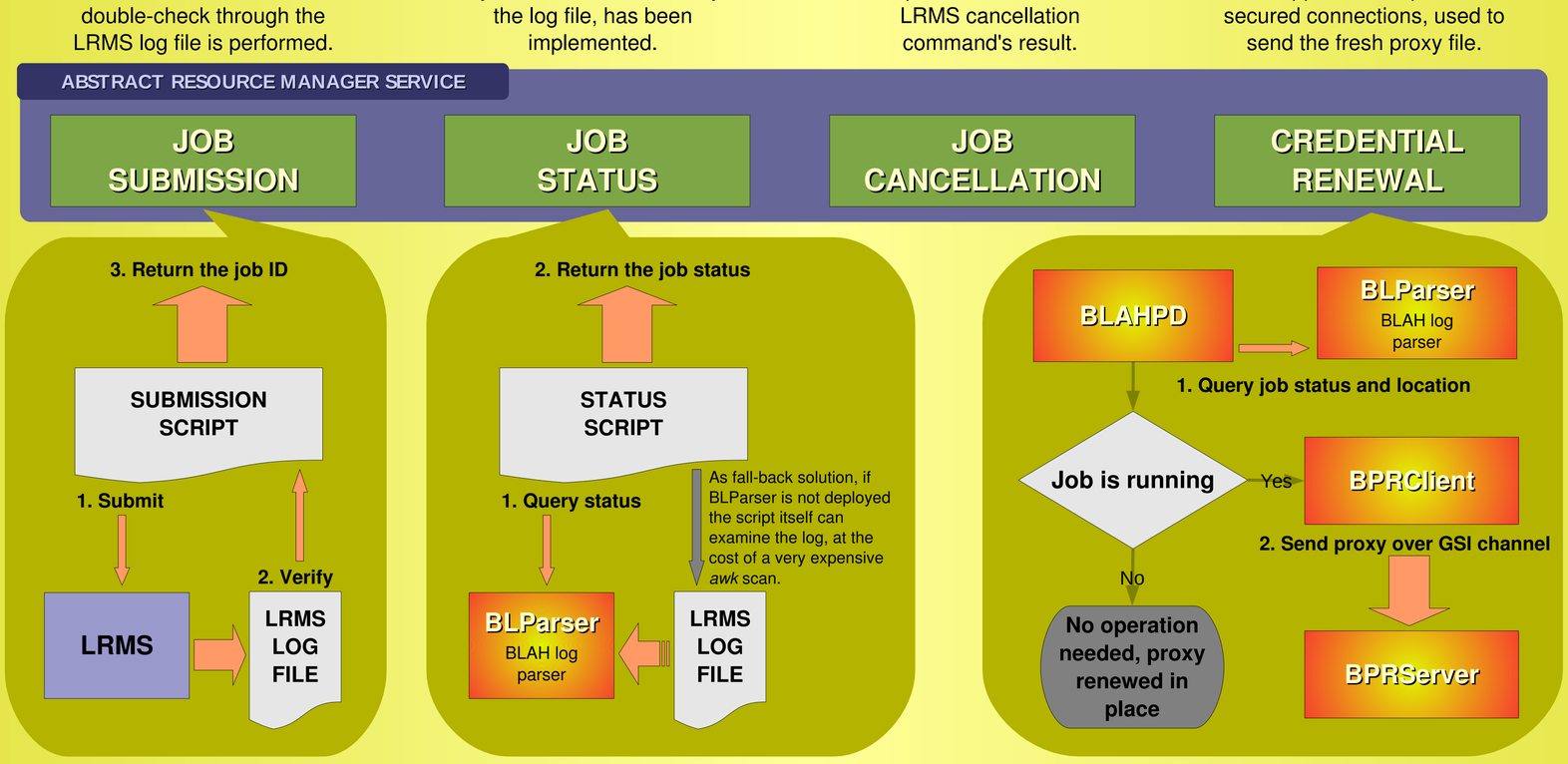
Batch system abstraction

Submission must be considered successful only if the job is actually accepted and enqueued by the LRMS. Therefore a double-check through the LRMS log file is performed.

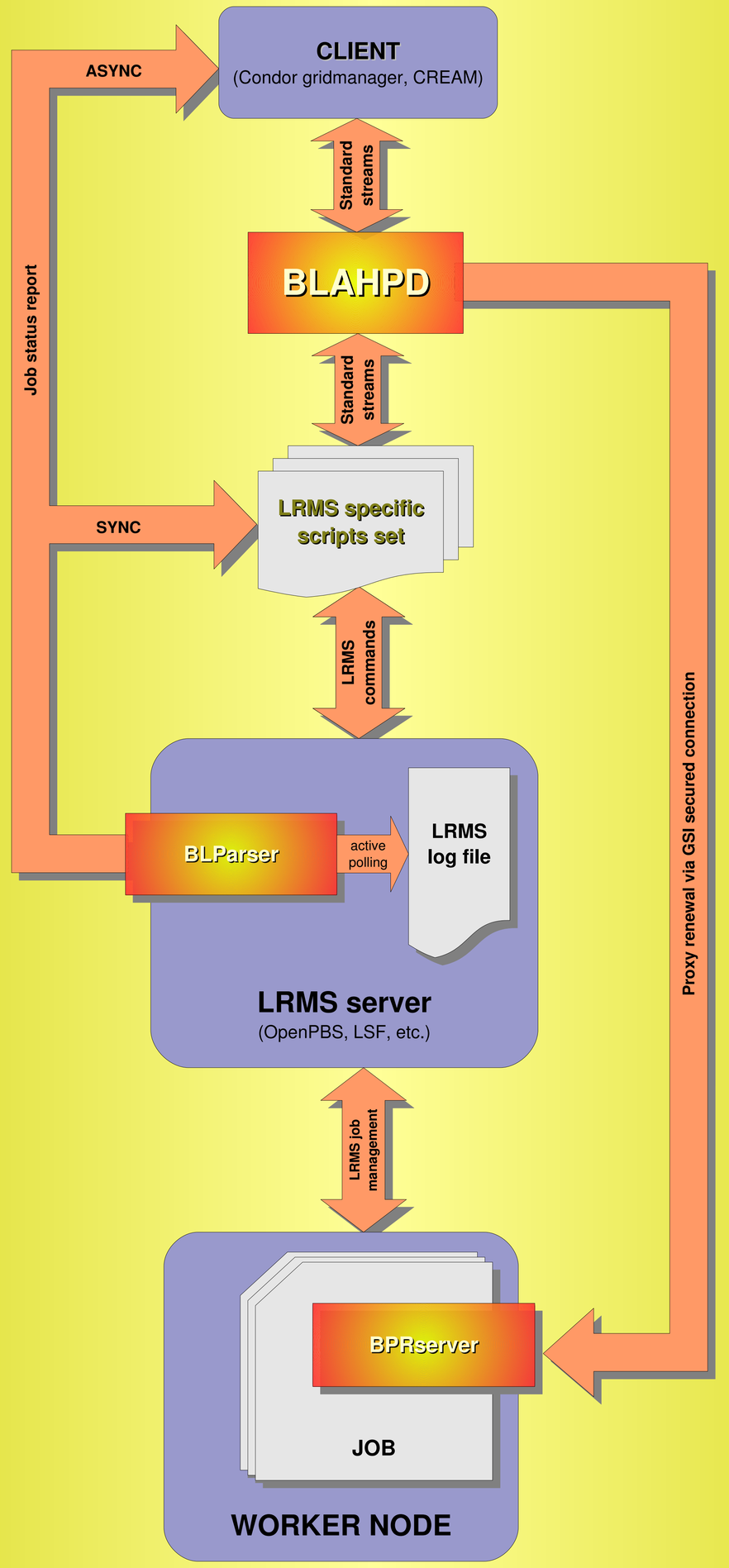
The LRMS server can be easily overloaded if flooded by status requests. To avoid direct queries to the server, a job state machine, fed by the log file, has been implemented.

No assumption are made on job cancellation. BLAHPD relies on LRMS capability for this operation, and trusts the LRMS cancellation command's result.

LRMSes offer no way of dispatching files to the workdir of a running job, so a small daemon is sent along with the job and started by the job wrapper. It accept GSI secured connections, used to send the fresh proxy file.



BLAHP Architecture



BLAHP commands[3]

COMMANDS
Provide the list of supported commands

VERSION
Return BLAHPD version

ASYNC_MODE_ON
Enable async notification of unread results

ASYNC_MODE_OFF
Disable async notification of unread results

RESULTS
Read and empty the result buffer

BLAH_JOB_SUBMIT
Submit a job

BLAH_JOB_STATUS
Retrieve jobs' status

BLAH_JOB_CANCEL
Cancel a job

BLAH_JOB_HOLD
Suspend a job (if supported by LRMS)

BLAH_JOB_RESUME
Resume a suspended job

BLAH_JOB_REFRESH_PROXY
Refresh job credentials

BLAH_SET_GLEXEC_DN
Use user's credential to issue LRMS commands

BLAH_SET_GLEXEC_OFF
Stop using user's credential

BLAHP implementation details

- The **blahpd daemon** runs on the submission machine of the batch system. It translates BLAHP commands into abstract LRMS actions.
  Most job management commands are served in dedicated threads: upon completion the results are queued in a (thread-safe) buffer and can be retrieved by the client later on.
- The LRMS specific **scripts set** is where the abstraction effort is concentrated. The scripts for different LRMS must have the same semantic and syntax, offering a uniform interface toward the core daemon. This architecture allows to make BLAHPD support new batch systems without going through the daemon code.
- The **BLParser** (BLAH Log Parser) implements a state machine, updated by a constant watching of the LRMS log file. It accepts synchronous queries via a socket connection, and can asynchronously notify job status changes to subscribed clients (e.g. CREAM).
- The **BPRServer** (BLAH Proxy Renewal Server) is a small daemon, sent along with the job to the worker node. The server is started by the job wrapper when the job start running, and when an incoming connection is detected, it answers with the jobID. The connecting client, if the jobID is correct, open a GSI secured channel and sends the fresh proxy, which replaces the old one.

[1] Condor Project: http://www.cs.wisc.edu/condor/

[2] Cream web page: http://grid.pd.infn.it/cream/field.php

[3] For a detailed description of BLAH protocol and syntax, see http://egee-jra1-wm.mi.infn.it/egee-jra1-wm/ce_blahp.shtml