Journal of Physics: Conference Series **119** (2008) 052013 doi:10.1088/1742-6596/119/5/052013

The ATLAS Software Installation System for LCG/EGEE

Alessandro De Salvo[(1)], Alessandro Barchiesi[(2)], Kondo Gnanvo[(3)], Carl Gwilliam[(4)], John Kennedy[(5)], Gernot Krobath[(5)], Andrzej Olszewski[(6)], Grigory Rybkine[(7)]

[(1)] Istituto Nazionale di Fisica Nucleare, sez. Roma 1,
[(2)] Università di Roma I "La Sapienza", [(3)] Queen Mary and Westfield College,
[(4)] University of Liverpool, [(5)] Ludwig-Maximilians-Universität München,
[(6)] Henryk Niewodniczanski Institute of Nuclear Physics, Polish Academy of Sciences,
[(7)] Royal Holloway College

Abstract. The huge amount of resources available in the Grids, and the necessity to have the most up-to-date experimental software deployed in all the sites within a few hours, have driven the need for an automatic installation system for the LHC experiments. In this work we describe the ATLAS system for the experiment software installation in LCG/EGEE, based on the Light Job Submission Framework for Installation (LJSFi), an independent job submission framework for generic submission and job tracking in EGEE. LJSFi is able to automatically discover, check, install, test and tag the full set of resources made available in LCG/EGEE to the ATLAS Virtual Organization in a few hours, depending on the site availability.

1. Introduction

The huge amount of resources available in the Grids, and the necessity to have the most up-to-date experimental software deployed in all the sites within a few hours, have driven the need for an automatic installation system for the LHC experiments. To cope with these needs within the ATLAS collaboration, a fully automatic software installation system has been developed for LCG/EGEE. The installation system is based on the Light Job Submission Framework for installation (LJSFi), an independent job submission framework for generic submission and job tracking in EGEE. In this paper we will show the architecture and main features of the LJSFi framework.

2. The LJSFi architecture

LJSFi is a VO-independent framework for job tracking and task management in LCG/EGEE. The framework is a thin layer over the Grid middleware, built partially of shell scripts, to wrap the Grid commands, and python scripts to interface to the database.

The core system is based on the **Installation DataBase** and the command line (**CLI**) interface.

The CLI interface is responsible of the interaction with the Grid middleware. It uses the DB to store and retrieve the task definitions to perform the installation tasks. The task and job status are updated in the DB when a CLI command is invoked.

The Installation DataBase is the central store of the information about the resources, the jobs and the software deployment status. All the LJSFi components use the DB in order to get the configurations for specific tasks and to store the current status of the actions performed. The current DB implementation is based on MySQL v4. Other DB backend types are also supported by the system, including MySQL v5 cluster and Oracle.

The installation tasks and the data display are handled by the LJSFi modules and extensions.

The LJSFi architecture is modular and several extensions are possible by adding independent modules or extending the core to add different Grid middleware environments or setups.

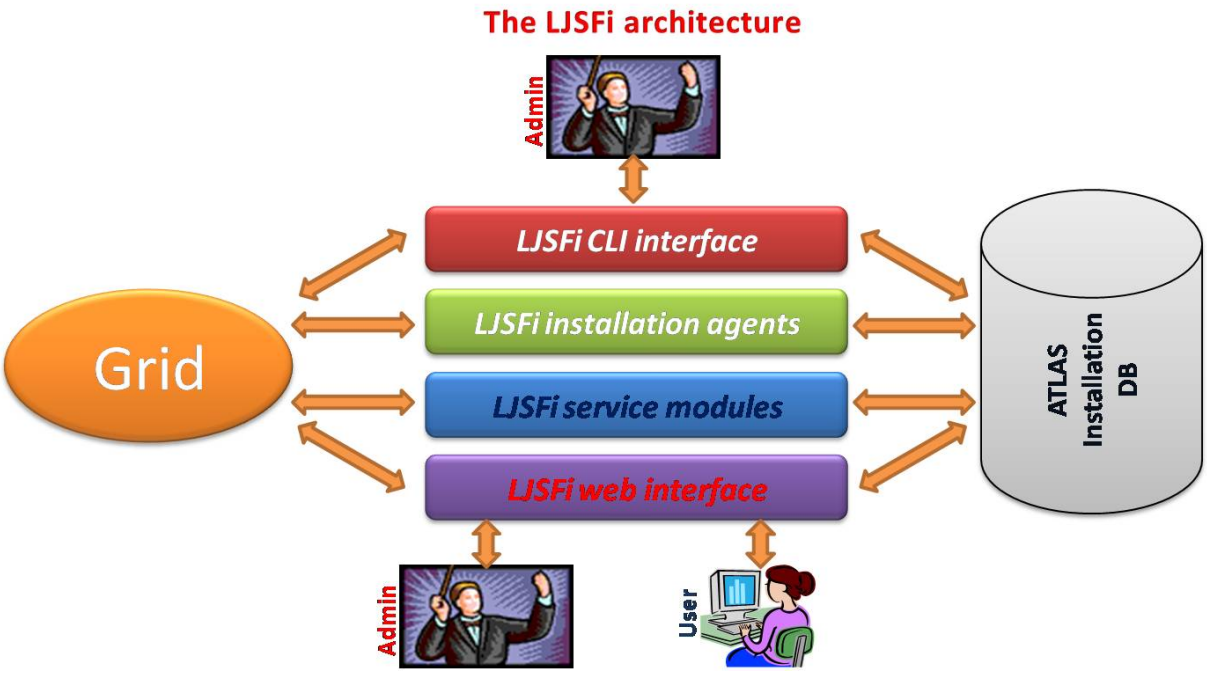

Figure 1
The LJSFi architecture

3. LJSFi modules

The LJSFi system has been instrumented with modules and extensions to handle installation requests. The installations or removals may be centrally triggered as well as requested by the end-users for each site. For this purpose the following components have been developed:

1. The RAI (Request An Installation) module;
2. The AIR (Automatic Installation Requester) module;
3. The InAgent (Installation Agent) module.

3.1. The **RAI** module

The RAI (Request An Installation) web interface is the portal to the user-driven installation requests. Each user is recognized by their credentials, obtained by checking the X509 personal certificate from the browser used to access the page. The page implements a SSL X509 security model, so only the users with a valid certificate, issued by a Certification Authority recognized by EGEE, are allowed to enter.

3.2. The **AIR** module

The centralised installations are triggered by the information stored in the installation database. When a release is tagged as production or obsolete and set to autoinstall, the LJSFi AIR (Automatic Installation Requester) module starts, respectively, the deployment or removal of the specified release, in all the sites where the relevant software tag is not yet published. LJSFi provides the needed locking mechanisms to avoid collisions among the installation jobs on the same resource or site. The AIR module is usually invoked each hour, in order to handle the installation requests promptly.

3.3. The **InAgent** module

The InAgent module is used to provide a fully automatic installation system. It reads the installation database every 10 minutes and starts the installation processes, interacting with the CLI interface. Each process is then handled by the installation agent, which updates the database in real-time with the task status, giving the online view of the software deployment.

The automatic installation tasks are performed by one or more agents, which may operate in concurrent mode.

A fallback option to manual operations is also available, in case of problems. The people from the Installation Team are able to manage all the installation tasks directly from the web interface, changing dynamically the parameters of the auto-installer for the specific task, or manually using the LJSFi CLI interface.

The first version of the InAgent module was introduced in late 2006, improving the scalability of the system and reducing the amount of manual work.

4. User services

Individual users may also ask the system for specific services:

1. *Pinning Releases*: each user may pin a release in one or more sites, to avoid the central system removing them while the pin is active. This is extremely important for releases used also locally at the sites. The pins may be released by any users, and the action is logged by the system;
2. *Action Notification*: each user may subscribe to the notification emails for one or more sites. Each time an action on a release on the specified site is performed, the system sends an email to all the subscribers to the given site.

5. Web interface

The installation data, status and job history are centrally kept in the MySQL installation DB and can be browsed via the LJSFi **Webber**[1] web interface. Also installation logfiles and summary files are made available via the same web portal, protecting the sensitive information of the sites by restricting the access only to the users with a valid certificate, recognized by LCG.

6. Grid Big Brother (GriBB)

All the installation tasks, including the LJSFi framework actions, are supervised by the GriBB (Grid Big Brother) watchdog agent.

GriBB provides an easy and comfortable way to introduce timeouts, limits and partial output retrieval for jobs still running in the Grid. In particular GriBB may be used to:

- Terminate the process when it exceeds the given limits for total time, memory size, stdout/sterr size, CPU % used
- Dump, on demand, the partial stdout/stderr of a job still running in the Grid without any need for installing external servers but only using existing DPM SEs (other SE types are being tested)
- Provide statistics on the CPU, disk and memory usage of the running job, at the end of the task or while still running.

7. The installation process

The software installation is performed in 3 steps:

1. Site checks;
2. Installation task;
3. Output validation.

The site checks are performed by sending a pilot job to each site where to install the software. After the site checks have been successfully executed, the actual installation process may start. The installation actions are performed in the target nodes (Worker Nodes) by the software management script, handling installations and removals of the ATLAS software distribution kit with pacman[2]. For each experiment software release installed at a site, a tag is published to the corresponding Computing Element at the end of the installation task. Several installation tasks are defined, corresponding to the actions to perform, including the software installation, testing, removal and tag management. At the end of the installation tasks, LJSFi retrieves the job output and exit code from the Grid middleware: the job output is uploaded to the web server via the Webber module, while the exit code is used to update the task status in the DB.

8. Current status and perspectives

The LJSFi version 1.0 system has been successfully used by ATLAS since 2003 to deploy around 15 different software releases. The system has been upgraded to LJSFi version 1.2 in 2005 and has performed more than 160000 installation jobs to deploy more than 70 releases so far.

In Figure 2 the number of successful and failed installation jobs per month of LJSFi v1.2 are shown. The plot does not include the site check jobs. The installations have been done either manually or through the use of a single instance of the automatic installation agent. The use of a single InAgent instance is adequate for the current needs and does not introduce any bottleneck in the system, anyway using more than one instance is also supported. The installation efficiency is shown in Figure 3. The average efficiency in the selected period is ~64% and is mostly dependent on the status of the sites and readiness to the ATLAS software. The installation failures due to problems in the framework are < 0.1%. The drop in the efficiency seen in the figure is mostly due to the activation of the AIR module in late 2006, requesting the installation of several new sites, not yet ready for the ATLAS software

deployment. The situation has become more stable since May 2007. The introduction of the fully automatic system gives the possibility to process more installation requests (Figure 4), reducing the amount of time needed to fully complete each request. In Figure 5 the average times needed to complete a request are shown. The values include the time to solve the problems in the sites, when needed. The time needed to install a site with a software release as a function of the time elapsed to complete the task is show in Figure 6: most of the sites may be installed in < 24 hours.

The LJSFi framework is currently being extended to the other WLCG Grids (NDGF and OSG), by directly using Condor as a submission system.

The latest version of LJSFi is available for download as pacman[2] package from the INFN ATLAS cache[3].

9. Conclusions

The LJSFi system has been proven to be flexible and robust enough to cope with the software installation needs of the ATLAS experiment. The evolution of the framework, with the introduction of the automatic installation agents and the user services, has improved the installation procedure scalability and speed, while decreasing the amount of manual work.

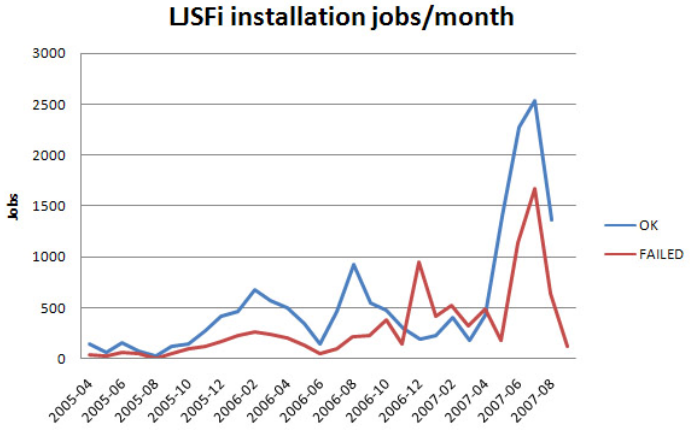

Figure 2
LJSFi v1.2 installation jobs per month. The plot does not include the pilot jobs used for the site checks.

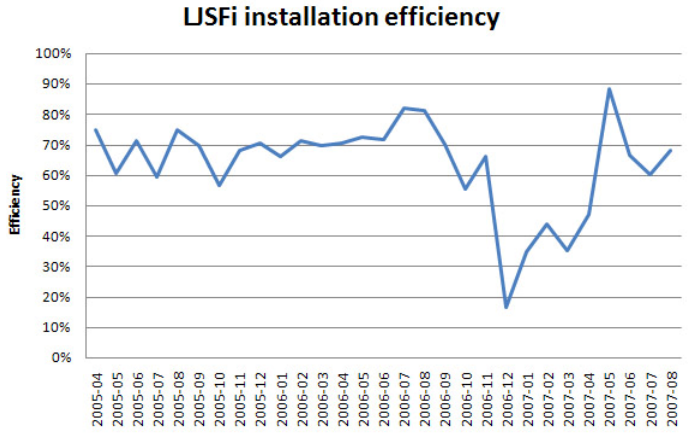

Figure 3
LJSFi v1.2 installation efficiency. The efficiency is dominated by the site status. The drop in the efficiency is due to the activation of the AIR module in late 2006, requesting the installation of several new sites, not yet ready for the ATLAS software deployment. The situation has become stable from May 2007.

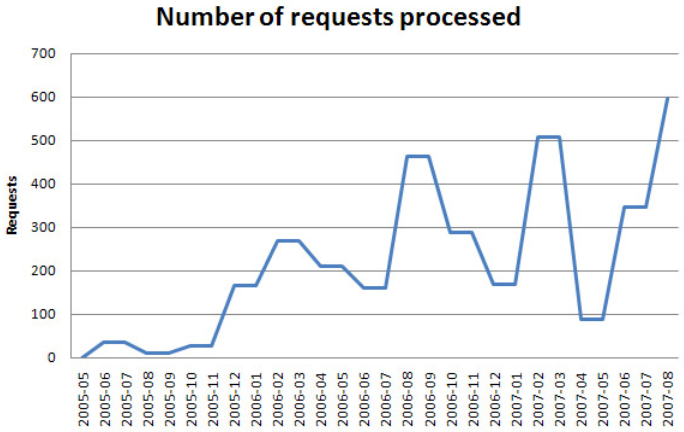

Figure 4
Number of requests processed by LJSFi, in bins of 2 months. The introduction of the fully automatic system has made possible to increase the number of installation requests processed per month.

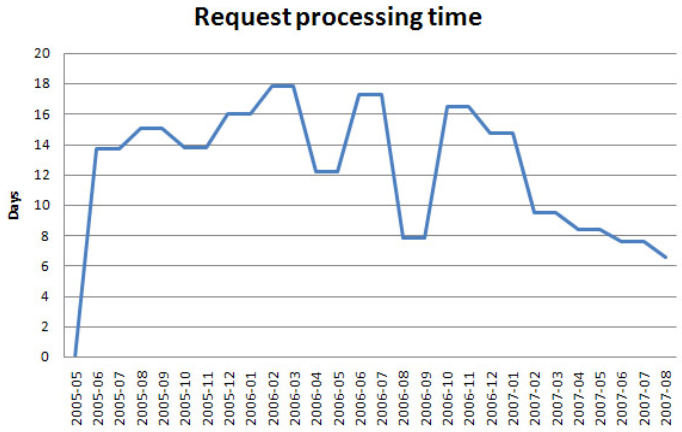

Figure 5
Request processing times, averaged over 2 months. The introduction of the fully automatic system has decreased the processing times per request up to a factor 2.

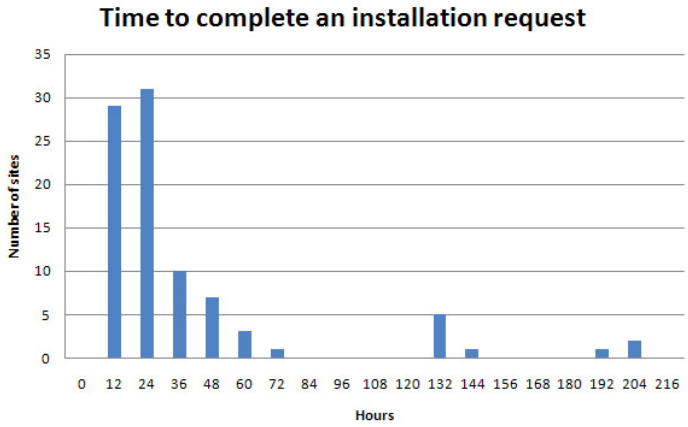

Figure 6
Time needed to complete an installation request, in hours. Most of the sites are installed within 24 hours after the installation request has been placed in the system.

References

[1] https://atlas-install.roma1.infn.it/atlas_install
[2] http://physics.bu.edu/~youssef/pacman/htmls/
[3] pacman -get http://classis01.roma1.infn.it/pacman/cache:LJSFi