Next Generation PanDA Pilot for ATLAS and Other Experiments

P Nilsson[1], F Barreiro Megino[2], J Caballero Bejar[3], K De[1], J Hover[3], P Love[4], T Maeno[3], R Medrano Llamas[2], R Walker[5], T Wenaus[3] for the ATLAS Collaboration

[1]University of Texas at Arlington (US), [2]CERN, [3]Brookhaven National Laboratory (US), [4]Lancaster University (UK), [5]Ludwig-Maximilians-Univ. Muenchen (DE)

Abstract

The Production and Distributed Analysis system (PanDA) has been in use in the ATLAS Experiment since 2005. It uses a sophisticated pilot system to execute submitted jobs on the worker nodes. While originally designed for ATLAS, the PanDA Pilot has recently been refactored to facilitate use outside of ATLAS. Experiments are now handled as plug-ins such that a new PanDA Pilot user only has to implement a set of prototyped methods in the plug-in classes, and provide a script that configures and runs the experiment specific payload. We will give an overview of the Next Generation PanDA Pilot system and will present major features and recent improvements including live user payload debugging, data access via the Federated XRootD system, stage-out to alternative storage elements, support for the new ATLAS DDM system (Rucio), and an improved integration with glExec, as well as a description of the experiment specific plug-in classes. The performance of the pilot system in processing LHC data on the OSG, LCG and Nordugrid infrastructures used by ATLAS will also be presented. We will describe plans for future development on the time scale of the next few years.

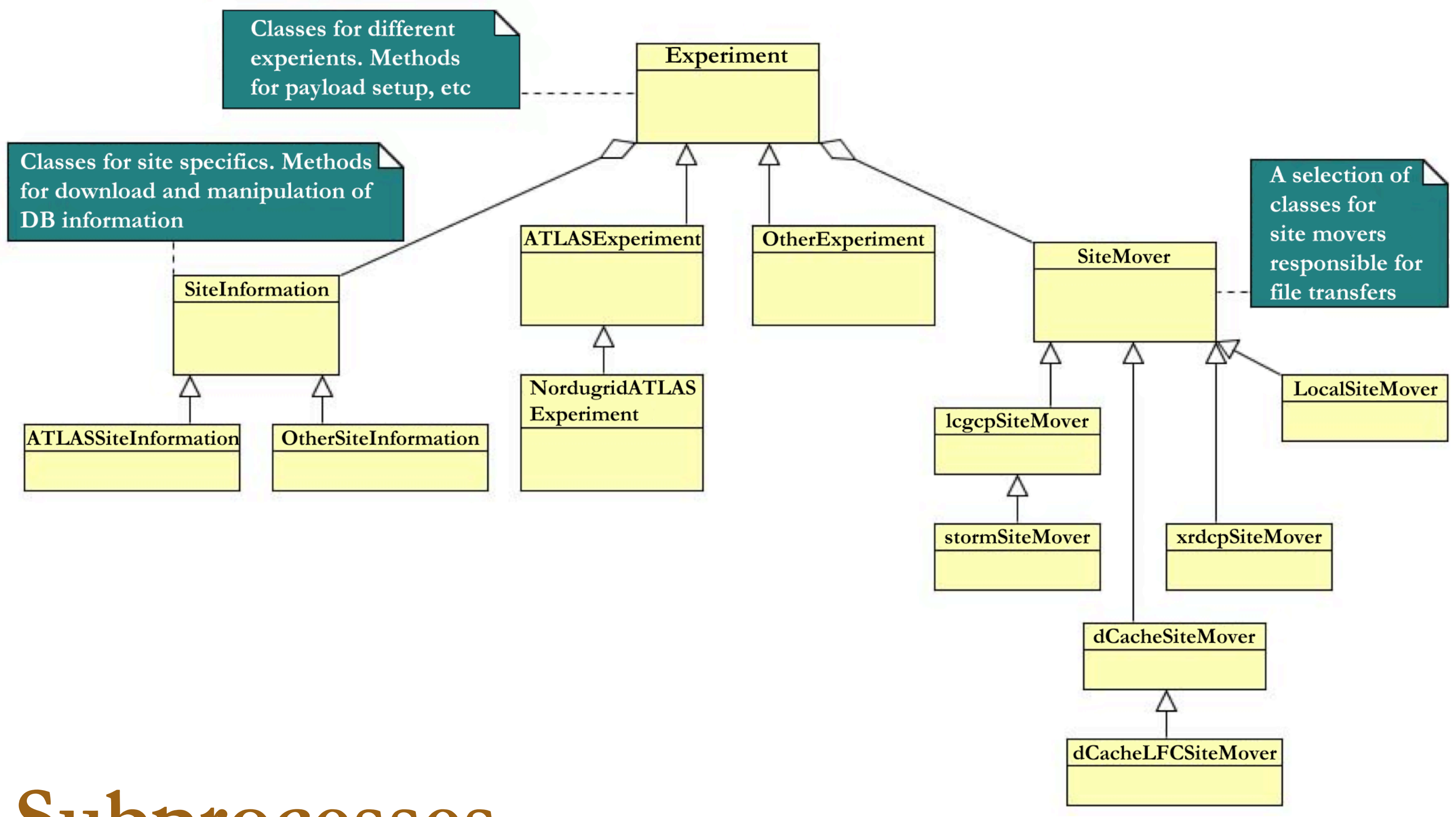
Introduction

A common approach in grid computing is to use pilot jobs. In the case of ATLAS [1], pilot factories are used to send special lightweight jobs, called pilot wrappers, to the batch systems that execute them on the worker nodes. The pilot wrappers download the PanDA Pilot [2] and launch it using pilot options that are relevant to the site in question. The responsibility of the pilot is to download the actual payload from the PanDA Server and any input file from the local Storage Element (SE), execute the payload, upload the output to the SE, and send the final job status to the server.

PanDA [3] has been very successful in managing the distributed analysis and production requirements across all ATLAS grids; OSG [4], EGI [5] and Nordugrid [6]. Today PanDA is being considered for use beyond ATLAS by several other experiments. To meet this need, it has been necessary to refactor the PanDA Pilot which until recently has been ATLAS specific.

Plug-in Mechanism

An adopting PanDA Pilot user should in principle only have to implement certain methods in the plug-in Experiment and SiteInformation classes. The plug-in classes contain methods that are experiment specific but sorted into two different classes. The Experiment classes contain methods related to payload setup, how the subprocess (responsible for the payload) should be launched, how metadata should be handled, which files should be removed from the payload work directory before the job log file is created, etc. The SiteInformation classes contain methods for handling site information from a DB, how it should be downloaded, from where, and how to verify its integrity, as well as how to manipulate it.


Classes for different experients. Methods for payload setup, etc
Experiment
Classes for site specifics. Methods for download and manipulation of DB information
SiteInformation
ATLASExperiment
OtherExperiment
SiteMover
A selection of classes for site movers responsible for file transfers
ATLASSiteInformation
OtherSiteInformation
NordugridATLAS Experiment
lcgcpSiteMover
LocalSiteMover
stormSiteMover
xrdcpSiteMover
dCacheSiteMover
dCacheLFCSiteMover

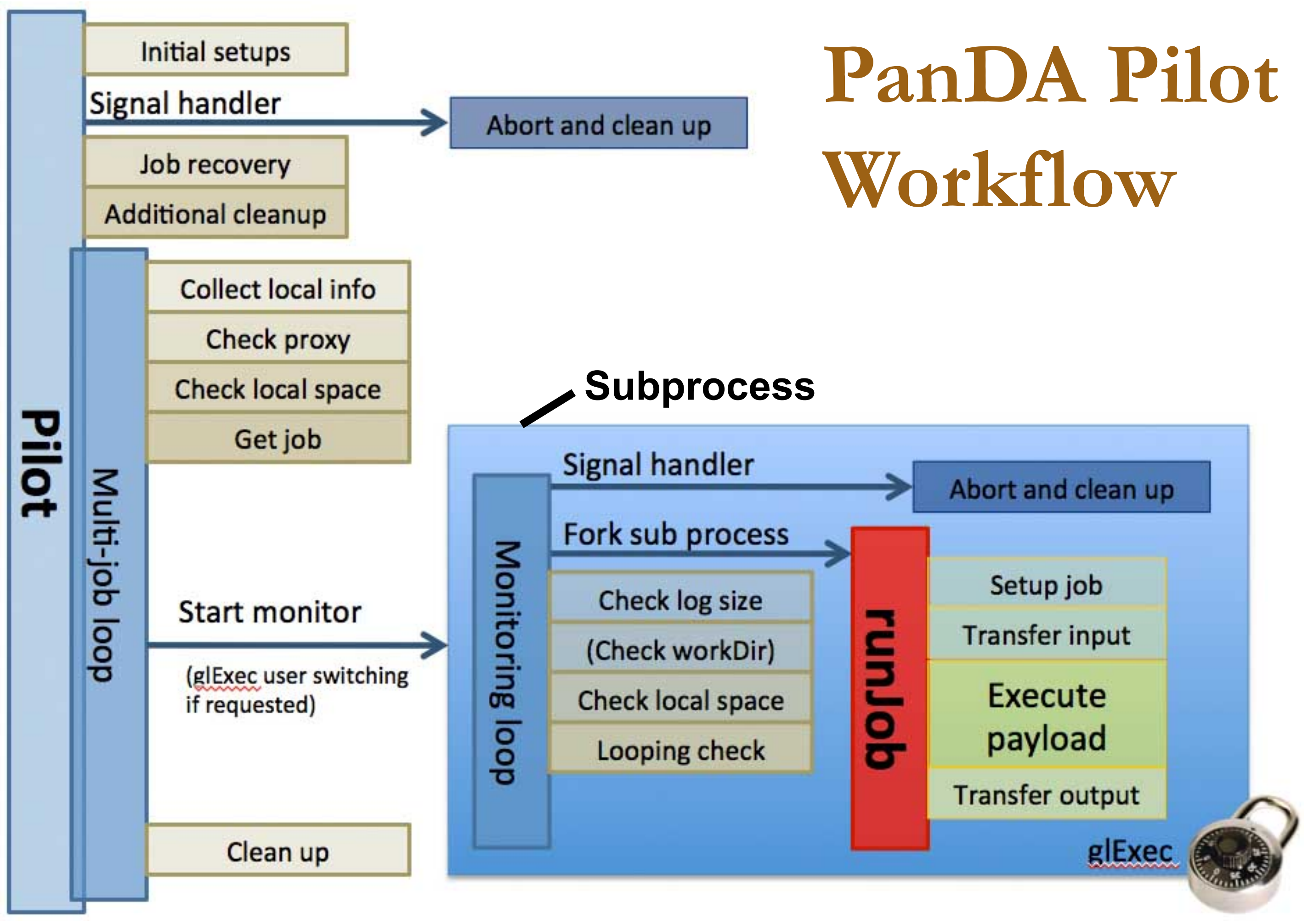
Subprocesses

A pilot module (called Monitor) forks and monitors a subprocess that is responsible for the payload. This subprocess can in principle be any module that needs the full attention and supervision of the Monitor. The primary PanDA Pilot subprocess module is called runJob, and is responsible for payload setup, stage-in of input files, execution of the payload and stage-out of output files. Two additional subprocess modules are currently in development; runEvent will be used to read and process events from an Event Server, and runHPCJob will be used for handling HPC jobs.

Plans for Future Developments

The PanDA Pilot has been refactored to facilitate the development of experiment specific classes. The Common Analysis Framework collaboration between ATLAS and CMS [7], has resulted in the development of experiment classes for CMS. Furthermore, AMS [8] is planning to adopt the PanDA system. Improving and further developing the PanDA Pilot for serving multiple experiments is of highest priority. To this end, several projects are foreseen including a new version of the job recovery mechanism [2], providing a full PanDA Pilot documentation, improving error reporting in a multi-experiment environment, and having pilot support for Event Server jobs and HPC's.

PanDA Pilot Workflow



Recent New Features

✖ **Live payload debugging.** The user can request live payload debugging when launching the grid jobs using pathena or prun [9]. When the pilot downloads a corresponding job, it will receive a special instruction in the job definition to frequently upload debugging information about the payload. This information will be made available on the PanDA Monitor job page for viewing by the user.

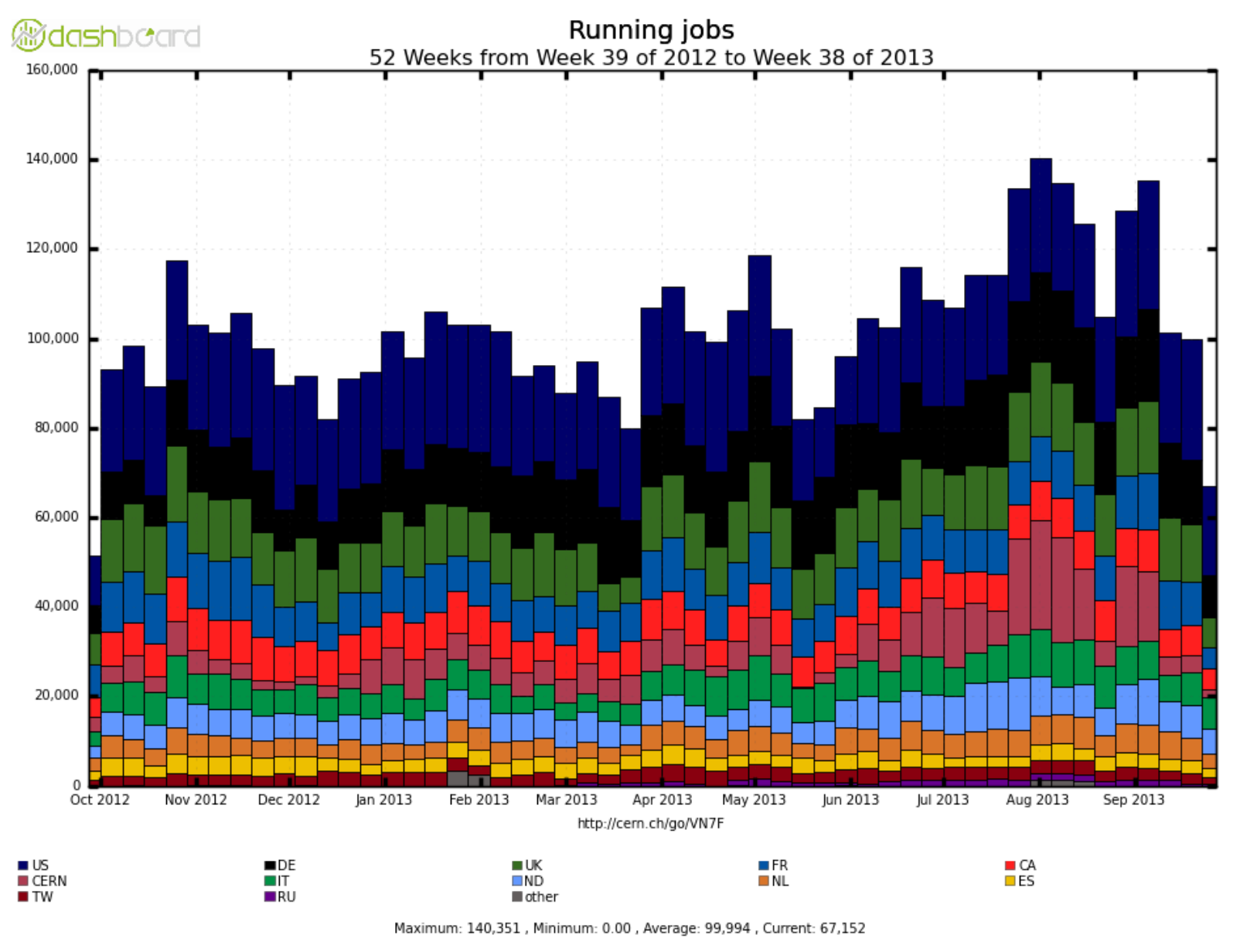
✖ **Data access via the Federated XRootD system.** The pilot has the option to attempt stage-in from a remote SE using the Federated ATLAS XRootD (FAX) [10] system. The FAX system consists of several dozens of sites accessed by hundreds of clients that act like a single storage resource. A special FAX Site Mover was developed for the PanDA Pilot, which means that the pilot can also use it as a primary copy tool, and not only as a fail-over mechanism which makes it interesting for "diskless" sites.

✖ **Stage-out to alternative storage elements.** The pilot is equipped with a mechanism for stage-out to an alternative Storage SE. The idea is that if the pilot fails (partially or completely) to stage-out the output files at the primary SE, it can re-attempt the stage-out on an alternative SE in the same cloud.

✖ **Rucio.** The pilot now has support for the new ATLAS DDM system (Rucio [11]), i.e. file paths used during stage-in/out can follow Rucio convention.

✖ **glExec.** The pilot has been refactored to enable proper introduction of glExec [12], while keeping the highly useful multi-job functionality (the ability to run several jobs from different users sequentially).

Performance



The PanDA system is serving over 100k production jobs and 35k user analysis jobs concurrently. It is performing very well with high job efficiency. The error rate in the entire system is very low. For production jobs the majority of the errors are site or system related, while for user analysis jobs the most common issues are related to application software. Pilot mechanisms like job recovery and FAX failover contribute to the robustness against site related failures.

References

[1] The ATLAS Experiment: *ATLAS Technical Proposal*. ATLAS Collaboration. CERN/LHCC/94-43, 1994

[2] P. Nilsson et al, Proc. of the 19th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2012)

[3] T. Maeno et al, Proc. of the 18th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP 2010)

[4] Open Science Grid: http://www.opensciencegrid.org

[5] European Grid Initiative: http://www.egi.eu

[6] M. Ellert et al., NIM A, 2003, Vol. 502

[7] CMS Collaboration, *JINST* **3** (2008) S08004, doi:10.1088/1748-0221/3/08/S08004

[8] R. Battiston, Nucl. Instrum. Methods Phys. Res., Sect. A 588, 227 (2008)

[9] Distributed analysis on PanDA: https://twiki.cern.ch/twiki/bin/viewauth/AtlasComputing/DAonPanda

[10] R. Gardner et al., Proc. of the 19th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP 2012)

[11] Rucio: http://rucio.cern.ch

[12] D. Groep, O. Koeroo, G. Venekamp, J.Phys.:Conf.Series 119 (2008) 062032