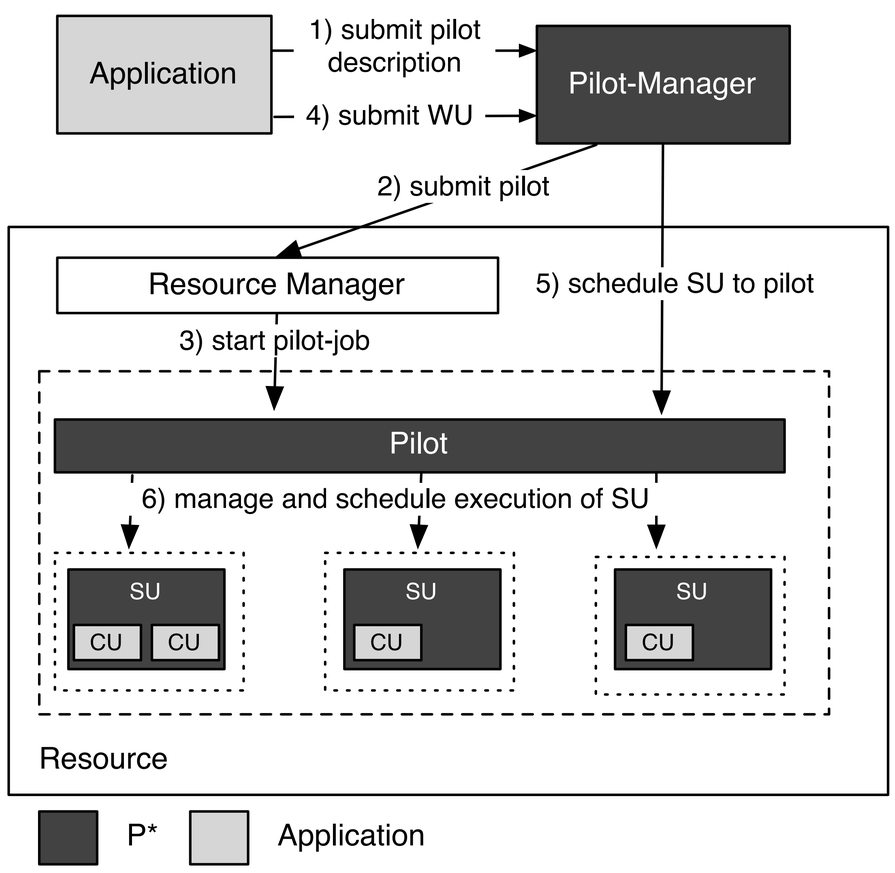
P*: Towards a Common Model for Pilot-Jobs

Andre Luckow, Mark Santcroos, Ole Weidner, Andre Merzky, Pradeep Mantha, Shantenu Jha

Pilot-Jobs (PJ) have become one of the most successful abstractions in distributed computing. In spite of extensive uptake, there does not exist a well defined, unifying conceptual model of Pilot-Jobs, which can be used to define, compare and contrast PJ implementations. This presents a barrier to extensibility and interoperability. This work is an attempt to, provide a minimal but complete model (P*) of Pilot-Jobs and to demonstrate the interoperable and concurrent usage of distinct pilot-job frameworks on different production distributed cyberinfrastructures via the use of an extensible API for the P* Model (Pilot-API).

Pilot-Jobs provide an effective abstraction for dynamic execution and resource utilization. As a consequence of providing a simple approach for decoupling workload management and resource assignment/scheduling, Pilot-Jobs have been one of the most successful abstractions in distributed computing.



P* Model

The **P* Model** provides a minimal, but complete model of Pilot-Jobs and provides a conceptual basis to compare and contrast different PJ frameworks. The P* Model comprises of the following elements:

Pilot: The pilot is the entity that actually gets submitted and scheduled on a resource. The PJ provides application (user) level control and management of the set of allocated resources.

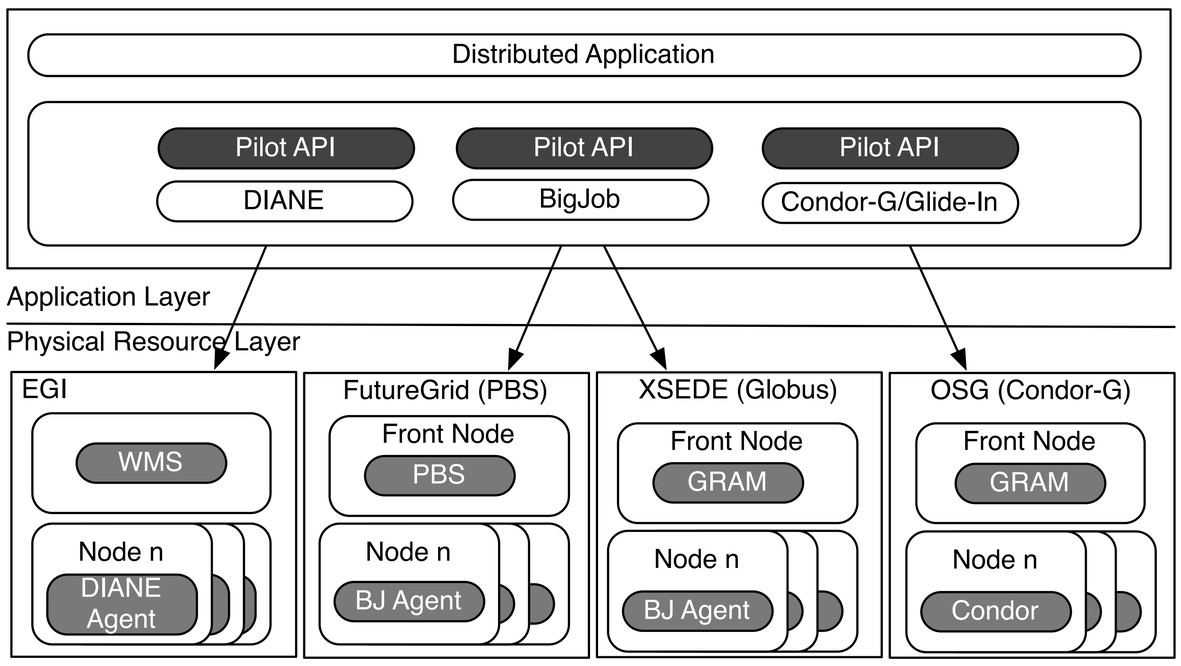
Compute Unit (CU): A CU encapsulates a self-contained piece of work (a task) specified by the application that is submitted to the Pilot-Job framework. There is no intrinsic notion of resource associated with a CU.

Scheduling Unit (SU): SUs are the units of scheduling internal to the P* Model, i.e., it is not known by or visible to an application.

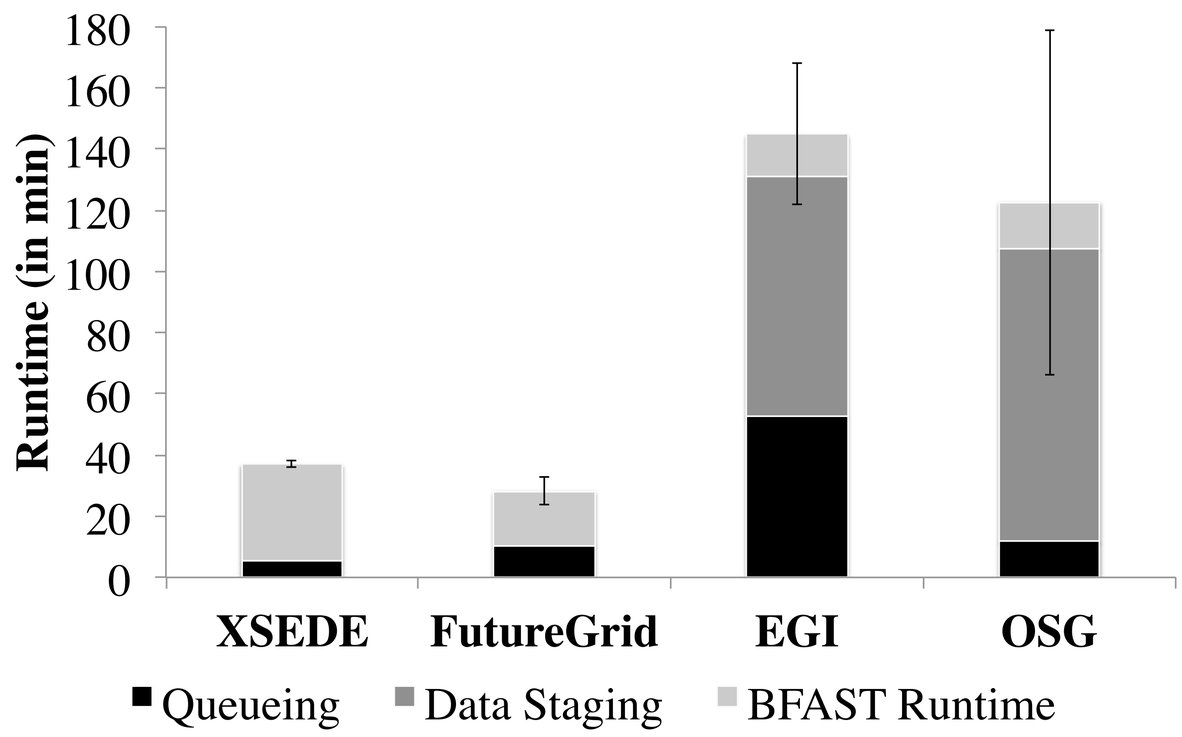
Pilot-Manager (PM): The PM is responsible for (i) orchestrating the interaction between the Pilots as well as the different components of the P* Model (CUs, SUs) and (ii) decisions related to internal resource assignment.

Pilot-API

The aim of the Pilot-API [2] is to provide an abstract, unified interface to PJ frameworks that adhere to the P* Model, e.g. BigJob [1], DIANE and Condor/Condor-G (see figure).



The Pilot-API has been used to support a wide range of infrastructures, applications types and usage modes. For example, we successfully utilized the Pilot-API and different distributed infrastructures with BFAST, a genome sequencing application. BFAST is very I/O sensitive, which leads to an I/O bottleneck if many BFAST CUs run on the same shared file system. The Pilot-API enables applications to scale to different infrastructures in such cases. The following figure shows how the Pilot-API is used to run BFAST interoperably on different production and research infrastructures.



References:

[1] A. Luckow, L. Lacinski, and S. Jha, “SAGA BigJob: An Extensible and Interoperable Pilot-Job Abstraction for Distributed Applications and Systems,” in IEEE/ACM CCGrid, 2010

[2] Pilot API, https://github.com/saga-project/BigJob/ blob/master/pilot/api/, 2012.

Acknowledgements:

This work is funded by NSF CHE-1125332 (Cyber-enabled Discovery and Innovation), HPCOPS NSF-OCI 0710874 award, NSF-ExTENCI (OCI- 1007115) and NIH Grant Number P20RR016456 from the NIH National Center For Research Resources. Important funding for SAGA has been provided by the UK EPSRC grant number GR/D0766171/1 (via OMII-UK) and the Cybertools project (PI Jha) NSF/LEQSF (2007-10)-CyberRII-01, NSF EPSCoR Cooperative Agreement No. EPS-1003897 with additional support from the Louisiana Board of Regents. SJ acknowledges the e-Science Institute, Edinburgh for supporting the research theme. “Distributed Programming Abstractions” & 3DPAS. MS is sponsored by the program of BiG Grid, the Dutch e-Science Grid, which is financially supported by the Netherlands Organization for Scientific Research, NWO. SJ acknowledges useful related discus- sions with Jon Weissman (Minnesota) and Dan Katz (Chicago). We thank J Kim (CCT) for assistance with BFAST. This work has also been made possible thanks to computer resources provided by TeraGrid TRAC award TG-MCB090174 (Jha) and BiG Grid. This document was developed with support from the US NSF under Grant No. 0910812 to Indiana University for “FutureGrid: An Experimental, High-Performance Grid Test-bed”. PM acknowledges OSG and XSEDE organizers for supporting travel to XSEDE'12.

RUTGERS