Pilot Abstractions for Compute, Data, and Network

Mark Santcroos
Academic Medical Center
University of Amsterdam
The Netherlands
m.a.santcroos@amc.uva.nl

Silvia Delgado Olabarriaga
Academic Medical Center
University of Amsterdam
The Netherlands
s.d.olabarriaga@amc.uva.nl

Daniel S. Katz
University of Chicago &
Argonne National Laboratory
USA
d.katz@ieee.org

Shantenu Jha
Rutgers University
USA
shantenu.jha@rutgers.edu

***Index Terms*—Pilot-Data, Pilot Abstraction, Distributed and Data-intensive, Networking**

Abstract

Scientific experiments in a variety of domains are producing increasing amounts of data that need to be processed efficiently. Distributed Computing Infrastructures are increasingly important in fulfilling these large-scale computational requirements.

Data is distributed for a variety of reasons. Sometimes data is distributed by nature, such as in large science experiments involving federations of institutes and related equipment. Such data is not produced (nor stored, therefore) at a single location, and needs to be transported for analysis elsewhere. Also, security and/or privacy concerns can restrict where data can be stored, archived, and/or processed. Having control over the network paths where this data is distributed (transported) provides a means to distribute data only among trusted locations. Finally, processing data that is spread across multiple compute resources benefits data intensive applications that would otherwise strain local I/O systems.

With the rapid growth of data volumes, it is often no longer practical to simply move all the "data" to the "job"; instead, data needs to be processed on a nearby compute resource. In this new paradigm where both compute and data are dynamically scheduled and placed, it is desirable to take network conditions into account, or to dynamically provision the network. In other words, networks should be considered as a third degree of freedom, in addition to compute and storage resources, for optimal scheduling of the processing of scientific data.

When the network topology is static, network sensing can be used to decide to move data and if so, which network paths to use. When frequent sensing/measuring capabilities are in place, the decisions and planning can be extended by actively reserving network paths. "Dynamic lightpaths" provide the ability to perform such reservations at a very fine grained level.

For the execution of scientific analysis software, Pilot-Jobs support effective distributed resource utilization, and are arguably one of the most widely used distributed computing abstractions on production Distributed Computing Infrastructures. Pilot-Jobs enable the separation of task management and resource allocation, and are successful because they free applications/users from the challenging procedure of mapping specific tasks onto heterogeneous and dynamic resource pools. Pilot-Jobs thus shield applications from having to load-balance tasks across such resources.

In earlier work we presented a well defined, unifying conceptual model of Pilot-Jobs, called P*, which can be used to define, compare and contrast different implementations [1]. From the model and the currently existing Pilot-Job frameworks, we derived an interoperable and extensible API for the P* Model (Pilot-API). We also established a natural and logical extension of the P* Model to include data in addition to computational tasks [2]. This leads to a new abstraction analogous to Pilot-Job: Pilot-Data, where a data item (e.g., a file) is separated from its physical location in some distributed resource.

Using the Pilot-Job and Pilot-Data abstractions exposed through the Pilot-API, an application can express affinity between data and compute. For example a user can specify that the data item produced by one task will be consumed by another task without explicitly specifying its physical location. This affinity can be taken into account by the Pilot-Manager to route the data and tasks to the physical resources and make more efficient use of the infrastructure.

The Pilot-Manager that is responsible for scheduling can be extended to also take into account network resources. For example, the decision on where to locate (or co-locate) a pair of computational jobs that produce/consume common intermediate data may depend on expected network characteristics at the time the first job will finish. Similarly, data security characteristics of the network can be used to decide upon data transfers between resources.

Figure 1 shows the components involved. The application interfaces to the Pilot-Data Manager and Pilot-Job Manager using the Pilot-API. The Pilot-Data Manager and Pilot-Job Manager interact with the resource schedulers to obtain the required resources, and schedule the application tasks and data accordingly. The Network Manager senses, maintains the state of, and schedules the network. It is used by the Pilot-Job Manager and Pilot-Data Manager for their scheduling decisions.

Using a well defined and integrated API for compute, data, and network relieves an application programmer from having to program against multiple and specific API for various resources.

In this work we will explore the abstractions to effectively program these three types of resources together using the Pilot-API. We will illustrate the benefit of our proposal using a

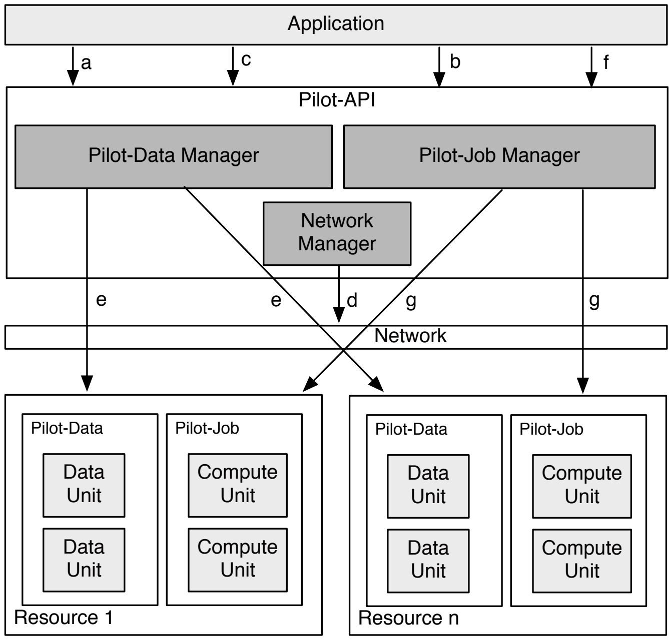

Fig. 1. Pilot-Data stack with Network Management. The application allocates Pilot-Job (a) and Pilot-Data (b) resources through the Pilot-API. The application also describes the Data Units and passes these to the Pilot-Data Manager (c). The Network Manager knows the state of the network and possibly interacts with it (d). The Pilot-Data Manager is responsible for transferring (e) the Data Units to their physical locations (Pilot-Datas). When the data is in place (or earlier), the Application can submit Compute Units (f) that will run in Pilot-Jobs (g).

typical Next Generation Sequencing pipeline based on BWA and Picard to attain faster time to completion without adding complexity for the user and/or the application programmer. This pipeline has also been used previously to process data from a large Dutch population study [3], where 770 persons were sequenced, producing a total of 45 TB of data that had to be aligned to the Human Genome. A full description of that project is out of scope, but because of the large scale, many problems arose from the inability to efficiently schedule data and compute on the Dutch e-science infrastructure [4]. Our experiments use SURFnet's Dynamic Lightpaths, which connect several sites of the Dutch Life Science Grid. We use the Network Service Interface (NSI) [5] to schedule the network from our system.

References

[1] Andre Luckow, Mark Santcroos, Ole Weider, Andre Merzky, Sharath Maddineni and Stantenu Jha, Towards a Common Model for Pilot-Jobs, Proceedings of The International ACM Symposium on High-Performance Parallel and Distributed Computing, 2012, Delft, The Netherlands

[2] Andre Luckow, Mark Santcroos, Ole Weidner, Andre Merzky, Pradeep Mantha and Shantenu Jha, P*: A Model of Pilot-Abstractions, Proceedings of the 8th IEEE International Conference on e-Science, 2012, Chicago, IL, USA

[3] Genome of the Netherlands project, http://www.nlgenome.nl/

[4] BiG Grid: the dutch e-science grid, http://www.biggrid.nl/

[5] Network Service Interface WG, Open Grid Forum, https://forge.ogf.org/sf/projects/nsi-wg

Authors Biographies

Mark Santcroos

Mark is a researcher in the e-BioScience group of the Bioinformatics Laboratory at the Academic Medical Center of the University of Amsterdam. In addition to that he is affiliated to the SAGA project. A computer scientist by training with a strong interest in bridging practice and theory by developing novel methods and technologies and applying those to real world problems. His involvement with distributed systems goes back to 2000, first in the field of networking and measurements and later in distributed computing infrastructures. Currently he participates in Dutch, European and US e-science projects.

Silvia Delgado Olabarriaga

Silvia is an Assistant Professor at the Academic Medical Center (AMC) of the University of Amsterdam. She leads the e-science group, which bridges technical and communication gaps between available public e-infrastructures (the Dutch e-science grid), biomedical users that need these infrastructures for big data analysis, and novel concepts from distributed computing. Silvia has a background in computer science (computer animation and visualization) and medical imaging (interactive image segmentation), and has been active in e-science research since 2005, taking part in various Dutch and European projects.

Daniel S. Katz

Dan is a Senior Fellow at the Computation Institute (CI) of the University of Chicago and Argonne National Laboratory, and an Adjunct Associate Professor at Louisiana State University. The CI is an intellectual nexus and resource center for scholars from multiple disciplines building and applying computational platforms for science. Dan has a background in computational electromagnetics, and experience in a wide range of computational sciences, computer science, and cyberinfrastructure. He has been active in cyberinfrastructure/e-science research since 1987, taking part in various US and international projects.

Shantenu Jha

Shantenu is an Assistant Professor at Rutgers University, a member of the Graduate Faculty in the School of Informatics at the University of Edinburgh (UK), and a Visiting Scientist at University College London. He is also the Associate Director for Advanced Research Cyberinfrastructure at the nascent Rutgers Discovery Informatics Institute. His research interests lie at the triple point of Applied Computing, Cyberinfrastructure R&D and Computational Science. Shantenu is the lead investigator of the SAGA project (http://www.saga-project.org). His research has been funded by multiple NSF awards, US Department of Energy (DoE), US National Institute for Health (NIH) as well as the UK EPSRC (OMII-UK project and Research theme at the e-Science Institute).