Towards Scalable Distributed Applications and Systems: The P* Model of Pilot-Abstractions

Shantenu Jha
The RADICAL Group
http://radical.rutgers.edu

22nd June 2012
Grand Challenge: Building the Global Cyberinfrastructure

ExTENCI Home

Extending Science Through Enhanced National CyberInfrastructure (ExTENCI) is a joint project of the Open Science Grid (OSG) and TeraGrid/XSEDE, funded by the National Science Foundation.
A Pore Man’s View of the TeraGrid/XSEDE

2005-09: TG Did not help manage the “distributed workload”!
The Challenge of Integrating Compute and Data at Scale
Overview

• Grand Challenge: How to build the CI for Sciences?
  – Understanding the Landscape of Distributed CI and Applications
  – Lack of a Theoretical Foundation of Middleware

• P* Model of Pilot-Abstractions
  – SAGA: A Standards layer upon which other abstractions are built
  – Uniform abstraction for compute and data
  – Interoperability and Extensibility

• P* Implementations are Supporting Large-scale S&E
  – Map-Reduce based next-generation gene sequencing data
  – Generalized ensemble and advanced sampling algorithms
    • Structural biology and patient-specific medicine
Landscape of DCI and Applications: Empirical Assertions

1. Developing DA is a hard undertaking, the space of possible DA is large (and rich), but the number of effective and extensible DA small
   - More than just submitting jobs here and there!
   - Intrinsic and Extrinsic Factors; coordinating across resources

2. Accept, if not embrace “distributedness”
   - Manage it, but also exploit it
   - The ability to reason: distributed performance, decomposition or aggregation!

3. There are missing abstractions (and often poor implementations)
   - Many local solutions, lack of end-to-solutions, especially tools
   - Missing conceptual frameworks and systems approach to tools

4. Clouds are an important and useful development but not panacea
DCI Challenges

- **Infrastructure Perspective: Complexity**
  - Middleware: Heterogeneity and semantic incompatibility
  - Slow/lack of agreement towards a standards-based solution
  - Difficult to integrate services & software (service-level Interop)
  - This is not Google (i.e., uniform, single top-down solution), as much of a socio-cultural challenge as a technical one

- **Application Perspective:**
  - End-to-end application support missing Many moving and changing parts, insufficient / partial coverage
  - Well defined interfaces Standard-layers handle hard parts, allowing innovation elsewhere

- **Effective dynamic resource utilization and execution models**
  Need a middleware framework that is extensible and interoperable
“Integrated” End-to-End Capabilities

L0: Applications

L1: Programming Abstractions and Models
   e.g. pipelines, workflows, loosely-coupled

L2: Middleware and Services
   e.g. task-data schedulers, data-caching, accelerators...

L3: Execution Platform (Underlying Infrastructure)
   e.g. Compute-resources, Distributed file-systems, Cloud technologies
Middleware IDEAS: Development Objectives

- **Interoperable**: Heterogeneity and Environmental Complexity
  - Ability to work within and across multiple DCI concurrently
  - Horizontal (Service) and Vertical Interoperability (Application)
- **Dynamic**: Beyond static execution & resource allocation models
  - Decisions at both deployment and run-time
  - Dynamical execution is almost fundamental at scale
- **Extensible**: Support new functionality & infrastructure without wholesale refactoring, i.e., lower coupling to tools & infrastructure
  - Abstractions based tools and systems building
- **Adaptive/Autonomic**: Flexible formulation, self-*
- **Scalable**: Along many dimensions and design points

**Challenge**: To develop DA and provide DCI, effectively and efficiently with IDEAS as first class objectives, with simplicity an over-aching concern
P* Model of Pilot-Abstractions

- Applications
- Tools
- Pilot-MapReduce
- Pilot-API
- P*
  - Pilot-Jobs
  - Pilot-Data
- SAGA
- Resource (Network, Compute & Data) Access Layer
- Physical Infrastructure
  - e.g. XSEDE, EGI, OSG, ESNet, Clouds
SAGA – An Overview

SAGA provides the base layer upon which other abstractions and capabilities are provided

http://www.saga-project.org
SAGA: Abstraction upon which other abstractions are built

• HOW SAGA is Used?
  – Uniform Access-layer to DCI
    • EGI, XSEDE, DATAONE, UK NGS and NAREGI/RENEKI and Clouds
  – Build tools, middleware services and capabilities that use DCI
    • e.g. Gateways, *Pilot-Jobs*
  – Application “Scripting Layer” to DCI
    • Improved and enhanced HTHP ensembles

• WHAT is SAGA Used for?
  – Support production-grade science and engineering
  – Research tool to design, implement reason about distributed programming models, systems and applications
Understanding Pilot-Jobs: Abstractions for Dynamic Execution

SAGA Pilot-Job (BigJob)
P* Model of Pilot-Abstractions
Towards a common model for Pilot-jobs, Luckow, Santcroos, Merzky, Weidner and Jha, to appear in proceedings of HPDC’12

<table>
<thead>
<tr>
<th>P* Element</th>
<th>BigJob</th>
<th>DIANE</th>
<th>Condor-G/Glide-in</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pilot-Manager</td>
<td>BigJob Manager</td>
<td>RunMaster</td>
<td>condor_master</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>condor_collector</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>condor_negotiator</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>condor_schedd</td>
</tr>
<tr>
<td>Pilot</td>
<td>BigJob Agent</td>
<td>Worker Agent</td>
<td>condor_master</td>
</tr>
<tr>
<td>Compute Unit (CU)</td>
<td>Task</td>
<td>Task</td>
<td>Job</td>
</tr>
<tr>
<td>Scheduling (SU)</td>
<td>Unit</td>
<td>Sub-Job</td>
<td>Task</td>
</tr>
</tbody>
</table>

**TABLE 1: Mapping P* elements and PJ Frameworks:** While each PJ framework maintains its own vocabulary, each of the P* elements can be mapped to one (or more) components of the different PJ frameworks.
<table>
<thead>
<tr>
<th>Properties</th>
<th>BigJob</th>
<th>DIANE</th>
<th>Condor-G/Glide-in</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coordination</td>
<td>M/W</td>
<td>M/W</td>
<td>M/W</td>
</tr>
<tr>
<td>Communication</td>
<td>Advert Service</td>
<td>CORBA</td>
<td>TCP</td>
</tr>
<tr>
<td>Scheduling</td>
<td>FIFO, custom</td>
<td>FIFO, custom</td>
<td>Matchmaking, priority-based scheduler</td>
</tr>
<tr>
<td>Binding</td>
<td>Early/Late</td>
<td>Late</td>
<td>Late</td>
</tr>
<tr>
<td>Agent Submission</td>
<td>API</td>
<td>GANAG Submission Script</td>
<td>Condor CLI</td>
</tr>
<tr>
<td>End User Environment</td>
<td>API</td>
<td>API and M/W Framework</td>
<td>CLI Tools</td>
</tr>
<tr>
<td>Fault Tolerance</td>
<td>Error propagation</td>
<td>Error propagation, retries</td>
<td>Error propagation, retries</td>
</tr>
<tr>
<td>Resource abstraction</td>
<td>SAGA</td>
<td>GANAG/SAGA</td>
<td>Globus</td>
</tr>
<tr>
<td>Security</td>
<td>Multiple (GSI, User/-Pass.)</td>
<td>Multiple (GSI)</td>
<td>Multiple (GSI, Kerberos)</td>
</tr>
</tbody>
</table>

*TABLE II: P* Characteristics and Properties of Different Pilot-Job Frame-*
Exposing the P* Model: The Pilot-API

**Diagram: Pilot-Job Framework**

1. **Application**
   - `create_pilot` (PilotComputeDescription)
2. **PilotComputeService**
   - `start` (Pilot)
3. **Application**
   - `submit` (ComputeUnitDescription)
4. **ComputeUnitService**
   - `schedule` (SchedulingUnit)

**Distributed Application**

- **Application Layer**
  - Pilot API
  - DIANE
  - Pilot API
  - BigJob
  - Pilot API
  - Condor-G/Glide-In

- **Physical Resource Layer**
  - EGI
    - WMS
    - DIANE Agent
  - FutureGrid (PBS)
    - Front Node
    - PBS
    - Node n
    - BJ Agent
  - XSEDE (Globus)
    - Front Node
    - GRAM
    - Node n
    - BJ Agent
  - OSG (Condor-G)
    - Front Node
    - GRAM
    - Node n
    - Condor
Accessing Multiple DCI & Pilot-Jobs via Pilot-API

128 BFAST tasks, O(10) GB

Runtime (in min)

<table>
<thead>
<tr>
<th></th>
<th>Staging</th>
<th>Compute</th>
</tr>
</thead>
<tbody>
<tr>
<td>BigJob/XSEDE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Trestles)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BigJob/FG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(India)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BigJob/FG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(India/Sierra)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIANE/EGI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GlideInWMS/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OSG</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Pilot-Job Interoperability via Pilot-API

![Graph showing runtime (in min) for different systems](image-url)
Pilot-Abstraction for Dynamic Distributed Data?

- Similar levels of heterogeneity in the data infrastructure
  - File systems, storage, transport protocols, ...
- Support application level capabilities to specify dependencies at a logical level rather than specific file level
  - First class support for Affinities (D-C, D-D)
- Typically placement and scheduling of data is decoupled from the compute-tasks
  - Integrated approach to compute and data?
- Dynamic decision for data
  - Analogous to late-binding of data
  - Fluctuating resources as a fundamental property of DCI
- Abstraction for other factors and not application specific way:
  - Varying data sources, fluctuating data rates, etc
Pilot-Data: Abstraction for Dynamic Distributed Data

In analogy with BigJob - BigData (before Big Data was BigData!)
Pilot-Data: Coupled Data Management and Efficient Transfer
Overview

• Understanding the Landscape of Distributed CI and Applications
  – Introduce IDEAS as design objectives
• SAGA: A Standards-based Approach to DCI and Applications
  – Layer upon which other abstractions are built
• P* Model of Pilot-Abstractions
  – Uniform abstraction for compute and data
  – Interoperability and Extensibility

• P* Implementations are Supporting Large-scale S&E
  – Map-Reduce based next-generation gene sequencing data
  – Structural biology and patient-specific medicine
  – Generalized ensemble and advanced sampling algorithms
Pilot-MapReduce (for NGS)
Pilot-MapReduce

Stage 1: Map

Stage 2: Shuffle

Stage 3: Reduce
Distributed (DMR) versus Hierarchical (HMR)
DMR vs HMR vs Iterative MR?
Depends upon the Application characteristics

<table>
<thead>
<tr>
<th>Application</th>
<th>Input</th>
<th>Intermediate</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>GS/PMR</td>
<td>80 GB</td>
<td>71 GB</td>
<td>17 GB</td>
</tr>
<tr>
<td>Word Count (English)</td>
<td>16 GB</td>
<td>26 GB</td>
<td>20 MB</td>
</tr>
<tr>
<td>Word Count (random)</td>
<td>16 GB</td>
<td>30 GB</td>
<td>30 GB</td>
</tr>
</tbody>
</table>
Hadoop vs PMR (for distributed data scenarios)

Runtime in sec

- Hadoop
- Hierarchical Hadoop
- Distributed PMR
- Hierarchical PMR

Natural language

Random
Hadoop vs PMR (for distributed data scenarios)

Run Time in sec

Read Size in GB

Hierarchical MR
Distributed PMR
# PMR versus other MapReduce Implementations

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Key Parallelism</strong></td>
<td>Pilot-based Distributed MR</td>
<td>Hadoop-based/ MR (Pydoop)</td>
<td>Hadoop Streaming</td>
<td>Hadoop-based MR</td>
<td>MR-based Structured Programming Framework</td>
</tr>
<tr>
<td><strong>Strategy</strong></td>
<td>No</td>
<td>Yes</td>
<td>Yes[^1]</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><strong>Hadoop Requirement</strong></td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><strong>Multiple Cluster</strong></td>
<td>Yes</td>
<td>Limited by Hadoop</td>
<td>Limited by Hadoop</td>
<td>Limited by Hadoop</td>
<td>Limited by JVM</td>
</tr>
<tr>
<td><strong>Support</strong></td>
<td>Support</td>
<td>Not allowed by Hadoop</td>
<td>Not allowed by Hadoop</td>
<td>Not allowed by Hadoop</td>
<td>Not Easy</td>
</tr>
<tr>
<td><strong>Multiple Node</strong></td>
<td>Yes</td>
<td>Limited by Hadoop</td>
<td>Limited by Hadoop</td>
<td>Limited by Hadoop</td>
<td>Limited by JVM</td>
</tr>
<tr>
<td><strong>Task Support</strong></td>
<td>Yes</td>
<td>Limited by Hadoop</td>
<td>Limited by Hadoop</td>
<td>Limited by Hadoop</td>
<td>Limited by JVM</td>
</tr>
<tr>
<td><strong>Distributed Job</strong></td>
<td>Yes</td>
<td>Limited by Hadoop</td>
<td>Limited by Hadoop</td>
<td>Limited by Hadoop</td>
<td>Limited by JVM</td>
</tr>
<tr>
<td><strong>Data Coordination</strong></td>
<td>Yes</td>
<td>Limited by Hadoop</td>
<td>Limited by Hadoop</td>
<td>Limited by Hadoop</td>
<td>Limited by JVM</td>
</tr>
<tr>
<td><strong>Primary Aligner</strong></td>
<td>BWA, Bowtie, and Others (coming)</td>
<td>BWA</td>
<td>Bowtie</td>
<td>RMAP</td>
<td>BWA</td>
</tr>
<tr>
<td><strong>Multiple Aligner</strong></td>
<td>Straightforward</td>
<td>Not Straight-forward</td>
<td>Possible</td>
<td>Not Straight-forward</td>
<td>Straight-forward</td>
</tr>
<tr>
<td><strong>Support</strong></td>
<td>Straightforward</td>
<td>Not Straight-forward</td>
<td>Possible</td>
<td>Not Straight-forward</td>
<td>Straight-forward</td>
</tr>
<tr>
<td><strong>Primary Tasks</strong></td>
<td>Alignment/Duplicate Removal (and Extensible for RNA-Seq)</td>
<td>Alignment/ Duplicate Removal</td>
<td>Alignment/ SNP Discovery</td>
<td>Alignment</td>
<td>Various NGS Data &amp; Downstream Analysis</td>
</tr>
<tr>
<td><strong>Extensibility</strong></td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td><strong>for Supporting</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Multiple Tools</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>(e.g. Multiple Aligners)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
PMR Scalability

![Graph showing PMR Scalability with different read sizes (2, 4, 8 GB) and tasks: Reduce, Shuffle, Map, Setup. The graph illustrates the runtime in seconds for each task under varying conditions.]
PMR Extensibility

- SEQAL
- Local PMR
- Distributed PMR

Runtime (in sec)

Read Size (in GB)

- 2
- 4
- 8

Run time (in sec)

Read size (in GB)

- Crossbow(Bowtie)
- local PMR(Bowtie)
- distributed PMR(Bowtie)
Deployment Context: Limitations of Abstractions
SAGA-based PJ: HT-HPC
Redux: P* Model of Pilot-Abstractions

- Applications
- Tools
- Pilot-MapReduce
- Pilot-API
- P*
  - Pilot-Jobs
  - Pilot-Data
- SAGA
- Resource (Network, Compute & Data) Access Layer
- Physical Infrastructure
  - e.g. XSEDE, EGI, OSG, ESNet, Clouds
Acknowledgments

• SAGA and RADICAL Group Members
  – http://saga-project.org
• NSF/LEQSF (2007-10)-CyberRII-01
• NSF HPCOPS NSF- OCI 0710874 award
• NSF CHE 1125332
• NSF-ExTENCI (OCI-1007115)
• UK EPSRC (GR/D0766171/1) and e-Science Institute, UK
• NSF OCI 1059635
• NIH Grant Number P20RR016456
• NSF TeraGrid TRAC award TG-MCB090174
• NSF FutureGrid Award (No. 42)
Some of the People Who Make it Happen

- Andre Merzky
- Ole Weidner
- Andre Luckow
- Mark Santcroos
- Ashley Zebrowski
- Melissa Romanus
- Pradeep Mantha
- Hugh Martin (PhD 2010)
- Sharath Maddineni
- Nayong Kim
- Abhinav Thota
- Joohyun Kim
- Yaakoub el-Khamra

Collaborators:
- Peter Coveney
- Jon Weissman
- Dan Katz
- M Parashar
- G Allen
- T Bishop
- C Laughton
- R Levy
- Darrin York
- ..