From Web Services Toward Grid Services

Building Grid Computing Applications

Eric Yen
Computing Centre, Academia Sinica
Outline

• Objective and Introduction
• GT3 for Grid Services
• Grid Services Development
• Service Discovery
• Related Work in the World
• Summary
Objectives

• Find out details of issues for designing and building Grid Services, such as
  – design methodologies and principles,
  – tooling,
  – application requirements,
  – benchmarking,
  – reuse,
  – debugging,
  – service orchestration,
  – scalability, and
  – interoperability.

• Find out what makes for a good Grid service (or a bad service),
• Learn how to organize service data, approaches to factoring problems into services, views on service lifetime and granularity, experiences with building Grid services,
• Think about how to teach Grid service design.
Now we know the Grid, how to take advantage of it
Deployment of Grid Technology

• Application Drive
• Realizing the Grid vision requires major enhancements to our scientific infrastructure in three key areas: high-speed networks, Grid services, and Grid-enabled terascale facilities.
Developing Grid Applications

• **Resources, Services, Sites**

• Major Grid Building “activities”, (by Rick Stevens)
  - Grid software infrastructure and toolkit development
  - Grid hardware resources [systems, networks, data, instruments]
  - Grid applications development and deployment
  - Grid resource allocation and policy development
Grid Services

• Grid services are the glue that transforms a collection of distributed, independent resources into a coherent computing, storage, and collaboration fabric.

• Providing uniform mechanisms for user authentication and authorization, accounting, resource access, data transfer, system monitoring, and resource management, Grid services make it possible for users, applications, and tools to discover and use disparate resources in coordinated ways.

• The emergence of the Globus Toolkit as a de facto standard for Grid services makes it feasible for a Grid system to provide coherent internal services and interoperate with other Grid systems.
Grid Service
Grid Service

• Definition
  – Any service offered to clients in a Grid environment
  – A service that is compliant with the OGSI spec, and which exposes itself through a WSDL interface.
  – A standard Web service adapted to the requirements typically found in a Grid environment.

• OGSI
  – A set of service primitives are specified, rather than a full-fledged services
  – Ensures that a nucleus of behavior in common to all Grid services that can be leveraged by meta- and system level infrastructure services.
  – Useful to dynamically cope with changing runtime properties of the network.
Web Service

• Address the discovery and invocation of persistent services
• Offers a powerful application-to-application integration framework
• What’s missing
  – State Management
  – Global Service Naming
  – Reference Resolution
  – Common behaviors and semantics defined for services by some interface description language
### Common Services is Necessary

<table>
<thead>
<tr>
<th>Absolutely Minimal Set</th>
<th>Information Grid Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Simple Registry</td>
<td>- Distributed Query Processing</td>
</tr>
<tr>
<td>- GridFTP/Reliable File Transfer</td>
<td>- Data Cache</td>
</tr>
<tr>
<td>- Job Submission</td>
<td>- Data Replication</td>
</tr>
<tr>
<td>- Data Access</td>
<td>- Mobile Code</td>
</tr>
<tr>
<td>- Authentication</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Key Services</th>
<th>Compute/File Grid Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Authorization Services</td>
<td>- Scheduler</td>
</tr>
<tr>
<td>- Notification</td>
<td>- Dynamic deployment of APs</td>
</tr>
<tr>
<td>- Workflow</td>
<td></td>
</tr>
<tr>
<td>- Dependency Management</td>
<td></td>
</tr>
<tr>
<td>- Registry</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Portal Services</th>
<th>Data Mining</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Session Management</td>
<td>- Visualization</td>
</tr>
<tr>
<td>- State Management</td>
<td></td>
</tr>
<tr>
<td>- Dynamic Deployment of Services</td>
<td></td>
</tr>
<tr>
<td>- Reservation</td>
<td></td>
</tr>
<tr>
<td>- Transaction</td>
<td></td>
</tr>
<tr>
<td>- Dynamic Deployment of Services</td>
<td></td>
</tr>
</tbody>
</table>
GT3 for Grid Services
GT3 Core Architecture
Overall Approach

- Create a simple testbed with the GT3 toolkit.
  - First release was June 30th
- Create some new simple services.
  - “learn by doing”
- Demonstrate the results and measure performance.
- Start to work with the AliEn components to understand them.
  - Can we envisage one framework and “competitive” services?
- Report on the activity
- Plan the future work lists

Courtesy of “OGSA/GT3 Evaluation Status Report”, by D. Foster et. al, CERN
GT3 Testbed

 Courtesy of “OGSA/GT3 Evaluation Status Report”, by D. Foster et. al, CERN
GT3 TestBed

- Simple system to distribute jobs and retrieve output
  - No security (for most services)
  - The user asks the Resource Broker (RB) to select the best Computing Element (CE)
  - The user submits the job to the CE
  - The Information and the Logging & Bookkeeping services exchange information mainly with the RB

- Why did we do it this way?
  - Simple scheme
    - As already mentioned: no architectural ambitions…
    - Learn by doing!

- What did we learn out of it?
  - See next slides
GT3 TestBed

- Resource broker and L&B (**Custom service**)
  - Surprisingly fast to set-up
- A few computing elements (**GT3-GRAM, with modifications**)
  - 2 PC boxes in the CERN Computing Centre
  - In a second phase, one PC located in Moscow was added
  - Some problems (solved) in data stage-in/stage-out
  - See GRAM comments in the performance part
- Information service (**GT3-IS**)
  - Native GT3 service
  - In this TestBed talks only with other services

Courtesy of “OGSA/GT3 Evaluation Status Report”, by D. Foster et. al, CERN
### GT3 TestBed coverage

<table>
<thead>
<tr>
<th>PortType</th>
<th>Operation</th>
<th>OSGA/GT3 evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gridservice</td>
<td>FindServiceData</td>
<td>GT3 TestBed-RB uses it to retrieve data from IS IS performance tests (C-client)</td>
</tr>
<tr>
<td></td>
<td>SetTerminationTime</td>
<td>Not Used Yet (directly)</td>
</tr>
<tr>
<td></td>
<td>Destroy</td>
<td>Everywhere, e.g. GRAM</td>
</tr>
<tr>
<td>NotificationSource</td>
<td>SubscribeToNotificationTopic</td>
<td>IS perf. Tests (data sources)</td>
</tr>
<tr>
<td>NotificationSink</td>
<td>DeliveryNotification</td>
<td>IS perf. Tests (listener)</td>
</tr>
<tr>
<td>Registry</td>
<td>RegisterService</td>
<td>Code examples</td>
</tr>
<tr>
<td></td>
<td>UnRegisterService</td>
<td>Code examples</td>
</tr>
<tr>
<td>Factory</td>
<td>CreateService</td>
<td>Via GRAM (first tests) Specific tests using DummyService</td>
</tr>
<tr>
<td>HandleMap</td>
<td>FindByHandle</td>
<td>Not Used Yet</td>
</tr>
</tbody>
</table>

Every service must implement this PortType

"pull" data access

"push" data access

Courtesy of “OGSA/GT3 Evaluation Status Report”, by D. Foster et. al, CERN
Summary

• GT3 is the first OGSI 1.0 implementation
  • Main focus of all activity so far
  • GT3 (ToolKit + doc) is in a status that allow a quick start
    - Not everything is perfect, but GT3 is more mature than expected
  • Development experience and quantitative measurements and in the next section of the presentation

• GT3 provides a few OGSA services by now
  • GRAM and RLS (GT2)
  • IS (Information Service)
  • RFT (Reliable File Transfer; GridFTP based)

• GT3 encourages to create custom services
  • The OGSI system provides the building blocks to provide a variety of services

Courtesy of “OGSA/GT3 Evaluation Status Report”, by D. Foster et. al, CERN
Grid Service Development
What we get:

- From Web Services
  - Interoperability
    - standard for message creation and definition - XML
    - standard for protocol-independent message passing – SOAP
    - standard for service definition – WSDL
  - result: choice on hosting environment is left to the service provider
  - Service Oriented Design approach

- From OGSI
  - Stateful Services (Service Data)
  - Other common features on independent services
    - Different from GT2 where nothing is common between services apart from GSI
    - Straightforward development: common framework for service usage and management
Grid Service Development

• What we get:
  - From the Globus Toolkit 3
    • Security Infrastructure
      - Authentication, authorization, delegation, message integrity and encryption
    • Higher-Level Services
      - Information Services – Index Service
      - Data Management – RLS and RFT
      - Master Managed Job Factory – GT3 interface for GRAM

• In summary:
  • Interoperable and environment independent services

Courtesy of “OGSA/GT3 Evaluation Status Report”, by D. Foster et. al, CERN
Grid Service Development

- **Current options:**
  - **Hosting Environments:**
    - J2EE Application Server – Jakarta Tomcat, GT3 Standalone Container, Websphere, …
    - Microsoft .NET Platform
  - **OGSI implementations:**
    - J2EE Servers: Globus Toolkit 3
    - Microsoft .NET: OGSI.NET (Virginia Univ.); MS.NETGrid (EPCC)
  - **Others are appearing**
    - Any environment with an existing implementation of a Web Services engine is one single step away from providing Grid Services
    - Ex: OGSI::Lite (Perl), pyGridWare (Python)

Courtesy of “OGSA/GT3 Evaluation Status Report”, by D. Foster et al, CERN
Designing Grid Services

• Important concepts when designing Grid Services:
  - Factories and Instances

  ![Diagram]

  1. Client
  2. Factory

• Factories create instances and respond to instance creation requests by clients
• Instances respond to client’s service specific interaction requests
• Advantages:
  – Workload balancing between pools of instances
  – User dependent instances
• Disadvantages:
  – Instance creation overhead
Designing Grid Services

• Approach:
  - Service Data, Subscriptions and Notifications

  GRID SERVICE A
  SDE A1  SDE A2

  GRID SERVICE B
  SDE B1

  1 - SUBSCRIPTION

  2... - NOTIFICATIONS

• Each Grid Service has it’s own Service Data Set - collection of Service Data Elements (SDEs)
• Every SDE has a set of associated values concerning its validity in time – goodFrom, goodUntil, availableUntil
• A service or client may declare interest in a SDE by issuing a Subscription
• Service Data flows by means of Notifications – normally when a change occurs or the value lifetime has expired
Writing Grid Services in GT3

You need:

1. A service interface – GWSDL (WSDL extended)
   • manually written or generated from existing Java code
2. The service implementation
   • directly extending a basic Grid Service or using Operation Providers (delegation) – in Java
3. A deployment descriptor
   • defined using WSDD (Web Service Deployment Descriptor)
4. A build file
   • For use by the Jakarta Ant build tool

• RESULT: A JAR file to use for deployment (GAR)
Using Grid Services

- Grid Services in action

HOSTING ENVIRONMENT

GRID CONTAINER

SERVICE IMPLEMENTATION

WSDL DESCRIPTION

STUBS

APPLICATION

CLIENT
Globus Toolkit 3 Overview

- The Globus Toolkit 3 is a complete implementation of the OGSI specification
- The development process is much easier when compared with previous versions of the toolkit
- Some additional components to what is in OGSI proved essential to achieve this:
  - Security Infrastructure:
    - GSI3 is an easy to use security provider, abstracting the developer from the major issues it deals with
  - Deployment Tools:
    - By using Ant and providing sample build files for service deployment, the developer can focus most of his time on the implementation of the service features
- Backward compatibility:
  - All GT2 components are shipped with the GT3 full bundle
  - Some services remain usable – those where only an OGSI-compliant interface was provided (e.g. GRAM)
  - Others are completely independent implementations (e.g. MDS2 and MDS3)
- A large user community is being built
Globus Toolkit 3 Overview

- Steep learning curve - it represents a new approach to service design and implementation (many small details that take time)
- Incomplete documentation – this is a real problem being faced by developers at this time
- Several bugs found in these exercises
  - Core implementation related - due to framework short lifetime
  - From tools deployed with the framework – hard to solve (e.g. Axis)
  - From the outside – easy to solve (e.g. Tomcat)
- Resource Management services still based on GRAM – with an OGSI-compliant but complex architecture behind

- Good resources for documentation and good interaction for problem solving:
  - OGSI 1.0 Specification
  - GT3 Tutorial http://www.casa-sotomayor.net/gt3-tutorial/
  - Globus Discuss discuss@globus.org
  - Globus Bugzilla
Service Discovery
Service Discovery Problem

Service Discovery Survey showed the two problems Service Discovery System face:
- Lack of description languages
- Lack of inexact matches

Solution:
- XML as a descriptive language
- Usage of Ontologies
Grid for Digital Archive

• Requirements
  – Facilitate the semantic-level integration of distributed and disparate archives
    • --> Resource Registry with Complete Metadata
    • --> Organization of Information Flow
    • --> Content Mediation
  – Accurate Search
    • More Accurate Search
    • More easy and precise way to describe the target
    • --> Thesaurus --> Ontology Building
    • --> User Interface & Indexing
    • --> Semantic Grid
  – Storage Management
    • --> Data Grid
Grid for Digital Archive

• Impacts
  – Overcome the barriers in different levels of IT APs
    • lack of interdisciplinary communication
    • isolated solutions
    • local differences in the organization of digital archives
Service Ontology – DAML-S

- **Resource** provides **Service**
- **Service Profile** presents **Service Model** and **Service Grounding**
  - **Service Model** describedBy **Service Grounding**
  - **Service Grounding** supports **Service Model**
- **Service Profile**
  - description
  - functional attributes
  - functionalities
- **What it does**
- **How it works**
- **How to access it**
Service Discovery Interactions

SD Matchmaker

Service Requester

Grid Application

Service Provider

Grid Service

Query

Publish

Invoke
Service Discovery Framework

1. Service Provider
2. Service Requester
3. Context Matching
4. Syntactic Matching
5. Semantic Matching
6. Service Registry DB
7. Auxiliary DB
8. Service Ontology DB
9. Advertisement
10. Word Distance, Type Hierarchy
11. Ontology Reasoner
12. Concept
13. Task Ontology
14. Domain Ontology

Parsing

[Diagram of service discovery process with boxes and arrows indicating the flow of information and interactions between service requesters, providers, and various matching and reasoning components.]
Solution Examples
## TeraGrid Service Architecture

<table>
<thead>
<tr>
<th>Service Layer</th>
<th>Functionality</th>
<th>TeraGrid Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced Grid Services</td>
<td>super schedulers, resource discovery services, repositories, etc.</td>
<td>SRB, MPICH-G2, distributed accounting, etc.</td>
</tr>
<tr>
<td>Core Grid Services (Collective layer)</td>
<td>TeraGrid information service, advanced data movement, job scheduling, monitoring</td>
<td>GASS, MDS, Condor-G, NWS</td>
</tr>
<tr>
<td>Basic Grid Services (Resource layer)</td>
<td>Authentication and access Resource allocation/Mgmt Data access/Mgmt Resource Information Service Accounting</td>
<td>GSI-SSH, GRAM, Condor, GridFTP, GRIS</td>
</tr>
</tbody>
</table>
## TeraGrid Application Service Example

<table>
<thead>
<tr>
<th>Service</th>
<th>Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Batch Runtime</td>
<td>Supports running static-linked binaries</td>
</tr>
<tr>
<td>Advanced Batch Runtime</td>
<td>Supports running dynamic-linked binaries</td>
</tr>
<tr>
<td>Scripted Batch Runtime</td>
<td>Supports scripting (including compile)</td>
</tr>
<tr>
<td>On-Demand / Interactive Runtime</td>
<td>Supports interactive applications</td>
</tr>
<tr>
<td>Large-Data</td>
<td>Supports very large data sets, data pre-staging, etc.</td>
</tr>
<tr>
<td>File-Based Archive</td>
<td>Supports GridFTP interface to data services</td>
</tr>
</tbody>
</table>

### Specification

<table>
<thead>
<tr>
<th>Basic Batch Runtime</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Objective</strong></td>
<td>To support execution of compiled binaries</td>
</tr>
<tr>
<td><strong>Assumptions / Policy</strong></td>
<td>• 1 user per node, dedicated during runtime</td>
</tr>
<tr>
<td></td>
<td>• nodes accessible for login (minimal environment)</td>
</tr>
<tr>
<td></td>
<td>• static linked binaries</td>
</tr>
<tr>
<td></td>
<td>• attributes and values published in MDS (using XML schema currently being developed)</td>
</tr>
<tr>
<td>System Software</td>
<td>Query for values (e.g. OS release, kernel level)</td>
</tr>
<tr>
<td>Node Hardware</td>
<td>Query for values (e.g. IA64 nodes, n CPU/node)</td>
</tr>
<tr>
<td>Software and Libraries</td>
<td>MPICH-G (min version)</td>
</tr>
<tr>
<td>I/O</td>
<td>• Local space: $TG_Local</td>
</tr>
<tr>
<td></td>
<td>• Home space: $TG_Home</td>
</tr>
<tr>
<td></td>
<td>• Global (shared) space: $TG_Global</td>
</tr>
<tr>
<td></td>
<td>• Node communication (protocol(s), naming)</td>
</tr>
<tr>
<td></td>
<td>• Note in/out streaming (to other notes within cluster, to other clusters, outside clusters)</td>
</tr>
<tr>
<td></td>
<td>• GridFTP</td>
</tr>
<tr>
<td>System Commands &amp; Utilities</td>
<td>• Path $TG_UTIL</td>
</tr>
<tr>
<td></td>
<td>• Min set (e.g. tar, gzip, gsi-scp, gsi-ssh, etc.)</td>
</tr>
</tbody>
</table>
PPDG Analysis Service Model
ICENI: An Open Grid Service Architecture Implemented with Jini

Web Service Client

ICENI Service

SOAP

RMI

Jini

Jini Service

Jini Lookup

Web Services

SOAP

Web Service Registry
GSP within PROGRESS

Content Provider

Webservice

Session Bean

Entity Beans

Data Management System (DMS)
Summary & Conclusion
Conclusions

• GT3 is the natural partner for new middleware initiatives
  - Encouraging results so far
  - Requires experience of large scale deployment
• OGSI seems to be already the lingua franca in this field
• Grid Application Service Provider should provide attractive high level services
  - HEP-specific services will be missing
  - A convincing backbone of services should materialise (also with HEP contribution)
  - OGSA/OGSI concept validate when serious challengers will be deployed on large scale
Conclusions

• A best practice guideline for Grid Services should be developed
• Need for Standard Data Definition, and “Grid object” definition
• Need for debugging tools appropriate for the Grid environment
• Need for common metadata catalog schema
• Matchmaking services
• User Interface Portal