The Physiology of the Grid
OGSA for distributed system integration

Foster, Kesselman, Nick, and Tuecke
Presented by Beomseok Nam
CMSC818S: Parallel and Distributed Data Intensive Computing

Overview
- Grid
- Globus Toolkit
- Web services
- Open Grid Service Architecture
- Grid service interfaces
- Conclusion

The Grid problem
- "The Grid problem" is about enabling geographically distributed communities (Virtual Organizations) to share resources as they pursue common goals - E-science and E-business.
- It is assumed the absence of...
  - Central location
  - Central control
  - Omniscience
  - Existing trust relationships

Why Grid? (e-Science)
- A biochemist exploits 10,000 computers to screen 100,000 compounds in an hour
- 1,000 physicists worldwide pool resources for peta-op analyses of petabytes of data
- Civil engineers collaborate to design, execute, & analyze shake table experiments
- Climate scientists visualize, annotate, & analyze terabyte simulation datasets
- An emergency response team couples real time data, weather model, population data

Why Grid? (e-Business)
- Engineers at a multinational company collaborate on the design of a new product
- A multidisciplinary analysis in aerospace couples code and data in four companies
- An insurance company mines data from partner hospitals for fraud detection
- An application service provider offloads excess load to a compute cycle provider
- An enterprise configures internal & external resources to support eBusiness workload

Definition of a Grid architecture
- Grid has been defined as
  - A collection of distributed services [Globus]
  - Distributed operating systems [Legion]
  - Distributed resource management frameworks [Condor]
- OGSA
  - Open Grid Services Architecture
  - Grid: an extensible set of Grid services
  - An extension and a refinement of the emerging Web Services architecture
  - Incorporation of Grid functionality into a Web services framework
  - Two technologies: Globus Toolkit & Web services
**Globus Toolkit**

- Tools enabling resource sharing within VOs
  - GRAM (Grid Resource Allocation and Management)
  - Tool for remote job and resource management
  - GSI (Grid Security Infrastructure)
  - Authentication based on grid-wide credential
  - Single sign-on, delegation
  - Authorization
  - MDS (Meta Directory Service)
  - Grid-wide information on the state of resources
  - Data Grid Technologies
    - GridFTP
    - Replica Management
  - Virtual data

**Globus Toolkit (2)**

- These components provide the essential elements of a service-oriented architecture in OGS

- Globus Toolkit services

**Refactoring Globus Toolkit**

- Globus Toolkit has no standard means of invocation, notification, error propagation, authorization, termination
- Virtualization of end systems is missing → Refactoring

**Web services framework**

- What is Web services?
  - Definition: A Web service is a software system identified by a URI, whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML-based messages conveyed by internet protocols. (W3C)
  - Two advantages
    - Support dynamic discovery and composition of services in heterogeneous environments
    - A framework based on Web services can exploit numerous tools and extant services
- Web services standards
  - XML, WSDL, WSIL, SOAP, UDDI, etc.

**Web Services architecture**

**WS example: Google Web API**

- Google Web APIs service work:
  - Software developers can
    - query more than 3 billion web documents directly from their own programs.
    - receive results as structured data, access information in the Google cache, and check the spelling of words.
  - Google uses the SOAP and WSDL standards so a developer can program in his or her favorite environment
- http://www.google.com/apis/
Web Services standards

- **WSDL**
  - Web Services Description Language
  - XML schema to describe a Web service interface.
- **UDDI**
  - Universal Description, Discovery and Integration
  - UDDI registry is a place to publish WSDL documents
  - Directory for Web services
- **SOAP**
  - Simple Object Access Protocol
  - XML based RPC protocol
  - Independent of the underlying transfer protocol

Web Services standards(2)

- **WS-Inspection**
  - XML language and conventions for locating Web service descriptions
  - Service description
    - Usually a URL to WSDL document
    - Occasionally reference to an entry in UDDI
  - **WSIL**: WS-Inspection Language
- **WSFL**
  - Web Services Flow Language
  - Web services orchestration
    - Building of sophisticated Web services by composing simple Web services

Example: WS description

```
<types>
  <schema targetNamespace="http://samples.ogsa.globus.org/database/database.xsd"
           xmlns="http://www.w3.org/2001/XMLSchema">
    <complexType name="query">
      <sequence>
        <element name="send_query" type="string"/>
      </sequence>
    </complexType>
  </schema>
</types>

<message name="myDatabaseQuery">
  <part name="query_parameter" type="xsd1:query"/>
</message>

<message name="myDatabaseResponse">
  <part name="response_parameter" type="string"/>
</message>

<portType name="Database_PortType">
  <operation name="databaseQueryOperation">
    <input message="tns:myDatabaseQuery"/>
    <output message="tns:myDatabaseResponse"/>
  </operation>
</portType>
```

Example: WS implementation

```
<binding name="Database_Binding" type="tns:Database_PortType">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="databaseQueryOperation">
    <input>
      <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded" namespace="http://samples.ogsa.globus.org/database"/>
    </input>
    <output>
      <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded" namespace="http://samples.ogsa.globus.org/database"/>
    </output>
  </operation>
</binding>

<service name="Database_Service">
  <port name="Database_Port" binding="tns:Database_Binding">
    <soap:address location="http://ept.mcs.anl.edu:8080/axis/services/Database_Port"/>
  </port>
</service>
```

Open Grid Services Architecture

- **Service oriented architecture**
  - Everything is a service with well-defined interfaces
- **Virtualization**
  - Encapsulation behind a common interface of diverse implementations
  - Allows the composition of lower-level services to form more sophisticated services

---

**WS example : Database Service**

- WSDL definition for "Database_PortType" defines operations and bindings, e.g.:
  - QueryOperation(Query, Response)
  - Accessible over SOAP

- PortType is a set of operations

---

**Example: Database Service**

```
Database_PortType
```

---

**Example: WS description**

```
Type
  <complexType name="query">
    <sequence>
      <element name="send_query" type="string"/>
    </sequence>
  </complexType>

Input/Output Message
  <message name="myDatabaseQuery">
    <part name="query_parameter" type="xsd1:query"/>
  </message>

  <message name="myDatabaseResponse">
    <part name="response_parameter" type="string"/>
  </message>

  <operation name="databaseQueryOperation">
    <input message="tns:myDatabaseQuery"/>
    <output message="tns:myDatabaseResponse"/>
  </operation>
```

---

**Example: WS implementation**

```
use SOAP
  interpreted as RPC call
    use http for transport
    Network Protocol Binding
      Service
        Service name: "Database_Service";
          <port name="Database_Port" Binding="tns:Database_Binding">
            <soap:address location="http://ept.mcs.anl.edu:8080/axis/services/Database_Port"/>
          </port>
```
Web services as Grid services

- **Grid Service**: A Web service that follows the following conventions
  - OGSA specifies three things that a web service must have before it qualifies as a Grid Services
    1. must be an instance of a service implementation of some service type
    2. must have a Grid Services Handle (GSH), which is bound to Grid Services Reference (GSR).
    3. must implement a portType called "GridService" which has three operations: FindServiceData, Destroy, SetTerminationTime.

GSH/GSR and HandleMap

- **GSH (Grid Service Handle)**
  - Uniquely identifies a service instance
  - Basically just a URL
  - e.g., http://server.ggf.org:2345/gridservices/example/instance
- **GSR (Grid Service Reference)**
  - Contains all the information a client needs in order to communicate with a service
  - GSH must be mapped to GSR in order to use a service
  - http GET operation on GSH returns a GSR
- **HandleMap** interface allows a client to map from a GSH to a GSR

Transient service instances

- Web services address discovery and invocation of persistent services
- In Grid, must also support transient service instances, created/destroyed dynamically
- The interfaces of OGSA address Globus Toolkit functionalities
- Discovery
- Dynamic service creation
- Lifetime management
- Notification
- Manageability

Grid service interfaces

- **GridService**
  - FindServiceData / Destroy / SetServiceTime
- **Factory**
  - CreateService
- **NotificationSource & NotificationSink**
  - SubscribeToNotificationTopic / DeliverNotification
- **Registry**
  - RegisterService / UnregisterService
- **HandleMap** (= Mapper, HandleResolver)
  - FindByHandle

Example: GS description

```xml
<portType name="GridServicePortType">
  <operation name="findServiceData">
    <input message="tns:FindServiceDataInputMessage"/>
    <output message="tns:FindServiceDataOutputMessage"/>
    <fault name="QueryNotSupportedFault" message="ogsa-faults:QueryNotSupportedFault"/>
  </operation>
  <operation name="setTerminationTime">
  </operation>
  <operation name="destroy">
  </operation>
</portType>

<portType name="Database_PortType" extends="gsdl:GridService">
  <operation name="databaseQueryOperation">
    <input message="tns:myDatabaseQuery"/>
    <output message="tns:myDatabaseResponse"/>
  </operation>
</portType>
```

Grid Service example (2)

- **GridService** (required)
- **Database** other interfaces (optional)
- **Notification** Service creation Service registry

Implementation

Hosting environment/runtime
- "C", J2EE, .NET, Python...
Service data

- A Grid service instance may maintain a set of service information
  - Service data
    - A set of service data elements
    - <name, type, TTL> meta-data, state-data </name>
  - FindServiceData operation within GridService interface retrieves service data from Grid service instances.
  - Clients query the Grid service instance for this information

Registry & Factory

- Registry interface may be used to discover a set of Grid service instances
  - Returns a WS-Inspection document containing the GSHs of a set of Grid services
  - Returns policy associated with the set
  - Also available through Discovery interface
- Factory creates a new Grid service instance
  - Dynamic & reliable creation with Create operation
  - Returns a GSH

Lifetime management

- GS instances are created by Factory or manually, and destroyed via soft-state or explicitly
  - Negotiation of initial lifetime with Factory
  - Requesting a lifetime extension is allowed
  - Resource consumption is bounded in hosting environment
- SoftStateDestruction interface supports
  - GetTerminationTime for inquiry
  - SetTerminationTime for keepalive
  - Explicit destruction is also available
  - Destroy operation in GridService

Notification

- Messaging service
  - Direct service-to-service notification message delivery
  - If a particular service wishes to support subscription of notification messages, it must support NotificationSource interface
  - A service that wishes to receive notification messages must implement NotificationSink interface
  - Close integration with service data
    - "push" delivery of service data that meet specified condition (FindServiceData op : pull model)
  - A wide variety of uses are possible
    - E.g. dynamic discovery/registry services, monitoring, application error notification, ...

Notification example

- "push" delivery of service data
- Any service data can be associated with notification

Notification example

- "push" delivery of service data
- Any service data can be associated with notification
Notification example

- "push" delivery of service data
- Any service data can be associated with notification

 transient service instance

- What services can you create?
- What database services exist?

Application Example: Data Mining for Bioinformatics

- Find me a data mining service, and somewhere to store data

Application Example: Data Mining for Bioinformatics

- I want to create a personal database containing data on e.coli metabolism
Application Example: Data Mining for Bioinformatics

"Create a data mining service with initial lifetime 10"

"Create a database with initial lifetime 1000"

Database Service Provider

Database Factory

Mining Factory

Community Registry

User Application

Storage Service Provider

Database Service

Results

Query

Keepalive

Application Example: Data Mining for Bioinformatics

"Create a database with initial lifetime 1000"

Database Service Provider

Database Factory

Mining Factory

Community Registry

User Application

Storage Service Provider

Database Service

Results

Query

Keepalive
Application Example: Data Mining for Bioinformatics

Higher-Level Grid Services

- Distributed data management services
  - DB access, data translation, replica management, replica location, etc
- Workflow services
  - multiple application execution on multiple distributed grid resources
- Auditing services
  - Fraud and intrusion detection
- Instrumentation and monitoring services
  - Unusual condition detection
- Problem determination services
  - Dump, trace, and log mechanisms
- Security protocol mapping services
  - Map distributed security protocol onto native platform security service

Summary

- The Grid Problem
  - Resource sharing & coordinated problem solving in dynamic, multi-institutional virtual organizations
- Globus Toolkit
  - Tools enabling resource sharing within VOs
- Web Services
  - Dynamic discovery and composition of services
- Open Grid Services Architecture
  - Next step in evolution

Network protocol bindings

- Web services framework can be initiated on various protocol bindings
  - E.g. SOAP + HTTP with TLS
- Four requirements in selecting network protocol bindings
  - Reliable transport
  - Authentication and delegation
  - Ubiquity
  - GSR Format
- Can InterGrid protocol be defined and widely accepted?