The Open Grid Services Architecture

Summary of Grid 2, Chapter 17

Nick Rutar and David Wang
CMSC 818S

All figures in this presentation are taken from Grid 2 by Foster, Kesselman.
Roadmap

- Introduction of primary elements of the core service interfaces/behaviors
- Introduction of various OGSA elements
  - Web services technologies (WSDL)
  - OGSI, set of WSDL that supports basic Grid behaviors
- Detailed application study
What is a service?

- Provides capabilities to clients by exchanging messages.
- Identifies sequences of specific message exchanges to perform some operation.
- Since these operations are defined by only the sequence of message exchanges, the actual implementation of the services are flexible.
Service-Oriented Architecture

Service-oriented architecture

- All entities are services
- All operations visible to the architecture are the results of message exchanges
Service Examples

- **Storage service**
  - provide ops for storing and retrieving data, reserving space.

- **Data transfer service**
  - provide ops for requesting the transfer of data from one storage device to another.

- **Troubleshooting service**
  - monitor the status of various other services.

- **Have common behaviors as well**
  - Monitor status
  - Query and define access policy
OGSA Design Goals

- Operations are grouped to form interfaces, and interfaces are combined to specify a service.
  - Encourages code-reuse
  - Simplifies application design
- Ease of composition of services
- Service Virtualization: isolate users from details of service implementation and location.
Interaction with Services

- Usage of a standard interface definition language (IDL) such as WSDL (Web Services Description Language) to describe the services interfaces.
  - Defines ops supported by a service
  - Defines messages consumed and produced by such service.
  - Describes the messages the service expects, but not the service’s response to these messages.
    - Behavior is described in user doc.
Why IDL and Service Virtualization?

- Simplifies manipulation and management of services
  - Service discovery
    - allows clients to query and find suitable services in an unfamiliar environment.
  - Service composition
    - code-reusage, dynamic construction of complex systems from simple components.
  - Specialization
    - Use of different implementation of a service interface on different platforms.
  - Interface extension
    - Allows extensions to specialized service interfaces
OGSA Components

- Open Grid Services Infrastructure (OGSI)
- OGSA services
- OGSA schemas
- Built on Web services
  - Extended by OGSI to specify
    - How services are created
    - How long services live
    - How to manage faults
    - How to manage long-lived state, etc.

- A Web service that adheres to OGSI is called a Grid service.
Role of OGSA

OGSI does not define everything:
- How to establish identity and authenticate?
- How is policy expressed/negotiated?
- How do I discover services? …

OGSA needs to pick up the slack:
- Define additional services
- Define standard schema to achieve interoperability
Example

- Multiple storage services
- Multiple file transfer services
- Many auxiliary services (storage monitor service, storage broker service)
- Multiple clients

Figure 17.2 from Grid 2
Web Services

- Defined within W3C and its partners
- Supports dynamic discovery and composition in heterogeneous environments (WSDL).
- Provides standard mechanism for defining interfaces independent of their implementations.
- Widespread adoption
  - Allows exploitation of numerous tools and extant services
- Already defined standards useful for the Grid
  - Service registry
  - Security
  - Policy, etc.
Web Services Description Language (WSDL)

- Used by OGSA to describe software components independent of any programming language/implementation
- WSDL service definition is encoded using XML
  - Service description
    - defines the service interface
  - Implementation details
    - Describes how the interface maps to protocol messages and concrete endpoint addresses.
WSDL in Detail

Service interfaces \((\text{portType})\)

- Defined abstractly in terms of
  - Message structures
  - Sequences of message exchanges \((\text{Operations})\)

- Bound to concrete
  - Network protocol
  - Data-encoding format
  - Endpoint address
Example

```xml
<wSDL:definitions xmlns:tns="..." targetNamespace="...">
  <message name="get_file_request">
    <part name="term" type="xs:string"/>
  </message>
  <message name="get_file_response">
    <part name="value" type="xs:string"/>
  </message>
  <portType name="StorageServiceChange">
    <operation name="get_file">
      <input message="get_file_request"/>
      <output message="get_file_response"/>
    </operation>
  </portType>
</wSDL:definitions>
```

**<portType>:** defines service interface (class)

**<operation>:** defines messages exchanged (methods)

**<message>:** defines messages used (arguments)
Example (continued)

<types>: defines additional data types used by the service.

<binding>: used to specify protocols, data-encoding, transport.

Separation of portType and binding allows for multiple binding of the same service

- Remote/local file access using the same protocol have different binding for efficiency reasons
Open Grid Services Infrastructure (OGSI)

- Provides standard interfaces for OGSA
  - Grid Service, HandleResolver, NotificationSource, NotificationSubscription, NotificationSink, Factory, ServiceGroup, ServiceGroupRegistration, ServiceGroupEntry

- Features
  - Grid service description and instances
    - Distinguish between definition and instances
  - Service state, metadata, introspection
    - Allows clients to receive states of a particular service
  - Naming conventions/ resolution
  - Service life cycle management
  - Fault type
    - Standard base type for fault messages
  - Service groups
OGSI continued

- Provides a core set of operations that every OGSI compliant service must support
  - GridService
    - FindServiceData
    - setServiceData
    - requestTerminationAfter
    - requestTerminationBefore
    - destroy
Grid Service Descriptions and Instances

A service description consists of a WSDL that defines the Grid service’s interfaces and semantics.

A service instance consists of an addressable, potentially stateful, potentially transient instantiation of the description.
Service Data

- State data associated with a service
- OGSI defines standard in WSDL to allow explicit access to these states.
- Service data is associated with each interface.
- Each interface has can specify service data elements (SDE’s).
- Each SDE is a XML-typed element with properties allowing for modification.
- SDE’s can represent static and dynamic data about a service, and allows for introspection.
Accessing SDE’s

OGSI defines 2 mechanisms for accessing SDE’s

- Pull mode (query): clients can make complex queries/searches for SDE’s of a particular service.
- Push mode (subscription): clients get notified when a SDE change if the clients subscribe to that service.
- Pull mode is required, while as push mode is optional.
  - NotificationSource (services that support push mode)
  - NotificationSink (services that receives notifications)
Service Data Element Schema

- Monitoring and management functions requires standards for SDE’s
- Various schemas are being defined within OGSA.
Naming

- Need to distinguish instances from one another.
  - Grid service handles (GSH)
    - Long-term reference to a service (virtual)
    - Global and always refers to its corresponding GSR
  - Grid service reference (GSR)
    - Transient reference to a service (physical)
    - May migrate within the network
    - Network-wide pointers to a specific Grid service
  - Client uses GSH to find out where a particular service is, and uses handle resolution mechanisms to obtain the updated GSR to send requests.
Service Life Cycle

Creating Transient Services: Factories

- OGSI defines a standard, extensible Factory interface to accommodate many common and simple service instantiations.
- The Factory interface’s `createService` operation creates a requested Grid service and returns the corresponding GSH and initial GSR.
- Can be virtualized to accommodate various hosting environments
  - The hosting environment is responsible for creating new service instances and returning the GSH
Service Lifetime Management

- Negotiating initial lifetime
  - Client-specified
  - Can be infinity

- Requesting lifetime modification
  - Send periodic keep-alive messages
    - requestTerminationAfter
    - requestTerminationBefore

- Termination
  - Explicit
    - By client or its proxy
    - By the service itself
  - Lifetime expires without keep-alive message
Service Lifetime Management

Guarantees

- Client always knows when a service instance expects to terminate
- Client knows the service instance’s expiration time
- System faults will terminate the service and release all resources claimed by that service
OGSI fault model comprises a standard XSD type: ogsi:FaultType

- **Required elements**
  - Originating service
  - Timestamp

- **Optional elements**
  - Description
  - Underlying cause
  - Fault code for legacy support.
  - ...
Service Groups

- A service group is a Grid service that maintains information about a group of Grid service instances
- Can be grouped arbitrarily
- Interfaces that concern with organizing groups of service instances
  - ServiceGroup
  - ServiceGroupEntry
  - ServiceGroupRegistration
Service Groups

- **ServiceGroup**
  - Defines entry service data elements to represent members
  - Defines optional SDE membershipContentRule to describe member service prerequisite.

- **ServiceGroupEntry**
  - Provides independent lifetime management functions for individual service entries

- **ServiceGroupRegistration**
  - Add/remove entries from group
OGSA Services

Core Services
- Some variant of them should eventually be supported by any OGSA service

Data and Information Services
- Management of data and information

Resource and Service Management
Core Services

- Name resolution and discovery
  - Capable of resolving GSHs to GSRs
- Service Domains
  - Coordination of internal services
- Security
- Policy
- Messaging, queuing, and logging
- Events
- Metering and accounting
Data and Information Services

- Data naming and access
  - Naming and accessing data resources across relational & XML databases, file systems

- Replication
  - Tracking, discovering, and replicating data resources

- Metadata and provenance
  - Maintaining metadata
  - Describing and tracking how data is created
Resource and Service Management

- Provisioning and Resource Management
  - Negotiating SLAs (Service-Level agreements)

- Service orchestration
  - Describing and managing choreography of a set of interacting services

- Transactions
  - Assure all parties that transactions have executed correctly

- Administration and deployment
  - Software deployment, change & identity management
  - Deploying needed software and data into hosting environments
Case Study

Components of the file transfer network

Figure 17.10 from Grid 2
Case Study (continued)

SLA negotiation process

Figure 17.11 from Grid 2
Case Study (still going)

Establish delivery service instances

Figure 17.12 from Grid 2
Case Study (last one, promise)

- Monitoring infrastructure
  - Be notified if something goes wrong

Figure 17.13 from Grid 2
OGSA Implementations

Principles

- OGSA does not dictate a particular implementation architecture or “hosting environment”

Globus Toolkit Version 3

- Open source developed since 1996
- GT3 has complete implementation of OGSI
- Other OGSI implementations
  - pyGlobus package (Python)
  - UNICORE (Java)
Future Directions

Services and Tools
- Required to facilitate the integration of Grids into different application scenarios

Implementation
- High-performance protocols
- Lightweight service instance implementations

Semantics
- Need formal mechanism for individual services

Scalability
- Must be able to handle increasing # of entities