NDL-OWL and its implementation in GENI ORCA project

Yufeng Xin, Ilia Baldine RENCI/UNC-CH
Networked Clouds

Cloud Providers

Observatory

Petri dish

Wind tunnel
Application-driven resource orchestration

Application directly operates on resources

User-facing APIs

Application-specific resource mapping

Provisioning

Stitching

Resource Co-Scheduling

Substrate slivering APIs

Eucalyptus/OpenStack
Virtualized Compute

Xcat/PERCEUS

NSI

Network

Storage

Sherpa

OSCARS

OSCARS

Application Slice

Internet2 2010 SMM
ORCA is a “wrapper” for off-the-shelf cloud and circuit networks etc., enabling federated orchestration:

+ Resource brokering
+ VM image distribution
+ Topology embedding
+ Stitching
+ Authorization

• GENI, DOE, NSF SDCI+TC
  • http://networkedclouds.org
  • http://geni-orca.renci.org
ORCA Architecture

- ORCA Actors
  - Broker (CH)
  - Slice Manager (SM)
- Web portal
- Substrate owners
- Authority/AM
- Users and tools
- Java
- Web portal
- XML – RPC

key actions:
- request
- redeem
- delegate

operators
- Operators

web portal
Slicing of a network
Link slivering
Resource representation life-cycle

Application

Collective possibly filtered manifest

ORCA

Possibly fuzzy request

ORCA

Substrate provider

More specific request to substrate provider(s)

Substrate models from substrate providers

Detailed manifest from substrate provider

Networks, clouds, other resources
1. Ethernet connection: compA ----- EthernetSwitch -------compB
   can be request or can be added to substrate description after the request is satisfied
3. Ethernet connection request form one Euca cluster to another
Manifest example
Creating virtual topologies in ExoGENI
Discussion points (use as needed)

- RDF Primer
- RDF Vs. XML
- Scalability
RDF Primer

- RDF is declarative
- RDF is represented as triples
  - `<subject> <predicate> <object>`
- Trivial example
  2. `dc:creator`
  3. `http://www.renci.org/People#IliaBaldine`
- `dc` is a shorthand for
  - `http://purl.org/dc/elements/1.1/`
  - Dublin Core Metadata Initiative (DMCI)
- An open organization engaged in the development of interoperable metadata standards that support a broad range of purposes and business models
RDF Graphs

http://geni-orca.renci.org/trac/gec7/WorkshopPresentation.pptx

http://www.renci.org/People#IliaBaldine  en
RDF Representations

- **NTriples**

- **Turtle: Terse RDF Triple Language**
  - Extension of NTriples
  - @prefix, @base and other shorthand notation

- **RDF-XML**
  <rdf:RDF>
  <rdf:Description rdf:about="http://geni-orca-renci.org/trac/gec7/WorkshopPresentation.pptx">
    <dc:creator rdf:resource="http://www.renci.org/People#IliaBaldine">
  </rdf:Description>
  </rdf:RDF>
XML vs. RDF

XML representation

DOM

In-memory objects, XPath

Application logic

RDF representation (RDF-XML, N3, Turtle)

RDF triple store in-memory or persistent

Inference engine

SPARQL query engine

Application logic

Queries

Assertions

Rules

In-memory objects

Presentation title goes here
XML vs. RDF (continued)

• Information model
  – XML DOM is tree-structured
    • Forces tree-like structure on information schema
  – RDF/OWL is a multi-colored graph
    • More natural way of thinking about information schema

• Data model
  – ‘The curse of XML’: almost ‘too easy’ to create data schema with no incentive to share schemas across projects
  – RDF has multiple standard data model representations (RDF-XML, N3) and encourages sharing, merging of schemas through the use of URLs for resource naming

• Toolage
  – RDF/OWL: Fewer tools to go from model to programming language constructs/objects
  – XML: rich set of tools that make it easy (see ‘the curse of XML’ above)
Naming

• XML resource descriptions (NML, GENI RSpecs) use URNs – flat or hierarchical names
• RDF uses URLs
• URLs combine name and location, URNs do not
  – URL advantage is in encouraging sharing of
Scalability challenges

• **Persistent storage: performance**
  – In-memory, file, SQL database
  – Triple store: Parliament™ (BBN), SHARD,…

• **Query vs. inference rules**
  – Expressiveness of graph pattern match
    • Web scalable data processing techniques
    • Parallel data processing techniques (Map-Reduce)
  – Fact-finding towards knowledge discovery
    Multi-stage query and answering process
Parliament

![Diagram of Parliament components and dependencies]

- **SPARQL**
  - Jetty + Joseki
  - Named Graph Support
- **SERQL**
  - Jena
  - Jena Graph for Indexes
  - Jena Graph for Parliament
- **Java Native Interface**
  - Parliament
  - Rule Engine
- **Operating System**
  - Berkeley DB

**Parliament Components**
- **Third-Party Components**