Explicit Resource Usage Policy Management and Enforcement in Grid Computing

A PhD Proposal by Jun Feng
Outline

- Introduction
- Problem statement
- Proposed solutions
- Research schedule
- Expected contributions
- Related work
- Questions
Outline

• Introduction
• Problem statement
• Proposed solutions
• Research schedule
• Expected contributions
• Related work
• Questions
Grid Computing

• Grid - resource pool shared by all
  – Distributed heterogeneous resources
    • CPU, storage, database, ... 
  – Across administration domains 
  – Service oriented architecture
    • OGSA, WSRF, ... 
  – Virtual Organization (VO)
Grid Example - OSG

- Open Science Grid
  - Inter-continental Grid
  - Scientific applications
  - Large number of resources and users
  - Multiple VOs
    - ATLAS
    - CMS
Grid Depends on Participants

• Resource providers (RPs)
  – Should be able to control *how* their resource can be consumed
  – Grid behavior should be *predictable*

• Grid users
  – Preferences and QoS requirements enforced by Grids
Outline

• Introduction
• *Problem statement*
• Proposed solutions
• Research schedule
• Expected contributions
• Related work
• Questions
My Research Problems

#1 Insufficient control for RPs
#2 Unpredictable Grid/VO behavior
#3 Lack of support for user policies
#4 Hard to know what was wrong in case of policy violation
Problem #1
Insufficient Control for RPs

- Access control is not enough
- Usage policies are different from access control policies
  - “Purge all Grid data on /scratch every week”
- Usage policies are not supported well in current Grid software
  - Globus GRAM, GridFTP
  - Condor
  - Legion
Problem #2
Unpredictable VO Behavior

• VOs today are merely membership plus access control, e.g., VOMS, CAS
  – Unfair resource sharing
  – Inappropriate data storage

• VO should be more than that
  – *Fairness* policies
  – Acceptable usage policies directly tied to software mechanisms
Problem #3
Lack of Support for User Policies

• Grid users cannot have policies and Grid software does not incorporate user policies
  – “Prefer site which charges less”
  – “Prefer site which supports some privacy protocols”
  – “Notify me whenever Marty Humphrey reads my data”
Problem #4
Hard to Know What is Going On

• Implicit policies are there, e.g., scheduling, auditing
• Precise policy violation information can not be reported
• Hard to do Grid debugging, error propagation, etc
Outline

• Introduction
• Problem statement
• *Proposed solutions*
• Research schedule
• Expected contributions
• Related work
• Questions
Shape of Solution

- Policy language to express explicit usage policies
- New policy engine to consider information outside of security contexts
- Policy management MyPolMan
- Enforcement of representative usage policies in GridFTP and GRAM
- Metrics to evaluate VO fairness
- Fairness policy enforcement
Usage Policy Types

• Configuration policies
  – “Provide 40% of disk space to Grid”
  – “Service is open between 9:00AM-5:00PM”

• Conditional policies
  – “If keyboard is touched, kill Grid jobs”
XACML & WS-Policy

• XACML
  – Suitable for conditional policies
  – Access control policies only
  – Security context information only
  – Request/response interactions only

• WS-Policy
  – Defines “and”, “or” semantics
Proposed Work: Usage Policy Language

• Investigate extending XACML for usage policy
• Configuration policies
  – Name-value pair
• Elements
  – <event>, <operation>, <status>
• Rules
  – <event><operation>
  – <subject><resource><action><request><status><operation>
Proposed Work: Policy Engine

- Solicited style
  - Request/response
- Unsolicited style
  - Event
  - Timer
Proposed Work: Policy Management

- Policy authoring
- Policy repository
- Policy management operations
  - Retrieve, update, delete
  - Possible query
  - Access control on policies
- Policy transfer
MyPolMan Components

Grid Users
Grid Resource Providers
Grid Admins

Policy Authoring

Policy Services
VO-Level Policy Services
Site-Level Policy Services

Policy Agents
Policy Consuming

Policy Repository
Policy Caches
Policy Document

Grid Services
Authorization
Job Scheduler
Data Movement
...
Possible MyPolMan Deployment

- Site level MyPolMan service
  - Site resource provider usage policies
- VO level MyPolMan service
  - Grid user policies
  - Possible VO policies
Proposed Work: Grid Fairness

• Consider these values for all sites
  – Average job delay/advance for each site
  – Ratio of consumed grid resource and total consumed resource for each site
  – Ratio of consumed grid resource and contributed resource for each site
Enforcement Mechanisms of Grid Fairness

- Local accounting
- Exchange information between sites
- Job migration
  - Migrate previously placed jobs to new places for execution
  - Must consider user policies as well
- Priority adjustment
  - Meta-scheduler and local scheduler
Evaluation Plan

• Implementation and evaluation on UVaCG
  – Leverage and extend .NET based Grid software (e.g., GridFTP .NET)

• Discrete event simulation
  – Simulate a Grid with several sites submitting jobs over a long period
  – Workload driven
  – With/without meta-scheduler

• Investigate
  – Effectiveness of enforcement mechanisms
  – Cost of fairness policy in Grid in terms of metrics such as Grid utilization rate, etc.
Put Everything Together
Put Everything Together

Site Administrators → Publish Policies → Site MyPolMan

Site MyPolMan → Retrieve Policies

Retrieve Policies

Policy Engine → Guidances

Guidances

Grid Services
Put Everything Together

1: Login

2: Ask for User Policy
3: Default User Policy
4: ProxyCert w/ User Policy
5: Job Description
   Default User Policy
   Per Job Policy

6: VO Policy
7: Site Resource/Service Policy
8: Place Job
9: Site Resource/Service Policy

Site Submission

Site 1
Site 2

VO and user policy enforcement
Early Progress - Policy Directed Data Movements on Grids (ICPADS 2006)

- Prototype implementation of MyPolMan
  - Based on CredEx
  - Policies associated with credentials
  - Upload, retrieve, delete
  - Push based policy distribution
- Representative policies for storage resources
  - Service available time
  - Some quota policies
  - Maximum number of streams
- Policies enforcement in GridFTP
  - GridFTP implementation on .NET (GRID 2005)
  - Enforcement of some policies
  - Precise policy violation on clients
Early Progress - Fairness Simulation

- Simulator being built using SimJava
  - FCFS + backfilling local scheduling
  - Real workload from Parallel Workload Archive
  - Target at TeraGrid setting
Related Work

- Policy framework
  - IEEE/DMTF policy framework
  - GGF policy group documents
- Grid authorization (access control)
  - XACML, Akenti, CAS, PRIMA, ...
- Grid resource partitioning policies
  - VO allocation, C. Dumitrescu, et. al
  - Grid wide partitioning, E. Elmroth, et. Al
  - SGAS, GridBank, DGAS, Gold, ...
- Policy work of Prof. Minsky at Univ. of Rutgers
Related Work (cont.)

• Grid scheduling policies
  – Maui, Silver, CSF, LSF, SGE, ...

• Web services policies
  – WS-Policy, WS-SecurityPolicy, ...

• Grid site policies
  – Condor ClassAds, Legion, ...

• Autonomic computing
  – PMAC, ...

• Grid Economy
  – Gridbus, R. Buyya, 2000, ...
## Research Schedule

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Quarters (Starting from Jan 2006)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Identify resource usage policies at site and VO level</td>
<td>X</td>
</tr>
<tr>
<td>Define policy model and languages to describe policies</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Build, test and evaluate MyPolMan, Enforce resource providers policies in GridFTP and GRAM services</td>
<td>X</td>
</tr>
<tr>
<td>Enforcement and evaluation of VO operational policies</td>
<td>X</td>
</tr>
<tr>
<td>Thesis writing</td>
<td></td>
</tr>
</tbody>
</table>
Expected Contribution

- Policy language and associated policy engine to express and evaluate explicit resource usage policies
- Algorithms and systems to enforce some representative usage policies in Grid software and protocol such as GRAM
- Grid policy management system MyPolMan
- Fairness in Grid and enforcement mechanisms
Impact

- Better resource usage control for RPs
- Fair sharing to encourage provider participation in Grids
- User policies are properly reflected in Grid operations
- Better Grid debugging and error propagation
Questions
Policy

• *A plan to guide* entity behaviors
  – Rules
  – Constraints
  – Preferences
  – Obligations
Why Explicit Policies

• Separation of policy and mechanism
  – Flexible
  – Manageable
• Better scheduling
• Predictable Grid behavior
  – Error propagation
Problems with Policies in Current Grid Software

• Implicit policies
  – Not flexible
  – Hard to know what went wrong in case of policy violation

• Insufficient policies
  – Support only certain resource types
  – Limited policy terms
  – No VO level policies
Policies on Grids

• Virtual Organization policies
  – Acceptable usage
  – Resource partitioning
  – ...

• Resource provider policies
  – Access control
  – Usage constraints
  – Auditing
  – Pricing
  – ...

• Grid user policies
  – Preferences
Access Control Policies

- Permit/deny decisions
- Typically depends on attributes of
  - Subject: identity, role, ...
  - Resource: identify, resource content, ...
  - Action: read, write, execute, ...
  - Environment: current date, time, ...
- Request/Response interactions between PDP/PEP
Usage Policies in Grids

• “Provide 40% of disk space to Grid”
• “Grid job can not take more than 8 nodes”
• “Purge all Grid data on /scratch every week”
• “If keyboard mouse is touched, preempt all jobs on the host”
How Usage Policies Differ From Access Control Policies

• More than permit/deny decisions
  – “Maximum download rate is 3Mb/s”

• Depends on the information outside of the security context
  – “Provide 40% disk space to Grid”

• Event/timer interactions
  – “If keyboard is touched, kill Grid jobs”
  – “Purge Grid data on /scratch every day 5:00 PM”
Importance of Problems

• Discourage RPs & users participation
  – Insufficient control for RPs
  – Unpredictable VO behaviors
  – None or less user policies

• Grid debugging, error tracing and propagation become difficult
Grid Fairness

• How can a Grid be “fair” to its participants?
• How to achieve fairness in Grid?
  – Capabilities of individual resources
  – Computational and data Grid
  – Volatility of Grid
  – With/without meta-schedulers
  – Local resource provider policies
• Does fairness come with any cost?
A “Fair” Grid

• All participants take the costs & benefits of resource sharing evenly to a certain degree
  – Costs, e.g. job is delayed because of resource sharing
  – Benefits, e.g. job is started earlier on other sites
Research

• Looking for appropriate policies for various resources types
  – Identifying implicit policies
  – Identifying desired policies

• Explicit language to express usage policies

• Algorithms and system to enforce some challenging policies such as grid wide quota, and fairness policies

• Management of Grid policies