Broker Overlay Architecture for Decentralized Grid Management

Abdulrahman Azab and Hein Meling

{abdulrahman.azab,hein.meling}@uis.no
Issues of Large Scale Grid Computing - 1

- **Machine Organization** - Flat
• **Machine Organization** - Hierarchical
• **Machine Organization** - Cell
• **Scheduling** - Centralized
• **Scheduling** - Centralized
• **Scheduling** - Decentralized
Issues of Large Scale Grid Computing - 5

- **Scheduling** - Decentralized
Issues of Large Scale Grid Computing - 6

- **Failure Handling** – Worker Failure
Issues of Large Scale Grid Computing - 7

- **Failure Handling** – Broker Failure

![Diagram](image-url)
# Related Work

<table>
<thead>
<tr>
<th></th>
<th>Condor</th>
<th>BOINC</th>
<th>gLite</th>
<th>UNICORE</th>
<th>NorduGrid</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Machine Organization</strong></td>
<td>Flat</td>
<td>Hierarchical</td>
<td>Hierarchical</td>
<td>Hierarchical</td>
<td>Hierarchical</td>
</tr>
<tr>
<td><strong>Scheduling</strong></td>
<td>Centralized</td>
<td>Centralized</td>
<td>Centralized</td>
<td>Centralized</td>
<td>Centralized</td>
</tr>
<tr>
<td><strong>Handling Worker Failures</strong></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td><strong>Handling Broker Failures</strong></td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>
Proposed Architecture
Proposed Architecture

- Consider:
  - Grid as bus
  - Grid node as bus seat
  - Grid job as passenger
Problem
Problem
Problem
Problem
Solution

Load balancing
Solution

Load balancing
How to achieve load balancing?
How to achieve load balancing?
How to achieve load balancing?
How to achieve load balancing?
How to achieve load balancing?
How to achieve load balancing?
How to achieve load balancing?

Client

Request

Grids with vacant Matching workers
How to achieve load balancing?
Resource Information Exchange - 1

- Resource information for nodes is stored in a three field **Resource Information Data Block**, **RIDB**.
- Each broker maintains a set of RIDBs for all nodes in the system.

<table>
<thead>
<tr>
<th>Resource Index</th>
<th># information</th>
<th>Time of Last Read</th>
</tr>
</thead>
</table>

![Diagram of a VO network with a broker (B) and nodes connected]
Resource Information Exchange - 2

**Broker**

- GetState(Rs,x,node)
- Update Rs(node)

**Regular Node**

- Start
- Rs = Current Resource State
- My Broker Alive?
  - yes
  - Call MyBroker.GetState(Rs, Me)
  - End
Each broker performs one exchange operation with a single neighbor broker each time unit. The exchange operation is done by updating each resource information data set in each of the two brokers with the newest data blocks.
Service Allocation (Scheduling) - 1

Service Allocator (regular node)

Service Allocator Submits a service S

Priority(S) = Normal

Add new service to the local queue (Broker)

Service S Received

Priority(S) = High

Allocate in Q(n+1)

Allocate in Q(0)

Allocate from local queue (Broker)

Start

Check Service Queue

Empty

yes

Set S = Q(0), Remove Q(0)

no

Matching Local Node (Nx)

S.DeploymentAttempts - = 1

no

Matching Global Node (Ny)

End

Deploy to Nx

Pass S to a neighbor Broker Bx

Pass S to Broker(Ny)
Service Allocation (Scheduling) - 2
Failure Handling – Broker
Failure Handling – Broker
Failure Handling – Broker
Failure Handling – Broker

**Broker**

1. GetState(Rs, x, node)
2. Update Rs(node)

**Regular Node**

1. Start
2. Rs = Current Resource State
3. My Broker Alive?
   - yes: Call MyBroker.GetState(Rs, Me)
   - no: Request Membership From another Broker B
4. Request Granted?
   - no: Set MyBroker = B
   - yes: End
Simulation Model
Performance Evaluation

- Validity of the stored resource information.
- Efficiency of service allocation.
- Impact of broker failure on resource information updating.

\[ N \rightarrow \text{Total Grid size}, \quad M \rightarrow \text{Number of VOs} \]
Performance Evaluation

- Broker Overlay Topologies

- Ring

- Hyper-Cube

- Fully connected

- Wire-\( k \)-out
Validity of the stored resource information

- The deviation of the reading time values of resource information data blocks, RIDBs, stored in the resource information data set, from the current cycle in a broker, with the simulation cycles.

- The deviation value for cycle (c):

\[
D(c) = \sqrt{\frac{\sum_{i=1}^{N} (\text{Time(RIDB(i))} - c)^2}{N}}
\]
Validity of the stored resource information

N = 100, M = 20

N = 500, M = 100 (log scale)
Efficiency of Service Allocation

- One broker periodical allocation.
Impact of Broker Failures on Resource Information Updating \( (N = 500, M = 100) \)

- a) Ring broker overlay topology
- b) Fully Connected broker overlay topology
- c) Wire-k-Out broker overlay topology, \( k = 60 \)
- d) Hyper-cube broker overlay topology
Conclusions and Future work

• Broker overlay Grid management model retains the system decentralization and increases the scalability.

• Hyper-cube topology provides scalability similar to the fully connected topology. Ring topology is not applicable in case of broker failures.

• As a future work, other collaboration aspects in a multi-virtual organization environment (e.g. security and rules of sharing) will be considered.
Questions