The LA Grid Meta-Scheduling Project

Team: Liana Fong[1], S. Masoud Sadjadi[2], Yanbin Liu[1], Ivan Rodero[3], David Villegas[2], Selim Kalayci[2], Norman Bobroff[1], and Julita Corbalan[3]

1: IBM T. J. Watson Research Center, Yorktown Heights, NY 10598

2: Florida International University, Miami, FL 33199

3: Barcelona Supercomputing Center, Barcelona, Spain

I: Objectives

Objective

- Support interoperation and cooperation of network of distributed schedulers

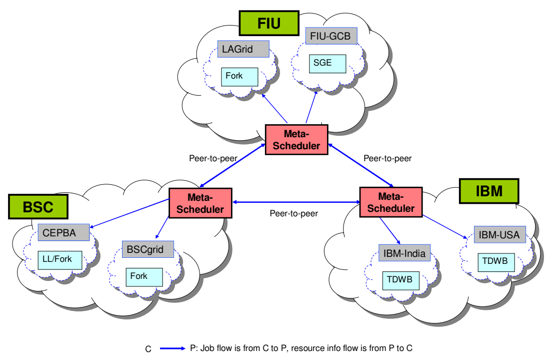
Strategic Importance

- Enhance usability: common job control language to different resource domains
- Drive interoperability of schedulers: proprietary and open-source
- Provide integrated scheduling views for enterprise and grid customers

Technology Benefits

- Meet various user service objectives: policy driven (e.g. capability based, response time based)
- Maximize resource availability to users with transparency of locations
- Optimize utilization of resources across domains

II: P2P Meta scheduling



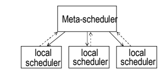
Some key aspects of the Metascheduler Protocol:

- Heterogeneous sites; inner structure of domains doesn't effect the functionality of the protocol.
- Site autonomy; each metascheduler is responsible from its own site, and offers as much information as it wants to other sites.
- Peer-to-peer; no centralized body, no single-point of failure.

III: Related Work

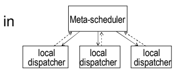
Centralized model:

- Meta-scheduling has direct information of all resources available at the various institutes of the virtual organization
- Responsible for scheduling job execution on all resources
- Local schedulers at individual institutes will act as job dispatchers.



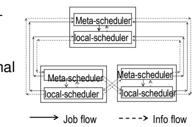
Hierarchical model:

- Meta-scheduling has no direct access to resources in the virtual organization
- Assign jobs to the local schedulers of the various institutes
- Local schedulers will match jobs to resources.



Distributed model:

- Multiple local schedulers with a companion meta-scheduling functional entity
- Local schedulers can submit jobs to each others through their respective meta-scheduling functional entities.



Ref: "Distributed job scheduling on computational grid using multiple Simultaneous Requests" by Vijay Subramani, Rajkumar Kettimuthu, Srividya Srinivasan, and P. Sadayappan

IV: System architecture

Connection API

- Establish and terminate connections between domain meta-schedulers.
- Negotiate roles and connection parameters using the interface
  - Provider roles: provide resources for job execution; is responsible of sending out resource information
  - Consumer roles: use resources provided by providers; route job request to providers.
- Send heart beats: exchanged to guarantee the healthy state of the connection.

Resource exchange API

- Exchange the scheduling capability and capacity of the domain controlled by the meta-scheduler
  - Exchanged information can be a complete or incremental set of data

Job management API

- Submit, re-route and monitor job executions across schedulers