The LA Grid Meta-Scheduling Project

Team: Liana Fong¹, S. Masoud Sadjadi², Yanbin Liu¹, Ivan Rodero³, David Villegas², Selim Kalayci², Norman Bobroff¹, and Julita Corbalan³

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

- 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 affect 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 other through their respective meta-scheduling functional entities.

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

Connection API
1. User Client takes the job request from the local User. This request is forwarded to Global Scheduling Manager (GSM).
2. GSM queries the Resource Manager (RM) for resources. RM stores information about local and remote resources.
3. If available resources are found on local site, job request is forwarded to Site Scheduling Manager (SSM).
4. SSM leverages Gridway functionality to submit the job to the Grid Middleware (Globus).
5. If there are not available resources locally, job request is sent to a remote site through WS Client
6. Alternatively, job requests from other peers can be received from the WS layer.

Resource exchange API
1. The eNANOS Client forwards the user requests to the eNANOS Broker.
2. The remote request from the P2P infrastructure are managed by regular WS (Axis2) acting as a wrapper to a GT4 service that implements the LAGrid APIs and protocols. Connections and other data is stored in Resource Properties.
3. Jobs and resources (aggregated data) obtained from local and remote sites are used in the eNANOS Resource Broker scheduling. Jobs are executed under the local domain through Globus services, or are forwarded to other meta-scheduler.
4. eNANOS provides its resources data, forwards jobs and performs other operations (such as sending heart beats) through a WS Client.