

Some use cases for Graap

Jim Pruyne (pruyne@hpl.hp.com)

HP Labs

The very long running service or application

- ◆ A user has developed a service that they wish to provide for the grid community at large (e.g. Netlib).
 - Require resources to host the service
 - No known or expected completion time
- ◆ Must be able to specify indeterminate run-length or completion time.

The variable demand long running service

◆ As before, but:

- Demand varies, so resource demand varies. Netlib still works here presumably

◆ Must specify “demand profile.”

- Deterministic
- Probabilistic

The "bottom-feeder" application

- ◆ An application that can consume all available resources
 - E.G. [SETI@Home](#), distributed.net, ...
 - Low priority
 - Flexible to variance
- ◆ Is this even a case of advance reservation?
 - How to arbitrate between multiple apps. of this class?

The configurable application

- ◆ An application that can be configured relative to its resource availability
 - E.g. a parallel computation that requires a power of 2 number of compute resources, but can be configured to use any power of 2.
 - Can wait until run-time to choose the level for this run, and may be sensitive to other QoS criteria.
- ◆ Must be able to specify alternatives.
 - Alternative selection requires user's criteria (e.g. cost, turnaround time, etc) for choosing.
 - Does this require negotiation to get it right?

The “Templated” Application

- ◆ Application developer and application user are distinct
 - Developer provides “template” describing app. Demands to be filled in by user based on their needs, environment, ...
 - E.g. A proprietary scientific app.
- ◆ Introduce “free variables” in to specification
 - Ideally templates are “pre-registered” and named allowing updates by developer

Monitoring of resource utilization

◆ On-line:

- State
- Accumulation
- Events
- E.g. accounting/monitoring app.

◆ On-going communication with RM system

- Authentication, etc.

Reservation modification

- ◆ Need to change reservation request after initial submission
 - E.g. Change based on observed change in state of RM system
- ◆ Modification should be “incremental,” not equivalent to a cancel/re-submit operation

Allocation modification

- ◆ Need to change reservation request after job started
 - Change based on observed progress
 - E.g. Weather forecasting simulation
- ◆ RM system must not “lose contact” with job after it starts
 - App. informed of change in resource state