Fault Detection of TeraGrid Resources Using Inca

Chin Hua Kong\textsuperscript{1}, Sangmi Lee Pallickara\textsuperscript{2} and Marion Pierce\textsuperscript{1}
\textsuperscript{1}Community Grids Laboratory and \textsuperscript{2}Research Technology, UITS, Indiana University

TeraGrid 2009 Conference

Motivation
- Submission 10,000’s of jobs may only require 25 minutes but processing required a day to a week.
- Some jobs might fail during processing or in queue.
- Resource down
- Resource up but Gram service down
- No mechanism to automatically recover the failure
- User has to manually check the status periodically and decide if job failed and re-assign job to other machine.
- Problems:
  - Solution inefficient: time consuming and cost of human resource.
  - Hard to identify remaining jobs since huge submission.

Background: Swarm

Swarm Features Include
- Monitoring framework for large scale jobs.
- User based job scheduling.
- Ranking resources based on predicted wait times.
- Standard Web Service interface for web applications.
- Extensible design for the domain specific software logics.

Challenges
- Target failures: i) TeraGrid resource down for service.
- ii) TeraGrid resource up but gram job down.
- Down time varies as between an hour a week.
- Maintenance of resource is specific to the organization. Each organization plans down time in site-specific ways.
- No standard shut down mechanism from TeraGrid.
- Condor job status unclear during down time. Statuses are shown in queue or hold.
- Notification is slow and inefficient through email.
- Unable to implement fault recovery mechanism without reliable information.
- Building self-checking mechanism is not trivial as network ping, due to resource might up but service down or service might up but execution failed.

Fault Detection I: Fault Manager
- Using Inca TeraGrid Resources Management Service.
  - Three operations (RESTful WS interface):
    - Pre-WS-Gram: updated every 3 minutes
    - External-Ping: updated every 5 minutes
    - Pre-WS-Gram batch: updated every 12 hours
  - XML handler.
  - Update incident report.
  - Monitoring the reported resource.
  - Periodically query to Inca TeraGrid Resource Portal.
  - Parsing status information that is encoded in XML.
  - Detecting changes of status
    - Down (Up -> down)
    - Up (down -> up)
  - Notify mechanism: observable & observer design pattern
  - Periodically update every 3 minutes.

Fault Detection II: Job Distributor
- Self-Tracking the Job status.
  - Job Distributed report abnormal behavior to Resource Manager.
  - Job in hold status.
  - Processing time more than user specified.
  - Notify fault manager.
  - Portable for other application with observer.

Conclusion
- We can detect possible system faults in TeraGrid HPC cluster and notify to the other software component within Swarm.
- Inca provides us easy-to-access interface for recent status of TeraGrid HPC clusters.
- Self-detecting scheme detects faults which can happen between Inca’s monitoring schedule.
- Inca limitation:
  - Periodical update: 3 minutes - 12 hour.
  - Reliability and accuracy.
  - Inca service down or traffic busy.
- Network latency problem: Cannot update status from Inca immediately.
- Dependency on Inca XML schema, changes will effect parsing failed.

Future Work
- Fault Tolerance.
  - Migrate job if machine is down.
  - Discover new resource for the task.
  - Implement job clean up when the resource is back.
  - Expend to other available service to avoid dependent on single service.
  - E.g. GPIR.
- Support for Cloud computing clusters or Condor cluster.
- Develop Web service interface for the resource monitoring.