Fault Tolerance and Recovery for Grid Workflow Systems

Anirban Mandal
Gopi Kandaswamy
Dan Reed

VGrADS Workshop
April 2007
Presentation Outline

• Motivation and rationale
• Fault tolerance and recovery service
• Fault tolerance and recovery algorithms
  – over-provisioning, simple restart, migration
• Proposed next steps
• Demonstration
  – LEAD-VGrADS integration
Motivation

- **Reliability and performance are related**
  - failure is the limiting case of poor performance
  - both involve measures of behavior over time

- **Large, complex workflows are sensitive to failures**
  - faults are the norm, rather than exceptions
    - distributed systems, services and resources
  - completion “guarantees” are problematic
    - workflow completion is probabilistic in the presence of faults

- **Many time-critical workflows are deadline driven**
  - severe weather events, disaster response, …
Fault Tolerance/Recovery (FTR) Service

- Application Performance Models
- Resource Reliability Models
- NWS, MDS
- BQP

Fault Tolerance/Recovery Service

Deadline & Success Probability

Over-provisioning? Simple restart? Migration?
FTR Algorithms

- **Notation**
  
  - $p_i$: one hour failure probability of resource $i$
  - $h_i$: expected execution cost of application on resource $i$
    - queue wait time
    - expected computation time
    - expected communication time
  
  - $x$: required success probability
  
  - $d$: required deadline

- If the reliability function is linear, probability of failure
  
  $$m_i = \max (p_i \times h_i, 1)$$

- Resource $i$ represents (queue, #nodes) combinations
Over Provisioning

• Find
  – degree and resources for over-provisioning

• Number of application copies
  – meet a deadline $d$ with a success probability $x$

• Solve the following optimization problem

For given $[1..M]$ resources, find a partition $P = \{s_1, s_2 \ldots s_n\}$ of $[1..M]$ such that

\[ 1 - m_{s_1} * m_{s_2} * \ldots m_{s_n} \geq x \ \land \ |P| \text{ is minimum} \ \land \ \min \{h_{s_1} \ldots h_{s_n}\} \leq d \]

Probability of failure

Minimum number of resources meeting deadline
Simple Restart

• Among the current available resources …
  – (universe – faulty resources)
• Find the best resource that
  – meets the deadline and success probability
• Mechanism can also be used for
  – runtime resource selection
• Two FTR mechanisms based on simple restart
  – restart with retries
  – restart if there is no progress
Migration

• Find the best migration path
  – more complex optimization problem
  – need to optimize orderings

• Solve the following optimization problem

Find a partition $P = \{s_1, s_2 \ldots s_n\}$ and ordering $\{s_1 > s_2 > \ldots s_n\}$ such that

$$1 - m_{s_1} * m_{s_2} * \ldots m_{s_n} \geq x \land |P| \text{ is minimum} \land (t_{s_1} + \ldots t_{s_n} + q) \leq d,$$

where $t_{s_i} = (p_{s_i} * h_{s_i} > 1)? 1/p_{s_i} : h_{s_i}$ and $q$ is the migration overhead.
Proposed Next Steps

- Modeling resource reliability
  - draw data from job log files in the resources
  - maintain constant updates of reliability estimates

- Reliability Information Service (viz. BQP)
  - “What is the probability that a job, when submitted to a queue at a resource will die before completion because of a failure?”

- Balancing multiple mechanisms
  - given resource constraints
    - e.g., over-provisioning and migration
  - quantifying resource “wastage” with common metric
Proposed Next Steps (2)

• Develop runtime rescheduling strategies
  – requires following changes in VGrADS software
    • ability to re-acquire slots
    • ability to relinquish unnecessary slots
    • ability to accommodate FTR directives
    • ability to schedule remaining workflow
  – requires following change in workflow engine
    • ability to run a re-scheduled workflow
Proposed Next Steps (3)

• **Deadline estimates**
  – *given a deadline for the entire workflow*
    • find deadlines for individual applications
  – *requires critical-path analysis*
FTR Demonstration

• Launch workflow from the LEAD portal
  – build new experiment
  – select forecast region and workflow
  – fetch data

• FTR service manages application execution by
  – choosing the fault-tolerance mode (OVP, RST)
  – invoking application service(s) as per current mode
  – monitoring for failures

• Workflow composer shows workflow progress
  – status of application currently running
  – workflow execution status via notifications etc.
Screenshots

GETTING STARTED WITH THE LEAD PORTAL

Now that you are logged into the LEAD Portal, you have several additional capabilities available to you. Below is a list of links to those capabilities along with accompanying tutorials and documentation to help you get started using the LEAD Portal. More extensive documentation is available under the Help tab.

**LEAD Portal Overview**
- Take a tour of the LEAD Portal (requires Flash plugin)

**My Workspace**
My Workspace provides a tree-based view of your projects, experiments, and data that are stored in MyLEAD.
- Take a tour of My Workspace (requires Flash plugin)
- Getting Started documentation

**Experiment Builder**
Experiment Builder provides the ability to create, manage, configure, and monitor LEAD experiments.
- Running a KRFS Forecast Tutorial (requires Flash plugin)

**Geographic Data Search**
The Geographic Data Search tool allows you to perform searches for data by specifying any combination of spatial, temporal, or data category parameters. You can access the tool without logging into the LEAD Portal, but now that you are logged in, you can also save data that you have found into your personal workspace.
- Take a tour of Geographic Data Search (requires Flash plugin)
Screenshots
Screenshots
Screenshots
Questions ?