Resource Management I

- Resource Allocation is a venerable problem
  - within CS, dates back roughly 60 years
  - elsewhere, pre-dates human existence
- Four lectures on Resource Management in Grid Computing:
  - Begin with breadth of problem and overview of variety of issues/approaches
  - then two lectures on specific aspects and relevant techniques
  - final lecture on more unusual perspectives
- Will draw on published surveys and prior work

Primary Source Material

- You must read this paper:
  - “A taxonomy and survey of grid resource management systems for distributed computing”
  - Krauter, Buyya & Maheswaran
  - Software — Practice and Experience 2002, vol.32, pp.135–164

  - taxonomies focus on resource management system issues
  - survey covers a broad range of systems
- Should also visit the GRAM website:

  `http://www.globus.org/toolkit/docs/4.0/execution/wsgram`

Resource Management System structure

- See figure 3 in the Taxonomy/Survey paper
- Note that this is an *abstract* structure/architcture
- It identifies:
  - ten internal components/functional units
  - two sorts of internally held information
  - four different interfaces
    - Resource Provider
    - Resource Consumer
    - Related services (support interface)
    - Peers (Resource Managers on other systems)
  - interactions between these parts
- Useful for appreciating the complexity of the issue
- Helpful for identifying tractable sub-problems
- Note that this is managing *some* of the resources (one provider)

Requirements

- resources are reusable (CPU not specific CPU second)
- provider controls the resource
- consumer utlises the resource
- resource manager optimizes use of resource pool w.r.t some metric

- Distribution Issues: scalability, responsiveness, fault-tolerance, stability
- & Grid Issues: extensibility, adaptibility, site autonomy, QoS, co-allocation
- plus Security: e.g. may wish to minimise number of domains used

- Must always remember autonomy of participating organisations
  - must agree to, and abide by, contracts for using/providing resources
  - each organisation has its own preferences, approaches, perspectives
  - common problems may be resolved in incompatible ways locally
  - inter-dependence introduces fragility

Taxonomy I

- Organising machines in the Grid (fig.4):
  - flat
  - hierarchical
  - cells: themselves flat or hierarchical
- Describing resources and operations on them (fig 5):
  - schema based (fixed or extensible)
  - object model (fixed or extensible)
- Naming resources, and organizing those names (fig 6):
  - hierarchical
  - relational
  - hybrid hierarchical/relational
  - graph

Taxonomy II

- Storing resource information (fig.8):
  - network directory (X.500/LDAP, relational, other)
  - distributed objects (language or object-model based)
- Discovering resource info (fig.9)
  - agents
  - queries (centralized or distributed)
- Disseminating resource info (fig.10)
  - online/on demand
  - batch/periodic (pull or push)
- QoS support (fig.7):
  - hard, soft or none

Scheduling and Allocation I

- Scheduler Structure (fig.11)
  - centralized
  - hierarchical
  - decentralized
- State Estimation (fig.12)
  - predictive
    - heuristics
    - pricing models (see 4th lecture in this group)
    - machine learning (see ML-M module)
  - non-predictive
    - heuristics
    - probability distribution

Scheduling and Allocation II

- Re-scheduling
  - periodic
  - event-driven/online
- Scheduling Policy
  - fixed
    - system-oriented
    - application-oriented
  - extensible
    - ad-hoc
    - structured

System Survey Overview

- This is most valuable part of the paper, summarised in table 1
  You will be assumed to have read it thoroughly for the exam.
- Covers 15 systems, including:
  - Condor (typical cluster approach)
  - Globus (more of an umbrella)
  - European Data Grid (typical hierarchical approach)
- broader perspective will inform your insights into the practical work undertaken in tutorial sessions
- Observations (ask yourselves “why?”):
  - extensibility is almost always provided (but degree varies)
  - dissemination is usually by periodic push
- Meta-issue:
  - notice how a thorough survey allows the authors to identify combinations of features that have not yet been explored

WS GRAM

- Grid Resource Allocation and Management service
- part of the Globus toolkit
- suite of web services and a "pre-web services" Unix server suite
  - submit, monitor, and cancel jobs on Grid computing resources
- “Grid computing resources are typically operated under the control of a scheduler which implements allocation and prioritization policies while optimizing the execution of all submitted jobs for efficiency and performance. GRAM is not a resource scheduler, but rather a protocol engine for communicating with a range of different local resource schedulers using a standard message format.”

Community Scheduler Framework

- “The Community Scheduler Framework 4.0 (CSF4.0) is a WSRF-compliant Grid level meta-scheduling framework built upon the Globus Toolkit. CSF provides interface and tools for Grid users to submit jobs, create advanced reservations and define different scheduling policies at the Grid level. Using CSF, Grid users are able to access different resource managers, such as LSF, PBS, Condor and SGE, via a single interface.”
- How (and why) is this different from/similar to the GRAM description?