Resource and Application Models for Advanced Grid Schedulers

Aleksandar Lazarevic, Lionel Sacks

Dept. of Electrical and Electronic Engineering,
University College London

Problem

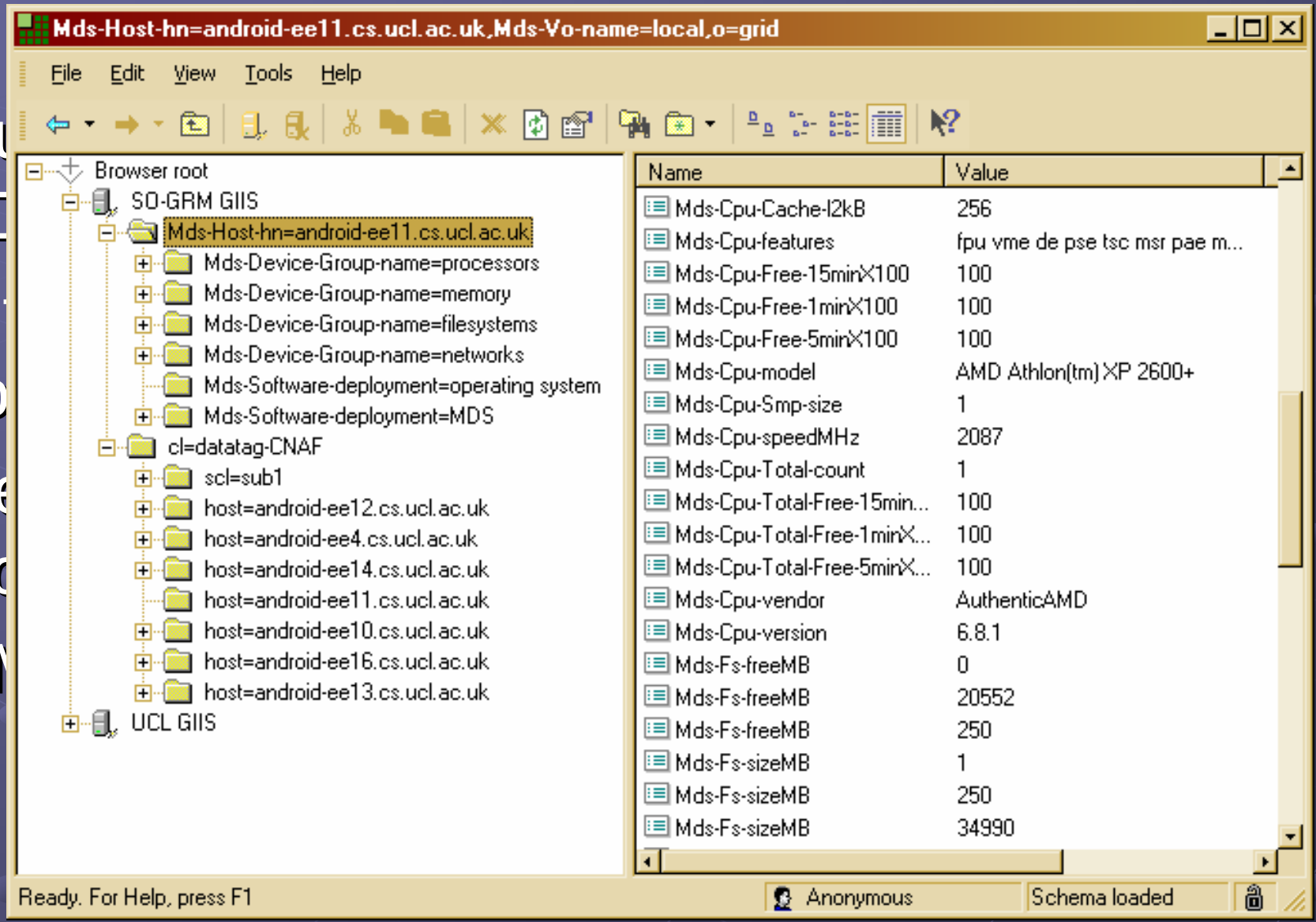
- Heterogeneous and dispersed systems
- Quest for effective scheduling technique
- Good scheduling decisions depend on quality and availability of information
- Importance of resource-efficient information dissemination.

Motivation - Scheduling

- Scheduling on distributed, heterogeneous and dynamic Grid resources.
- Current Schedulers
  - Queuing or Batch:
    - NQE, PBS, LSF, Load Leveler
  - Application Level:
    - AppLeS, MARS, SEA, DOME
  - Dynamic, Ranking:
    - Condor ClassAd language and matchmaker

Motivation – Info Distribution

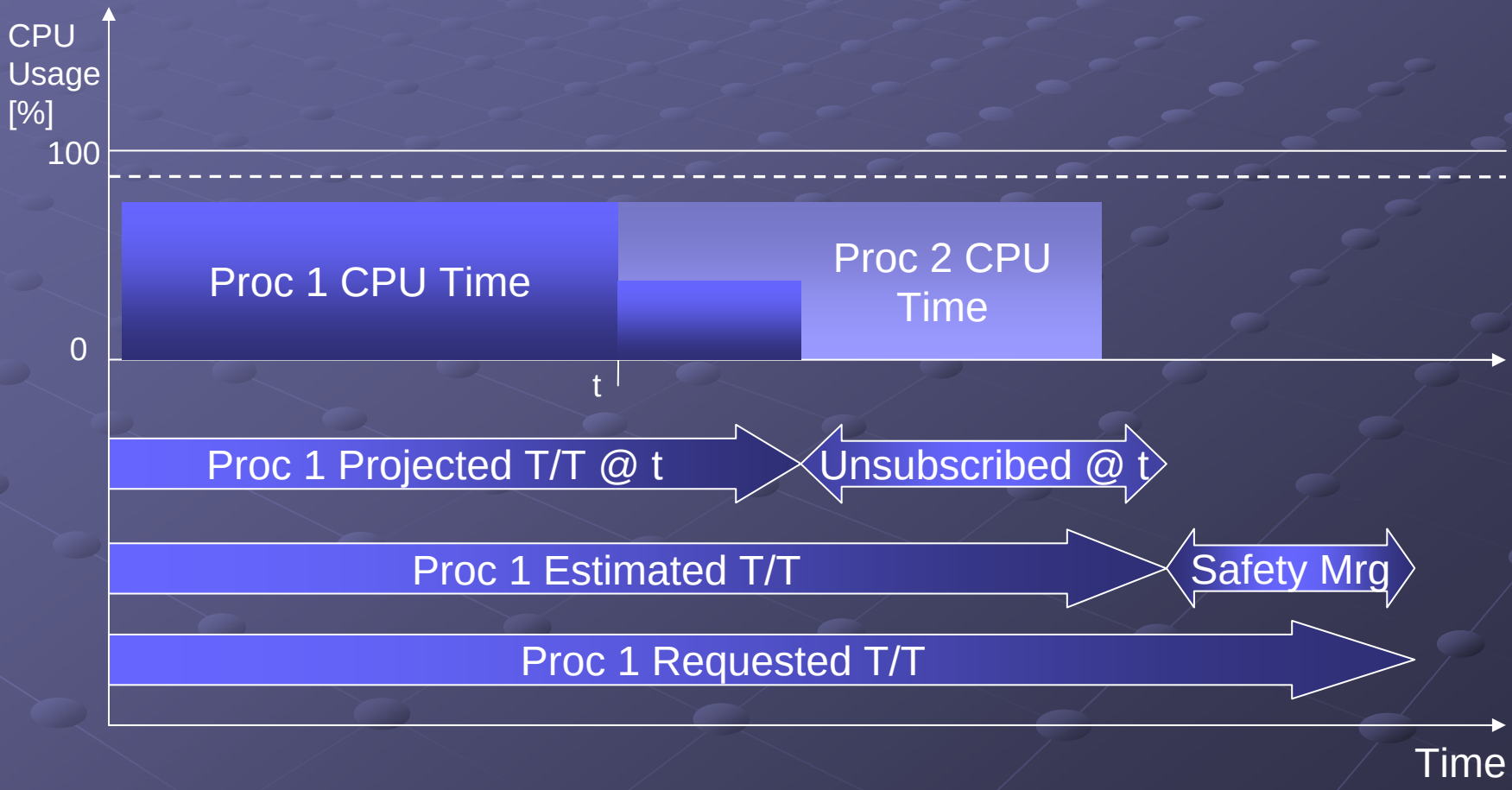
- Cu
  LD
- Li
  fo
- Ce
  bo
- SM



Bright Ideas - Scheduling

- Advance reservation and partitioning of resources complex and wasteful.
- Low-level scheduling in multitasking OS can distort machine loading info.
- Decouple application load and node computational output
- Assign jobs based on requested turnaround and unsubscribed capacity.

Subscribed Load Scheduling



Application & Node Profiles

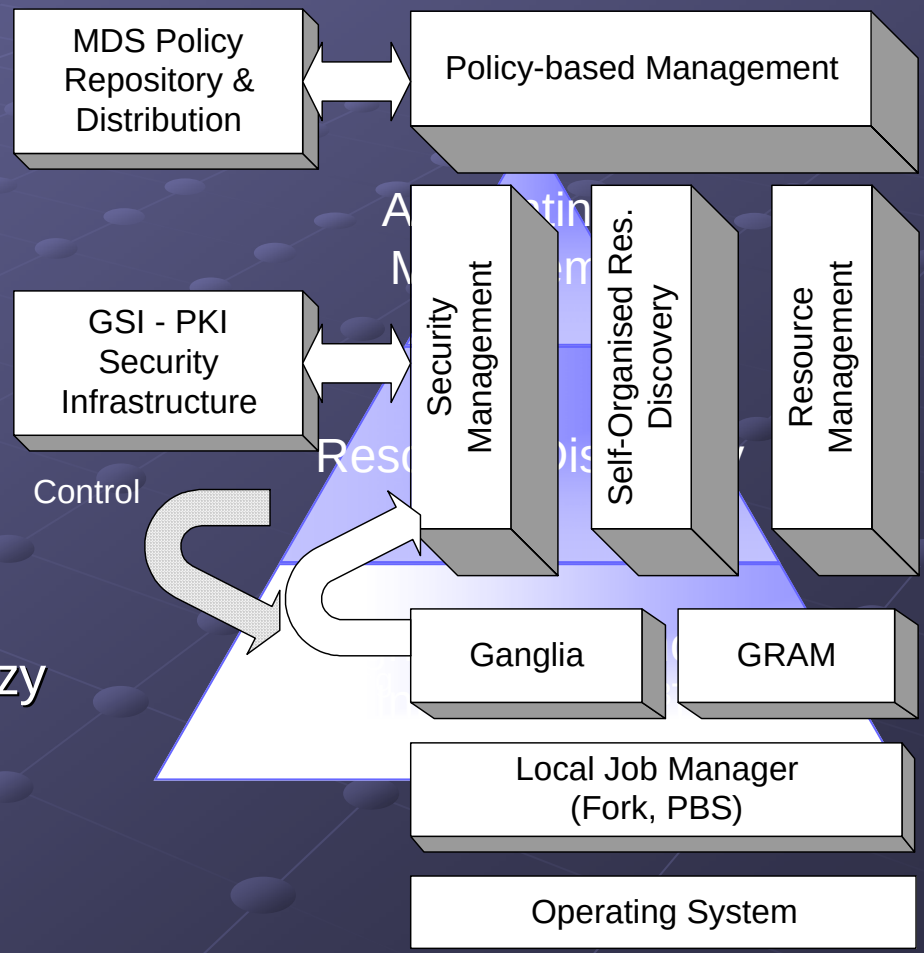
- Distinction between volatile and non-volatile resources.
- Profiles in XML with modular matchmaker.
- Nodes self asses the level of fitness for a given request and return a Bid Value.
- Monitoring and feedback improve confidence levels and reduce safety margins

Bright Ideas - Information

- Small-Worlds principle – information shared among several neighbours and few distant nodes.
- Fuzzy picture of the Grid environment – enables “good” but not necessarily “best” decisions.
- Gaining credibility, good resilience to random node failures

Information Flows

- Localised, need-to-know information flow policy
- 3-Tier Information Flow:
  - Node Current State
    - Low-latency, short shelf life
  - Volatile Resources State
    - Self-organized, distributed, fuzzy
  - Accounting
    - Centralized, reliable, accurate



Conclusions & Future Work

- New approaches needed to handle dynamic and heterogeneous resource pool.
- Reduce complexity and possible points of failure.
- Develop a prototype meta-scheduler and test on 200 CPU UCL Grid