Reinforcement learning for utility-based grid scheduling

J. Perez, B. Kégl, C. Germain-Renaud

Université Paris-Sud, LRI, TAO Team

November 29, 2007

Outline

Motivations

Previous approaches

Proposed methodology

Experiments

Perspectives

Motivations

- Need of a general and efficient method for **dynamically allocating grid resources** to optimize the satisfaction of both **end-users** and participating **institutions**.
- **Differentiated QoS** must be possible: Interactive and Batch

High level objectives driven scheduling

- **Overhead minimization**:

$$\frac{\textit{Time spent in the system} - \textit{Execution time}}{\textit{Execution time}}$$

- **Fairshare constraint**: difference between **allocated** resources, w_k, and actually **used** resources, S_k, for each group of users, k, also called Virtual Organizations (VO).

$$1 - \frac{\underset{k}{\operatorname{argmax}} (w_k - S_k)_+}{\underset{k}{\operatorname{argmax}} (w_k)}$$

Previous approches

- **Greedy policies**[1]: Unable to ensure trade-off between several objectives in the long term.
- **Queueing models**[2]: Complex queueing model may be required to obtain good performances in real grid systems that are dynamic and non-steady.

[1] E.D. Jensen, 1985, **A time driven scheduling model for real-time operating systems**

[2] R.Doyle and al, 2003,**Model-based resource provisioning in a web service utility**

Proposed methodology

- Scheduling considered as a **Continuous Markov Decision Process**.

- The goal is to find a **stationary policy** that chooses the action to take in each state which **maximizes** the long-term **expectation of utility**.
  - **State**: a set of real **variables measured in the grid**.
  - **Action**: each **job waiting** to be scheduled.
  - **Rewards**: **utility functions** that allow the users and the system administrators to **configure the priority between jobs**.

- Use of a feed-forward back propagated **neural net** to regress Q via **SARSA** algorithm.

Experiments

- **Synthetic case**, theoretical traffic simulation: 40 machines, 5000 jobs.
- **Realistic case**, simulated activity extracted from EGEE logs of April 2006: 110 machines, 5000 jobs.
- 2 types of utility function.

Time utility functions

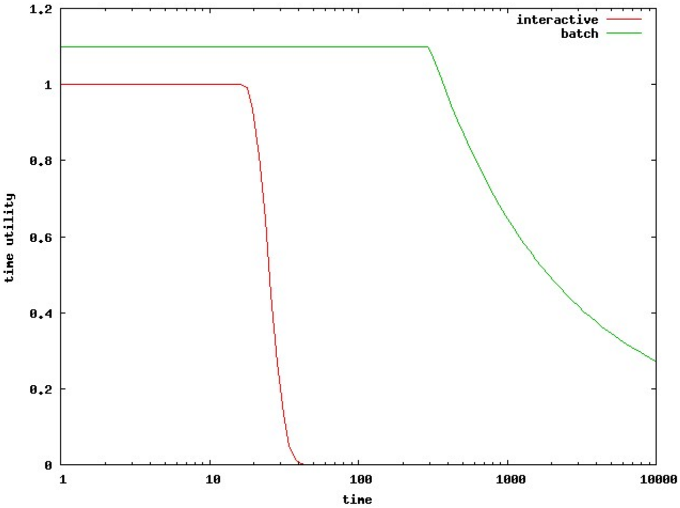
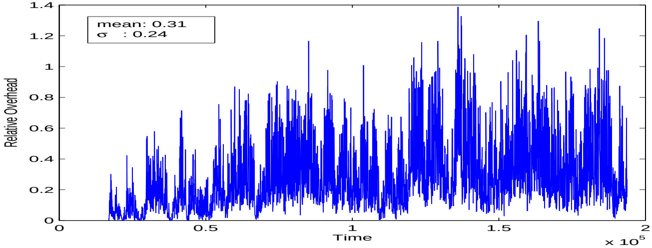


Figure: Example of Time Utility function used by the jobs

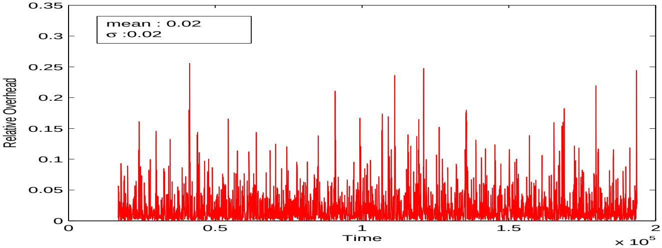
Synthetic case

- **Jobs**:
  - Poisson distribution of inter-arrival times.
  - Exponential distribution of service times.
  - 4 VOs (70%, 20%, 5%, 5%)
- **Utilities**: Fairshare and Overhead.
- **Methodologies**: RL and FIFO.

Synthetic case, details of overhead measures (1/2)

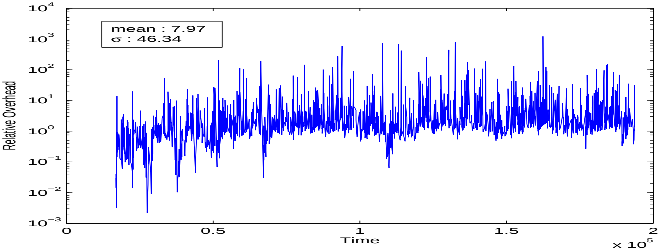


(a) FIFO batch jobs

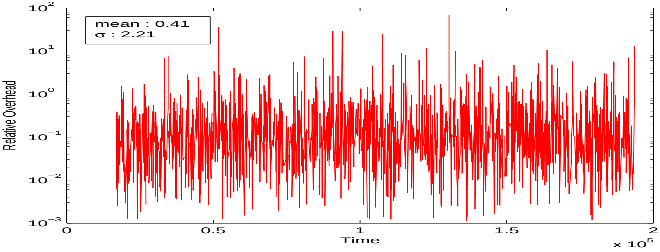


(b) RL-based batch jobs

Synthetic case, details of overhead measures (2/2)

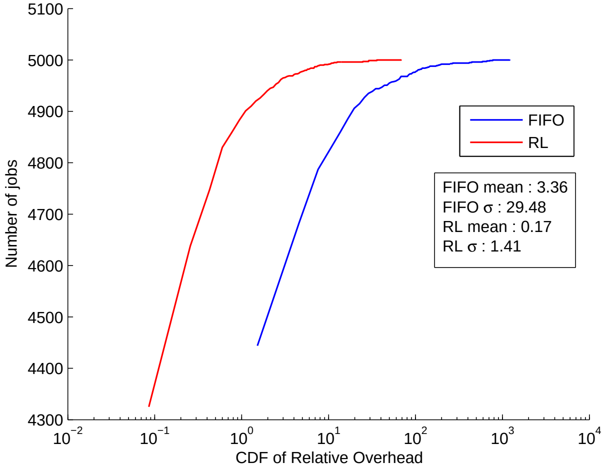


(c) FIFO interactive jobs

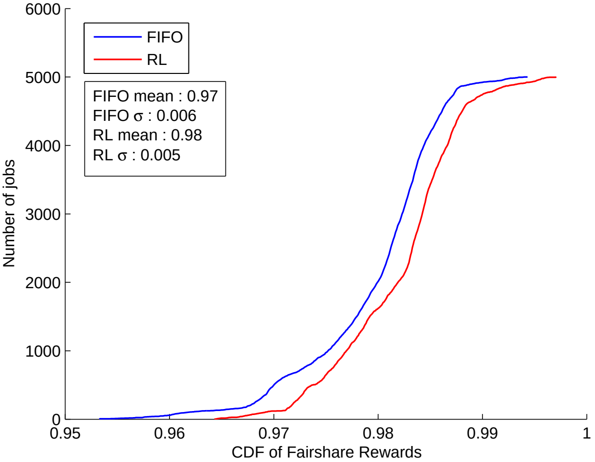


(d) RL-based interactive jobs

Summary of synthetic case, overhead



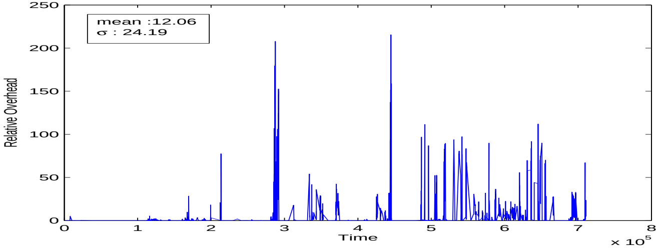
Summary of synthetic case, fairshare



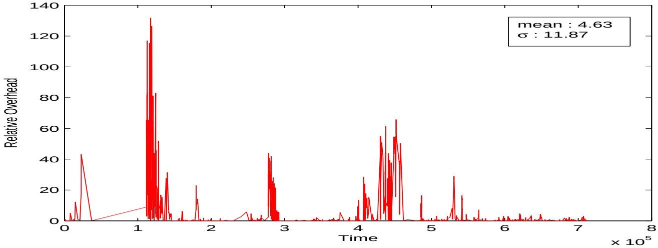
EGEE case

- **Jobs**: extracted from **Torque logs** of the **LAL**.
  - 7 VOs
  - Fairshare objective : (20%, 12%, 12%, 6%, 6%, 9%, 35%)
  - VO distribution : (72%, 7%, 5%, 2% , 1% , 4%, 9%)
- **Utilities**: Fairshare and Overhead.
- **Schedulers**: RL and EGEE's gLite.

EGEE case, details of overhead measures (1/2)

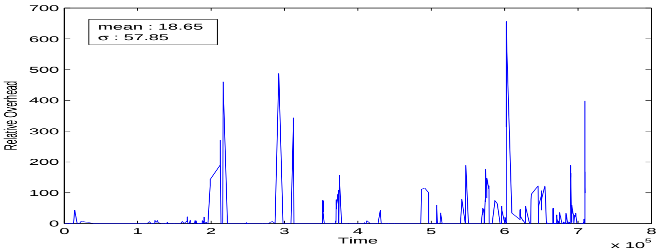


(e) EGEE batch jobs

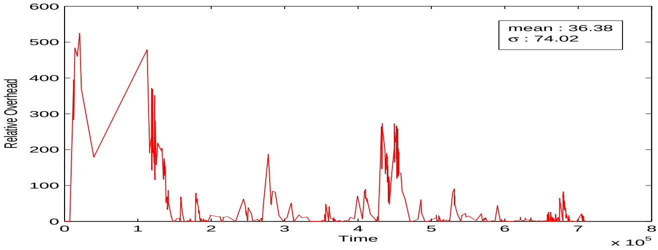


(f) RL-based batch jobs

EGEE case, details of overhead measures (2/2)

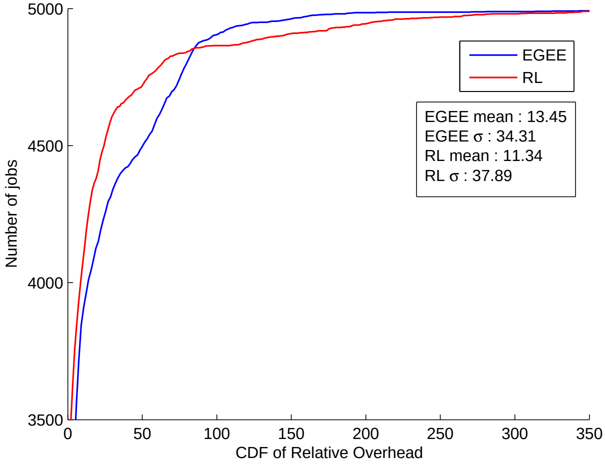


(g) EGEE interactive jobs

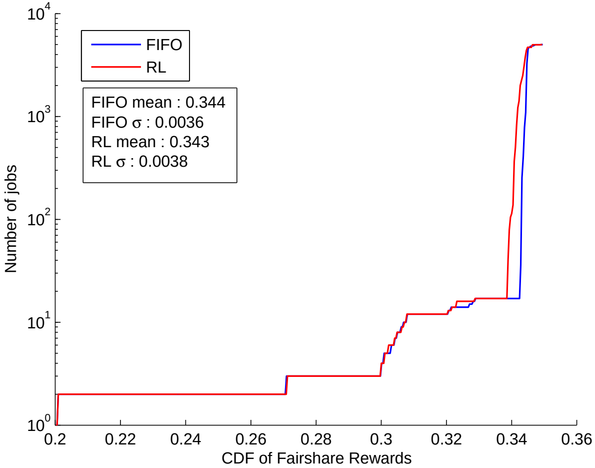


(h) RL-based interactive jobs

Summary of EGEE case, overhead



Summary of EGEE case, fairshare



Perspectives

Grid perspectives

- Improving grid state description
- Implementation in a grid infrastructure

Learning perspectives

- New generalization algorithms:
  - Deep Belief Network
  - Echo State Machine
- Multi-objective reinforcement learning
- Distributed reinforcement learning