A decentralized utility-based grid scheduling algorithm

João Luís Vazão Vasques

Dissertation to obtain the Master Degree in Communication Networks Engineering

Agenda

- Introduction
- Proposed Solution
- Implementation details
- Evaluation
- Conclusions
- Future work

Introduction

Motivation

- Shortcomings of current solutions
  - Centralized or hierarchical architectures [Buyya et al. 2000,Frey et al. 2002, Xhafa et al. 2010]
  - Some do not consider *QoS* or utility [Izakian et al. 2009,Yun-Han et al. 2011]
  - Jobs' requirements lack f exible characterization [Chauhan 2010,XiaoShan et al. 2003]
  - Non-f exible scheduling policies [Amudha et al. 2011, Chunlin et al. 2007]

Introduction

Problem statement

- Is it possible to devise a decentralized scheduling architecture where the scheduling decisions take into account:
  - the grid resources and the user's requirements, in a flexible way,
  - in order to improve the system's performance and user satisfaction ?

Introduction

Thesis goals

- New decentralized utility-driven scheduling algorithm that provides partial requirement fulfillment based on user's requirements

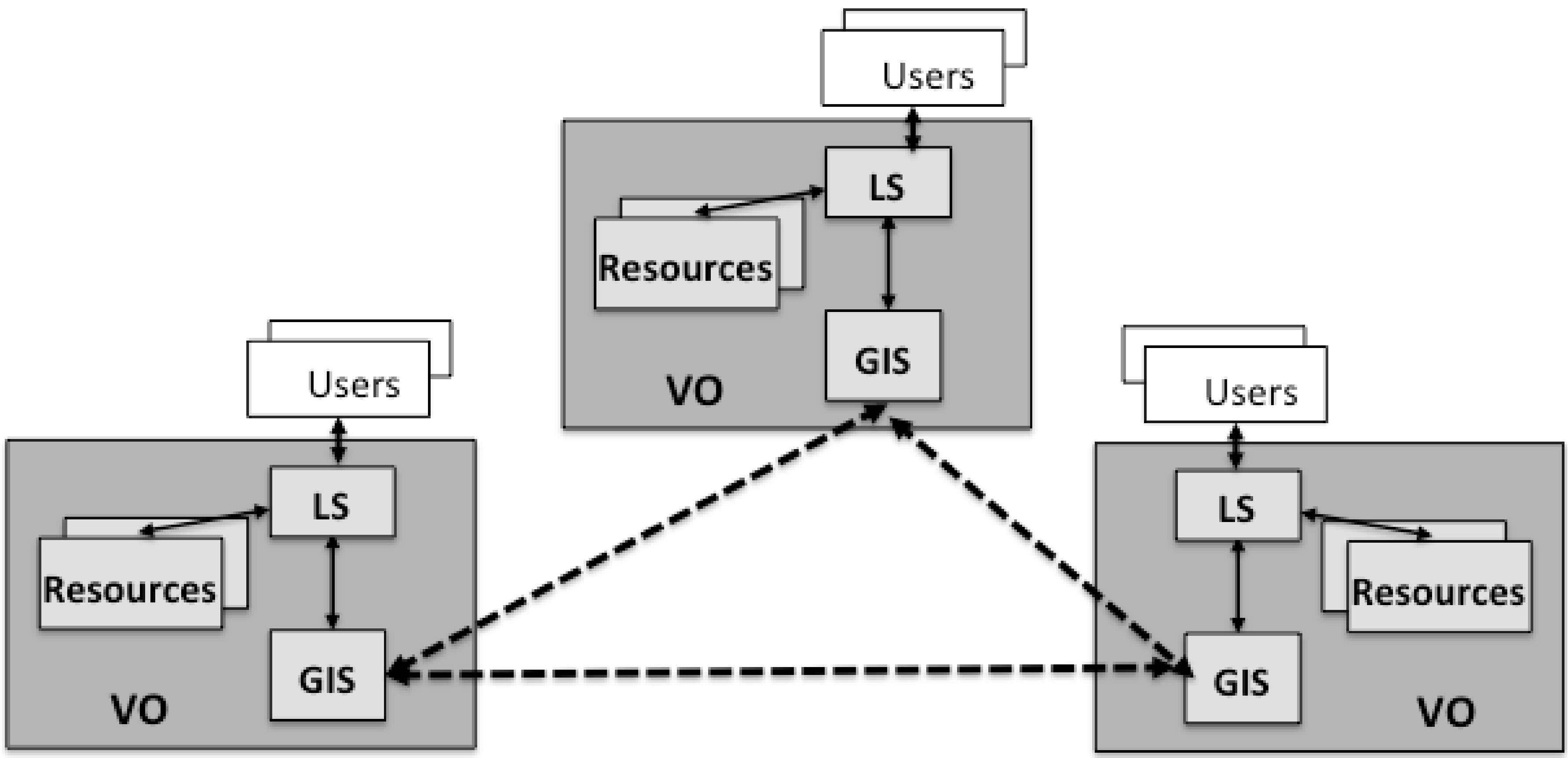
Introduction

Contributions

- New decentralized utility scheduling algorithm that provides partial utility fulfillment
- Extensive set of extensions to *GridSim* to support decentralized architectures
- A new resource allocation policy in *GridSim*
- Modifications to *GridSim* to support multiple schedulers
- Validation of the proposed solution

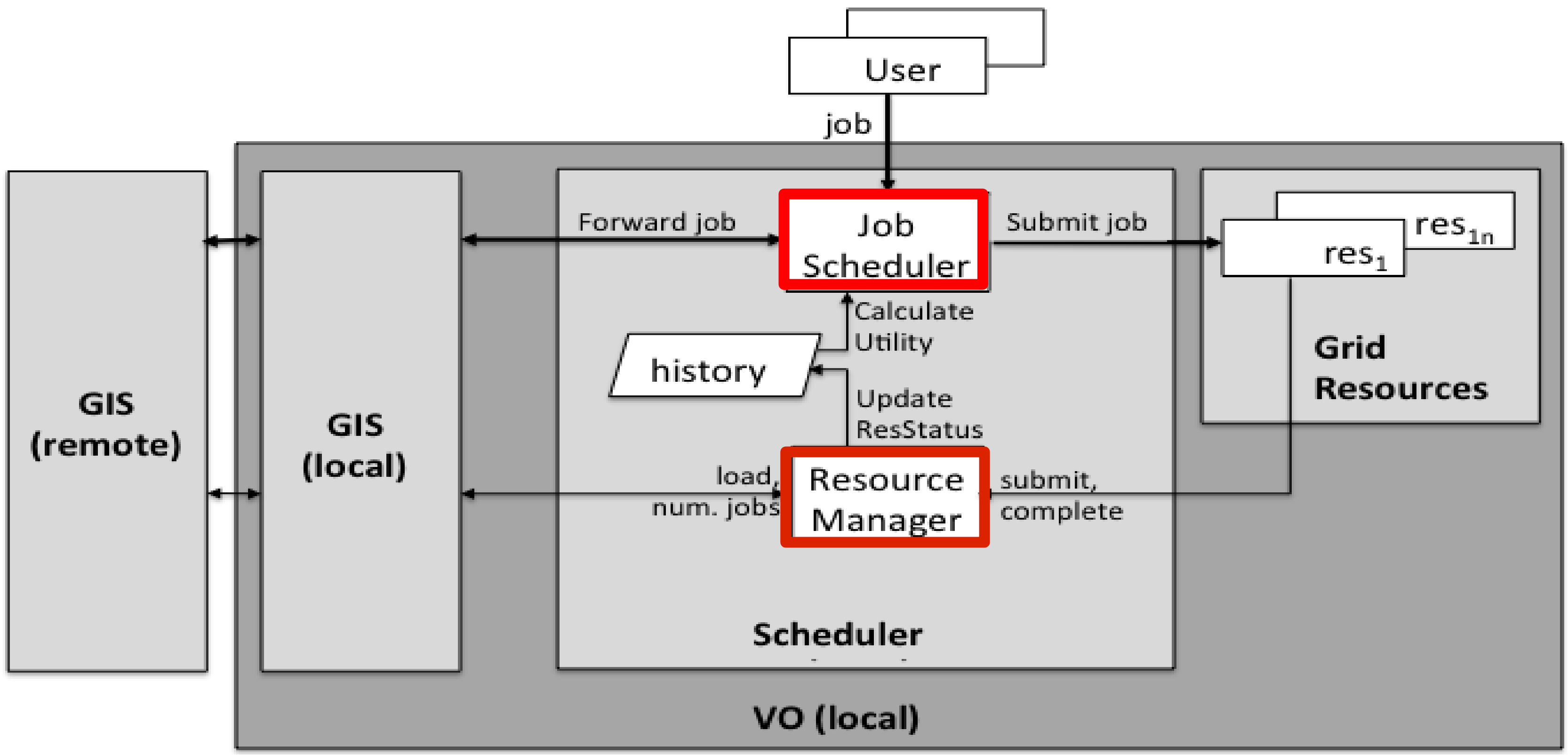
Proposed solution

Grid Organization



Proposed solution

Virtual Organization architecture



Proposed solution

Utility function

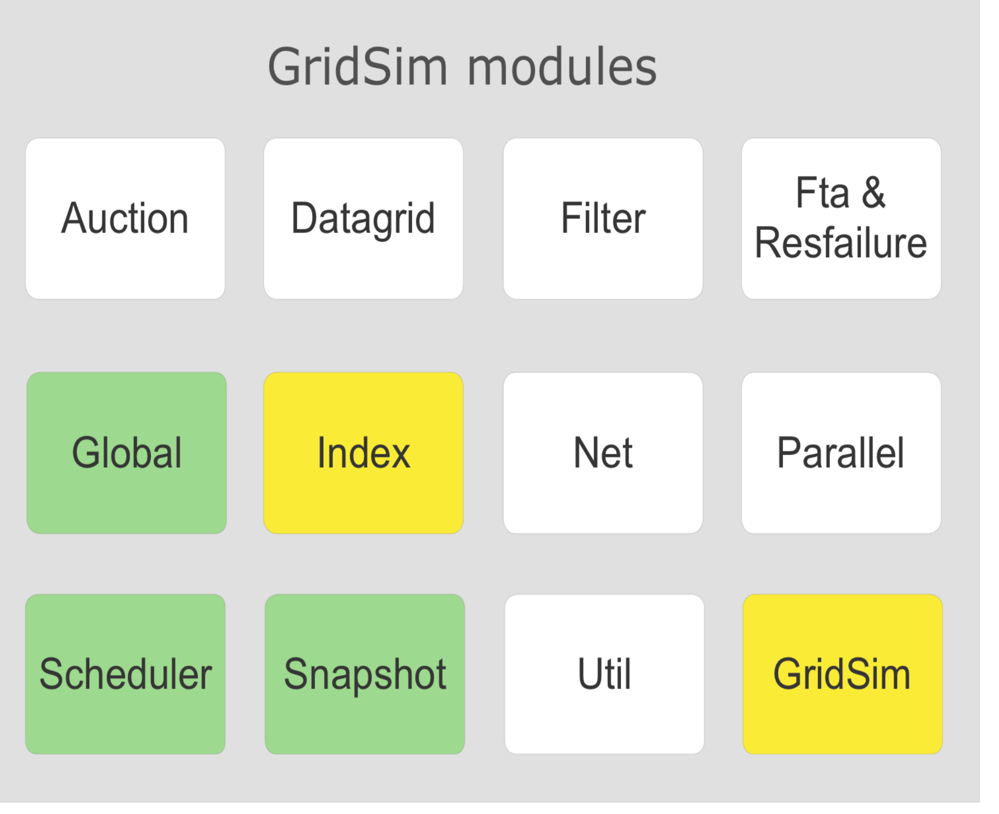
- Calculates the global utility value of a resource
  - N – number of requirements
  - α – set of utility values of requirement's, *req*$_i$, *options*, *opt*$_i$
  - β – weight assigned to each requirement

$$res\_util = \frac{\sum_{i=1}^{N} max(\alpha_{(i)} * \beta(i))}{N}$$

Implementation

GridSim modules – additions and extensions

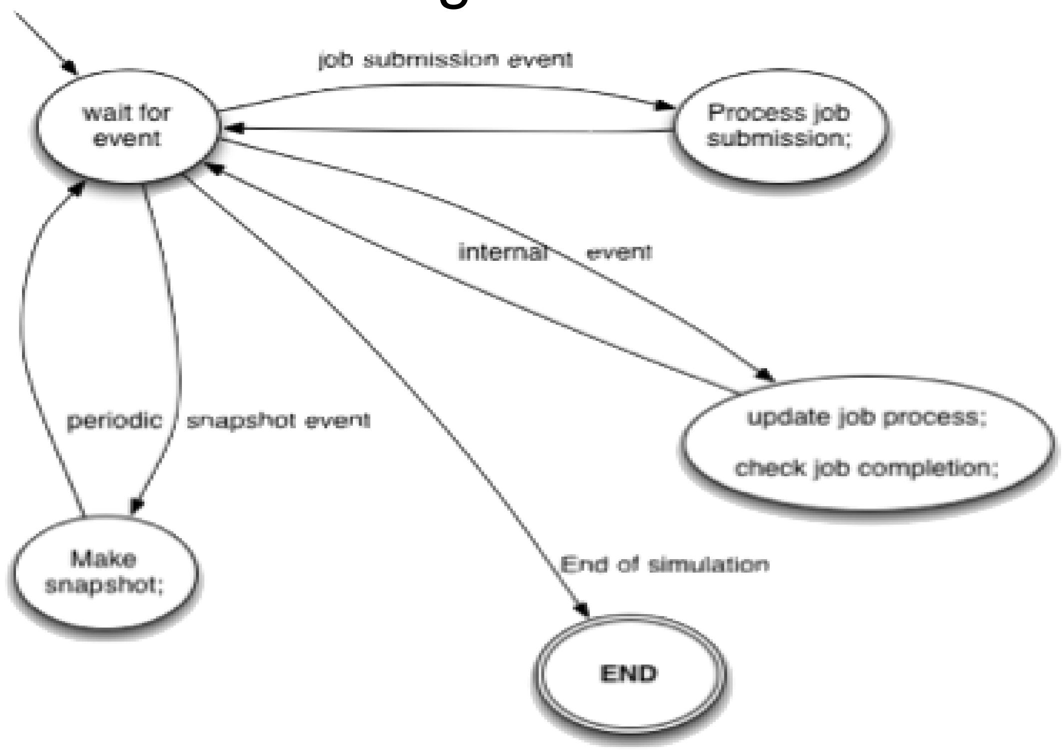
- New modules
  - Scheduler
  - Snapshot
  - Global
- Extensions
  - Index
  - GridSim



Implementation

Utility Allocation Policy

- A Processing Element (PE) can process multiple jobs in parallel
- Jobs have priorities according to their time constrains



Evaluation

Goals

- Goal I: Scheduling algorithms comparison
- Goal II: Decentralized architecture validation

Scenario

- Variable number of: VOs, resources, users and jobs (per user)
- Resource characterization: architecture, OS, PE speed, Number of PEs
- Job's requirements: architecture, OS, maximum execution time
- Job's generation properties: size and inter-departure rate

Evaluation

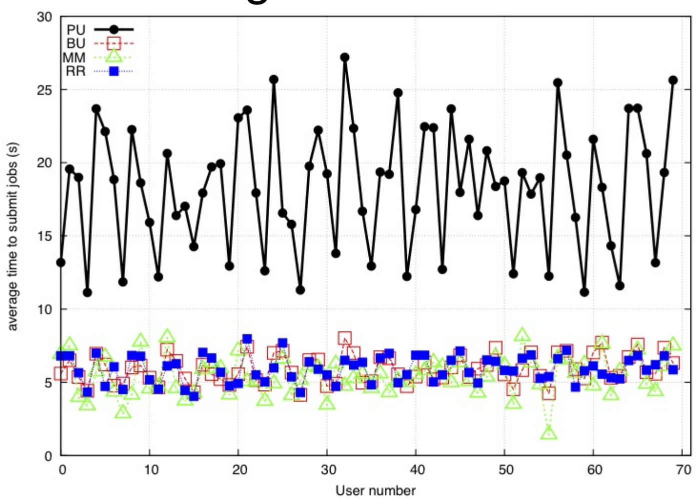
Goal I: Scheduling algorithms comparison

- **Scheduling algorithms comparison**
  - PU – Partial Utility (proposed solution)
  - BU – Binary Utility (strict fulf llment)
  - MM – Matchmaking (full requirement match)
  - RR – Round Robin
- **Tests**
  - Variable number of VOs
  - Variable number of jobs per user
  - Variable job inter-departure time

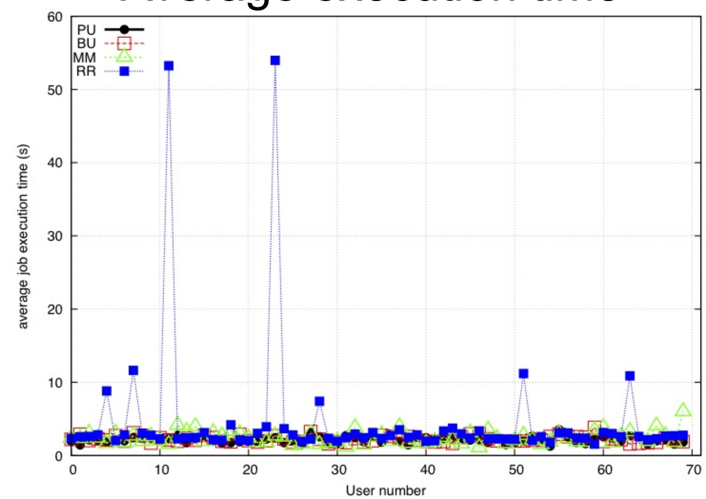
Evaluation

Goal I: Variable number of jobs/user

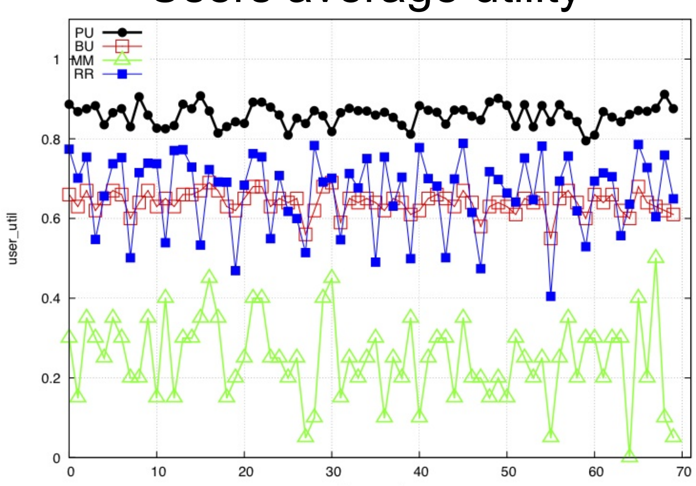
Average time to submit



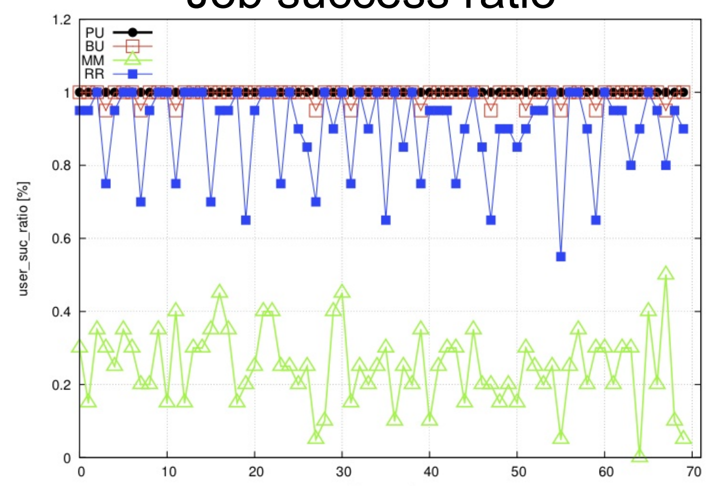
Average execution time



Users average utility

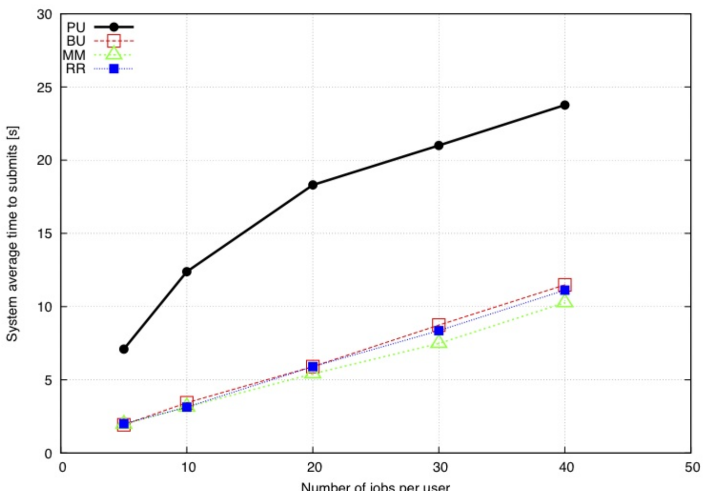


Job success ratio

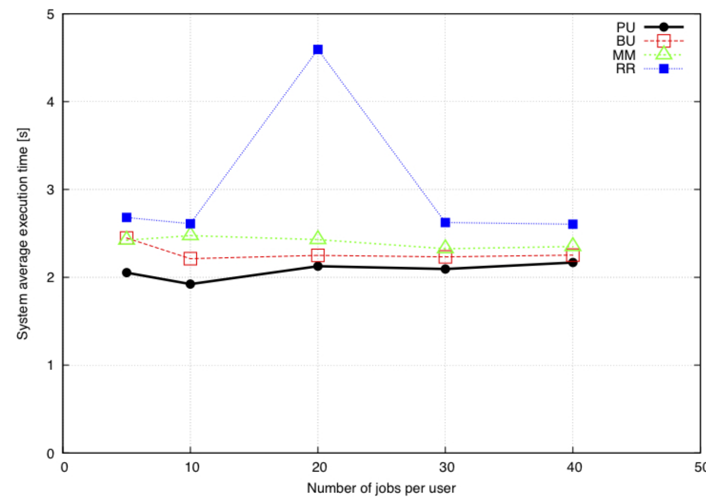


Goal I: Variable number of jobs/user - comparison

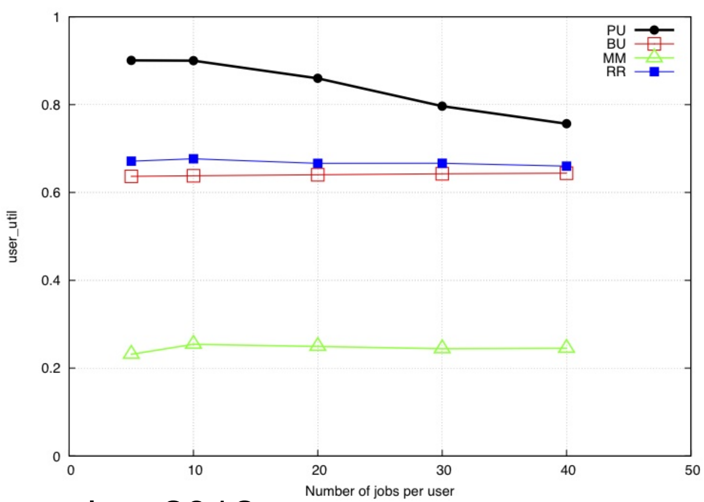
Average time to submit



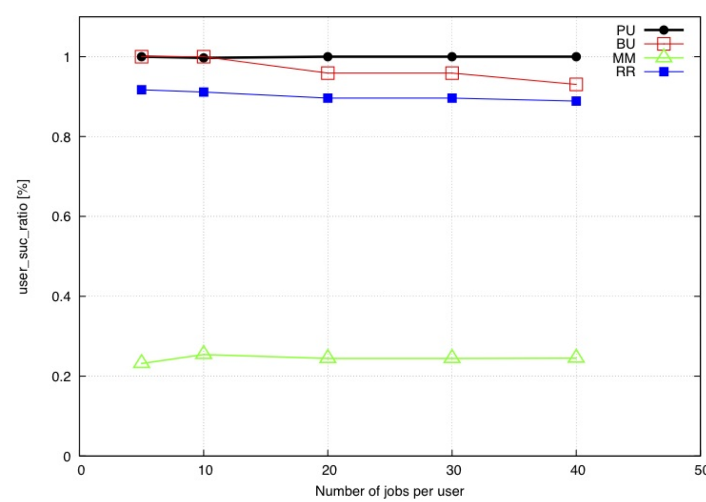
Average execution time



Users average utility



Job success ratio



Conclusions

- Decentralized scheduling architecture
- Partial Utility scheduling
- Extension of the *GridSim* simulator
- Better performance when compared with other algorithms

Future work

- Prioritization of the requirements
- Support for a more dynamic environment
- Test the solution in a real grid scenario (Globus) or on an emulator (PlanetLab)
- Propose the solution to a cloud environment

Publications

- The work in this dissertation is partially described in:
  - J. Vasques and L. Veiga, *A Decentralized Utility-based Grid Scheduling Algorithm*, accepted to ACM SAC 2013, 28th Symposium On Applied Computing, ACM.

References

1.Buyya, R., D. Abramson, & J. Giddy (2000). Nimrod/G: an Architecture for a Resource Management and Scheduling System in a Global Computational Grid. Proceedings Fourth International Conference/Exhibition on High Performance Computing in the Asia-Pacif c Region, 283–289 vol.1.

2.James Frey, Todd Tannenbaum, Miron Livny, Ian Foster, and Steven Tuecke. 2002. Condor-G: A Computation Management Agent for Multi-Institutional Grids. Cluster Computing 5, 3 (July 2002)

3.Xhafa, F. & A. Abraham (2010, April). Computational models and heuristic methods for Grid scheduling problems. Future Generation Computer Systems 26(4), 608–621

4.Izakian, H., A. Abraham, & V. Snasel (2009, April). Comparison of heuristics for scheduling independent tasks on heterogeneous distributed environments. In Computational Sciences and Optimization, 2009. CSO 2009. International Joint Conference on, Volume 1, pp. 8 –12

5.Yun-Han Lee, Seiven Leu, and Ruay-Shiung Chang. 2011. Improving job scheduling algorithms in a grid environment. Future Gener. Comput. Syst. 27, 8 (October 2011), 991-998

6.Chauhan, S. & R. Joshi (2010, February.). A weighted mean time min-min max-min selective scheduling strategy for independent tasks on grid. In Advance Computing Conference (IACC), 2010 IEEE 2nd International

7.XiaoShan He, XianHe Sun, and Gregor von Laszewski. 2003. QoS guided min-min heuristic for grid task scheduling. J. Comput. Sci. Technol. 18, 4 (July 2003), 442-451

8.Amudha, T. & T. Dhivyaprabha (2011, June). Qos priority based scheduling algorithm and proposed framework for task scheduling in a grid environment. In Recent Trends in Information Technology (ICRTIT), 2011 International Conference on, pp. 650 –655

9.Chunlin, L. & L. Layuan (2007). An optimization approach for decentralized qos-based scheduling based on utility and pricing in grid computing. Concurrency Computation Practice And Experience 19(1), 107–128

Thank you for your attention.

Questions?