Achieving Better Performance Through True Best Effort in Scavenging Grid Computing

Raphael de Aquino Gomes
Fábio Moreira Costa
Fouad Joseph Georges

Institute of Informatics
Federal University of Goiás, Brazil

September, 2008

Outline

1. Introduction
2. Related Work
3. InteGrade Overview
4. Our Approach
5. Experiments
6. Conclusion

Introduction

Grid environment

- Heterogeneous components (software and hardware)
- Extremely dynamic
  - Can exists unexpected join/exit of nodes
  - Their topology is not fixed and therefore cannot be predefined
- No-dedicated resources (this is truer for scavenging grid)
- Usually **best-effort**
  - Same priority for all users
  - Absence of guarantees

Introduction
Typical actions and their problems

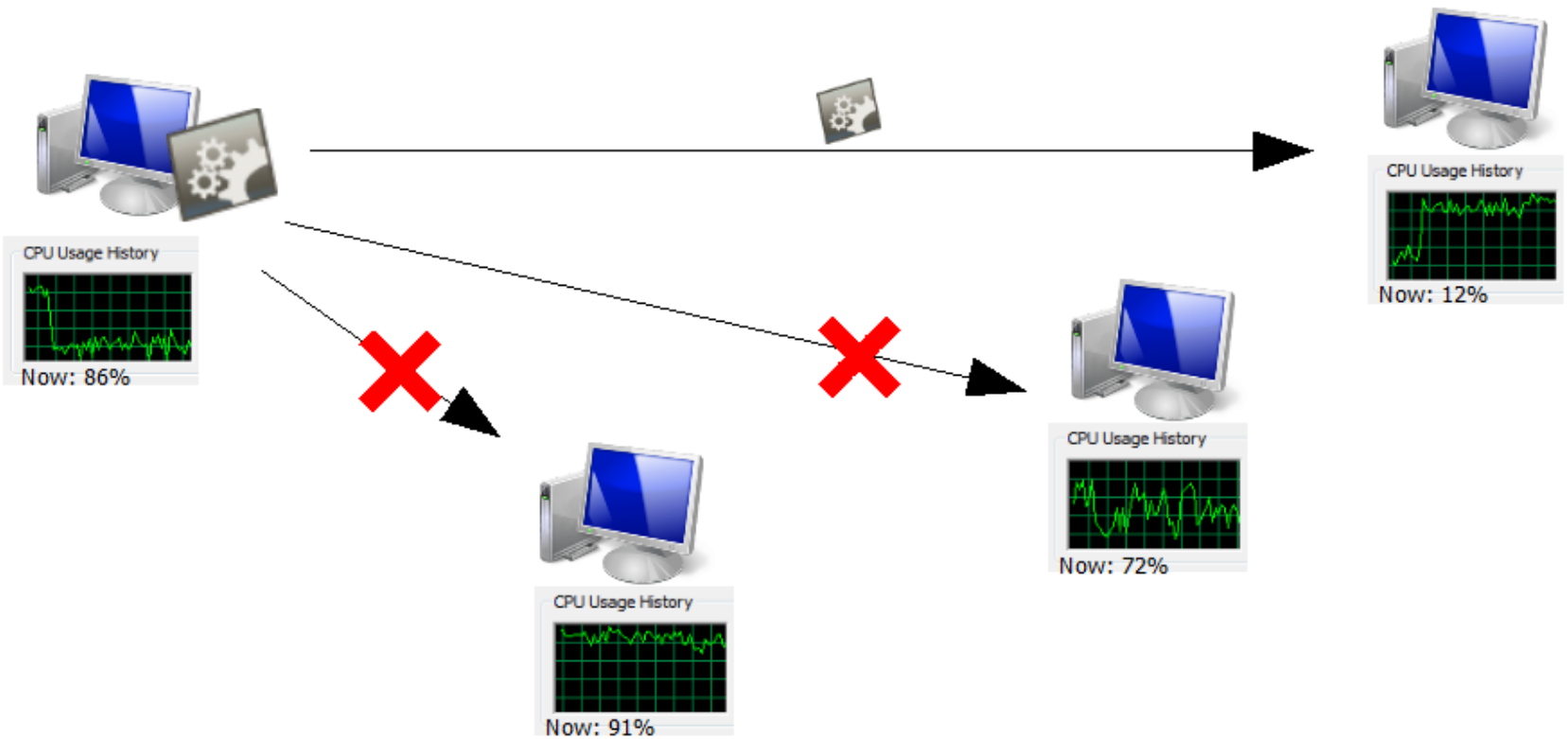
To act upon task schedules where tuning is applied through the identification of behavior patterns

- Pattern anomalies: Resource utilization diverges from the statistical values
- The arithmetic functions used to estimate the reference behavior often rule out sporadic bursts.
- Non-determinism: no guarantees that resources will stay idle

To use task migration in lack of resources. Problems:

- Constant migration affects application execution performance
- There are costs: data transfer and rollback of the tasks
- Can cause migration chaining (Migrating a task back and forth two nodes may happen)
- Lack of extra resources

Introduction
Typical actions and their problems



Introduction
Our Approach

1. Analysis of the execution pattern of local applications. Wishes:
   - Enhances the migration process of grid tasks
   - Improves the performance through **real best-effort**
2. Use of dynamic adaptation techniques as an alternative to migration
3. Research on grid middleware: InteGrade[1]

[1] http://www.integrade.org.br

Related Work

"Pre-execution" operations

- Choi *et al* analyzes the influence of the heterogeneity of grid nodes on the performance
- Scheduling algorithms that take into account the properties of volunteer nodes
- Profiling for the whole node (not for individual applications)

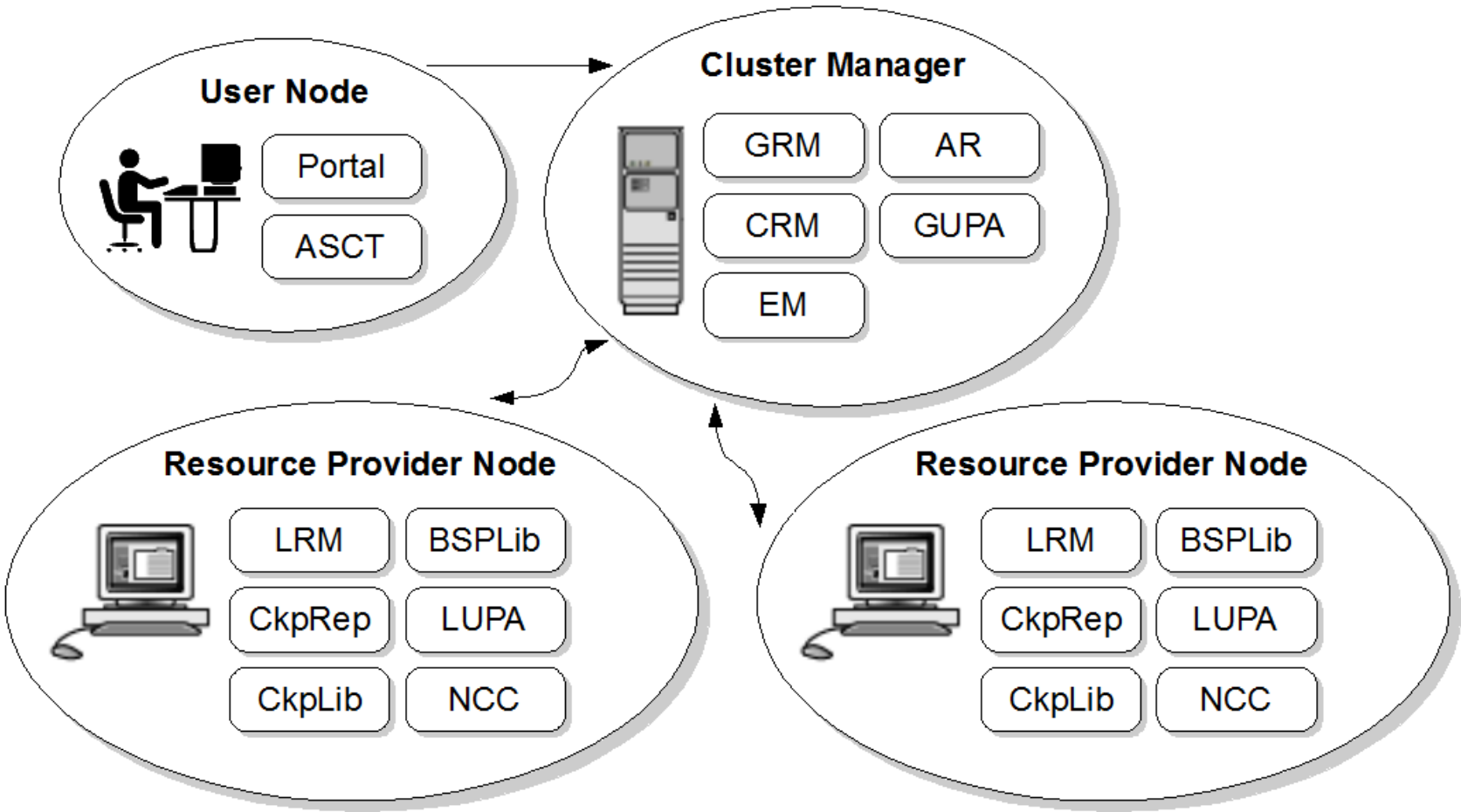
Anomaly-aware prediction

- Grid Harvest Service (GHS): parallel application scheduling system
- The system is monitored aiming identify anomalies about the identified patterns. However the only operation is scheduling
- We hope to get better results through a wider analysis and taking alternatives actions, such as adaptation

InteGrade Overview

- Grid middleware that uses the idle computing power of personal computers
- Object-oriented architecture (CORBA standard)
- Sequential, parameter sweeping, and parallel (BSP and MPI) applications
- Usage pattern collection and analysis
- Preserves resource provider's QoS at all costs
  - Grid users' QoS: currently not a concern

InteGrade Overview

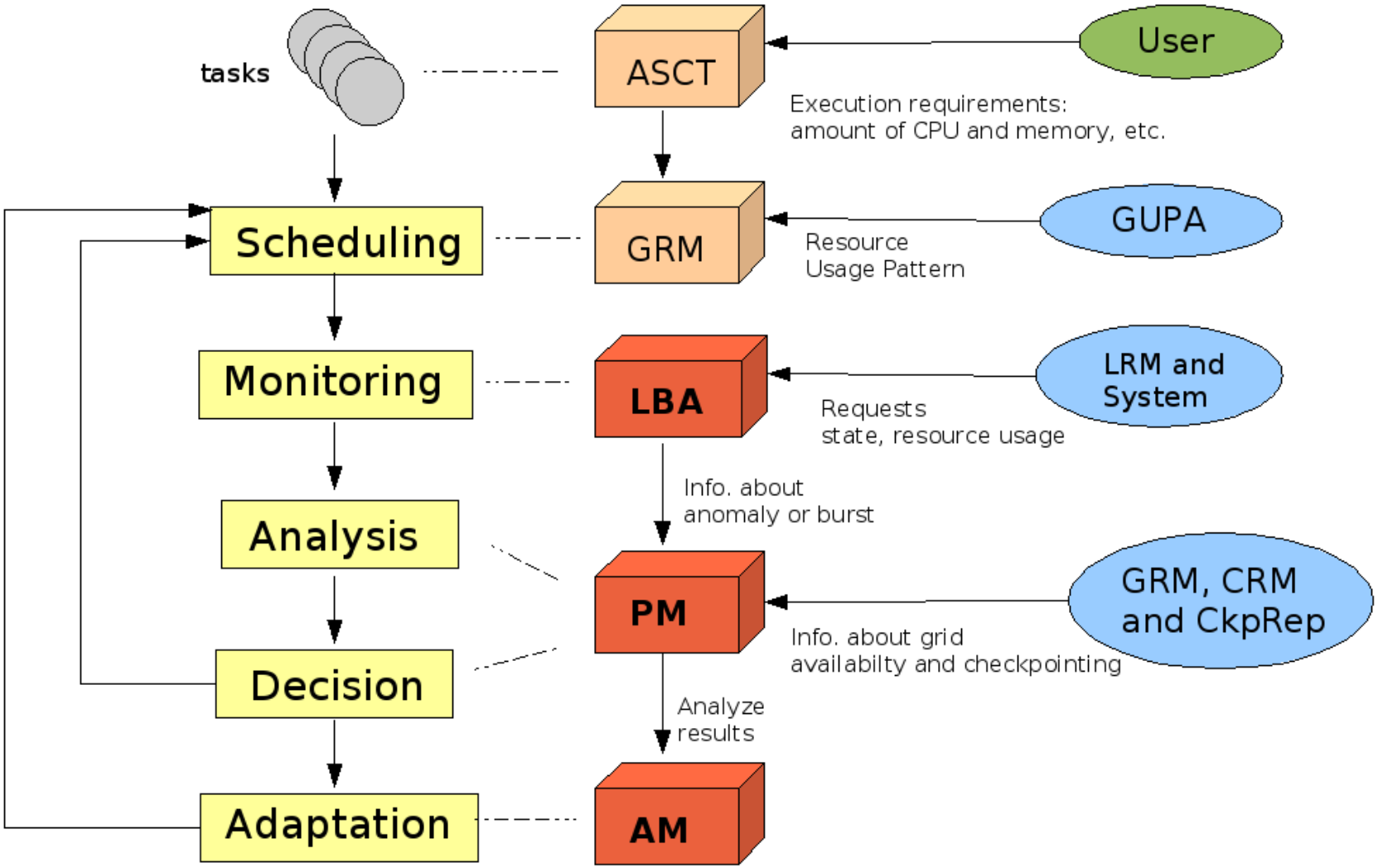


Our Approach
Achieving True Best-Effort

LUPA and GUPA optimize the performance but they have some limitations:

- Are based on standard/usual resource behaviour
  - Do not capture anomalous behavior (which deviates from the standard)
- Information about resource usage bursts is eliminated

Our Approach
Architecture

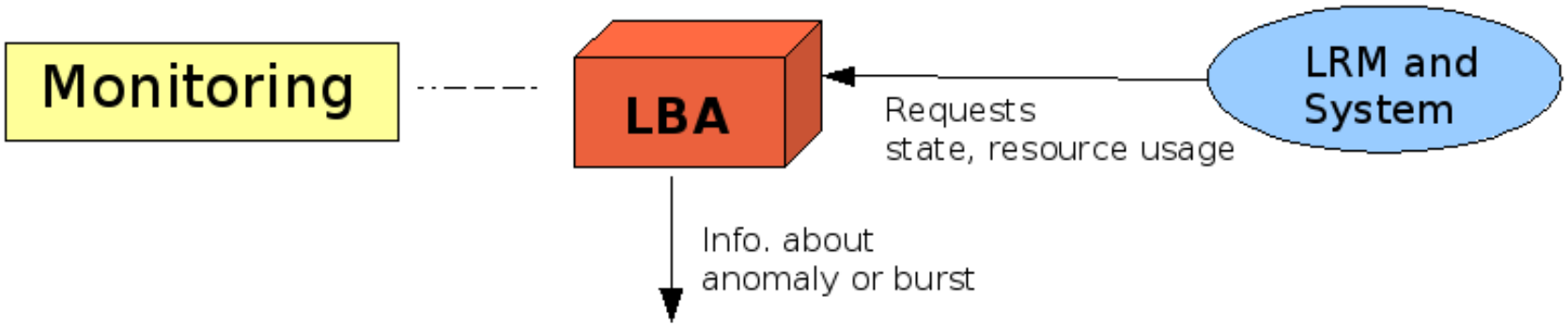


Our Approach
Architecture

LBA (Local usage pattern anomaly and Burst Analiser)

Goal: Estimate resource usage (during a burst) and identify anomalous behavior

- It must be able to run on the presence of resource scarcity
- Requirements: simplicity and high priority

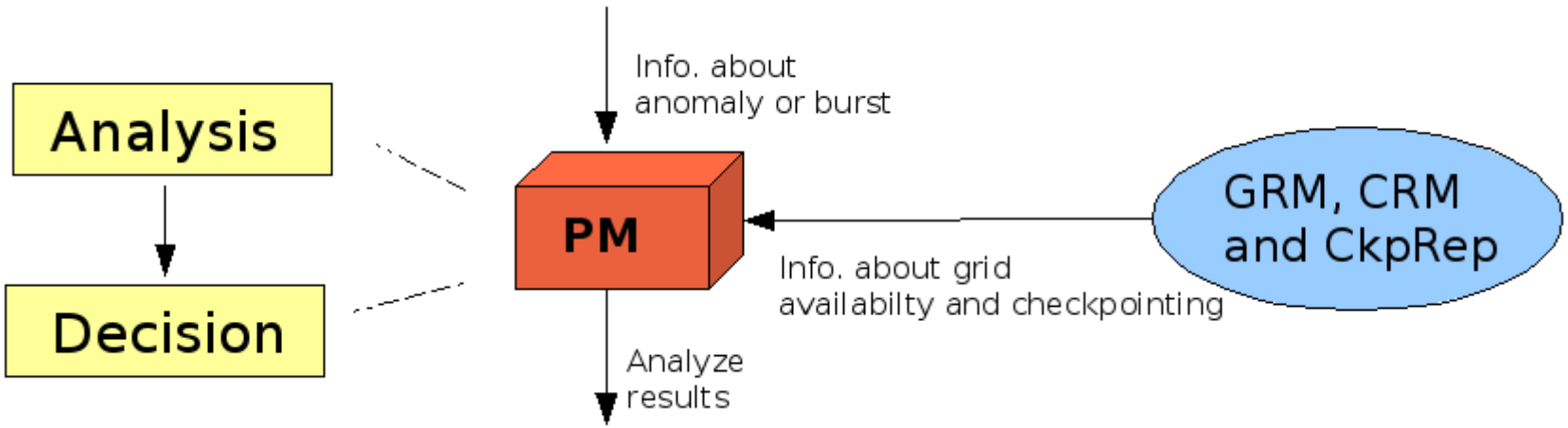


Our Approach
Architecture

PM (Performance Manager)

Goal: Decide the best strategy – migration or adaptation (can be both)

Migration is not the best solution for all case!

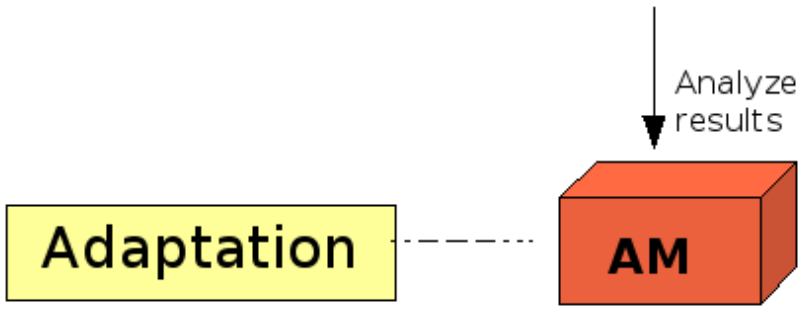


Our Approach
Architecture

AM (Adaptation Manager)

Goal: Implement adaptive operations as an alternative (or a complement) to migration

- The adaptation is performed on the node where the task is running (without migration) or on the grid middleware components

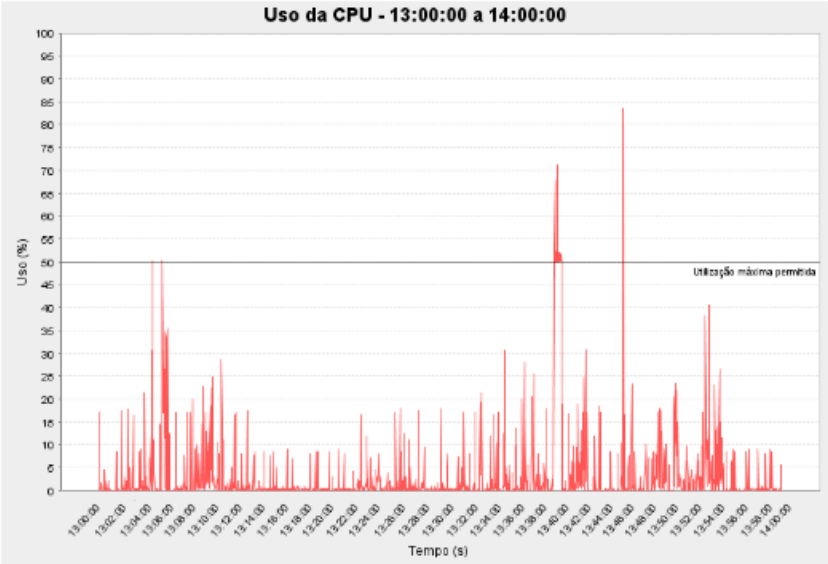


Experiments

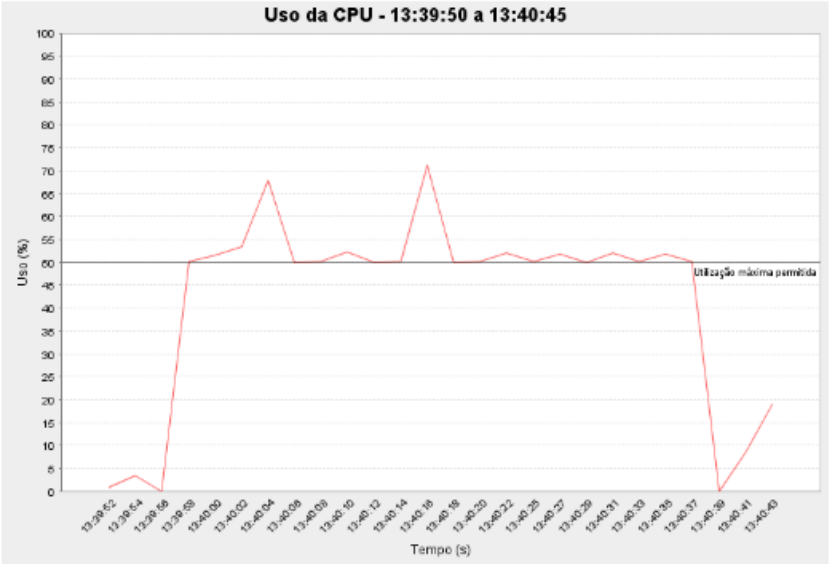
There is no implementation yet to alow a thorough evaluation of our approach. But we've carried out experiments that demonstrate the existence of the problem

- Testbed: DELL Optiplex 745, Intel Core 2 Duo 2.2GHz, 4 GB
- We collected CPU and memory usage measurements at intervals of 1 second
  - CPU data extracted from */proc/stat*
  - Memory data obtained with the *free* command

Experiments



(a)



(b)

CPU usage - average level (13:00-14:00)

2.98 %

Standard deviation (13:00-14:00)

7.98 %

Usage average between 13:39:00 and 13:44:00 (taken at intervals of 5 min)

8.43 %

13:39:58 a 13:40:38

> 50 % (40 sec)

Conclusion

- We proposed to investigate a more efficient task recovery strategy to improve grid applications performance
- Another contribution is the inclusion of adaptive methods in the InteGrade middleware – allows the dynamic adaptation of grid services aiming to enhance global performance
- This work is in progress – implementation of the architecture
  - Implementation status: LBA prototype nearly completed

References

Choi, S.
Group-based Adaptive Scheduling Mechanism in Desktop Grid
Department of Computer Science and Engineering Graduate School – Korea University, 2007.

Kon. F.; Goldman, A.; Finger, M. and Bezerra, G.C.
InteGrade: Object-oriented grid middleware leveraging idle computing power of desktop machines
Concurrency and Computation: Practice & Experience, 2004.

Wu, M. and Sun, X.H.
A general self-adaptive task scheduling system for non-dedicated heterogeneous computing
Cluster Computing, 2003. Proceedings. 2003 IEEE International Conference on, 2003.

Questions

Thank You!

For more information:
raphael@inf.ufg.br