VI HPCLATAM - Latin American Symposium on High Performance Computing

Heterogeneous Resource Allocation in the OurGrid Middleware: A Greedy Approach

Miguel Da Silva   Sergio Nesmachnow

Centro de Cálculo, Facultad de Ingeniería
Universidad de la República, Uruguay

Mendoza, Argentina, 2013

Contents

1. Introduction: The OurGrid Middleware and motivation
2. Traditional resource allocation in OurGrid
3. The Greedy scheduling approach
4. Experimental analysis
5. Conclusions and future work

Introduction

The OurGrid Middleware

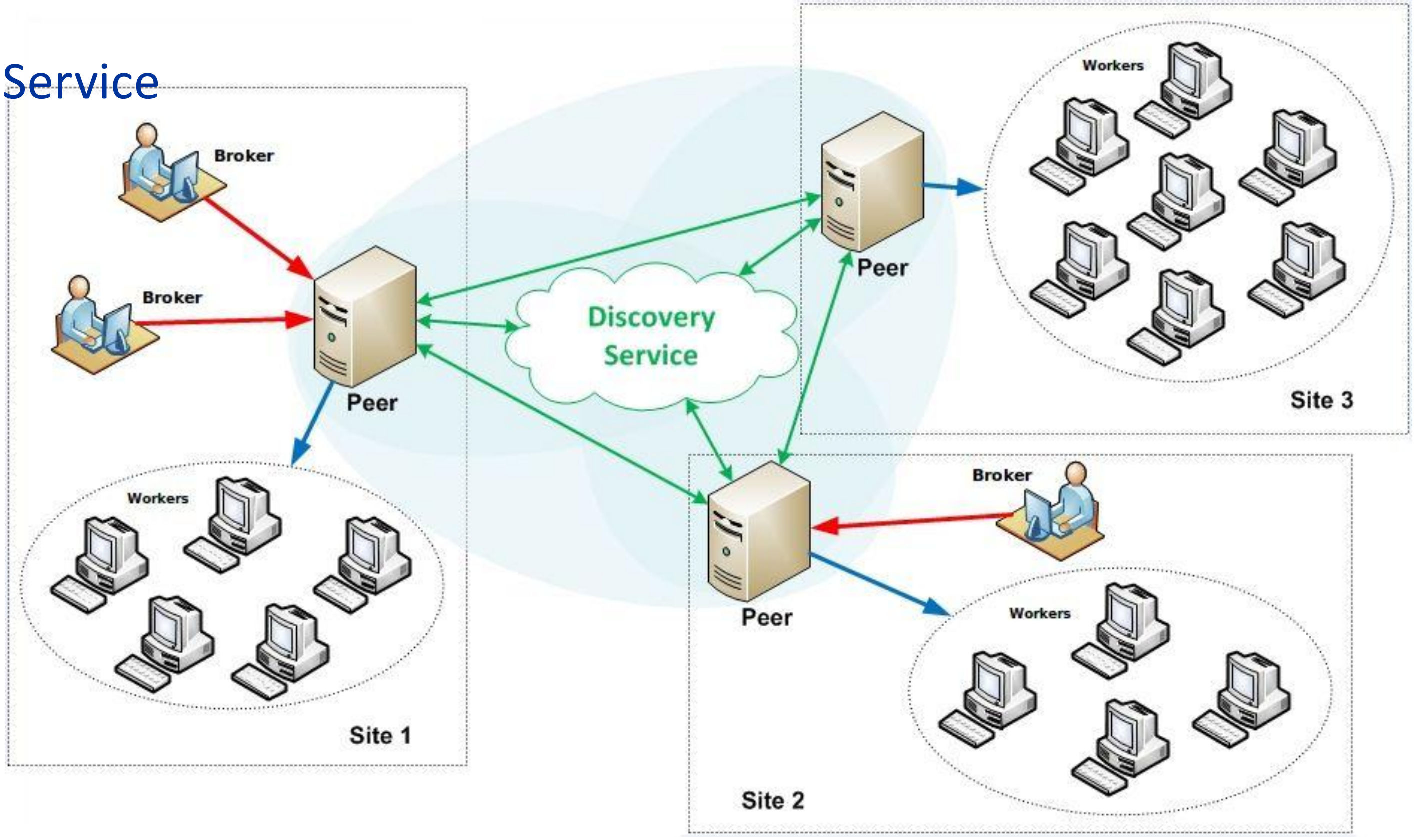
- OurGrid: open source middleware that enables the creation of peer-to-peer grids and volunteer-computing platforms
- Developed by researchers from *Universidade Federal de Campina Grande* (UFCG)
- Support applications following the Bag-of-Tasks (BoT) model; typically arises in grid and volunteer-based computing infrastructures
- Communications using the *eXtensible Messaging and Presence Protocol* (XMPP)
  - Internet-friendly, simple and efficient protocol
- Federations can be created easily and many sites can use the same XMPP server

Introduction

The OurGrid Middleware

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Four components for a regular OurGrid site:
  1. Peer
  2. Broker
  3. Worker
  4. Discovery Service



Introduction

The OurGrid Middleware

- All the components are integrated in a transparent way to the user; allowing the grid to provide a single-image of an infrastructure with a large computing power
- For experimentation, the OurSim simulator is available
  - Discrete-events simulator that implements a *virtual grid site* similar as deployed using the middleware
  - Realistic workloads (synthetic applications) can be used as input
  - Infrastructure can be described using data from different sources

OurSim is not a metascheduler; the whole infrastructure of a grid site is simulated. For each component of a regular site there will be an entity to simulate its behavior.

Introduction

Motivation

- OurGrid schedules tasks in a *round-robin* fashion; the *first* available node complying users specifications is taken
- This scheduling policy may not assign the most *suitable* resources, especially for heterogeneous environments
- Is it possible to change the scheduling policy in order to maximize/minimize some specific criterion?
- How is load balancing affected by the scheduling policy?

The main purpose of this work is to provide OurGrid with new allocation methods, and support for heterogeneity. A simple approach, yet very used and powerful, is chosen: a *Greedy* resource scheduler. The criteria is *processing time*.

Resource allocation in OurGrid

Traditional vs greedy scheduler

- Traditional scheduler: does not provide support for heterogeneuos environments
  - Resources within each site are implicitly ordered according to the time the corresponding peer registers them
  - OurGrid uses a *Network of Favours* to encourage the resource contribution; this feature prioritizes sites requesting the same resource
- Greedy scheduler: *dynamic priority scheduling* algorithm, provides support for heterogeneuos environments
  - Resources are sorted according to processing capabilities
  - OurGrid paradigm is not changed: the Network of Favours is still used to rank sites
  - Support for volunteer-computing: resource contribution is encouraged

The Greedy scheduling approach

Implementation details

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Implementation is done by changing specific Java classes; the peer is aware of more information about the workers and functions according to the defined metric
- Modifications:
  - Method *takeNeededWorkers* totally rewritten
  - Classes *AllocationHelper*, *SamePriorityAllocationHelper*, *LowerPriorityAllocationHelper* modified
- Changes to OurGrid/OurSim code are now available within the official distribution

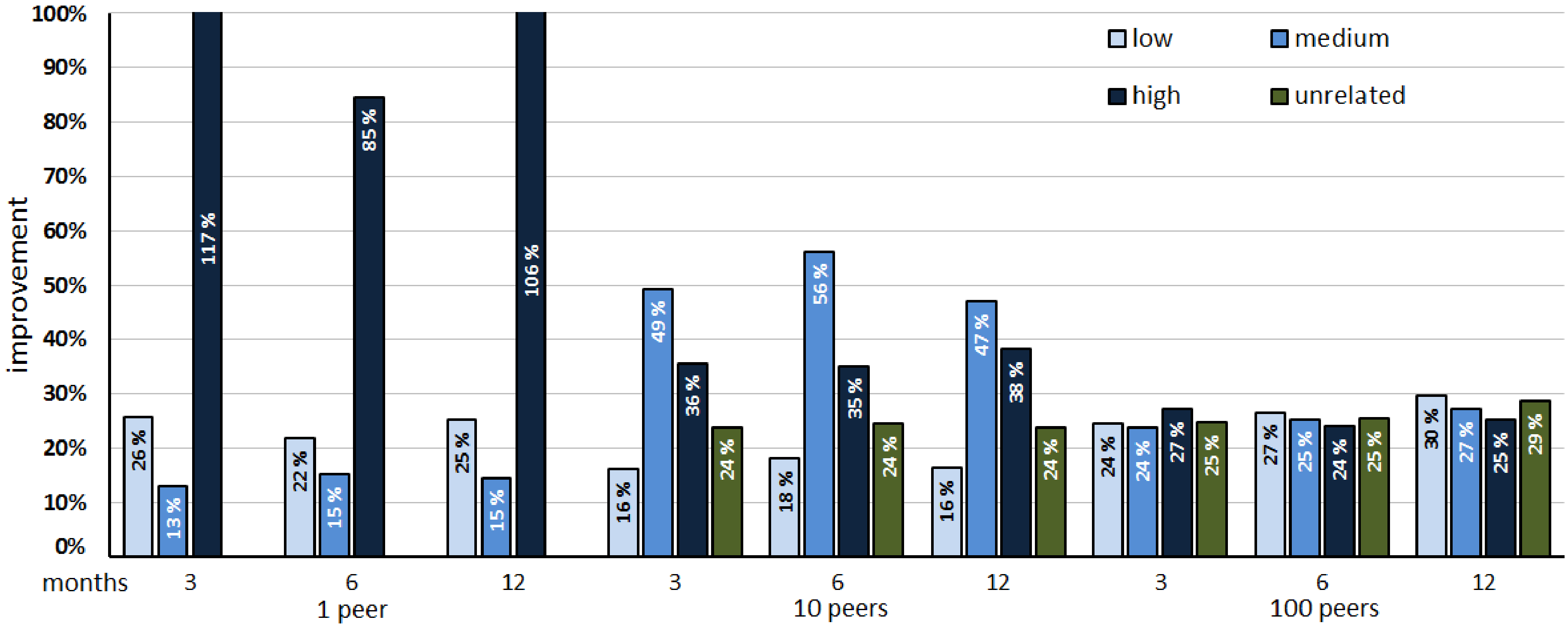
Experimental analysis

Results and discussion: Grid scenarios

- The greedy approach was tested using OurGrid simulated infrastructures (OurSim simulator)
- Methodology: the analysis was carried out for both *related machines* (low/medium/high heterogeneity) and *unrelated machines* models
- Resources within each site and the corresponding heterogeneity levels are defined based on the SSJ SPEC benchmark results
- Site dimensions: 1, 10, and 100 peers are considered
- A total of 30 grid instances per scenario were simulated and the number of workers per site are 8, 16, 32 or 64 (uniform distribution)
- Workloads: 30 instances representing 3, 6 and 12 months are created
- TOTAL: 390 grid scenarios and 90 workload instances in the experimental evaluation

Experimental analysis

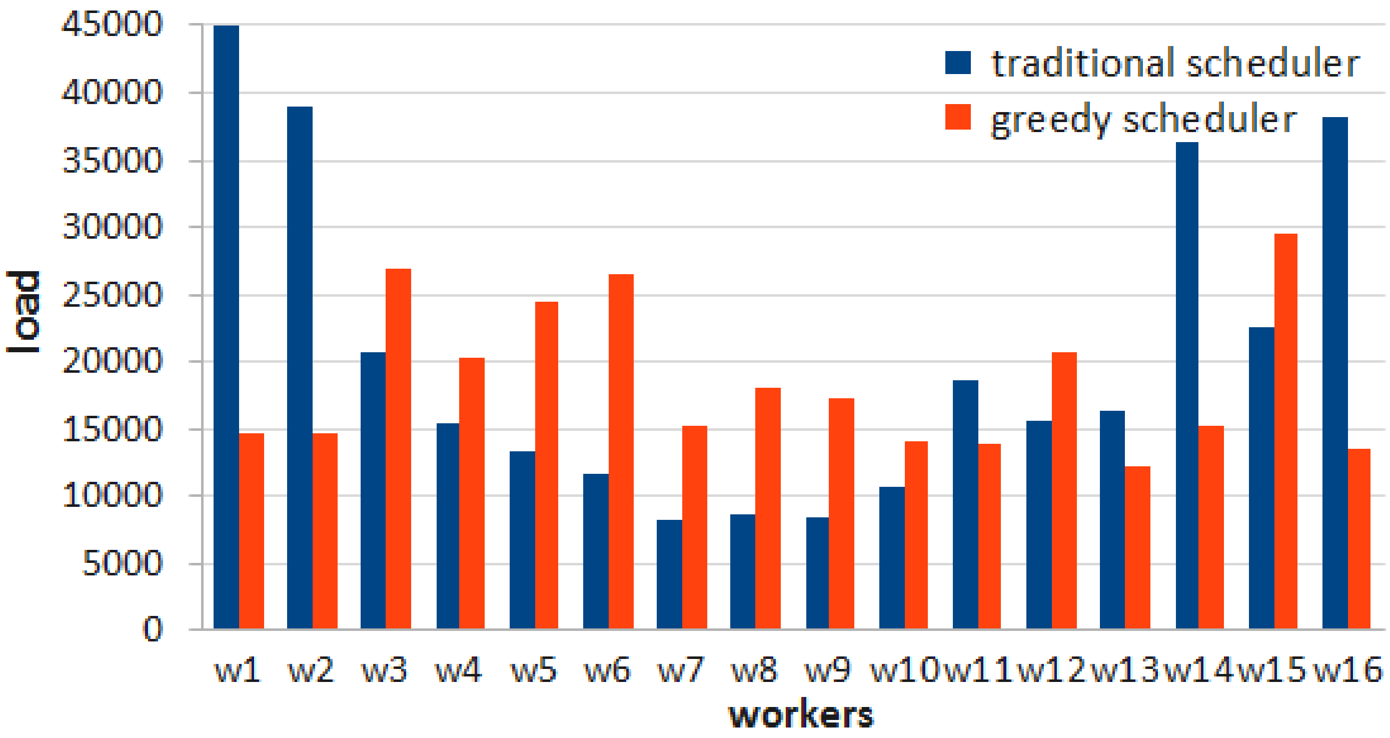
Results and discussion: greedy scheduling policy



- The greedy scheduler outperformed the traditional mechanism; just in a few cases the traditional scheduler obtained shorter (<2.5%) times
- As the number of peers increases, results stabilize (due to increasing number of network communication) near **25-30%** improvement

Experimental analysis

Results and discussion: load balancing



load balancing: representative execution

- The greedy scheduling method distributes the tasks over the workers of a specific peer in a more balanced way
- This behavior is observed in all grid instances and workloads of different sizes

Conclusions and future work

Main contribution: new scheduling method

- The experimental results demonstrate that the greedy scheduler is an effective method for reducing overall execution time of BoT jobs
- Load balancing is also improved
- Main contribution: the proposed greedy scheduling is now available within the official distribution of the OurGrid/OurSim code

- Future works:
  - New scheduling policies based on different criterion (e.g., node reliability and energy consumption)
  - Improvements for the methods of remote resources selection
  - Additional scalability studies regarding the numbers of brokers per site for both *static* and *dynamic* scenarios

THANKS FOR YOUR ATTENTION

FACULTAD DE INGENIERÍA, UNIVERSIDAD DE LA REPÚBLICA, URUGUAY