

A Computing Resource Discovery Mechanism Over a P2P Tree Topology

D. Castellà H. Blanco F. Giné F. Solsona

Distributed Computing Group
University of Lleida
Spain



VECPAR 2010

BERKELEY (CALIFORNIA)

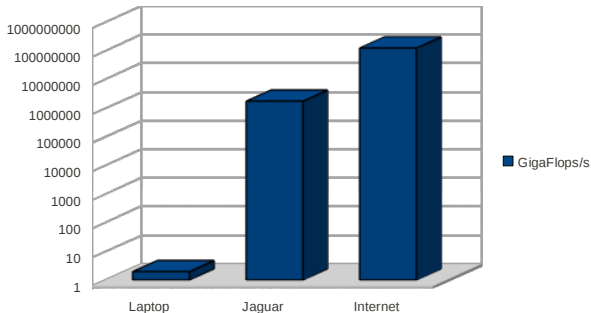
- 1 Introduction
- 2 System model
- 3 Rebalancing algorithm
- 4 Resource Discovery Service
- 5 Experimental results
- 6 Conclusions and future work

Outline

- 1 Introduction
- 2 System model
- 3 Rebalancing algorithm
- 4 Resource Discovery Service
- 5 Experimental results
- 6 Conclusions and future work

Starting point

Distributed computing



Main goal

To design and implement a P2P distributed computing system.

Main functionalities of CoDiP2P

Previous Work

CodiP2P features:

- 1 Connecting algorithm.
- 2 Updating algorithm.
- 3 Peer removal algorithm.
- 4 Job scheduling algorithm.



Main functionalities of CoDiP2P

Proposals

Our proposal in this presentation:

- 1 Rebalancing algorithm.
- 2 Searching algorithm.



Outline

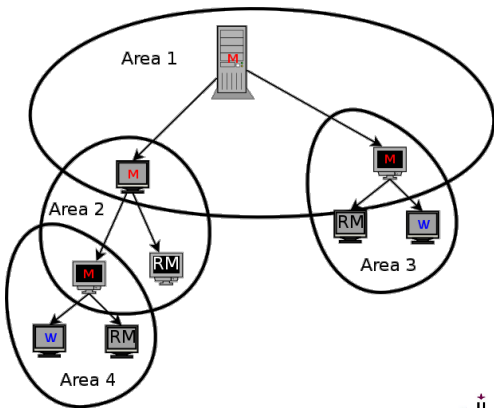
- 1 Introduction
- 2 System model**
- 3 Rebalancing algorithm
- 4 Resource Discovery Service
- 5 Experimental results
- 6 Conclusions and future work

System model

Miscellaneous

- Area of peers (A_j).
- Peer Roles:
 - Manager (M_j)
 - Worker (W_j)
 - Replicated Manager (RM_j)

Tree-Like Topology



Outline

- 1 Introduction
- 2 System model
- 3 Rebalancing algorithm**
- 4 Resource Discovery Service
- 5 Experimental results
- 6 Conclusions and future work

System restructuring

Rebalancing algorithm Introduction

Causes of having unbalanced tree:

The churn of peers can unbalance the system:

- Massive input of peers into the system.
- Massive output of peers from the system.

Solution

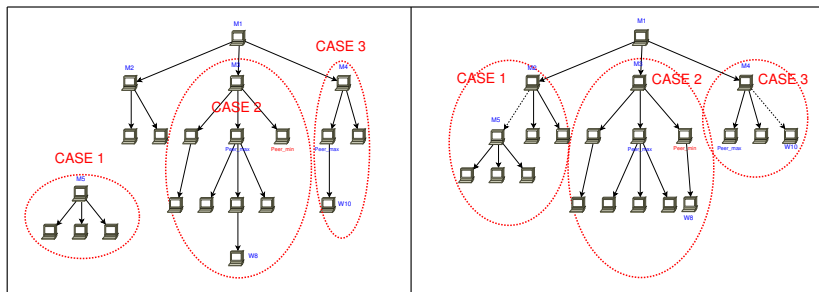
Using a rebalancing algorithm for restructuring the system.

System restructuring

Rebalancing algorithm

Three cases to consider:

- **Case 1:** Area A_i isolated from its above manager
- **Case 2:** Area A_i full with $(LPeer_{max} - LPeer_{min}) > 1$
- **Case 3:** Area A_i not full and $LPeer_{max} > 0$

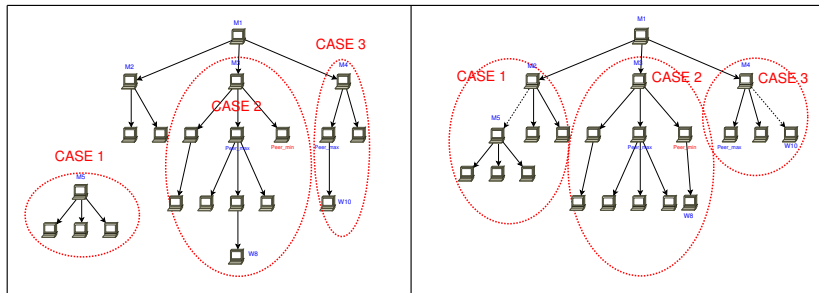


System restructuring

Rebalancing algorithm

Cost of the algorithm:

$$\log_{|Area|}(N)$$



Outline

- 1 Introduction
- 2 System model
- 3 Rebalancing algorithm
- 4 Resource Discovery Service**
- 5 Experimental results
- 6 Conclusions and future work

Resource Discovery Service

Overview

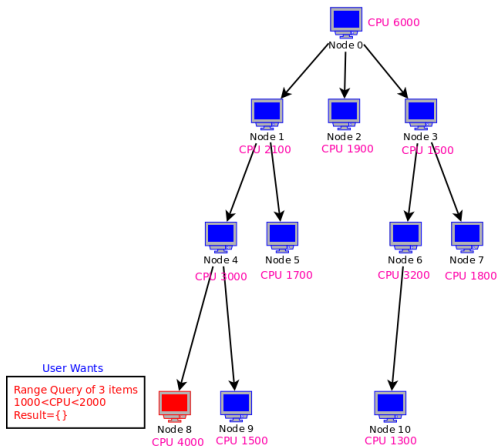
In order to schedule and execute tasks:

We have implemented a resource discovery service taking into account:

- 1 A tree-like topology.
- 2 The information provided by the updating algorithm to the managers.
- 3 Able to perform exact and range queries.
- 4 Lookup CPU power available and returns the address of peers owners.

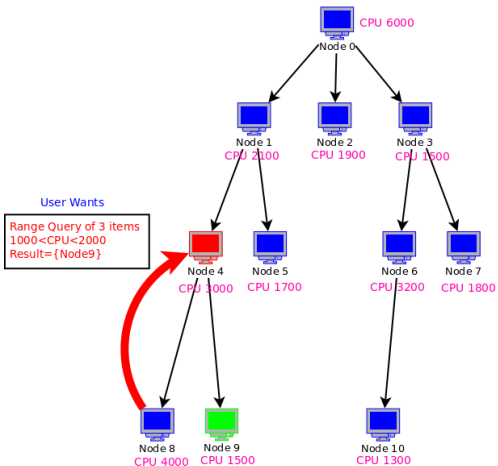
Resource discovery service

Searching algorithm: step 0



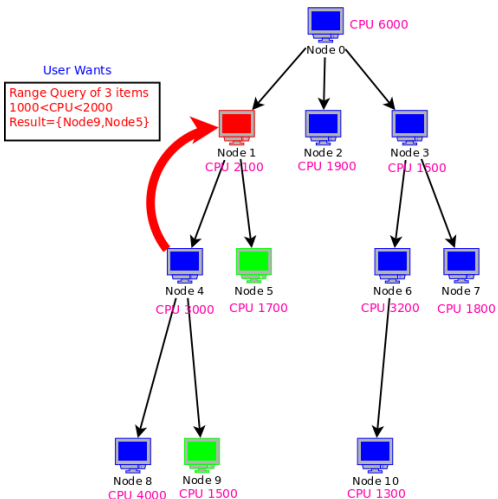
Resource discovery service

Searching algorithm: step 1



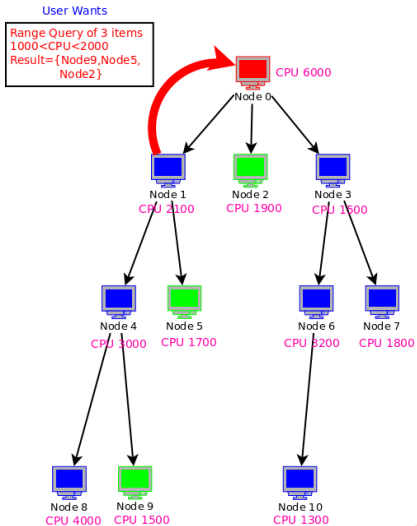
Resource discovery service

Searching algorithm: step 2



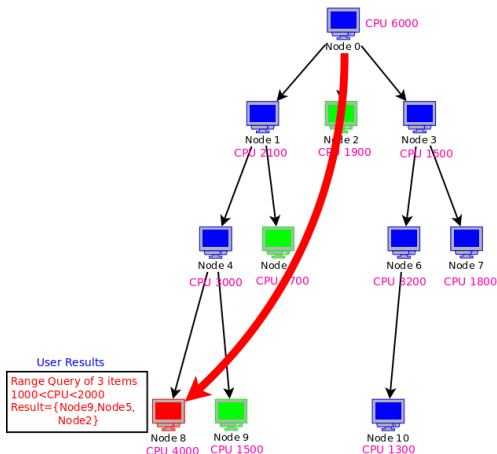
Resource discovery service

Searching algorithm: step 3



Resource discovery service

Searching algorithm: step 4

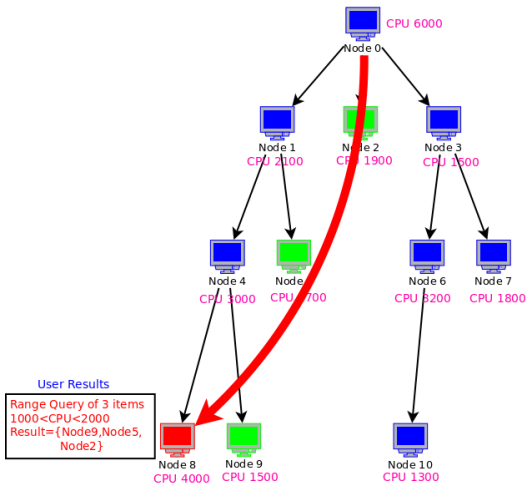


Resource discovery service

Searching algorithm:

Cost:

$$\log_{|Area|}(N)$$



Outline

- 1 Introduction
- 2 System model
- 3 Rebalancing algorithm
- 4 Resource Discovery Service
- 5 Experimental results**
- 6 Conclusions and future work

Experimental results

Simulation

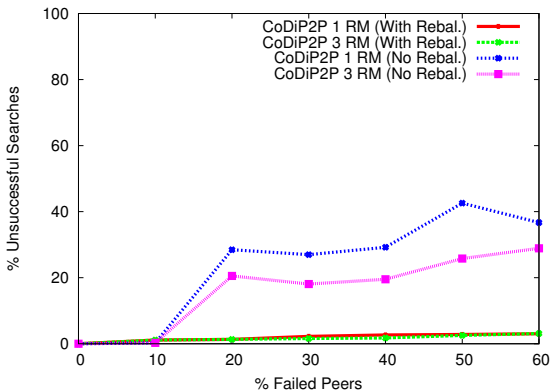
Environment

We use a GridSim based Simulator to design, implement and test all proposals.

- Number of nodes: 10000.
- Number of searches: 125.000 searches/s with a Poisson distribution with $\lambda = 125$ searches/s
- Network latency: 100ms.
- Network bandwidth: 1Mbps.

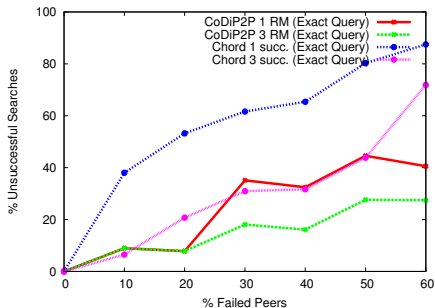
Experimental results

Influence of the rebalancing algorithm over the searching algorithm

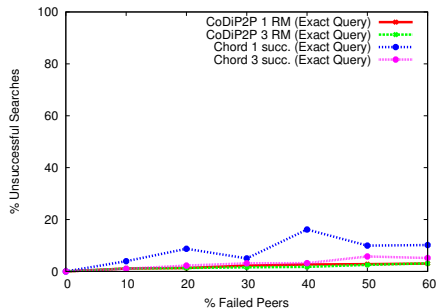


Experimental results

Reliability of exact queries searching algorithm



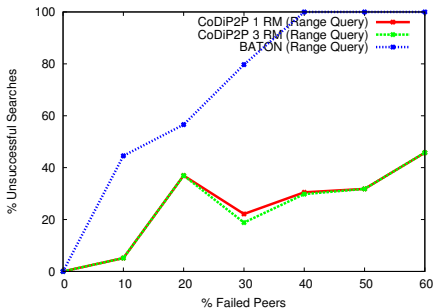
● *Freq = 125000 searches/s*



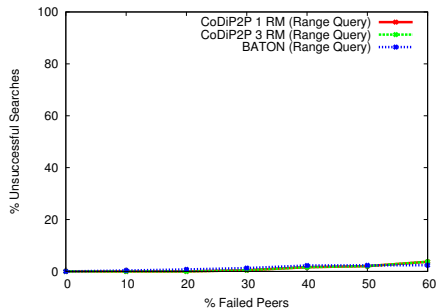
● *Freq = 125 searches/s*

Experimental results

Reliability of range queries searching algorithm



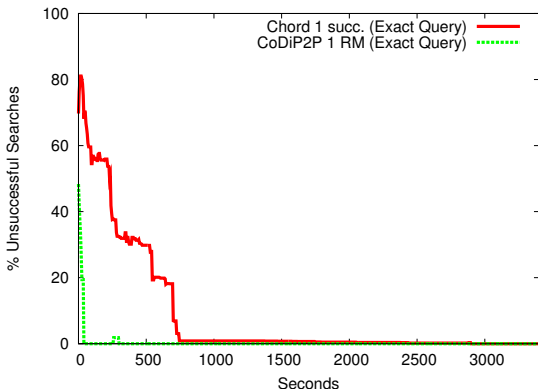
● *Freq = 125000 searches/s*



● *Freq = 125 searches/s*

Experimental results

System time response



- CoDiP2P and Chord comparative of system time response due to a 50% of Failed Peers.

Outline

- 1 Introduction
- 2 System model
- 3 Rebalancing algorithm
- 4 Resource Discovery Service
- 5 Experimental results
- 6 Conclusions and future work**

Conclusions

Achievements

- Over CoDiP2P system (tree topology):
 - We have developed the main features: connection, updating, and peer removal algorithms.
 - We have proposed two new algorithms: rebalancing and searching algorithms.
- The rebalancing algorithm ensures stability to the system.
- The reliability of lookups in CoDiP2P is higher than other topologies, specially in high frequencies.
- The system time response of CoDiP2P is lower than other topologies.

Future work

Avenues

- Testing and monitoring the tree topology and algorithms under real conditions and network.
- Extending the tree topology with a second level topology (Bruijn).



UdL

VECPAR 2010



Indra



Universitat de Lleida