Computer Science & Engineering Department

Performance Analysis of Grid DAG Scheduling Algorithms using MONARC Simulation Tool

Florin Pop, Ciprian Dobre, Valentin Cristea

Faculty of Automatics and Computer Science
University Politehnica of Bucharest, Romania

Emails: florin.pop@cs.pub.ro, ciprian.dobre@cs.pub.ro, valentin.cristea@cs.pub.ro

Computer Science
& Engineering
Department

Outline

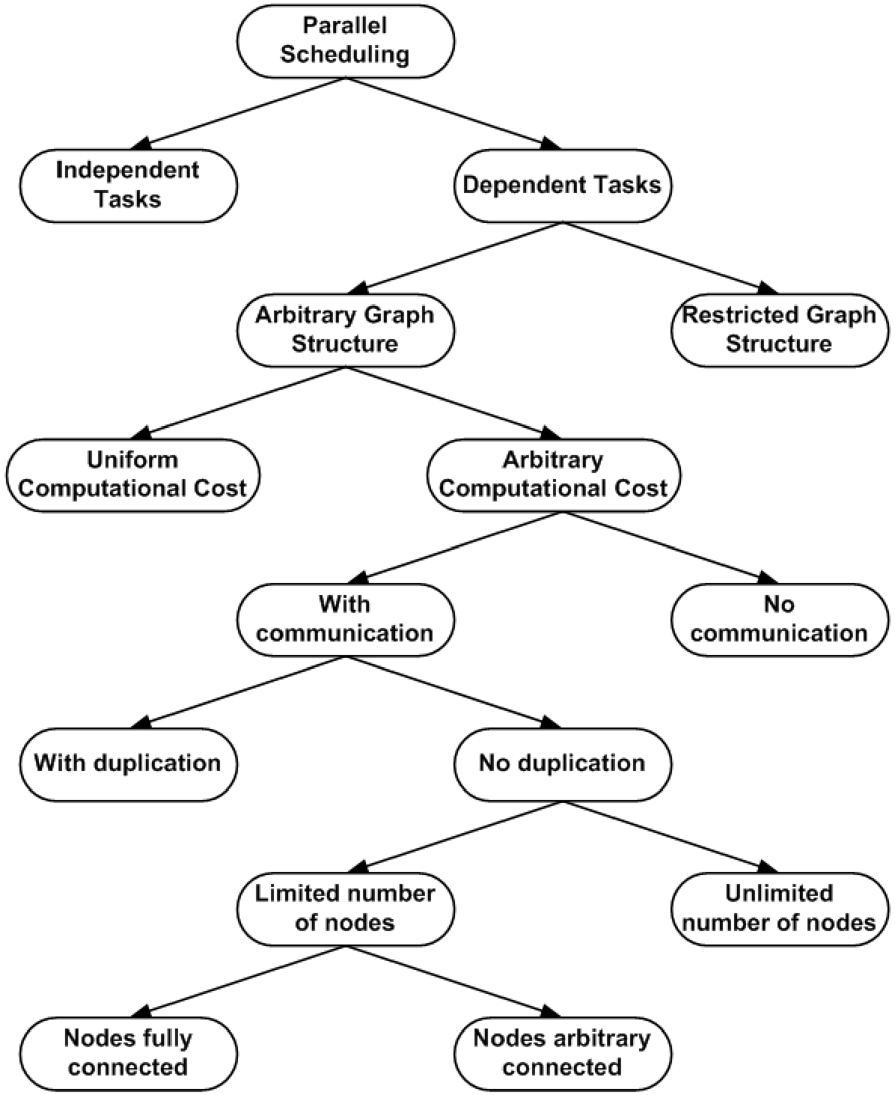
- Importance of DAG Scheduling
- DAG Scheduling Algorithms
- Importance of Simulation for Grid Systems
- Grid Simulation Tools
- MONARC Architecture
- Simulating DAG Scheduling Algorithms using MONARC
- Performance analysis
- Experimental results
- Future work
- Conclusions
- HiPerGRID!

Computer Science
& Engineering
Department

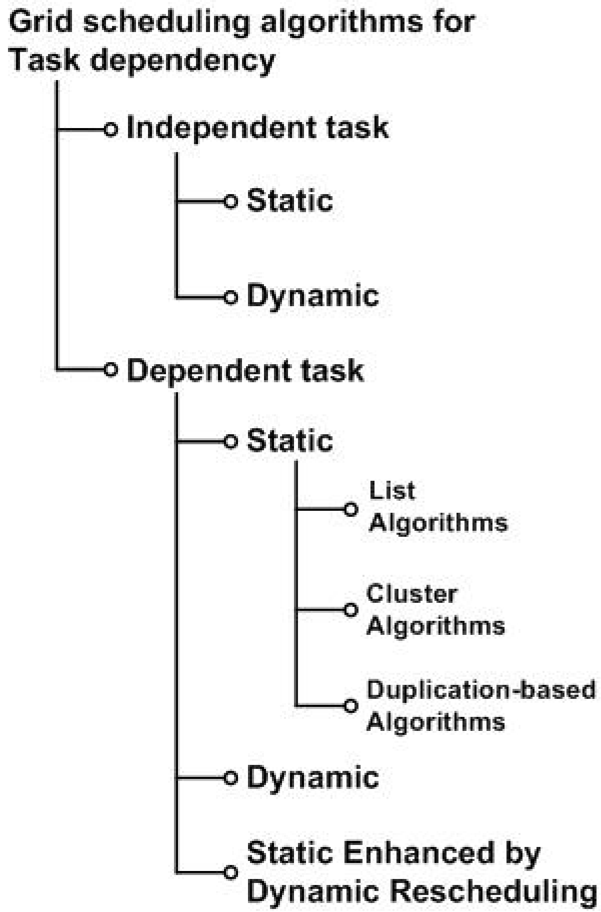
Importance of DAG Scheduling

- **The DAG scheduling problem**
  - Allocate a group of *different tasks with dependencies* to *available resources*
  - NP-Complete problem
  - Describe the scientific workflow
- **Environment: Computational GRID**
  - Heterogeneous – resources with different processing capacities
  - Public - resources can be used by multiple users
- **Objectives**
  - Efficient scheduling
    - Minimize the total execution time of the tasks
    - Uniform loading of computing resources
  - Successful completion of tasks: Deadline restrictions and Resource requirements

DAG Tasks Scheduling



A taxonomy of the DAG scheduling problem



Taxonomy of task dependency scheduling algorithms in Grid environments

DAG Scheduling Algorithms

- Scheduling algorithms based on priorities:
  - *tlevel* (top-level) - the weight of the longest path from the source node (the first scheduled task) to a node *u*
  - *blevel* (bottom-level) - the weight of the longest path from a node *u* to an exit node (the latest scheduled task).
- Static Algorithm
  - *HLFET (Highest Level First with Estimated Times)*
  - *ETF (Earliest Time First)*
- Dynamic Algorithm
  - *CCF (Cluster ready Children First)*
  - *Hybrid Remapper*

Importance of Simulation for Grid Systems

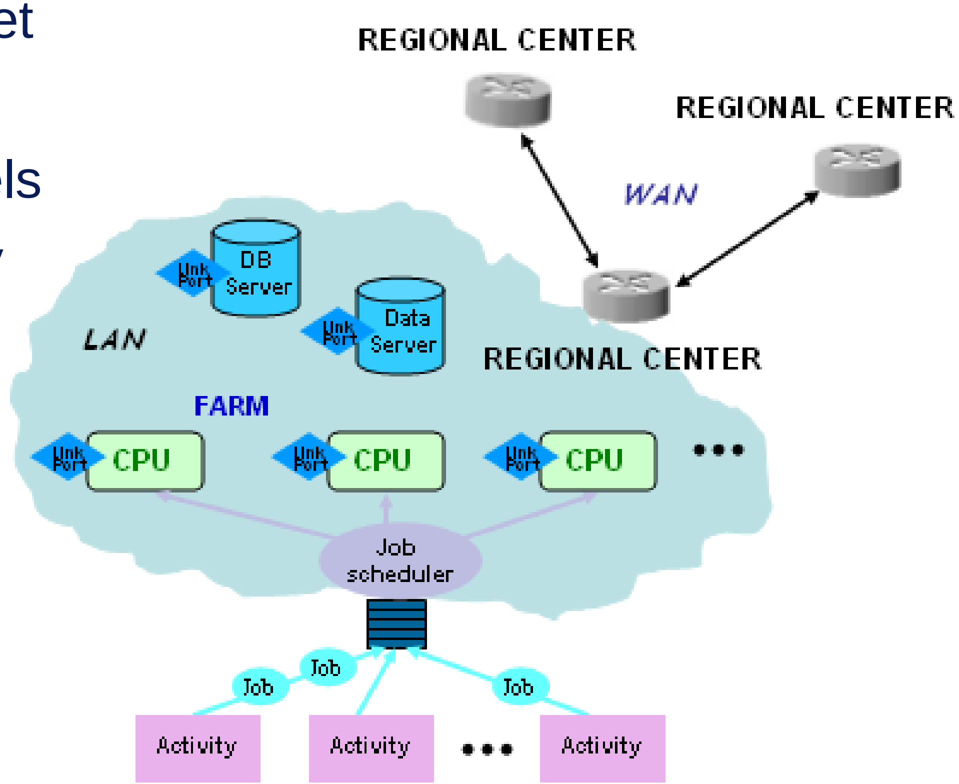
- Simulation tools
  - Discrete-event simulators
  - Use for large scale distributed systems
  - Efficient and scalable tools
  - Realistic simulation scenarios
- Importance for Grid Systems
  - Grid systems are complex: many resources and many jobs being concurrently executed
  - In Grid is difficult to made tests and evaluate performances
  - Modeling and simulation => reduces costs for evaluation
  - Flexibility for high level of parameter configuration
- Problems
  - Can you rely on result?
  - Does the experiment resembles real world systems?

Grid Simulation Tools

- *SimGrid*
  - provides core functionalities for the evaluation of scheduling algorithms and the right model and level of abstraction for studying Grid-based scheduling.
- *GridSim*
  - investigate effective resource allocation techniques based on computational economy.
- *OptorSim*
  - Data Grid simulator project designed specifically for testing various optimization technologies to access data in Grid environments.
- *… Others …*

MONARC Architecture

- Realistic simulation
  - Abstraction for all the system components and their interactions
  - Physical resources of the distributed system (Regional Center).
  - Modeling the behavior of the applications and their interaction.
  - Dynamically instantiate a set of users or activity objects.
  - Build a wide range of models
  - arbitrary level of complexity

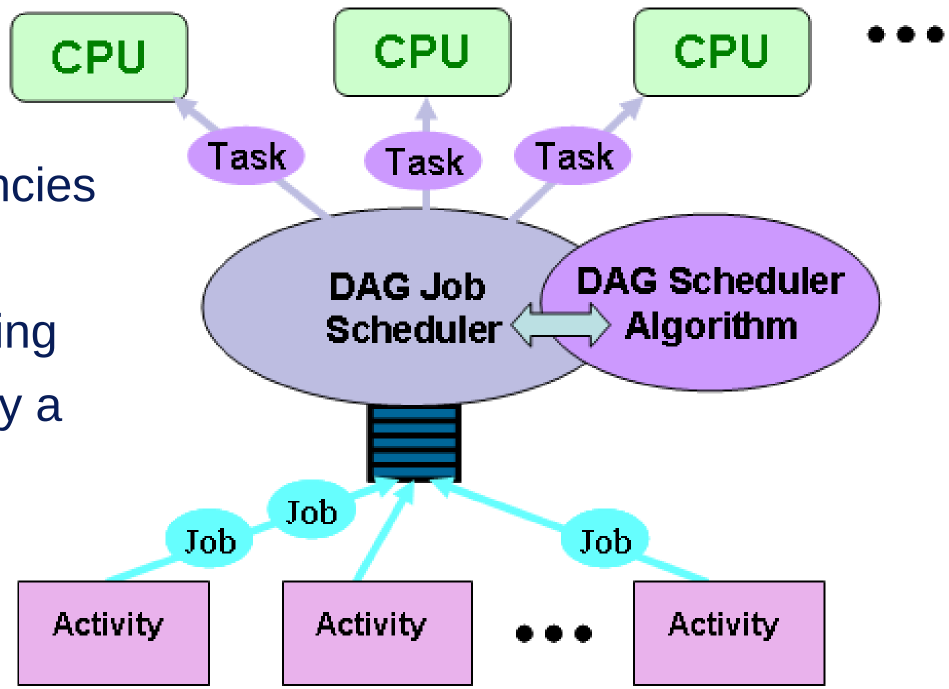


Simulating DAG Scheduling Algorithms using MONARC

- The simulation model for DAG Scheduling Algorithms
  - MONARC allows the evaluation of advanced scheduling algorithms.
  - extend behavior of the job model in MONARC
  - allow for more realistic scheduling decisions
  - underlying topology

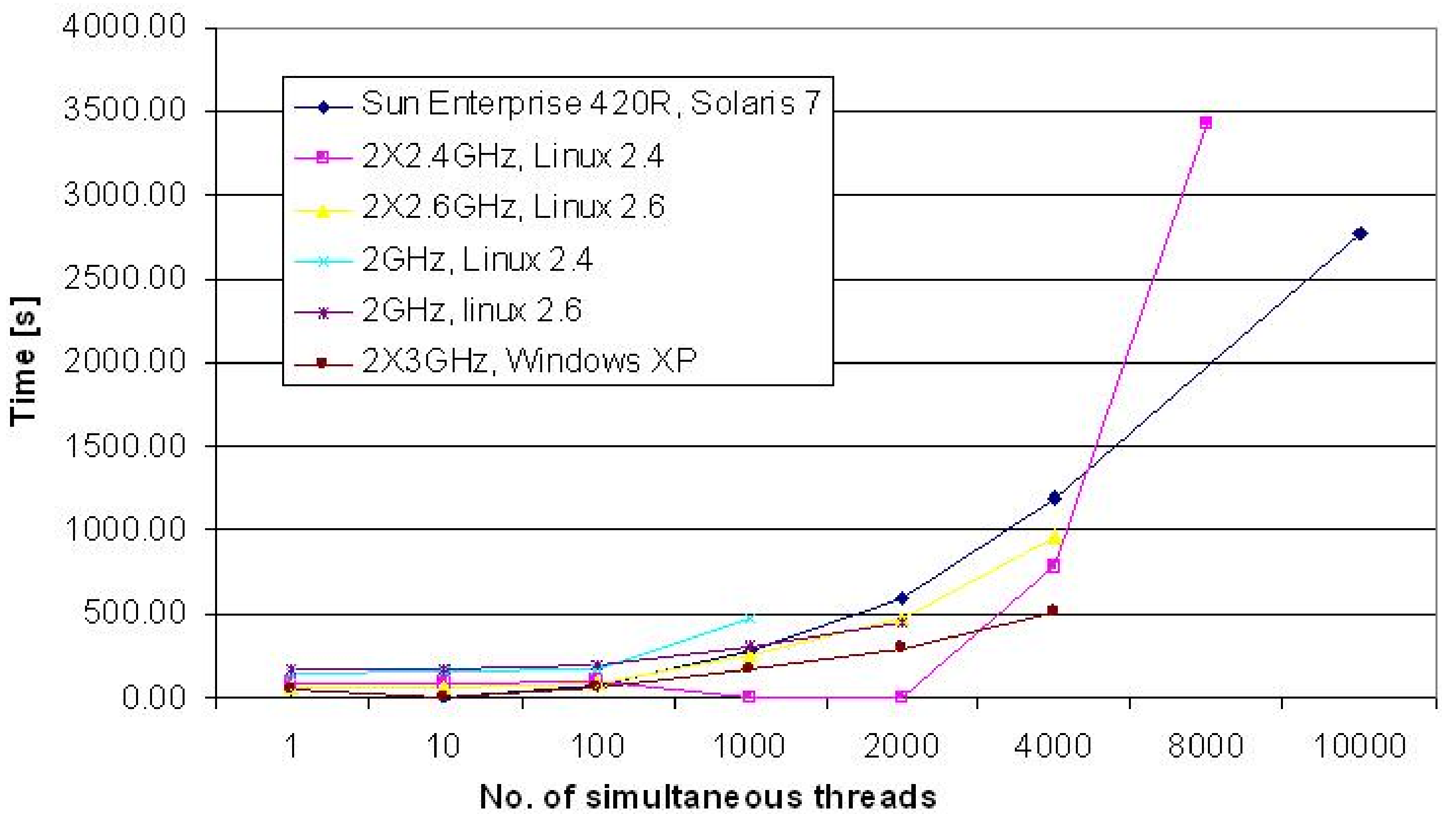
  (input parameter for the DAG job scheduler algorithm)
  - full topology of tasks dependencies
  - allow a job to carry enough

  information about the corresponding input DAG topology as required by a DAG scheduling algorithm

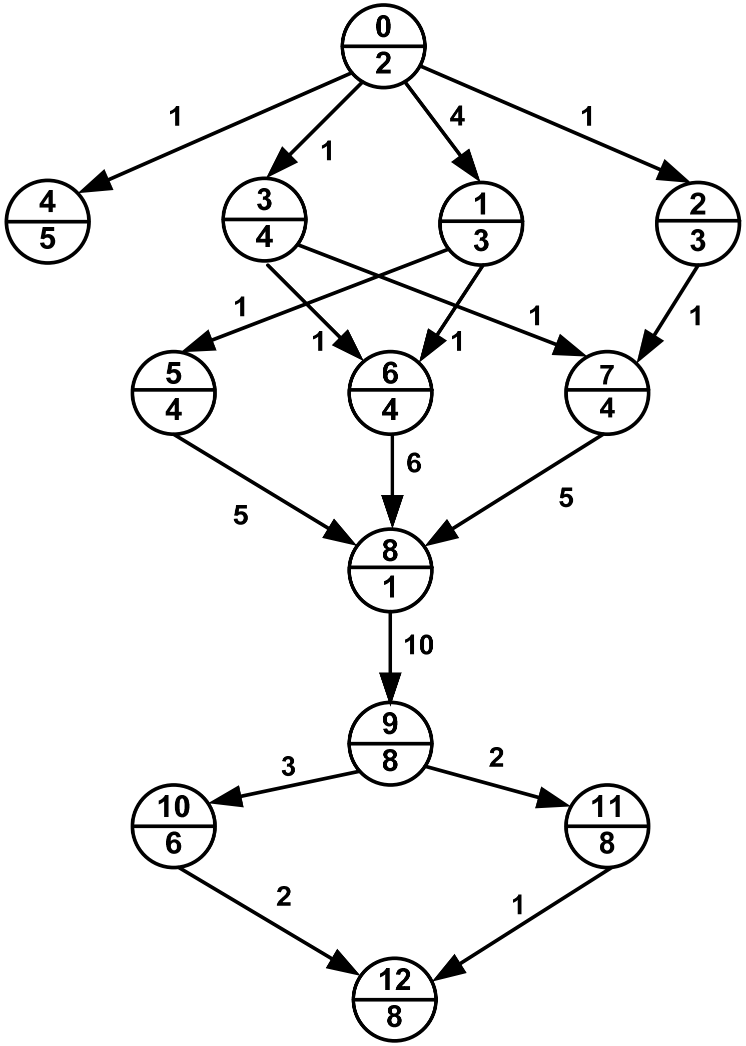


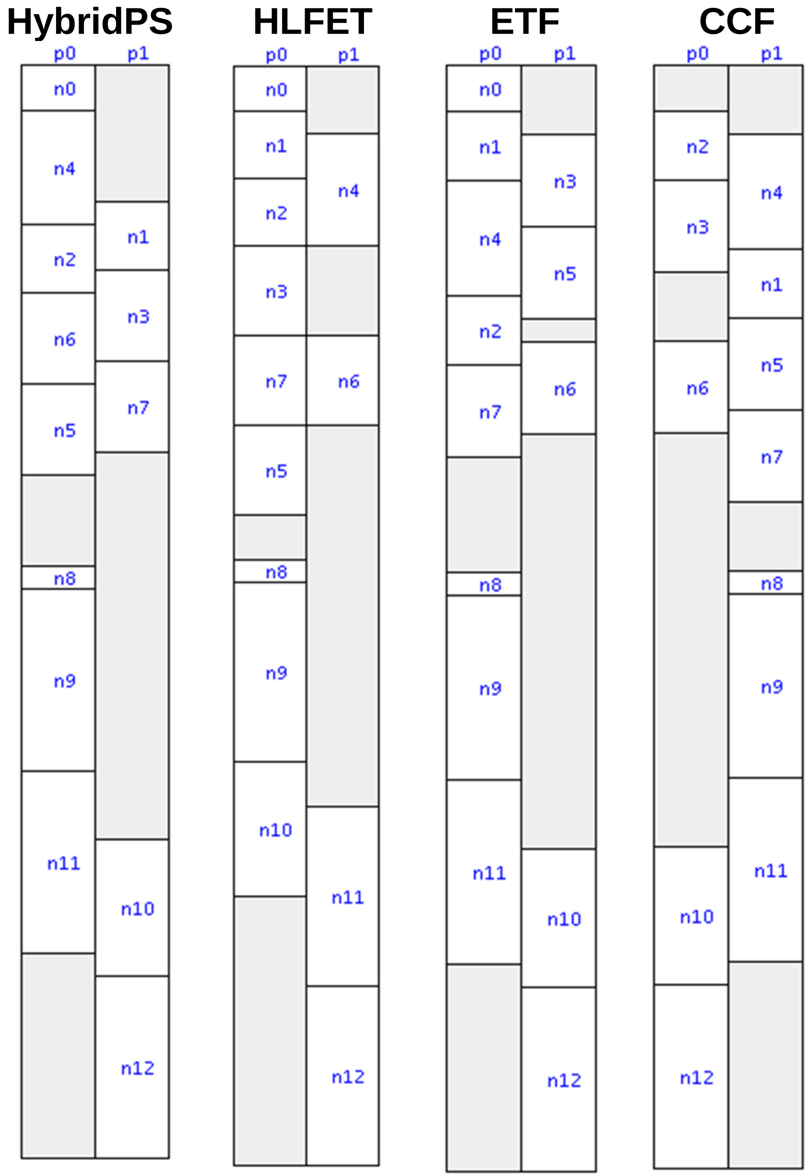
Performance analysis

- Simulation of different type of architecture => testing algorithm in a heterogeneous environments.
- Very good performance results modeling very large distributed systems



Experimental results (1/2)

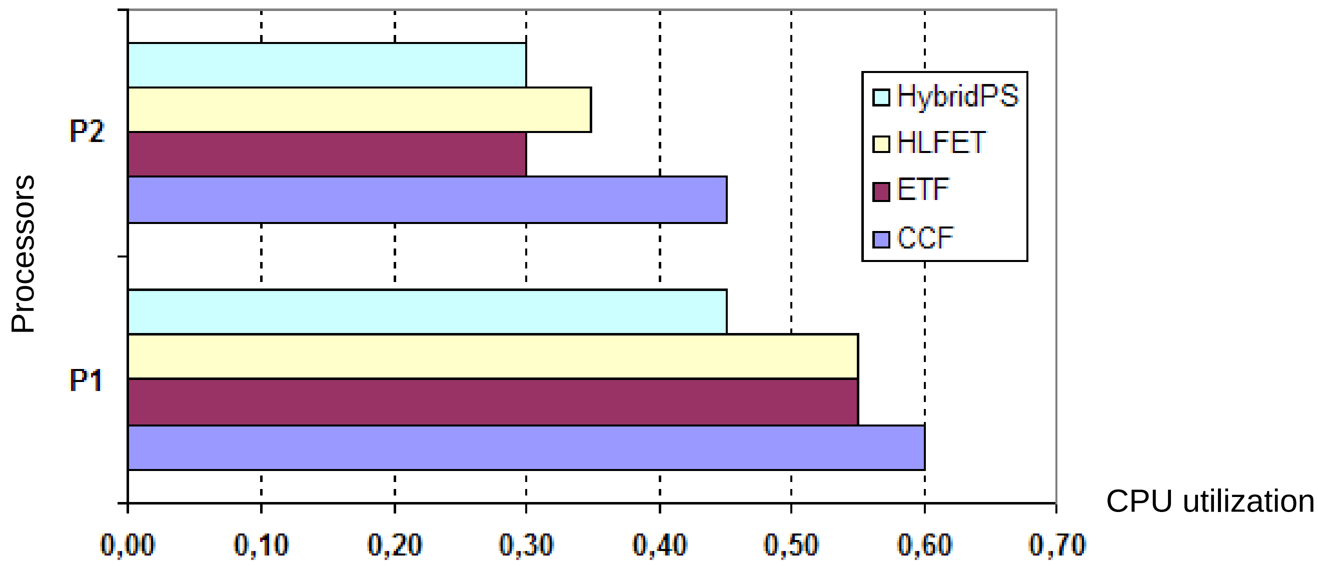




Scheduling results for proposed DAG scheduling algorithms =>

Experimental results (2/2)

- The CPU utilization results in each of the four experiments.
- The highest CPU utilization values were obtained for the CCF scheduling algorithm, explained by the lower number of transfers between different processors involved.



	P1	P2
HybridPS	0,45	0,30
HLFET	0,55	0,35
ETF	0,55	0,30
CCF	0,60	0,45

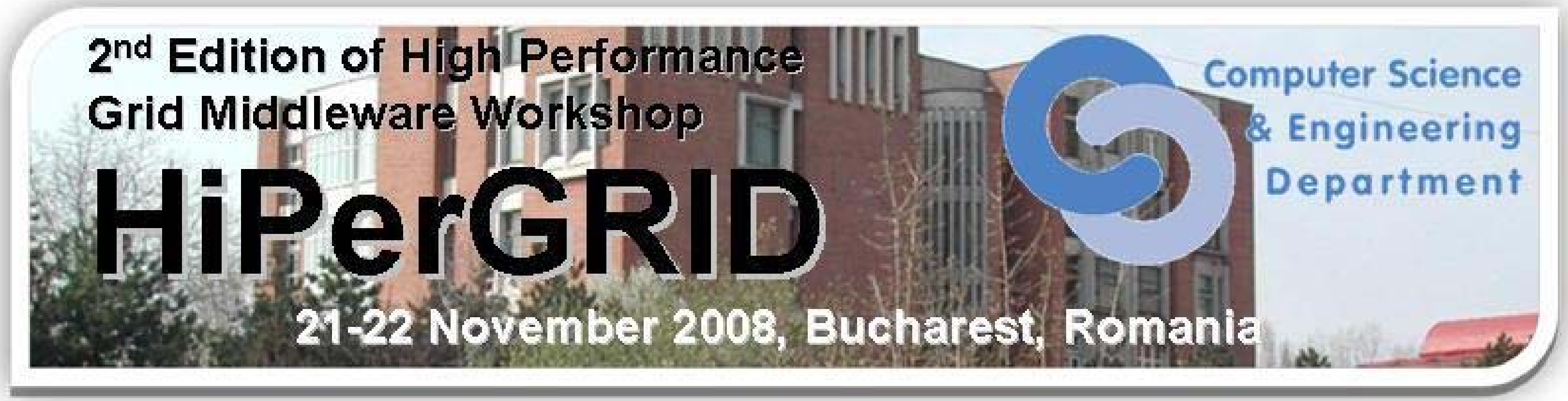
Future work

- The analysis of a wider set of scheduling algorithms currently used in Grid systems.
- Scheduling algorithms for real-time scenarios.
- Solutions for backup and recovery from error (re-scheduling) and solving the problem of co-scheduling and multi-criteria constraints scheduling

Conclusions

- In this paper we proposed a simulation-based solution to evaluate the performances of Grid scheduling algorithms.
- The results could be used in decisions regarding optimizations to existing Grid DAG Scheduling and for selecting the proper algorithm for DAG scheduling in various actual situations.
- The main contribution of the presented research consists in the development of the simulation layer in MONARC that is appropriate for DAG scheduling algorithms evaluation
- The simulator represented an adequate instrument to approach a key problem of ensuring the high performance behavior of the Grid: the scheduling dependent activities.

Thank you!



http://hipergrid.grid.pub.ro