



Universitat
Autònoma
de Barcelona

UNIVERSITY OF
WISCONSIN
MADISON

Scheduling for MW applications

E. Heymann, M.A. Senar and E. Luque

Computer Architecture and Operating Systems Group

Universitat Autònoma of Barcelona

SPAIN

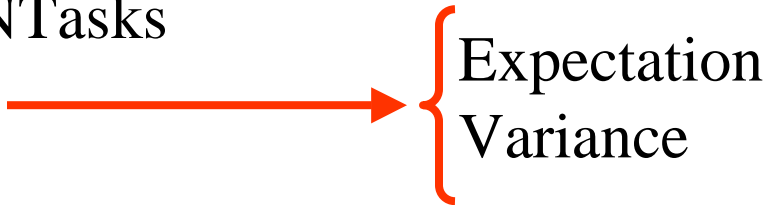
Contents

- ↗ Statement of the problem.
- ↗ Simulation framework.
- ↗ Simulation results.
- ↗ Implementation on MW.
- ↗ Future work.

Objective

- To develop and evaluate efficient scheduling policies for parallel applications with the following MW model:

```
for i=1 to M
  for j=1 to NTasks
    do F(j)
  end
  [computation]
end
```



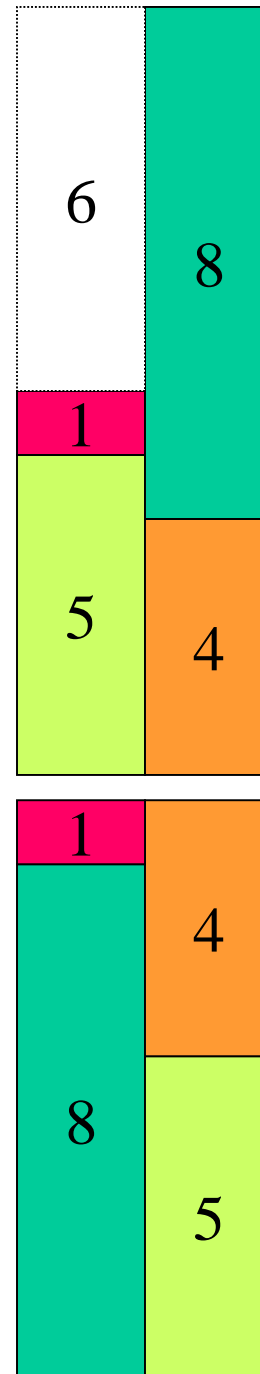
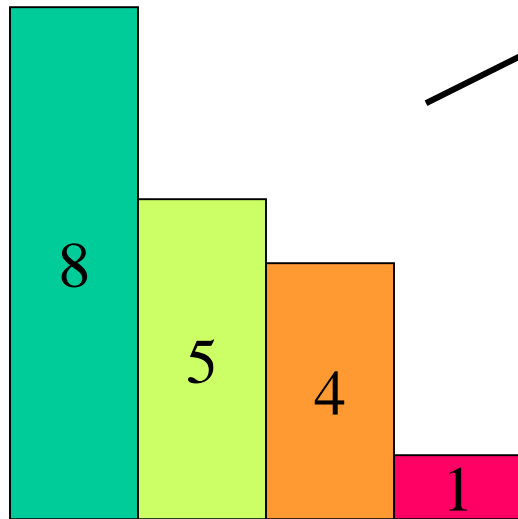
The diagram shows a red arrow pointing from the text 'do F(j)' in the code block to a red bracket on the right. The bracket encloses the words 'Expectation' and 'Variance' stacked vertically.

In an opportunistic environment

Objective

- How many workers ?
- How to assign tasks to workers ?
- Efficiency without forgetting performance.
- How sensitive is efficiency with respect to variance changes.

Example



Without preemption

$$\text{Efficiency} = \frac{6 + 12}{12 + 12} = \frac{3}{4}$$

LPTF Policy

$$\text{Efficiency} = \frac{9 + 9}{9 + 9} = 1$$

Platforms

- Dedicated homogeneous machines.
- Non-dedicated (such as Condor) homogeneous machines.
- Non-dedicated (such as Condor) heterogeneous machines.

Policies Simulated

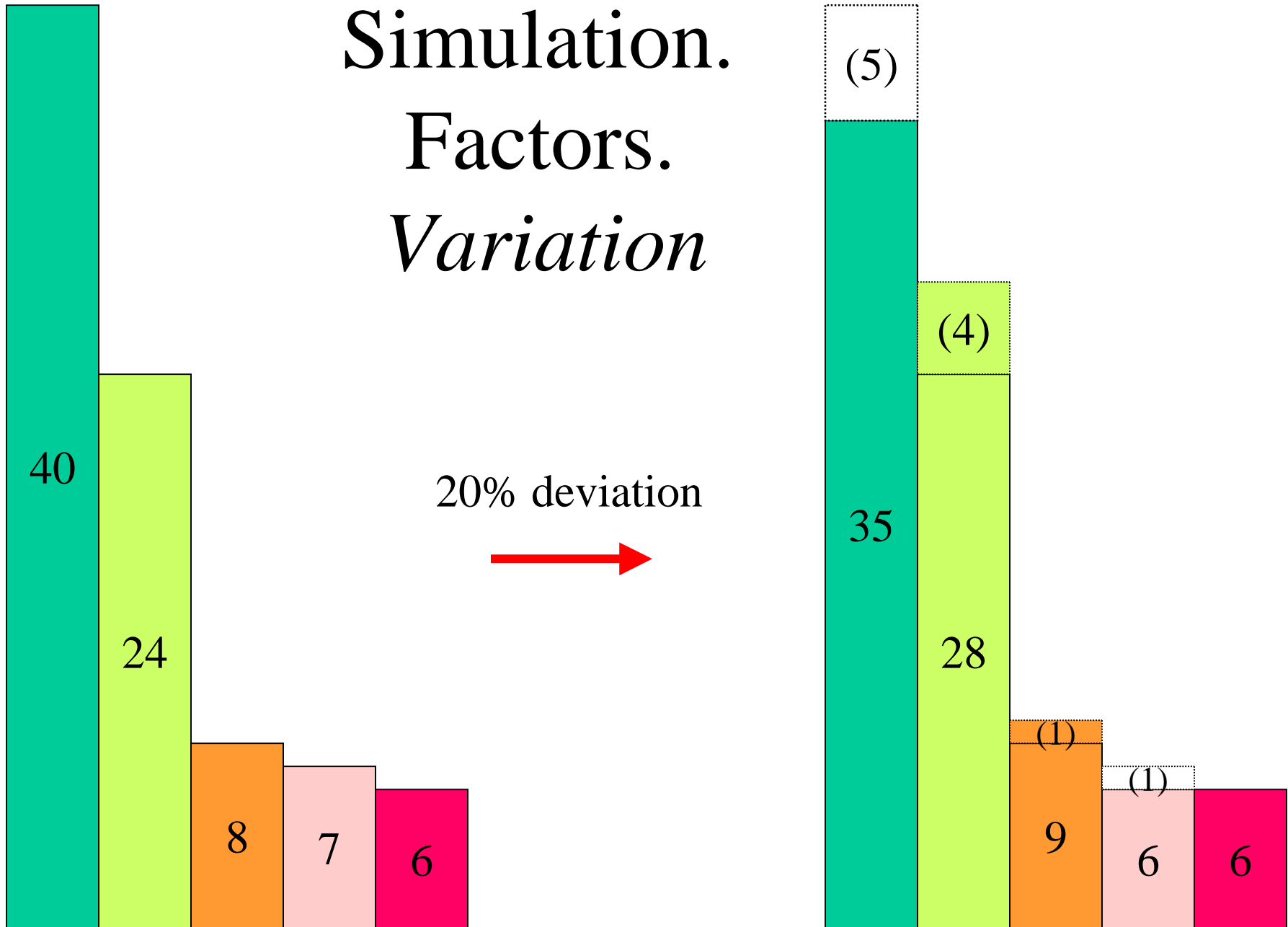
- LPTF: Largest processing time first
- LPTF on Average
- Random
- Random and Average

Policy	Next task to be assigned
LPTF	The biggest task
LPTF on Average	The biggest task considering execution times without variation
Random	A random task
Random & Average	<i>1st iteration:</i> A rand task <i>Next iterations:</i> The biggest task based on execution times of previous iterations

Simulation

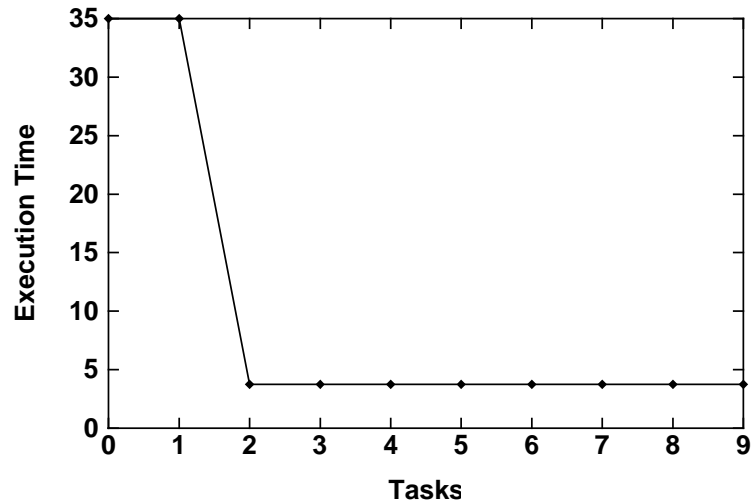
- Response Variables:
 - Efficiency
 - Execution Time

Simulation. Factors. *Variation*

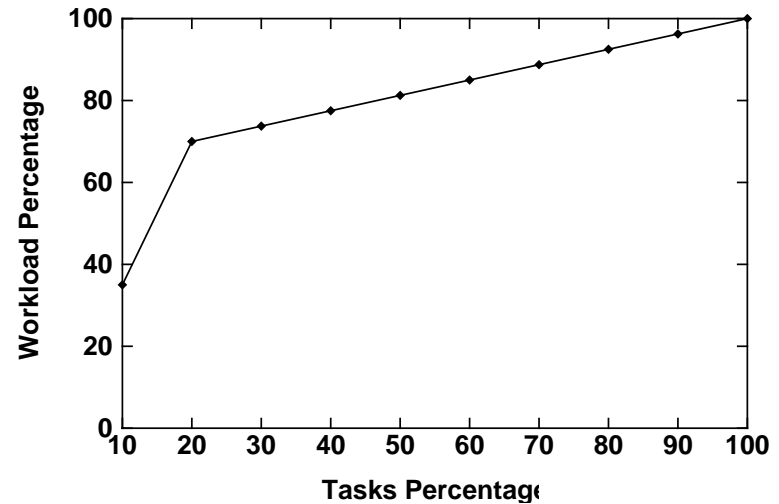


Simulation. Factors. Workload

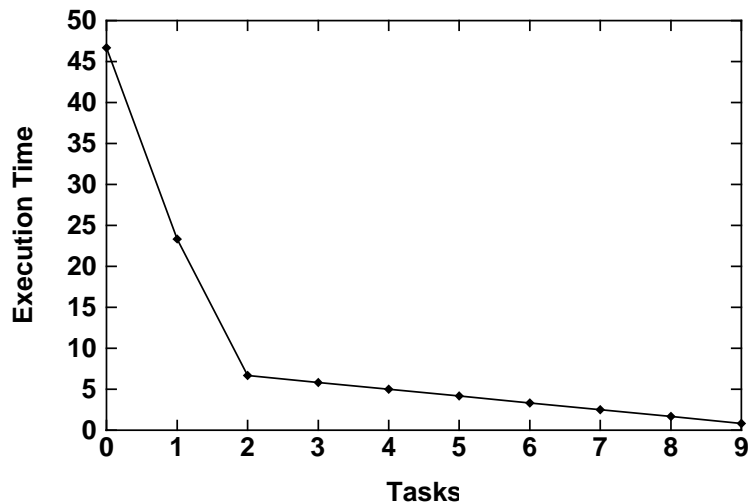
WorkPercentage: 70% 0-0 TASKS=10



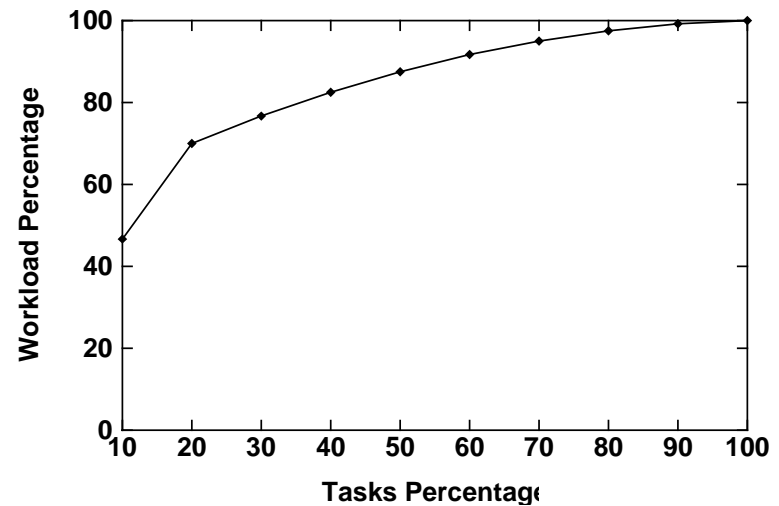
WorkPercentage: 70% 0-0 TASKS=10



WorkPercentage: 70% 1-1 TASKS=10



WorkPercentage: 70% 1-1 TASKS=10

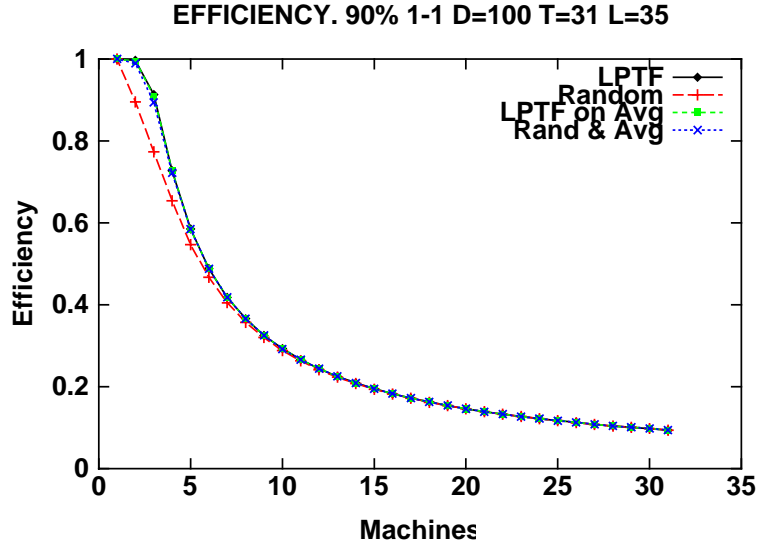
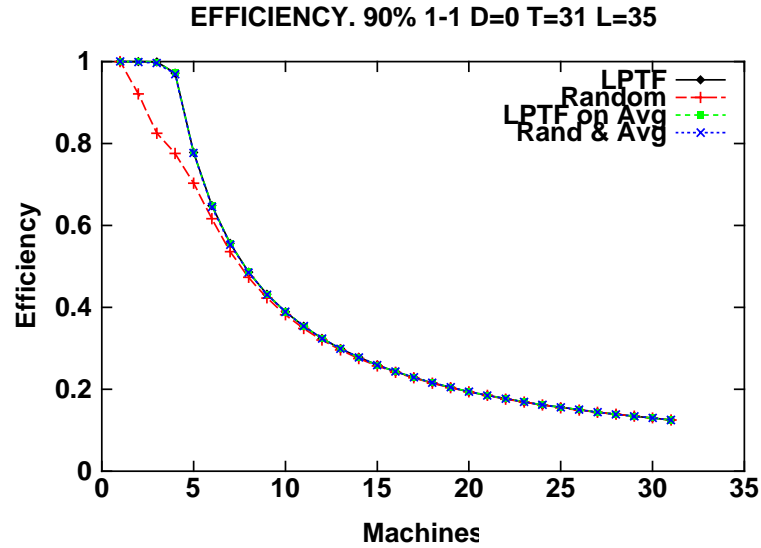
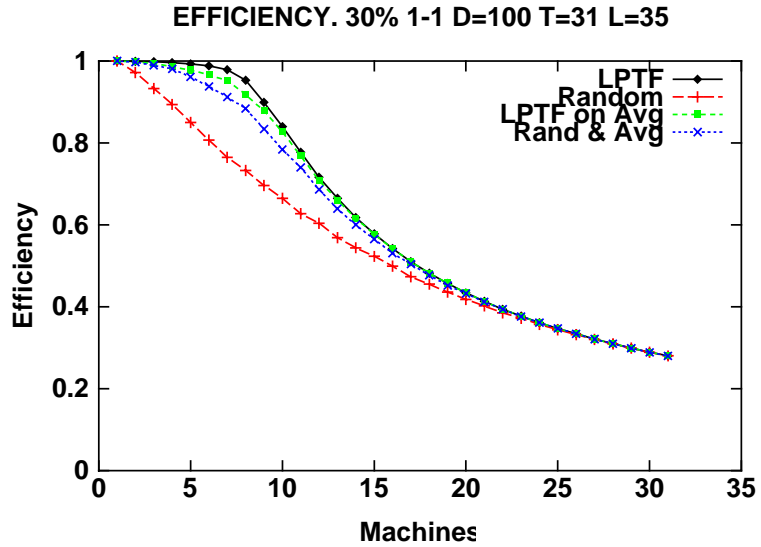
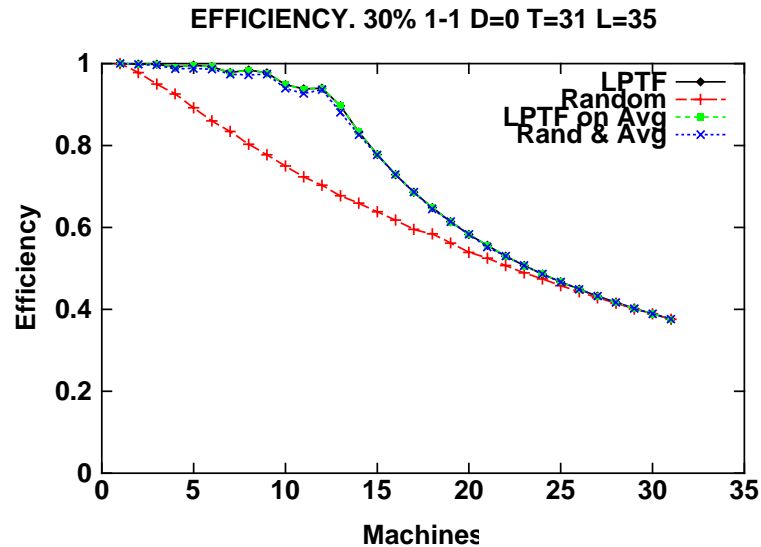


Simulation. Factors

- Processor Number → 31, 100, 300
- Standard Deviation → 0, 10, 30, 60, 100
- External Loop → 10, 35, 50, 100
- Workload
 - 20% load: 30, 40, 50, 60, 70, 80, 90
 - 20% dist: equal, decreasing
 - 80% dist: equal, decreasing

Simulation Results

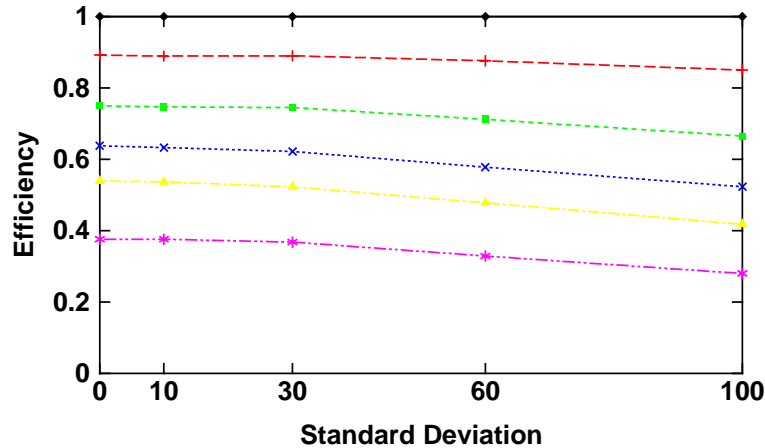
Dedicated homogeneous machines



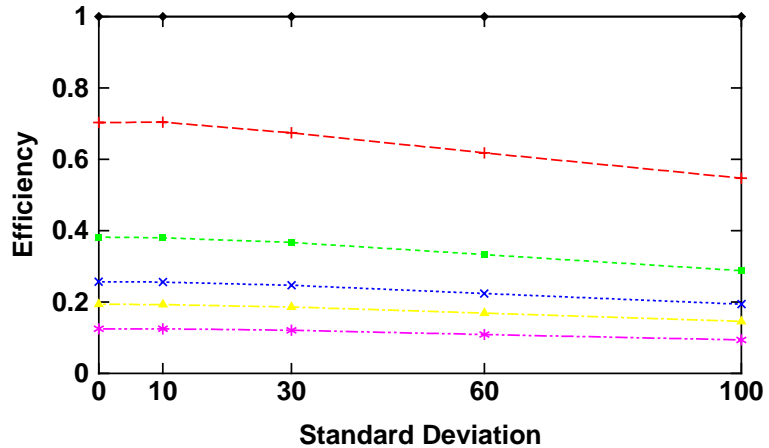
Simulation Results

Dedicated homogeneous machines

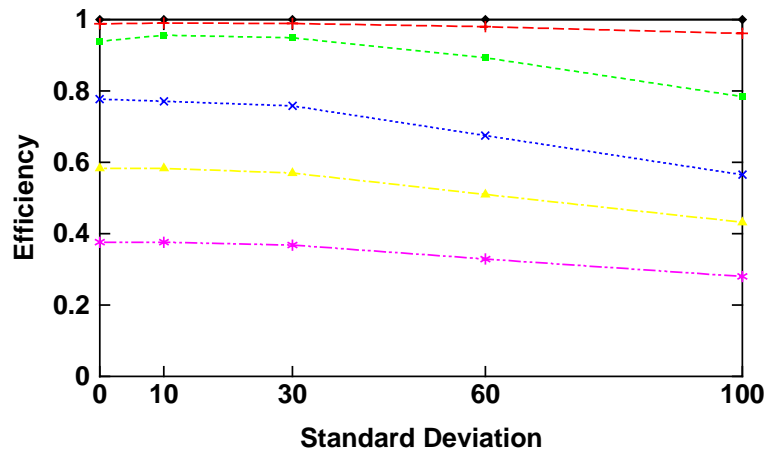
RANDOM. W: 30% 1-1 T=31 L=35



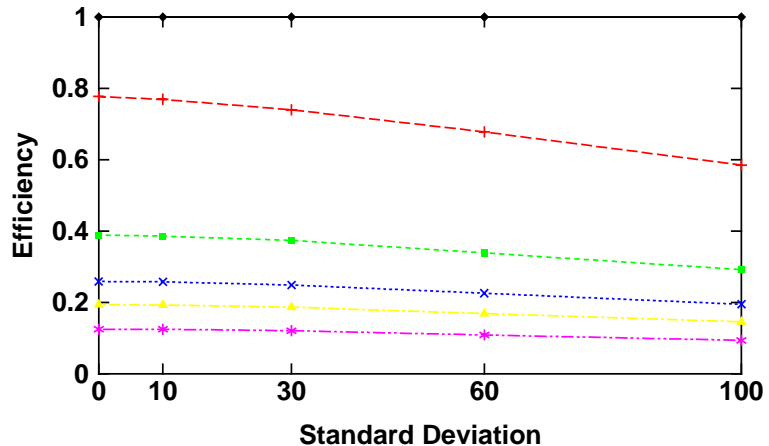
RANDOM. W: 90% 1-1 T=31 L=35



RAND & AVG. W: 30% 1-1 T=31 L=35



RAND & AVG. W: 90% 1-1 T=31 L=35

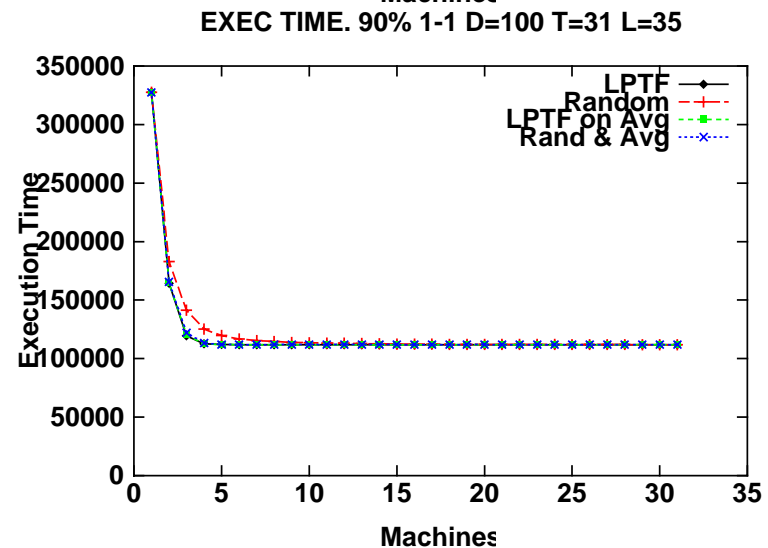
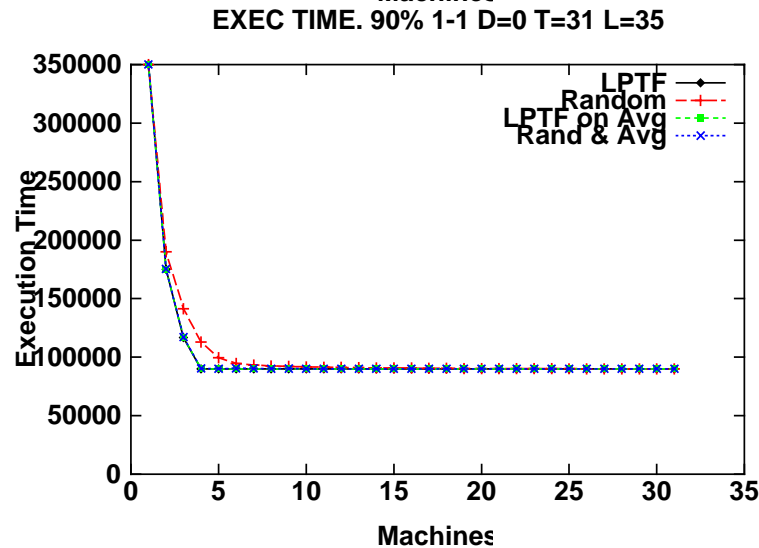
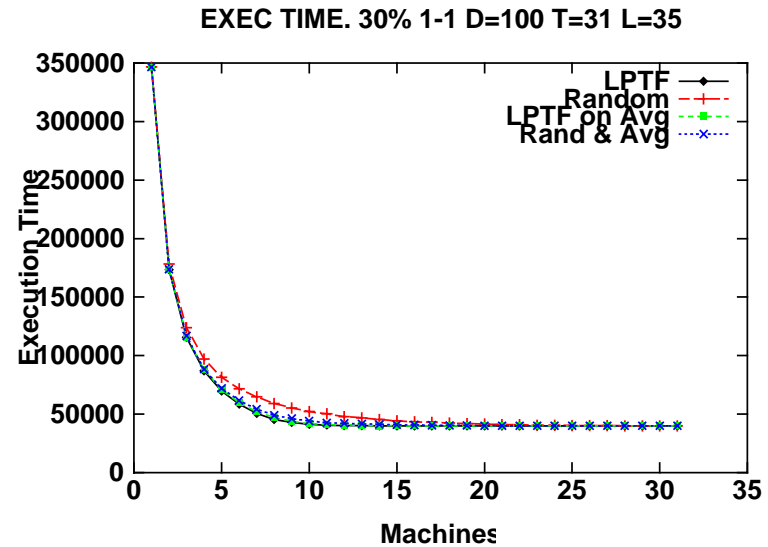
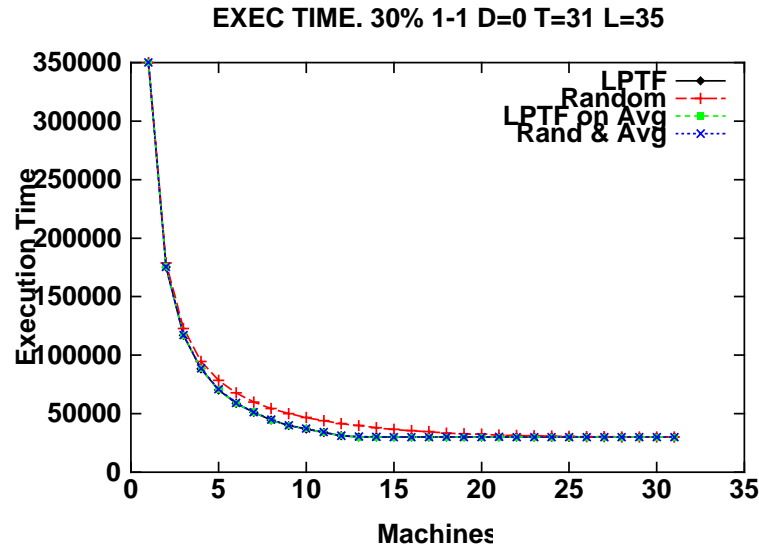


1 Worker —●— 10 Workers —●— 20 Workers —●—
5 Workers —+— 15 Workers —x— 31 Workers —*—

1 Worker —●— 10 Workers —●— 20 Workers —●—
5 Workers —+— 15 Workers —x— 31 Workers —*—

Simulation Results

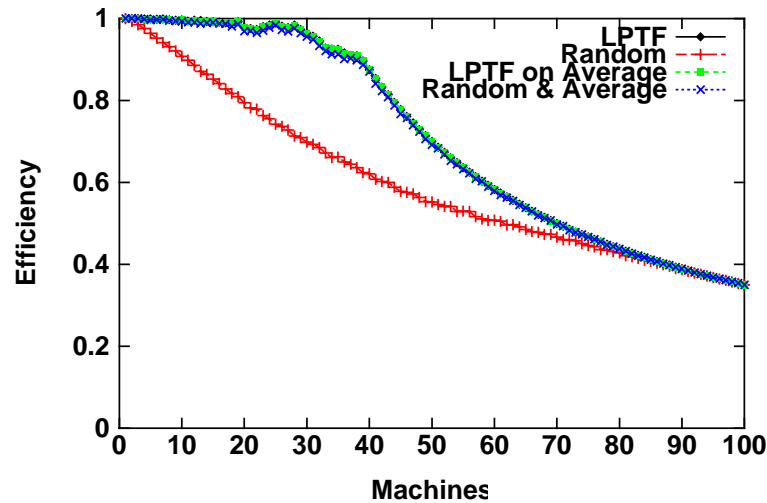
Dedicated homogeneous machines



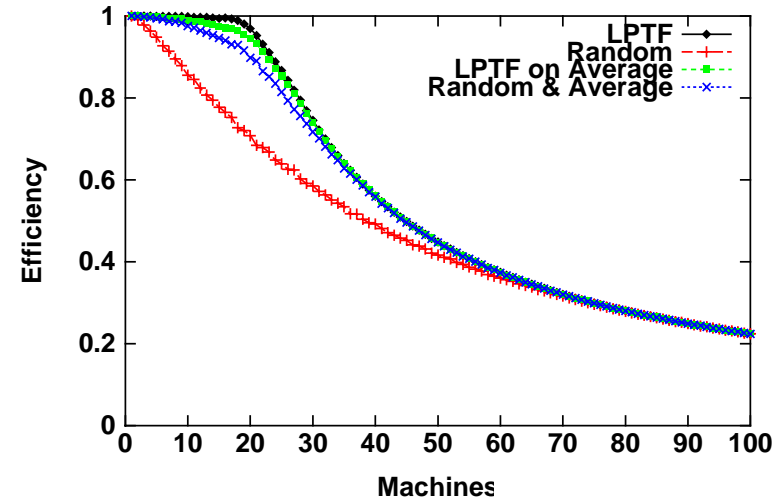
Simulation Results

Dedicated homogeneous machines

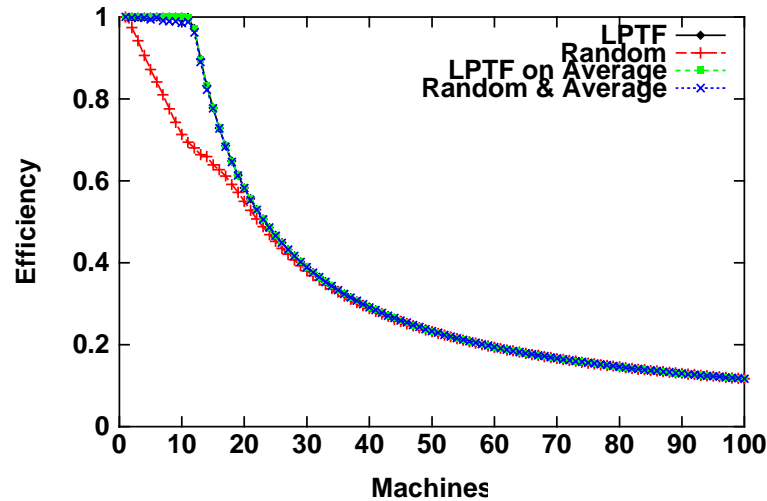
EFF. 30% 1-1 D=0 T=100 L=35



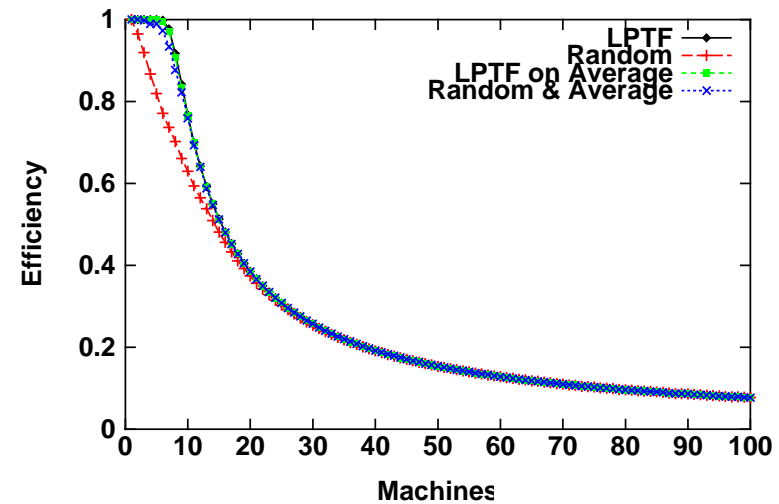
EFF. 30% 1-1 D=100 T=100 L=35



EFF. 90% 1-1 D=0 T=100 L=35



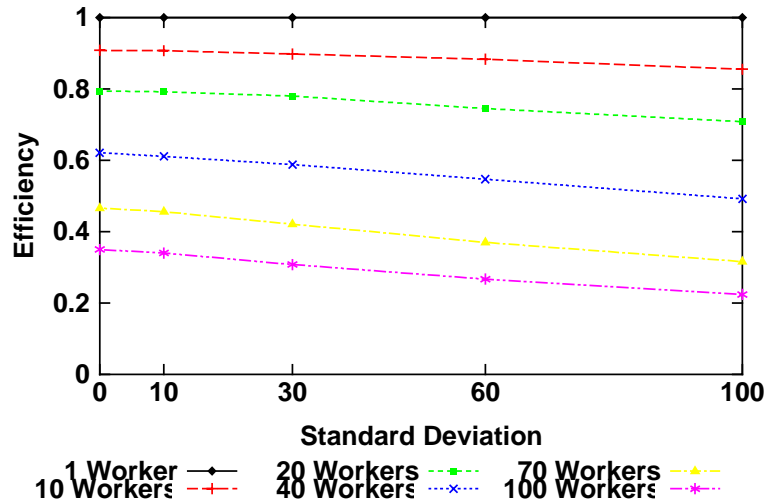
EFF. 90% 1-1 D=100 T=100 L=35



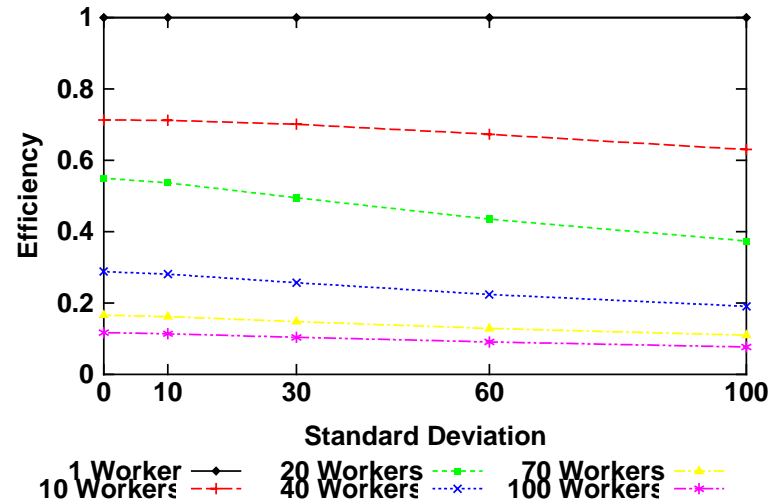
Simulation Results

Dedicated homogeneous machines

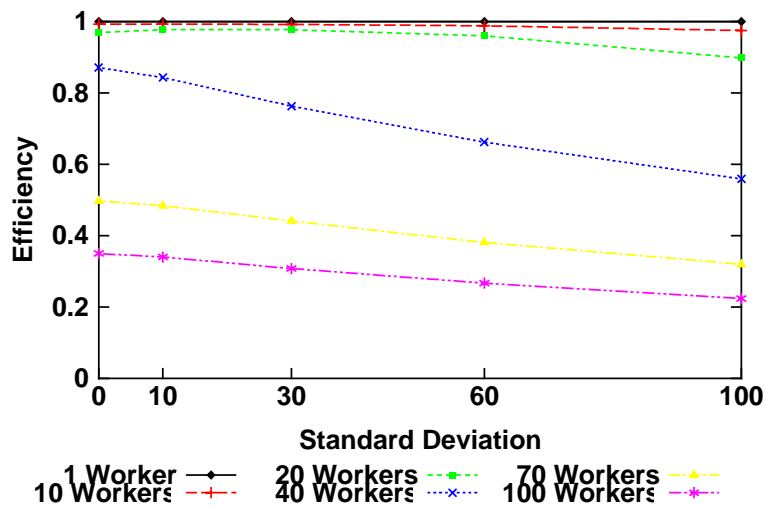
RANDOM. 30% 1-1 T=100 L=35



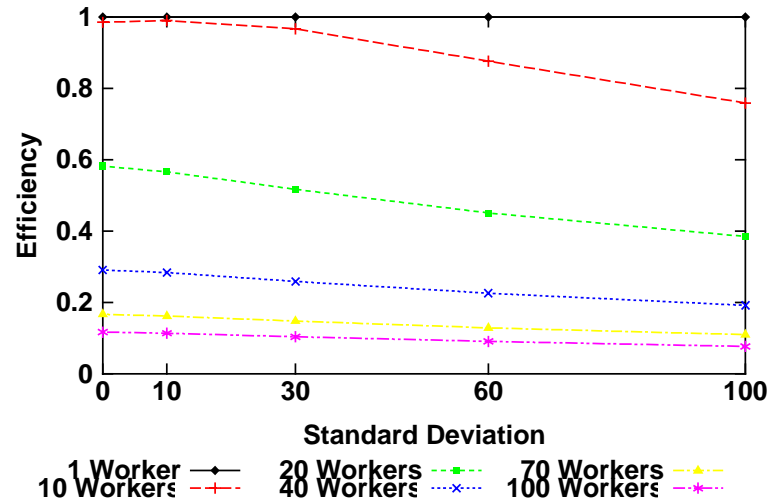
RANDOM. 90% 1-1 T=100 L=35



RAND & AVERAGE. 30% 1-1 T=100 L=35



RAND & AVERAGE. 90% 1-1 T=100 L=35



Simulation Conclusions

Dedicated homogeneous machines

Rough analysis:

- Variance does not seem to make efficiency significantly worse.
- External loop does not affect efficiency.
- To achieve an efficiency $> 80\%$ and an execution time < 1.1 respect to LPTF execution time, the number of workers should range between 15% of the tasks number (90% load) and 40% (30% load).

Simulation

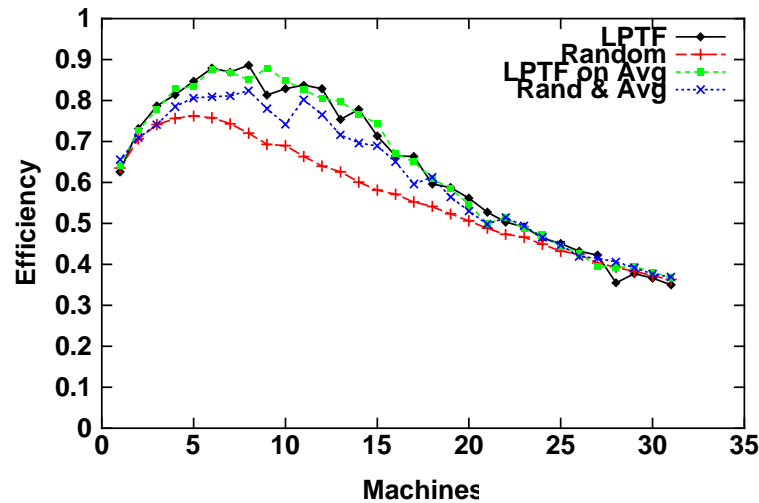
Non-dedicated homogeneous machines

- Factors:
 - Processor Number ✓
 - Standard Deviation ✓
 - External Loop → Only 35
 - Workload ✓
 - Probability of loosing and getting machines
 - Checkpoint
 - Always
 - Never
 - Only for “big” tasks that have been “a long time” in execution

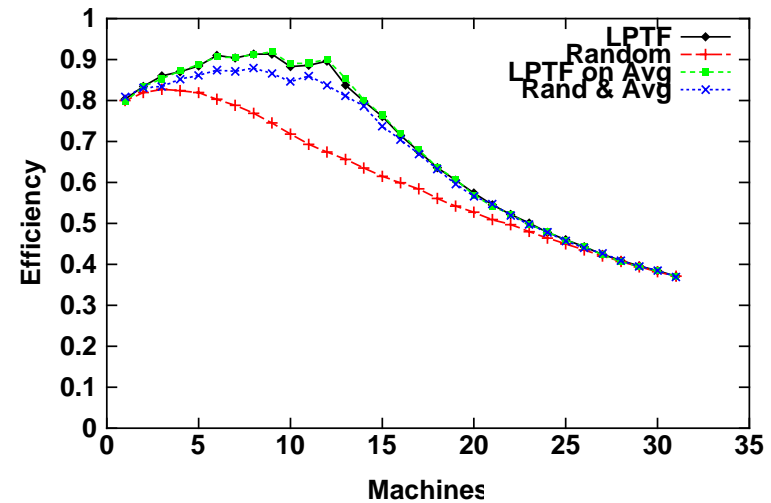
Simulation Results

Non-dedicated homogeneous machines

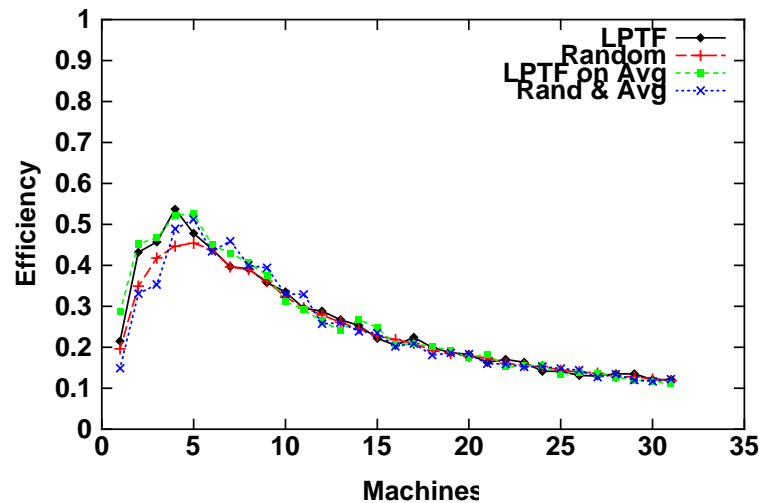
EFF ck=NO 30% 1-1 D=0 T=31 L=35



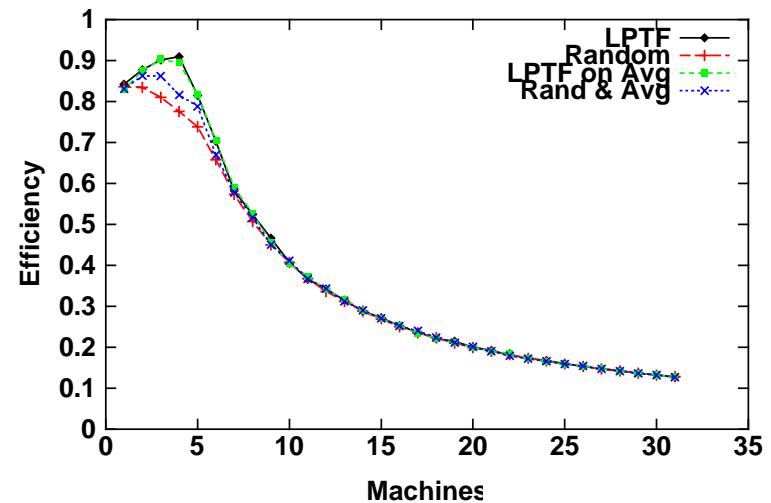
EFF ck=YES 30% 1-1 D=0 T=31 L=35



EFF ck=NO 90% 1-1 D=0 T=31 L=35



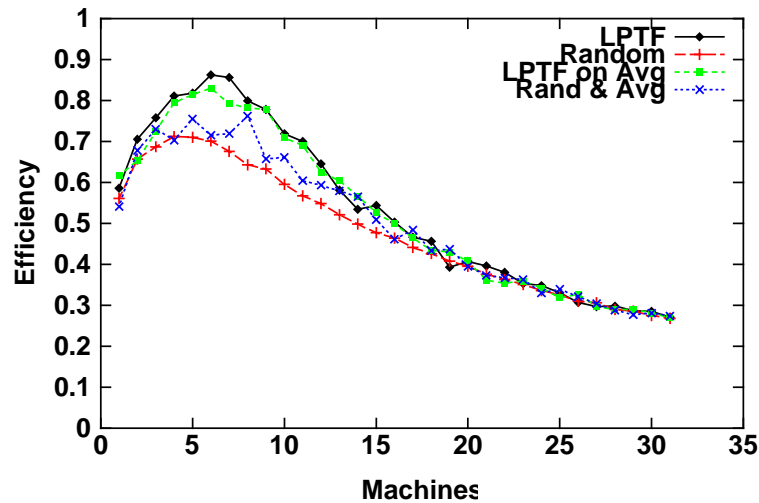
EFF ck=YES 90% 1-1 D=0 T=31 L=35



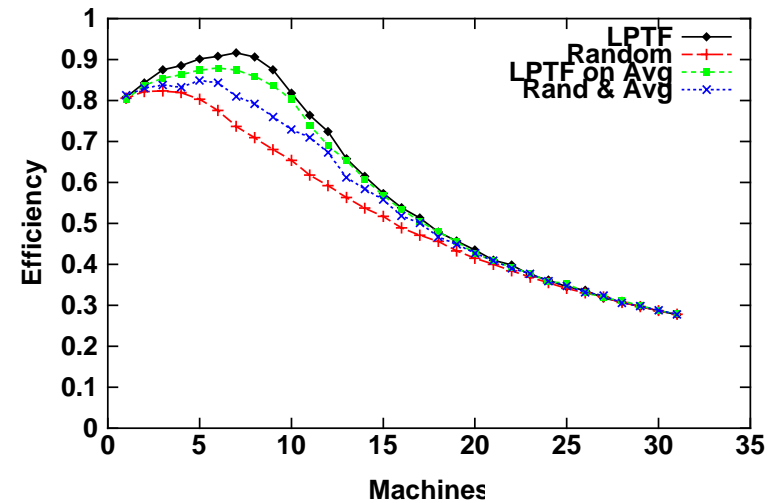
Simulation Results

Non-dedicated homogeneous machines

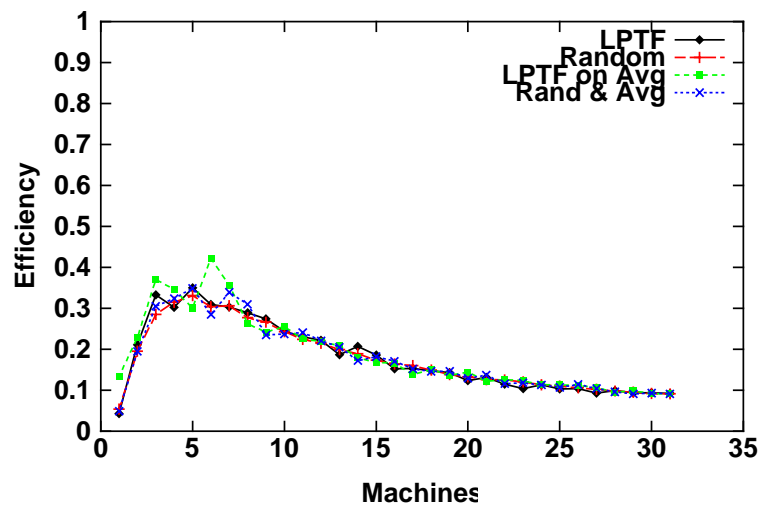
EFF ck=NO 30% 1-1 D=100 T=31 L=35



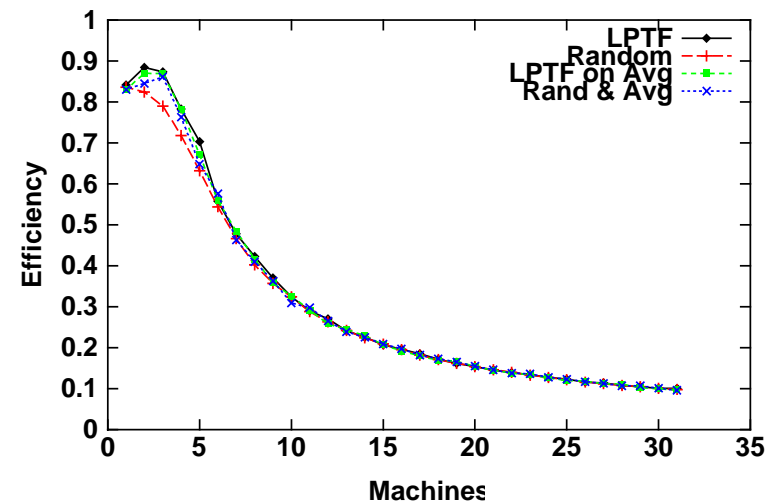
EFF ck=YES 30% 1-1 D=100 T=31 L=35



EFF ck=NO 90% 1-1 D=100 T=31 L=35



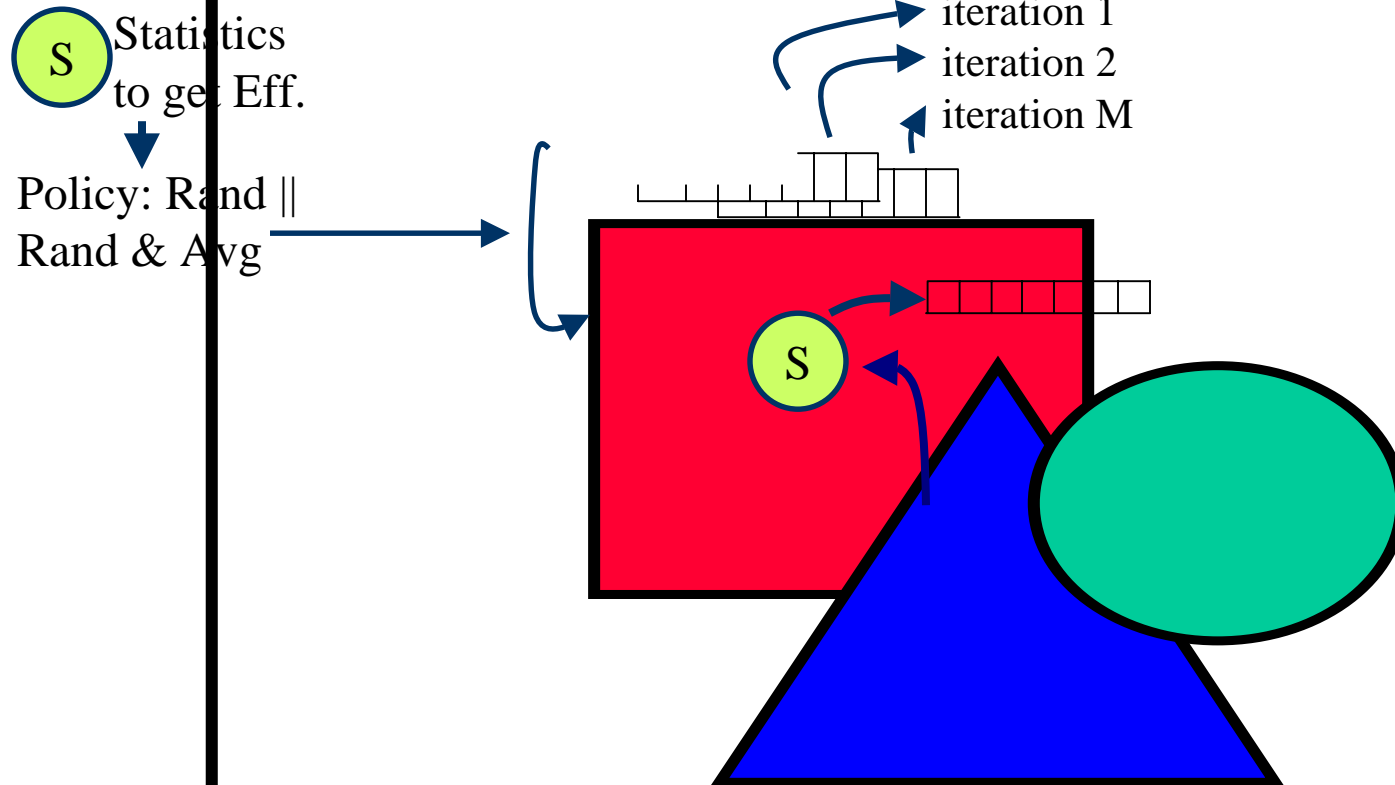
EFF ck=YES 90% 1-1 D=100 T=31 L=35



Implementation on MW

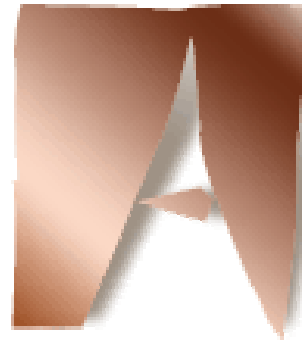
- Support to the desired Program Model.
- Computation of the Efficiency.
- Scheduling policies → Random
 - Random & Average

Implementation on MW



Future

- Non-dedicated homogeneous machines:
 - Complete the simulations.
 - Duplication of large tasks. (?)
- Non-dedicated heterogeneous machines:
 - Use dynamic load information (provided by Condor) to rank machines.
- Implementation on MW:
 - Test the MW scheduling policies with large applications.



**Universitat
Autònoma
de Barcelona**

UNIVERSITY OF
WISCONSIN
MADISON
