The impact of heterogeneity on master-slave on-line scheduling

Jean-François PINEAU, Yves ROBERT and Frédéric VIVIEN

Laboratoire de l'Informatique du Parallélisme
École Normale Supérieure de Lyon, France

Jean-Francois.Pineau@ens-lyon.fr

http://graal.ens-lyon.fr/~jfpineau

March 16, 2006
Outline

1. Scheduling
   - On-line competitiveness
     - Homogenous problem
     - Heterogeneous problem
     - General approach
     - Results

2. Experiments

3. Conclusion
The processors

- Parallel
  - Identical
  - Uniform
Background on Scheduling

The processors
- Parallel
  - Identical
  - Uniform

The tasks
- described by:
  - their amount of computation
  - their amount of communication
  - their release date
Background on Scheduling

The master
- Receive the tasks
- Send them to the processors
Goal

Scheduling tasks onto processors
- according to the constraints,
  - of the processors
  - of the tasks
- and optimizing some objective function
Background on Scheduling

Notations

- $n$ tasks, $m$ processors
- $p_{i,j}$: processing time of task $i$ on processor $j$
- $c_{i,j}$: sending time of task $i$ from master to processor $j$
- $r_i$: release date
- $C_i$: date of end of execution

Main objective functions:
- makespan: $\max C_i$
- maximum flow time: $\max (C_i - r_i)$
- average flow time: $\sum (C_i - r_i)$
Definition

An algorithm $\mathcal{X}$ has a lower bound on its competitive ratio of $\rho$ for the minimization of one objective function (for example makespan) if for one set of tasks:

$$(\max C_i)_{\mathcal{X}} \geq \rho (\max C_i)_{Opt}$$
Background on Scheduling

Let’s specify the problem

- Identical independent tasks,

Otherwise, problem NP-hard even for 2 processors.
Let’s specify the problem

- Identical independent tasks,
- Fast communications.
Let’s specify the problem

- Identical independent tasks,
- Fast communications.

If $c_{j_0} = \min c_j$ and $c_{j_0} > p_{j_0}$, then the optimal algorithm is trivial.
Outline

1. Scheduling

2. On-line competitiveness
   - Homogenous problem
   - Heterogeneous problem
   - General approach
   - Results

3. Experiments

4. Conclusion
Outline

1. Scheduling

2. On-line competitiveness
   - Homogenous problem
   - Heterogeneous problem
   - General approach
   - Results

3. Experiments

4. Conclusion
On homogeneous platforms

Round-Robin

is an optimal algorithm to minimize all three
- \textit{makespan},
- max flow time,
- sum flow time,

for an on-line problem with release dates.
Outline

1. Scheduling

2. On-line competitiveness
   - Homogeneous problem
   - Heterogeneous problem
   - General approach
   - Results

3. Experiments

4. Conclusion
On heterogeneous platforms

**Optimal algorithm**
does not exist, to minimize one objective function among
- makespan,
- max flow time,
- sum flow time,

This can be proved by an adversary method.
There is no scheduling algorithm for the problem $Q, MS | online, r_i, p_j, c_j = c | \max C_i$ with a competitive ratio less than $\frac{5}{4}$.

Jean-François Pineau (LIP)
Theorem

There is no scheduling algorithm for the problem $Q, MS \mid online, r_i, p_j, c_j = c \mid \max C_i$ with a competitive ratio less than $\frac{5}{4}$. 

Jean-François Pineau (LIP)

Heterogeneity & master-slave scheduling
Proof

1. Suppose the existence of an on-line algorithm $\mathcal{X}$ with a competitive ratio $\rho = \frac{5}{4} - \epsilon$, with $\epsilon > 0$.

2. Let’s study the behavior of $\mathcal{X}$ opposed to our adversary on a platform composed of two processors, where $p_1 = 3$, $p_2 = 7$, and $c = 1$. 

Jean-François Pineau (LIP)

On-line competitiveness  Heterogeneous problem

Heterogeneity & master-slave scheduling
Adversary sends a single task $i$ at time 0: best makespan = 4
At time $t_1 = c$, we check the decision of $X$. 
Adversary sends a single task $i$ at time 0: best makespan = 4
At time $t_1 = c$, we check the decision of $\mathcal{X}$.
- adversary does not send other tasks.
Adversary sends a single task $i$ at time 0: best makespan $= 4$

At time $t_1 = c$, we check the decision of $X$.

- adversary does not send other tasks.

competitive ratio: $\frac{t_1 + c + p_1}{4} = \frac{5}{4} > \rho$
Adversary sends a single task \( i \) at time 0: best makespan = 4
At time \( t_1 = c \), we check the decision of \( \mathcal{X} \).

- adversary does not send other tasks.
  competitive ratio : \( \frac{c + p_2}{4} = 2 > \rho \)
Proof

Adversary sends a single task $i$ at time 0: best makespan = 4
At time $t_1 = c$, we check the decision of $\mathcal{X}$.

- $\mathcal{X}$ has no choice but to schedule task $i$ on $P_1$ to enforce its competitive ratio.
At time $t_1 = c$, adversary sends task $j$. At time $t_2 = 2c$: 
At time $t_1 = c$, adversary sends task $j$. At time $t_2 = 2c$:

- adversary sends no more task.
- competitive ratio: $\frac{2c+p_2}{7} = \frac{9}{7} > \frac{5}{4} > \rho$. 

Optimal

Algo X

Comm = 1
$P_2 : p_2 = 7$
$P_1 : p_1 = 3$

At time $t_1 = c$, adversary sends task $j$. At time $t_2 = 2c$:

- adversary sends no more task.
- competitive ratio: $\frac{2c+p_2}{7} = \frac{9}{7} > \frac{5}{4} > \rho$. 

Optimal

Algo X

Comm = 1
$P_2 : p_2 = 7$
$P_1 : p_1 = 3$
At time $t_1 = c$, adversary sends task $j$. At time $t_2 = 2c$:

- adversary sends a last task at time $t_2 = 2c$. 

Jean-François Pineau (LIP)

Heterogeneity & master-slave scheduling
Proof

At time $t_1 = c$, adversary sends task $j$. At time $t_2 = 2c$:
- adversary sends a last task at time $t_2 = 2c$.
- competitive ratio: $\frac{10}{8} = \frac{5}{4} > \rho$. 

Jean-François Pineau (LIP)
At time $t_1 = c$, adversary sends task $j$. At time $t_2 = 2c$:  
- adversary sends a last task at time $t_2 = 2c$. 
Proof

At time $t_1 = c$, adversary sends task $j$. At time $t_2 = 2c$:

- adversary sends a last task at time $t_2 = 2c$. 
At time $t_1 = c$, adversary sends task $j$. At time $t_2 = 2c$:
- adversary sends a last task at time $t_2 = 2c$. 
Proof

At time $t_1 = c$, adversary sends task $j$. At time $t_2 = 2c$:

- adversary sends a last task at time $t_2 = 2c$.

competitive ratio : $\frac{10}{8} = \frac{5}{4} > \rho$. 

Jean-François Pineau (LIP)
Outline

1. Scheduling

2. On-line competitiveness
   - Homogenous problem
   - Heterogeneous problem
   - General approach
   - Results

3. Experiments

4. Conclusion
How does it work?

Let’s see how we find the worst platform for an on-line algorithm.

Example
- Fully heterogeneous platform
- Minimization of max flow
General approach

How does it work?
Let’s see how we find the worst platform for an on-line algorithm.

Example
- Fully heterogeneous platform
- Minimization of max flow
Generalisation

Idea:
- one fast processor with slow communication \((c_1 > 1)\);
- two slow identical processors with fast communication;
- if only one task, send it on fast processor \((c_1 + p_1 < 1 + p_2)\);
- if more than one task, do not send the first task on the fast processor.
On-line competitiveness

General approach

Generalisation

Idea:

- one fast processor with slow communication ($c_1 > 1$);
- two slow identical processors with fast communication;
- if only one task, send it on fast processor ($c_1 + p_1 < 1 + p_2$);
- if more than one task, do not send the first task on the fast processor.
At time $\tau \geq 1$ we look at what happened:
At time $\tau \geq 1$ we look at what happened:

1. Optimal: max flow $= c_1 + p_1$. 

$P_3(1, p_2)$

$P_2(1, p_2)$

$P_1(c_1, p_1)$
At time $\tau \geq 1$ we look at what happened:

1. Optimal: max flow $= c_1 + p_1$.
2. $\max \text{ flow} \geq \tau + c_1 + p_1$, ratio $\geq \frac{\tau + c_1 + p_1}{c_1 + p_1}$.
At time $\tau \geq 1$ we look at what happened:

1. Optimal: max flow = $c_1 + p_1$.
2. $\max\text{ flow} \geq \tau + c_1 + p_1$, ratio $\geq \frac{\tau + c_1 + p_1}{c_1 + p_1}$.
3. $\max\text{ flow} \geq 1 + p_2$, ratio $\geq \frac{1 + p_2}{c_1 + p_1}$. 

Jean-François Pineau (LIP)
On-line competitiveness

General approach

Generalisation

We choose $\tau$, $c_1$, $p_1$ and $p_2$ to have:

$$\min \left\{ \frac{1 + p_2}{c_1 + p_1}, \frac{\tau + c_1 + p_1}{c_1 + p_1} \right\} \geq \rho$$

So algorithm has to execute the first task on $P_1$. 
At time $\tau$ we send two new tasks. Let's see all possible schedulings.
three tasks on $P_1$:

$$\max\{c_1 + p_1, \max\{\max\{c_1, \tau\} + c_1 + p_1, c_1 + 2p_1\} - \tau, \max\{\max\{c_1, \tau\} + c_1 + p_1 + \max\{c_1, p_1\}, c_1 + 3p_1\} - \tau\}$$
Generalisation

Last task on $P_1$.

$$\max\{c_1 + p_1, \max\{\max\{c_1, \tau\} + c_2 + p_2\} - \tau, \max\{\max\{c_1, \tau\} + c_2 + c_1 + p_1, c_1 + 2p_1\} - \tau\}$$
First task on $P_1$.

\[
\max\{c_1 + p_1, \max\{\max\{c_1, \tau\} + c_1 + p_1, c_1 + 2p_1\} - \tau, \max\{c_1, \tau\} + c_1 + c_2 + p_2\} - \tau\}
\]
On-line competitiveness

General approach

Generalisation

No more tasks on $P_1$.

$$\max \{ c_1 + p_1, (\max \{ c_1, \tau \} + c_2 + p_2) - \tau, (\max \{ c_1, \tau \} + c_2 + c_2 + p_2) - \tau \}$$
The case where two tasks are allocated on $P_2$ is even worse than the previous case.
Better solution: 1\textsuperscript{st} task on $P_2$, 2\textsuperscript{nd} on $P_3$ and 3\textsuperscript{rd} on $P_1$.

$$\max\{c_2+p_2, (\max\{c_2, \tau\}+c_2+p_2)-\tau, (\max\{c_2, \tau\}+c_2+c_1+p_1)-\tau\}$$
How we found lower bound of competitiveness (1)

Lower bound of competitiveness:

\[
\min \left\{ \frac{\tau + c_1 + p_1}{c_1 + p_1}, \frac{1 + p_2}{c_1 + p_1}, \right. \\
\left. \quad \min \left\{ \max\{c_1 + p_1, \max\{\max\{c_1, \tau\} + c_1 + p_1, c_1 + 2p_1\} - \tau, \max\{\max\{c_1, \tau\} + c_1 + p_1 + \max\{c_1, p_1\}, c_1 + 3p_1\} - \tau\right\} \\
\right. \\
\left. \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \max\{c_1 + p_1, (\max\{c_1, \tau\} + c_2 + p_2) - \tau, \max\{\max\{c_1, \tau\} + c_2 + c_1 + p_1, c_1 + 2p_1\} - \tau\right\} \\
\right. \\
\left. \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \max\{c_1 + p_1, \max\{\max\{c_1, \tau\} + c_1 + p_1, c_1 + 2p_1\} - \tau, (\max\{c_1, \tau\} + c_1 + c_2 + p_2) - \tau\} \\
\right. \\
\left. \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \max\{c_1 + p_1, (\max\{c_1, \tau\} + c_2 + p_2) - \tau, (\max\{c_1, \tau\} + c_2 + c_2 + p_2) - \tau\} \\
\right. \\
\left. \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \max\{c_2 + p_2, (\max\{c_2, \tau\} + c_2 + c_1 + p_1) - \tau\} \right\} \\
\]

Problem

Find \( \tau, c_1, p_1 \) and \( p_2 \) (\( c_2 = 1 \)) which maximize this lower bound, such as: \( c_1 + p_1 < p_2 \).
How we found lower bound of competitiveness (1)

Lower bound of competitiveness:

\[
\min \begin{cases}
\frac{\tau + c_1 + p_1}{c_1 + p_1}, \\
\frac{1 + p_2}{c_1 + p_1}, \\
\max \{ c_1 + p_1, \max \{ \max \{ c_1, \tau \} + c_1 + 2p_1 \} - \tau, \max \{ \max \{ c_1, \tau \} + c_1 + p_1 + \max \{ c_1, p_1 \}, c_1 + 3p_1 \} - \tau \} \\
\max \{ c_1 + p_1, (\max \{ c_1, \tau \} + c_2 + p_2) - \tau, \max \{ \max \{ c_1, \tau \} + c_2 + c_1 + p_1, c_1 + 2p_1 \} - \tau \} \\
\max \{ c_1 + p_1, \max \{ \max \{ c_1, \tau \} + c_1 + p_1, c_1 + 2p_1 \} - \tau, (\max \{ c_1, \tau \} + c_1 + c_2 + p_2) - \tau \} \\
\max \{ c_1 + p_1, (\max \{ c_1, \tau \} + c_2 + p_2) - \tau, (\max \{ c_1, \tau \} + c_2 + c_2 + p_2) - \tau \} \\
\max \{ c_2 + p_2, (\max \{ c_2, \tau \} + c_2 + c_1 + p_1) - \tau \}
\end{cases}
\]

Problem

Find \( \tau, c_1, p_1 \) and \( p_2 \) (\( c_2 = 1 \)) which maximize this lower bound, such as: \( c_1 + p_1 < p_2 \).
How we found lower bound of competitiveness (2)

1 Numerical resolution
2 Characterization of optimal: $\tau < c_1$, $p_1 = 0$, etc.
3 New system:

$$\min \begin{cases} \frac{\tau + c_1}{c_1}, \\ \frac{1 + p_2}{c_1}, \\ \min \begin{cases} 3c_1 - \tau, \\ c_1 + 1 - \tau + p_2, \\ 2c_1 - \tau + 1 + p_2 \\ c_1 + 2 + p_2 - \tau \end{cases}, \\ \frac{\tau + c_1}{c_1}, \\ \frac{1 + p_2}{c_1}, \\ \frac{c_1 + 1 - \tau + p_2}{1 + p_2} \end{cases} = \min \begin{cases} \frac{\tau + c_1}{c_1}, \\ \frac{1 + p_2}{c_1}, \\ \frac{c_1 + 1 - \tau + p_2}{1 + p_2} \end{cases}$$

4 Solution: $c_1 = 2(1 + \sqrt{2})$, $p_2 = \sqrt{2}c_1 - 1$, $\tau = 2$, $\rho = \sqrt{2}$. 

Jean-François Pineau (LIP)
How we found lower bound of competitiveness (2)

1. Numerical resolution
2. Characterization of optimal: $\tau < c_1$, $p_1 = 0$, etc.
3. New system:

$$
\min \begin{cases}
\frac{\tau + c_1}{c_1}, \\
\frac{1 + p_2}{c_1}, \\
3c_1 - \tau, \\
\min \begin{cases}
c_1 + 1 - \tau + p_2, \\
2c_1 - \tau + 1 + p_2 \\
c_1 + 2 + p_2 - \tau \\
1 + p_2
\end{cases}
\end{cases}
= \min \begin{cases}
\frac{\tau + c_1}{c_1}, \\
\frac{1 + p_2}{c_1}, \\
\frac{c_1 + 1 - \tau + p_2}{1 + p_2}
\end{cases}
$$

4. Solution: $c_1 = 2(1 + \sqrt{2})$, $p_2 = \sqrt{2}c_1 - 1$, $\tau = 2$, $\rho = \sqrt{2}$. 
How we found lower bound of competitiveness (2)

1. Numerical resolution
2. Characterization of optimal: \( \tau < c_1, p_1 = 0 \), etc.
3. New system:

\[
\min \begin{cases} 
\frac{\tau + c_1}{c_1}, \\
\frac{1 + p_2}{c_1}, \\
\min \begin{cases} 
3c_1 - \tau, \\
c_1 + 1 - \tau + p_2, \\
2c_1 - \tau + 1 + p_2 \\
c_1 + 2 + p_2 - \tau \\
\frac{1}{1 + p_2}
\end{cases}, \\
\frac{\tau + c_1}{1 + p_2}, \\
\frac{1 + p_2}{1 + p_2}, \\
c_1 + 1 - \tau + p_2,
\end{cases}
\]

4. Solution: \( c_1 = 2(1 + \sqrt{2}), p_2 = \sqrt{2}c_1 - 1, \tau = 2, \rho = \sqrt{2} \).
How we found lower bound of competitiveness (2)

1. Numerical resolution
2. Characterization of optimal: \( \tau < c_1, p_1 = 0, \text{etc.} \)
3. New system:

\[
\begin{align*}
\min \left\{ \frac{\tau + c_1}{c_1}, \frac{1 + p_2}{c_1}, \min \left\{ 3c_1 - \tau, c_1 + 1 - \tau + p_2, 2c_1 - \tau + 1 + p_2 \right\} \right. \\
\left. \frac{c_1 + 2 + p_2 - \tau}{1 + p_2} \right\} = \min \left\{ \frac{\tau + c_1}{c_1}, \frac{1 + p_2}{c_1}, \frac{c_1 + 1 - \tau + p_2}{1 + p_2} \right\}
\end{align*}
\]

4. Solution: \( c_1 = 2(1 + \sqrt{2}), p_2 = \sqrt{2}c_1 - 1, \tau = 2, \rho = \sqrt{2} \).
Outline

1. Scheduling
2. On-line competitiveness
   - Homogenous problem
   - Heterogeneous problem
   - General approach
   - Results
3. Experiments
4. Conclusion
### On-line competitiveness

### Results

<table>
<thead>
<tr>
<th>Platform type</th>
<th>Objective function</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Makespan</td>
</tr>
<tr>
<td>Homogeneous</td>
<td></td>
</tr>
<tr>
<td>Communication homogeneous</td>
<td>$\frac{5}{4}$ = 1.250</td>
</tr>
<tr>
<td>Computation homogeneous</td>
<td>$\frac{6}{5}$ = 1.200</td>
</tr>
<tr>
<td>Heterogeneous</td>
<td>$\frac{1+\sqrt{3}}{2} \approx 1.366$</td>
</tr>
</tbody>
</table>

**Table:** Lower bounds on the competitive ratio of on-line algorithms, depending on the platform type and on the objective function.
Outline

1. Scheduling

2. On-line competitiveness
   - Homogenous problem
   - Heterogeneous problem
   - General approach
   - Results

3. Experiments

4. Conclusion
## The platform

### Hardware
- 5 computers (1 master, 4 slaves)
- 1 Fast-Ethernet switch

### Software
- MPI communications
- Modification of slave parameters
Algorithms

- **Algorithm 1** is a dynamic one
- **Algorithm 4 and 7** take into account communication heterogeneity
- **Algorithms 5 and 6** take into account computation heterogeneity
- **Algorithms 2 and 3** take into account both communication and computation heterogeneity

Algorithm 6 is optimal to minimize *makespan* if it knows the total number of tasks. Algorithm 7 is meant to be used on computation homogeneous platform.
Algorithm 1 is a dynamic one
Algorithm 4 and 7 take into account communication heterogeneity
Algorithms 5 and 6 take into account computation heterogeneity
Algorithms 2 and 3 take into account both communication and computation heterogeneity

Algorithm 6 is optimal to minimize *makespan* if it knows the total number of tasks.
Algorithm 7 is meant to be used on computation homogeneous platform.
Algorithms

- Algorithm 1 is a dynamic one
- Algorithm 4 and 7 take into account communication heterogeneity
- Algorithms 5 and 6 take into account computation heterogeneity
- Algorithms 2 and 3 take into account both communication and computation heterogeneity

Algorithm 6 is optimal to minimize *makespan* if it knows the total number of tasks.
Algorithm 7 is meant to be used on computation homogeneous platform
Experiments

Results

General case:

Figure: Normalized objective functions
Results

Homogeneous processors:

Figure: Normalized objective functions
The heuristic meant to be used on a communication heterogeneous platform is better than the other most part of the time (95%), and close to the best found algorithm (2%) elsewhere.

SLJF is outperformed by some classical algorithms.
Experiments

Results

Summary

- The heuristic meant to be used on a communication heterogeneous platform is better than the other most part of the time (95%), and close to the best found algorithm (2%) elsewhere.

- *SLJF* is outperformed by some classical algorithms.

Point out the importance to take into account the relative speed of communication links when searching a close-to-optimal solution to our scheduling problem.
Outline

1. Scheduling

2. On-line competitiveness
   - Homogenenous problem
   - Heterogeneous problem
   - General approach
   - Results

3. Experiments

4. Conclusion
Contributions and perspectives

Contributions

- Comprehensive set of lower bounds for the competitive ratio of any deterministic scheduling algorithm, for each source of heterogeneity and for each target objective function,
- Experiments on real small-size master-slave platform.

Perspectives

- See which bounds can be met, if any, and design the corresponding approximation algorithms,
- Theoretical study of off-line scheduling problems,
- Detailed comparison of all previous heuristics on significantly larger platforms,
- Widen the scope of the MPI experiments.
Contributions and perspectives

Contributions

- Comprehensive set of lower bounds for the competitive ratio of any deterministic scheduling algorithm, for each source of heterogeneity and for each target objective function,
- Experiments on real small-size master-slave platform.

Perspectives

- See which bounds can be met, if any, and design the corresponding approximation algorithms,
- Theoretical study of off-line scheduling problems,
- Detailed comparison of all previous heuristics on significantly larger platforms,
- Widen the scope of the MPI experiments.