# Offline and online master-worker scheduling of concurrent bags-of-tasks on heterogeneous platforms

Anne BENOIT, Loris MARCHAL, 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

Alpage, February 1, 2008

# Outline

# Outline

# Bag-of-tasks Applications

## Bag of tasks

described by:

- the number of independent identical tasks
- the amount of computation of a task
- the amount of communication of a task
- their release date

Online scheduling.

# Bag-of-tasks Applications

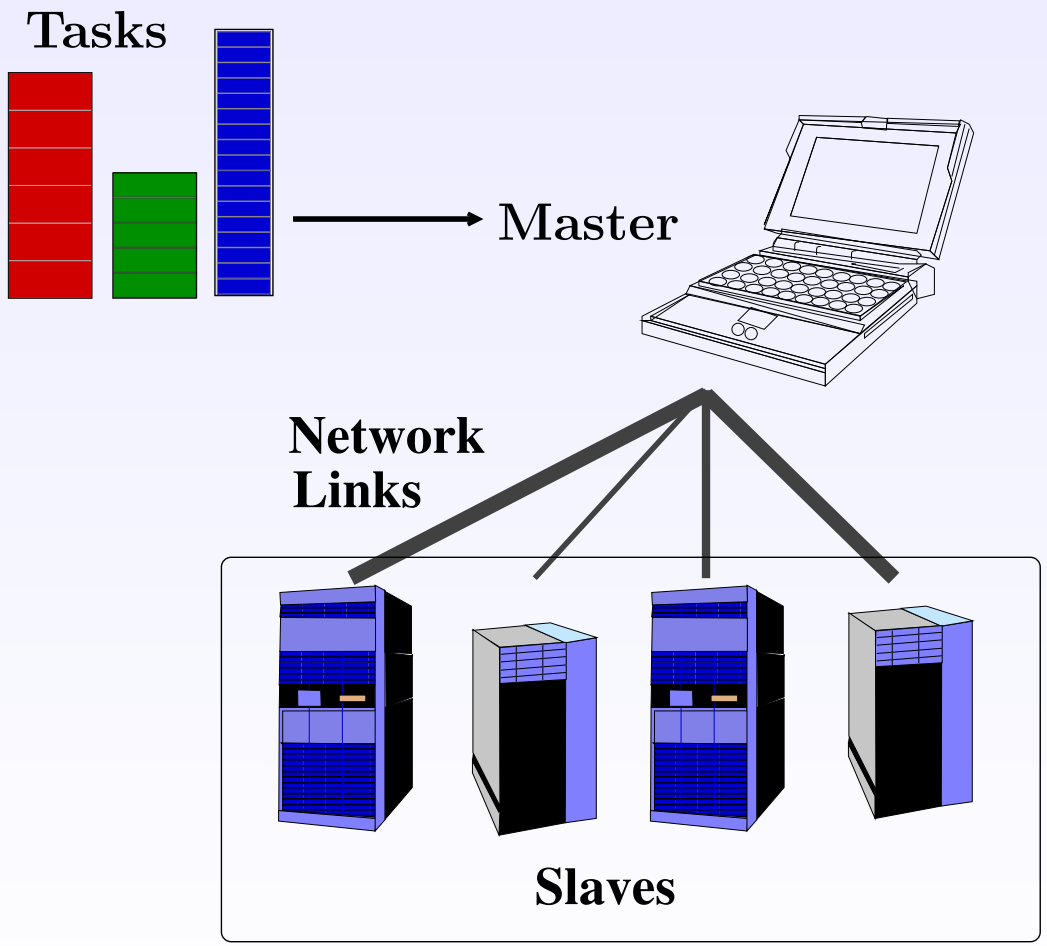
## Bag of tasks

described by:

- the number of independent identical tasks
- the amount of computation of a task
- the amount of communication of a task
- their release date

Online scheduling.

# Platform model



# Master-slaves platform

## The master

- Receive the bags of tasks
- Send the tasks to the processors
- Bounded multi-port model

## The processors

- Parallels
  - Identical
  - Uniform
- Related

# Master-slaves platform

## The master

- Receive the bags of tasks
- Send the tasks to the processors
- Bounded multi-port model

## The processors

- Parallels
  - Identical
  - Uniform
- Related

# Notations

## Tasks

- $n$ bags-of-tasks applications $\mathcal{A}_k$
- $\mathcal{A}_i$ is composed of $\Pi^{(i)}$ tasks.
- $\omega^{(i)}$: amount of computation of a task of $\mathcal{A}_i$
- $\delta^{(i)}$: amount of communication of a task of $\mathcal{A}_i$
- $r^{(i)}$: release date of $\mathcal{A}_i$
- $\mathcal{C}^{(i)}$: completion date of $\mathcal{A}_i$

# Notations

## Platform

- $p$ processors,
- $\mathcal{B}$: bound of the multi-port model.
- $b_u$: bandwidth of the link between the master and $P_u$,
- $s_u$: computational speed of worker $P_u$,

# Notations

## Platform

- $p$ processors,
- $\mathcal{B}$: bound of the multi-port model.
- $b_u$: bandwidth of the link between the master and $P_u$,
- $s_u^{(k)}$: computational speed of related worker $P_u$ with tasks of $\mathcal{A}_k$,

# Objective function

## Objective function

- Makespan

$$\max \ \mathcal{C}^{(i)} \text{ or } \mathcal{C}^{(max)}$$

# Objective function

## Objective function

- Makespan

$$\max \ \mathcal{C}^{(i)} \text{ or } \mathcal{C}^{(max)}$$

*Problem of satisfaction of the clients*

# Objective function

## Objective function

- ~~Makespan~~
- Sum flow

$$\sum \{C^{(i)} - r^{(i)}\}$$

# Objective function

## Objective function

- ~~Makespan~~
- Sum flow

$$\sum \{C^{(i)} - r^{(i)}\}$$

*Problem of starvation*

# Objective function

## Objective function

- ~~Makespan~~
- ~~Sum flow~~
- Max flow

$$\max \{C^{(i)} - r^{(i)}\}$$

# Objective function

## Objective function

- ~~Makespan~~
- ~~Sum flow~~
- ~~Max flow~~

$$\max \{C^{(i)} - r^{(i)}\}$$

*Small applications can wait a long time*

# Objective function

## Objective function

- ~~Makespan~~
- ~~Sum flow~~
- ~~Max flow~~
- Max Stretch

$$\max \frac{\mathcal{C}^{(i)} - r^{(i)}}{Size\ of\ \mathcal{A}_i}$$

# Objective function

## Objective function

- ~~Makespan~~
- ~~Sum flow~~
- ~~Max flow~~
- Max Stretch

$$\max \frac{\mathcal{C}^{(i)} - r^{(i)}}{\textit{Size of } \mathcal{A}_i}$$

$\textit{Size of } \mathcal{A}_i = \Pi^{(i)}$ ?

# Objective function

## Objective function

- ~~Makespan~~
- ~~Sum flow~~
- ~~Max flow~~
- Max Stretch

$$\max \frac{\mathcal{C}^{(i)} - r^{(i)}}{\textit{Size of } \mathcal{A}_i}$$

$\textit{Size of } \mathcal{A}_i = \omega^{(i)}$ ?

# Objective function

## Objective function

- ~~Makespan~~
- ~~Sum flow~~
- ~~Max flow~~
- Max Stretch

$$\max \frac{\mathcal{C}^{(i)} - r^{(i)}}{\textit{Size of } \mathcal{A}_i}$$

$$\textit{Size of } \mathcal{A}_i = \Pi^{(i)} \times \omega^{(i)} \ ?$$

# Objective function

## Objective function

- ~~Makespan~~
- ~~Sum flow~~
- ~~Max flow~~
- Max Stretch

$$\max \frac{\mathcal{C}^{(i)} - r^{(i)}}{Size\ of\ \mathcal{A}_i}$$

$Size\ of\ \mathcal{A}_i$ =Makespan $MS^{*(i)}$ of $\mathcal{A}_i$ if alone

# $MS^{*(0)}$

**Problem**

- Unique bag-of-tasks $\mathcal{A}_0$
- Large $\Pi^{(0)}$

# $MS^{*(0)}$

**Problem**

- Unique bag-of-tasks $\mathcal{A}_0$
- Large $\Pi^{(0)}$

**Objective**

- Minimizing the makespan

# $MS^{*(0)}$

**Problem**

- Unique bag-of-tasks $\mathcal{A}_0$
- Large $\Pi^{(0)}$

**Objective**

- ~~Minimizing the makespan~~
- Maximizing the throughput

# $MS^{*(0)}$

## Problem

- Unique bag-of-tasks $\mathcal{A}_0$
- Large $\Pi^{(0)}$

## Objective

- ~~Minimizing the makespan~~
- Maximizing the throughput
- Throughput of worker $P_u$: $\rho_u^{*(0)}$
- Total throughput $\rho^{*(0)} = \sum_{u=1}^{p} \rho_u^{*(0)}$

# Linear program

$$\begin{cases} \text{MAXIMIZE } \rho^{*(0)} = \sum_{u=1}^{p} \rho_u^{*(0)} \\ \text{SUBJECT TO} \\ \rho_u^{*(0)} \frac{\omega^{(0)}}{s_u^{(0)}} \leq 1 \\ \rho_u^{*(0)} \frac{\delta^{(0)}}{b_u} \leq 1 \\ \sum_{u=1}^{p} \rho_u^{*(0)} \frac{\delta^{(0)}}{\mathcal{B}} \leq 1 \end{cases} \tag{1}$$

Rational solution

$$\rho^{*(0)} = \min\left\{ \frac{\mathcal{B}}{\delta^{(0)}}, \sum_{u=1}^{p} \min\left\{ \frac{s_u^{(0)}}{\omega^{(0)}}, \frac{b_u}{\delta^{(0)}} \right\} \right\}.$$

# Linear program

$$\begin{cases} \text{MAXIMIZE } \rho^{*(0)} = \sum_{u=1}^{p} \rho_u^{*(0)} \\ \text{SUBJECT TO} \\ \rho_u^{*(0)} \frac{\omega^{(0)}}{s_u^{(0)}} \leq 1 \\ \rho_u^{*(0)} \frac{\delta^{(0)}}{b_u} \leq 1 \\ \sum_{u=1}^{p} \rho_u^{*(0)} \frac{\delta^{(0)}}{\mathcal{B}} \leq 1 \end{cases} \quad (1)$$

Rational solution

$$\rho^{*(0)} = \min \left\{ \frac{\mathcal{B}}{\delta^{(0)}}, \sum_{u=1}^{p} \min \left\{ \frac{s_u^{(0)}}{\omega^{(0)}}, \frac{b_u}{\delta^{(0)}} \right\} \right\}.$$

# Feasible schedule

Resource selection ($\rho_u^{*(0)} = 0$)

# Feasible schedule

Resource selection ($\rho_u^{*(0)} = 0$)

In theory:

- While there are tasks to process on the master, send tasks to processor $P_u$ with rate $\rho_u^{*(0)}$.
- As soon as processor $P_u$ starts receiving a task it processes at the rate $\rho_u^{*(0)}$.

# Feasible schedule

Resource selection ($\rho_u^{*(0)} = 0$)

In theory:

- While there are tasks to process on the master, send tasks to processor $P_u$ with rate $\rho_u^{*(0)}$.
- As soon as processor $P_u$ starts receiving a task it processes at the rate $\rho_u^{*(0)}$.

# Feasible schedule

Resource selection ($\rho_u^{*(0)} = 0$)

In practice, master uses the 1D-load balancing algorithm:

- the first worker to receive a task is the one with largest throughput
- each participating worker $P_u$ has already received $n_u$ tasks, the next worker to receive a task is chosen as the one minimizing

$$\frac{n_u + 1}{\rho_u^{*(0)}}$$

# Back on multi-applications problem

Approximation of the best execution time:

$$MS^{*(k)} = \frac{\Pi^{(k)}}{\rho^{*(k)}}.$$

Real execution time:

$$C^{(k)} = r^{(k)} + MS^{(k)}$$

In general:

$$MS^{(k)} \geq MS^{*(k)}$$

# Back on multi-applications problem

Approximation of the best execution time:

$$MS^{*(k)} = \frac{\Pi^{(k)}}{\rho^{*(k)}}.$$

Real execution time:

$$\mathcal{C}^{(k)} = r^{(k)} + MS^{(k)}$$

In general:

$$MS^{(k)} \geq MS^{*(k)}$$

# Back on multi-applications problem

Approximation of the best execution time:

$$MS^{*(k)} = \frac{\Pi^{(k)}}{\rho^{*(k)}}.$$

Real execution time:

$$\mathcal{C}^{(k)} = r^{(k)} + MS^{(k)}$$

In general:

$$MS^{(k)} \geq MS^{*(k)}$$

# Stretch

Stretch:

$$\mathcal{S}^k = \frac{MS^{(k)}}{MS^{*(k)}}$$

Throughput $\rho^{(k)}$ defined by:

$$MS^{(k)} = \frac{\Pi^{(k)}}{\rho^{(k)}}.$$

**Objective**: max-stretch:

$$\mathcal{S} = \max_{1 \leq k \leq n} \mathcal{S}^k$$

# Stretch

Stretch:

$$\mathcal{S}^k = \frac{MS^{(k)}}{MS^{*(k)}} = \frac{\rho^{*(k)}}{\rho^{(k)}}$$

Throughput $\rho^{(k)}$ defined by:

$$MS^{(k)} = \frac{\Pi^{(k)}}{\rho^{(k)}}.$$

**Objective**: max-stretch:

$$\mathcal{S} = \max_{1 \leq k \leq n} \mathcal{S}^k$$

# Stretch

Stretch:

$$\mathcal{S}^k = \frac{MS^{(k)}}{MS^{*(k)}} = \frac{\rho^{*(k)}}{\rho^{(k)}}$$

Throughput $\rho^{(k)}$ defined by:

$$MS^{(k)} = \frac{\Pi^{(k)}}{\rho^{(k)}}.$$

**Objective**: max-stretch:

$$\mathcal{S} = \max_{1 \leq k \leq n} \mathcal{S}^k$$

# Outline

# Offline

- Computing all the $MS^{*(k)}$, $\forall\ 1 \leq k \leq n$
- Binary search on the max-stretch
- For each candidate value $\mathcal{S}'$, we know that:

$$\forall\ 1 \leq k \leq n,\ \frac{MS^{(k)}}{MS^{*(k)}} \leq \mathcal{S}'$$

$$\forall\ 1 \leq k \leq n,\ C^{(k)} = r^{(k)} + MS^{(k)} \leq r^{(k)} + \mathcal{S}' \times MS^{*(k)}$$

# Offline

- Computing all the $MS^{*(k)}$, $\forall\ 1 \leq k \leq n$
- Binary search on the max-stretch
- For each candidate value $\mathcal{S}'$, we know that:

$$\forall\ 1 \leq k \leq n,\ \frac{MS^{(k)}}{MS^{*(k)}} \leq \mathcal{S}'$$

$$\forall\ 1 \leq k \leq n,\ C^{(k)} = r^{(k)} + MS^{(k)} \leq r^{(k)} + \mathcal{S}' \times MS^{*(k)}$$

# Offline

- Computing all the $MS^{*(k)}$, $\forall\ 1 \leq k \leq n$
- Binary search on the max-stretch
- For each candidate value $\mathcal{S}'$, we know that:

$$\forall\ 1 \leq k \leq n,\ \frac{MS^{(k)}}{MS^{*(k)}} \leq \mathcal{S}'$$

$$\forall\ 1 \leq k \leq n,\ C^{(k)} = r^{(k)} + MS^{(k)} \leq r^{(k)} + \mathcal{S}' \times MS^{*(k)}$$

# Offline

- Computing all the $MS^{*(k)}$, $\forall\ 1 \leq k \leq n$
- Binary search on the max-stretch
- For each candidate value $\mathcal{S}'$, we know that:

$$\forall\ 1 \leq k \leq n,\ \frac{MS^{(k)}}{MS^{*(k)}} \leq \mathcal{S}'$$

$$\forall\ 1 \leq k \leq n,\ \mathcal{C}^{(k)} = r^{(k)} + MS^{(k)} \leq r^{(k)} + \mathcal{S}' \times MS^{*(k)}$$

# Deadlines

We set:

$$d^{(k)} = r^{(k)} + \mathcal{S}' \times MS^{*(k)} \tag{2}$$

**Definition:** Epochal times

$$t_j \in \{r^{(1)}, ..., r^{(n)}\} \cup \{d^{(1)}, ..., d^{(n)}\}$$

such that

$$t_j \leq t_{j+1},\ 1 \leq j \leq 2n-1$$

Divide the total execution time into intervals whose bounds are epochal times.

# Deadlines

We set:

$$d^{(k)} = r^{(k)} + \mathcal{S}' \times MS^{*(k)} \tag{2}$$

**Definition:** Epochal times

$$t_j \in \{r^{(1)}, ..., r^{(n)}\} \cup \{d^{(1)}, ..., d^{(n)}\}$$

such that

$$t_j \leq t_{j+1},\ 1 \leq j \leq 2n-1$$

Divide the total execution time into intervals whose bounds are epochal times.

# Deadlines

We set:

$$d^{(k)} = r^{(k)} + \mathcal{S}' \times MS^{*(k)} \tag{2}$$

**Definition:** Epochal times

$$t_j \in \{r^{(1)}, ..., r^{(n)}\} \cup \{d^{(1)}, ..., d^{(n)}\}$$

such that

$$t_j \leq t_{j+1},\ 1 \leq j \leq 2n-1$$

Divide the total execution time into intervals whose bounds are epochal times.

# Intervals

- run each application $\mathcal{A}_k$ during its whole execution window $[r^{(k)}, d^{(k)}]$,
- use a different throughput on each interval $[t_j, t_{j+1}]$, $r^{(k)} \leq t_j$ and $t_{j+1} \leq d^{(k)}$.
- for communication
- for computation

# Intervals

- run each application $\mathcal{A}_k$ during its whole execution window $[r^{(k)}, d^{(k)}]$,
- use a different throughput on each interval $[t_j, t_{j+1}]$, $r^{(k)} \leq t_j$ and $t_{j+1} \leq d^{(k)}$.
- for communication
- for computation

# Intervals

- run each application $\mathcal{A}_k$ during its whole execution window $[r^{(k)}, d^{(k)}]$,
- use a different throughput on each interval $[t_j, t_{j+1}]$, $r^{(k)} \leq t_j$ and $t_{j+1} \leq d^{(k)}$.

- for communication
- for computation

# Intervals

- run each application $\mathcal{A}_k$ during its whole execution window $[r^{(k)}, d^{(k)}]$,
- use a different throughput on each interval $[t_j, t_{j+1}]$, $r^{(k)} \leq t_j$ and $t_{j+1} \leq d^{(k)}$.

- for communication

- for computation

# Intervals

- run each application $\mathcal{A}_k$ during its whole execution window $[r^{(k)}, d^{(k)}]$,
- use a different throughput on each interval $[t_j, t_{j+1}]$, $r^{(k)} \leq t_j$ and $t_{j+1} \leq d^{(k)}$.

- for communication
- for computation
- state of buffers

# Intervals

- run each application $\mathcal{A}_k$ during its whole execution window $[r^{(k)}, d^{(k)}]$,
- use a different throughput on each interval $[t_j, t_{j+1}]$, $r^{(k)} \leq t_j$ and $t_{j+1} \leq d^{(k)}$.

- for communication : $\rho^{(k)}_{M \to u}(t_j, t_{j+1})$
- for computation : $\rho^{(k)}_{u}(t_j, t_{j+1})$
- state of buffers at time $t_j$: $B^{(k)}_u(t_j)$

◂ Short version

# Positive values

- **Non-negative throughputs.**

$$\forall 1 \leq u \leq p, \forall 1 \leq k \leq n, \forall 1 \leq j \leq 2n-1,$$
$$\rho^{(k)}_{M \to u}(t_j, t_{j+1}) \geq 0 \text{ and } \rho^{(k)}_{u}(t_j, t_{j+1}) \geq 0. \quad (3)$$

- **Non-negative buffers.**

$$\forall\ 1 \leq k \leq n, \forall 1 \leq u \leq p, \forall 1 \leq j \leq 2n,$$
$$B^{(k)}_{u}(t_j) \geq 0. \quad (4)$$

# Physical constraints

- **Bounded link capacity.**

$$\forall 1 \leq j \leq 2n-1, \forall 1 \leq u \leq p,$$

$$\sum_{k=1}^{n} \rho_{M \to u}^{(k)}(t_j, t_{j+1}) \frac{\delta^{(k)}}{b_u} \leq 1. \quad (5)$$

- **Limited sending capacity of master.**

$$\forall 1 \leq j \leq 2n-1,$$

$$\sum_{u=1}^{p} \sum_{k=1}^{n} \rho_{M \to u}^{(k)}(t_j, t_{j+1}) \frac{\delta^{(k)}}{\mathcal{B}} \leq 1. \quad (6)$$

- **Bounded computing capacity.**

$$\forall 1 \leq j \leq 2n-1, \forall 1 \leq u \leq p,$$

$$\sum_{k=1}^{n} \rho_{u}^{(k)}(t_j, t_{j+1}) \frac{\omega^{(k)}}{s_u^{(k)}} \leq 1. \quad (7)$$

# Buffer constraints

- **Buffer initialization.**

$$\forall\ 1 \leq k \leq n, \forall 1 \leq u \leq p,$$

$$B_u^{(k)}(r^{(k)}) = 0. \quad (8)$$

- **Emptying Buffer.**

$$\forall\ 1 \leq k \leq n, \forall 1 \leq u \leq p,$$

$$B_u^{(k)}(d^{(k)}) = 0. \quad (9)$$

- **Bounded size**

$$\forall 1 \leq u \leq p, \forall 1 \leq j \leq 2n,$$

$$\sum_{k=1}^{n} B_u^{(k)}(t_j)\delta^{(k)} \leq M_u. \quad (10)$$

# Tasks constraints

- **Task conservation.**

$$\forall\ 1 \leq k \leq n, \forall 1 \leq j \leq 2n-1, \forall 1 \leq u \leq p,$$
$$B_u^{(k)}(t_{j+1}) = B_u^{(k)}(t_j) + \left(\rho_{M\to u}^{(k)}(t_j, t_{j+1}) - \rho_u^{(k)}(t_j, t_{j+1})\right) \times \left(t_{j+1} - t_j\right). \quad (11)$$

- **Total number of tasks.**

$$\forall\ 1 \leq k \leq n,$$
$$\sum_{\substack{1\leq j\leq 2n-1 \\ t_j \geq r^{(k)} \\ t_{j+1} \leq d^{(k)}}} \sum_{u=1}^{p} \rho_{M\to u}^{(k)}(t_j, t_{j+1}) \times (t_{j+1} - t_j) = \Pi^{(k)}. \quad (12)$$

# Polyhedron

$$\begin{cases} \text{find } \rho^{(k)}_{M\to u}(t_j, t_{j+1}), \rho^{(k)}_{u}(t_j, t_{j+1}), \\ \forall k, u, j \text{ such that } 1 \leq k \leq n, 1 \leq u \leq p, 1 \leq j \leq 2n-1 \\ \text{under the constraints } (3), (4), (5), (6), (7), (8), (9), (10), (11) \text{ and } (12) \end{cases} \quad (K)$$

A given max-stretch $\mathcal{S}'$ is achievable if and only if the Polyhedron $(K)$ is not empty

In practice, we add a fictitious linear objective function.

# Polyhedron

$$\begin{cases} \text{find } \rho^{(k)}_{M\to u}(t_j, t_{j+1}), \rho^{(k)}_{u}(t_j, t_{j+1}), \\ \forall k, u, j \text{ such that } 1 \leq k \leq n, 1 \leq u \leq p, 1 \leq j \leq 2n-1 \\ \text{under the constraints } (3), (4), (5), (6), (7), (8), (9), (10), (11) \text{ and } (12) \end{cases} \quad (K)$$

A given max-stretch $\mathcal{S}'$ is achievable if and only if the Polyhedron $(K)$ is not empty

In practice, we add a fictitious linear objective function.

# Polyhedron

$$\begin{cases} \text{find } \rho^{(k)}_{M\to u}(t_j, t_{j+1}), \rho^{(k)}_{u}(t_j, t_{j+1}), \\ \forall k, u, j \text{ such that } 1 \leq k \leq n, 1 \leq u \leq p, 1 \leq j \leq 2n-1 \\ \text{under the constraints } (3), (4), (5), (6), (7), (8), (9), (10), (11) \text{ and } (12) \end{cases} \quad (K)$$

A given max-stretch $\mathcal{S}'$ is achievable if and only if the Polyhedron $(K)$ is not empty

In practice, we add a fictitious linear objective function.

# Algorithm

1: Computing all the $MS^{*(k)}$, $\forall\ 1 \leq k \leq n$
2: $\mathcal{S}_{\text{inf}} \leftarrow 1$
3: $\mathcal{S}_{\text{sup}} \leftarrow \mathcal{S}_{\text{max}}$
4: **while** $\mathcal{S}_{\text{sup}} - \mathcal{S}_{\text{inf}} > \epsilon$ **do**
5: $\quad \mathcal{S} \leftarrow (\mathcal{S}_{\text{sup}} + \mathcal{S}_{\text{inf}})/2$
6: $\quad$ **if** Polyhedron $(K)$ is empty **then**
7: $\quad\quad \mathcal{S}_{\text{inf}} \leftarrow \mathcal{S}$
8: $\quad$ **else**
9: $\quad\quad \mathcal{S}_{\text{sup}} \leftarrow \mathcal{S}$
10: Return $\mathcal{S}_{\text{sup}}$

**Theorem**

The previous algorithm finds the optimal max-stretch in polynomial time.

# Algorithm

1: Computing all the $MS^{*(k)}$, $\forall\ 1 \leq k \leq n$
2: $\mathcal{S}_{\text{inf}} \leftarrow 1$
3: $\mathcal{S}_{\text{sup}} \leftarrow \mathcal{S}_{\text{max}}$
4: **while** $\mathcal{S}_{\text{sup}} - \mathcal{S}_{\text{inf}} > \epsilon$ **do**
5: $\quad \mathcal{S} \leftarrow (\mathcal{S}_{\text{sup}} + \mathcal{S}_{\text{inf}})/2$
6: $\quad$ **if** Polyhedron $(K)$ is empty **then**
7: $\quad\quad \mathcal{S}_{\text{inf}} \leftarrow \mathcal{S}$
8: $\quad$ **else**
9: $\quad\quad \mathcal{S}_{\text{sup}} \leftarrow \mathcal{S}$
10: Return $\mathcal{S}_{\text{sup}}$

**Theorem**

The previous algorithm finds the optimal max-stretch in polynomial time.

# Stretch-intervals

$$d^{(k)}(\mathcal{S}) = r^{(k)} + \mathcal{S} \times MS^{*(k)}.$$

# Stretch-intervals

$$d^{(k)}(\mathcal{S}) = r^{(k)} + \mathcal{S} \times MS^{*(k)}.$$

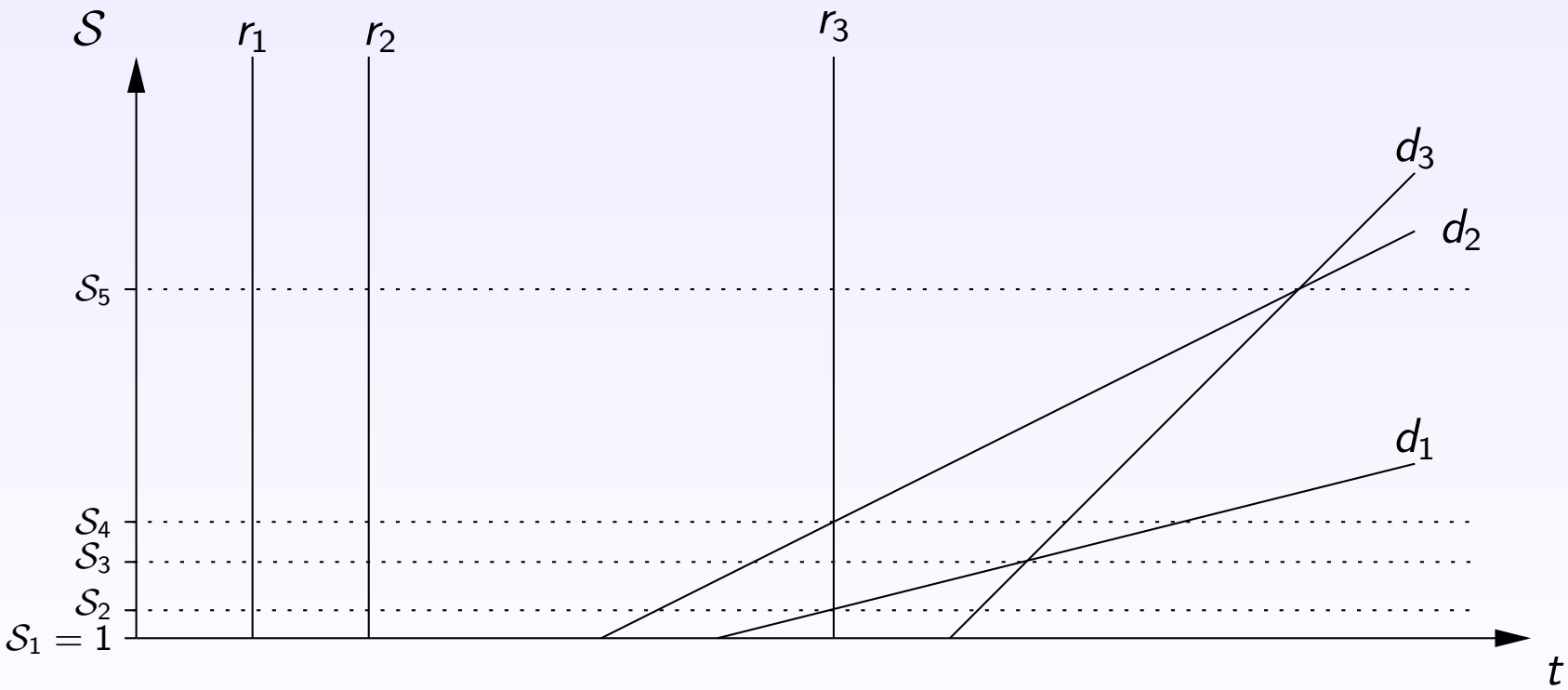



Figure: Relation between stretch and deadlines

# Notations

**Problem:** Quadratic constraints!

New notations:

$$A^{(k)}_{M\to u}(t_j, t_{j+1}) = \rho^{(k)}_{M\to u}(t_j, t_{j+1}) \times (t_{j+1} - t_j)$$
$$A^{(k)}_{u}(t_j, t_{j+1}) = \rho^{(k)}_{u}(t_j, t_{j+1}) \times (t_{j+1} - t_j)$$

# Notations

**Problem:** Quadratic constraints!

New notations:

$$\begin{aligned} A^{(k)}_{M\to u}(t_j, t_{j+1}) &= \rho^{(k)}_{M\to u}(t_j, t_{j+1}) \times (t_{j+1} - t_j) \\ A^{(k)}_{u}(t_j, t_{j+1}) &= \rho^{(k)}_{u}(t_j, t_{j+1}) \times (t_{j+1} - t_j) \end{aligned}$$

# New constraints

- **Bounded link capacity.**

$$\forall 1 \leq j \leq 2n-1, \forall 1 \leq u \leq p,$$
$$\sum_{k=1}^{n} A_{M \to u}^{(k)}(t_j, t_{j+1}) \frac{\delta^{(k)}}{b_u} \leq (\alpha_{j+1} - \alpha_j)\mathcal{S} + (\beta_{j+1} - \beta_j)$$

# New constraints

- **Bounded link capacity.**
- **Limited sending capacity of master.**

$$\forall 1 \leq j \leq 2n-1,$$
$$\sum_{u=1}^{p}\sum_{k=1}^{n} A_{M\to u}^{(k)}(t_j, t_{j+1})\delta^{(k)} \leq \mathcal{B} \times \left((\alpha_{j+1}-\alpha_j)\mathcal{S} + (\beta_{j+1}-\beta_j)\right)$$

# New constraints

- **Bounded link capacity.**
- **Limited sending capacity of master.**
- **Bounded computing capacity.**

$$\forall 1 \leq j \leq 2n-1, \forall 1 \leq u \leq p,$$
$$\sum_{k=1}^{n} A_u^{(k)}(t_j, t_{j+1}) \frac{\omega^{(k)}}{s_u^{(k)}} \leq (\alpha_{j+1} - \alpha_j)\mathcal{S} + (\beta_{j+1} - \beta_j)$$

# New constraints

- **Bounded link capacity.**
- **Limited sending capacity of master.**
- **Bounded computing capacity.**
- **Total number of tasks.**

$$\forall\ 1 \leq k \leq n,$$

$$\sum_{\substack{1 \leq j \leq 2n-1 \\ t_j \geq r^{(k)} \\ t_{j+1} \leq d^{(k)}}} \sum_{u=1}^{p} A_{M \to u}^{(k)}(t_j, t_{j+1}) = \Pi^{(k)}$$

# New constraints

- **Bounded link capacity.**
- **Limited sending capacity of master.**
- **Bounded computing capacity.**
- **Total number of tasks.**
- **Task conservation.**

$$\forall\ 1 \leq k \leq n, \forall 1 \leq j \leq 2n-1, \forall 1 \leq u \leq p,$$
$$B_u^{(k)}(t_{j+1}) = B_u^{(k)}(t_j) + A_{M\to u}^{(k)}(t_j, t_{j+1}) - A_u^{(k)}(t_j, t_{j+1})$$

# New constraints

- **Bounded link capacity.**
- **Limited sending capacity of master.**
- **Bounded computing capacity.**
- **Total number of tasks.**
- **Task conservation.**
- **Non-negative buffer.**
- **Buffer initialization.**
- **Emptying Buffer.**

# New constraints

- **Bounded link capacity.**
- **Limited sending capacity of master.**
- **Bounded computing capacity.**
- **Total number of tasks.**
- **Task conservation.**
- **Non-negative buffer.**
- **Buffer initialization.**
- **Emptying Buffer.**
- **Bounded stretch**

$$\mathcal{S}_a \leq \mathcal{S} \leq \mathcal{S}_b \tag{13}$$

# Linear programm

$$(\text{LP}) \begin{cases} \text{Minimize } \mathcal{S}, \\ \text{under all constraints} \end{cases}$$

At most $n(n-1)$ stretch intervals

# Linear programm

$$(\text{LP}) \begin{cases} \text{Minimize } \mathcal{S}, \\ \text{under all constraints} \end{cases}$$

At most $n(n-1)$ stretch intervals

# Algorithm offline

$L \leftarrow 1$ and $U \leftarrow \max$
**while** $U - L > 1$ **do**
  $M \leftarrow \left\lfloor \frac{L+U}{2} \right\rfloor$
  Solve the linear program (LP) for interval $[\mathcal{S}_M, \mathcal{S}_{M+1}]$
  **if** there is a solution with objective value $\mathcal{S}_{\text{opt}}$ **then**
    **if** $\mathcal{S}_{\text{opt}} > \mathcal{S}_M$ **then**
      Return $\mathcal{S}_{\text{opt}}$
    **else**
      $U \leftarrow M$
  **else**
    $L \leftarrow M$
Solve the linear program (LP) for interval $[\mathcal{S}_L, \mathcal{S}_U]$
Return the objective value $\mathcal{S}_{\text{opt}}$ of the solution

# Online

Offline algorithm at each release dates.
For each application $\mathcal{A}_k$:

- update $\Pi^{(k)}$
- update $MS^{*(k)}$
- determine the new optimal stretch that can be achieved as in the offline case

# Outline

# Experiments

## Hardware

- 1 master SuperMicro servers 6013PI, with processors P4 Xeon 2.4 GHz;
- 8 workers SuperMicro servers 5013-GM, with processors P4 2.4 GHz;
- 100Mbps Fast-Ethernet switch

## Software

- MPI communications
- Modification of slave parameters

The linear programs are solved using `glpk`.

# Experiments

## Hardware

- 1 master SuperMicro servers 6013PI, with processors P4 Xeon 2.4 GHz;
- 8 workers SuperMicro servers 5013-GM, with processors P4 2.4 GHz;
- 100Mbps Fast-Ethernet switch

## Software

- MPI communications
- Modification of slave parameters

The linear programs are solved using `glpk`.

# Experiments

## Hardware

- 1 master SuperMicro servers 6013PI, with processors P4 Xeon 2.4 GHz;
- 8 workers SuperMicro servers 5013-GM, with processors P4 2.4 GHz;
- 100Mbps Fast-Ethernet switch

## Software

- MPI communications
- Modification of slave parameters

The linear programs are solved using `glpk`.

# Simulations

- Simulated platforms as close as possible to actual experimental framework
- Experiments in 2 steps:
  - use of the same platform configuration and application scenario than during MPI experiments,
  - launch extensive set of simulations with larger applications

# Simulations

- Simulated platforms as close as possible to actual experimental framework
- Experiments in 2 steps:
  - use of the same platform configuration and application scenario than during MPI experiments,
  - launch extensive set of simulations with larger applications

# Simulations

- Simulated platforms as close as possible to actual experimental framework
- Experiments in 2 steps:
  - use of the same platform configuration and application scenario than during MPI experiments,
  - launch extensive set of simulations with larger applications

# Simulations

- Simulated platforms as close as possible to actual experimental framework
- Experiments in 2 steps:
  - use of the same platform configuration and application scenario than during MPI experiments,
  - launch extensive set of simulations with larger applications

# Heuristics

FIFO
SPT
SRPT
SWRPT

# Heuristics

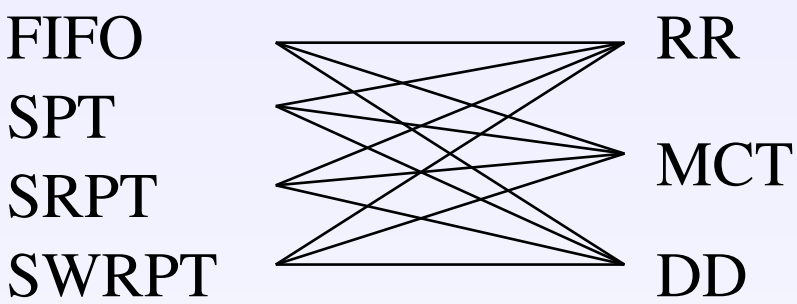
FIFO
SPT
SRPT
SWRPT

RR

MCT

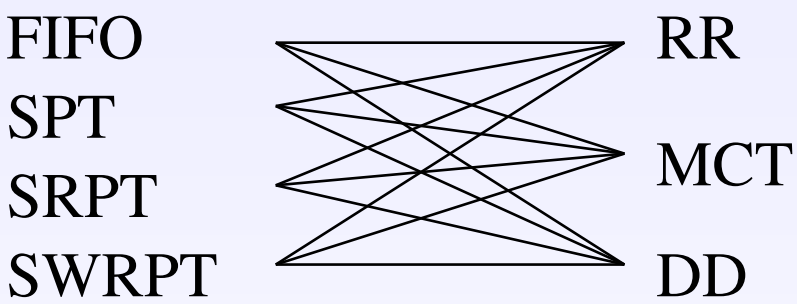
DD

# Heuristics



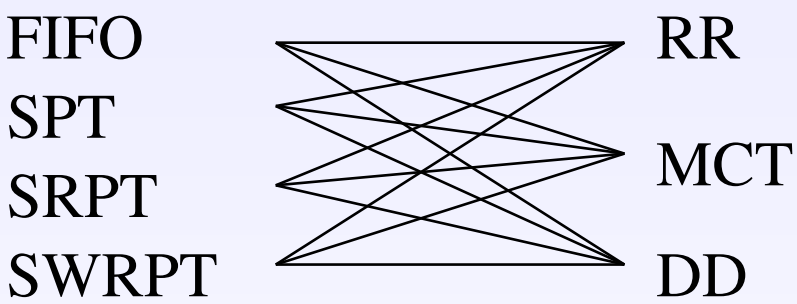

- Steady-state MWMA (Master Worker Multi-applications) on each time interval
- CBS3M (*Clever Burst Steady-State Stretch Minimizing*)

# Heuristics




- Steady-state MWMA (Master Worker Multi-applications) on each time interval
- CBS3M (*Clever Burst Steady-State Stretch Minimizing*)

# Heuristics




- Steady-state MWMA (Master Worker Multi-applications) on each time interval
- CBS3M (*Clever Burst Steady-State Stretch Minimizing*)

# MPI experiment results

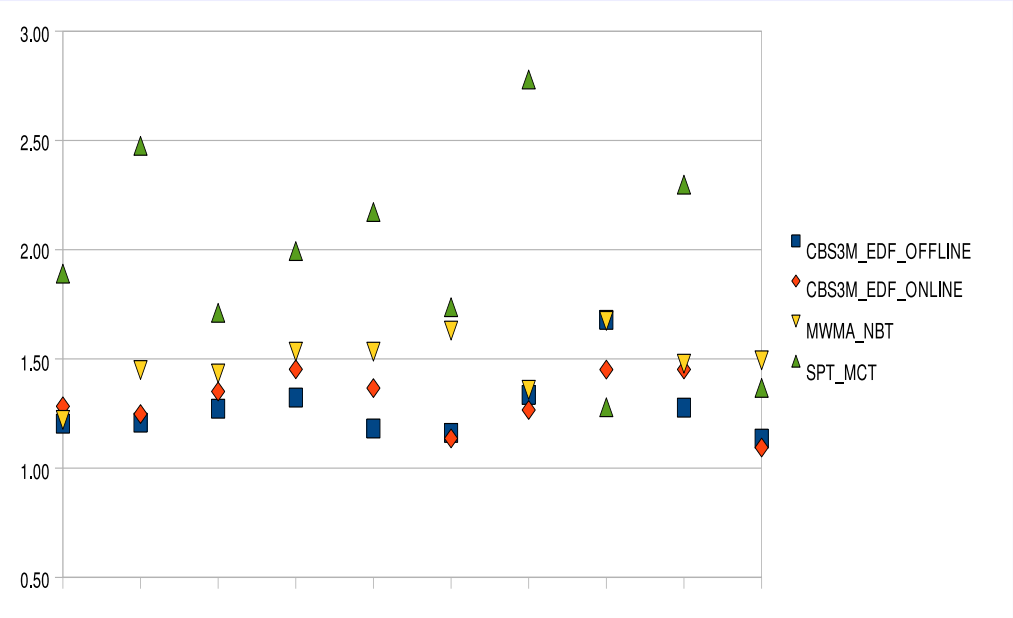



Figure: Relative max-stretch of best four heuristics.

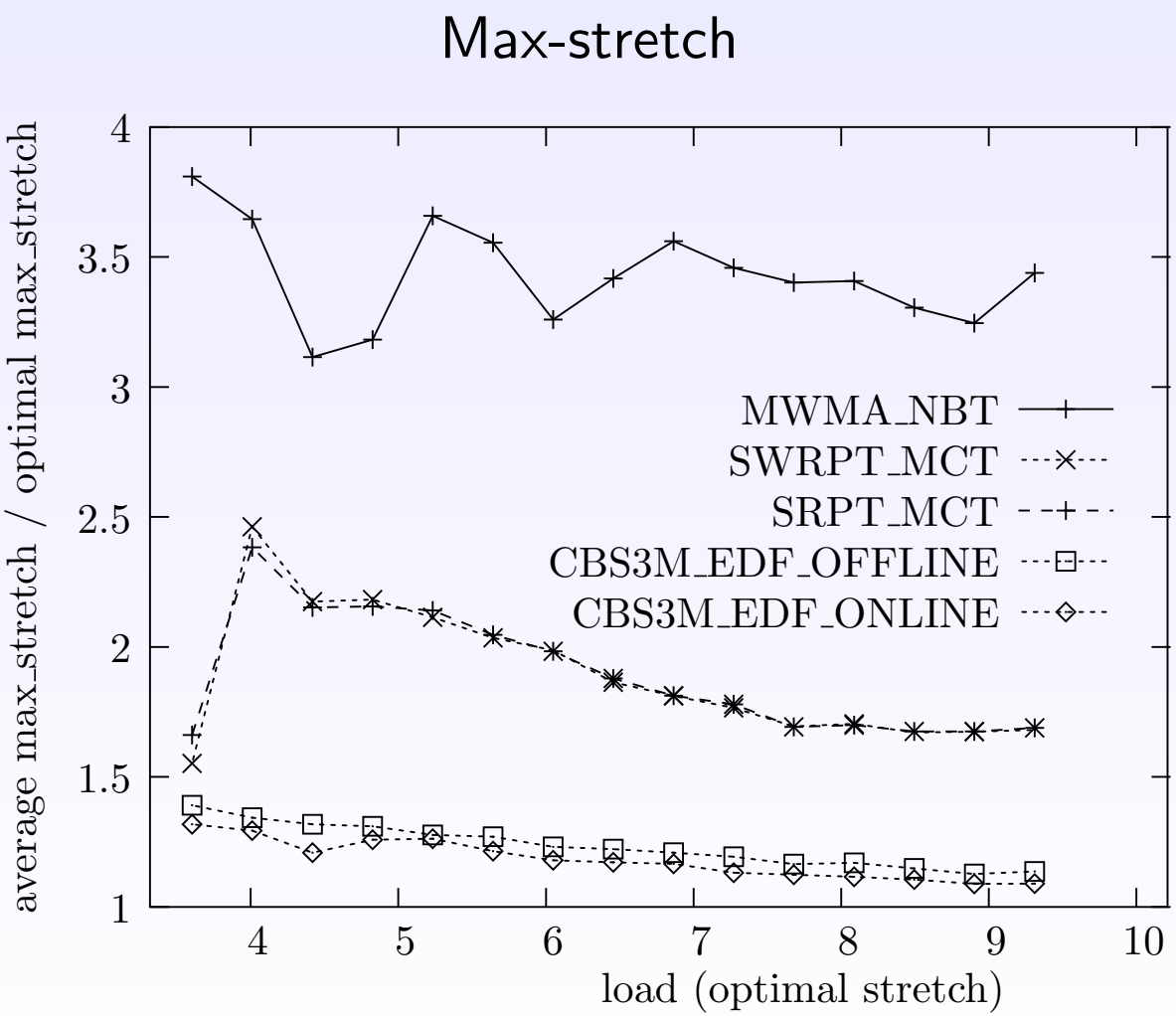
- Resource selection
- CBS3M online competitive agains CBS3M offline

# MPI experiments vs simulations

Comparison of relative max-stretch

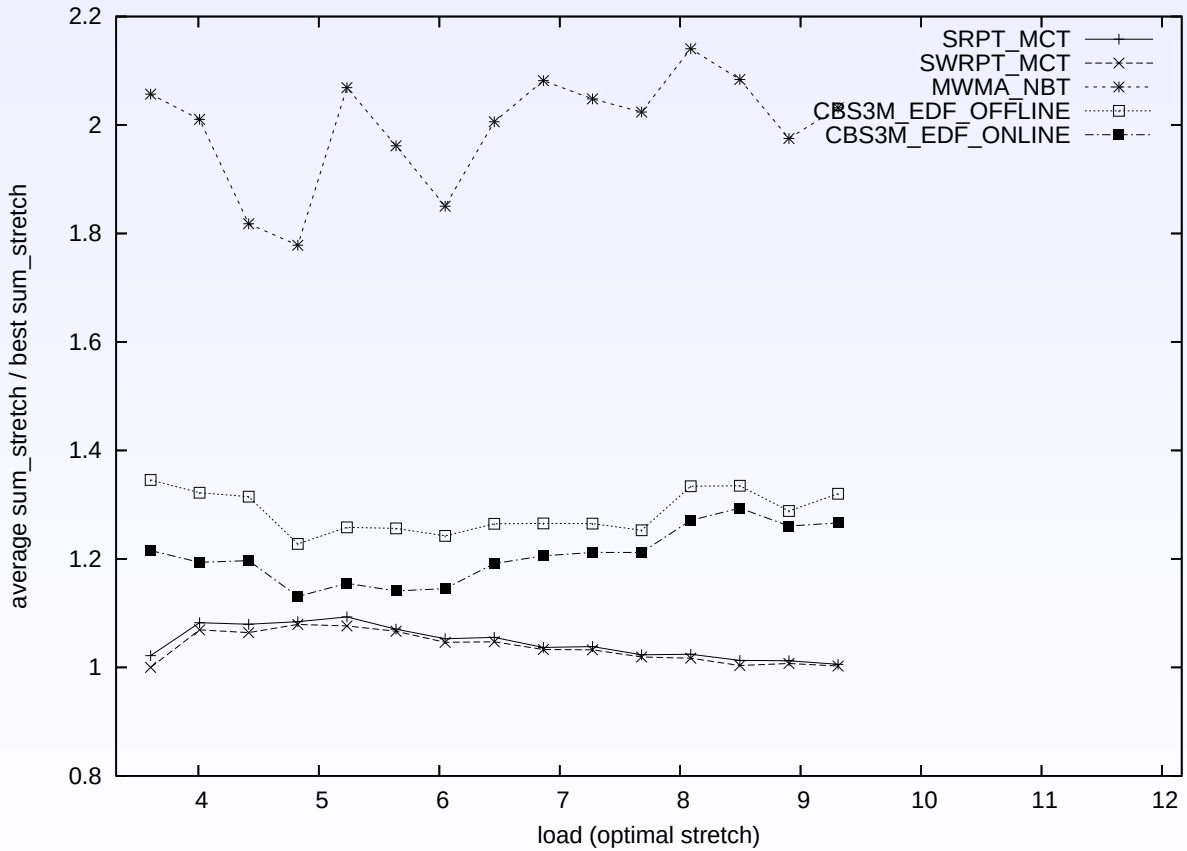
- average difference around 16%
- standard deviation of 14% (maximum of 72%).

# Simulation results



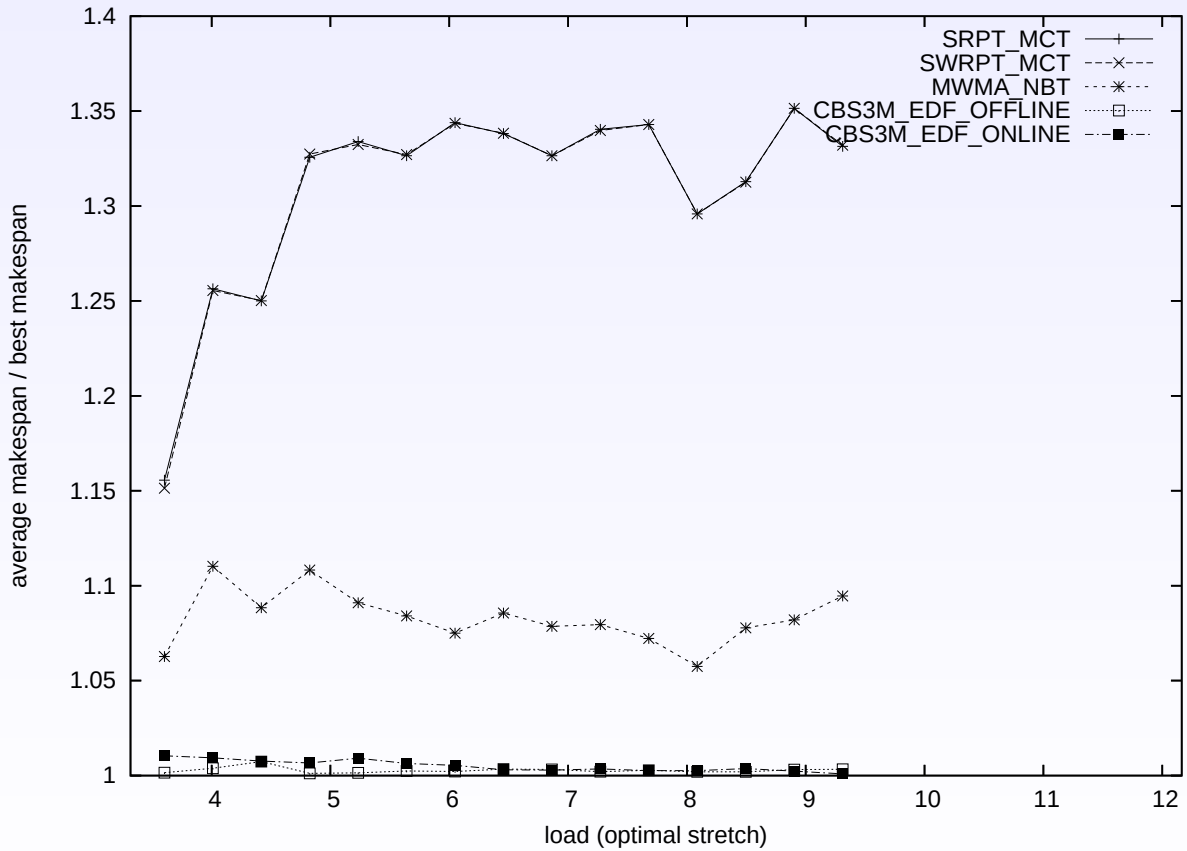
# Simulation results

## Sum-stretch



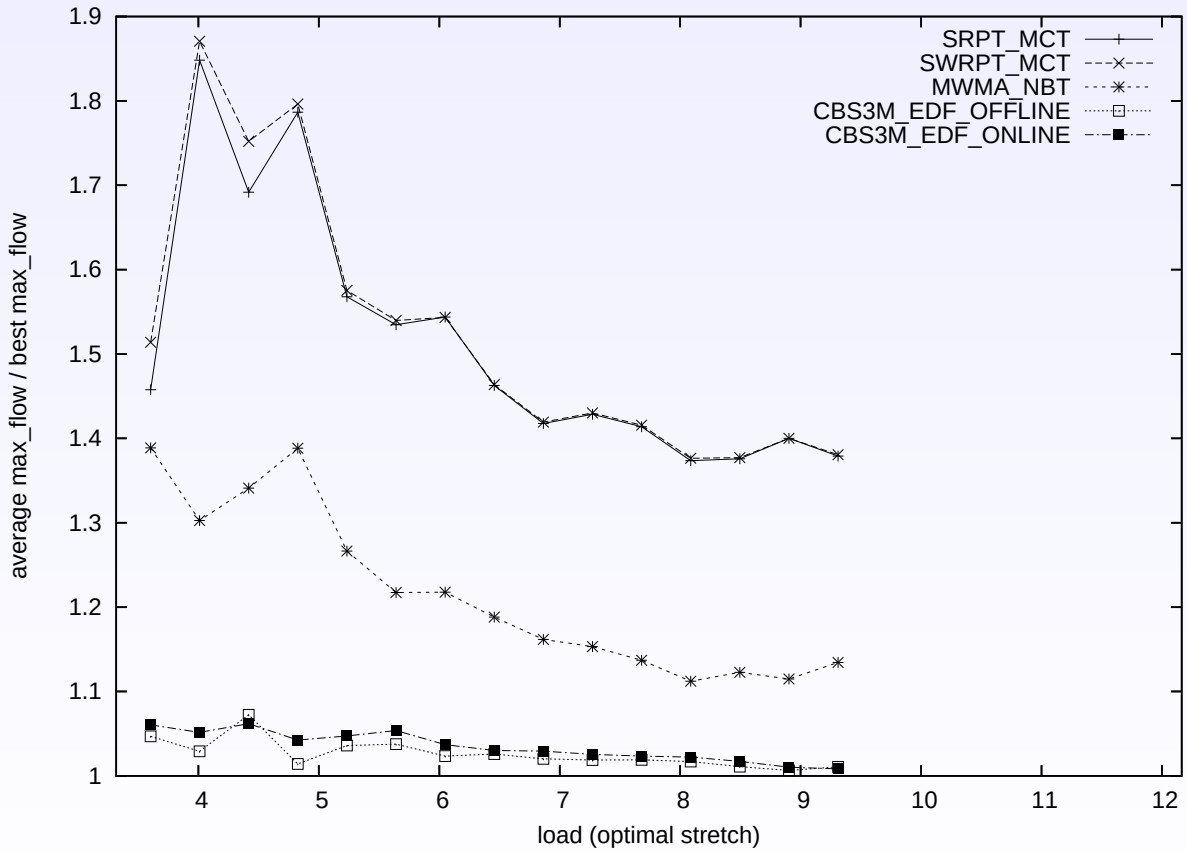
# Simulation results

## Makespan



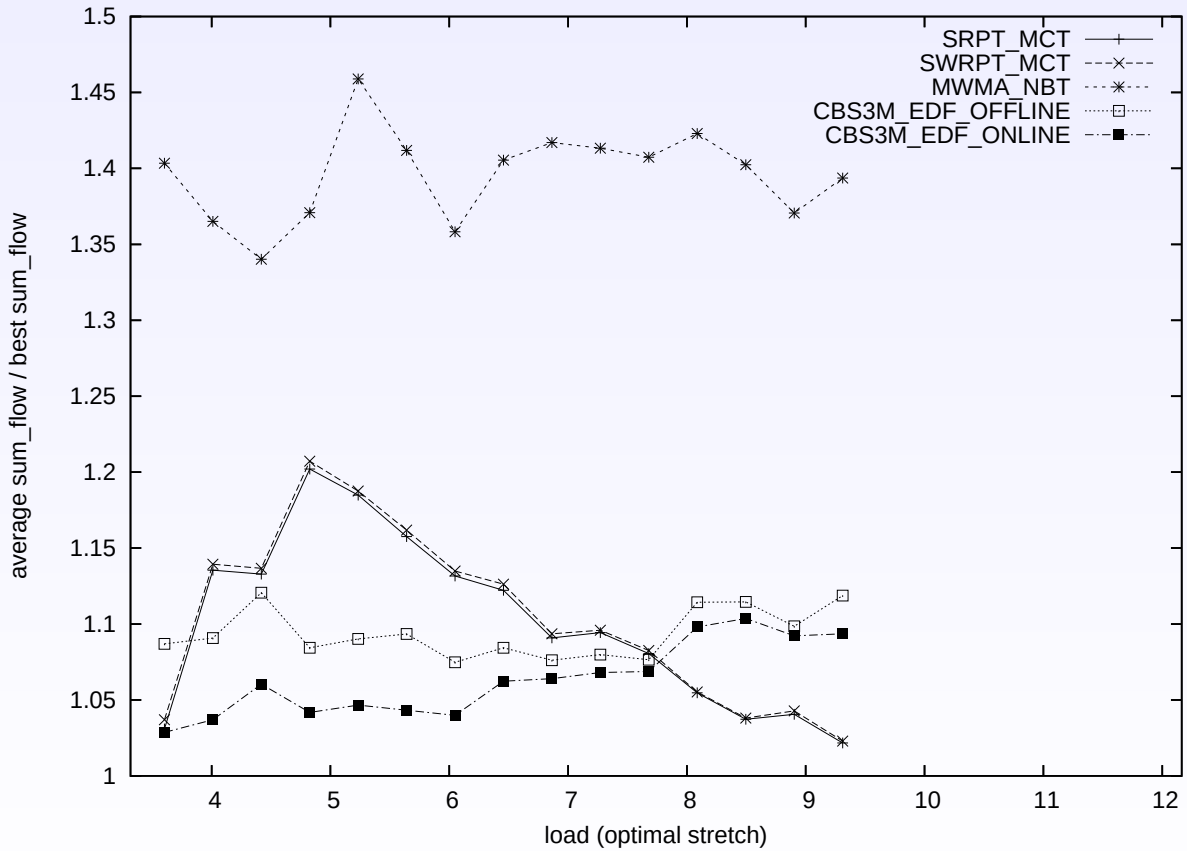
# Simulation results

## Max-flow



# Simulation results

## Sum-flow



# Outline

# Conclusion

- Key points:
  - Realistic platform model
  - Optimal offline algorithm
  - Efficient online algorithm based on offline study
- Extensions:
  - Extend the simulation to larger platform
  - Bicriteria

# Conclusion

- Key points:
  - Realistic platform model
  - Optimal offline algorithm
  - Efficient online algorithm based on offline study
- Extensions:
  - Extend the simulation to larger platform
  - Bicriteria

# Conclusion

- Key points:
  - Realistic platform model
  - Optimal offline algorithm
  - Efficient online algorithm based on offline study
- Extensions:
  - Extend the simulation to larger platform
  - Bicriteria

# For any questions...

- Platform parameters
- Detailled results

# Parameters

| | | |
|---|---|---|
| general | number of workers | 8 |
| | number of applications | 12 |
| arrival dates | mean of the distribution in the log space | 4.0 |
| | standard deviation in the log space | 1.2 |
| computations | maximum amount of work application | 76.8 Gflops |
| | minimum amount of work per task | 3.1 Gflops |
| communications | maximum amount of communication per application | 800 MB |
| | minimum amount of communication per task | 40 MB |
| number of tasks | minimum number of tasks per application | 10 |

Table: Parameters for the MPI experiments

# Parameters

| | | |
|---|---|---|
| general | number of workers | 10 |
| | number of applications | 20 |
| arrival dates | mean of the distribution in the log space | 4.0 |
| | standard deviation in the log space | 1.2 |
| computations | maximum amount of work application | 409 Gflops |
| | minimum amount of work per task | 3.1 Gflops |
| communications | maximum amount of communication per application | 6 GB |
| | minimum amount of communication per task | 40 MB |
| number of tasks | minimum number of tasks per application | 10 |

Table: Parameters for the SimGrid simulations

Back

# Detailled results

| Algorithm | Exp1 | Exp2 | Exp3 | Exp4 | Exp5 | Exp6 | Exp7 | Exp8 | Exp9 | Exp10 | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **CBS3M** EDF OFFLINE | **1.20** | **1.21** | **1.27** | **1.32** | **1.18** | 1.16 | 1.34 | 1.68 | 1.28 | 1.13 | 1.28 |
| **CBS3M** EDF ONLINE | 1.28 | 1.25 | 1.35 | 1.45 | 1.37 | **1.14** | **1.27** | 1.45 | 1.45 | **1.09** | 1.31 |
| **CBS3M** FIFO OFFLINE | 1.38 | 1.25 | 1.28 | 1.37 | 1.34 | 1.22 | 1.35 | 1.64 | **1.27** | 1.37 | 1.35 |
| **CBS3M** FIFO ONLINE | 1.42 | 1.26 | 1.48 | 1.43 | 1.47 | 1.15 | 1.54 | 1.55 | 1.36 | 1.16 | 1.38 |
| **FIFO MCT** | 1.71 | 2.46 | 1.87 | 2.54 | 1.53 | 1.28 | 2.77 | 1.66 | 2.27 | 1.37 | 1.95 |
| **FIFO RR** | 5.06 | 3.03 | 2.88 | 3.58 | 4.31 | 4.42 | 3.75 | 9.37 | 3.70 | 2.55 | 4.26 |
| **MWMA MS** | 1.66 | 1.99 | 2.42 | 1.80 | 2.17 | 2.18 | 1.80 | 2.98 | 2.28 | 3.18 | 2.24 |
| **MWMA NBT** | 1.22 | 1.45 | 1.43 | 1.53 | 1.53 | 1.63 | 1.36 | 1.67 | 1.48 | 1.49 | 1.48 |
| **SPT DD** | 4.27 | 3.06 | 2.36 | 2.74 | 5.00 | 9.20 | 4.18 | 11.17 | 3.33 | 2.32 | 4.76 |
| **SPT MCT** | 1.89 | 2.48 | 1.71 | 1.99 | 2.17 | 1.74 | 2.78 | **1.28** | 2.30 | 1.37 | 1.97 |
| **SRPT MCT** | 1.91 | 2.41 | 1.72 | 2.00 | 2.17 | 1.76 | 2.79 | 1.64 | 2.27 | 1.38 | 2.00 |
| **SWRPT MCT** | 1.92 | 2.44 | 1.72 | 1.99 | 2.17 | 1.76 | 2.97 | 1.63 | 2.28 | 1.38 | 2.03 |

Table: Results of the MPI experiments.

# Detailled results

| Algorithm | minimum | average | (± stddev) | maximum | (fraction of best result) |
|---|---|---|---|---|---|
| FIFO_RR | 4.550 | 16.689 | (± 7.897) | 62.6 | (the best in 0.0 %) |
| FIFO_MCT | 1.857 | 6.912 | (± 2.404) | 17.9 | (the best in 0.0 %) |
| FIFO_DD | 4.550 | 16.689 | (± 7.897) | 62.6 | (the best in 0.0 %) |
| SPT_RR | 1.348 | 4.274 | (± 1.771) | 13.8 | (the best in 0.0 %) |
| SPT_MCT | 1.007 | 1.928 | (± 0.610) | 5.99 | (the best in 1.3 %) |
| SPT_DD | 1.348 | 4.274 | (± 1.771) | 13.8 | (the best in 0.0 %) |
| SRPT_RR | 1.348 | 4.121 | (± 1.737) | 13.8 | (the best in 0.0 %) |
| SRPT_MCT | 1.007 | 1.861 | (± 0.601) | 6.87 | (the best in 2.2 %) |
| SRPT_DD | 1.348 | 4.121 | (± 1.737) | 13.8 | (the best in 0.0 %) |
| SWRPT_RR | 1.344 | 4.119 | (± 1.739) | 13.8 | (the best in 0.0 %) |
| SWRPT_MCT | 1.007 | 1.857 | (± 0.601) | 6.87 | (the best in 1.9 %) |
| SWRPT_DD | 1.344 | 4.119 | (± 1.739) | 13.8 | (the best in 0.0 %) |
| MWMA_NBT | 1.477 | 3.433 | (± 1.044) | 8.49 | (the best in 0.0 %) |
| MWMA_MS | 2.435 | 8.619 | (± 2.420) | 20.4 | (the best in 0.0 %) |
| CBS3M_FIFO_ONLINE | 1.003 | 1.322 | (± 0.208) | 2.83 | (the best in 6.9 %) |
| **CBS3M_EDF_ONLINE** | **1.003** | **1.163** | **(± 0.118)** | **1.93** | **(the best in 64.0 %)** |
| CBS3M_FIFO_OFFLINE | 1.022 | 1.379 | (± 0.276) | 3.74 | (the best in 3.8 %) |
| **CBS3M_EDF_OFFLINE** | **1.011** | **1.213** | **(± 0.125)** | **2.06** | **(the best in 26.2 %)** |

Table: Max-stretch of all heuristics in the simulations.

# Detailled results

| Algorithm | minimum | average | (± stddev) | maximum | (fraction of best result) |
|---|---|---|---|---|---|
| FIFO_RR | 2.064 | 6.783 | (± 3.210) | 30.7 | (the best in 0.0 %) |
| FIFO_MCT | 1.322 | 2.754 | (± 0.670) | 6.45 | (the best in 0.0 %) |
| FIFO_DD | 2.064 | 6.783 | (± 3.210) | 30.7 | (the best in 0.0 %) |
| SPT_RR | 1.019 | 2.942 | (± 1.221) | 10.1 | (the best in 0.0 %) |
| SPT_MCT | 1.000 | 1.182 | (± 0.183) | 2.53 | (the best in 2.4 %) |
| SPT_DD | 1.019 | 2.942 | (± 1.221) | 10.1 | (the best in 0.0 %) |
| SRPT_RR | 1.007 | 2.607 | (± 1.071) | 8.93 | (the best in 0.0 %) |
| **SRPT_MCT** | **1.000** | **1.045** | **(± 0.098)** | **1.92** | **(the best in 25.5 %)** |
| SRPT_DD | 1.007 | 2.607 | (± 1.071) | 8.93 | (the best in 0.0 %) |
| SWRPT_RR | 1.000 | 2.596 | (± 1.068) | 8.96 | (the best in 0.1 %) |
| **SWRPT_MCT** | **1.000** | **1.038** | **(± 0.098)** | **1.92** | **(the best in 60.1 %)** |
| SWRPT_DD | 1.000 | 2.596 | (± 1.068) | 8.96 | (the best in 0.1 %) |
| MWMA_NBT | 1.051 | 2.013 | (± 0.644) | 5.41 | (the best in 0.0 %) |
| MWMA_MS | 1.663 | 4.183 | (± 1.269) | 11.5 | (the best in 0.0 %) |
| CBS3M_FIFO_ONLINE | 1.000 | 1.294 | (± 0.208) | 2.16 | (the best in 0.4 %) |
| **CBS3M_EDF_ONLINE** | **1.000** | **1.201** | **(± 0.190)** | **2.08** | **(the best in 20.2 %)** |
| CBS3M_FIFO_OFFLINE | 1.000 | 1.332 | (± 0.227) | 2.57 | (the best in 0.1 %) |
| CBS3M_EDF_OFFLINE | 1.000 | 1.272 | (± 0.214) | 2.49 | (the best in 3.8 %) |

Table: Sum-stretch of all heuristics in the simulations.

# Detailled results

| Algorithm | minimum | average | (± stddev) | maximum | (fraction of best result) |
|---|---|---|---|---|---|
| FIFO_RR | 1.343 | 2.716 | (± 0.684) | 5.31 | (the best in 0.0 %) |
| FIFO_MCT | 1.000 | 1.329 | (± 0.202) | 2.11 | (the best in 0.1 %) |
| FIFO_DD | 1.343 | 2.716 | (± 0.684) | 5.31 | (the best in 0.0 %) |
| SPT_RR | 1.325 | 2.714 | (± 0.685) | 5.33 | (the best in 0.0 %) |
| SPT_MCT | 1.000 | 1.329 | (± 0.202) | 2.1 | (the best in 0.0 %) |
| SPT_DD | 1.325 | 2.714 | (± 0.685) | 5.33 | (the best in 0.0 %) |
| SRPT_RR | 1.325 | 2.714 | (± 0.686) | 5.32 | (the best in 0.0 %) |
| SRPT_MCT | 1.000 | 1.328 | (± 0.202) | 2.1 | (the best in 0.0 %) |
| SRPT_DD | 1.325 | 2.714 | (± 0.686) | 5.32 | (the best in 0.0 %) |
| SWRPT_RR | 1.322 | 2.715 | (± 0.686) | 5.32 | (the best in 0.0 %) |
| SWRPT_MCT | 1.000 | 1.328 | (± 0.202) | 2.1 | (the best in 0.0 %) |
| SWRPT_DD | 1.322 | 2.715 | (± 0.686) | 5.32 | (the best in 0.0 %) |
| MWMA_NBT | 1.000 | 1.079 | (± 0.070) | 1.45 | (the best in 4.6 %) |
| MWMA_MS | 1.000 | 1.078 | (± 0.067) | 1.42 | (the best in 2.1 %) |
| CBS3M_FIFO_ONLINE | 1.000 | 1.029 | (± 0.029) | 1.17 | (the best in 7.5 %) |
| **CBS3M_EDF_ONLINE** | **1.000** | **1.004** | **(± 0.006)** | **1.05** | **(the best in 35.0 %)** |
| **CBS3M_FIFO_OFFLINE** | **1.000** | **1.018** | **(± 0.023)** | **1.22** | **(the best in 17.6 %)** |
| **CBS3M_EDF_OFFLINE** | **1.000** | **1.003** | **(± 0.006)** | **1.07** | **(the best in 53.0 %)** |

Table: Makespan of all heuristics in the simulations.

# Detailled results

| Algorithm | minimum | average | (± stddev) | maximum | (fraction of best result) |
|---|---|---|---|---|---|
| FIFO_RR | 1.146 | 3.097 | (± 1.135) | 10.2 | (the best in 0.0 %) |
| **FIFO_MCT** | **1.000** | **1.281** | **(± 0.258)** | **2.83** | **(the best in 14.4 %)** |
| FIFO_DD | 1.146 | 3.097 | (± 1.135) | 10.2 | (the best in 0.0 %) |
| SPT_RR | 1.386 | 3.282 | (± 1.222) | 10.9 | (the best in 0.0 %) |
| SPT_MCT | 1.002 | 1.460 | (± 0.287) | 3.09 | (the best in 0.0 %) |
| SPT_DD | 1.386 | 3.282 | (± 1.222) | 10.9 | (the best in 0.0 %) |
| SRPT_RR | 1.386 | 3.289 | (± 1.225) | 10.9 | (the best in 0.0 %) |
| SRPT_MCT | 1.003 | 1.473 | (± 0.306) | 4.28 | (the best in 0.0 %) |
| SRPT_DD | 1.386 | 3.289 | (± 1.225) | 10.9 | (the best in 0.0 %) |
| SWRPT_RR | 1.382 | 3.291 | (± 1.225) | 10.9 | (the best in 0.0 %) |
| SWRPT_MCT | 1.000 | 1.477 | (± 0.309) | 4.28 | (the best in 0.1 %) |
| SWRPT_DD | 1.382 | 3.291 | (± 1.225) | 10.9 | (the best in 0.0 %) |
| MWMA_NBT | 1.000 | 1.181 | (± 0.153) | 1.99 | (the best in 7.0 %) |
| MWMA_MS | 1.000 | 1.261 | (± 0.189) | 2.32 | (the best in 1.1 %) |
| CBS3M_FIFO_ONLINE | 1.000 | 1.054 | (± 0.061) | 1.52 | (the best in 5.8 %) |
| **CBS3M_EDF_ONLINE** | **1.000** | **1.031** | **(± 0.057)** | **1.48** | **(the best in 23.2 %)** |
| **CBS3M_FIFO_OFFLINE** | **1.000** | **1.037** | **(± 0.058)** | **1.48** | **(the best in 21.6 %)** |
| **CBS3M_EDF_OFFLINE** | **1.000** | **1.023** | **(± 0.055)** | **1.48** | **(the best in 48.7 %)** |

Table: Max-flow of all heuristics in the simulations.

# Detailled results

| Algorithm | minimum | average | (± stddev) | maximum | (fraction of best result) |
|---|---|---|---|---|---|
| FIFO_RR | 1.644 | 4.020 | (± 1.567) | 16.3 | (the best in 0.0 %) |
| FIFO_MCT | 1.134 | 1.652 | (± 0.264) | 3.33 | (the best in 0.0 %) |
| FIFO_DD | 1.644 | 4.020 | (± 1.567) | 16.3 | (the best in 0.0 %) |
| SPT_RR | 1.196 | 2.811 | (± 1.081) | 9.21 | (the best in 0.0 %) |
| SPT_MCT | 1.000 | 1.149 | (± 0.171) | 2.32 | (the best in 3.5 %) |
| SPT_DD | 1.196 | 2.811 | (± 1.081) | 9.21 | (the best in 0.0 %) |
| SRPT_RR | 1.079 | 2.704 | (± 1.048) | 9.03 | (the best in 0.0 %) |
| **SRPT_MCT** | **1.000** | **1.105** | **(± 0.151)** | **2.23** | **(the best in 32.1 %)** |
| SRPT_DD | 1.079 | 2.704 | (± 1.048) | 9.03 | (the best in 0.0 %) |
| SWRPT_RR | 1.079 | 2.706 | (± 1.049) | 9.03 | (the best in 0.0 %) |
| **SWRPT_MCT** | **1.000** | **1.108** | **(± 0.152)** | **2.23** | **(the best in 15.4 %)** |
| SWRPT_DD | 1.079 | 2.706 | (± 1.049) | 9.03 | (the best in 0.0 %) |
| MWMA_NBT | 1.000 | 1.404 | (± 0.217) | 2.29 | (the best in 0.1 %) |
| MWMA_MS | 1.359 | 2.333 | (± 0.355) | 3.7 | (the best in 0.0 %) |
| CBS3M_FIFO_ONLINE | 1.000 | 1.122 | (± 0.101) | 1.62 | (the best in 1.4 %) |
| **CBS3M_EDF_ONLINE** | **1.000** | **1.065** | **(± 0.090)** | **1.53** | **(the best in 35.6 %)** |
| CBS3M_FIFO_OFFLINE | 1.000 | 1.120 | (± 0.103) | 1.67 | (the best in 0.3 %) |
| **CBS3M_EDF_OFFLINE** | **1.000** | **1.087** | **(± 0.101)** | **1.66** | **(the best in 18.7 %)** |

Table: Sum-flow of all heuristics in the simulations.

Back