An Optimal Semi-Partitioned Scheduler
Assuming Arbitrary Affinity Masks

Sergey Voronov, James H. Anderson

The University of North Carolina at Chapel Hill
Department of Computer Science

39th IEEE Real-Time Systems Symposium
Nashville, Tennessee, USA

December 14, 2018
Motivation

Peter Zijlstra’s keynote talk (ECRTS’17).
Introduction

Motivation
Peter Zijlstra’s keynote talk (ECRTS’17).

Problem (Informal)
How to schedule a task set on an identical multiprocessor if some tasks have affinity constraints, so they can be scheduled only on specific cores?

Outline:
- Necessary definitions
- Feasibility test
- Affinity reduction
- AM-Red scheduler
Soft Real-Time

Time

Task tardiness: \( \max_{\text{jobs}} (\text{job tardiness}) \)

Approach

Goal

Hard Real-Time

no tardiness at all

Soft Real-Time

bounded tardiness for every task

Sergey Voronov, James H. Anderson (UNC-CH)  An Optimal Semi-Partitioned Scheduler Assuming Arbitrary Affinity Masks  December 14, 2018
Soft Real-Time

- Release
- Deadline

Time

Soft Real-Time

- Definitions
- Feasibility Test
- Affinity Reduction
- AM-Red
- Conclusion

Sergey Voronov, James H. Anderson (UNC-CH) | An Optimal Semi-Partitioned Scheduler Assuming Arbitrary Affinity Masks | December 14, 2018
### Soft Real-Time

- **Deadline**
- **Release**
- **Execution intervals**

#### Definitions
- **Soft Real-Time**
- **Hard Real-Time**

#### Approach
- **Goal**:
  - **Hard Real-Time**: no tardiness at all
  - **Soft Real-Time**: bounded tardiness for every task

---

_Sergey Voronov, James H. Anderson (UNC-CH) An Optimal Semi-Partitioned Scheduler Assuming Arbitrary Affinity Masks December 14, 2018_
Soft Real-Time

- Definitions
- Feasibility Test
- Affinity Reduction
- AM-Red
- Conclusion

**Soft Real-Time**

- Introduction
- Definitions
- Feasibility Test
- Affinity Reduction
- AM-Red
- Conclusion

**Soft Real-Time**

- Release
- Execution intervals
- Deadline
- Completion

**Time**

- **Deadline**
- **Completion**

**Response time**

**Tardiness**

**Task tardiness:** max jobs (job tardiness)

**Approach**

**Hard Real-Time**

- no tardiness at all

**Soft Real-Time**

- bounded tardiness for every task
Soft Real-Time

- **Release**
- **Deadline**
- **Completion**
- **Response time**
- **Tardiness**

**Definitions**
- **Soft Real-Time**
- **Hard Real-Time**

**Approach**
- **Goal**
  - **Hard Real-Time**: no tardiness at all
  - **Soft Real-Time**: bounded tardiness for every task

Sergey Voronov, James H. Anderson (UNC-CH)

**An Optimal Semi-Partitioned Scheduler Assuming Arbitrary Affinity Masks**

December 14, 2018
Soft Real-Time

Response time

Task tardiness:
\[
\max_{\text{jobs}} (\text{job tardiness})
\]
Soft Real-Time

Approach | Goal

Task tardiness:
\[ \max_{\text{jobs}} (\text{job tardiness}) \]

Hard Real-Time
- no tardiness at all

Soft Real-Time
- bounded tardiness for every task

Definitions
- **Deadline**
- **Execution intervals**
- **Response time**
- **Deadline**
- **Completion**
- **Tardiness**

**Introduction**

**Definitions**

**Feasibility Test**

**Affinity Reduction**

**AM-Red**

**Conclusion**

An Optimal Semi-Partitioned Scheduler Assuming Arbiitary Affinity Masks

Sergey Voronov, James H. Anderson (UNC-CH)

December 14, 2018
Soft Real-Time

<table>
<thead>
<tr>
<th>Approach</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hard Real-Time</td>
<td>no tardiness at all</td>
</tr>
</tbody>
</table>

Task tardiness: \( \max_{\text{jobs}}(\text{job tardiness}) \)

<table>
<thead>
<tr>
<th>Release</th>
<th>Deadline</th>
<th>Completion</th>
<th>Execution intervals</th>
<th>Tardiness</th>
</tr>
</thead>
</table>

Time

Response time
**Soft Real-Time**

<table>
<thead>
<tr>
<th>Approach</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hard Real-Time</td>
<td>no tardiness at all</td>
</tr>
<tr>
<td>Soft Real-Time</td>
<td>bounded tardiness for every task</td>
</tr>
</tbody>
</table>

**Task tardiness:**

$$\max_{\text{jobs}}(\text{job tardiness})$$
Affinity Masks

**Affinity Mask**

A set of cores where a task can be scheduled.
Affinity Masks

**Affinity Mask**

A set of cores where a task can be scheduled.

**Usage:**

- performance improvement
  (ensure shared cache)
- load balancing
- task isolation
Affinity Masks

Affinity Mask

A set of cores where a task can be scheduled.

Usage:
- performance improvement (ensure shared cache)
- load balancing
- task isolation

Affinity Graph (AG)

Bipartite graph that shows affinity masks.

\[
\begin{align*}
\pi_1 & \rightarrow \pi_2 & \pi_3 & \rightarrow n \text{ tasks in total} \\
\tau_1 & \rightarrow \tau_2 & \tau_3 & \rightarrow m \text{ cores in total}
\end{align*}
\]
### Feasibility Test

#### Task Set Feasibility Problem (Formal)

Can any algorithm schedule a given task set $\tau$ of size $n$ with a given affinity masks set on an identical multiprocessor with $m$ cores?
Feasibility Test

Task Set Feasibility Problem (Formal)

Can any algorithm schedule a given task set $\tau$ of size $n$ with a given affinity masks set on an identical multiprocessor with $m$ cores?

Feasibility without affinities

\[ \sum \text{utilizations of tasks} \leq m \]
Feasibility Test

Task Set Feasibility Problem (Formal)

Can any algorithm schedule a given task set $\tau$ of size $n$ with a given affinity masks set on an identical multiprocessor with $m$ cores?

Feasibility without affinities

$$\sum \text{utilizations of tasks} \leq m$$

Feasibility with affinities

For every subset of tasks $\tau' \subset \tau$:

$$\sum_{\tau'} \text{utilizations of tasks} \leq \text{size of } \tau' \text{ affinities union}$$
Feasibility Test

Task Set Feasibility Problem (Formal)
Can any algorithm schedule a given task set $\tau$ of size $n$ with a given affinity masks set on an identical multiprocessor with $m$ cores?

Feasibility without affinities
$$\sum \text{utilizations of tasks} \leq m$$

Feasibility with affinities
For every subset of tasks $\tau' \subset \tau$:
$$\sum_{\tau'} \text{utilizations of tasks} \leq \text{size of } \tau' \text{ affinities union}$$

Existing test#
LP-based, $\tilde{O}(mn(m + n)^{2.9})$

Proposed test
Flow-based, $\tilde{O}(mn\sqrt{m + n})$

Feasibility Test: Overview

Proposed test: Max Flow is \( \sum_{\tau_i \in \tau} U_i \Rightarrow \text{task set is feasible.} \)
Feasibility Test: Overview

Proposed test: Max Flow is \( \sum_{\tau_j \in \tau} U_i \Rightarrow \) task set is feasible.

Affinity graph (AG).

AG-based flow network.
Proposed test: Max Flow is $\sum_{\tau_i \in \tau} U_i \Rightarrow$ task set is feasible.

Affinity graph (AG).

AG-based flow network.
Proposed test: Max Flow is \[
\sum_{\tau_i \in \tau} U_i \Rightarrow \text{task set is feasible.}
\]
Feasibility Test: Overview

Proposed test: Max Flow is \( \sum_{\tau_j \in \tau} U_i \Rightarrow \) task set is feasible.

Affinity graph (AG).

AG-based flow network.

Sergey Voronov, James H. Anderson (UNC-CH)  
An Optimal Semi-Partitioned Scheduler Assuming Arbitrary Affinity Masks  
December 14, 2018
Feasibility Test: Overview

Proposed test: Max Flow is $\sum_{\tau_i \in \tau} U_i \Rightarrow$ task set is feasible.

Affinity graph (AG).

AG-based flow network.

All edges’ capacities are equal to 1.
Affinity Reduction

General idea

Edge without flow can be removed from AG.
Affinity Reduction

General idea
Edge without flow can be removed from AG.

Change rules:
- Total flow from a task should not change
- Total flow to a core should not change

Sergey Voronov, James H. Anderson (UNC-CH) An Optimal Semi-Partitioned Scheduler Assuming Arbitrary Affinity Masks December 14, 2018
Affinity Reduction

General idea

Edge without flow can be removed from AG.

Change rules:

- Total flow from a task should not change
- Total flow to a core should not change
Affinity Reduction

General idea
Edge without flow can be removed from AG.

Change rules:
- Total flow from a task should not change
- Total flow to a core should not change
Affinity Reduction

General idea

Edge without flow can be removed from AG.

Change rules:
- Total flow from a task should not change
- Total flow to a core should not change
Affinity Reduction

General idea

Edge without flow can be removed from AG.

Change rules:
- Total flow from a task should not change
- Total flow to a core should not change
Affinity Reduction

General idea

Edge without flow can be removed from AG.

Change rules:

- Total flow from a task should not change
- Total flow to a core should not change
Affinity Reduction

General idea

Edge without flow can be removed from AG.

Change rules:
- Total flow from a task should not change
- Total flow to a core should not change
Affinity Reduction

General idea
Edge without flow can be removed from AG.

Change rules:
- Total flow from a task should not change
- Total flow to a core should not change
Affinity Reduction

General idea

Edge without flow can be removed from AG.

Change rules:
- Total flow from a task should not change
- Total flow to a core should not change
Affinity Reduction

General idea

Edge without flow can be removed from AG.

Change rules:
- Total flow from a task should not change
- Total flow to a core should not change
Affinity Reduction

General idea

Edge without flow can be removed from AG.

Change rules:
- Total flow from a task should not change
- Total flow to a core should not change
Affinity Reduction

General idea

Edge without flow can be removed from AG.

Change rules:
- Total flow from a task should not change
- Total flow to a core should not change
Affinity Reduction

General idea
Edge without flow can be removed from AG.

Change rules:
- Total flow from a task should not change
- Total flow to a core should not change
AM-Red (Affinity Mask Reduction): Overview

- Compute Max Flow (checks feasibility)
AM-Red (Affinity Mask Reduction): Overview

- Compute Max Flow (checks feasibility)
- Apply affinity reduction (ensures at most \( m - 1 \) migrating tasks)
AM-Red (**Affinity Mask Reduction**): Overview

- Compute Max Flow (checks feasibility)
- Apply affinity reduction (ensures at most $m - 1$ migrating tasks)

Tasks which are scheduled on at least two different cores
AM-Red (Affinity Mask Reduction): Overview

- Compute Max Flow (checks feasibility)
- Apply affinity reduction (ensures at most $m - 1$ migrating tasks)
- Construct a schedule template of size $|F|$ (ensures a tardiness bound of $|F|$)

Tasks which are scheduled on at least two different cores
AM-Red (Affinity Mask Reduction): Overview

- Compute Max Flow (checks feasibility)
- Apply affinity reduction (ensures at most \( m - 1 \) migrating tasks)
- Construct a schedule template of size \( |F| \) (ensures a tardiness bound of \( |F| \))
- Repeat template to get actual schedule

tasks which are scheduled on at least two different cores
AM-Red: Template Construction

A diagram illustrating the template construction process.

The diagram shows a network of tasks labeled $\tau_1$, $\tau_2$, $\tau_3$, and $\tau_4$, with edges connecting them, and nodes labeled $\pi_1$, $\pi_2$, $\pi_3$, and $\pi_4$. The edges are labeled with values 0.5, indicating some form of resource allocation or task connection. The diagram also includes a time axis with values 0, 1, 2, 3, 4, and 5, and a set of tasks $F$ with labels $\pi_1$, $\pi_2$, and $\pi_3$. The tasks are to be scheduled over time.
AM-Red: Template Construction

The diagram illustrates the template construction process for cores’ fill order and time allocation.

- τ₁, τ₂, τ₃, τ₄ represent the time intervals for different tasks.
- π₁, π₂, π₃ represent the cores.
- F denotes the task execution order.
- The time scale ranges from 0 to 5 for the x-axis, representing time.

The diagram shows how tasks are assigned to cores based on affinity masks, ensuring an optimal semi-partitioned scheduler assumption for arbitrary affinity masks.
AM-Red: Template Construction

![Diagram showing cores' fill order and time]
AM-Red: Template Construction

Core's fill order

Sergey Voronov, James H. Anderson (UNC-CH)  An Optimal Semi-Partitioned Scheduler Assuming Arbitrary Affinity Masks  December 14, 2018
AM-Red: Template Construction

- \( \pi_1 \rightarrow \pi_2 \rightarrow \pi_3 \rightarrow \pi_4 \)
- \( \tau_1 \rightarrow \tau_2 \rightarrow \tau_3 \rightarrow \tau_4 \)

Cores' fill order:

- \( F \)

- \( \tau_1 \rightarrow \tau_2 \rightarrow \tau_3 \rightarrow \tau_4 \)

Graphical representation:

- \( \pi_1 \rightarrow \pi_2 \rightarrow \pi_3 \rightarrow \pi_4 \)

- \( \tau_1 \rightarrow \tau_2 \rightarrow \tau_3 \rightarrow \tau_4 \)

- \( 0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \)

Sergey Voronov, James H. Anderson (UNC-CH)
AM-Red: Template Construction

cores’ fill order

0.5 0.5 0.5 0.5

τ₁ τ₂ τ₃ τ₄

π₁ τ₁ π₂ τ₂ π₃

0 1 2 3 4 5

time

F

π₁ τ₁ τ₂

π₂ τ₃ τ₁

π₃ τ₃

Sergey Voronov, James H. Anderson (UNC-CH)
AM-Red: Template Construction

![Diagram of cores' fill order and time]
AM-Red: Template Construction

- cores’ fill order

<table>
<thead>
<tr>
<th>time</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

- \( \tau_1 \), \( \tau_2 \), \( \tau_3 \), \( \tau_4 \)
- \( \pi_1 \), \( \pi_2 \), \( \pi_3 \)

Sergey Voronov, James H. Anderson (UNC-CH)

An Optimal Semi-Partitioned Scheduler Assuming Arbitrary Affinity Masks

December 14, 2018
AM-Red: Template Construction (mixed cores order)
AM-Red: Template Construction (mixed cores order)

\[ \pi_1 \rightarrow \pi_2 \rightarrow \pi_3 \rightarrow \tau_1 \rightarrow \tau_2 \rightarrow \tau_3 \rightarrow \tau_4 \]

cores' fill order

\[ \tau_1 \rightarrow \tau_2 \rightarrow \tau_3 \rightarrow \tau_4 \]

\[ F \]

\[ \pi_1 \]

\[ \pi_2 \]

\[ \pi_3 \]

\[ 0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \]

\[ \text{time} \]
AM-Red: Template Construction (mixed cores order)

- \( \tau_1 \)
- \( \tau_2 \)
- \( \tau_3 \)
- \( \tau_4 \)

cores’ fill order

- \( \pi_1 \)
- \( \pi_2 \)
- \( \pi_3 \)

0.5

\( \pi_1 \)

<table>
<thead>
<tr>
<th>( \tau_1 )</th>
<th>( \tau_2 )</th>
</tr>
</thead>
</table>

\( \pi_3 \)

\( \pi_2 \)

time

0 1 2 3 4 5

Sergey Voronov, James H. Anderson (UNC-CH)

An Optimal Semi-Partitioned Scheduler Assuming Arbitrary Affinity Masks

December 14, 2018
AM-Red: Template Construction (mixed cores order)
AM-Red: Template Construction (mixed cores order)

\[ \tau_1, \tau_2, \tau_3, \tau_4 \]

cores' fill order

\[ \pi_1, \pi_2, \pi_3 \]

\[ F \]

| \pi_1 | \tau_1 | \tau_2 |
| \pi_3 | \tau_3 | \tau_4 |

- \[ \pi_1 \]
- \[ \pi_3 \]
- \[ \pi_2 \]

\[ \tau_1, \tau_2, \tau_3, \tau_4 \]

\[ \pi_1 \]

\[ \pi_3 \]

\[ \pi_2 \]

\[ F \]

\[ \text{time} \]

0 1 2 3 4 5
AM-Red: Template Construction (mixed cores order)
AM-Red: Template Construction (mixed cores order)
AM-Red: Template Construction (mixed tasks order)
AM-Red: Template Construction (mixed tasks order)
AM-Red: Template Construction (mixed tasks order)
AM-Red: Template Construction (mixed tasks order)

$$\tau_1 \quad \tau_2 \quad \tau_3 \quad \tau_4$$

cores’ fill order

$$\pi_1 \quad \pi_2 \quad \pi_3$$

$$\pi_1 \quad \tau_2 \quad \tau_1$$

$$\pi_2 \quad \tau_1$$

$$\pi_3$$

$$F$$

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sergey Voronov, James H. Anderson (UNC-CH)  An Optimal Semi-Partitioned Scheduler Assuming Arbitrary Affinity Masks  December 14, 2018
AM-Red: Template Construction (mixed tasks order)

\[
\begin{align*}
\tau_2 & \quad 0.5 \\
\pi_1 & \quad \tau_1 & \quad 0.5 \\
\pi_2 & \quad 0.5 \\
\pi_3 & \quad \tau_4 & \quad 0.5 \\
\pi_3 & \quad \tau_3 & \quad
dots
\end{align*}
\]

cores’ fill order
AM-Red: Template Construction (mixed tasks order)

- $\tau_1$
- $\tau_2$
- $\tau_3$
- $\tau_4$

$\pi_1$, $\pi_2$, $\pi_3$

- Cores’ fill order

Diagram:

- τ4
- τ3
- τ2
- τ1

Time:

- 0 1 2 3 4 5

Sergey Voronov, James H. Anderson (UNC-CH)
AM-Red: Template Construction (mixed tasks order)
AM-Red: Template Construction (mixed tasks order)

Diagram showing the tasks and their fill order on different cores.

- \( \pi_1 \) and \( \pi_2 \) are scheduled first on cores 1 and 2 respectively.
- \( \pi_3 \) is scheduled next on core 3.
- \( \pi_4 \) is scheduled last on core 4.

The tasks are scheduled in the order of their affinities, with \( \tau_1 \) and \( \tau_2 \) scheduled first, followed by \( \tau_3 \) and \( \tau_4 \).

The diagram also shows the time axis with tasks being scheduled from time 0 to time 5.

Sergey Voronov, James H. Anderson (UNC-CH)
An Optimal Semi-Partitioned Scheduler Assuming Arbitrary Affinity Masks
December 14, 2018
**AM-Red: Final Schedule**

<table>
<thead>
<tr>
<th>Time</th>
<th>Job release</th>
<th>Job deadline</th>
<th>Job completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Time**
- **Allocation intervals for $\tau_3$**
- **Scheduled intervals of $\tau_3$**
- **Allocation intervals for all other tasks**
AM-Red: Final Schedule

- Allocation intervals for $\tau_3$
- Scheduled intervals of $\tau_3$
- Allocation intervals for all other tasks

Job release
Job deadline
Job completion
AM-Red: Final Schedule

<table>
<thead>
<tr>
<th>π₁</th>
<th>F</th>
<th>2F</th>
<th>3F</th>
</tr>
</thead>
<tbody>
<tr>
<td>π₂</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>π₃</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Job release**
- **Job deadline**
- **Job completion**

<table>
<thead>
<tr>
<th>Allocation intervals for τ₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scheduled intervals of τ₃</td>
</tr>
<tr>
<td>Allocation intervals for all other tasks</td>
</tr>
</tbody>
</table>
AM-Red: Final Schedule

- Job release
- Job deadline
- Job completion

- Allocation intervals for $\tau_3$
- Scheduled intervals of $\tau_3$
- Allocation intervals for all other tasks

Figure 1: AM-Red: Final Schedule

π₁

π₂

π₃

$\pi_1$ $\pi_2$ $\pi_3$

$\tau_3$

<table>
<thead>
<tr>
<th>T</th>
<th>F</th>
<th>2F</th>
<th>3F</th>
</tr>
</thead>
<tbody>
<tr>
<td>π₁</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>π₂</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>π₃</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sergey Voronov, James H. Anderson (UNC-CH)  
An Optimal Semi-Partitioned Scheduler Assuming Arbitrary Affinity Masks  
December 14, 2018
AM-Red: Final Schedule

Job release
Job deadline
Job completion

Allocation intervals for $\tau_3$
Scheduled intervals of $\tau_3$
Allocation intervals for all other tasks
AM-Red: Final Schedule

![Diagram of task scheduling]

- **Job release**
- **Job deadline**
- **Job completion**
- **Allocation intervals for \( \tau_3 \)**
- **Scheduled intervals of \( \tau_3 \)**
- **Allocation intervals for all other tasks**
AM-Red: Final Schedule

<table>
<thead>
<tr>
<th>Job release</th>
<th>Job deadline</th>
<th>Job completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allocation intervals for $\tau_3$</td>
<td>Scheduled intervals of $\tau_3$</td>
<td>Allocation intervals for all other tasks</td>
</tr>
</tbody>
</table>

$\pi_1$ | $\pi_2$ | $\pi_3$ | $\tau_3$

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
</tr>
</tbody>
</table>

Sergey Voronov, James H. Anderson (UNC-CH) | An Optimal Semi-Partitioned Scheduler Assuming Arbitrary Affinity Masks | December 14, 2018
AM-Red: Final Schedule

- **Job release**
- **Job deadline**
- **Job completion**

- Allocation intervals for $\tau_3$
- Scheduled intervals of $\tau_3$
- Allocation intervals for all other tasks
AM-Red: Final Schedule

Job release
Job deadline
Job completion
Allocation intervals for $\tau_3$
Scheduled intervals of $\tau_3$
Allocation intervals for all other tasks
AM-Red: Final Schedule

- Job release
- Job deadline
- Job completion

- Allocation intervals for $\tau_3$
- Scheduled intervals of $\tau_3$
- Allocation intervals for all other tasks
AM-Red: Final Schedule

- Job release
- Job deadline
- Job completion

- Allocation intervals for $\tau_3$
- Scheduled intervals of $\tau_3$
- Allocation intervals for all other tasks
AM-Red: Final Schedule

Job release
Job deadline
Job completion

Allocation intervals for $\tau_3$
Scheduled intervals of $\tau_3$
Allocation intervals for all other tasks
SRT-optimal ($|F|$ is a tardiness bound)
Conclusion

- SRT-optimal ($|F|$ is a tardiness bound)
- At most $2m - 2$ migrations per template
Conclusion

- SRT-optimal ($|F|$ is a tardiness bound)
- At most $2m - 2$ migrations per template
- Total complexity is $O(m^2 n^2)$
Conclusion

- SRT-optimal ($|F|$ is a tardiness bound)
- At most $2m - 2$ migrations per template
- Total complexity is $O(m^2n^2)$
  
  ($O(m + n)$ for hierarchical masks)
SRT-optimal ($|F|$ is a tardiness bound)
- At most $2m - 2$ migrations per template
- Total complexity is $O(m^2 n^2)$
  ($O(m + n)$ for hierarchical masks)

Results:
- Proposed exact feasibility test.
- Developed an affinity reduction function.
- AM-Red is an SRT-optimal semi-partitioned scheduler.