Response Time Analysis for G-EDF and G-DM Scheduling of Sporadic DAG-Tasks with Arbitrary Deadline

Andrea Parri, Alessandro Biondi and Mauro Marinoni

Scuola Superiore Sant’Anna – Pisa, Italy
Multicore revolution

New parallel programming models for expressing parallel computational activities
Introduction

Big Data

- Novel programming models based on the Map-Reduce paradigm that relies on parallel processing
Introduction

**JUNIPER EU Project** – supported this work

- **Goal:** enable application development with performance guarantees required for real-time exploitation of large streaming data sources and stored data;

- **Case-study:** applications for credit cards.
DAG-Task

- Task model for expressing parallel computations with precedence constraints
- A task is described with a Directed Acyclic Graph (DAG)
Vertex – sequential computation with WCET $e_i$
DAG-Task

Edge – *precedence constraint* among two computational activities
Note: this model allows to express parallelism
**DAG-Task**

**Release of a DAG-Task** $\tau_i$

- All the vertices are *released simultaneously* but it can be that not all of them are *enabled* due to precedence constrains.

---

[Diagram showing a directed acyclic graph (DAG) with vertices $e_1, e_2, e_3, e_4, e_5, e_6, e_7$ and edge labels $\tau_i$. The graph is structured with edges from $e_1$ to $e_3$, $e_2$ to $e_4$, $e_3$ to $e_5$, and $e_5$ to $e_6$ and $e_7$.]
Sporadic DAG-Task

- DAG-Task $\tau_i$
  - Released with a minimum inter-arrival time $T_i$
  - Each vertex must complete within a deadline $D_i$
Example

processor 1

\[ e_1 \quad e_3 \]

processor 2

\[ e_2 \]

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( e_1 )</td>
<td>2</td>
</tr>
<tr>
<td>( e_2 )</td>
<td>2</td>
</tr>
<tr>
<td>( e_3 )</td>
<td>1</td>
</tr>
<tr>
<td>( e_4 )</td>
<td>1</td>
</tr>
<tr>
<td>( e_5 )</td>
<td>2</td>
</tr>
<tr>
<td>( e_6 )</td>
<td>3</td>
</tr>
</tbody>
</table>
Example

processor 1

\[ e_1, e_3, e_4 \]

processor 2

\[ e_2 \]

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>2</th>
<th>1</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>( e_1 )</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( e_2 )</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( e_3 )</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( e_4 )</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( e_5 )</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( e_6 )</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Scheduling Problem

Given

- a set of N sporadic DAG-Tasks;
- A scheduling algorithm (G-EDF or G-DM);
- A platform with m identical processors;

verify if all deadlines are guaranteed.
State of The Art

Existing schedulability analysis can be split in 3 categories:

- Based on **resource augmentation** (speed-up);
  
  (Baruah et al., Bonifaci et al., Nilissen et al., ...)

- Based on **capacity augmentation**;
  
  (Kim et al., Li et al., Lakshmanan et al., ...)

- Based on **Response-Time Analysis**.
  
  (Maia et al., Chwa et al., Melani et al., ...)

This Work

- Response-Time Analysis of Sporadic DAG-Tasks under both G-EDF and G-DM

Contribution w.r.t. the state of the art:

- Vertices-oriented analysis;
- Tasks can have arbitrary deadlines;
- Vertices can have arbitrary utilization;
- Augmentation bounds proved for N=1.
Response-Time Analysis

- For each DAG-Task $\tau_i$,

- For each vertex $v$ of $\tau_i$,

- Each job of vertex $v$ must complete within a deadline $D_i$

$$e_v + I_v = R_v \leq D_i$$

Vertex WCET

Worst-case scheduling interference
Response-Time Analysis

- For each DAG-Task $\tau_i$,
  - For each vertex $v$ of $\tau_i$,
    - Each job of vertex $v$ must complete within a deadline $D_i$

$$e_v + I_v = R_v \leq D_i$$

Not easy to compute for multiprocessor systems!
Response-Time Analysis

- **Our approach**: compute an upper-bound $\overline{I_v}$ of the interference $I_v$ specific for each vertex $v$, so obtaining a response-time upper-bound $\overline{R_v}$

![Equation]

$$e_v + I_v = R_v \leq$$

$$e_v + \overline{I_v} \Rightarrow R_v \leq \overline{R_v}$$
Main result of this work: we proved that

\[ R_v \leq \overline{R}_v \]

\[
\overline{R}_v = \ell^+_v + \left\lfloor \frac{1}{m} \left( \sum_{v'} W_{v,v'}(\overline{R}_v,Y_{v'}) - \ell^+_v \right) \right\rfloor
\]

Critical path length: maximum sum of WCETs in a path ending with \( v \)
Critical Path

- **Critical path length**: maximum sum of WCETs in a path ending with \( \nu \)

\[
\ell_{\nu} = 13
\]

Diagram:

1. \( 1 \to 4 \to 2 \to 4 \) with \( 1+4+2+4 = 11 \)
2. \( 3 \to 6 \to 4 \) with \( 3+6+4 = 13 \)
Response-Time Analysis

**Main result:** we proved that

\[ R_v \leq \overline{R}_v \]

\[ \overline{R}_v = \ell^+_v + \left[ \frac{1}{m} \left( \sum_{v'} W_{v,v'}(\overline{R}_v, Y_{v'}) - \ell^+_v \right) \right] \]

Sum on all vertices \( v' \) in the task-set
Response-Time Analysis

**Main result**: we proved that

\[ R_v \leq \overline{R_v} \]

\[
\overline{R_v} = \ell^+_v + \left[ \frac{1}{m} \left( \sum_{v'} W_{v,v'}(\overline{R_v}, Y_{v'}) - \ell^+_v \right) \right]
\]

Upper-bound on the worst-case workload generated by \( v' \) interfering with \( v \)
Worst-Case Workload

- Upper-bound on the worst-case workload generated by \( \nu' \) interfering with \( \nu \)

\[
W_{\nu, \nu'}(\overline{R_\nu}, Y_{\nu'})
\]

Tentative response-time of vertex \( \nu \), used in the fixed-point iteration starting with \( \overline{R_\nu} = e_\nu \)

Response-time upper-bound
Must be always greater than the response-time
(\( Y_{\nu'} = D_\nu + 1 \) in the limit case)
Worst-Case Workload

- A generic vertex $v'$ interferes with $v$ released at $t$

If shifted more on the left the job of $v'$ will be completed when $v$ is released.

Release of a job of $v$
Worst-Case Workload

- A generic vertex $v'$ interferes with $v$ released at $t$

Interfering workload:

$$\left[\frac{Y_{v'} + R_v}{T_{v'}}\right] e_{v'}$$

In case of **G-DM**: Null for vertices of lower-priority tasks
Worst-Case Workload

- A generic vertex $v'$ interferes with $v$ released at $t$

**In case of G-EDF**

Jobs of $v'$ released after $t + D_v - D_{v'}$ will not interfere with $v$
Worst-Case Workload

- A generic vertex \( v' \) interferes with \( v \) released at \( t \)

In case of G-EDF

Interfering workload

\[
\left[ Y_{v'} + \min\{ \bar{R}_v, D_v - D_{v'} \} \right] \frac{e_{v'}}{T_{v'}}
\]
Response-Time Analysis

- **Successors** in the same job of a DAG-task cannot interfere
Main result: we proved that

\[ R_v \leq \overline{R}_v \]

\[ \overline{R}_v = \ell_v^+ + \left[ \frac{1}{m} \left( \sum_{v'} w_{v,v'}(\overline{R}_v, Y_{v'}) - \ell_v^+ \right) \right] \]
Schedulability Test

**Algorithm** $\text{RTA}(N)$  
Maximum number of iterations
Schedulability Test

**Algorithm RTA\((N)\)**

1. We start with \( Y_v = D_v + 1, \forall v, i = 1 \)

2. Compute the fixed-point of

\[
\overline{R_v} = \ell_v^+ + \left[ \frac{1}{m} \left( \sum_{v'} W_{v,v'}(\overline{R_{v'}},Y_{v'}) - \ell_v^+ \right) \right]
\]

3. If \( \overline{R_v} \leq D_v \) return **SCHEDULABLE**

4. If \( Y_v = \overline{R_v}, \forall v \) OR \( i==N \) return **NOT SCHEDULABLE**

5. Else, update response-times as \( Y_v = \overline{R_v}, \forall v \) and go to step 2

\( i++ \)

Pseudo-Polynomial Complexity
Polynomial-Time Schedulability Test

- If we set $Y_v = D_v + 1$ and $R_v = D_v$ it is possible to obtain a simple polynomial-time schedulability test without involving any iteration.

\[
R_v = \ell_v^+ + \left[ \frac{1}{m} \left( \sum_{v'} W_{v,v'}(D_v, D_{v'} + 1) - \ell_v^+ \right) \right]
\]

Polynomial Complexity
Augmentation Bound

In case of a task-set composed of a single DAG-Task \((N=1)\) we proved that

- Our test based on response-time analysis has
  - Augmentation bound \(< 3\) for G-EDF;
  - Augmentation bound \(< 5\) for G-DM.
Experimental Results

- The proposed schedulability tests have been evaluated by using *synthetic workload*.

- *libdag* – DAG-Tasks generator and schedulability test. *Soon publicly available online!*

- Comparison against the test based on augmentation bound proposed in V. Bonifaci, A. Marchetti-Spaccamela, S. Stiller, and A. Wiese. “Feasibility analysis in the sporadic DAG task model”. In proc. of ECRTS 2013

  *To the best of our knowledge it is the only test dealing with arbitrary deadlines*
Experimental Results

Number $N$ of external iterations in our algorithm

The test of Bonifaci et al. is based on a workload approximation up to an $\epsilon$-error with $2^{-\delta}$
### Experimental Results

#### Running times of the schedulability tests

<table>
<thead>
<tr>
<th></th>
<th>Min (s)</th>
<th>Max (s)</th>
<th>Avg (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTA(64)</td>
<td>0.001</td>
<td>0.397</td>
<td>0.014</td>
</tr>
<tr>
<td>RTA(16)</td>
<td>0.001</td>
<td>0.225</td>
<td>0.014</td>
</tr>
<tr>
<td>RTA(4)</td>
<td>0.001</td>
<td>0.050</td>
<td>0.009</td>
</tr>
<tr>
<td>RTA(1)</td>
<td>0.001</td>
<td>0.014</td>
<td>0.005</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Min (s)</th>
<th>Max (s)</th>
<th>Avg (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BON(10)</td>
<td>0.000</td>
<td>17.855</td>
<td>3.357</td>
</tr>
<tr>
<td>BON(6)</td>
<td>0.000</td>
<td>1.160</td>
<td>0.214</td>
</tr>
<tr>
<td>BON(4)</td>
<td>0.000</td>
<td>0.292</td>
<td>0.051</td>
</tr>
<tr>
<td>BON(2)</td>
<td>0.000</td>
<td>0.142</td>
<td>0.012</td>
</tr>
</tbody>
</table>

RTA test has running time lower of two orders of magnitude.

Exponential increase of the running time as the test precision increases.

*(Intel Xeon @ 3.5 Ghz)*
Experimental Results

**Take-away messages**

- RTA test outperforms the speed-up based test in all the tested configurations;
- In some cases our polynomial-time test performs better than the speed-up based test that has pseudo-polynomial complexity.
Conclusions

- We proposed a new **Response-Time Analysis** for the sporadic DAG-Task model under both G-EDF and G-DM scheduling;

- The analysis handles DAG-Tasks with **arbitrary deadline** and arbitrary utilization;

- Two schedulability tests have been derived (pseudo-polynomial and polynomial complexity);

- Extensive set of **experimental results** confirmed the effectiveness of the test.
Future Work

- More accurate characterization of the *interfering workload*;
- Support for *conditional* statements in the DAG-Task;
- Integration of *locking protocols* in the analysis;
- Handle *distributed* computations.
Thank you!

Alessandro Biondi
alessandro.biondi@sssup.it