Matrix-DBP for (m,k)-firm real time guarantee
Ye-Qiong Song

To cite this version:
Ye-Qiong Song. Matrix-DBP for (m,k)-firm real time guarantee. Séminaire du Programme de Recherches Avancées Franco-Chinois PRA SI01-04, Oct 2002, Zhejiang University, China. pp.35. inria-00107605

HAL Id: inria-00107605
https://inria.hal.science/inria-00107605
Submitted on 19 Oct 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Matrix-DBP For (m, k)-firm Real-Time Guarantee

Based on a common work with E. Poggi, YQ. Song, Z. Wang, A. Koubâa
Partially supported by PRA SI01-04

• Part 1: context
• Part 2: where are the problems
• Part 3: Matrix-DBP
• Part 4: future work

Presentation at Zhejiang University
Part 1: (m,k)-firm background
Why we interest in \((m,k)\)-firm?

- HRT uses WC analysis leading to oversized system design (pessimistic performance estimation)
- In practice, many systems being classified HRT are not so « hard ». Occasional deadline misses can be tolerated if they are correctly distributed
Why we interest in (m,k)-firm?

- SRT can resolve the over-sizing problem but does not specify how deadlines can missed.
- A probabilistic guarantee of m/k may not be sufficient (e.g. MPEG video stream composed of I, B and P packets).

Video source

Can tolerate at most 1 deadline miss of B every 10 packets
Probability of 9/10 can not be suitable
Why we interest in (m,k)-firm?

• (m,k)-firm guarantees the deadline meeting of \( m \) among any \( k \) consecutive task invocations or message transmissions

• (m,k)-firm provides a general frame for RT constraints description:
  – HRT is a particular case of (m,k)-firm with \( m=k \)
  – SRT guarantee with probability \( p = m/k \) is a particular case of (m,k)-firm with:
    \[
    p = \lim_{m,k \to \infty} \frac{m}{k}
    \]
(m,k)-firm and WHRT

• (m,k)-firm (i.e. at least m of k meets) is firstly introduced by [Hamdaoui95]

• WHRT is defined by [Bernat98] and [Bernat&Burns01] to generalize (m,k)-firm:
  – \( (\overline{m}, \overline{k}) \): at most \( m \) misses. But \( (m, k) = (\overline{k - m}, \overline{k}) \)
  – \( <m,k> \): in any \( k \) consecutive jobs, at least \( m \) consecutive jobs are with deadline meet
  – \( \langle \overline{m}, \overline{k} \rangle = \langle \overline{m} \rangle \): never \( m \) consecutive deadline misses (notice that \( (m, k) \leq \langle k - m \rangle \) )
(m,k)-firm

- Failure state in (m,k)-firm context
- Example of (2,3)-firm
- \( \mu \)-pattern

111 | 110 | 101 | 011
---|---|---|---
0 | 1 | 1 | 0

1 meet deadline | 0 miss deadline

25-Oct-02 at Zhejiang Univ.  Yeqiong SONG
DBP: Distance Based Priority

• DBP of a μ-pattern is defined as the number of deadline misses to arrive to a failure state

• Example: (3,5)-firm
  – (11011) has a distance of 2
  – (10111) has a distance of 3
  – (10001) has a distance of 0
DBP in MIQSS model

\[
\tau_1 = (\delta_{i-1}^{1}, \delta_{i}^{1}, \delta_{i+1}^{1}), \ldots, j_{i+3}^{1}, j_{i+2}^{1}, j_{i+1}^{1} \quad \text{DBP} \quad j_{i+1}^{1}, p_{i+1}^{1}
\]

\[
\tau_2 = (\delta_{i-1}^{2}, \delta_{i}^{2}, \delta_{i+1}^{2}), \ldots, j_{i+3}^{2}, j_{i+2}^{2}, j_{i+1}^{2} \quad \text{DBP} \quad j_{i+1}^{2}, p_{i+1}^{2}
\]

\[
\vdots
\]

\[
\tau_N = (\delta_{i-1}^{N}, \delta_{i}^{N}, \delta_{i+1}^{N}), \ldots, j_{i+3}^{N}, j_{i+2}^{N}, j_{i+1}^{N} \quad \text{DBP} \quad j_{i+1}^{N}, p_{i+1}^{N}
\]

\(j_{i}^{x} \): \(i\)th instance of job \(x\)
\(p_{i}^{x} \): priority of \(i\)th instance of job \(x\)

- A job is modeled by: \(\tau_i = \{D_i, c_i, m_i, k_i\}\)
Gain of DBP for (m,k)-firm

- Dynamic QoS control is possible with:
  - In case of overload, DBP allows on-line drop down of jobs according to \((m_i, k_i)\), reducing thus the need of resources, i.e.:
    \[
    \sum_{i=1}^{N} \left[ \frac{c_i m_i}{T_i k_i} \right] \leq 1
    \]
    instead of
    \[
    \sum_{i=1}^{N} \left[ \frac{c_i}{T_i} \right] \leq 1
    \]
  - Can be extended to \((m(t), k(t))\) when on-line scheduling algorithm is adopted
Part 2: where are problems?
Problems of DBP in MIQSS for RT

• A job is modeled by: \( \tau_i = \{T_i, D_i, c_i, m_i, k_i\} \)
• Pb1: how to choose when several jobs have the same priority? (DBP+EDF?)
• Pb2: how to take into account the relative importance among jobs of all sources?
### Un example of Pb2

<table>
<thead>
<tr>
<th></th>
<th>(m,k)-constraint</th>
<th>Service time (ms)</th>
<th>Period/Deadline</th>
<th>Initial ( \mu )-pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sa</strong></td>
<td>(4,5)</td>
<td>15</td>
<td>30</td>
<td>{01111}</td>
</tr>
<tr>
<td><strong>Sb</strong></td>
<td>(2,5)</td>
<td>2</td>
<td>5</td>
<td>{00101}</td>
</tr>
</tbody>
</table>

Processing firstly Sa will lead to failure state of Sb!

So notion of relative criticality should be defined
Need of new non-preemptive scheduling algo.

• Should take into account:
  – (m,k)-firm constraint
  – Other temporal parameters & constraints: $T_i, D_i, c_i$
  – Relative criticality between streams (a matrix can be used to represent it)

• Should be of low implementation cost

Matrix-DBP can meet all these requirements
Part 3: Matrix-DBP
Offset between jobs invocations

(2,5)-firm

Case A

Case B
When happens 1 deadline miss

Best case for j
When happens 1 deadline miss

\[ c_i \left( T_j - c_j \right) \]
When happens 1 deadline miss

\[ c_i \leq T_j - c_j \]
When happens 1 deadline miss

\[ c_i - (T_j - c_j) > D_j - c_j, \]

then 1 miss in the best case.

i.e., at least 1 miss
When happens 1 deadline miss

\[ c_i - (T_j - c_j) \leq T_j + D_j - c_j \]

then no more than 1 miss
When happens 1 deadline miss

\[
\begin{align*}
    c_i - (T_j - c_j) &> D_j - c_j, & \text{At least 1 miss} \\
    c_i - (T_j - c_j) &\leq T_j + D_j - c_j. & \text{No more than 1 miss}
\end{align*}
\]
Generalization to $n$ deadline misses

\[
c_i - (T_j - c_j) > (n_{j,i} - 1) \cdot T_j + D_j - c_j,
\]
\[
c_i - (T_j - c_j) \leq n_{j,i} \cdot T_j + D_j - c_j.
\]

which gives \[x-1 \leq n < x\]
Minimum number of deadline misses

$$n_{j,i} = \left\lfloor \frac{c_i + 2c_j - D_j}{T_j} \right\rfloor - 1.$$ 

Minimum number of jobs that stream $j$ misses while a job of stream $i$ is being served

$$M = \begin{bmatrix} \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \end{bmatrix} \quad \text{N} \times \text{N}$$
Necessary conditions

Necessary Load condition

\[ \sum_{i=1}^{N} \left[ \frac{c_i}{T_i} \frac{m_i}{k_i} \right] \leq 1 \]

Mutual feasibility condition

\[ \forall i, j \leq N, \]
\[ n_{i,j} \leq k_i - m_i \]

New Necessary Condition
Matrix-DBP

<table>
<thead>
<tr>
<th></th>
<th>(m, k)</th>
<th>Execution time</th>
<th>Period Deadline</th>
<th>Initial μ-pattern</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stream a</td>
<td>(4, 5)</td>
<td>15ms</td>
<td>30ms</td>
<td>01111</td>
<td>2</td>
</tr>
<tr>
<td>Stream b</td>
<td>(2, 5)</td>
<td>2ms</td>
<td>5ms</td>
<td>00101</td>
<td>3</td>
</tr>
</tbody>
</table>

\[
M = \begin{bmatrix}
0 & 0 \\
2 & 0
\end{bmatrix}
\]

Priority_{a,b} = DBP_a - n_{a,b} = 2

Priority_{b,a} = DBP_b - n_{b,a} = 1
Simulation scenario

<table>
<thead>
<tr>
<th>Stream</th>
<th>(m,k)-constraints</th>
<th>Job Process time: c_i</th>
<th>Period (T_i)/Deadline(D_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stream 0</td>
<td>(2,5)</td>
<td>8/c</td>
<td>12</td>
</tr>
<tr>
<td>Stream 1</td>
<td>(4,5)</td>
<td>10/c</td>
<td>20</td>
</tr>
<tr>
<td>Stream 2</td>
<td>(3,6)</td>
<td>2/c</td>
<td>5</td>
</tr>
<tr>
<td>Stream 3</td>
<td>(1,5)</td>
<td>4/c</td>
<td>6</td>
</tr>
</tbody>
</table>
Matrix of the scenario

- When $c = 1$

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

- When $c = 1.5$

Matrix-DBP becomes DBP

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$
Simulation result

Failure State Pr.

Server resources

Distance Based Priority
Matrix Distance Based Priority

Server resources

0%
5%
10%
15%
20%
25%
Contributions on (m,k)-firm

• Pointed out the drawback of the classic DBP when it is applied to a more general real-time context
• Provided an additional necessary condition call mutual schedulability test
• Proposed matrix-DBP to correct DBP
Part 4: Future work
On Matrix-DBP for (m,k)-firm

- Analysis of necessary condition on $c$ for (m,k)-firm guarantee (i.e. find the value of $c$ from which the system always has 0 failure state)
- Taking other criteria than the global failure state percentage (e.g., fairness in failure state percentage for each stream)
- Extension to multi-hops case by taking global deadline and local deadlines
On \((m,k)\)-firm

- Enhancement of DBP for aperiodic streams
- \((m,k)\)-firm for a \((\sigma,\rho)\) upper bounded traffic
Application domains of (m,k)-firm

• Handling real-time traffic in Switched Industrial Ethernet (an Intranet) and in Internet

• Evaluating dependability in process control systems (in which we suppose that the state variables are over-sampled vs. Nyquist frequency)
Questions and remarks?