Stability Conditions of On-line Resource Managers for Systems with Execution Times Variations

Sergiu Rafiliu, Petru Eles, Zebo Peng

Department of Computer and Information Science, Linköping University, Sweden

July 29, 2011
Stability Conditions of On-line Resource Managers for Systems with Execution Times Variations
Preliminaries

Stability Conditions of On-line Resource Managers for Systems with Execution Times Variations

Real-Time System

Tasks (have variable resource requirements)
Stability Conditions of On-line Resource Managers for Systems with Execution Times Variations

Real-Time System

Tasks (have variable resource requirements)

Resource Manager
Preliminaries

Stability Conditions of On-line Resource Managers for Systems with Execution Times Variations

Is this system stable?
Motivation

On-line Resource Managers

- React to resource demand (variable execution times)
- Control the state of the system (response times, latency, jitter, throughput, load, ...).
- Other goals (e.g. Optimize some performance QoS metric)

Stability means that the Resource Manager controls the system such that the resource demand (state) is bounded.
Motivation

**On-line Resource Managers**

- **React to resource demand** *(variable execution times)*
- **Control the state of the system** *(response times, latency, jitter, throughput, load, ...)*.
- **Other goals** *(e.g. Optimize some performance QoS metric)*

**Stability** means that the Resource Manager controls the system such that the resource demand (state) is bounded.
Outline

1. System Description
2. Stability and Problem Formulation
3. Approach
   - System Model
   - Main Results
4. Applications
   - Usage
   - Stability of Existing Resource Managers
   - Bounds for Stable Systems
5. Conclusions and Future Work
System Description

- Execution times ($c$) vary in unknown ways.
- Task rates ($\rho$) are decided by the Resource Manager (RM).
- Jobs are scheduled through any scheduling policy that:
  - executes the jobs of each task in the order of their release.
  - is non-idling.
System Description

- Execution times \((c)\) vary in unknown ways
- Task rates \((\rho)\) are decided by the **Resource Manager** (RM)
- Jobs are scheduled through any scheduling policy that:
  - executes the jobs of each task in the order of their release
  - is non-idling.
System Description

Tasks are modeled as queues of jobs, that wait to be executed by the scheduler.
System Description

Tasks are modeled as queues of jobs, that wait to be executed by the scheduler.

- not overload
- overload
Resource Demand

\[ (\text{predicted}) D^u_{[k]} = \frac{C_{\text{previous}} + C_{\text{current}}}{t_{[k+1]} - t_{[k]}} \]
Resource Manager

- Runs at key moments in time: $t[k], t[k+1], t[k+2], \ldots$
- Reacts to variations in resource demand ($u^D$),
- Adjusts task rates ($\rho_i$) to new values.
<table>
<thead>
<tr>
<th></th>
<th>System Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Stability and Problem Formulation</td>
</tr>
<tr>
<td>3</td>
<td>Approach</td>
</tr>
<tr>
<td></td>
<td>System Model</td>
</tr>
<tr>
<td></td>
<td>Main Results</td>
</tr>
<tr>
<td>4</td>
<td>Applications</td>
</tr>
<tr>
<td></td>
<td>Usage</td>
</tr>
<tr>
<td></td>
<td>Stability of Existing Resource Managers</td>
</tr>
<tr>
<td></td>
<td>Bounds for Stable Systems</td>
</tr>
<tr>
<td>5</td>
<td>Conclusions and Future Work</td>
</tr>
</tbody>
</table>
Stability - Definition (from Control Theory)

**Definition (Stability in the sense of Lagrange)**

A system is stable, if at any moment in time, the system’s state is within a bounded distance from the set of stationary points.

**Stationary Points** are states where we wish our system to spend most of its time. In our case, any state where the *resource demand* \( (u^D) \) is 1 is a stationary point.
Problem Formulation

Determine a criterion that the Resource Manager must satisfy, in order for the system to be stable.
1 System Description

2 Stability and Problem Formulation

3 Approach
   - System Model
   - Main Results

4 Applications
   - Usage
   - Stability of Existing Resource Managers
   - Bounds for Stable Systems

5 Conclusions and Future Work
System Model

System of difference equations

\[ F(\overline{x}_{k+1}, \overline{x}_k) = 0 \]

\[ \overline{x}_k = \begin{pmatrix} \Phi_k \\ \rho_k \\ c_k \\ q_k \\ \vdots \end{pmatrix} \]

- offsets
- rates
- predicted execution times
- queue sizes
System Model

\[\sum_{i \in I} c_i[k+1] \cdot q_i[k+1] = h \cdot \max\{0, u_i^R[k+1] - 1\}\]

\[\phi_i[k+1] = \phi_i[k] + \frac{1}{\rho_i[k+1]} \left[\rho_i[k+1] \cdot \max\{0, h - \phi_i[k]\}\right] - h\]

\[\overline{c}[k+1] = f_p(c_i[k]) + \nu[k]\]

\[\overline{q}^p[k+1] = \overline{q}[k]\]

\[\overline{\phi}^p[k+1] = \overline{\phi}[k]\]

\[\overline{\rho}[k+1] = f_c(x[k])\]
System Model

- resource manager
  \[ \bar{\rho}_{k+1} = f_c(\bar{x}_k) \]

- execution time prediction
  \[ \bar{c}_{k+1} = f_p(\bar{x}_k) + \bar{\nu}_k \]

- model of the scheduling policy
  \[ \sum_{i \in I} c_{i[k+1]} \cdot q_{i[k+1]} = h \cdot \max\{0, u_{k+1}^R - 1\} \]
Main Results

Stability - Intuition

\[ \overline{x}[k] = \left( \begin{array}{c} \phi[k] \\ \rho[k] \\ c[k] \\ q[k] \\ \vdots \\ \vdots \end{array} \right) \]

\begin{align*}
\text{← offsets} & \quad \text{← rates} \\
\text{← predicted execution times} & \quad \text{← queue sizes}
\end{align*}

\{ \text{bounded above and below} \}

\{ \text{unbounded above} \}
Theorem 2 (Necessary condition)

A necessary condition for a system to be stable is:

\[ \sum_{i \in I} c_i^{\text{max}} \cdot \rho_i^{\text{min}} \leq 1 \]

- All jobs of all tasks could have their worst-case execution times
- There must exist a setting for task rates, such that, in this conditions, the load does not exceed 1
Main Results

Stability - Criterion

Theorem 2 (Necessary condition)

A necessary condition for a system to be stable is:

$$\sum_{i \in I} c_i \max \cdot \rho_i \min \leq 1$$

The set of all rate vectors that guarantee that the resource demand drops is:

$$\Gamma^\star = \left\{ \overline{\rho} \in \mathbb{P} \left| \sum_{i \in I} c_i \max \cdot \rho_i \leq 1 \right. \right\}$$
Main Results

Stability - Criterion

**Theorem 3**

For any general system that satisfies Theorem 2 a sufficient stability condition is that the Resource Manager satisfies:

\[
\bar{\rho}_{[k+1]} \in \begin{cases} 
\Gamma_\star, & \text{if } u_{[k]}^D \geq u_\Omega^D \\
\mathbf{P}, & \text{otherwise}
\end{cases}
\]  

(1)

\[
\rho_{i[k+1]} \leq \rho_{i[k]} \quad \text{if } u_{[k]}^D \geq u_\Omega^D, \quad \forall i \in I
\]  

(2)
Stability Criterion – Proof

**Theorem 1 (Uniform boundedness)**

A system is uniformly bounded if:

- $d(\bar{x}[k], M) \geq \Omega \Rightarrow \exists V : \mathcal{X} \rightarrow \mathbb{R}_+ \text{ and } \varphi_1, \varphi_2 : \mathbb{R}_+ \rightarrow \mathbb{R}_+; \varphi_1, \varphi_2 \uparrow,$
  \[\lim_{r \rightarrow \infty} \varphi_i(r) = \infty, i = 1, 2\]
  \[\varphi_1(d(\bar{x}, M)) \leq V(\bar{x}) \leq \varphi_2(d(\bar{x}, M)) \text{ and } V(\bar{x}_{[k+1]}) \leq V(\bar{x}_{[k]})\]

- $d(\bar{x}[k], M) < \Omega \Rightarrow \exists \Psi > 0 \text{ s.t } d(\bar{x}_{[k+1]}, M) < \Psi$

\[u^D_\Omega \Rightarrow \Omega, \Psi\]
\[u^D(\bar{x}) \Rightarrow V(\bar{x}) \text{ (upper bound)}\]

$\Omega$ – ultimate bound
$\Psi$ – bound on the real-time performance of the system
Stability Criterion – Proof

Theorem 1 (Uniform boundedness)

A system is uniformly bounded if:

1. \( d(\bar{x}[k], \mathcal{M}) \geq \Omega \Rightarrow \exists V : \mathcal{X} \rightarrow \mathbb{R}_+ \) and \( \varphi_1, \varphi_2 : \mathbb{R}_+ \rightarrow \mathbb{R}_+; \varphi_1, \varphi_2 \uparrow, \lim_{r \to \infty} \varphi_i(r) = \infty, i = 1, 2 \)

\[
\varphi_1(d(\bar{x}, \mathcal{M})) \leq V(\bar{x}) \leq \varphi_2(d(\bar{x}, \mathcal{M})) \quad \text{and} \quad V(\bar{x}_{[k+1]}) \leq V(\bar{x}_{[k]})
\]

2. \( d(\bar{x}[k], \mathcal{M}) < \Omega \Rightarrow \exists \psi > 0 \) s.t \( d(\bar{x}_{[k+1]}, \mathcal{M}) < \psi \)

\[
\begin{align*}
u_D^\Omega & \Rightarrow \Omega, \psi \\
u_D^D(\bar{x}) & \Rightarrow V(\bar{x}) \quad \text{(upper bound)}
\end{align*}
\]

\( \Omega \) – ultimate bound

\( \psi \) – bound on the real-time performance of the system
1 System Description

2 Stability and Problem Formulation

3 Approach
   - System Model
   - Main Results

4 Applications
   - Usage
   - Stability of Existing Resource Managers
   - Bounds for Stable Systems

5 Conclusions and Future Work
Usage

Three ways:

- Select $u^D_\Omega$ and construct a resource manager,
- Take an existing resource manager and show that there exists a $u^D_\Omega$, or
- Modify an existing resource manager by considering a $u^D_\Omega$

$u^D_\Omega \Rightarrow \Psi$ – bound on the real-time performance of the system.
Stability of Existing Resource Managers

Stability of Existing Resource Manager

- QRAM\textsuperscript{1}
- Corner-Case\textsuperscript{2}
- QoS Derivative\textsuperscript{2}


Stability of QRAM

Each time QRAM runs, it takes the following steps:

- set all rates to $\rho_i^{\text{min}}$ and compute $u^D$ with the assumed rates.
- if $u^D < u^D_\Omega$ select a tasks $\tau_i$ according with some given QoS curves
- increase $\tau_i$’s rate to the next defined rate point on the curve, or until $u^D = u^D_\Omega$
- repeat this process until $u^D = u^D_\Omega$ or the rates of all tasks are $\rho_i^{\text{max}}$

Observation

$u^D > u^D_\Omega \iff \bar{\rho} = (\rho_1^{\text{min}} \rho_2^{\text{min}} \rho_3^{\text{min}} \cdots)^T \in \Gamma_\ast \Rightarrow$ the system is stable.
Stability of Existing Resource Managers

Stability of QRAM

Each time QRAM runs, it takes the following steps:

- set all rates to $\rho_{i}^{\text{min}}$ and compute $u^{D}$ with the assumed rates.
- if $u^{D} < u_{\Omega}^{D}$ select a tasks $\tau_{i}$ according with some given QoS curves
- increase $\tau_{i}$’s rate to the next defined rate point on the curve, or until $u^{D} = u_{\Omega}^{D}
- repeat this process until $u^{D} = u_{\Omega}^{D}$ or the rates of all tasks are $\rho_{i}^{\text{max}}$

Observation

$u^{D} > u_{\Omega}^{D} \iff \bar{\rho} = (\rho_{1}^{\text{min}} \rho_{2}^{\text{min}} \rho_{3}^{\text{min}} \cdots)^{T} \in \Gamma_{\star} \Rightarrow$ the system is stable.
Bounds for Stable Systems

Assume

- EDF scheduler
- Stable system, with \( u_{[k]}^{D} \leq u^{D} \)

Results

- Bound on queue sizes:
  \[
  q_{j}^{\text{max}} = \frac{h}{c_{j}^{\text{min}}} \cdot u_{R}^{\max}(u^{D})
  \]

- Bound on response times:
  \[
  r_{j}^{\text{max}} = \frac{1}{\rho_{j}^{\text{min}}} + h \cdot \left( u_{\max}^{R}(u^{D}_{\Omega}) - \frac{1}{h} \sum_{i \in I} c_{i}^{\max} \right) + \sum_{i \in I} c_{i}^{\max} \cdot \left[ \frac{\rho_{i}^{\max}}{\rho_{j}^{\text{min}}} \right]
  \]
Conclusions

- Proposed a stability criterion for a uniprocessor system with independent task set.
- Shown how to apply the criterion for a number of existing Resource Managers.
- Shown how the stability criterion can be linked with real-time properties (e.g. bound on execution times).
Current/Future Work

- more classes of schedulers
- tasks with different task modes
- allow job dropping
- more flexible release method
- different real-time goals
Current/Future Work

Distributed architectures composed of

- several resources (CPUs, buses, ...)
- task graphs distributed on the resources

\[
\begin{align*}
    u_a^D(X) \\
    u_b^D(X) \\
    u_c^D(X)
\end{align*}
\]
Current/Future Work

Distributed architectures composed of
- several resources (CPUs, buses, …)
- task graphs distributed on the resources
Current/Future Work

Distributed architectures composed of

- several resources (CPUs, buses, …)
- task graphs distributed on the resources

Data dependencies impose huge problems.
Questions???
Theorem 1 (Uniform boundedness)

A system is uniformly bounded if:

- \( d(\bar{x}_{[k]}, \mathcal{M}) \geq \Omega \Rightarrow \exists V : \mathcal{X} \rightarrow \mathbb{R}_+ \text{ and } \varphi_1, \varphi_2 : \mathbb{R}_+ \rightarrow \mathbb{R}_+; \varphi_1, \varphi_2 \uparrow, \lim_{r \rightarrow \infty} \varphi_i(r) = \infty, i = 1, 2 \)

\[
\varphi_1(d(\bar{x}, \mathcal{M})) \leq V(\bar{x}) \leq \varphi_2(d(\bar{x}, \mathcal{M})) \text{ and } V(\bar{x}_{[k+1]}) \leq V(\bar{x}_{[k]})
\]

- \( d(\bar{x}_{[k]}, \mathcal{M}) < \Omega \Rightarrow \exists \psi > 0 \text{ s.t } d(\bar{x}_{[k+1]}, \mathcal{M}) < \psi \) (3)
Stability - Intuition

\[ x[0] \] – initial state

\[ \mathcal{M} \] – region of stationary points

\[ x[0] \]
Stability - Intuition

Ball of size $\Omega$ around $\mathcal{M}$
Stability - Intuition

Ball of size $\Psi$ around $\mathcal{M}$
**Comparison of Models**

<table>
<thead>
<tr>
<th>Model</th>
<th>Controller</th>
<th>Example 1</th>
<th>Example 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$h$</td>
<td>$r_1^{\text{max}}$</td>
</tr>
<tr>
<td>Constrained</td>
<td>C1</td>
<td>20</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>C2</td>
<td>20</td>
<td>28</td>
</tr>
<tr>
<td>General</td>
<td>C1</td>
<td>4</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>C2</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>C1</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>C2</td>
<td>2</td>
<td>10</td>
</tr>
</tbody>
</table>
Response Time Result

$$r_j^{\text{max}} = \frac{1}{\rho_j^{\text{min}}} + h \cdot \left( u_R^{\text{max}}(u_D^\Omega) - \frac{1}{h} \sum_{i \in I} c_i^{\text{max}} \right) + \sum_{i \in I} c_i^{\text{max}} \cdot \left[ \frac{\rho_i^{\text{max}}}{\rho_j^{\text{min}}} \right]$$