Component-Based Software Design
Hierarchical Real-Time Scheduling
lecture 1/4

Enrico Bini

March 10, 2015

Outline

1. Introduction to the course
2. Composition of real-time systems
3. Scheduling problems, background
4. Supply function, definition
5. Supply function, properties
Info

- Please interrupt me anytime if something is unclear;
- Topics: hierarchical real-time scheduling over single/multi-cores.
- My classes:
  
<table>
<thead>
<tr>
<th>Day</th>
<th>Date</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tue</td>
<td>2015-03-10</td>
<td>11:00–13:00</td>
</tr>
<tr>
<td>Wed</td>
<td>2015-03-11</td>
<td>10:00–13:00</td>
</tr>
<tr>
<td>Tue</td>
<td>2015-03-24</td>
<td>11:00–13:00</td>
</tr>
<tr>
<td>Wed</td>
<td>2015-03-25</td>
<td>10:00–13:00</td>
</tr>
</tbody>
</table>

  - start at +10 min,
  - depending on the topic break for b minutes, during classes;
  - end around $-15 \times h + 10 + b$ minutes, with $h$ number of hours.

- Exam: one/two questions in the written examination.
- Material: slides, your notes.

Outline

1. Introduction to the course
2. Composition of real-time systems
3. Scheduling problems, background
4. Supply function, definition
5. Supply function, properties
### Motivations

Applications sharing the same machine. Issues:
- a misbehavior in App \( a \) (running longer than expected, etc.) may cause misbehaviors on Apps \( b, c \)
- priority of application tasks needs to be comparable
  - may violate intellectual properties
  - when a new application joins, priorities of new tasks need to be assigned w.r.t. other tasks

### Solution

- *global scheduler* \( \mathcal{A} \) assigns physical machine to *virtual machines*
- tasks scheduled over VMs by *local scheduler*
- misbehavior confined at app boundary
- task prio meaningful only within app
Hierarchical real-time scheduling

Focus of my lectures:

*how can real-time constraints of components be guaranteed over VMs?*

Starting point:

1. Model of a real-time component (periodic tasks, deadlines, etc, we assume you are familiar with it)
2. Model of a scheduler (rules that determine what applications, tasks to execute over the platform)
3. Model of the VM: how is a virtual machine for real-time application characterized? (main focus of my classes)

Outline

1. Introduction to the course
2. Composition of real-time systems
3. Scheduling problems, background
4. Supply function, definition
5. Supply function, properties
A virtual machine is characterized by the amount of resource assigned to it by the global scheduler.

Review of scheduling problems (to be applied at global scheduler level).

Basic elements of a scheduling problem:

- a set \( \mathcal{N} \) of VMs requiring work to be made. Since they are finite, we represent them by \( \mathcal{N} = \{1, 2, \ldots, n\} \);
- a set \( \mathcal{M} \) of (physical) machines capable to perform some work. Since they are finite we represent them by \( \mathcal{M} = \{1, 2, \ldots, m\} \);
- a time set \( \mathcal{T} \), over which the scheduling is performed (in my classes always \([0, \infty)\), in general it may be \( \mathbb{N} \));

A schedule \( S \) of \( \mathcal{N} \) over \( \mathcal{M} \) is a function

\[
S : \mathcal{M} \times \mathcal{T} \rightarrow \mathcal{N} \cup \{0\}
\]

with “0” denoting idle.

Graphical representation of a schedule

- If \( S(k, t) = i \) then the machine \( k \) is assigned to the \( i \)-th VM at time \( t \).
- If \( S(k, t) = 0 \) then the machine \( k \) is not assigned at time \( t \) (we say that the \( k \)-th machine is idle at \( t \)).

This definition of schedule implies that at every instant \( t \) each machine is assigned to at most one VM.

Conversely, at every instant each VM may run over any number of machines in \( \mathcal{M} \).
VM schedule

An equivalent representation of a schedule $S$ is given by a set of $n + 1$ VM schedule functions

$$i \in \mathcal{N} \cup \{0\} \quad s_i : \mathcal{T} \rightarrow 2^\mathcal{M}$$

such that

$$\forall t \in \mathcal{T}, \forall i \neq j, \quad s_i(t) \cap s_j(t) = \emptyset$$

$$\bigcup_{i=0}^{n} s_i(t) = \mathcal{M}$$

that is no machine can be used simultaneously by two VMs.

- A schedule $S$ and a set of VMs schedules $\{s_1, \ldots, s_n\}$ are equivalent through

$$\forall t \in \mathcal{T}, k \in \mathcal{M}, \quad S(k, t) = i \iff k \in s_i(t)$$

Outline

1. Introduction to the course
2. Composition of real-time systems
3. Scheduling problems, background
4. Supply function, definition
5. Supply function, properties
From now on, we assume a single processor. Also, we drop the index $i$ of the VM, since we consider only one in isolation.

- $m = |M| = 1$,
- the schedule is just $S : T \rightarrow N \cup \{0\}$
- the VM schedule functions $s_i(t)$ made by the global scheduler are
  - if $s_i(t) = 1$ the processor is allocated to the VM,
  - if $s_i(t) = 0$ the processor is not allocated to the VM.

Legal schedules

Given a global scheduling algorithm $A$, we denote by $\mathcal{L}_*$ the set of all possible legal VM scheduling functions $s_i(t)$, defined as

$$\mathcal{L}_* = \{s_0, s_1, \ldots, s_n\},$$

$s_i(t)$ may be generated by $A$,

while the set $\mathcal{L}_i$ of legal scheduling functions for $i$-th VM is defined as a projection of $\mathcal{L}_*$, that is

$$\mathcal{L}_i(P) = \{s_i: 1 \leq i \leq n, (s_1, \ldots, s_n) \in \mathcal{L}_*\}.$$

Example:

- $\mathcal{L}_i$ legal schedules of a VM implemented by a periodic servers with period $P_i$ and budget $Q_i$.
  - then $s_i \in \mathcal{L}_i$ is such that...
Supply functions

Definition (supply lower bound function of a VM)
We define the supply lower bound function \( \text{slbf}(t) \) of a VM as

\[
\text{slbf}(t) = \min_{t_0 \in T, s \in \mathcal{L}} \int_{t_0}^{t_0+t} s(x) \, dx.
\]

Informally, “minimum amount of resource made available by the VM in every interval of length \( t \)”.

Definition (supply upper bound function of a VM)
We define the supply upper bound function \( \text{subf}(t) \) of a VM as

\[
\text{subf}(t) = \max_{t_0 \in T, s \in \mathcal{L}} \int_{t_0}^{t_0+t} s(x) \, dx.
\]

Outline

1. Introduction to the course
2. Composition of real-time systems
3. Scheduling problems, background
4. Supply function, definition
5. Supply function, properties
Usage of supply bounds

Theorem (Resource bounds)

For any legal VM schedule $s \in \mathcal{L}$, it is always

$$\text{slbf}(b - a) \leq \int_{a}^{b} s(t) \, dt \leq \text{subf}(b - a).$$

How long does it take to compute a work $W$ over a VM?

- **worst-case response time** $R_w(W)$ of work $W$ over a VM
  $$R_w(W) = \sup \{ t : \text{slbf}(t) < W \},$$

- **best-case response time** $R_b(W)$ of work $W$ over a VM
  $$R_b(W) = \inf \{ t : \text{subf}(t) \geq W \}.$$
Example slbf(t), subf(t): static partition 1

- Let us assume the static VM schedule
  \[ s(t) = \begin{cases} 1 & 0 \leq (t \mod 4) < 1 \\ 0 & \text{otherwise} \end{cases} \]

- What are its slbf(t) and subf(t)?
  - Remember: “any interval of length t” not necessarily [0, t]
  - What are \( R_b(2) \) and \( R_w(2) \)?

Resource assigned with a given pattern

If resource/idle alternate according to a known sequence:
- \( S(k) \), with \( k = 1, 2, \ldots \) is the sequence of budgets
- \( Z(k) \), with \( k = 1, 2, \ldots \) is the sequence of idle intervals

Let \( t = 0 \) coincide with the start of the first budget \( S(1) \)
slbf(t) with given pattern

**Theorem (slbf of sequence of budgets/idle)**

\[
\text{slbf}(t) \geq \min \{t - \sigma_Z(n), \sigma_S(n)\}, \quad t \in I_n, n \in \mathbb{N}
\]

with the sequence of intervals \(\{I_n\}_{n \in \mathbb{N}}\) defined as

\[
I_n = \begin{cases} 
[0, \sigma_Z(1)] & n = 0 \\
[\sigma_Z(n) + \sigma_S(n - 1), \sigma_Z(n + 1) + \sigma_S(n)] & n \geq 1
\end{cases}
\]

and with

\[
\sigma_S(n) = \inf_{n_0} \sum_{k=n_0}^{n_0+n-1} S(k),
\]

\[
\sigma_Z(n) = \sup_{n_0} \sum_{k=n_0}^{n_0+n-1} Z(k).
\]

**Illustration of slbf**

\[
\text{slbf}(t) \geq \min \{t - \sigma_Z(n), \sigma_S(n)\}, \quad t \in I_n, n \in \mathbb{N}
\]

\[
I_n = \begin{cases} 
[0, \sigma_Z(1)] & n = 0 \\
[\sigma_Z(n) + \sigma_S(n - 1), \sigma_Z(n + 1) + \sigma_S(n)] & n \geq 1
\end{cases}
\]

**Remark:** the worst cases for \(S(k)\) and \(Z(k)\) are considered independently, leading to potential extra pessimism (see next example)
Example slbf \((t)\), subf \((t)\): static partition 2

- Let us assume the static VM schedule

\[
s(t) = \begin{cases} 
1 & \text{if } (t \mod 12) \in [2, 3) \cup [5, 7) \cup [10, 12) \\
0 & \text{otherwise}
\end{cases}
\]

- What are its slbf \((t)\) and subf \((t)\)?
  - Remember: the interval with the minimum supply can start at different points for different \(t\)

- What is \(W\) such that \(R_w(W) - R_b(W)\) over the VM, is maximized?

Example slbf \((t)\), subf \((t)\): periodic server

- Let us assume that
  - the VM is implemented by a periodic server
  - the global scheduler guarantees a budget \(Q\) every period \(P\)

- What is its slbf \((t)\)? Remember: scenario of minimum possible supply must be assumed, among all legal schedules in \(\mathcal{L}\)
Example slbf\( (t) \), subf\( (t) \): periodic server

- Let us assume that
  - the VM is implemented by a periodic server
  - the global scheduler guarantees a budget \( Q \) every period \( P \)

- What is its slbf\( (t) \)? Remember: scenario of minimum possible supply must be assumed, among all legal schedules in \( \mathcal{L} \)

\[
\text{slbf}(t) = \begin{cases} 
0 & t \in [0, P - Q] \\
(k - 1)Q & t \in (kP - Q, (k + 1)P - 2Q] \\
(t - (k + 1)(P - Q)) & \text{otherwise}
\end{cases}
\]

with \( k = \left\lfloor \frac{t - (P - Q)}{P} \right\rfloor \).

Properties of slbf\( (t) \)

1. \( \text{slbf}(0) = 0 \) (no resource in empty interval),
2. \( \forall s \geq t \geq 0, \text{slbf}(s) \geq \text{slbf}(t) \) (non-decreasing)
3. \( \forall s, t \geq 0, \text{slbf}(s + t) \geq \text{slbf}(s) + \text{slbf}(t) \) (super-additivity)
4. \( \forall s \geq t \geq 0, \text{slbf}(s) - \text{slbf}(t) \leq s - t \) (Lipschitz continuous with factor 1):
Exploiting properties

Let us assume that at some instant $t^*$ we know that

$$\text{slbf}(t^*) \geq c^*, \quad (1)$$

From Lipschitz-continuity and $(1)$, we find

$$\forall t \in [0, t^*] \quad \text{slbf}(t) \geq \begin{cases} 
0 & 0 \leq t \leq t^* - c^* \\
(\text{slbf}(t^*) + (t - t^*)) & t^* - c^* < t \leq t^*
\end{cases}$$

Let $q$ and $r$ be quotient and rest of the Euclidean division of any $t \geq 0$ by $t^*$, that is

$$t = qt^* + r, \quad q \in \mathbb{N}, \ r \in [0, t^*).$$

Then, for any $t \geq 0$, the following lower bound holds

$$\text{slbf}(t) = \text{slbf}(t^* + \cdots + t^* + r) \geq$$

$$\geq \underbrace{\text{slbf}(t^*) + \cdots + \text{slbf}(t^*)}_{q \text{ times}} + \text{slbf}(r) = qc^* + \text{slbf}(r).$$