Component-Based Software Design

Hierarchical Real-Time Scheduling
lecture 2/4

Enrico Bini

March 11, 2015

Component-Based Software Design
Enrico Bini
Linear supply bounds
The design problem
Parameters of the VM
Feasible VM parameters
Cost of a VM

Outline

Component-Based Software Design
Enrico Bini
Linear supply bounds
The design problem
Parameters of the VM
Feasible VM parameters
Cost of a VM

Linear lower bounds

- The exact supply functions may be too complicated to be handled.
- Supply lower bound function slbf(t) can be lower bounded by a linear function (**add drawing**):

The supply lower bound function slbf(t) is lower bounded by any of the following functions

$$\text{lslbf}(t) = \max\{0, \alpha(t - \Delta)\}$$

with

$$\alpha \le \lim_{t\to\infty} \frac{\text{slbf}(t)}{t}, \qquad \Delta \ge \sup_{t\ge 0}\left\{t - \frac{\text{slbf}(t)}{\alpha}\right\}$$

although $\Delta > \dots$ introduces a unnecessary pessimism.
Notice that the linear lower bound is not unique.

Component-Based Software Design
Enrico Bini
Linear supply bounds
The design problem
Parameters of the VM
Feasible VM parameters
Cost of a VM

Linear lower bounds: properties

- α is often called *bandwidth* of the VM;
- Δ is often called *delay* of the VM, as the application may need to wait up to Δ before receiving the resource with rate α
  - $\Delta \ge \sup\{t : \text{sbf}(t) = 0\}$ always.
- gain in simplicity: only two numbers (α, Δ) to abstract a VM;
- gain in generality: these two numbers can be computer for any global scheduler $\mathcal{A}$
- loss in accuracy: there is some gap between the exact slbf(t) and its linear lower bound

Component-Based Software Design
Enrico Bini
Linear supply bounds
The design problem
Parameters of the VM
Feasible VM parameters
Cost of a VM

lblsf: Example

1. What is the lslbf(t) of a periodic (P, Q) server?

Component-Based Software Design
Enrico Bini
Linear supply bounds
The design problem
Parameters of the VM
Feasible VM parameters
Cost of a VM

lblsf: Example

1. What is the lslbf(t) of a periodic (P, Q) server?

$$\alpha = \frac{Q}{P}, \qquad \Delta = 2(P - Q)$$

lslsf: Example

1. What is the lslbf(t) of a periodic (P, Q) server?

$$\alpha = \frac{Q}{P}, \qquad \Delta = 2(P - Q)$$

2. What is the lslbf(t) of

$$[1,2] \cup [3,6] \quad \text{with period } 6$$

lslsf: Example

1. What is the lslbf(t) of a periodic (P, Q) server?

$$\alpha = \frac{Q}{P}, \qquad \Delta = 2(P - Q)$$

2. What is the lslbf(t) of

$$[1,2] \cup [3,6] \quad \text{with period } 6$$

$$\alpha = \frac{2}{3}, \qquad \Delta = 1.5 \ (> \text{longest idle time})$$

Outline

1. Linear supply bounds
2. The design problem
3. Parameters of the VM
4. Feasible VM parameters
5. Cost of a VM

The problem

What is the **best** *VM which can* **schedule** *a given* **real-time application***?*

The choice of the "best" VM depends of:

1. how is the application modeled
2. what is the local scheduling algorithm to schedule the application over the VM
3. what do we mean by "best"

Application and local scheduler

- We model an application by a set of n tasks (C_i, T_i, D_i)
  - T_i period
  - D_i deadline
  - C_i computation time $(C \leq D_i)$
- Local scheduler may be FP, EDF

Outline

1. Linear supply bounds
2. The design problem
3. Parameters of the VM
4. Feasible VM parameters
5. Cost of a VM

VM model

- The VM is characterized by a set of parameters ψ
  - could be budget Q and period P if VM implemented by a periodic server
  - could be bandwidth α and delay Δ if resource provided by VM is abstracted by linear supply function
  - we can transform a (P,Q) model into a (α,Δ) model via the following transformations:

$$\begin{cases}\alpha &= \frac{Q}{P}\\ \Delta &= 2(P-Q)\end{cases} \Leftrightarrow \begin{cases}Q &= \frac{\alpha\Delta}{2(1-\alpha)}\\ P &= \frac{\Delta}{2(1-\alpha)}\end{cases}$$

- To solve the design problem a cost $J(\psi)$, function of the VM parameters ψ must be specified
  - typical example: the consumed bandwidth. If $\psi = (P,Q)$, with Q budget and P period of the VM, then

$$J(Q,P) = \frac{Q}{P}$$

Feasible VM parameters

- A selection of VM parameters ψ determines the supply functions. Let us denote by

$$\mathsf{slbf}_\psi(t)$$

the supply lower bound function of the VM with param ψ.

- For any pair (t,w), $t,w \ge 0$, let us define the region $F_\psi(t,w)$ of feasible parameters ψ as

$$F_\psi(t,w) = \{\psi : \mathsf{slbf}_\psi(t) \ge w\}$$

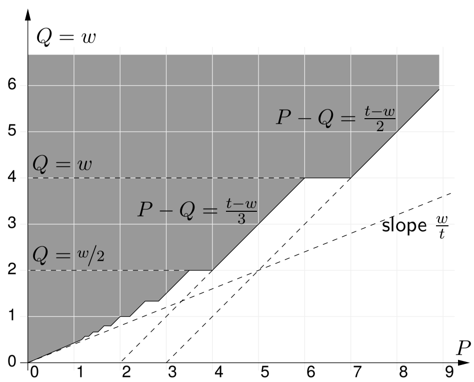
- Later, by combining regions $F_\psi(t,w)$, we are going to formulate the exact region Φ_ψ of feasible VM parameters.

Examples:

- $\psi = (P,Q)$, VM with budget Q and period P
- what is $F_{(P,Q)}(10,4)$?
- **draw the supply function**

Example of $F_{(P,Q)}(t,w)$

Plotting $F_{(P,Q)}(10,4)$



Expression of $F_{(P,Q)}(t,w)$

In general

$$F_{(P,Q)}(t,w) = \bigcup_{k=1}^{\infty} \left\{(P,Q) \in \mathbb{R}^2 : Q \ge \frac{w}{k},\ Q \ge P - \frac{t-w}{k+1}\right\}$$

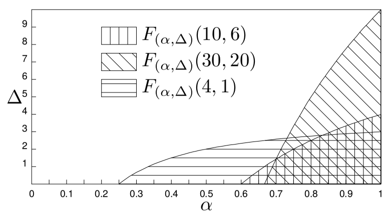
VM parameters are (α,Δ)

If the VM parameters are $\psi = (\alpha,\Delta)$, with

- α, bandwidth of the VM, and
- Δ, delay of the VM,

then the expression of the $F_{(\alpha,\Delta)}(t,w)$ can be computed more simply

$$F_{(\alpha,\Delta)}(t,w) = \{(\alpha,\Delta) : \overbrace{\alpha(t-\Delta)}^{\mathsf{slbf}_{(\alpha,\Delta)}(t)} \ge w\} = \left\{(\alpha,\Delta) : \Delta \le t - \frac{w}{\alpha}\right\}$$



Outline

FP-schedulable VM parameters 1/2

Theorem (FP-schedulability)

A constrained deadline (with $D_i \le T_i$) task set is FP-schedulable over a VM with parameters ψ, **if and only if**

$$\forall i = 1,\dots,n,\ \exists t \in \mathcal{P}_{i-1}(D_i),\quad C_i + \sum_{j=1}^{i-1}\left\lceil \frac{t}{T_j}\right\rceil C_j \le \mathsf{slbf}_\psi(t)$$

with $\mathcal{P}_j(t)$ being a set recursively defined as

$$\begin{cases} \mathcal{P}_0(t) = \{t\} \\ \mathcal{P}_j(t) = \mathcal{P}_{j-1}\left(\left\lfloor \frac{t}{T_j}\right\rfloor T_j\right) \cup \mathcal{P}_{j-1}(t). \end{cases}$$

Component-Based Software Design
Enrico Bini
Linear supply bounds
The design problem
Parameters of the VM
Feasible VM parameters
Cost of a VM

FP-schedulable VM parameters 2/2

$$\forall i = 1,\dots,n,\ \exists t \in \mathcal{P}_{i-1}(D_i),\quad C_i + \sum_{j=1}^{i-1}\left\lceil \frac{t}{T_j}\right\rceil C_j \le \mathsf{slbf}_\psi(t)$$

$$\bigwedge_{i=1}^{n} \bigvee_{t\in\mathcal{P}_{i-1}(D_i)} \quad C_i + \sum_{j=1}^{i-1}\left\lceil \frac{t}{T_j}\right\rceil C_j \le \mathsf{slbf}_\psi(t)$$

$$\bigwedge_{i=1}^{n} \bigvee_{t\in\mathcal{P}_{i-1}(D_i)} \quad \psi \in F_\psi\left(t, C_i + \sum_{j=1}^{i-1}\left\lceil \frac{t}{T_j}\right\rceil C_j\right)$$

$$\psi \in \bigcap_{i=1}^{n} \bigcup_{t\in\mathcal{P}_{i-1}(D_i)} F_\psi\left(t, C_i + \sum_{j=1}^{i-1}\left\lceil \frac{t}{T_j}\right\rceil C_j\right),$$

that is the set of VM param ψ guaranteeing the app $\{(C_1,T_1,D_1),\dots,(C_n,T_n,D_n)\}$ when the local sched is FP

Component-Based Software Design
Enrico Bini
Linear supply bounds
The design problem
Parameters of the VM
Feasible VM parameters
Cost of a VM

EDF-schedulable VM parameters

Theorem (EDF-schedulability)

A constrained deadline (with $D_i \le T_i$) task set is EDF-schedulable over a VM with parameters ψ, **if and only if**

$$\forall t \in \mathcal{D} \quad \sum_{i=1}^{n} \max\left\{0, \left\lfloor \frac{t+T_i-D_i}{T_i}\right\rfloor\right\} C_i \le \mathsf{slbf}_\psi(t)$$

with

$$\mathcal{D} = \{d_{i,k} : d_{i,k} = kT_i + D_i,\ i = 1,\dots,n,\ k\in\mathbb{N},\ d_{i,k} \le D^*\}$$

and $D^* = \mathsf{lcm}(T_1,\dots,T_n) + \max_i\{D_i\}$.

Analogously to the FP case the set of feasible VM parameters ψ is

$$\psi \in \bigcap_{t\in\mathcal{D}} F_\psi\left(t, \sum_{i=1}^{n}\max\left\{0,\left\lfloor\frac{t+T_i-D_i}{T_i}\right\rfloor\right\}C_i\right)$$

Component-Based Software Design
Enrico Bini
Linear supply bounds
The design problem
Parameters of the VM
Feasible VM parameters
Cost of a VM

Summary of feasible ψ

Let us denote by Φ_ψ the set of feasible VM parameters ψ. Then:

- if FP is local scheduler, then

$$\Phi_\psi = \bigcap_{i=1}^{n} \bigcup_{t\in\mathcal{P}_{i-1}(D_i)} F_\psi\left(t, C_i + \sum_{j=1}^{i-1}\left\lceil \frac{t}{T_j}\right\rceil C_j\right)$$

- if EDF is local scheduler, then

$$\Phi_\psi = \bigcap_{t\in\mathcal{D}} F_\psi\left(t, \sum_{i=1}^{n}\max\left\{0,\left\lfloor\frac{t+T_i-D_i}{T_i}\right\rfloor\right\}C_i\right)$$

Component-Based Software Design
Enrico Bini
Linear supply bounds
The design problem
Parameters of the VM
Feasible VM parameters
Cost of a VM

Example of EDF-schedulable VM parameters

- Let us assume to have the following task set (C_i, T_i, D_i)
  - $\{(1,3,3),(1,4,4),(1,12,12)\}$
- Local scheduler is EDF
- Let us choose the parameters $\psi = (\alpha,\Delta)$ to represent a VM
- We aim at computing

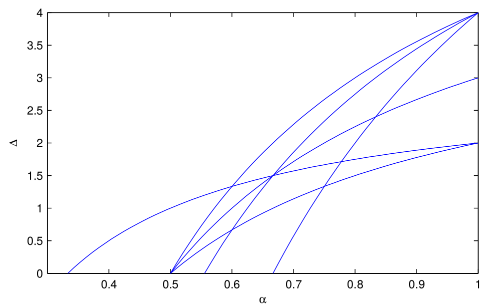
$$(\alpha,\Delta) \in \bigcap_{t\in\mathcal{D}} F_{(\alpha,\Delta)}\left(t, \sum_{i=1}^{n}\max\left\{0,\left\lfloor\frac{t+T_i-D_i}{T_i}\right\rfloor\right\}C_i\right)$$

- we need to compute the set of deadlines $\mathcal{D}$
- all the (t,w) pairs for $t\in\mathcal{D}$.

Component-Based Software Design
Enrico Bini
Linear supply bounds
The design problem
Parameters of the VM
Feasible VM parameters
Cost of a VM

Illustration of EDF-sched (α,Δ)

- $\mathcal{D} = \{3,4,6,8,9,12\}$
- pairs are $(t,w) \in \{(3,1),(4,2),(6,3),(8,4),(9,5),(12,8)\}$
- the corresponding region is



- the only relevant pairs are in $(4,2)$ and $(12,8)$

Outline

VM design problem: recap

- The cost $J(\psi)$ of a VM with parameters ψ determines the "best" parameters ψ^*
- The design problem of a VM is formulated as optimization problem

$$\text{minimize } J(\psi) \quad \text{subject to } \psi \in \Phi_\psi$$

with Φ_ψ being the set of feasible VM parameters, that is function of
  - the application parameters C_i, T_i, D_i
  - the local scheduler FP/EDF

Cost $J(\psi)$ if $\psi = (P, Q)$

- If $\psi = (P, Q)$, what is the cost function that it is reasonable to minimize?
- $J = \frac{Q}{P}$
  - good motivation (consumed resource is the cost)
  - however, it may lead to impractical implementations
  - $J = Q/P$ implies that $P^{\text{opt}}, Q^{\text{opt}} \to 0$ (**prove based on showing that slbf is never smaller if period is divided by 2**)
  - $P = 0$ can never be physically achieved.
- To prevent $P = 0$ cases, we can consider to minimize the really consumed bandwidth that is

$$J = \underbrace{\frac{Q}{P}}_{\text{useful}} + \underbrace{\frac{c}{P}}_{\text{wasted}}$$

with c equal to the overhead of switching in and out a VM.

Cost $J(\psi)$ if $\psi = (\alpha, \Delta)$

- If $\psi = (\alpha, \Delta)$, then the cost

$$J = \frac{Q}{P} + \frac{c}{P}$$

can be written as function of (α, Δ), though the transformation $(P, Q) \Rightarrow (\alpha, \Delta)$, as follows

$$J = \alpha + c\frac{2(1-\alpha)}{\Delta}$$

Solving optimal design problem, $\psi = (\alpha, \Delta)$

- If $\psi = (\alpha, \Delta)$ then optimal VM design is solved by

$$\text{minimize } \alpha + c\frac{2(1-\alpha)}{\Delta} \quad \text{subject to } (\alpha, \Delta) \in \Phi_{(\alpha,\Delta)}$$

  - if local scheduler is EDF, then $\Phi_{(\alpha,\Delta)}$ is convex because intersection of convex sets. Cost function is quasi-convex. Hence the problem can be solved efficiently
  - if local scheduler is FP, then $\Phi_{(\alpha,\Delta)}$ is not convex. Ad-hoc methods needed

Solving optimal design problem, $\psi = (P, Q)$

- If $\psi = (P, Q)$ then optimal VM design is solved by

$$\text{minimize } \frac{Q}{P} + \frac{c}{P} \quad \text{subject to } (P, Q) \in \Phi_{(P,Q)}$$

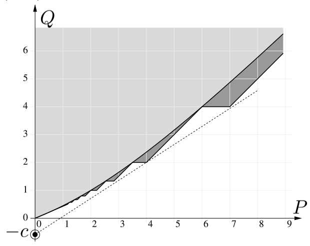
- **Remark**: the level curves of $\frac{Q}{P} + \frac{c}{P}$ in the (P, Q)-plane are lines through the point $(P, Q) = (0, -c)$.

$$\frac{Q}{P} + \frac{c}{P} = k$$
$$Q + c = kP$$

which is true for any k if $Q = -c$ and $P = 0$.

Solving optimal design problem, $\psi = (P, Q)$

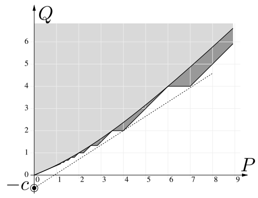
- Problem illustrated below in the simple case of $\Phi_{(P,Q)} = F_{(P,Q)}(10, 4)$



- in light grey the $F_{(\alpha,\Delta)}(10, 4)$ transformed into the (P, Q) variables
- The nature of the problem does not change as $\Phi_{(P,Q)}$ is intersection/union of some $F_{(P,Q)}(t, w)$

Component-Based Software Design
Enrico Bini
Linear supply bounds
The design problem
Parameters of the VM
Feasible VM parameters
Cost of a VM

Solving optimal design problem, $\psi = (P, Q)$



Optimal (P, Q) is:

- convex hull of the outer points $\mathcal{C} = [(P_1^*, Q_1^*), (P_2^*, Q_2^*), \ldots]$ ordered from right to left
- if $c > P_1^* - Q_1^*$, then best is $Q = P$ (fully dedicated VM)
- (P_k^*, Q_k^*) is opt **iff**

$$c \in \left[\frac{P_{k+1}^* Q_k^* - Q_{k+1}^* P_k^*}{P_k^* - P_{k+1}^*}, \frac{P_k^* Q_{k-1}^* - Q_k^* P_{k-1}^*}{P_{k-1}^* - P_k^*} \right]$$