Advanced Real-Time Systems

Lecture 6/6

Enrico Bini

November 9, 2012

Outline

Motivations

- Applications often need to be isolated, otherwise a misbehaviour on one (such as taking longer than expected) can cause misbehaviors on other applications
- Example of high priority task executing for longer

Solutions

- The execution of each application is controlled by a mechanism (aka server, resource reservation), which prevents the application from running more than planned
- Such a mechanism provides the abstraction of a *virtual resource*: a resource which is not always fully available
- How can we guarantee the timing constraints in presence of such a mechanism?

Outline

Application Model

- We model an application by a set of n tasks (C_i, T_i, D_i)
  - C_i computation time
  - T_i period
  - D_i deadline
- All tasks are scheduled by some scheduling algorithm (FP, EDF, etc.) **over a the virtual resource**. Such a scheduling alg. is often called *local scheduler* (local to the virtual resource)
- How to we check schedulability over virtual resources?

Resource model

- The model of a virtual resource must capture the "not-fully-available" characteristic.

Definition

We define the *supply bound function* $\mathsf{sbf}(t)$ of a virtual resource as the minimum amount of execution time available in any interval of length t.

Example 1: dbf(t)

- Let us assume a virtual resource that provides execution time (to the application) in

$$\bigcup_{k\in\mathbb{Z}}[4k, 4k+1]$$

- What is its sbf(t)?
- Remember: "any interval of length t" not necessarily $[0, t]$

Example 2

- Let us now assume that the virtual resource is implemented by a periodic server with period (=deadline) P, time budget Q
- What is its sbf(t)
- Remember: scenario of minimum possible supply must be assumed

Example 2

- Let us now assume that the virtual resource is implemented by a periodic server with period (=deadline) P, time budget Q
- What is its $\mathrm{sbf}(t)$
- Remember: scenario of minimum possible supply must be assumed

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

with $k = \left\lceil \frac{t-(P-Q)}{P} \right\rceil$.

- This example explains why this is often called *hierarchical scheduling* since it is about to schedule a task set within another task

Example 3

- Let us assume a virtual resource that provides execution time (to the application) in

$$[2,3] \cup [5,7] \cup [10,12] \quad \text{with period } 12$$

- Remember: the interval with the minimum supply can start at different points for different t

Outline

Schedulability conditions

Theorem (FP-schedulability [?])

A constrained deadline (with $D_i \leq T_i$) task set is FP-schedulable over a virtual resource with $\mathsf{sbf}(t)$, **if and only if**

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

Theorem (EDF-schedulability [?])

The task set $\mathcal{N}$ is EDF-schedulable over a virtual resource with $\mathsf{sbf}(t)$, **if and only if***:*

$$\forall t \geq 0 \quad \sum_{i=1}^{n} \max\left\{0, \left\lfloor \frac{t + T_i - D_i}{T_i} \right\rfloor \right\} C_i \leq \mathsf{sbt}(t)$$

Lower bounding sbf

- Testing with the exact sbf(t), for example

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

  with $k = \left\lceil \frac{t-(P-Q)}{P} \right\rceil$,
  may be too complicated.
- It is safe (sufficient) to make the test by using a **lower** bound to sbf(t)

Linear lower bound

Given a supply bound function $\mathsf{sbf}(t)$, it can be lower bounded [**?**] by any of the following functions

$$\mathsf{lsbf}(t) = \max\{0, \alpha(t - \Delta)\}$$

with

$$\alpha \leq \lim_{t\to\infty} \frac{\mathsf{sbf}(t)}{t}$$

and

$$\Delta = \sup_{t\geq 0} \left\{ t - \frac{\mathsf{sbf}(t)}{\alpha} \right\}$$

Typically we take $\alpha = \ldots$.

- α is often called *bandwidth* of the virtual resource;
- Δ is often called *delay* of the virtual resource.
    - $\Delta \geq \sup\{t : \mathsf{sbf}(t) = 0\}$ always.

lbsf: Example

1. What is the $\mathsf{lsbf}(t)$ of a periodic (Q, P) server?

lbsf: Example

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

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

lbsf: Example

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

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

2. What is the lsbf(t) of

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

lbsf: Example

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

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

2. What is the lsbf(t) of

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

$$\alpha = \frac{5}{12}, \qquad \Delta = 1.5 \ (> \text{longest idle time})$$

Exercise

1. Can we schedule by FP the two (implicit deadline) tasks $C_1 = 2$, $T_1 = 7$ and $C_2 = 2$, $T_2 = 15$, over a virtual resource implemented by a periodic server with period $P = 4$ and budget $Q = 2$?
2. Is it FP-schedulable over the virtual resource, if it is abstracted by the lsbf?
3. Is it EDF-schedulable over the virtual resource, if abstracted by the sbf?
4. Is it EDF-schedulable over the virtual resource, if abstracted by the lsbf?

Outline

1. Hierarchical Scheduling: intro
2. System model
3. Schedulability conditions in hierarchical scheduling
4. Designing a virtual resource
5. Conclusions

Motivation

- Often the task set is given
- We just have to design the virtual platform parameters such that the application is schedulable, and
- the minimum amount of (real, physical) resource is consumed (the bandwidth α is minimal).

Design using lsbf

For any pair (t, w), $t, w \geq 0$, let us define the region of feasible parameters as

$$\begin{aligned} F_{\alpha,\Delta}(t,w) &= \{(\alpha,\Delta) \in \mathbb{R}^2 : \mathsf{lsbf}(t) \geq w, \alpha \geq 0\} \\ &= \{(\alpha,\Delta) \in \mathbb{R}^2 : \alpha(t-\Delta) \geq w, \alpha \geq 0\} \end{aligned}$$

If local scheduler is EDF, then the problem is:

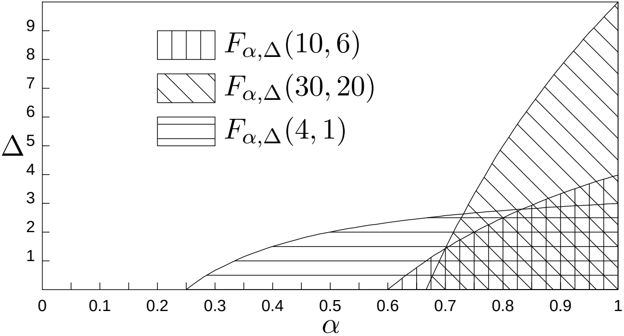
$$\begin{aligned} &\text{minimize } J(\alpha,\Delta) \\ &\text{s.t. } (\alpha,\Delta) \in \bigcap_{t\in\mathcal{D}} F_{\alpha,\Delta}(t,\mathsf{dbf}(t)) \end{aligned}$$

If local scheduler is FP, then the problem is:

$$\begin{aligned} &\text{minimize } J(\alpha,\Delta) \\ &\text{s.t. } (\alpha,\Delta) \in \bigcap_{i\in\mathcal{N}} \bigcup_{t\in\mathcal{P}_{i-1}(D_i)} F_{\alpha,\Delta}(t, C_i + I_i(t)) \end{aligned}$$

Geometry of the region

- $F_{\alpha,\Delta}(t, w)$ is convex



- If local scheduler is EDF the feasible region is convex.

Cost function

- What is the cost function that it is reasonable to minimize?
- $J = \alpha$
    - good motivation (as little bandwidth as possible)
    - however, it may lead to impractical implementations
    - $J = \alpha$ implies that $\Delta^{\text{opt}} = 0$
    - $\Delta = 0$ can never be physically achieved. For example, if a periodic server is used, then

$$P = \frac{\Delta}{2(1-\alpha)}$$

- To prevent $\Delta = 0$ cases, we can consider to minimize the really consumed bandwidth

$$J = \underbrace{\alpha}_{\text{useful}} + \underbrace{\frac{c}{P}}_{\text{wasted}} = \alpha + 2c\frac{1-\alpha}{\Delta}$$

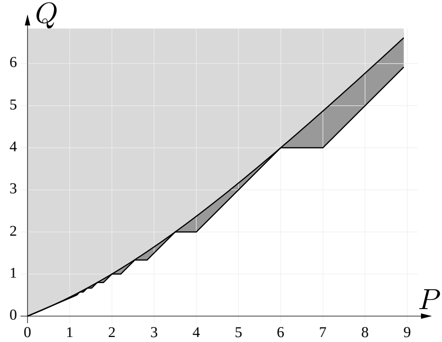
with c equal to some overhead. It is quasiconvex [**?**], hence good for optimization.

Design using sbf

- lsbf is a lower bound, hence the previous analysis is not optimal. Exact analysis should consider the exact shape of the sbf

$$F_{P,Q}(t,w) = \{(P,Q) \in \mathbb{R}^2 : \mathsf{sbf}_{(Q,\,P)\ \mathsf{server}}(t) \geq w\}$$

In the figure below $F_{P,Q}(10,4)$



sbf: minimum bandwidth problem

If local scheduler is EDF, then the problem is:

$$\begin{aligned}\text{minimize}&\frac{Q}{P}+\frac{c}{P}\\ \text{s.t. } &(P,Q)\in\bigcap_{t\in\mathcal{D}}F_{P,Q}(t,\mathsf{dbf}(t))\end{aligned}$$

If local scheduler is FP, then the problem is:

$$\begin{aligned}\text{minimize}&\frac{Q}{P}+\frac{c}{P}\\ \text{s.t. } &(P,Q)\in\bigcap_{i\in\mathcal{N}}\bigcup_{t\in\mathcal{P}_{i-1}(D_i)}F_{P,Q}(t,C_i+I_i(t))\end{aligned}$$

Level sets of the cost function are lines going through $P=0$ and $Q=-c$.

Outline

1. Hierarchical Scheduling: intro
2. System model
3. Schedulability conditions in hierarchical scheduling
4. Designing a virtual resource
5. Conclusions

Conclusion of the lecture

- Hierarchical scheduling is well motivated by the need to compose application which are designed and guaranteed in isolation
- The optimal selection of the parameters of the virtual resource was discussed
    - EDF over lsbf easy because convex
    - FP over lsbf complicated, but approachable
    - EDF over sbf (exact supply) messy
    - FP over sbf messy mess

Conclusion of the course

- With the simplest possible task model
- FP over uniprocessor resource
    - schedulability analysis
    - sensitivity analysis
    - optimal design of the task set
- EDF over uniprocessor resource
    - schedulability analysis
    - sensitivity analysis
    - optimal design of the task set
- FP+EDF over uniprocessor virtual resource
    - schedulability analysis
    - optimal design of the virtual resource
- If you want to publish in this area, you can complicate the task model, the scheduling algorithm, or the resource model according to your taste (possibly getting closer to the reality).