

Component-Based Software Design

Hierarchical Real-Time Scheduling

lecture 3/4

Enrico Bini

March 24, 2015

Outline

① Design problem

② Model of VM

The design problem over parallel VM

Premise: the terms multi-core, multiprocessor, parallel machine, etc. are all used interchangeably, as they all are machines capable to perform computation simultaneously.

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

Needed ingredients:

- A model of a parallel VM
- An application model
- Local scheduler of application over VM
- Notion of “best”
- Optimization method

Outline

① Design problem

② Model of VM

Main characteristic of multicore

Resource is available along *two dimensions*

- *horizontal* dimension over time
- *vertical* dimension over the number of processing units

In short:

- two machines of speed 0.5 and
- one machine of speed 1

are different.

Migration hypothesis:

- 1 migration/identical multiprocessor: the work can freely migrate over all CPUs and it takes the same amount of time
 - reasonable as first approximation
 - in reality migration has a non-uniform (cache hierarchies) cost

VM schedule

A schedule is modeled by a set of n *VM schedule functions*

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

such that

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

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

Equivalently: $s_{i,k}(t)$ schedule of i -th VM over k -th CPU at t

$$i \in \mathcal{N}, \quad s_{i,k} : \mathcal{T} \rightarrow \{0, 1\}$$

$$\forall t, k, \quad \sum_{i=1}^n s_{i,k}(t) \leq 1 \quad \text{core } k \text{ at } t \text{ to at most 1 VM}$$

Legal schedules

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

$$\mathcal{L}_* = \{(s_{i,k})_{i \in \mathcal{N}, k \in \mathcal{M}} : \text{scheduling functions } (s_{i,k})_{i \in \mathcal{N}, k \in \mathcal{M}} \text{ may be generated by } \mathcal{A}\},$$

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

$$\mathcal{L}_i = \{(s_{i,k})_{k \in \mathcal{M}} : (s_{i,k})_{i \in \mathcal{N}, k \in \mathcal{M}} \in \mathcal{L}_*\}.$$

Since we focus on one VM only, we drop the index i of the VM

$$\mathcal{L}_i \Rightarrow \mathcal{L}, \quad s_{i,k}(t) \Rightarrow s_k(t)$$

k is the index of the assigned physical machine

Parallel supply lower bound function

Definition (*parallel supply lower (bound) function of a VM*)

Given a VM, we define its *parallel supply lower bound function* $\text{pslf}_k(t)$ with maximum parallelism k as

$$\text{pslf}_k(t) = \min_{t_0 \in \mathcal{T}, (s_1, \dots, s_m) \in \mathcal{L}} \int_{t_0}^{t_0+t} \min \left\{ k, \sum_{\ell=1}^m s_\ell(x) \right\} dx.$$

[identical hyp means $\sum s_\ell$]

“minimum amount of resource made available by the VM in every interval of length t , with parallelism at most k ”.

Notice that $\text{pslf}_k(t)$ depends on

- 1 the length of the time interval t , and
 - 2 the level of parallelism k ,
- to represent the bi-dimensional nature of the resource.

Parallel supply upper bound function

Definition (*parallel supply upper (bound) function of a VM*)

Given a VM, we define its *parallel supply upper bound function* $\text{psuf}_k(t)$ with maximum parallelism k as

$$\text{psuf}_k(t) = \max_{t_0 \in \mathcal{T}, (s_1, \dots, s_m) \in \mathcal{L}} \int_{t_0}^{t_0+t} \min \left\{ k, \sum_{\ell=1}^m s_{\ell}(x) \right\} dx.$$

Properties

- 1 $\forall k, \text{pslf}_k(0) = 0$ (no resource in empty interval),
- 2 $\forall k, \forall s \geq t \geq 0, \text{pslf}_k(s) \geq \text{pslf}_k(t)$ (non-decreasing)
- 3 $\forall k, \forall s, t \geq 0, \text{pslf}_k(s+t) \geq \text{pslf}_k(s) + \text{pslf}_k(t)$
(super-additivity)
- 4 $\forall k, \forall s \geq t \geq 0, \text{pslf}_k(s) - \text{pslf}_k(t) \leq k(s-t)$ (Lipschitz
continuous with factor k):
- 5 when needed, we extend $\forall k \leq 0, \text{pslf}_k(t) = 0$
- 6 $\forall k, \forall t \geq 0, \text{pslf}_k(t) \leq \text{pslf}_{k+1}(t)$ (non-decr with k)
- 7 (def): $\bar{m} = \min\{k : \forall t, \text{pslf}_k(t) = \text{pslf}_{k+1}(t)\}$ is *max
parallelism* of the VM;
- 8 $\forall k, \forall t \geq 0, \text{pslf}_k(t) - \text{pslf}_{k-1}(t) \geq \text{pslf}_{k+1}(t) - \text{pslf}_k(t)$
(concavity) **[explanation]**
implies $\forall k, \forall t \geq 0, \frac{\text{pslf}_k(t)}{k} \geq \frac{\text{pslf}_{k+1}(t)}{k+1}$ **[by induction]**
- 9 $\text{pslf}_{\bar{m}}(b-a) \leq \int_a^b \sum_{k=1}^{\bar{m}} s_k(t) dt \leq \text{psuf}_{\bar{m}}(b-a)$

More properties

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

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

From Lipschitz-continuity and (1), we find

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

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

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

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

$$\begin{aligned} \text{pslf}_k(t) &= \underbrace{\text{pslf}_k(t^* + \dots + t^* + r)}_{q \text{ times}} \geq \\ &\geq \underbrace{\text{pslf}_k(t^*) + \dots + \text{pslf}_k(t^*)}_{q \text{ times}} + \text{pslf}_k(r) = qc^* + \text{pslf}_k(r). \end{aligned}$$

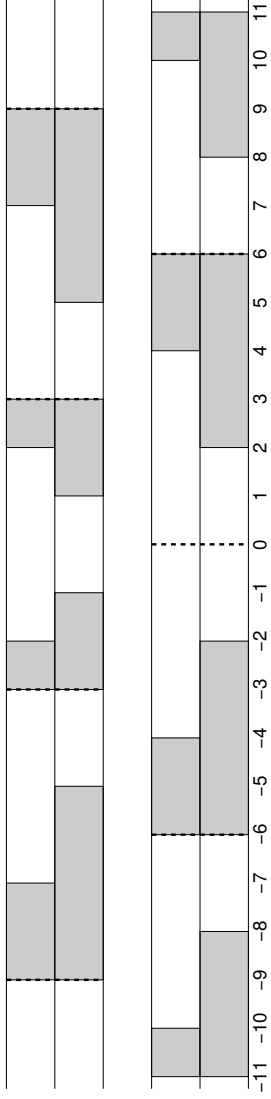
Examples

- 1 $\text{pslf}_k(t)$ of a VM with $\bar{m}$ dedicated processors?
- 2 $\text{pslf}_k(t)$ of a VM which provides Q budget, every period P , with parallelism at most $\bar{m}$?
- 3 $\text{pslf}_k(t)$ of a VM which provides $Q_1 = 4, Q_2 = 2$ budgets, on two CPUs, every period $P = 6$?

Examples

- 1 $\text{pslf}_k(t)$ of a VM with $\bar{m}$ dedicated processors?
- 2 $\text{pslf}_k(t)$ of a VM which provides Q budget, every period P , with parallelism at most $\bar{m}$?
- 3 $\text{pslf}_k(t)$ of a VM which provides $Q_1 = 4, Q_2 = 2$ budgets, on two CPUs, every period $P = 6$?

[Symmetric worst-case supply pattern (idea by Artem Burmyakov)]



How does it compare against $Q_1 = Q_2 = 3$ or against $Q_1 = 6, Q_2 = 0$?

Is even budget always worst?

If periods are not synchronized... ($P = 8, Q_1 + Q_2 = 8$)

