Multiprocessor Scheduling II: Global Scheduling

Prof. Dr. Jian-Jia Chen

LS 12, TU Dortmund

01, July, 2014

Global Scheduling

- We will only focus on identical multiprocessors in this module.
- The system has a global queue.
- A job can be migrated to any processor.
- Priority-based global scheduling:
  - Among the jobs in the global queue, the M highest priority jobs are chosen to be executed on M processors.
  - Task migration here is assumed no overhead.
  - Global-EDF: When a job finishes or arrives to the global queue, the M jobs in the queue with the shortest absolute deadlines are chosen to be executed on M processors.
  - Global-FP, Global-DM, Global-RM: When a job finishes or arrives to the global queue, the M jobs in the queue with the highest priorities (defined by fixed-priority ordering, deadline-monotonic strategy, or rate-monotonic strategy) are chosen to be executed on M processors.
- Pfair scheduling, and the variances (not discussed in this lecture).

Good News for Global Scheduling

- McNaughton's wrap-around rule for $P|pmtn|C_{\max}$ on M processors (historically, task migration is also called task preemption in the literature)
  - Compute $C_{\max}$ as $\max\{\max_{\tau_i \in \mathcal{T}} C_i, \frac{\sum_{\tau_i \in \mathcal{T}} C_i}{M}\}$
    - Assign the tasks according to any order from time 0 to $C_{\max}$
    - If a task's processing exceeds $C_{\max}$, the task is migrated to a new processor from time 0
    - Repeat the assignment of tasks until all the tasks are assigned
  - The resulting schedule minimizes $C_{\max}$

R. McNaughton. Scheduling with deadlines and loss functions. Management Science, 6:1-12, 1959.

Weakness of Partitioned Scheduling

- Restricting a task on a processor reduces the schedulability
- Restricting a task on a processor makes the problem $\mathcal{N}P$-hard
- The $\mathcal{N}P$-completeness for EDF does no hold any more if the migration has *no overhead*.
  - Proportionate Fair (pfair) algorithm introduced by Baruah et al. provides an optimal utilization bound for schedulibility
  - A task set with implicit deadlines is schedulable on M identical processors if the total utilization of the task set is no more than M.
  - The idea is to divide the time line into quanta, and execute tasks proportionally in each quanta.
  - It has very high overhead.
  - There are several variances to reduce the overhead.

Sanjoy K. Baruah, N. K. Cohen, C. Greg Plaxton, Donald A. Varvel: Proportionate Progress: A Notion of Fairness in Resource Allocation. Algorithmica 15(6): 600-625 (1996)

Bad News for Global Scheduling

For Global-EDF or Global-RM, the least upper bound for schedulability analysis is at most 1.

Input:

$M + 1$ tasks:

- One heavy task τ_k: $D_k = T_k = C_k$
- M light tasks τ_is: $C_i = \epsilon$ and $D_i = T_i = C_k - \epsilon$, in which ϵ is a positive number, very close to 0.

Sudarshan K. Dhall, C. L. Liu, On a Real-Time Scheduling Problem, OPERATIONS RESEARCH Vol. 26, No. 1, January-February 1978, pp. 127-140.

Bad News for Global Scheduling

For Global-EDF or Global-RM, the least upper bound for schedulability analysis is at most 1.

Input:

$M+1$ tasks:

- One heavy task τ_k: $D_k = T_k = C_k$
- M light tasks τ_is: $C_i = \epsilon$ and $D_i = T_i = C_k - \epsilon$, in which ϵ is a positive number, very close to 0.

Result:

The M light tasks (with higher priority than the light task) will be scheduled on M processors. The heavy task misses the deadline even when the utilization is $1 + M\epsilon$.

Sudarshan K. Dhall, C. L. Liu, On a Real-Time Scheduling Problem, OPERATIONS RESEARCH Vol. 26, No. 1, January-February 1978, pp. 127-140.

Gold Approach: Resource Augmentation

- The bad news on the least upper bound was very important in 80's, since the research in this direction suffered from the so called "Dhall effect".
- With resource augmentation, by Phillips et al., the "Dhall effect" disappears
  - For Global-EDF, the resource augmentation factor by "speeding up" is $2 - \frac{1}{M}$.
  - That is, if a feasible schedule exists on M processors, applying Global-EDF is also feasible on M processors by speeding up the execution speed with $2 - \frac{1}{M}$.
  - We will focus on schedulability test here first (for the first two parts) and the resource augmentation at the end.

Cynthia A. Phillips, Clifford Stein, Eric Torng, Joel Wein: Optimal Time-Critical Scheduling via Resource Augmentation. STOC 1997: 140-149

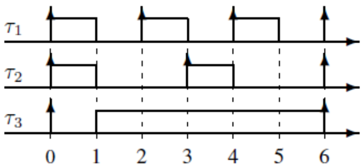
Articles for This Module

- Theodore P. Baker: Multiprocessor EDF and Deadline Monotonic Schedulability Analysis. RTSS 2003: 120-129 (*First part*)
- Marko Bertogna, Michele Cirinei, Giuseppe Lipari: Improved Schedulability Analysis of EDF on Multiprocessor Platforms. ECRTS 2005: 209-218
- Sanjoy K. Baruah: Techniques for Multiprocessor Global Schedulability Analysis. RTSS 2007: 119-128 (*Second part*)
- Nan Guan, Martin Stigge, Wang Yi, Ge Yu: New Response Time Bounds for Fixed Priority Multiprocessor Scheduling. IEEE Real-Time Systems Symposium 2009: 387-397
  - The analysis is tighter than the approaches presented here.
  - We won't have time to go through it.
- Vincenzo Bonifaci, Alberto Marchetti-Spaccamela, Sebastian Stiller, Andreas Wiese: Feasibility Analysis in the Sporadic DAG Task Model. ECRTS 2013: 225-233 (*Third part*)
  - Vincenzo Bonifaci, Alberto Marchetti-Spaccamela, Sebastian Stiller: A Constant-Approximate Feasibility Test for Multiprocessor Real-Time Scheduling. ESA 2008: 210-221

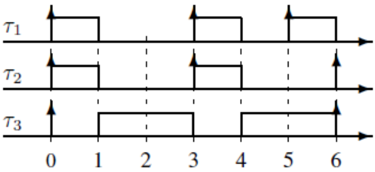
We will mainly focus on task sets with constrained deadlines.

Critical Instants?

- The analysis for uniprocessor scheduling is based on the gold critical instant theorem.
- Synchronous release of events does not lead to the critical instant for global multiprocessor scheduling
  - Suppose that there are 3 tasks: (C_i, D_i, T_i) are $\tau_1 = (1, 1, 2), \tau_2 = (1, 1, 3), \tau_3 = (5, 6, 6)$



Feasible for τ_3.



Infeasible for τ_3.

Outline

Introduction

Schedulability Analysis: Baker's Approach

Schedulability Analysis: Improved Results

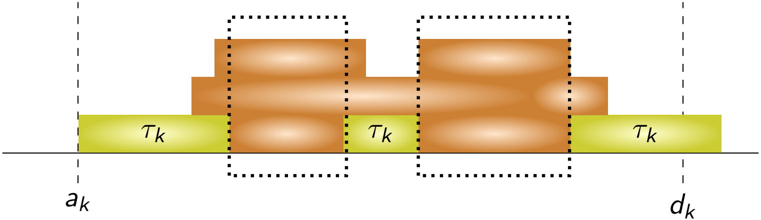
Schedulability Analysis: Augmentation Factor

Appendix

Strategy

- We are looking for the necessary condition such that a deadline miss happens.
- Suppose that Global scheduling fails by missing the deadline d_k of task τ_k, which is the first instant with deadline missing.
- The job with the earliest deadline miss arrives at time a_k, in which $D_k = d_k - a_k$.

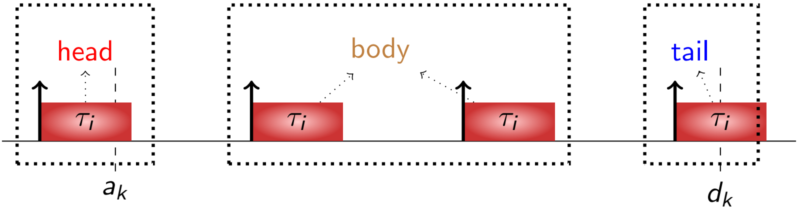
First Necessary Condition



- If τ_k misses the deadline at d_k, there must be at least $D_k - C_k$ units of time in which all M processors are executing other higher-priority jobs.
- Definition: *demand* in a time interval with length Δ is the total amount of computation that needs to be completed within the interval.
- Definition: *load* $W(\Delta)$ is the demand with interval length Δ divided by Δ.
- If τ_k misses its deadline at time d_k, then

$$W(D_k) > M(1 - \frac{C_k}{D_k}) + \frac{C_k}{D_k}.$$

Identifying Interference



- Problem window (interval) is defined in $[a_k, d_k)$.
- The jobs of task τ_i in the problem window can be categorized into three types:
  - Head job (at most one): some computation demand is *carried in* to the problem window for a job arrival before a_k.
  - Body jobs: the computation demand has to be done in the problem window.
  - Tail job (at most one): some computation demand can be *carried out* from the problem window.

Bound Carry-In Interference

- Baker's approach tries to bound the carry-in interference by extending the busy-interval to the left hand side while satisfying some load condition.
- This step is called *downward extension of an interval* for global EDF. For your reference, the procedures are included in the appendix.
- Here, I am presenting a very simple strategy to analyze the schedulability for global RM.
- This is based on the schedulability analysis we did earlier in the utilization bound analysis for global RM.

A Pessimistic Sufficient Test for Global RM

For all $0 < t \leq T_k$

$$W_k(t) = \sum_{i=1}^{k-1} \left(\left\lceil \frac{t}{T_i} \right\rceil - 1 \right) C_i + 2C_i.$$

This implies that we just greedily take a head job immediately.
Clearly, lower-priority jobs have no effect for the unschedulability or schedulability.

Theorem

A system $\mathcal{T}$ of periodic, independent, preemptable tasks is schedulable by Global-RM on M processors if

$$\forall \tau_i \in \mathcal{T} \ \exists t \text{ with } 0 < t \leq D_i \ \ \textit{and} \ \ C_i + \frac{W_i(t)}{M} \leq t$$

holds. This condition is NOT a necessary condition.

Worst-Case: $T_k \leq 2T_1$

By the test, we know that for all $0 < t \leq T_k$

$$W_k(t) = \sum_{i=1}^{k-1}(\left\lceil \frac{t}{T_i} \right\rceil - 1)C_i + 2C_i > t.$$

Now, suppose again F_i is $\left\lfloor \frac{T_k}{T_i} \right\rfloor$. For all $0 < t \leq T_k$

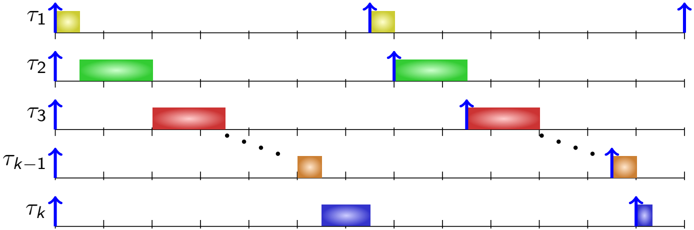
$$\left\lceil \frac{t}{T_i} \right\rceil C_i \leq \left\lceil \frac{t}{F_i T_i} \right\rceil F_i C_i$$

Therefore, by changing the period of task τ_i to $F_i T_i$ and the execution time from C_i to $F_i C_i$, task τ_k in the new task set remains unschedulable under Global RM.

After changing the periods, we reorder the tasks according to their new periods. Does this affect the non-schedulability of task τ_k?

Hyperbolic Bound: Structure

For the rest of the proof, we only consider $T_k \leq 2T_1$. The non-schedulability also implies the following structure:



$$C_k + \frac{\sum_{j=1}^{k-1} 2C_j + \sum_{j=0}^{i-1} C_j}{M} > T_i, \forall i = 1, 2, \ldots, k-1,$$

$$C_k + \frac{3\sum_{j=1}^{n-1} C_j}{M} > T_k,$$

where C_0 is defined as 0 for brevity.

Hyperbolic Bound: Structure

That is, C_k must be *sufficiently large* to enforce the above conditions.

Let's now recall what we were doing:

- We were given a set of tasks, in which the utilization U_i of each task τ_i is given.
- We wanted to prove that the task set is always schedulable under Global RM no matter how the periods are assigned under certain utilization constraints.
- What we have done so far is the contraposition that when the utilization above certain conditions, there exists at least one assignment of periods to make task τ_k not schedulable under Global RM.

Hyperbolic Bound: U_k

For a critical value of U_k, if we reduce U_k by a small value, the above non-schedulability condition will not be satisfied any more. So, the critical value is equivalent to the minimum U_k to enforce the following condition:

$$C_k + \frac{2\sum_{j=1}^{k-1} C_j + \sum_{j=0}^{i-1} C_j}{M} \geq T_i, \forall i = 1, 2, \ldots, k-1,$$

$$C_k + \frac{3\sum_{j=1}^{k-1} C_j}{M} = T_k,$$

In fact, we can also normalize T_k to 1. The above condition to get the minimum U_k is equivalent to the following linear programming

$$\text{minimize } C_k = T_k - \frac{3\sum_{j=1}^{k-1} U_j T_j}{M}$$

$$\text{s.t.} T_k - \frac{3\sum_{j=1}^{k-1} U_j T_j}{M} + \frac{2\sum_{j=1}^{k-1} U_j T_j + \sum_{j=0}^{i-1} U_j T_j}{M} \geq T_i, \forall i = 1, \ldots, k -$$

Extreme Point Theory in Linear Programming

$$\text{minimize } C_k = T_k - \frac{3\sum_{j=1}^{k-1} U_j T_j}{M}$$
$$\text{s.t. } T_i = T_k - \frac{\sum_{j=i}^{k-1} U_j T_j}{M} \geq T_i, \forall i = 1, \ldots, k-1$$

The optimal solution of the above linear programming is achieved when all the $k-1$ linear constraints are with $=$ instead of $\geq$ by the extreme point theory. That is, the minimum U_k is achieved when

$$T_i = T_k - \frac{\sum_{j=i}^{k-1} U_j T_j}{M}, \forall i = 1, \ldots, k-1$$
$$C_i = (T_{i+1} - T_i)M, \forall i = 1, \ldots, k-1$$

Hyperbolic Bound: Final

$$U_i = \frac{C_i}{T_i} = M\frac{T_{i+1} - T_i}{T_i} = M(\frac{T_{i+1}}{T_i} - 1), \forall i = 1, \ldots, n-1$$

$$\Rightarrow \frac{T_i}{T_{i+1}} = U_i/M + 1, \forall i = 1, \ldots, n-1$$

$$C_k^* = T_k - 3(T_k - T_1) = 3T_1 - 2T_k$$

$$\Rightarrow U_k^* = 3\left(\frac{T_1}{T_k}\right) - 2,$$

where C_k^* is the optimal solution of the above linear programming. The non-schedulability of task τ_k implies that

$$U_k > U_k^* = 3\left(\frac{T_1}{T_2}\frac{T_2}{T_3}\cdots\frac{T_{k-1}}{T_k}\right) - 2$$

$$= \frac{3}{\prod_{i=1}^{k-1}(U_i/M+1)} - 2$$

Hyperbolic Bound: Final

The task set is schedulable under Global RM if

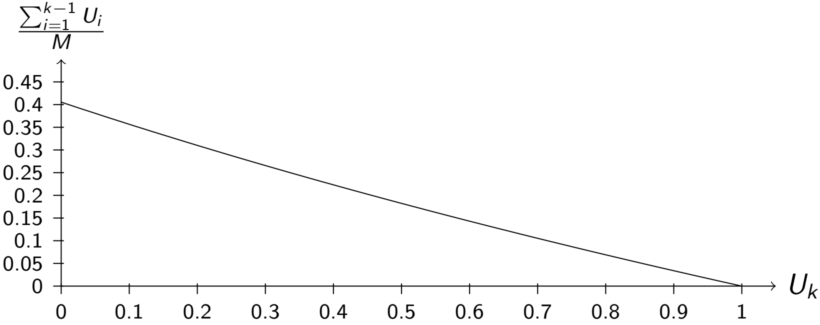
$$\forall k, \qquad (2 + U_k)\,\Pi_{i=1}^{k-1}(U_i/M + 1) \leq 3.$$

Hyperbolic Bound: Final

The task set is schedulable under Global RM if

$$\forall k, \qquad (2 + U_k) \Pi_{i=1}^{k-1} (U_i / M + 1) \leq 3.$$

The following figure is the hyperbolic bound for the extreme case when k goes to ∞, in which $(2 + U_k) e^{\frac{\sum_{i=1}^{k-1} U_i}{M}} \leq 3$



Capacity Augmentation Bound

Given a task set $\mathcal{T}$ with total utilization of $U_{\sum}$, a scheduling algorithm $\mathcal{A}$ with capacity augmentation bound b can always schedule this task set on M processors of speed b as long as $\mathcal{T}$ satisfies the following conditions:

$$\text{Utilization does not exceed total cores, } \sum_{\tau_i \in \mathcal{T}} U_i \leq M \tag{1}$$

$$\text{For each task } \tau_i \in \mathcal{T}, \text{the critical path } U_i \leq 1 \tag{2}$$

Capacity Augmentation Bound

Given a task set $\mathcal{T}$ with total utilization of $U_{\sum}$, a scheduling algorithm $\mathcal{A}$ with capacity augmentation bound b can always schedule this task set on M processors of speed b as long as $\mathcal{T}$ satisfies the following conditions:

$$\text{Utilization does not exceed total cores, } \sum_{\tau_i \in \mathcal{T}} U_i \leq M \tag{1}$$

$$\text{For each task } \tau_i \in \mathcal{T}, \text{the critical path } U_i \leq 1 \tag{2}$$

This means that the algorithm guarantees the schedulability if the following conditions are satisfied:

$$\text{Utilization does not exceed total cores, } \sum_{\tau_i \in \mathcal{T}} U_i \leq \frac{M}{b} \tag{3}$$

$$\text{For each task } \tau_i \in \mathcal{T}, \text{the critical path } U_i \leq \frac{1}{b} \tag{4}$$

Capacity Augmentation Bound of Global RM

The task set is schedulable under Global RM if

$$\forall k, (2 + U_k) \Pi_{i=1}^{k-1} (U_i/M + 1) \leq 3. \tag{5}$$

$$\Rightarrow \left(2 + \frac{1}{b}\right) \left(\frac{1}{(k-1)b} + 1\right)^{k-1} \leq 3. \tag{6}$$

$$\Rightarrow \left(2 + \frac{1}{b}\right) e^{1/b} \leq 3. \tag{7}$$

Again, we use the worst cases by setting all the tasks with the same utilization as we did in the analysis for uniprocessor systems. This concludes that $b \geq 3.6215$ enforces the above inequality.

Remarks

The bounds provided here are looser than the best results. These better results use more precise ways to analyze the workload of the head jobs. Here, we just greedily take them.

Outline

Introduction

Schedulability Analysis: Baker's Approach

Schedulability Analysis: Improved Results

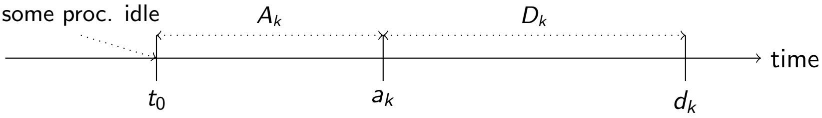
Schedulability Analysis: Augmentation Factor

Appendix

Baruah's Approach

Again, assume that τ_k misses its deadline at time d_k with release time a_k.

- Bound the carry-in computation demand more precisely.
- Let t_0 be the earliest time instant such that the system executes jobs on M processors from t_0 to a_k.
  - The ready queue before t_0 is with less than M jobs.
  - The ready queue has at least M jobs in time interval $[t_0, a_k)$.
- Let $\mathcal{I}$ be the set of intervals in $[t_0, d_k)$, in which all the M processors are executing. By considering the worst cases, the job of task τ_k arriving at time a_k is not executed at all in $\mathcal{I}$.
- Let A_k be $a_k - t_0$.



Necessary Condition

- Let $W_i(\mathcal{I})$ be the demand executed in the set $\mathcal{I}$ of time intervals. The necessary condition for τ_k to miss its deadline is

$$\sum_{i=1}^{N} W_i(\mathcal{I}) > M(A_k + D_k - C_k).$$

- Let's consider two types of interferences in $W_i(\mathcal{I})$.
  - Type 1: tasks that are not executing at time t_0. There will be no carry-in demand at time t_0.
  - Type 2: tasks that are executing at time t_0. There might be carry-in demand at time t_0.

Interference Type 1: No Carry-In at Time t_0

- Case 1: $i \neq k$
  - The demand of τ_i to be done in the time intervals in $\mathcal{I}$ is at most
    $$\min\left\{dbf(\tau_i, A_k + D_k), A_k + D_k - C_k\right\}.$$
- Case 2: i is k
  - The demand of τ_k to be done in the time intervals in $\mathcal{I}$ is at most
    $$\min\left\{dbf(\tau_i, A_k + D_k) - C_k, A_k\right\}.$$
  - Specifically, we need to remove the job that arrives at a_k since its execution is not counted as part of $\mathcal{I}$.

Therefore,

$$W_i^1(\mathcal{I}) =^{\text{def}} \begin{cases} \min\left\{dbf(\tau_i, A_k + D_k), A_k + D_k - C_k\right\} & \text{if } i \neq k \\ \min\left\{dbf(\tau_i, A_k + D_k) - C_k, A_k\right\} & \text{if } i = k. \end{cases}$$

Interference Type 2: With Carry-In at Time t_0

- Case 1: $i \neq k$
  - The demand of τ_i to be done in the time intervals in $\mathcal{I}$ is at most
$$\min\left\{dbf^{\dagger}(\tau_i, A_k + D_k), A_k + D_k - C_k\right\},$$
where $dbf^{\dagger}(\tau_i, \delta)$ is $\left\lfloor \frac{\delta}{T_i} \right\rfloor C_i + \min\{C_i, \delta \mod T_i\}$.
- Case 2: i is k
  - The demand of τ_k to be done in the time intervals in $\mathcal{I}$ is at most
$$\min\left\{dbf^{\dagger}(\tau_i, A_k + D_k) - C_k, A_k\right\}.$$

Therefore,

$$W_i^2(\mathcal{I}) =^{\text{def}} \begin{cases} \min\left\{dbf^{\dagger}(\tau_i, A_k + D_k), A_k + D_k - C_k\right\} & \text{if } i \neq k \\ \min\left\{dbf^{\dagger}(\tau_i, A_k + D_k) - C_k, A_k\right\} & \text{if } i = k. \end{cases}$$

Putting Together

- Let $W_i^{diff}(\mathcal{I})$ be $W_i^2(\mathcal{I}) - W_i^1(\mathcal{I})$.
- The necessary condition for τ_k to miss its deadline becomes

$$\sum_{i=1}^{N} W_i^1(\mathcal{I}) + \sum_{M-1 \text{ largest}} W_i^{diff}(\mathcal{I}) > M(A_k + D_k - C_k).$$

Putting Together

- Let $W_i^{diff}(\mathcal{I})$ be $W_i^2(\mathcal{I}) - W_i^1(\mathcal{I})$.
- The necessary condition for τ_k to miss its deadline becomes

$$\sum_{i=1}^{N} W_i^1(\mathcal{I}) + \sum_{M-1 \text{ largest}} W_i^{diff}(\mathcal{I}) > M(A_k + D_k - C_k).$$

Theorem

A task set is schedulable under Global-EDF if for every task τ_k and for all $A_k \geq 0$

$$\sum_{i=1}^{N} W_i^1(\mathcal{I}) + \sum_{M-1 \text{ largest}} W_i^{diff}(\mathcal{I}) \leq M(A_k + D_k - C_k).$$

Complexity

Theorem

If there exists $A_k \geq 0$ with

$$\sum_{i=1}^{N} W_i^1(\mathcal{I}) + \sum_{M-1 \text{ largest}} W_i^{diff}(\mathcal{I}) > M(A_k + D_k - C_k),$$

then

$$A_k \leq \frac{SumMaxC - D_k(M - \sum_{i=1}^{N} \frac{C_i}{T_i}) + \sum_{i=1}^{N}(T_i - D_i)\frac{C_i}{T_i} + MC_k}{M - \sum_{i=1}^{N} \frac{C_i}{T_i}},$$

where $SumMaxC$ is the sum of the $(M-1)$ largest C_i's.

Outline

Introduction

Schedulability Analysis: Baker's Approach

Schedulability Analysis: Improved Results

Schedulability Analysis: Augmentation Factor

Appendix

Normal Collection of Jobs

A job collection $\mathcal{J}$ is a set of jobs that are revealed online over time:

- a job $j \in \mathcal{J}$ becomes known upon the release date of j
- Each job $j \in \mathcal{J}$ is characterized by its arrival time r_j, absolute deadline d_j, and an unknown execution time c_j.

Note that the actual execution time c_j of a job is discovered by the scheduler only after the job signals completion.

Optimal Schedule for $\mathcal{J}$

Given $\mathcal{J}$, suppose that infinitely many (or, say, $|\mathcal{J}|$) processors of unit speed were available.

Optimal Schedule for $\mathcal{J}$

Given $\mathcal{J}$, suppose that infinitely many (or, say, $|\mathcal{J}|$) processors of unit speed were available.

Then, the following scheduling algorithm S_∞ is optimal:

- just allocate one processor to each job and schedule each job as early as possible.

Schedulability for EDF

Theorem

Consider a normal collection $\mathcal{J}$ of jobs and let $\alpha \geq 1$. Then at least one of the following conditions holds:

1. all jobs in $\mathcal{J}$ are completed within their deadline under EDF on M processors of speed α, or
2. $\mathcal{J}$ is infeasible under S_∞, or
3. there is an interval I such that any feasible schedule for $\mathcal{J}$ must finish more than $(\alpha M - M + 1) \cdot |I|$ units of work within I.

Proof

- The details are omitted, please refer to Bonifaci et al. in ECRTS 2013 (Lemma 3 in Page 228).

Speedup for Normal Collection of Jobs

Theorem

Any normal collection of jobs that is feasible on M processors of unit speed is EDF-schedulable on M processors of speed $2 - 1/M$.

Proof

The feasibility on M processors of unit speed implies that the demand at any interval I is at most $M \cdot |I|$. By setting α to $2 - \frac{1}{M}$, for any interval I, we have

$$(\alpha M - M + 1) \cdot |I| = M \cdot |I|.$$

Hence, this implies that EDF finishes all jobs by their respective deadline at speed $2 - \frac{1}{M}$.

Putting Together

Theorem

If $\forall t > 0$, we have $dbf(\tau_i, t) \leq t$ for every task τ_i and $\sum_{i=1}^{N} dbf(\tau_i, t) \leq M \cdot t$, then this task set with N tasks is EDF-schedulable on M processors of speed $2 - 1/M$.

Theorem

If $\forall t > 0$, we have $dbf(\tau_i, t) \leq \frac{t}{2-1/M}$ for every task τ_i and $\sum_{i=1}^{N} dbf(\tau_i, t) \leq M \cdot \frac{t}{2-1/M}$, then this task set with N tasks is EDF-schedulable on M processors.

This analysis also works for arbitrary deadlines.

Outline

Introduction

Schedulability Analysis: Baker's Approach

Schedulability Analysis: Improved Results

Schedulability Analysis: Augmentation Factor

Appendix

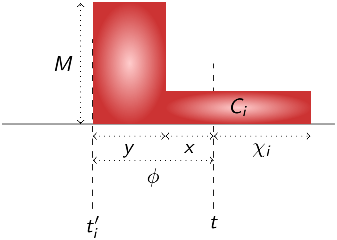
Bound Carry-In Interference

Let t_i' be the last release time of task τ_i before $t \stackrel{\text{def}}{\equiv} a_k$ and ϕ be $t - t_i'$. Suppose that y is the sum of the lengths of all the intervals in $[t_i', t)$, where all M processors are executing jobs that preempt τ_i, then

- If the carry-in χ_i of task τ_i is non-zero, we have

$$\chi_i = C_i - (\phi - y).$$

- The load in interval $[t_i', t)$ is at least $(M-1)\frac{(\phi - C_i + \chi_i)}{\phi} + 1$.



$$y = \phi - (C_i - \chi_i).$$

$$\frac{My + (\phi - y)}{\phi} = (M-1)\frac{y}{\phi} + 1.$$

Busy Intervals

- An interval is said λ-busy if its load is at least $M(1-\lambda)+\lambda$.
- A downward extension of an interval is to expand the starting point of the interval with the original ending point.
- A maximum λ-busy downward extension is the maximum one that can not be extended further to maintain the λ-busy property.
- Example:
  - $[a_k, a_k + D_k)$ is a problem window and is a $\frac{C_k}{D_k}$-busy interval.
- It can be proved that any problem interval for task τ_k has a unique maximal λ-busy downward extension for $\lambda = \frac{C_k}{D_k}$.
  - Let $[t, t+\Delta)$ be the maximal λ-busy downward extension (Note that, therefore, $\Delta \geq D_k$).
  - *t is no longer the arrival time, but $t+\Delta$ is still the deadline.*
- Suppose that the head job of task τ_i $(i \neq k)$ arrives at time $t-\phi$. If $\phi \geq D_i$, there is no carry-in computation of τ_i; otherwise, the carry-in at time t is at most $C_i - \lambda\phi$.

Body and Tail Jobs

- Let n_i be the maximum number of jobs of task τ_i released as the body and tail jobs (that must be completed) in time interval $[t, t+\Delta)$.
- Suppose that $t_i^\dagger$ is the release time of the tail job.
- Let σ_i be $t + \Delta - t_i^\dagger$.

Theorem

Let $n_i = \left\lfloor \frac{\Delta - D_i}{T_i} \right\rfloor + 1$. The EDF demand W_i in the busy window $[t, t+\Delta)$ (the maximal λ-busy downward extension) resulting from τ_i is at most

$$n_i C_i + \max\{0, C_i - \phi\lambda\}.$$

Upper Bound on EDF Load

Theorem

The EDF load $W_i(\Delta)$ in the busy window $[t, t+\Delta)$ (the maximal λ-busy downward extension) for any $\Delta \geq D_k$ is at most

$$\beta_i = \begin{cases} \frac{C_i}{T_i}(1 + \frac{T_i - D_i}{D_k}) & \text{if } \lambda \geq \frac{C_i}{T_i} \\ \frac{C_i}{T_i}(1 + \frac{T_i - D_i}{D_k}) + \frac{C_i - \lambda T_i}{D_k} & \text{if } \lambda < \frac{C_i}{T_i} \end{cases}$$

Proof

- Let $f_i(\Delta)$ be $\frac{n_i C_i + \max\{0, C_i - \phi\lambda\}}{\Delta}$.
- We have known that $W_i(\Delta) \leq f_i(\Delta)$, and would like to prove $f_i(\Delta) \leq \beta_i$

Proof Case 1: $\max\{0, C_i - \phi\lambda\}$ Is 0

- $C_i - \phi\lambda \leq 0 \Rightarrow \lambda \geq \frac{C_i}{\phi}$.
- By definition $\phi < T_i$, and we know $\lambda \geq \frac{C_i}{T_i}$.
- Therefore, we have

$$
\begin{aligned}
f_i(\Delta) = \frac{n_i C_i}{\Delta} &= \frac{\left(\left\lfloor \frac{\Delta - D_i}{T_i} \right\rfloor + 1\right) C_i}{\Delta} \leq \frac{\frac{\Delta - D_i + T_i}{T_i} C_i}{\Delta} \\
&= \frac{C_i}{T_i}\left(1 + \frac{T_i - D_i}{\Delta}\right) \leq \frac{C_i}{T_i}\left(1 + \frac{T_i - D_i}{D_k}\right),
\end{aligned}
$$

where the last inequality comes from the fact $\Delta \geq D_k$.

Proof Case 2: $\max\{0, C_i - \phi\lambda\} > 0$

- $C_i - \phi\lambda > 0$.
- By taking $\phi + \Delta = n_i T_i + D_i$ (i.e., σ_i is D_i) to re-formulate $f_i(\Delta)$, $f_i(\Delta) = \frac{n_i(C_i - \lambda T_i) + C_i - \lambda(D_i - \Delta)}{\Delta}$.
- If $C_i - \lambda T_i > 0$, i.e., $\lambda < \frac{C_i}{T_i}$, we know that ($n_i \leq \frac{\Delta - D_i}{T_i} + 1$)

$$
\begin{aligned}
f_i(\Delta) &\leq \frac{\frac{\Delta - D_i + T_i}{T_i}(C_i - \lambda T_i) + C_i - \lambda(D_i - \Delta)}{\Delta} \\
&\leq \frac{C_i}{T_i}(1 + \frac{T_i - D_i}{\Delta}) + \frac{C_i - \lambda T_i}{\Delta} \\
&\leq \frac{C_i}{T_i}(1 + \frac{T_i - D_i}{D_k}) + \frac{C_i - \lambda T_i}{D_k}
\end{aligned}
$$

- If $C_i - \lambda T_i \leq 0$, i.e., $\lambda \geq \frac{C_i}{T_i}$, we know that ($n_i > \frac{\Delta - D_i}{T_i}$)

$$
f_i(\Delta) < \frac{\frac{\Delta - D_i}{T_i}(C_i - \lambda T_i) + C_i - \lambda(D_i - \Delta)}{\Delta} \leq \frac{C_i}{T_i}(1 + \frac{T_i - D_i}{D_k})
$$

Putting Everything Together

Theorem

A set of periodic tasks $\tau_1, \tau_2, \ldots, \tau_N$ with constrained deadlines is schedulable on M processors by using preemptive Global EDF scheduling if for every task τ_k

$$\sum_{i=1}^{N} \min\{1, \beta_i\} \leq M(1 - \frac{C_k}{D_k}) + \frac{C_k}{D_k}.$$

Theorem

A set of periodic tasks $\tau_1, \tau_2, \ldots, \tau_N$ with implicit deadlines is schedulable on M processors by using preemptive Global EDF scheduling if

$$\sum_{i=1}^{N} \frac{C_i}{T_i} \leq M(1 - \frac{C_k}{T_k}) + \frac{C_k}{T_k},$$

where τ_k is the task with the largest utilization $\frac{C_k}{T_k}$

Upper Bound on DM (Deadline Monotonic) Load

Theorem

The DM load $W_i(\Delta)$ in the busy window $[t, t+\Delta)$ (the maximal λ-busy downward extension) is at most

$$\beta_i = \begin{cases} \frac{C_i}{T_i}(1 + \frac{T_i - \delta_i}{D_k}) & \text{if } \lambda \geq \frac{C_i}{T_i} \\ \frac{C_i}{T_i}(1 + \frac{T_i - \delta_i}{D_k}) + \frac{C_i - \lambda T_i}{D_k} & \text{if } \lambda < \frac{C_i}{T_i} \end{cases},$$

where δ_i is C_i for $i < k$ and δ_k is D_k

Theorem

A set of periodic tasks $\tau_1, \tau_2, \ldots, \tau_N$ with constrained deadlines is schedulable on M processors by using preemptive DM scheduling if for every task τ_k

$$\sum_{i=1}^{k-1} \beta_i \leq M(1 - \frac{C_k}{D_k}).$$