Push Forward: Global Fixed-Priority Scheduling of Arbitrary-Deadline Sporadic Task Systems

Jian-Jia Chen, Georg von der Brüggen and Niklas Ueter

TU Dortmund University, Germany

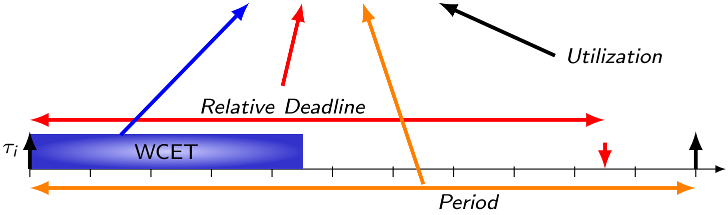
06.07.2018 at ECRTS

technische universität dortmund

SFB 876 Verfügbarkeit von Information durch Analyse unter Ressourcenbeschränkung

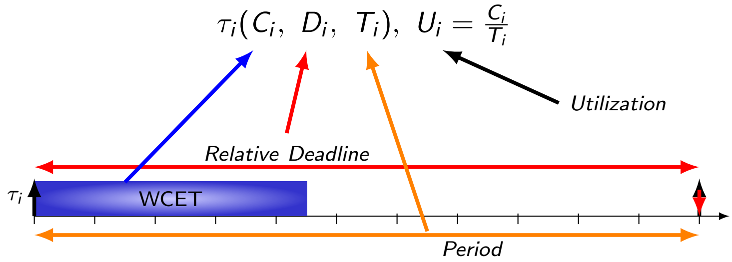
Sporadic Task Model

$$\tau_i(C_i,\ D_i,\ T_i),\ U_i = \frac{C_i}{T_i}$$



Constrained `Deadline`: $D_i \leq T_i, \forall \tau_i$

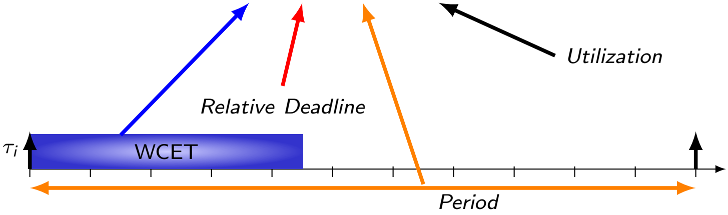
Sporadic Task Model



Constrained `Deadline`: $D_i \leq T_i, \forall\tau_i$
Implicit `Deadline`: $D_i = T_i, \forall\tau_i$

Sporadic Task Model

$$\tau_i(C_i,\ D_i,\ T_i),\ U_i = \frac{C_i}{T_i}$$



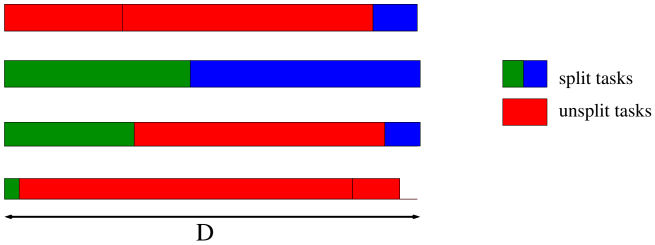
Constrained `Deadline`: $D_i \leq T_i, \forall \tau_i$
Implicit `Deadline`: $D_i = T_i, \forall \tau_i$
Arbitrary `Deadline`: otherwise. The jobs of a task must be executed in the FCFS manner.

Scheduling Model

- M identical multiprocessors: all the processors have the same characteristics
- Global scheduling:
  - A job may execute on any processor
  - The system maintains a global ready queue
  - Execute the M highest-priority jobs in the ready queue
- Basic Terms: EDF, FP, RM, DM
- Global Scheduling: Global-EDF, Global-FP, Global-RM, Global-DM

Good News for Global Scheduling

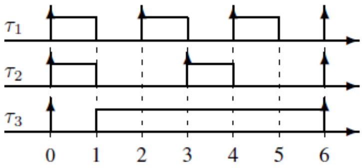
For frame-based task systems, McNaughton's wrap-around rule for $P|pmtn|C_{\max}$ is optimal



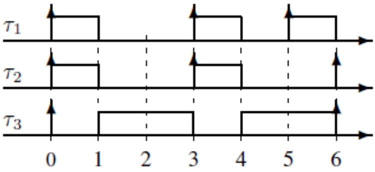
R. McNaughton. Scheduling with deadlines and loss functions. Management Science, 6:1-12, 1959.

Critical Instant?

- Synchronous release of higher-priority tasks and as early as possible for the following jobs
- Example: M=2 and 3 tasks: (C_i, D_i, T_i) are $\tau_1 = (1, 2, 2), \tau_2 = (1, 3, 3), \tau_3 = (5, 6, 6)$



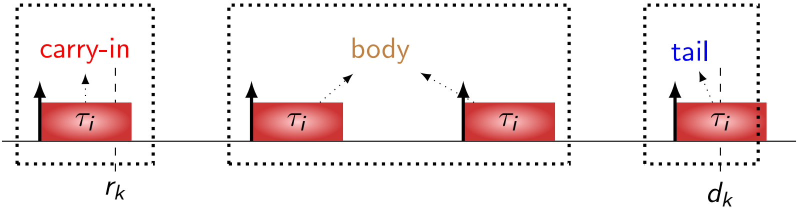
Feasible for τ_3.



Infeasible for τ_3.

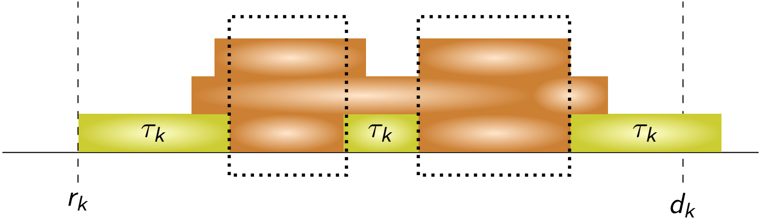
Baruah in RTSS 2007.

Identifying Interference for Constrained Deadlines



- For contrapositive, assume that a job of task τ_k misses its absolute deadline at time d_k with release time r_k
- Problem window (interval) is defined in $[r_k, d_k)$

Necessary Condition for Deadline Misses

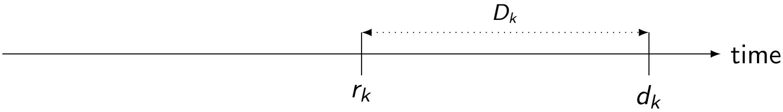


- *demand* $E(\Delta)$: high-priority computation *executed* in the interval of length Δ.
- If τ_k misses its deadline at time d_k, then

$$E(D_k) > M \times (D_k - C_k) + C_k$$

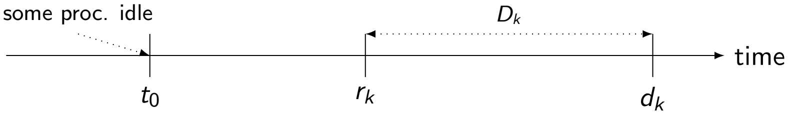
Carry-In Demand

- For contrapositive, assume that a job of task τ_k misses its absolute deadline at time d_k with release time r_k.



Carry-In Demand

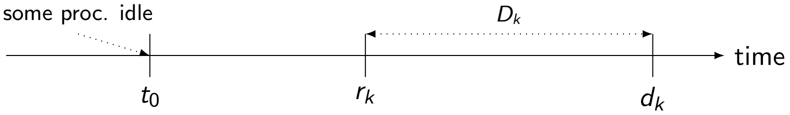
- For contrapositive, assume that a job of task τ_k misses its absolute deadline at time d_k with release time r_k.



- An innovative idea by Baruah in RTSS 2007 for Global-EDF, extended by Guan et al. in RTSS 2009 for Global-FP
- Let t_0 be the earliest time instant such that the system executes jobs on M processors from t_0 to r_k.
  - Prior to t_0, at least one processor idles

Carry-In Demand

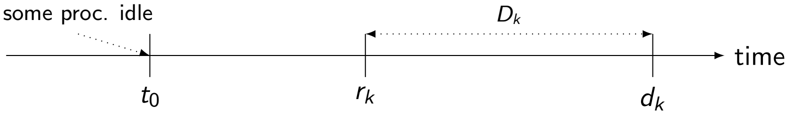
- For contrapositive, assume that a job of task τ_k misses its absolute deadline at time d_k with release time r_k.



- An innovative idea by Baruah in RTSS 2007 for Global-EDF, extended by Guan et al. in RTSS 2009 for Global-FP
- Let t_0 be the earliest time instant such that the system executes jobs on M processors from t_0 to r_k.
  - Prior to t_0, at least one processor idles
  - For constrained-deadline systems, at most M-1 carry-in jobs

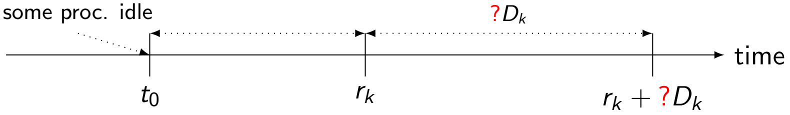
Carry-In Demand

- For contrapositive, assume that a job of task τ_k misses its absolute deadline at time d_k with release time r_k.



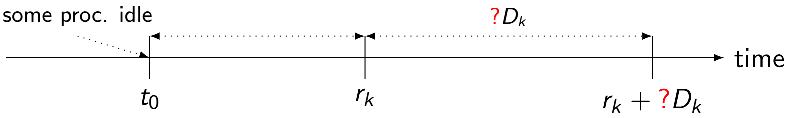
- An innovative idea by Baruah in RTSS 2007 for Global-EDF, extended by Guan et al. in RTSS 2009 for Global-FP
- Let t_0 be the earliest time instant such that the system executes jobs on M processors from t_0 to r_k.
  - Prior to t_0, at least one processor idles
  - For constrained-deadline systems, at most M-1 carry-in jobs
  - For arbitrary-deadline systems, at most M-1 carry-in tasks and there may be multiple jobs in a carry-in task

Global-FP: Arbitrary-Deadline



- Essential Problems
  - How to define the window of interest?
  - How many jobs should be considered in the window of interest?

Global-FP: Arbitrary-Deadline



- Essential Problems
  - How to define the window of interest?
  - How many jobs should be considered in the window of interest?
- Existing Results
  - Baker (RTSJ 2006): downward extension
  - Baruah and Fisher (OPODIS 2007, ICDCN 2008): to-be-detailed later
  - Guan et al. (RTSS 2009): M-1 carry-in jobs (flawed)
  - Sun et al. (RTCSA 2014): complex carry-in workload functions
  - Huang and Chen (RTNS 2015): a more precise quantification for the number of carry-in jobs

Resource Augmentation

Speedup (resource augmentation) factor ρ ($\rho \geq 1$):

$\Rightarrow$

If the task set (system) is schedulable (feasible), Algorithm A also returns a schedulable (feasible) answer when speeding up the system by a factor ρ.

Resource Augmentation

Speedup (resource augmentation) factor ρ ($\rho \geq 1$):

$\Rightarrow$

If the task set (system) is schedulable (feasible), Algorithm A also returns a schedulable (feasible) answer when speeding up the system by a factor ρ.

$\Leftarrow$

If Algorithm A does not return a schedulable (feasible) answer, the system is also unschedulable (infeasible) when slowing down by a factor ρ, i.e., at speed $1/\rho$.

Resource Augmentation of Global DM

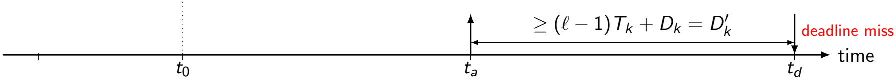
	implicit deadlines	constrained deadlines	arbitrary deadlines
upper bounds	2.668 (Lundberg 2002) 2.823 (Chen et al. 2016)	$3-1/M$ (Baruch et al. 2011) $3-1/M$ (Chen et al. 2016)	$\frac{2(M-1)}{4M-1-\sqrt{12M^2-8M+1}} \leq 3.73$ (Baruah and Fisher 2007)
lower bounds	2.668 (Lundberg 2002)	2.668 (Lundberg 2002)	2.668 (Lundberg 2002)

Resource Augmentation of Global DM

	implicit deadlines	constrained deadlines	arbitrary deadlines
upper bounds	2.668 (Lundberg 2002) 2.823 (Chen et al. 2016)	$3 - 1/M$ (Baruch et al. 2011) $3 - 1/M$ (Chen et al. 2016)	$\frac{2(M-1)}{4M-1-\sqrt{12M^2-8M+1}} \leq 3.73$ (Baruah and Fisher 2007) $3 - \frac{1}{M}$
lower bounds	2.668 (Lundberg 2002)	2.668 (Lundberg 2002)	2.668 (Lundberg 2002) $3 - \frac{3}{M+1}$

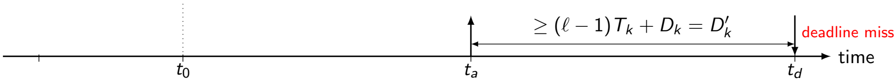
Outline

Push Forward (Basic Idea from Baruah and Fisher)



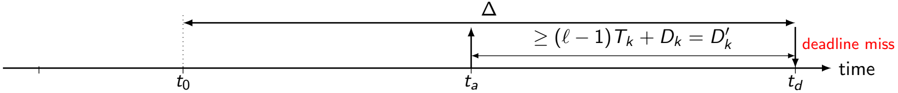
- τ_k is continuously active from t_a to t_d with deadline miss at t_d

Push Forward (Basic Idea from Baruah and Fisher)



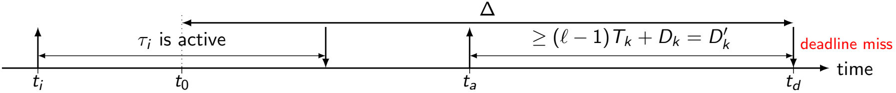
- τ_k is continuously active from t_a to t_d with deadline miss at t_d
- t_0 is the smallest value of $t \leq t_a$ such that $\Omega(t, t_d) \geq \mu_k$
  - $E(t, t_d)$: the amount of workload (sum of the execution times) of the higher-priority jobs, i.e., from $\tau_1, \tau_2, \ldots, \tau_{k-1}$, executed in the time interval $[t, t_d)$
  - C_k^*: amount of time that task τ_k is executed from t_a to t_d
  - ℓ-th job of task τ_k misses its deadline, i.e., $C_k^* < \ell C_k = C'_k$
  - $\Omega(t, t_d)$ is $\frac{C_k^* + E(t, t_d)}{t_d - t}$

Push Forward (Basic Idea from Baruah and Fisher)



- τ_k is continuously active from t_a to t_d with deadline miss at t_d
- t_0 is the smallest value of $t \leq t_a$ such that $\Omega(t, t_d) \geq \mu_k$
- Δ is $t_d - t_0$
- time instant t_i is the arrival time of a higher-priority carry-in task τ_i if τ_i is continuously active in time interval $[t_i, t_0 + \varepsilon]$, where $t_i < t_0$ and $\varepsilon > 0$ is an arbitrarily small number

Push Forward (Basic Idea from Baruah and Fisher)



- τ_k is continuously active from t_a to t_d with deadline miss at t_d
- t_0 is the smallest value of $t \leq t_a$ such that $\Omega(t, t_d) \geq \mu_k$
- Δ is $t_d - t_0$
- time instant t_i is the arrival time of a higher-priority carry-in task τ_i if τ_i is continuously active in time interval $[t_i, t_0 + \varepsilon]$, where $t_i < t_0$ and $\varepsilon > 0$ is an arbitrarily small number

Properties of Pushing Forward

Suppose that $\mu_k = M - (M-1)\rho$ for a certain ρ with $1 \geq \rho \geq \frac{C'_k}{D'_k}$.

Lemma

If τ_k misses its deadline at t_d, for any ρ with $1 \geq \rho \geq \frac{C'_k}{D'_k}$, the time t_0 always exists with $\Omega(t_0, t_d) \geq \mu_k$ and $t_0 \leq t_a$.

Properties of Pushing Forward

Suppose that $\mu_k = M - (M-1)\rho$ for a certain ρ with $1 \geq \rho \geq \frac{C'_k}{D'_k}$.

Lemma

If τ_k misses its deadline at t_d, for any ρ with $1 \geq \rho \geq \frac{C'_k}{D'_k}$, the time t_0 always exists with $\Omega(t_0, t_d) \geq \mu_k$ and $t_0 \leq t_a$.

Definition

A task τ_i is a carry-in task in the schedule S, if τ_i is continuously active in a time interval $[t_i, t_0 + \varepsilon]$, for $t_i < t_0$ and an arbitrarily small $\varepsilon > 0$.

Properties of Pushing Forward

Suppose that $\mu_k = M - (M-1)\rho$ for a certain ρ with $1 \geq \rho \geq \frac{C'_k}{D'_k}$.

Lemma

If τ_k misses its deadline at t_d, for any ρ with $1 \geq \rho \geq \frac{C'_k}{D'_k}$, the time t_0 always exists with $\Omega(t_0, t_d) \geq \mu_k$ and $t_0 \leq t_a$.

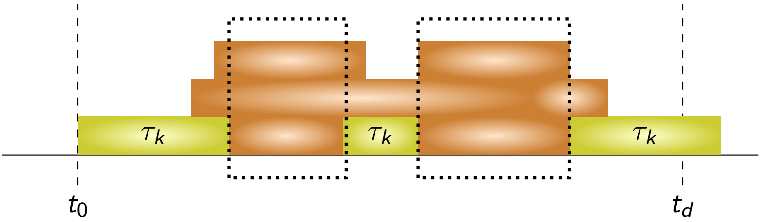
Definition

A task τ_i is a carry-in task in the schedule S, if τ_i is continuously active in a time interval $[t_i, t_0 + \varepsilon]$, for $t_i < t_0$ and an arbitrarily small $\varepsilon > 0$.

Lemma

For $1 \geq \rho \geq \frac{C'_k}{D'_k}$, there are at most $\lceil M - (M-1)\rho \rceil - 1$ carry-in tasks at t_0 in schedule S.

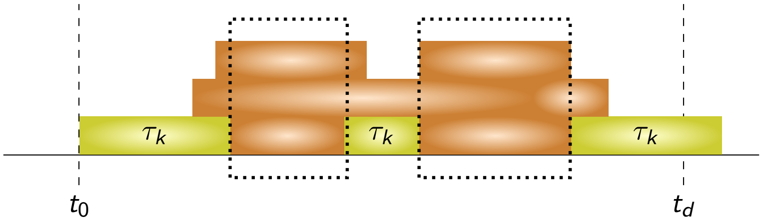
Necessary Condition for Deadline Misses



If task τ_k misses its deadline, then

- $\exists$ a positive integer ℓ, i.e., ℓ-th job of τ_k (in a continuously actively interval) misses its deadline

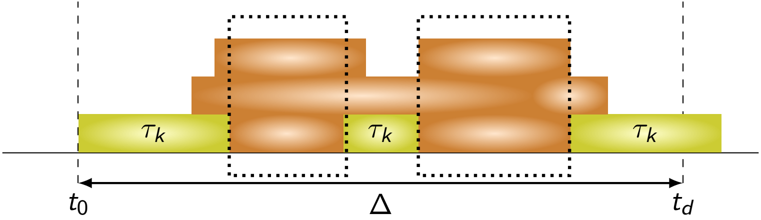
Necessary Condition for Deadline Misses



If task τ_k misses its deadline, then

- $\exists$ a positive integer ℓ, i.e., ℓ-th job of τ_k (in a continuously actively interval) misses its deadline
- $\forall 1 \geq \rho \geq \frac{C'_k}{D'_k}$,

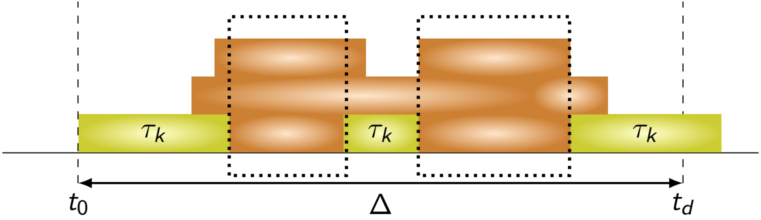
Necessary Condition for Deadline Misses



If task τ_k misses its deadline, then

- $\exists$ a positive integer ℓ, i.e., ℓ-th job of τ_k (in a continuously actively interval) misses its deadline
- $\forall 1 \geq \rho \geq \frac{C'_k}{D'_k}$, $\exists \Delta \geq D'_k = (\ell - 1) T_k + D_k$

Necessary Condition for Deadline Misses



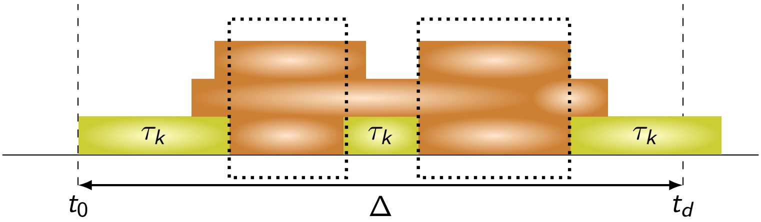
If task τ_k misses its deadline, then

- $\exists$ a positive integer ℓ, i.e., ℓ-th job of τ_k (in a continuously actively interval) misses its deadline
- $\forall 1 \geq \rho \geq \frac{C'_k}{D'_k}$, $\exists \Delta \geq D'_k = (\ell - 1)T_k + D_k$
- such that

$$\ell C_k + E(\Delta) > (M - (M-1)\rho)\Delta$$

$E(\Delta)$: high-priority computation *executed* in interval length Δ.

Sufficient Condition for τ_k



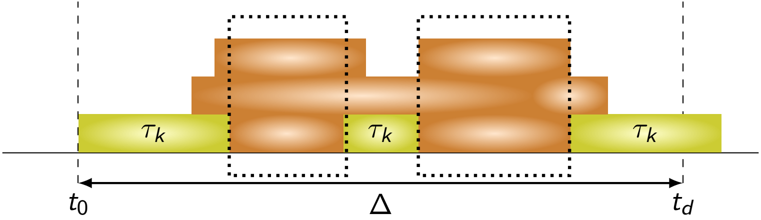
If task τ_k misses its deadline, then

- $\exists$ a positive integer ℓ, i.e., ℓ-th job of τ_k (in a continuously actively interval) misses its deadline
- $\forall 1 \geq \rho \geq \frac{C'_k}{D'_k}$, $\exists \Delta \geq D'_k = (\ell - 1)T_k + D_k$
- such that

$$\ell C_k + E(\Delta) > (M - (M-1)\rho)\Delta$$

$E(\Delta)$: high-priority computation executed in interval length Δ.

Sufficient Condition for τ_k



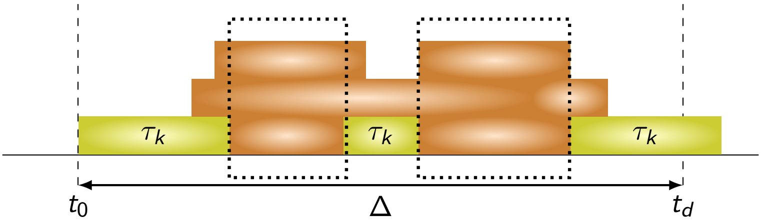
Task τ_k always meets its deadline, if

- $\exists$ a positive integer ℓ, i.e., ℓ-th job of τ_k (in a continuously actively interval) misses its deadline
- $\forall 1 \geq \rho \geq \frac{C'_k}{D'_k}$, $\exists \Delta \geq D'_k = (\ell - 1)T_k + D_k$
- such that

$$\ell C_k + E(\Delta) > (M - (M-1)\rho)\Delta$$

$E(\Delta)$: high-priority computation executed in interval length Δ.

Sufficient Condition for τ_k



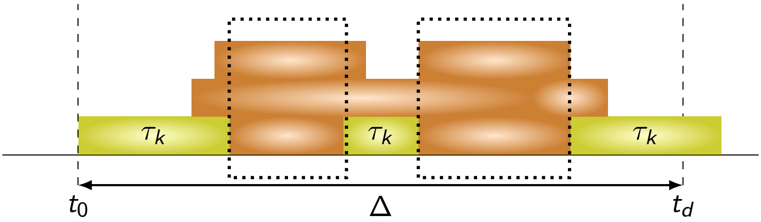
Task τ_k always meets its deadline, if

- $\forall$ positive integer ℓ, i.e., ℓ-th job of τ_k (in a continuously actively interval) misses its deadline
- $\forall 1 \geq \rho \geq \frac{C'_k}{D'_k}$, $\exists \Delta \geq D'_k = (\ell - 1)T_k + D_k$
- such that

$$\ell C_k + E(\Delta) > (M - (M-1)\rho)\Delta$$

$E(\Delta)$: high-priority computation executed in interval length Δ.

Sufficient Condition for τ_k



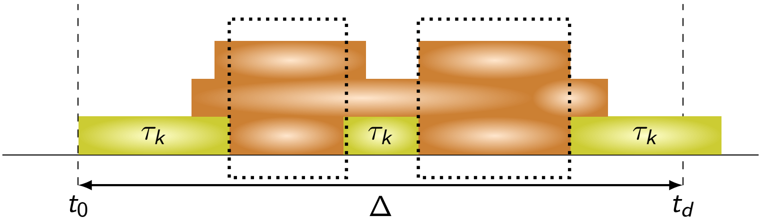
Task τ_k always meets its deadline, if

- $\forall$ positive integer ℓ, i.e., ℓ-th job of τ_k (in a continuously actively interval) misses its deadline
- $\exists 1 \geq \rho \geq \frac{C'_k}{D'_k}$, $\exists \Delta \geq D'_k = (\ell - 1)T_k + D_k$
- such that

$$\ell C_k + E(\Delta) > (M - (M-1)\rho)\Delta$$

$E(\Delta)$: high-priority computation executed in interval length Δ.

Sufficient Condition for τ_k



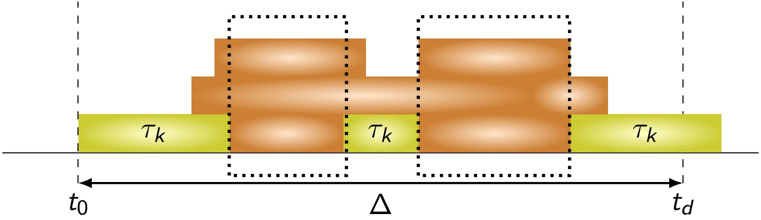
Task τ_k always meets its deadline, if

- $\forall$ positive integer ℓ, i.e., ℓ-th job of τ_k (in a continuously actively interval) misses its deadline
- $\exists 1 \geq \rho \geq \frac{C'_k}{D'_k}$, $\forall \Delta \geq D'_k = (\ell - 1)T_k + D_k$
- such that

$$\ell C_k + E(\Delta) > (M - (M-1)\rho)\Delta$$

$E(\Delta)$: high-priority computation executed in interval length Δ.

Sufficient Condition for τ_k



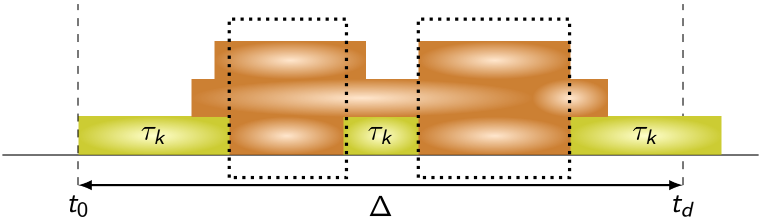
Task τ_k always meets its deadline, if

- $\forall$ positive integer ℓ, i.e., ℓ-th job of τ_k (in a continuously actively interval) misses its deadline
- $\exists 1 \geq \rho \geq \frac{C'_k}{D'_k}$, $\forall \Delta \geq D'_k = (\ell - 1)T_k + D_k$
- such that

$$\ell C_k + E(\Delta) \leq (M - (M-1)\rho)\Delta$$

$E(\Delta)$: high-priority computation executed in interval length Δ.

Baruah and Fisher's Analysis for Global-DM



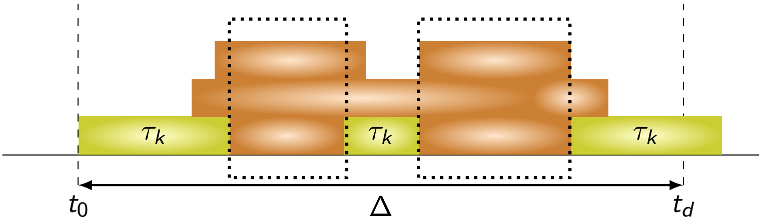
Task τ_k always meets its deadline, if

- $\forall$ positive integer ℓ, i.e., ℓ-th job of τ_k (in a continuously actively interval) misses its deadline
- $\exists 1 \geq \rho \geq \frac{C'_k}{D'_k}$, $\forall \Delta \geq D'_k = (\ell - 1)T_k + D_k$
- such that

$$\ell C_k + E(\Delta) \leq M - (M-1)\rho$$

$E(\Delta)$: high-priority computation executed in interval length Δ.

Baruah and Fisher's Analysis for Global-DM



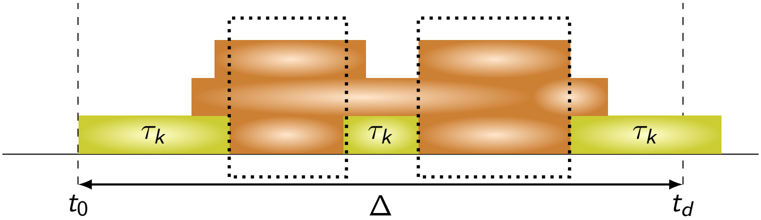
Task τ_k always meets its deadline, if

- $\forall$ positive integer ℓ, i.e., ℓ-th job of τ_k (in a continuously actively interval) misses its deadline
- for $\rho = \max_i^k\{\frac{C_i}{T_i}, \frac{C_i}{D_i}\}$, $\forall \Delta \geq D'_k = (\ell - 1)T_k + D_k$
- such that

$$\ell C_k + E(\Delta) \leq M - (M-1)\rho$$

$E(\Delta)$: high-priority computation executed in interval length Δ.

Baruah and Fisher's Analysis for Global-DM

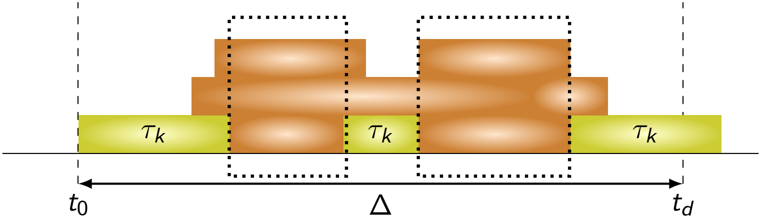


Task τ_k always meets its deadline, if

- $\forall$ positive integer ℓ, i.e., ℓ-th job of τ_k (in a continuously actively interval) misses its deadline
- for $\rho = \max_i^k\{\frac{C_i}{T_i}, \frac{C_i}{D_i}\}$, $\forall \Delta \geq D'_k = (\ell - 1)T_k + D_k$
- such that

$$\ell C_k + (\lceil M - (M-1)\rho \rceil - 1)\rho\Delta + \left(\sum_{i=1}^{k-1} dbf_i(\Delta + D_i)\right) \leq (M-(M-1)\rho)\Delta$$

Baruah and Fisher's Analysis for Global-DM



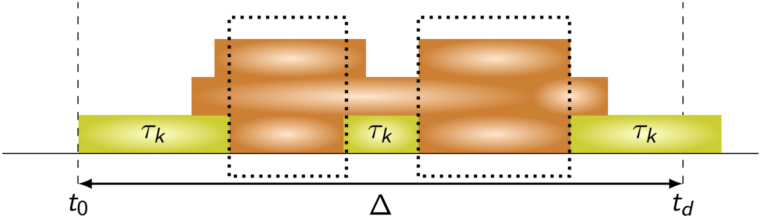
Task τ_k always meets its deadline, if

- $\forall$ positive integer ℓ, i.e., ℓ-th job of τ_k (in a continuously actively interval) misses its deadline
- for $\rho = \max_i^k\{\frac{C_i}{T_i}, \frac{C_i}{D_i}\}$, $\forall \Delta \geq D'_k = (\ell - 1)T_k + D_k$
- such that

$$(\lceil M - (M-1)\rho \rceil - 1)\,\rho\Delta + \sum_{i=1}^{k} 2\times dbf_i(\Delta) \leq (M - (M-1)\rho)\Delta$$

since $D_i \leq \Delta$

Baruah and Fisher's Analysis for Global-DM



Task τ_k always meets its deadline, if

- $\forall$ positive integer ℓ, i.e., ℓ-th job of τ_k (in a continuously actively interval) misses its deadline
- for $\rho = \max_i^k\{\frac{C_i}{T_i}, \frac{C_i}{D_i}\}$, $\forall \Delta \geq D'_k = (\ell - 1)T_k + D_k$
- such that

$$(\lceil M-(M-1)\rho \rceil - 1)\,\rho + 2\times \sum_{i=1}^{k} \frac{dbf_i(\Delta)}{\Delta} \leq (M-(M-1)\rho)$$

since $D_i \leq \Delta$

Enforcement Techniques (My Slide at ECRTS'17)

- Strong and/or early-design/analytical enforcements
  - simplify the analysis

Enforcement Techniques (My Slide at ECRTS'17)

- Strong and/or early-design/analytical enforcements
  - simplify the analysis
  - at the expense of poor performance in practical settings when compared to other algorithms or tests

Enforcement Techniques (My Slide at ECRTS'17)

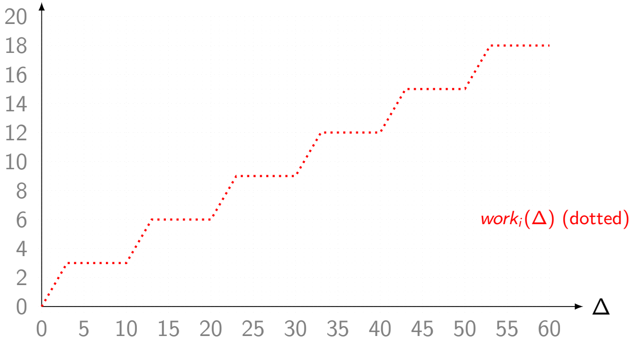
- Strong and/or early-design/analytical enforcements
  - simplify the analysis
  - at the expense of poor performance in practical settings when compared to other algorithms or tests

Observation 6

Adding enforcements tailoring the design of a scheduling algorithm or test to facilitate the derivation of a bounded speedup factor can be counterproductive; it may severely compromise performance in practical settings.

Workload Upper Bounds

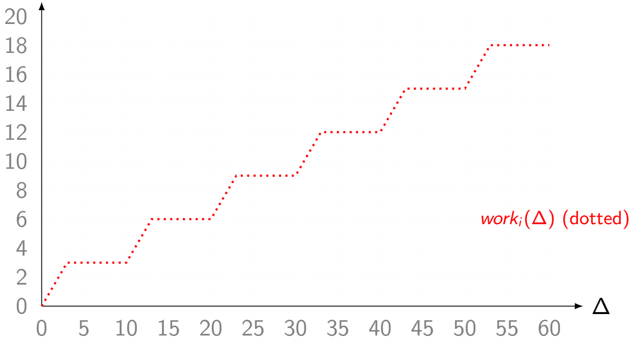
$T_i = 10, C_i = 3$



- Executed workload of a non-carry-in task τ_i from t_0 to $t_d = t_0 + \Delta$ is at most $work_i(\Delta)$
- Executed workload of a carry-in task τ_i from t_0 to $t_d = t_0 + \Delta$ is at most $work_i(\Delta + D_i)$

Workload Upper Bounds

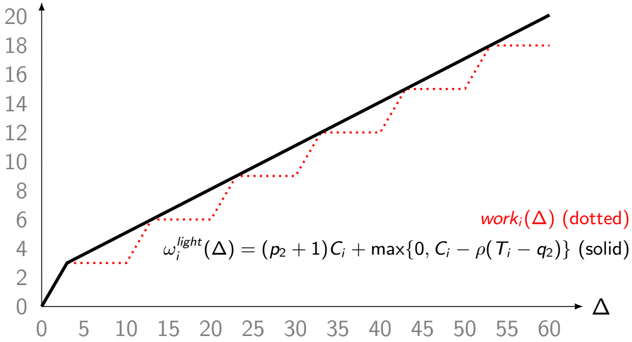
$T_i = 10, C_i = 3$



- Executed workload of a non-carry-in task τ_i from t_0 to $t_d = t_0 + \Delta$ is at most $work_i(\Delta)$
- Executed workload of a carry-in task τ_i from t_0 to $t_d = t_0 + \Delta$ is at most $\omega_i^{heavy}(\Delta) = work_i(\Delta + D_i)$ if $U_i > \rho$

Workload Upper Bounds

$T_i = 10, C_i = 3$ $\quad p_2 = \lceil (\Delta - C_i)/T_i \rceil - 1,\ q_2 = \Delta - C_i - p_2 T_i$

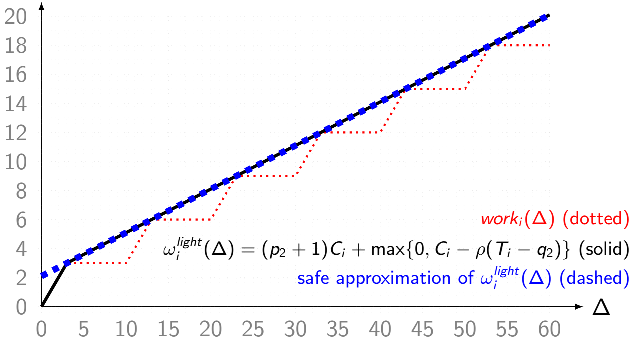


- Executed workload of a non-carry-in task τ_i from t_0 to $t_d = t_0 + \Delta$ is at most $work_i(\Delta)$
- Executed workload of a carry-in task τ_i from t_0 to $t_d = t_0 + \Delta$ is at most $\omega_i^{heavy}(\Delta) = work_i(\Delta + D_i)$ if $U_i > \rho$
- Executed workload of a carry-in task τ_i from t_0 to $t_d = t_0 + \Delta$ is at most $\omega_i^{light}(\Delta)$ if $U_i \leq \rho$

Workload Upper Bounds

$T_i = 10, C_i = 3$

$p_2 = \lceil (\Delta - C_i)/T_i \rceil - 1,\ q_2 = \Delta - C_i - p_2 T_i$



- Executed workload of a non-carry-in task τ_i from t_0 to $t_d = t_0 + \Delta$ is at most $work_i(\Delta) \leq (C_i - U_i D_i) + U_i \Delta$
- Executed workload of a carry-in task τ_i from t_0 to $t_d = t_0 + \Delta$ is at most $\omega_i^{heavy}(\Delta) = work_i(\Delta + D_i) \leq (C_i - U_i D_i) + U_i \Delta + U_i D_i$ if $U_i > \rho$
- Executed workload of a carry-in task τ_i from t_0 to $t_d = t_0 + \Delta$ is at most $\omega_i^{light}(\Delta) \leq (C_i - U_i D_i) + U_i \Delta$ if $U_i \leq \rho$

Putting Them Together (Theorem 4.4)

Task τ_k is schedulable by the given Global FP if

$$\forall \ell \in \mathbb{N}, \exists 1 \geq \rho \geq \ell C_k/((\ell-1)T_k + D_k), \forall \Delta \geq D'_k = (\ell-1)T_k + D_k,$$

$$\ell C_k + \sum_{\tau_i \in \mathbf{T}^{carry-approx}} \gamma_i U_i D_i + \sum_{i=1}^{k-1} (C_i - C_i U_i + U_i \Delta) \leq \mu_k \Delta$$

where $\mu_k = M - (M-1)\rho$,

$$\gamma_i = \begin{cases} 1 & \text{if } U_i > \rho \\ 0 & \text{if } U_i \leq \rho \end{cases}$$

and $\mathbf{T}^{carry-approx}$ is the set of the $\lceil \mu_k \rceil - 1$ tasks among the $k-1$ higher-priority tasks with the largest values of $\gamma_i U_i D_i$.

Putting Them Together (Theorem 4.4)

Task τ_k is schedulable by the given Global FP if

$$\forall \ell \in \mathbb{N}, \exists 1 \geq \rho \geq \ell C_k/((\ell-1)T_k + D_k), \forall \Delta \geq D'_k = (\ell-1)T_k + D_k,$$

$$\ell C_k + \sum_{\tau_i \in \mathbf{T}^{carry-approx}} \gamma_i U_i D_i + \sum_{i=1}^{k-1} (C_i - C_i U_i + U_i \Delta) \leq \mu_k \Delta$$

where $\mu_k = M - (M-1)\rho$,

$$\gamma_i = \begin{cases} 1 & \text{if } U_i > \rho \\ 0 & \text{if } U_i \leq \rho \end{cases}$$

and $\mathbf{T}^{carry-approx}$ is the set of the $\lceil \mu_k \rceil - 1$ tasks among the $k-1$ higher-priority tasks with the largest values of $\gamma_i U_i D_i$.

The worst case of Δ happens when $\Delta = D'_k$

Further Approximations

- Which ρ? (Theorem 4.6)

  Task τ_k is schedulable by the given Global-FP if

$$\forall \ell \in \mathbb{N}, \frac{\ell C_k}{D'_k} + \sum_{i=1}^{k-1} \left(\frac{C_i - C_i U_i}{D'_k} + U_i \right) \leq (M - (M-1) U_{\delta,k}^{\max})$$

D'_k is $(\ell - 1) T_k + D_k$ and $U_{\delta,k}^{\max} = \max\{\max_{i=1}^{k-1} U_i, \frac{C_k}{T_k}, \frac{C_k}{D_k}\}$

Further Approximations

- Which ρ? (Theorem 4.6)

  Task τ_k is schedulable by the given Global-FP if

$$\forall \ell \in \mathbb{N}, \frac{\ell C_k}{D'_k} + \sum_{i=1}^{k-1} \left(\frac{C_i - C_i U_i}{D'_k} + U_i \right) \leq (M - (M-1) U_{\delta,k}^{\max})$$

  D'_k is $(\ell - 1) T_k + D_k$ and $U_{\delta,k}^{\max} = \max\{\max_{i=1}^{k-1} U_i, \frac{C_k}{T_k}, \frac{C_k}{D_k}\}$

- Which ℓ? (Theorem 4.7)

  Task τ_k is schedulable by the given Global-FP if

$$\max\left\{ \frac{C_k}{T_k}, \frac{C_k}{D_k} \right\} + \sum_{i=1}^{k-1} \left(\frac{C_i - C_i U_i}{D_k} + U_i \right) \leq M - (M-1) U_{\delta,k}^{\max}$$

Conclusion

- We prove that Sanjoy's idea can yield a speedup factor of 3 for Global-DM

Conclusion

- We prove that Sanjoy's idea can yield a speedup factor of 3 for Global-DM

Future work

- Priority assignment: OPA-Compatible
- Proofs for the speedup bound of the optimal Global-FP
- Soft real-time tasks

(Have you seen Sanjoy during the talk?)

More materials: (links are in the paper)

- Impl. of Thm. 4.4 in $O(N \log N)$ for testing whether τ_k meets its deadline
- Evaluation results with many figures

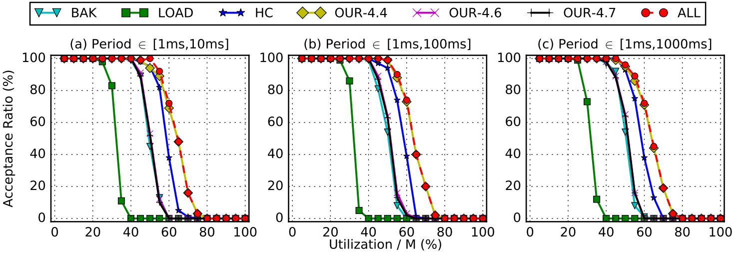


Figure: Global-DM, $M = 8$, $N = 40$, $\frac{D_i}{T_i} \in [0.8, 2]$.