Energy-aware scheduling of flow applications on master-worker platforms

Jean-François PINEAU
Yves ROBERT and Frédéric VIVIEN

Laboratoire d'Informatique, de Robotique, et de MicroÉlectronique de Montpellier

Jean-Francois.Pineau@ens-lyon.org

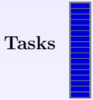
http://www.lirmm.fr/~pineau

ASTEC, June, 2009

Outline

1. Framework
   - Application and platform
   - Energy
   - Objective function
2. At the processor level
   - Minimizing the power consumption
   - Maximizing the throughput
3. At the system level
   - Ideal model
   - Model with start-up overheads
4. Conclusion

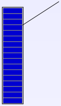
Applications and platform



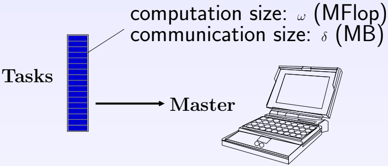
Applications and platform

computation size: ω (MFlop)
communication size: δ (MB)

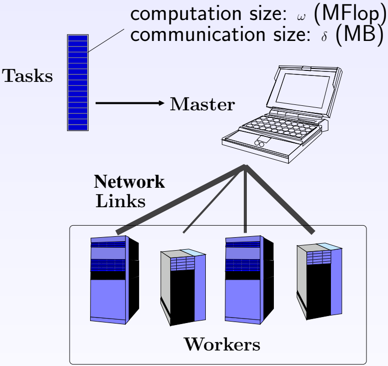
Tasks



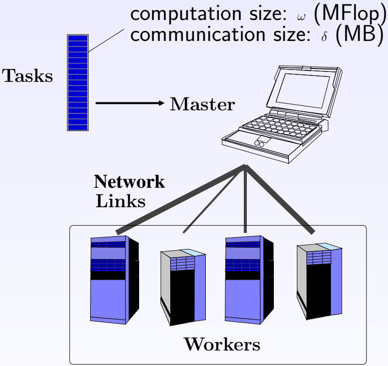
Applications and platform

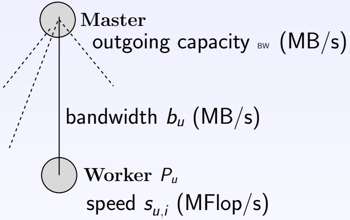


Applications and platform

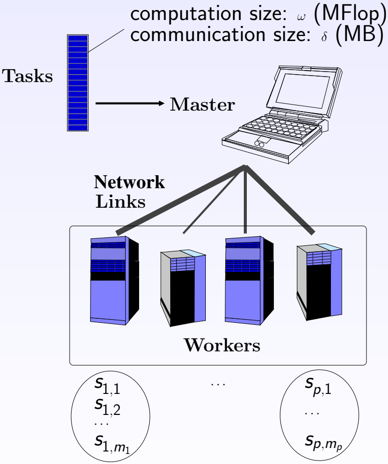


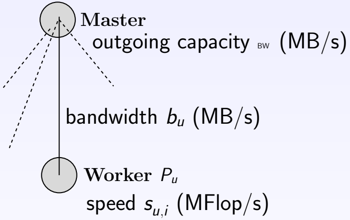
Applications and platform



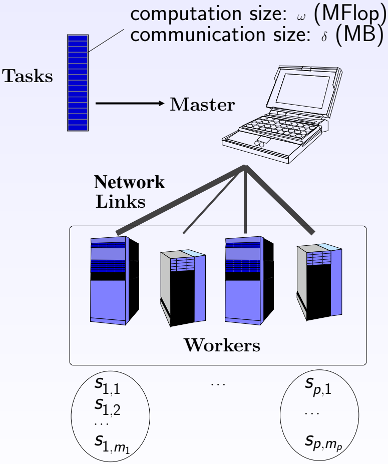


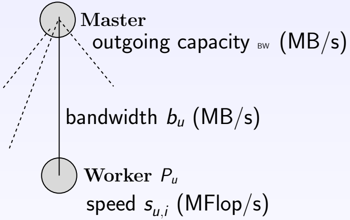
Applications and platform





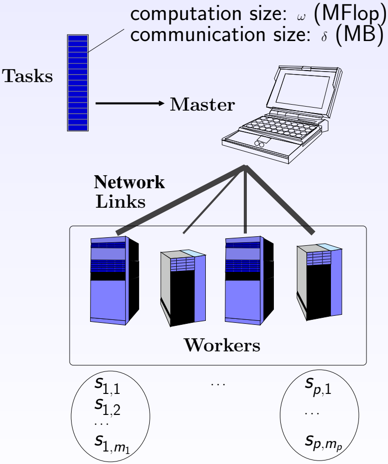
Applications and platform

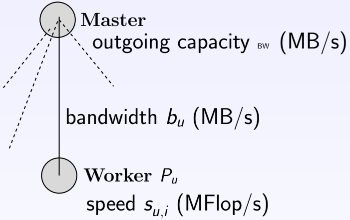




$$s_{u,1} \leq \dots \leq s_{u,i} \leq \dots \leq s_{u,m_u}$$

Applications and platform





$s_{u,1} \leq \dots \leq s_{u,i} \leq \dots \leq s_{u,m_u}$

All modes are exclusives

Outline

Energy model

- Most common power consumption formula:

$$P_d = s^{\alpha}, \text{ where } \alpha > 1.$$

- Our hypothesis:

Power consumption is a super-linear function.

- Power consumption of P_u at mode $s_{u,i}$ (fully used): $\mathfrak{P}_{u,i}$

Energy model

- Most common power consumption formula:

$$P_d = s^{\alpha}, \text{ where } \alpha > 1.$$

- Our hypothesis:

Power consumption is a super-linear function.

- Power consumption of P_u at mode $s_{u,i}$ (fully used): $\mathfrak{P}_{u,i}$

Energy model

- Most common power consumption formula:

$$P_d = s^\alpha, \text{ where } \alpha > 1.$$

- Our hypothesis:

Power consumption is a super-linear function.

- Power consumption of P_u at mode $s_{u,i}$ (fully used): $\mathfrak{P}_{u,i}$

Between modes

Fluid model:

switching among the modes does not cost any penalty.

Idle:

- an idle processor does not consume any power
- once a processor is on, it will always be above $s_{u,1}$

Overhead:

- time overhead
- power overhead
    - when turning on the worker
    - when turning it off
    - for each transition of mode

Between modes

Fluid model:

switching among the modes does not cost any penalty.

Idle:

- an idle processor does not consume any power
- once a processor is on, it will always be above $s_{u,1}$

Overhead:

- time overhead
- power overhead
  - when turning on the worker
  - when turning it off
  - for each transition of mode

Between modes

Fluid model:

switching among the modes does not cost any penalty.

Idle:

- an idle processor does not consume any power
- once a processor is on, it will always be above $s_{u,1}$

Overhead:

- time overhead
- power overhead
  - when turning on the worker
  - when turning it off
  - for each transition of mode

Between modes

Fluid model:

switching among the modes does not cost any penalty.

Idle:

- an idle processor does not consume any power
- once a processor is on, it will always be above $s_{u,1}$

Overhead:

- time overhead
- power overhead
  - when turning on the worker
  - when turning it off
  - for each transition of mode

Between modes

Fluid model:

switching among the modes does not cost any penalty.

Idle:

- an idle processor does not consume any power
- once a processor is on, it will always be above $s_{u,1}$

Overhead:

- ~~time overhead~~
- power overhead
  - when turning on the worker
  - when turning it off
  - for each transition of mode

Between modes

Fluid model:

switching among the modes does not cost any penalty.

Idle:

- an idle processor does not consume any power
- once a processor is on, it will always be above $s_{u,1}$

Overhead:

- ~~time overhead~~
- power overhead
  - when turning on the worker
  - when turning it off
  - for each transition of mode

Between modes

Fluid model:

switching among the modes does not cost any penalty.

Idle:

- an idle processor does not consume any power
- once a processor is on, it will always be above $s_{u,1}$

Overhead:

- ~~time overhead~~
- power overhead
  - when turning on the worker
  - when turning it off
  - for each transition of mode

Models

- Ideal model:
  - an idle processor does not consume any power
  - switching among the modes does not cost any penalty
- Model with start-up overheads
  - once a processor is on, it will always be above $s_{u,1}$
  - power consumption depends on the length of the interval

$$\mathfrak{P}_{u,i}(t) = \mathfrak{P}_{u,i}^{(1)} \cdot t + \mathfrak{P}_{u}^{(2)}$$

Models

- Ideal model:
  - an idle processor does not consume any power
  - switching among the modes does not cost any penalty
- Model with start-up overheads
  - once a processor is on, it will always be above $s_{u,1}$
  - power consumption depends on the length of the interval

$$\mathfrak{P}_{u,i}(t) = \mathfrak{P}_{u,i}^{(1)} \cdot t + \mathfrak{P}_{u}^{(2)}$$

Outline

Steady-state

Motivations

- Assume the number of tasks is huge
- Concentrate on *throughput* (fluid framework)
- Assume the workers can run at different speed
- Concentrate on *energy minimization*

Bi-criteria problem

Steady-state

Motivations

- Assume the number of tasks is huge
- Concentrate on *throughput* (fluid framework)

- Assume the workers can run at different speed
- Concentrate on *energy minimization*

Bi-criteria problem

Steady-state

Motivations

- Assume the number of tasks is huge
- Concentrate on *throughput* (fluid framework)

- Assume the workers can run at different speed
- Concentrate on *energy minimization*

Bi-criteria problem

Laptop or server ?

Laptop problem:
"What is the best schedule achievable using a particular energy budget, before battery becomes critically low?"

MaxThroughput ($\mathfrak{P}$): *maximizing the throughput while not exceeding the power consumption* $\mathfrak{P}$

Server problem:
"What is the least energy required to achieve a desired level of performance?"

MinPower (ρ): *minimizing the power consumption while achieving a throughput* ρ

Laptop or server ?

Laptop problem:
"What is the best schedule achievable using a particular energy budget, before battery becomes critically low?"

MaxThroughput ($\mathfrak{P}$): *maximizing the throughput while not exceeding the power consumption* $\mathfrak{P}$

Server problem:
"What is the least energy required to achieve a desired level of performance?"

MinPower (ρ): *minimizing the power consumption while achieving a throughput* ρ

Outline

Problem

Goal: Minimization of the power consumption of P_u.

Constraints:

- P_u has to ensure a given throughput,
- the processing capacity of $P_{u,i}$ cannot be exceeded, and the different modes are exclusive.

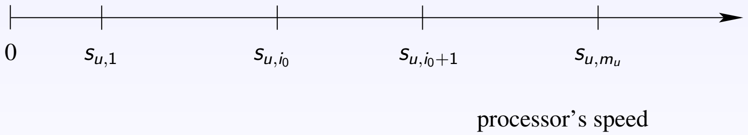
Problem

Goal: Minimization of the power consumption of P_u.

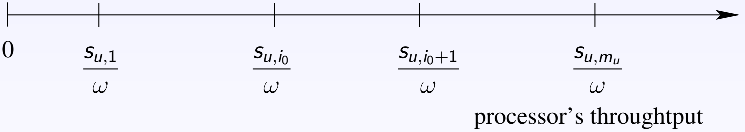
Constraints:

- P_u has to ensure a given throughput,
- the processing capacity of $P_{u,i}$ cannot be exceeded, and the different modes are exclusive.

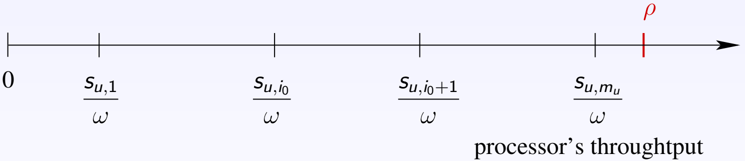
Optimal scheduling



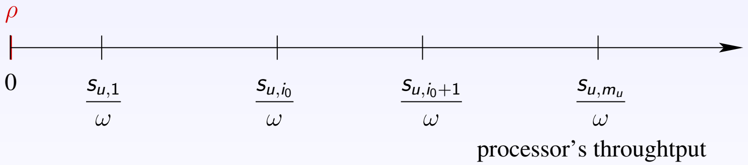
Optimal scheduling



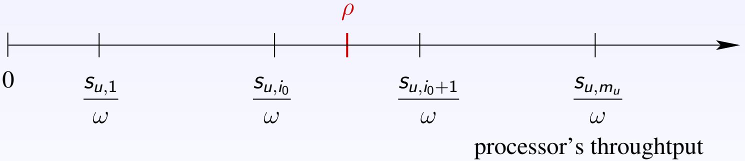
Optimal scheduling



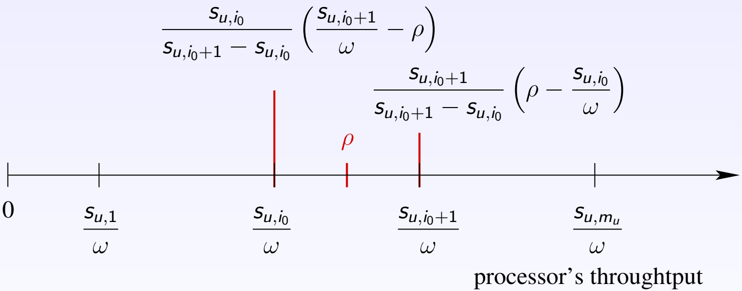
Optimal scheduling



Optimal scheduling



Optimal scheduling



Theorem 1

Theorem

This scheduling is optimal to minimize the power consumption while achieving a throughput of ρ.

The power consumption is then:

$$\mathfrak{P}_u(\rho) = (\omega\rho - s_{u,i_0})\frac{\mathfrak{P}_{u,i_0+1} - \mathfrak{P}_{u,i_0}}{s_{u,i_0+1} - s_{u,i_0}} + \mathfrak{P}_{u,i_0}$$

Theorem 1

Theorem

This scheduling is optimal to minimize the power consumption while achieving a throughput of ρ.

The power consumption is then:

$$\mathfrak{P}_u(\rho) = (\omega\rho - s_{u,i_0})\frac{\mathfrak{P}_{u,i_0+1} - \mathfrak{P}_{u,i_0}}{s_{u,i_0+1} - s_{u,i_0}} + \mathfrak{P}_{u,i_0}$$

Theorem 1

Theorem

This scheduling is optimal to minimize the power consumption while achieving a throughput of ρ.

The power consumption is then:

$$\begin{aligned} \mathfrak{P}_u(\rho) &= (\omega\rho - s_{u,i_0})\frac{\mathfrak{P}_{u,i_0+1} - \mathfrak{P}_{u,i_0}}{s_{u,i_0+1} - s_{u,i_0}} + \mathfrak{P}_{u,i_0} \\ &= \max_{0 \le i < m_u} \left\{ (\omega\rho - s_{u,i})\frac{\mathfrak{P}_{u,i+1} - \mathfrak{P}_{u,i}}{s_{u,i+1} - s_{u,i}} + \mathfrak{P}_{u,i} \right\}. \end{aligned}$$

because $\mathfrak{P}$ is super-linear.

Theorem 2

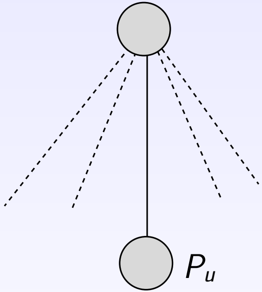
The maximum achievable throughput according to the power consumption limit $\mathfrak{P}$ is

$$\rho_u(\mathfrak{P}) = \min \left\{ \begin{array}{c} \dfrac{s_{u,m_u}}{\omega} \\ \max\limits_{1 \leq i \leq m_u} \left\{ \dfrac{\mathfrak{P}(s_{u,i+1} - s_{u,i}) + s_{u,i}\mathfrak{P}_{u,i+1} - s_{u,i+1}\mathfrak{P}_{u,i}}{\omega(\mathfrak{P}_{u,i+1} - \mathfrak{P}_{u,i})} \right\} \end{array} \right.$$

Outline

1. Framework
   - Application and platform
   - Energy
   - Objective function
2. At the processor level
   - Minimizing the power consumption
   - Maximizing the throughput
3. At the system level
   - Ideal model
   - Model with start-up overheads
4. Conclusion

Simplifying the problem

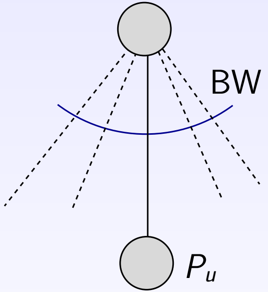


Constraint of the system: ρ

$\frac{BW}{\delta} \geq \rho$, or the system as no solution.

Use of the previous theorems

Simplifying the problem

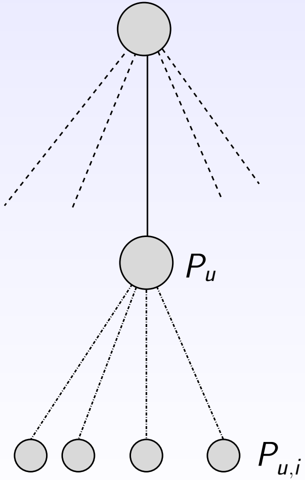


Constraint of the system: ρ

$\frac{\text{BW}}{\delta} \geq \rho$, or the system as no solution.

Use of the previous theorems

Simplifying the problem

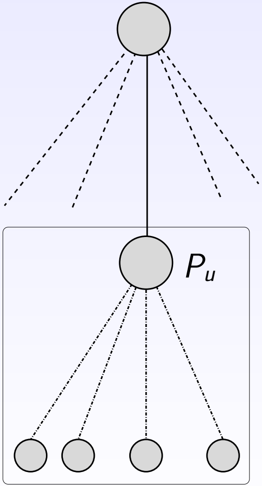


Constraint of the system: ρ

$\frac{\text{BW}}{\delta} \geq \rho$, or the system as no solution.

Use of the previous theorems

Simplifying the problem

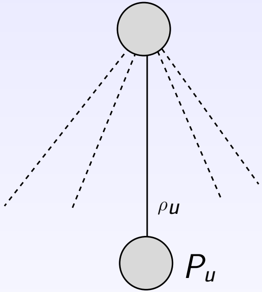


Constraint of the system: ρ

$\frac{\text{BW}}{\delta} \geq \rho$, or the system as no solution.

Use of the previous theorems

Simplifying the problem

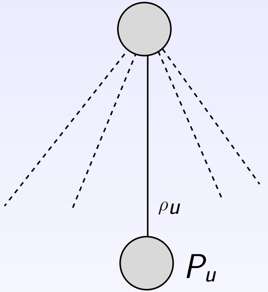


Constraint of the system: ρ

$\frac{\text{BW}}{\delta} \geq \rho$, or the system as no solution.

Use of the previous theorems

Simplifying the problem



$$\left(\omega\rho_u - s_{u,i}\right)\frac{\mathfrak{P}_{u,i+1} - \mathfrak{P}_{u,i}}{s_{u,i+1} - s_{u,i}} + \mathfrak{P}_{u,i}$$

Constraint of the system: ρ

$\frac{\mathrm{BW}}{\delta} \geq \rho$, or the system as no solution.

Use of the previous theorems

Algorithm

$$\left\{ \frac{\mathfrak{P}_{u,i+1} - \mathfrak{P}_{u,i}}{s_{u,i+1} - s_{u,i}} \right\}$$

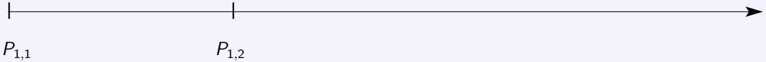
Algorithm

$$\left\{ \frac{\mathfrak{P}_{u,i+1} - \mathfrak{P}_{u,i}}{s_{u,i+1} - s_{u,i}} \right\}$$

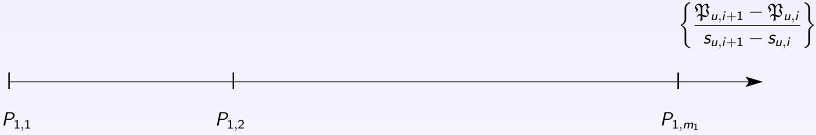
$P_{1,1}$

Algorithm

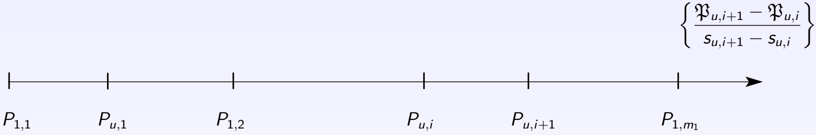
$$\left\{ \frac{\mathfrak{P}_{u,i+1} - \mathfrak{P}_{u,i}}{s_{u,i+1} - s_{u,i}} \right\}$$



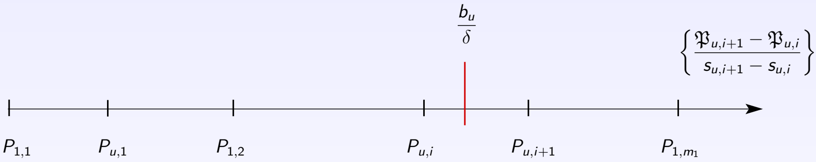
Algorithm


$\left\{ \frac{\mathfrak{P}_{u,i+1} - \mathfrak{P}_{u,i}}{s_{u,i+1} - s_{u,i}} \right\}$
$P_{1,1}$
$P_{1,2}$
P_{1,m_1}

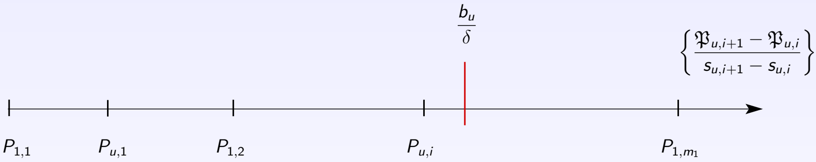
Algorithm


$\left\{ \frac{\mathfrak{P}_{u,i+1} - \mathfrak{P}_{u,i}}{s_{u,i+1} - s_{u,i}} \right\}$
$P_{1,1}$
$P_{u,1}$
$P_{1,2}$
$P_{u,i}$
$P_{u,i+1}$
P_{1,m_1}

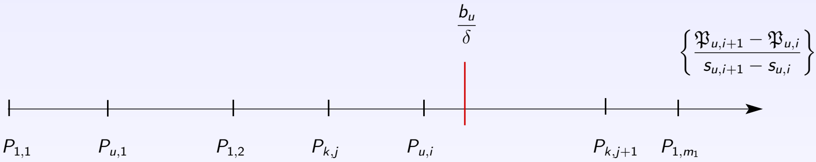
Algorithm


$\frac{b_u}{\delta}$
$\left\{ \frac{\mathfrak{P}_{u,i+1} - \mathfrak{P}_{u,i}}{s_{u,i+1} - s_{u,i}} \right\}$
$P_{1,1}$
$P_{u,1}$
$P_{1,2}$
$P_{u,i}$
$P_{u,i+1}$
P_{1,m_1}

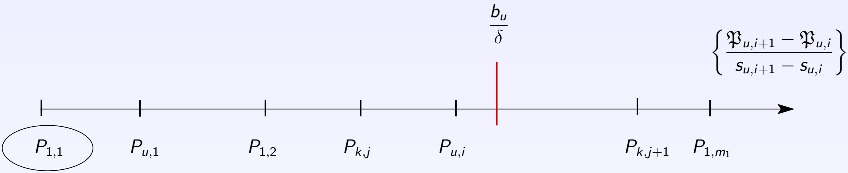
Algorithm


$\frac{b_u}{\delta}$
$\left\{ \frac{\mathfrak{P}_{u,i+1} - \mathfrak{P}_{u,i}}{s_{u,i+1} - s_{u,i}} \right\}$
$P_{1,1}$
$P_{u,1}$
$P_{1,2}$
$P_{u,i}$
P_{1,m_1}

Algorithm

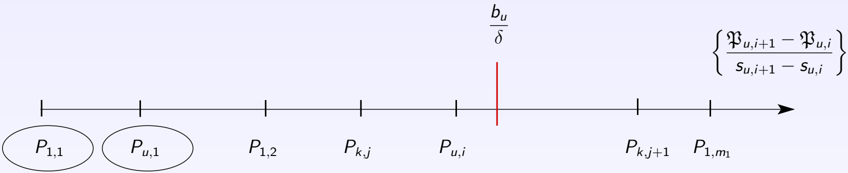

$\frac{b_u}{\delta}$
$\left\{ \frac{\mathfrak{P}_{u,i+1} - \mathfrak{P}_{u,i}}{s_{u,i+1} - s_{u,i}} \right\}$
$P_{1,1}$
$P_{u,1}$
$P_{1,2}$
$P_{k,j}$
$P_{u,i}$
$P_{k,j+1}$
P_{1,m_1}

Algorithm



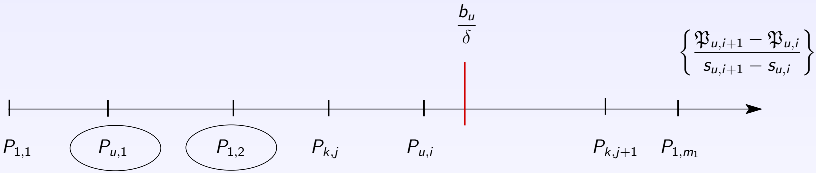
Throughput of the system : $\Phi = \frac{s_{1,1}}{\omega}$

Algorithm



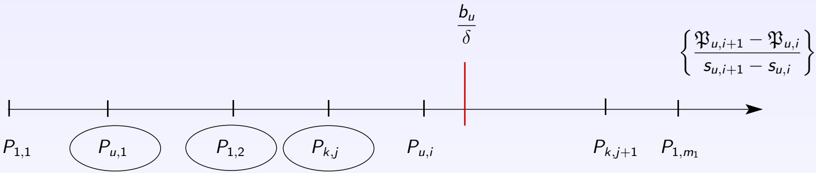
Throughput of the system : $\Phi = \quad \Phi + \frac{s_{u,1}}{\omega}$

Algorithm



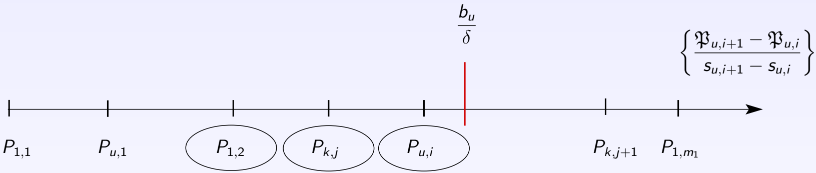
Throughput of the system : $\Phi = \quad \Phi - \frac{s_{1,1}}{\omega} + \frac{s_{1,2}}{\omega}$

Algorithm



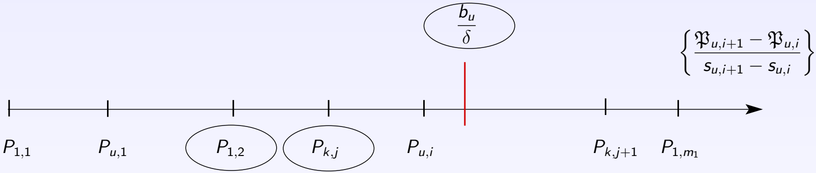
Throughput of the system : $\Phi = \quad \Phi - \frac{s_{k,j-1}}{\omega} + \frac{s_{k,j}}{\omega}$

Algorithm



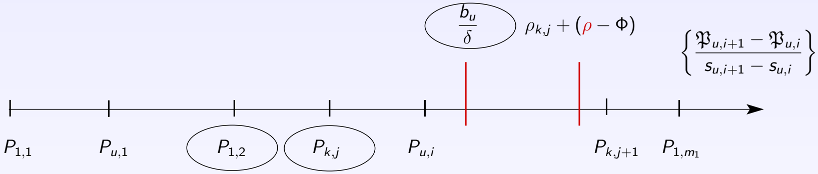
Throughput of the system : $\Phi = \quad \Phi - \frac{s_{u,i-1}}{\omega} + \frac{s_{u,i}}{\omega}$

Algorithm

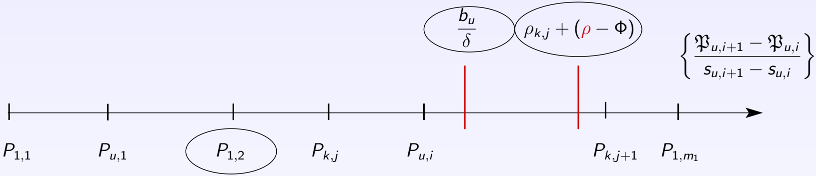


Throughput of the system : $\Phi = \quad \Phi - \frac{s_{u,i}}{\omega} + \frac{b_u}{\delta}$

Algorithm


$\frac{b_u}{\delta}$
$\rho_{k,j} + (\rho - \Phi)$
$\left\{ \frac{\mathfrak{P}_{u,i+1} - \mathfrak{P}_{u,i}}{s_{u,i+1} - s_{u,i}} \right\}$
$P_{1,1}$
$P_{u,1}$
$P_{1,2}$
$P_{k,j}$
$P_{u,i}$
$P_{k,j+1}$
P_{1,m_1}

Algorithm



Throughput of the system : $\Phi = \quad \rho$

Proof of the algorithm

Theorem

Our algorithm optimally solves problem MinPower *(ρ).*

- $\tilde{\mathcal{S}} = \{\tilde{\rho}_u\}$: throughput of each processor given by our algorithm
- $\mathcal{S} = \{\rho_u\}$: another optimal solution

Proof of the algorithm

Theorem

Our algorithm optimally solves problem $\textsc{MinPower}$ *(ρ).*

- $\tilde{\mathcal{S}} = \{\tilde{\rho}_u\}$: throughput of each processor given by our algorithm
- $\mathcal{S} = \{\rho_u\}$: another optimal solution

Proof (1/2)

	$\tilde{\mathcal{S}}$	$\mathcal{S}$	
P_1	$\tilde{\rho}_1$	ρ_1	
P_u	$\tilde{\rho}_u$	ρ_u	
P_p	$\tilde{\rho}_p$	ρ_p	

Proof (1/2)

	$\tilde{\mathcal{S}}$		$\mathcal{S}$	
P_1	$\tilde{\rho}_1$		ρ_1	
	$\tilde{\rho}_{\min}$	$>$	$\rho_{\min}$	
P_u	$\tilde{\rho}_u$		ρ_u	
P_p	$\tilde{\rho}_p$		ρ_p	

Proof (1/2)

	$\tilde{\mathcal{S}}$		$\mathcal{S}$	
P_1	$\tilde{\rho}_1$		ρ_1	
$P_{\min}$	$\tilde{\rho}_{\min}$	$>$	$\rho_{\min}$	
P_u	$\tilde{\rho}_u$		ρ_u	
P_p	$\tilde{\rho}_p$		ρ_p	

Proof (1/2)

	$\tilde{\mathcal{S}}$		$\mathcal{S}$	
P_1	$\tilde{\rho}_1$		ρ_1	
$P_{\min}$	$\tilde{\rho}_{\min}$	$>$	$\rho_{\min}$	
P_u	$\tilde{\rho}_u$		ρ_u	
	$\tilde{\rho}_{\mathrm{MAX}}$	$<$	ρ_{MAX}	
P_p	$\tilde{\rho}_p$		ρ_p	

Proof (1/2)

	$\tilde{\mathcal{S}}$		$\mathcal{S}$	
P_1	$\tilde{\rho}_1$		ρ_1	
P_{min}	$\tilde{\rho}_{\text{min}}$	$>$	ρ_{min}	
P_u	$\tilde{\rho}_u$		ρ_u	
P_{MAX}	$\tilde{\rho}_{\text{MAX}}$	$<$	ρ_{MAX}	
P_p	$\tilde{\rho}_p$		ρ_p	

Proof (1/2)

	$\tilde{\mathcal{S}}$		$\mathcal{S}$	
P_1	$\tilde{\rho}_1$		ρ_1	
P_{min}	$\tilde{\rho}_{\text{min}}$	$>$	ρ_{min}	
P_u	$\tilde{\rho}_u$		ρ_u	
P_{MAX}	$\tilde{\rho}_{\text{MAX}}$	$<$	ρ_{MAX}	
P_p	$\tilde{\rho}_p$		ρ_p	

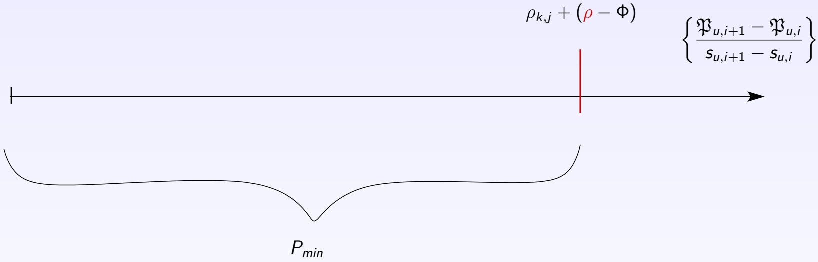
$$\epsilon = \min\{\tilde{\rho}_{\text{min}} - \rho_{\text{min}}; \rho_{\text{MAX}} - \tilde{\rho}_{\text{MAX}}\}.$$

Proof (1/2)

	$\tilde{\mathcal{S}}$		$\mathcal{S}$	$\mathcal{S}'$
P_1	$\tilde{\rho}_1$		ρ_1	ρ_1
P_{min}	$\tilde{\rho}_{\text{min}}$	$>$	ρ_{min}	$\rho_{\text{min}} + \epsilon$
P_u	$\tilde{\rho}_u$		ρ_u	ρ_u
P_{MAX}	$\tilde{\rho}_{\text{MAX}}$	$<$	ρ_{MAX}	$\rho_{\text{MAX}} - \epsilon$
P_p	$\tilde{\rho}_p$		ρ_p	ρ_p

$$\epsilon = \min\{\tilde{\rho}_{\text{min}} - \rho_{\text{min}}; \rho_{\text{MAX}} - \tilde{\rho}_{\text{MAX}}\}.$$

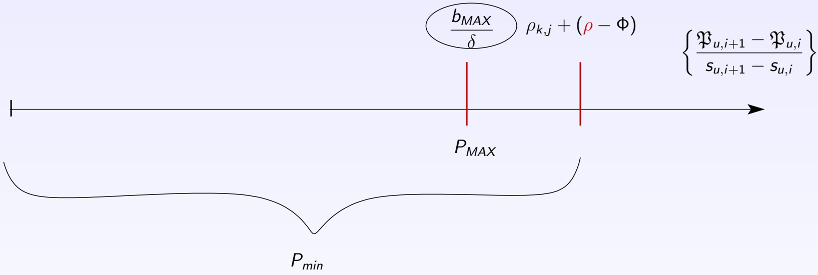
Proof (2/2)



Conclusion

$\mathcal{S}'$ consumes no more power than $\mathcal{S}$

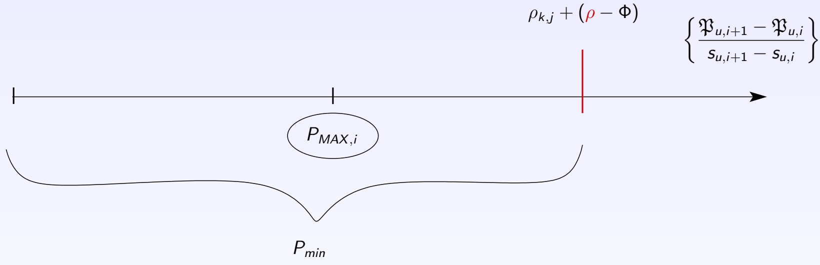
Proof (2/2)



Conclusion

$\mathcal{S}'$ consumes no more power than $\mathcal{S}$

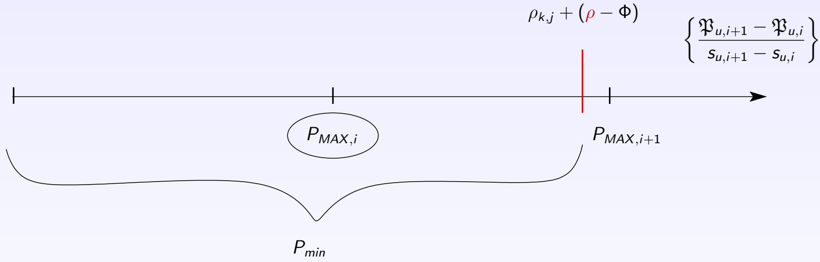
Proof (2/2)



Conclusion

$\mathcal{S}'$ consumes no more power than $\mathcal{S}$

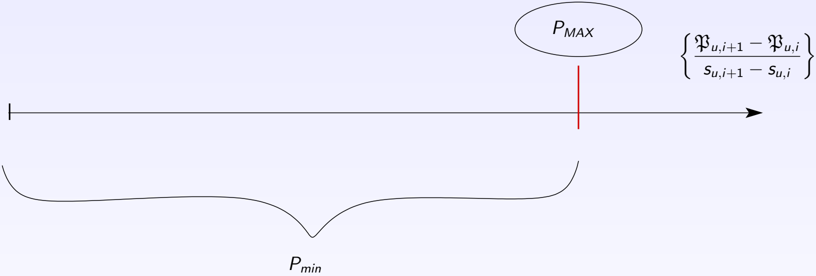
Proof (2/2)



Conclusion

$\mathcal{S}'$ consumes no more power than $\mathcal{S}$

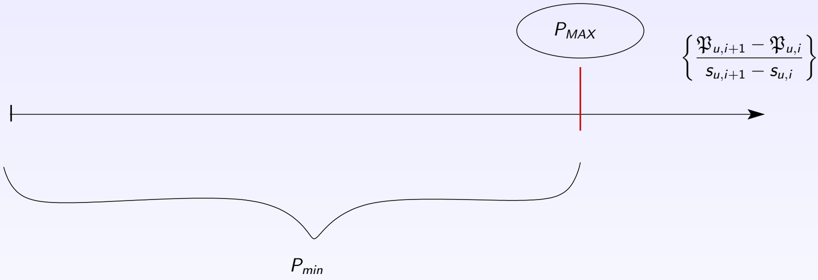
Proof (2/2)



Conclusion

$\mathcal{S}'$ consumes no more power than $\mathcal{S}$

Proof (2/2)



Conclusion

$\mathcal{S}'$ consumes no more power than $\mathcal{S}$

Maximizing the throughput

- Same principle
- Simple change of objective function:

$$\mathcal{T}[u_k] \leftarrow \min \begin{cases} \mathfrak{P}_{u_k,i_k} \\ \left(\omega \frac{b_{u_k}}{\delta} - s_{u_k,i_k}\right) \frac{\mathfrak{P}_{u_k,i_k+1} - \mathfrak{P}_{u_k,i_k}}{s_{u_k,i_k+1} - s_{u_k,i_k}} + \mathfrak{P}_{u_k,i_k} \\ \mathfrak{P}' + (\mathfrak{P} - \Psi) \end{cases}$$

(Ψ is the current power consumption)

Outline

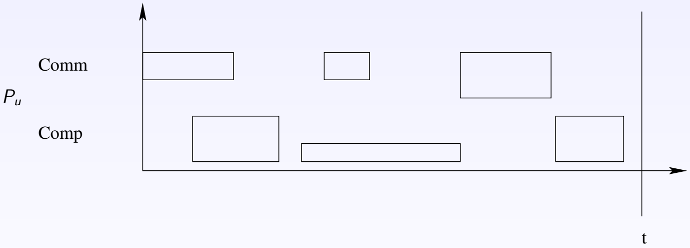
Property

Theorem

There exists an optimal schedule in which all processors, except possibly one, are used at a maximum throughput, i.e., either the throughput dictated by their bandwidth, or the throughput achieved by one of their execution modes.

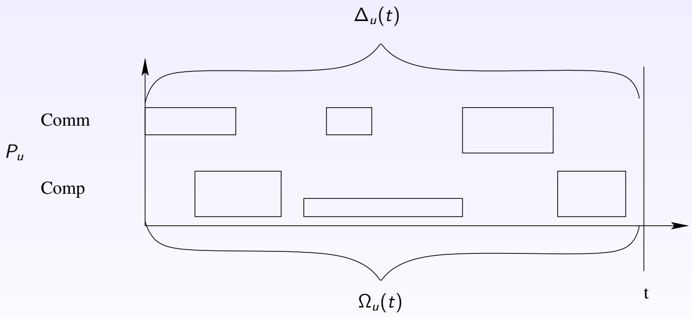
Proof

$\mathcal{S}$, optimal schedule, during t time-units:



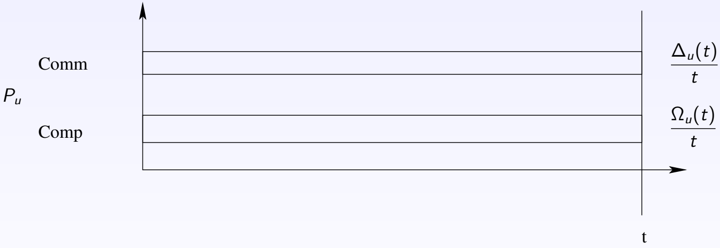
Proof

$\mathcal{S}$, optimal schedule, during t time-units:



Proof

$\mathcal{S}'$:



$\mathcal{S}'$ optimal

If $\mathcal{S}'$ does not have the desired property:

- P_{min}, P_{MAX}: not running at a maximum throughput.
- $\rho'_{\text{min}} \leftarrow P_{\text{min},i_{\text{min}}}, P_{\text{min},i_{\text{min}}+1}$
- $\rho'_{\text{MAX}} \leftarrow P_{\text{MAX},i_{\text{MAX}}}, P_{\text{MAX},i_{\text{MAX}}+1}$
-
$$\frac{\mathfrak{P}_{\text{min},i_{\text{min}}+1}(t) - \mathfrak{P}_{\text{min},i_{\text{min}}}(t)}{s_{\text{min},i_{\text{min}}+1} - s_{\text{min},i_{\text{min}}}} \leq \frac{\mathfrak{P}_{\text{MAX},i_{\text{MAX}+1}}(t) - \mathfrak{P}_{\text{MAX},i_{\text{MAX}}}(t)}{s_{\text{MAX},i_{\text{MAX}+1}} - s_{\text{MAX},i_{\text{MAX}}}}$$

$\mathcal{S}'$ optimal

If $\mathcal{S}'$ does not have the desired property:

- P_{min}, P_{MAX}: not running at a maximum throughput.
- $\rho'_{\text{min}} \leftarrow P_{\text{min},i_{\text{min}}},\ P_{\text{min},i_{\text{min}}+1}$
- $\rho'_{\text{MAX}} \leftarrow P_{\text{MAX},i_{\text{MAX}}},\ P_{\text{MAX},i_{\text{MAX}}+1}$
- $$\frac{\mathfrak{P}_{\text{min},i_{\text{min}}+1}(t) - \mathfrak{P}_{\text{min},i_{\text{min}}}(t)}{s_{\text{min},i_{\text{min}}+1} - s_{\text{min},i_{\text{min}}}} \leq \frac{\mathfrak{P}_{\text{MAX},i_{\text{MAX}+1}}(t) - \mathfrak{P}_{\text{MAX},i_{\text{MAX}}}(t)}{s_{\text{MAX},i_{\text{MAX}+1}} - s_{\text{MAX},i_{\text{MAX}}}}$$

$\mathcal{S}'$ optimal

If $\mathcal{S}'$ does not have the desired property:

- P_{min}, P_{MAX}: not running at a maximum throughput.
- $\rho'_{\text{min}} \leftarrow P_{\text{min},i_{\text{min}}}$, $P_{\text{min},i_{\text{min}}+1}$
- $\rho'_{\text{MAX}} \leftarrow P_{\text{MAX},i_{\text{MAX}}}$, $P_{\text{MAX},i_{\text{MAX}}+1}$
- $$\frac{\mathfrak{P}_{\text{min},i_{\text{min}}+1}(t) - \mathfrak{P}_{\text{min},i_{\text{min}}}(t)}{s_{\text{min},i_{\text{min}}+1} - s_{\text{min},i_{\text{min}}}} \leq \frac{\mathfrak{P}_{\text{MAX},i_{\text{MAX}+1}}(t) - \mathfrak{P}_{\text{MAX},i_{\text{MAX}}}(t)}{s_{\text{MAX},i_{\text{MAX}+1}} - s_{\text{MAX},i_{\text{MAX}}}}$$

$\mathcal{S}'$ optimal

If $\mathcal{S}'$ does not have the desired property:

- $P_{\min}$, P_{MAX}: not running at a maximum throughput.
- $\rho'_{\min} \leftarrow P_{\min,i_{\min}}$, $P_{\min,i_{\min}+1}$
- $\rho'_{\text{MAX}} \leftarrow P_{\text{MAX},i_{\text{MAX}}}$, $P_{\text{MAX},i_{\text{MAX}}+1}$
-
$$\frac{\mathfrak{P}_{\min,i_{\min}+1}(t) - \mathfrak{P}_{\min,i_{\min}}(t)}{s_{\min,i_{\min}+1} - s_{\min,i_{\min}}} \leq \frac{\mathfrak{P}_{\text{MAX},i_{\text{MAX}+1}}(t) - \mathfrak{P}_{\text{MAX},i_{\text{MAX}}}(t)}{s_{\text{MAX},i_{\text{MAX}+1}} - s_{\text{MAX},i_{\text{MAX}}}}$$

New schedule

$$\mathcal{S}'' : \begin{cases} \rho''_{\mathsf{min}} = \rho'_{\mathsf{min}} + \epsilon \\ \rho''_{\mathsf{MAX}} = \rho'_{\mathsf{MAX}} - \epsilon \\ \rho''_u = \rho'_u \text{ otherwise} \\ \epsilon = \min\left\{ \frac{b_u}{\delta} - \rho'_{\mathsf{min}}; \frac{s_{\mathsf{min}, i_{\mathsf{min}}+1}}{\omega} - \rho'_{\mathsf{min}}; \rho'_{\mathsf{MAX}} - \frac{s_{\mathsf{MAX}, i_{\mathsf{MAX}}}}{\omega} \right\} \end{cases}$$

$\mathcal{S}''$

- achieves the same throughput than $\mathcal{S}'$
- does not consume more power than $\mathcal{S}'$
- has less processor not at a maximum throughput than in $\mathcal{S}'$

New schedule

$$\mathcal{S}'' : \begin{cases} \rho''_{\text{min}} = \rho'_{\text{min}} + \epsilon \\ \rho''_{\text{MAX}} = \rho'_{\text{MAX}} - \epsilon \\ \rho''_u = \rho'_u \text{ otherwise} \\ \epsilon = \min\left\{ \frac{b_u}{\delta} - \rho'_{\text{min}}; \frac{s_{\text{min},i_{\text{min}}+1}}{\omega} - \rho'_{\text{min}}; \rho'_{\text{MAX}} - \frac{s_{\text{MAX},i_{\text{MAX}}}}{\omega} \right\} \end{cases}$$

$\mathcal{S}''$

- achieves the same throughput than $\mathcal{S}'$
- does not consume more power than $\mathcal{S}'$
- has less processor not at a maximum throughput than in $\mathcal{S}'$

New schedule

$$\mathcal{S}'' : \begin{cases} \rho''_{\text{min}} = \rho'_{\text{min}} + \epsilon \\ \rho''_{\text{MAX}} = \rho'_{\text{MAX}} - \epsilon \\ \rho''_u = \rho'_u \text{ otherwise} \\ \epsilon = \min\left\{ \frac{b_u}{\delta} - \rho'_{\text{min}} ; \frac{s_{\text{min},i_{\text{min}}+1}}{\omega} - \rho'_{\text{min}} ; \rho'_{\text{MAX}} - \frac{s_{\text{MAX},i_{\text{MAX}}}}{\omega} \right\} \end{cases}$$

$\mathcal{S}''$

- achieves the same throughput than $\mathcal{S}'$
- does not consume more power than $\mathcal{S}'$
- has less processor not at a maximum throughput than in $\mathcal{S}'$

New schedule

$$\mathcal{S}'' : \begin{cases} \rho''_{\text{min}} = \rho'_{\text{min}} + \epsilon \\ \rho''_{\text{MAX}} = \rho'_{\text{MAX}} - \epsilon \\ \rho''_u = \rho'_u \text{ otherwise} \\ \epsilon = \min\left\{ \frac{b_u}{\delta} - \rho'_{\text{min}} ; \frac{s_{\text{min}, i_{\text{min}}+1}}{\omega} - \rho'_{\text{min}} ; \rho'_{\text{MAX}} - \frac{s_{\text{MAX}, i_{\text{MAX}}}}{\omega} \right\} \end{cases}$$

$\mathcal{S}''$

- achieves the same throughput than $\mathcal{S}'$
- does not consume more power than $\mathcal{S}'$
- has less processor not at a maximum throughput than in $\mathcal{S}'$

Asympotic optimality (1/2)

- Algorithm look at the energy consumed during d time-units (d defined later)
- $\mathcal{A}$ is composed of $\mathcal{B}$ tasks
- optimal scheduling time: $\mathcal{T} = \frac{\mathcal{B}}{\rho}$, where ρ is the throughput bound.
- $\mathfrak{P}_{opt}$: the optimal power consumption that would be obtained in the ideal model
- $\mathfrak{P}^*$: the optimal power consumption that can be achieve under the model with start-up overheads
- $\mathfrak{P}$ the power consumption given by our algorithm.
- $\mathfrak{P}_{opt} \leq \mathfrak{P}^* \leq \mathfrak{P}$.
- during t time-units:

$$\mathfrak{P}(t) \leq \mathfrak{P}_{opt} \cdot t + \left\lceil \frac{t}{d} \right\rceil \sum_{u=1}^{p} \mathfrak{P}_u^{(2)}$$

Asympotic optimality (1/2)

- Algorithm look at the energy consumed during d time-units (d defined later)
- $\mathcal{A}$ is composed of $\mathcal{B}$ tasks
- optimal scheduling time: $\mathcal{T} = \frac{\mathcal{B}}{\rho}$, where ρ is the throughput bound.
- $\mathfrak{P}_{opt}$: the optimal power consumption that would be obtained in the ideal model
- $\mathfrak{P}^*$: the optimal power consumption that can be achieve under the model with start-up overheads
- $\mathfrak{P}$ the power consumption given by our algorithm.
- $\mathfrak{P}_{opt} \leq \mathfrak{P}^* \leq \mathfrak{P}$.
- during t time-units:

$$\mathfrak{P}(t) \leq \mathfrak{P}_{opt} \cdot t + \left\lceil \frac{t}{d} \right\rceil \sum_{u=1}^{p} \mathfrak{P}_u^{(2)}$$

Asympotic optimality (1/2)

- Algorithm look at the energy consumed during d time-units (d defined later)
- $\mathcal{A}$ is composed of $\mathcal{B}$ tasks
- optimal scheduling time: $\mathcal{T} = \frac{\mathcal{B}}{\rho}$, where ρ is the throughput bound.
- $\mathfrak{P}_{opt}$: the optimal power consumption that would be obtained in the ideal model
- $\mathfrak{P}^*$: the optimal power consumption that can be achieve under the model with start-up overheads
- $\mathfrak{P}$ the power consumption given by our algorithm.
- $\mathfrak{P}_{opt} \leq \mathfrak{P}^* \leq \mathfrak{P}$.
- during t time-units:

$$\mathfrak{P}(t) \leq \mathfrak{P}_{opt} \cdot t + \left\lceil \frac{t}{d} \right\rceil \sum_{u=1}^{p} \mathfrak{P}_u^{(2)}$$

Asympotic optimality (1/2)

- Algorithm look at the energy consumed during d time-units (d defined later)
- $\mathcal{A}$ is composed of $\mathcal{B}$ tasks
- optimal scheduling time: $\mathcal{T} = \frac{\mathcal{B}}{\rho}$, where ρ is the throughput bound.
- $\mathfrak{P}_{opt}$: the optimal power consumption that would be obtained in the ideal model
- $\mathfrak{P}^*$: the optimal power consumption that can be achieve under the model with start-up overheads
- $\mathfrak{P}$ the power consumption given by our algorithm.
- $\mathfrak{P}_{opt} \leq \mathfrak{P}^* \leq \mathfrak{P}$.
- during t time-units:

$$\mathfrak{P}(t) \leq \mathfrak{P}_{opt} \cdot t + \left\lceil \frac{t}{d} \right\rceil \sum_{u=1}^{p} \mathfrak{P}_u^{(2)}$$

Asympotic optimality (1/2)

- Algorithm look at the energy consumed during d time-units (d defined later)
- $\mathcal{A}$ is composed of $\mathcal{B}$ tasks
- optimal scheduling time: $\mathcal{T} = \frac{\mathcal{B}}{\rho}$, where ρ is the throughput bound.
- $\mathfrak{P}_{opt}$: the optimal power consumption that would be obtained in the ideal model
- $\mathfrak{P}^*$: the optimal power consumption that can be achieve under the model with start-up overheads
- $\mathfrak{P}$ the power consumption given by our algorithm.
- $\mathfrak{P}_{opt} \leq \mathfrak{P}^* \leq \mathfrak{P}$.
- during t time-units:

$$\mathfrak{P}(t) \leq \mathfrak{P}_{opt} \cdot t + \left\lceil \frac{t}{d} \right\rceil \sum_{u=1}^{p} \mathfrak{P}_u^{(2)}$$

Asympotic optimality (1/2)

- Algorithm look at the energy consumed during d time-units (d defined later)
- $\mathcal{A}$ is composed of $\mathcal{B}$ tasks
- optimal scheduling time: $\mathcal{T} = \frac{\mathcal{B}}{\rho}$, where ρ is the throughput bound.
- $\mathfrak{P}_{opt}$: the optimal power consumption that would be obtained in the ideal model
- $\mathfrak{P}^*$: the optimal power consumption that can be achieve under the model with start-up overheads
- $\mathfrak{P}$ the power consumption given by our algorithm.
- $\mathfrak{P}_{opt} \leq \mathfrak{P}^* \leq \mathfrak{P}$.
- during t time-units:

$$\mathfrak{P}(t) \leq \mathfrak{P}_{opt} \cdot t + \left\lceil \frac{t}{d} \right\rceil \sum_{u=1}^{p} \mathfrak{P}_u^{(2)}$$

Asympotic optimality (1/2)

- Algorithm look at the energy consumed during d time-units (d defined later)
- $\mathcal{A}$ is composed of $\mathcal{B}$ tasks
- optimal scheduling time: $\mathcal{T} = \frac{\mathcal{B}}{\rho}$, where ρ is the throughput bound.
- $\mathfrak{P}_{opt}$: the optimal power consumption that would be obtained in the ideal model
- $\mathfrak{P}^*$: the optimal power consumption that can be achieve under the model with start-up overheads
- $\mathfrak{P}$ the power consumption given by our algorithm.
- $\mathfrak{P}_{opt} \leq \mathfrak{P}^* \leq \mathfrak{P}$.
- during t time-units:

$$\mathfrak{P}(t) \leq \mathfrak{P}_{opt} \cdot t + \left\lceil \frac{t}{d} \right\rceil \sum_{u=1}^{p} \mathfrak{P}_u^{(2)}$$

Asympotic optimality (1/2)

- Algorithm look at the energy consumed during d time-units (d defined later)
- $\mathcal{A}$ is composed of $\mathcal{B}$ tasks
- optimal scheduling time: $\mathcal{T} = \frac{\mathcal{B}}{\rho}$, where ρ is the throughput bound.
- $\mathfrak{P}_{opt}$: the optimal power consumption that would be obtained in the ideal model
- $\mathfrak{P}^*$: the optimal power consumption that can be achieve under the model with start-up overheads
- $\mathfrak{P}$ the power consumption given by our algorithm.
- $\mathfrak{P}_{opt} \leq \mathfrak{P}^* \leq \mathfrak{P}$.
- during t time-units:

$$\mathfrak{P}(t) \leq \mathfrak{P}_{opt} \cdot t + \left\lceil \frac{t}{d} \right\rceil \sum_{u=1}^{p} \mathfrak{P}_u^{(2)}$$

Asympotic optimality (2/2)

If we fix $d = \sqrt{\mathcal{T}}$, we have

$$\mathfrak{P}(\mathcal{T}) \leq \mathfrak{P}^* \cdot \mathcal{T} + \left(1 + \sqrt{\mathcal{T}}\right) \cdot p \cdot \max_{u=1}^{p} \left\{\mathfrak{P}_u^{(2)}\right\}.$$

Then, when comparing $\mathfrak{P}$ and $\mathfrak{P}^*$ during the scheduling of the $\mathcal{B}$ tasks of application $\mathcal{A}$, we obtain:

$$\begin{aligned} \frac{\mathfrak{P}(\mathcal{T})}{\mathfrak{P}^*(\mathcal{T})} &\leq 1 + \left(\frac{1}{\mathcal{T}} + \frac{1}{\sqrt{\mathcal{T}}}\right) \frac{p \cdot \max_{u=1}^{p} \left\{\mathfrak{P}_u^{(2)}\right\}}{\mathfrak{P}^*} \\ &\leq 1 + \mathcal{O}\left(\frac{1}{\sqrt{\mathcal{T}}}\right). \end{aligned}$$

Outline

Conclusion

Contributions:

- scheduling a single application with power consumption constraints
- new closed-form relations between the throughput and the power consumption at the processor level
- optimal bi-criteria algorithm under the ideal power consumption model.
- asymptotically optimal algorithm under the model with start-up overheads

Perspectives:

- add memory constraints
- more realistic model with time overhead when turning on the worker.

Conclusion

Contributions:

- scheduling a single application with power consumption constraints
- new closed-form relations between the throughput and the power consumption at the processor level
- optimal bi-criteria algorithm under the ideal power consumption model.
- asymptotically optimal algorithm under the model with start-up overheads

Perspectives:

- add memory constraints
- more realistic model with time overhead when turning on the worker.

Conclusion

Contributions:

- scheduling a single application with power consumption constraints
- new closed-form relations between the throughput and the power consumption at the processor level
- optimal bi-criteria algorithm under the ideal power consumption model.
- asymptotically optimal algorithm under the model with start-up overheads

Perspectives:

- add memory constraints
- more realistic model with time overhead when turning on the worker.

Conclusion

Contributions:

- scheduling a single application with power consumption constraints
- new closed-form relations between the throughput and the power consumption at the processor level
- optimal bi-criteria algorithm under the ideal power consumption model.
- asymptotically optimal algorithm under the model with start-up overheads

Perspectives:

- add memory constraints
- more realistic model with time overhead when turning on the worker.

Conclusion

Contributions:

- scheduling a single application with power consumption constraints
- new closed-form relations between the throughput and the power consumption at the processor level
- optimal bi-criteria algorithm under the ideal power consumption model.
- asymptotically optimal algorithm under the model with start-up overheads

Perspectives:

- add memory constraints
- more realistic model with time overhead when turning on the worker.

Conclusion

Contributions:

- scheduling a single application with power consumption constraints
- new closed-form relations between the throughput and the power consumption at the processor level
- optimal bi-criteria algorithm under the ideal power consumption model.
- asymptotically optimal algorithm under the model with start-up overheads

Perspectives:

- add memory constraints
- more realistic model with time overhead when turning on the worker.