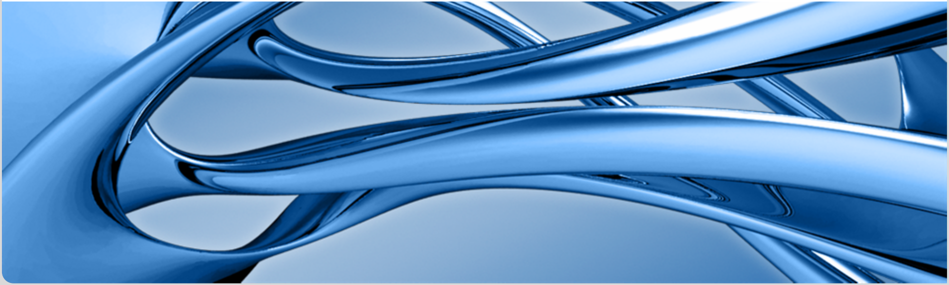
Karlsruhe Institute of Technology

Energy Efficient Task Partitioning based on the Single Frequency Approximation Scheme

Santiago Pagani and Jian-Jia Chen

2013 IEEE 34th Real-Time Systems Symposium (RTSS)
December 3-6 2013, Vancouver, Canada

KARLSRUHE INSTITUTE OF TECHNOLOGY (KIT)



Outline

- Introduction
- Motivation and Problem Definition
- Task Partitioning Scheme: Double-Largest-Task-First (DLTF)
- Approximation Factor Analysis (energy consumption): DLTF and SFA
  - Negligible Leakage Power Consumption
  - Non-negligible Leakage Power Consumption
  - Non-negligible Sleeping Overhead
- Simulations
- Conclusions

Outline

- Introduction
- Motivation and Problem Definition
- Task Partitioning Scheme: Double-Largest-Task-First (DLTF)
- Approximation Factor Analysis (energy consumption): DLTF and SFA
  - Negligible Leakage Power Consumption
  - Non-negligible Leakage Power Consumption
  - Non-negligible Sleeping Overhead
- Simulations
- Conclusions

Introduction

Karlsruhe Institute of Technology

Importance of Energy Efficiency:

- Slow increases of battery capacity.
  - Less Energy Consumption $\Rightarrow$ Prolong Battery Lifetime of Embedded Systems.
- Increasing costs of energy.
  - Less Energy Consumption $\Rightarrow$ Lower Power Bills for Servers.

Outcome for Computing Systems:

- Motivated to move from single-core to multi-core.
- Techniques for power management.

Introduction

Importance of Energy Efficiency:

- Slow increases of battery capacity.
  - Less Energy Consumption $\Rightarrow$ Prolong Battery Lifetime of Embedded Systems.
- Increasing costs of energy.
  - Less Energy Consumption $\Rightarrow$ Lower Power Bills for Servers.

Outcome for Computing Systems:

- Motivated to move from single-core to multi-core.
- Techniques for power management.

Introduction

Karlsruhe Institute of Technology

Dynamic Power Management (DPM):

- Technique for putting cores in a low-power mode: idle, sleep, off, etc.

Dynamic Voltage and Frequency Scaling (DVFS):

- Technique for scaling the voltage and frequency of cores.
- Per-core DVFS:
  - Individual voltage and frequency for cores.
  - Optimal, but too expensive to manufacture.
- Global DVFS:
  - All cores share the same voltage.
  - Energy inefficient.

Introduction

Dynamic Power Management (DPM):

- Technique for putting cores in a low-power mode: idle, sleep, off, etc.

Dynamic Voltage and Frequency Scaling (DVFS):

- Technique for scaling the voltage and frequency of cores.
- Per-core DVFS:
  - Individual voltage and frequency for cores.
  - Optimal, but too expensive to manufacture.
- Global DVFS:
  - All cores share the same voltage.
  - Energy inefficient.

Introduction

Karlsruhe Institute of Technology

Dynamic Power Management (DPM):

- Technique for putting cores in a low-power mode: idle, sleep, off, etc.

Dynamic Voltage and Frequency Scaling (DVFS):

- Technique for scaling the voltage and frequency of cores.
- Per-core DVFS:
  - Individual voltage and frequency for cores.
  - Optimal, but too expensive to manufacture.
- Global DVFS:
  - All cores share the same voltage.
  - Energy inefficient.

Introduction

Karlsruhe Institute of Technology

Dynamic Power Management (DPM):

- Technique for putting cores in a low-power mode: idle, sleep, off, etc.

Dynamic Voltage and Frequency Scaling (DVFS):

- Technique for scaling the voltage and frequency of cores.
- Per-core DVFS:
  - Individual voltage and frequency for cores.
  - Optimal, but too expensive to manufacture.
- Global DVFS:
  - All cores share the same voltage.
  - Energy inefficient.

Introduction

Dynamic Voltage and Frequency Scaling (DVFS):

- Multiple Voltage Islands:
  - Compromise between *Per-core DVFS* and *Global DVFS*.
  - Cores are grouped into *Voltage Islands*.
  - Islands can have different voltages.

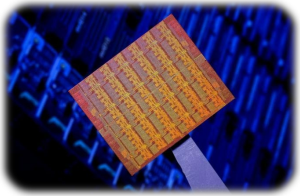


Figure: Intel's SCC snapshot

Intel Corporation. *Single-chip Cloud Computer (SCC)*. URL:
`http://www.intel.com/content/www/us/en/research/intel-labs-single-chip-cloud-computer.html`

Introduction

CMOS-core Power Model

$$P(s) = P_{\text{dynamic}}(s) + P_{\text{static}}$$

Considering that:

$$P_{\text{dynamic}}(s) = C_{\text{eff}} V_{dd}^2 s$$

$$s \propto \frac{(V_{dd} - V_t)^2}{V_{dd}}$$

We can approximate to:

$$P(s) = \alpha s^\gamma + \beta$$

Introduction

CMOS-core Power Model

$$P(s) = P_{\text{dynamic}}(s) + P_{\text{static}}$$

Considering that:

$$P_{\text{dynamic}}(s) = C_{\text{eff}} V_{dd}^2 s$$

$$s \propto \frac{(V_{dd} - V_t)^2}{V_{dd}}$$

We can approximate to:

$$P(s) = \alpha s^{\gamma} + \beta$$

Introduction

CMOS-core Power Model

$$P(s) = P_{\text{dynamic}}(s) + P_{\text{static}}$$

Considering that:

$$P_{\text{dynamic}}(s) = C_{\text{eff}} V_{dd}^2 s$$

$$s \propto \frac{(V_{dd} - V_t)^2}{V_{dd}}$$

We can approximate to:

$$P(s) = \alpha s^{\gamma} + \beta$$

Introduction

CMOS-core Power Model

$$P(s) = P_{\text{dynamic}}(s) + P_{\text{static}}$$

Considering that:

$$P_{\text{dynamic}}(s) = C_{\text{eff}} V_{dd}^2 s$$

$$s \propto \frac{(V_{dd} - V_t)^2}{V_{dd}}$$

We can approximate to:

$$P(s) = \alpha s^{\gamma} + \beta$$

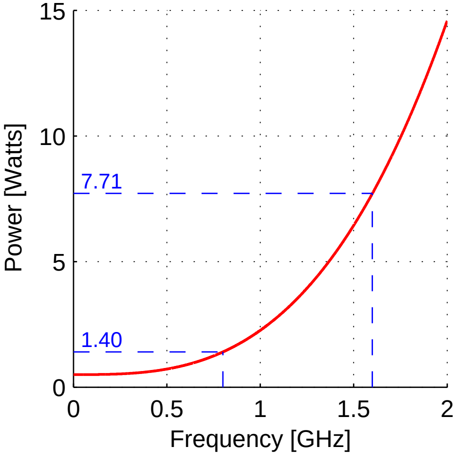


Figure: $\alpha = 1.76 \frac{\text{Watts}}{\text{GHz}^3}$, $\gamma = 3$ and $\beta = 0.5$ Watts

Introduction

Energy Consumption

$$E(s) = (\alpha s^{\gamma} + \beta)\frac{\Delta c}{s}$$

Critical Frequency:

$$s_{\text{crit}} = \sqrt[\gamma]{\frac{\beta}{(\gamma - 1)\,\alpha}}$$

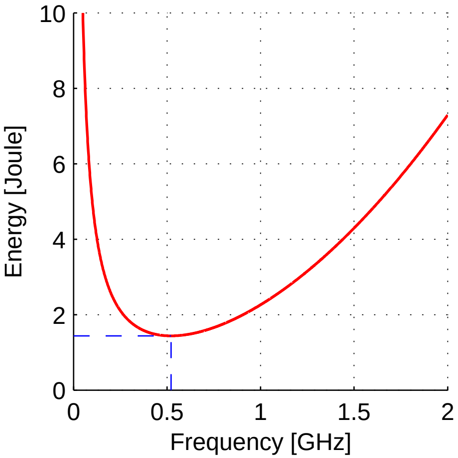


Figure: $\alpha = 1.76\,\frac{\text{Watts}}{\text{GHz}^3}$, $\gamma = 3$, $\beta = 0.5$ Watts and $\Delta c = 10^9$ cycles

Outline

- Introduction
- Motivation and Problem Definition
- Task Partitioning Scheme: Double-Largest-Task-First (DLTF)
- Approximation Factor Analysis (energy consumption): DLTF and SFA
  - Negligible Leakage Power Consumption
  - Non-negligible Leakage Power Consumption
  - Non-negligible Sleeping Overhead
- Simulations
- Conclusions

Motivation

In each voltage island (or Global DVFS), for energy minimization:

Task partitioning and DVFS schedule:
Play a major role in energy minimization.

Task Partitioning:

- Convexity of $E(s)$: For the same workload $\Rightarrow 2 \cdot E(s) \leq E(2 \cdot s)$.
- In general, load balancing reduces the dynamic energy consumption.
- Balanced solution: with high complexity and not always feasible.
- Good option: polynomial time algorithm based on load balancing, e.g., *Largest-Task-First* (LTF)[1] strategy.

---

[1] Chuan-Yue Yang, Jian-Jia Chen, and Tei-Wei Kuo. "An Approximation Algorithm for Energy-Efficient Scheduling on A Chip Multiprocessor". In: *Conference on Design, Automation, and Test in Europe (DATE)*. 2005, pp. 468–473

Motivation

In each voltage island (or Global DVFS), for energy minimization:

Task partitioning and DVFS schedule:
Play a major role in energy minimization.

Task Partitioning:

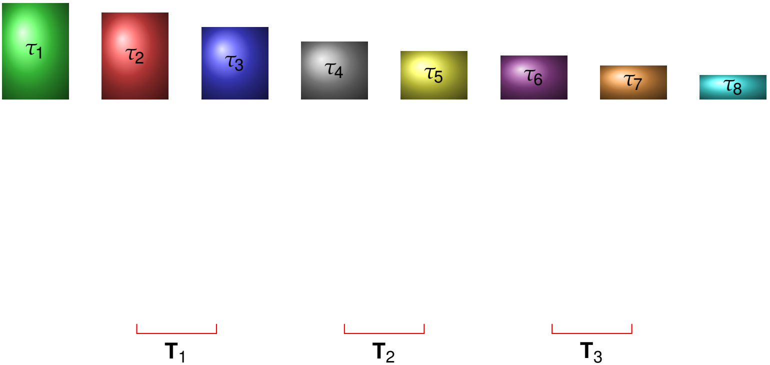
- Convexity of $E(s)$: For the same workload $\Rightarrow 2 \cdot E(s) \leq E(2 \cdot s)$.
- In general, load balancing reduces the dynamic energy consumption.
- Balanced solution: with high complexity and not always feasible.
- Good option: polynomial time algorithm based on load balancing, e.g., *Largest-Task-First* (LTF)[1] strategy.

[1] Chuan-Yue Yang, Jian-Jia Chen, and Tei-Wei Kuo. "An Approximation Algorithm for Energy-Efficient Scheduling on A Chip Multiprocessor". In: *Conference on Design, Automation, and Test in Europe (DATE)*. 2005, pp. 468–473

Motivation

Largest-Task-First (LTF) Strategy Example:

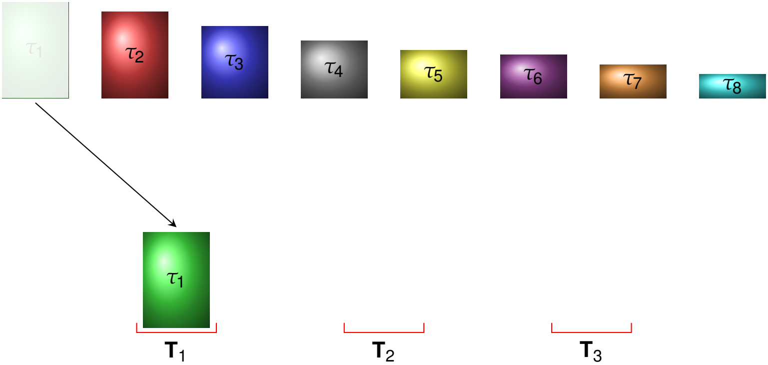
- 8 tasks: $\{\tau_1, \tau_2, \ldots, \tau_8\}$
- Partitioned into 3 task sets: $\{\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3\}$



Motivation

Largest-Task-First (LTF) Strategy Example:

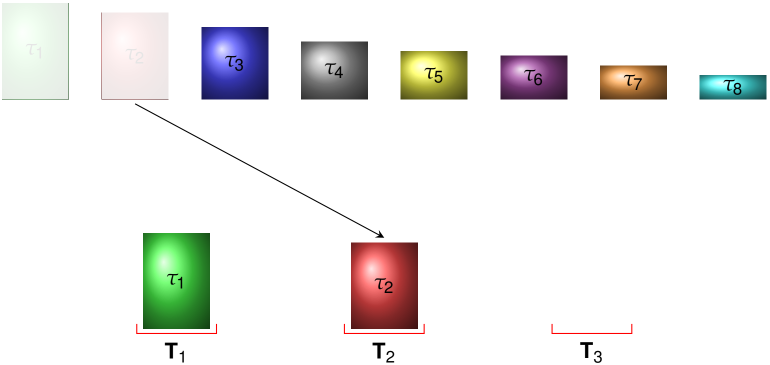
- 8 tasks: $\{\tau_1, \tau_2, \ldots, \tau_8\}$
- Partitioned into 3 task sets: $\{\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3\}$



Motivation

Largest-Task-First (LTF) Strategy Example:

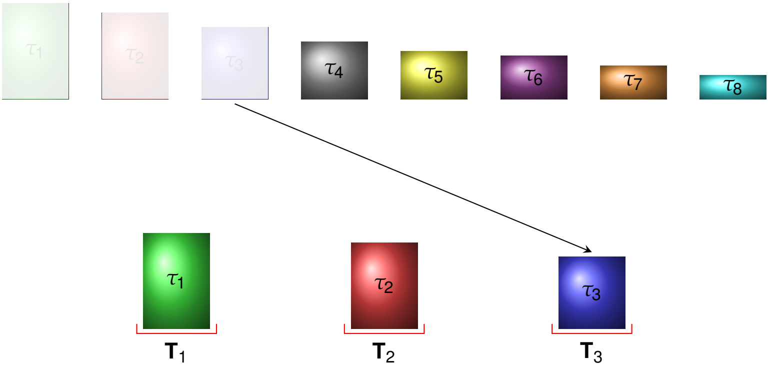
- 8 tasks: $\{\tau_1, \tau_2, \ldots, \tau_8\}$
- Partitioned into 3 task sets: $\{\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3\}$



Motivation

Largest-Task-First (LTF) Strategy Example:

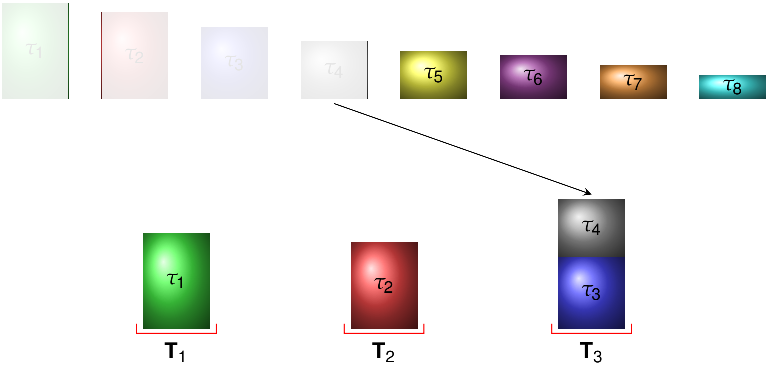
- 8 tasks: $\{\tau_1, \tau_2, \ldots, \tau_8\}$
- Partitioned into 3 task sets: $\{\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3\}$



Motivation

Largest-Task-First (LTF) Strategy Example:

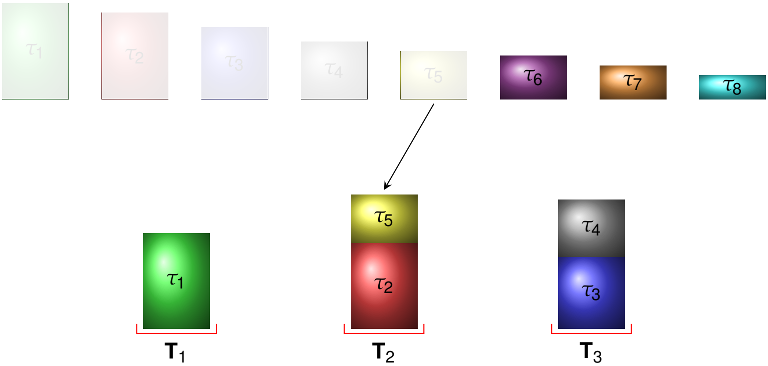
- 8 tasks: $\{\tau_1, \tau_2, \ldots, \tau_8\}$
- Partitioned into 3 task sets: $\{\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3\}$



Motivation

Largest-Task-First (LTF) Strategy Example:

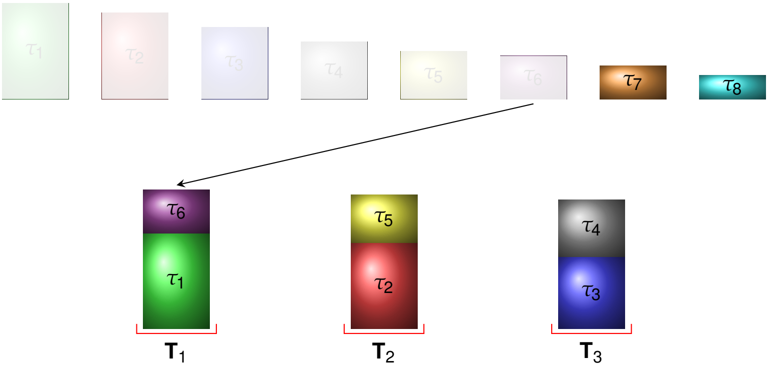
- 8 tasks: $\{\tau_1, \tau_2, \ldots, \tau_8\}$
- Partitioned into 3 task sets: $\{\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3\}$



Motivation

Largest-Task-First (LTF) Strategy Example:

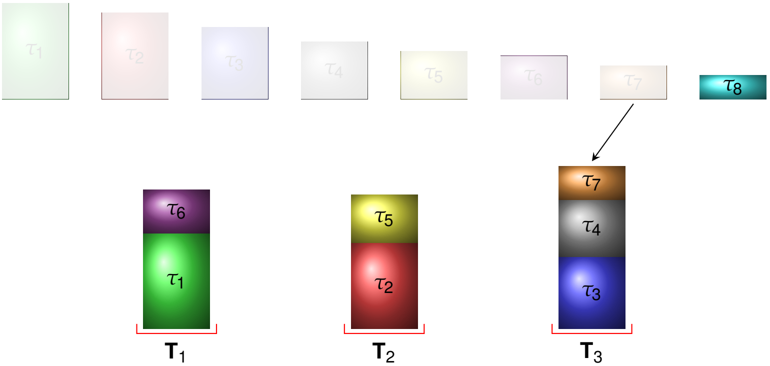
- 8 tasks: $\{\tau_1, \tau_2, \ldots, \tau_8\}$
- Partitioned into 3 task sets: $\{\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3\}$



Motivation

Largest-Task-First (LTF) Strategy Example:

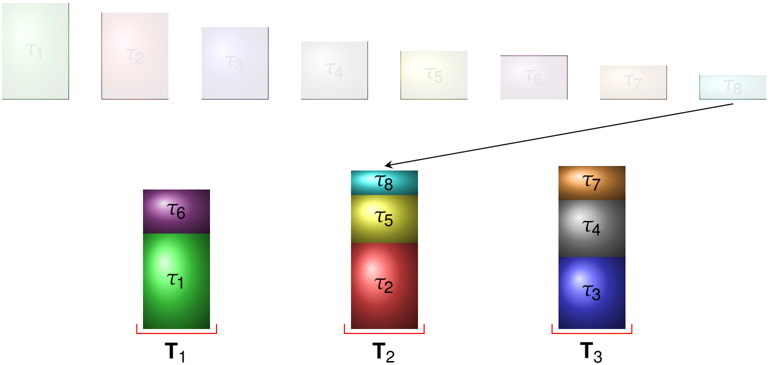
- 8 tasks: $\{\tau_1, \tau_2, \ldots, \tau_8\}$
- Partitioned into 3 task sets: $\{\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3\}$



Motivation

Largest-Task-First (LTF) Strategy Example:

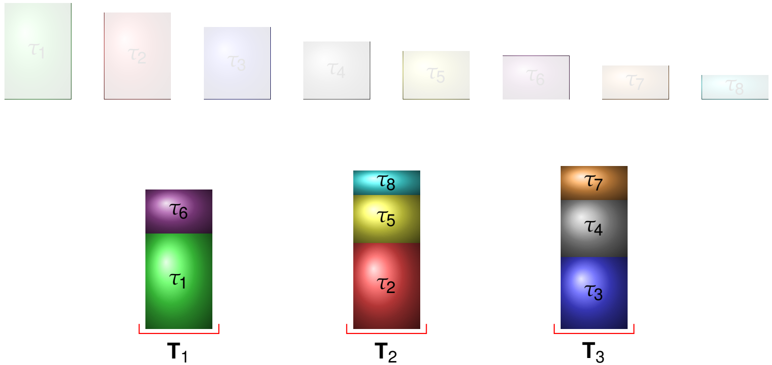
- 8 tasks: $\{\tau_1, \tau_2, \ldots, \tau_8\}$
- Partitioned into 3 task sets: $\{\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3\}$



Motivation

Largest-Task-First (LTF) Strategy Example:

- 8 tasks: $\{\tau_1, \tau_2, \ldots, \tau_8\}$
- Partitioned into 3 task sets: $\{\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3\}$



Motivation

DVFS Schedule → Single Frequency Approximation (SFA)[2] Scheme:

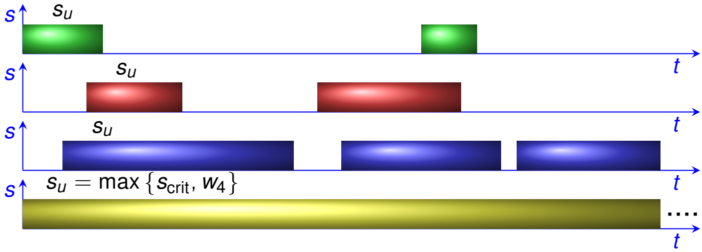
- Use the lowest voltage/frequency, satisfying the timing constraints.
- Linear time complexity. Is the *simplest* and *most intuitive* strategy.
- Not optimal, but significantly reduces the management overhead.
- No frequency alignment between cores ⟹ Any uni-core DPM technique can be adopted individually in each core.

Core 1:

Core 2:

Core 3:

Core 4:



[2] Santiago Pagani and Jian-Jia Chen. "Energy Efficiency Analysis for the Single Frequency Approximation (SFA) Scheme". In: *Proceedings of the 19th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. 2013

Motivation

Karlsruhe Institute of Technology

Combining LTF and SFA for energy minimization:

- Is a practical solution.
- Very easy to implement.

What is the worst-case performance in terms of energy efficiency?

Problem Definition

- For periodic real-time tasks, assigned on a voltage island.
- Using *Earliest-Deadline-First* (EDF) on individual cores.

Contributions

- Present a simple and practical solution for energy minimization.
  - Task Partitioning: *Double-Largest-Task-First* (DLTF).
  - DVFS schedule: SFA.
- Theoretically analyse such a solution:

$$\text{AF}_{\text{SFA}}^{\text{DLTF}} = \max \frac{E_{\text{SFA}}^{\text{DLTF}}}{E_{\text{OPT}}^{*}} \leq \max \frac{E_{\text{SFA}}^{\text{DLTF}}}{E_{\downarrow}^{*}}$$

Problem Definition

- For periodic real-time tasks, assigned on a voltage island.
- Using *Earliest-Deadline-First* (EDF) on individual cores.

Contributions

- Present a simple and practical solution for energy minimization.
  - Task Partitioning: *Double-Largest-Task-First* (DLTF).
  - DVFS schedule: SFA.
- Theoretically analyse such a solution:

$$\mathrm{AF}_{\mathrm{SFA}}^{\mathrm{DLTF}} = \max \frac{E_{\mathrm{SFA}}^{\mathrm{DLTF}}}{E_{\mathrm{OPT}}^{*}} \leq \max \frac{E_{\mathrm{SFA}}^{\mathrm{DLTF}}}{E_{\downarrow}^{*}}$$

Problem Definition

- For periodic real-time tasks, assigned on a voltage island.
- Using *Earliest-Deadline-First* (EDF) on individual cores.

Contributions

- Present a simple and practical solution for energy minimization.
  - Task Partitioning: *Double-Largest-Task-First* (DLTF).
  - DVFS schedule: SFA.
- Theoretically analyse such a solution:

$$\mathrm{AF}_{\mathrm{SFA}}^{\mathrm{DLTF}} = \max \frac{E_{\mathrm{SFA}}^{\mathrm{DLTF}}}{E_{\mathrm{OPT}}^{*}} \leq \max \frac{E_{\mathrm{SFA}}^{\mathrm{DLTF}}}{E_{\downarrow}^{*}}$$

Outline

- Introduction
- Motivation and Problem Definition
- Task Partitioning Scheme: Double-Largest-Task-First (DLTF)
- Approximation Factor Analysis (energy consumption): DLTF and SFA
  - Negligible Leakage Power Consumption
  - Non-negligible Leakage Power Consumption
  - Non-negligible Sleeping Overhead
- Simulations
- Conclusions

Double-Largest-Task-First (DLTF)

- Initial solution: Task partitioning with LTF.
- Tasks are regrouped, shutting down all possible cores.
- This reduces the energy consumption for idling under SFA:

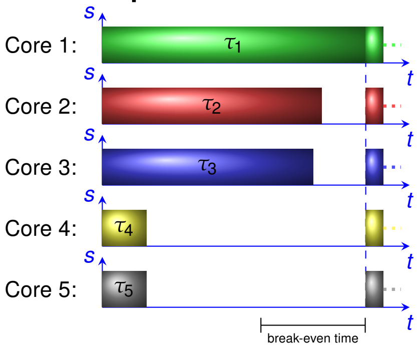
Example: LTF and SFA

Example: DLTF and SFA

Double-Largest-Task-First (DLTF)

- Initial solution: Task partitioning with LTF.
- Tasks are regrouped, shutting down all possible cores.
- This reduces the energy consumption for idling under SFA:

Example: LTF and SFA

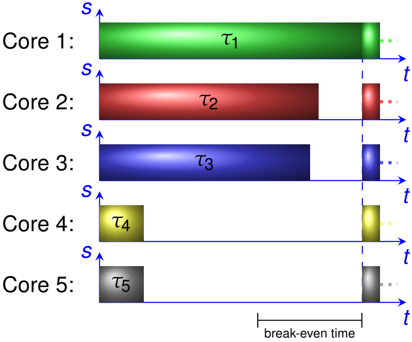


Example: DLTF and SFA

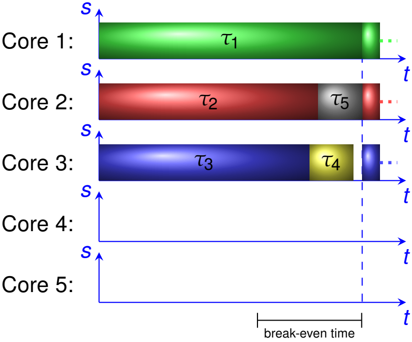
Double-Largest-Task-First (DLTF)

- Initial solution: Task partitioning with LTF.
- Tasks are regrouped, shutting down all possible cores.
- This reduces the energy consumption for idling under SFA:

Example: LTF and SFA



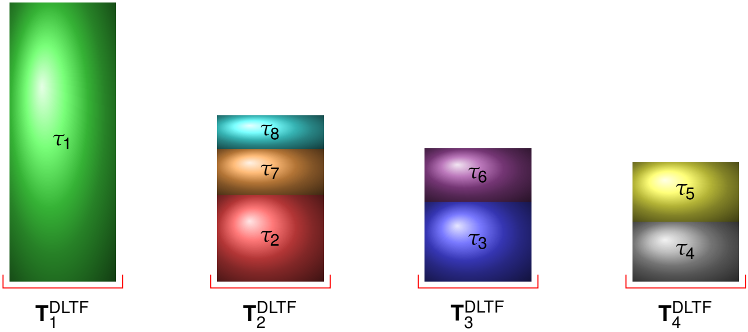
Example: DLTF and SFA



Double-Largest-Task-First (DLTF)

Example:

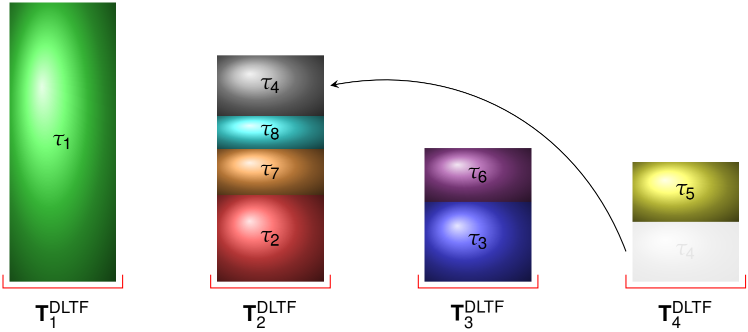
- 8 tasks: $\{\tau_1, \tau_2, \ldots, \tau_8\}$
- Partitioned into 4 task sets: $\{\mathbf{T}_1^{\mathrm{DLTF}}, \mathbf{T}_2^{\mathrm{DLTF}}, \ldots, \mathbf{T}_4^{\mathrm{DLTF}}\}$



Double-Largest-Task-First (DLTF)

Example:

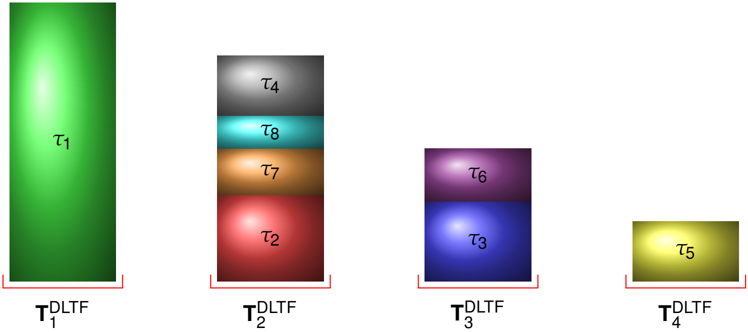
- 8 tasks: $\{\tau_1, \tau_2, \ldots, \tau_8\}$
- Partitioned into 4 task sets: $\{\mathbf{T}_1^{\text{DLTF}}, \mathbf{T}_2^{\text{DLTF}}, \ldots, \mathbf{T}_4^{\text{DLTF}}\}$



Double-Largest-Task-First (DLTF)

Example:

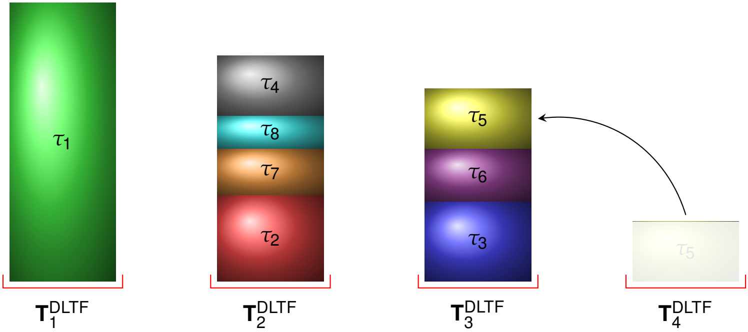
- 8 tasks: $\{\tau_1, \tau_2, \ldots, \tau_8\}$
- Partitioned into 4 task sets: $\{\mathbf{T}_1^{\mathrm{DLTF}}, \mathbf{T}_2^{\mathrm{DLTF}}, \ldots, \mathbf{T}_4^{\mathrm{DLTF}}\}$



Double-Largest-Task-First (DLTF)

Example:

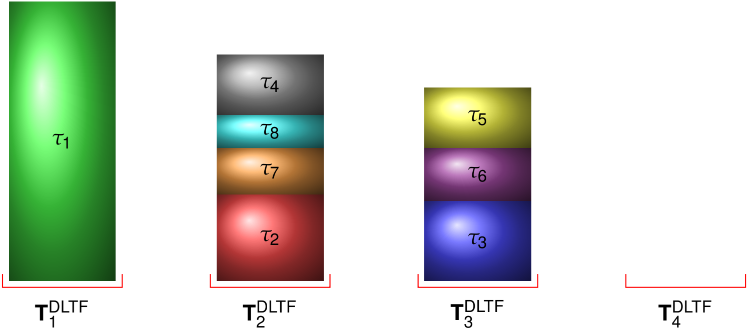
- 8 tasks: $\{\tau_1, \tau_2, \ldots, \tau_8\}$
- Partitioned into 4 task sets: $\{\mathbf{T}_1^{\text{DLTF}}, \mathbf{T}_2^{\text{DLTF}}, \ldots, \mathbf{T}_4^{\text{DLTF}}\}$



Double-Largest-Task-First (DLTF)

Example:

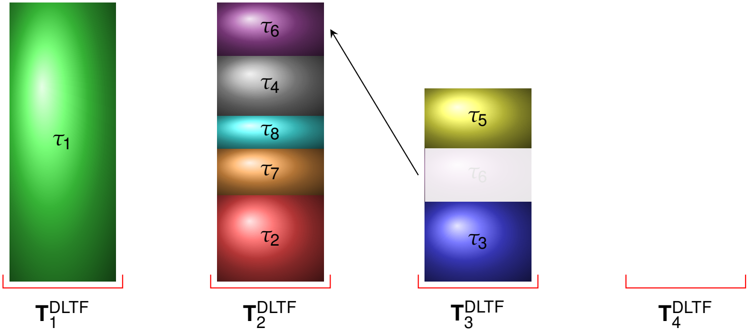
- 8 tasks: $\{\tau_1, \tau_2, \ldots, \tau_8\}$
- Partitioned into 4 task sets: $\{\mathbf{T}_1^{\text{DLTF}}, \mathbf{T}_2^{\text{DLTF}}, \ldots, \mathbf{T}_4^{\text{DLTF}}\}$



Double-Largest-Task-First (DLTF)

Example:

- 8 tasks: $\{\tau_1, \tau_2, \ldots, \tau_8\}$
- Partitioned into 4 task sets: $\{\mathbf{T}_1^{\text{DLTF}}, \mathbf{T}_2^{\text{DLTF}}, \ldots, \mathbf{T}_4^{\text{DLTF}}\}$

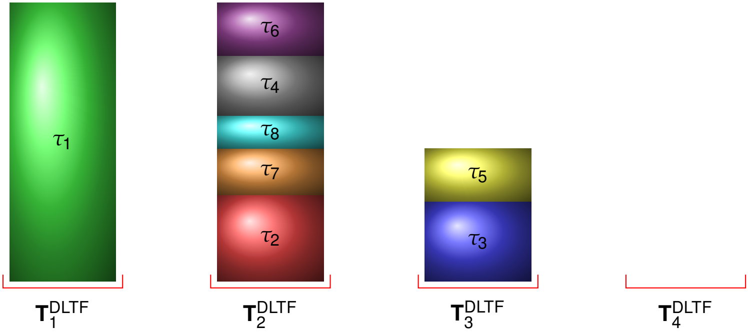


Double-Largest-Task-First (DLTF)

Karlsruhe Institute of Technology

Example:

- 8 tasks: $\{\tau_1, \tau_2, \ldots, \tau_8\}$
- Partitioned into 4 task sets: $\{\mathbf{T}_1^{\mathrm{DLTF}}, \mathbf{T}_2^{\mathrm{DLTF}}, \ldots, \mathbf{T}_4^{\mathrm{DLTF}}\}$



Properties of Double-Largest-Task-First (DLTF)

If $w_M^{\text{LTF}} < s_{\text{crit}}$, then:

- $w_1^{\text{DLTF}} \leq w_1^{\text{DLTF}} \leq \cdots \leq w_M^{\text{DLTF}} < s_{\text{crit}}$
- All cores run at the critical frequency: $s_u = s_{\text{crit}}$
- The energy consumption for execution is minimized.
- $E_{\text{SFA}}^{\text{DLTF}} = E_{\text{OPT}}^{*} \quad \Rightarrow \quad \text{AF}_{\text{SFA}}^{\text{DLTF}} = 1$

If $w_M^{\text{LTF}} \geq s_{\text{crit}}$, then:

- $w_M^{\text{DLTF}} = w_M^{\text{LTF}}$
- If there is *only one* task in $\mathbf{T}_M^{\text{DLTF}} \Rightarrow \quad w_M^{\text{DLTF}} \leq w_M^{*}$
- If there are *at least two* tasks in $\mathbf{T}_M^{\text{DLTF}} \Rightarrow \quad \frac{w_M^{\text{DLTF}}}{w_M^{*}} \leq \theta_{\text{LTF}} = \frac{4}{3} - \frac{1}{3M}$

Properties of Double-Largest-Task-First (DLTF)

Karlsruhe Institute of Technology

If $w_M^{\mathrm{LTF}} < s_{\mathrm{crit}}$, then:

- $w_1^{\mathrm{DLTF}} \leq w_1^{\mathrm{DLTF}} \leq \cdots \leq w_M^{\mathrm{DLTF}} < s_{\mathrm{crit}}$
- All cores run at the critical frequency: $s_u = s_{\mathrm{crit}}$
- The energy consumption for execution is minimized.
- $E_{\mathrm{SFA}}^{\mathrm{DLTF}} = E_{\mathrm{OPT}}^{*} \quad \Rightarrow \quad \mathrm{AF}_{\mathrm{SFA}}^{\mathrm{DLTF}} = 1$

If $w_M^{\mathrm{LTF}} \geq s_{\mathrm{crit}}$, then:

- $w_M^{\mathrm{DLTF}} = w_M^{\mathrm{LTF}}$
- If there is *only one* task in $\mathbf{T}_M^{\mathrm{DLTF}} \Rightarrow \quad w_M^{\mathrm{DLTF}} \leq w_M^{*}$
- If there are *at least two* tasks in $\mathbf{T}_M^{\mathrm{DLTF}} \Rightarrow \quad \frac{w_M^{\mathrm{DLTF}}}{w_M^{*}} \leq \theta_{\mathrm{LTF}} = \frac{4}{3} - \frac{1}{3M}$

Outline

- Introduction
- Motivation and Problem Definition
- Task Partitioning Scheme: Double-Largest-Task-First (DLTF)
- Approximation Factor Analysis (energy consumption): DLTF and SFA
  - Negligible Leakage Power Consumption
  - Non-negligible Leakage Power Consumption
  - Non-negligible Sleeping Overhead
- Simulations
- Conclusions

Negligible Leakage Power Consumption

Karlsruhe Institute of Technology

Energy Consumption for DLTF and SFA (when $\beta = 0$):

$$E_{\text{SFA}}^{\text{DLTF}} = P(s_u)\frac{L}{s_u}\sum_{i=1}^{M} w_i^{\text{DLTF}} \qquad \Rightarrow \qquad E_{\text{SFA}}^{\text{DLTF}} = \alpha L \left({w_M^{\text{DLTF}}}^{\gamma-1} \right) \sum_{i=1}^{M} w_i^{*}$$

Lower Bound Energy Consumption (when $\beta = 0$):

- Unroll periodic tasks in a hyper-period $\Rightarrow$ frame-based tasks.
- Use the results from the SFA analysis paper [3]:

$$E_{\downarrow}^{*(\beta=0)} = \alpha L \left[\sum_{i=1}^{M} \left(w_i^{*} - w_{i-1}^{*} \right) \sqrt[\gamma]{M-i+1} \right]^{\gamma}$$

[3] Santiago Pagani and Jian-Jia Chen. "Energy Efficiency Analysis for the Single Frequency Approximation (SFA) Scheme". In: *Proceedings of the 19th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA).* 2013

Negligible Leakage Power Consumption

Karlsruhe Institute of Technology

Energy Consumption for DLTF and SFA (when $\beta = 0$):

$$E_{\text{SFA}}^{\text{DLTF}} = P\left(s_u\right)\frac{L}{s_u}\sum_{i=1}^{M} w_i^{\text{DLTF}} \qquad \Rightarrow \qquad E_{\text{SFA}}^{\text{DLTF}} = \alpha L\left({w_M^{\text{DLTF}}}^{\gamma-1}\right)\sum_{i=1}^{M} w_i^*$$

Lower Bound Energy Consumption (when $\beta = 0$):

- Unroll periodic tasks in a hyper-period $\Rightarrow$ frame-based tasks.
- Use the results from the SFA analysis paper [3]:

$$E_{\downarrow}^{*(\beta=0)} = \alpha L\left[\sum_{i=1}^{M}\left(w_i^* - w_{i-1}^*\right)\sqrt[\gamma]{M-i+1}\right]^{\gamma}$$

[3] Santiago Pagani and Jian-Jia Chen. “Energy Efficiency Analysis for the Single Frequency Approximation (SFA) Scheme”. In: *Proceedings of the 19th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. 2013

Negligible Leakage Power Consumption

Approximation Factor for DLTF and SFA (when $\beta = 0$):

$$AF_{\text{SFA}}^{\text{DLTF}(\beta=0)} \leq \max \left\{ \frac{\left(w_M^{\text{DLTF}}\right)^{\gamma-1} \sum_{i=1}^{M} w_i^*}{\left[\sum_{i=1}^{M} \left(w_i^* - w_{i-1}^*\right) \sqrt[\gamma]{M-i+1}\right]^{\gamma}} \right\}$$

Critical Cycle Utilization Distribution: Minimizes $E_{\downarrow}^*$, for a fixed $\sum_{i=1}^{M} w_i^*$

- $w'_1 = w'_2 = \cdots = w'_{M-1} = \text{Average}\left(w_1^*, w_2^*, \ldots, w_{M-1}^*\right)$
- Utilization Ratio: $0 \leq \delta = \frac{\text{Average}\left(w_1^*, w_2^*, \ldots, w_{M-1}^*\right)}{w_M^*} \leq 1$

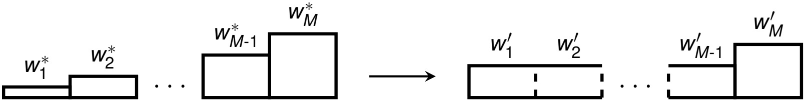
Negligible Leakage Power Consumption

Karlsruhe Institute of Technology

Approximation Factor for DLTF and SFA (when $\beta = 0$):

$$\mathrm{AF}_{\mathrm{SFA}}^{\mathrm{DLTF}(\beta=0)} \leq \max \left\{ \frac{\left(w_M^{\mathrm{DLTF}}\right)^{\gamma-1} \sum_{i=1}^{M} w_i^*}{\left[\sum_{i=1}^{M} \left(w_i^* - w_{i-1}^*\right) \sqrt[\gamma]{M-i+1}\right]^{\gamma}} \right\}$$

Critical Cycle Utilization Distribution: Minimizes $E_{\downarrow}^*$, for a fixed $\sum_{i=1}^{M} w_i^*$



- $w'_1 = w'_2 = \cdots = w'_{M-1} = \text{Average}\left(w_1^*, w_2^*, \ldots, w_{M-1}^*\right)$
- Utilization Ratio: $0 \leq \delta = \frac{\text{Average}\left(w_1^*, w_2^*, \ldots, w_{M-1}^*\right)}{w_M^*} \leq 1$

Negligible Leakage Power Consumption

Karlsruhe Institute of Technology

Approximation Factor for DLTF and SFA (when $\beta = 0$):

$$\mathrm{AF}_{\mathrm{SFA}}^{\mathrm{DLTF}(\beta=0)} \leq \max\left\{ \left(\frac{w_M^{\mathrm{DLTF}}}{w_M^*} \right)^{\gamma-1} \cdot h(\delta) \right\}$$

where

$$h(\delta) = \frac{1-\delta+\delta M}{\left(1-\delta+\delta\sqrt[\gamma]{M}\right)^{\gamma}} \leq h(\delta^{\max})$$

and

$$\delta^{\max} = \frac{\frac{1}{\gamma-1}\left(\gamma-1+M-\gamma\sqrt[\gamma]{M}\right)}{\left(M\sqrt[\gamma]{M}-M-\sqrt[\gamma]{M}+1\right)}$$

Negligible Leakage Power Consumption

Approximation Factor for DLTF and SFA (when $\beta = 0$):

$$\mathrm{AF}_{\mathrm{SFA}}^{\mathrm{DLTF}\,(\beta=0)} \leq \max\left\{\left(\frac{w_M^{\mathrm{DLTF}}}{w_M^*}\right)^{\gamma-1} \cdot h(\delta)\right\}$$

where

$$h(\delta) = \frac{1-\delta+\delta M}{\left(1-\delta+\delta\sqrt[\gamma]{M}\right)^{\gamma}} \leq h\left(\delta^{\max}\right)$$

and

$$\delta^{\max} = \frac{\frac{1}{\gamma-1}\left(\gamma-1+M-\gamma\sqrt[\gamma]{M}\right)}{\left(M\sqrt[\gamma]{M}-M-\sqrt[\gamma]{M}+1\right)}$$

Negligible Leakage Power Consumption

Approximation Factor for DLTF and SFA (when $\beta = 0$):

$$\mathrm{AF}_{\mathrm{SFA}}^{\mathrm{DLTF}(\beta=0)} \leq \max\left\{ \left(\frac{w_M^{\mathrm{DLTF}}}{w_M^*} \right)^{\gamma-1} \cdot h(\delta) \right\}$$

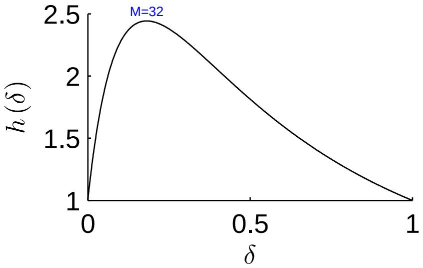
where

$$h(\delta) = \frac{1 - \delta + \delta M}{\left(1 - \delta + \delta \sqrt[\gamma]{M}\right)^{\gamma}} \leq h\left(\delta^{\max}\right)$$

and

$$\delta^{\max} = \frac{\frac{1}{\gamma-1}\left(\gamma - 1 + M - \gamma \sqrt[\gamma]{M}\right)}{\left(M \sqrt[\gamma]{M} - M - \sqrt[\gamma]{M} + 1\right)}$$

$h(\delta)$ for $\gamma = 3$:



Negligible Leakage Power Consumption

Karlsruhe Institute of Technology

Approximation Factor for DLTF and SFA (when $\beta = 0$):

$$\mathrm{AF}_{\mathrm{SFA}}^{\mathrm{DLTF}\,(\beta=0)} \leq \max \left\{ \left(\frac{w_M^{\mathrm{DLTF}}}{w_M^{*}} \right)^{\gamma - 1} \cdot h\left(\delta\right) \right\}$$

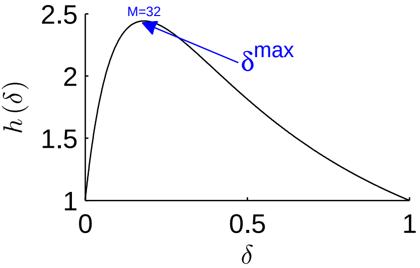
where

$$h\left(\delta\right) = \frac{1 - \delta + \delta M}{\left(1 - \delta + \delta \sqrt[\gamma]{M}\right)^{\gamma}} \leq h\left(\delta^{\max}\right)$$

and

$$\delta^{\max} = \frac{\frac{1}{\gamma - 1}\left(\gamma - 1 + M - \gamma \sqrt[\gamma]{M}\right)}{\left(M \sqrt[\gamma]{M} - M - \sqrt[\gamma]{M} + 1\right)}$$

$h\left(\delta\right)$ for $\gamma = 3$:



Negligible Leakage Power Consumption

Approximation Factor for DLTF and SFA (when $\beta = 0$):

$$\mathrm{AF}_{\mathrm{SFA}}^{\mathrm{DLTF}\,(\beta=0)} \leq \max\left\{ \left(\frac{w_M^{\mathrm{DLTF}}}{w_M^*} \right)^{\gamma-1} \cdot h(\delta) \right\}$$

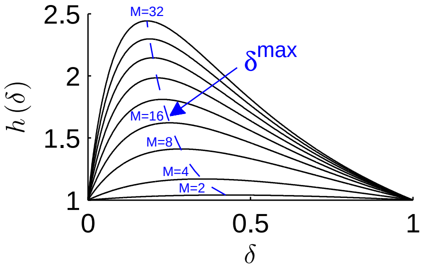
where

$$h(\delta) = \frac{1-\delta+\delta M}{\left(1-\delta+\delta\sqrt[\gamma]{M}\right)^{\gamma}} \leq h\left(\delta^{\max}\right)$$

and

$$\delta^{\max} = \frac{\frac{1}{\gamma-1}\left(\gamma-1+M-\gamma\sqrt[\gamma]{M}\right)}{\left(M\sqrt[\gamma]{M}-M-\sqrt[\gamma]{M}+1\right)}$$

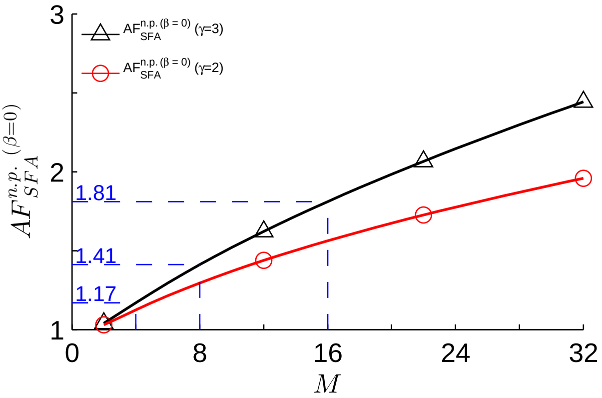
$h(\delta)$ for $\gamma = 3$:



Negligible Leakage Power Consumption

Approximation Factor for DLTF and SFA (when $\beta = 0$):

- If $w_M^* \geq w_M^{\text{DLTF}} \Rightarrow w'_M = w_M^{\text{DLTF}} \rightarrow$ Same AF as for fixed task sets



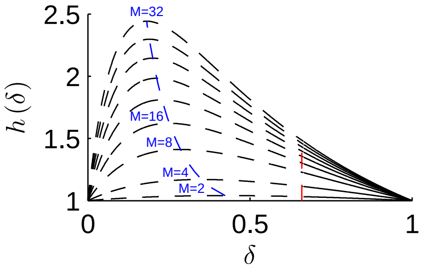
Note: $AF_{SFA}^{n.p.\,(\beta=0)}$ *only depends on the values of* γ *and* M*.*

Negligible Leakage Power Consumption

Approximation Factor for DLTF and SFA (when $\beta = 0$):

- If $w_M^* < w_M^{\text{DLTF}} \Rightarrow$

  If $\delta > \delta^{\max} \Rightarrow h(\delta) < h(\delta^{\max})$:



We prove that:

$$\delta \geq \frac{w_1'}{w_M'} \geq \frac{4M+1}{6M} \geq 0.66$$

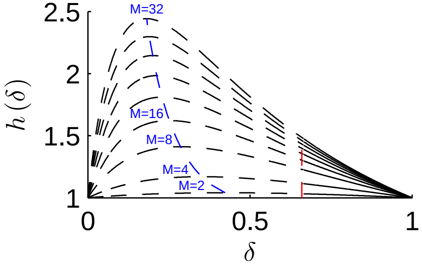
The AF for this case is $\leq \theta_{\text{LTF}}^{\gamma-1} \cdot h\left(\frac{4M+1}{6M}\right)$

Negligible Leakage Power Consumption

Approximation Factor for DLTF and SFA (when $\beta = 0$):

- If $w_M^* < w_M^{\text{DLTF}} \Rightarrow$

  If $\delta > \delta^{\max} \Rightarrow h(\delta) < h(\delta^{\max})$:



We prove that:

$$\delta \geq \frac{w_1'}{w_M'} \geq \frac{4M+1}{6M} \geq 0.66$$

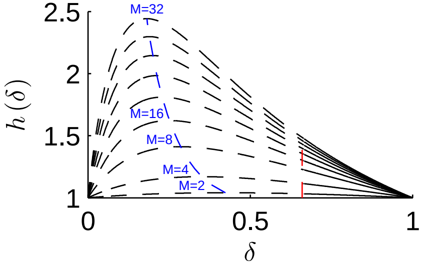
The AF for this case is $\leq \theta_{\text{LTF}}^{\gamma-1} \cdot h\left(\frac{4M+1}{6M}\right)$

Negligible Leakage Power Consumption

Approximation Factor for DLTF and SFA (when $\beta = 0$):

- If $w_M^* < w_M^{\mathrm{DLTF}} \Rightarrow$

  If $\delta > \delta^{\max} \Rightarrow h(\delta) < h(\delta^{\max})$:



We prove that:

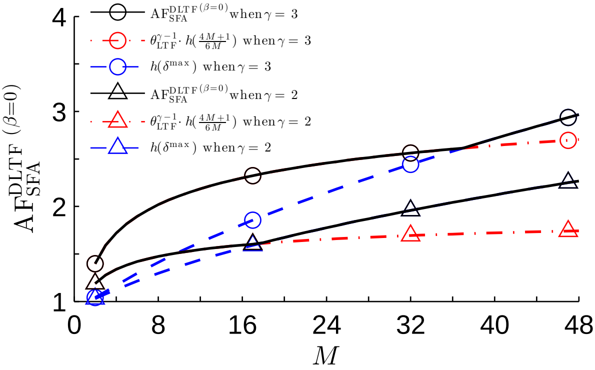
$$\delta \geq \frac{w_1'}{w_M'} \geq \frac{4M+1}{6M} \geq 0.66$$

The AF for this case is $\leq \theta_{\mathrm{LTF}}^{\gamma-1} \cdot h\left(\frac{4M+1}{6M}\right)$

Negligible Leakage Power Consumption

Worst-case Approximation Factor for DLTF and SFA (when $\beta = 0$):

$$AF_{SFA}^{DLTF^{(\beta=0)}} \leq \max\left\{h\left(\delta^{max}\right), \theta_{LTF}^{\gamma-1} \cdot h\left(\frac{4M+1}{6M}\right)\right\}$$



Note: $AF_{SFA}^{DLTF^{(\beta=0)}}$ *only depends on the values of* γ *and M.*

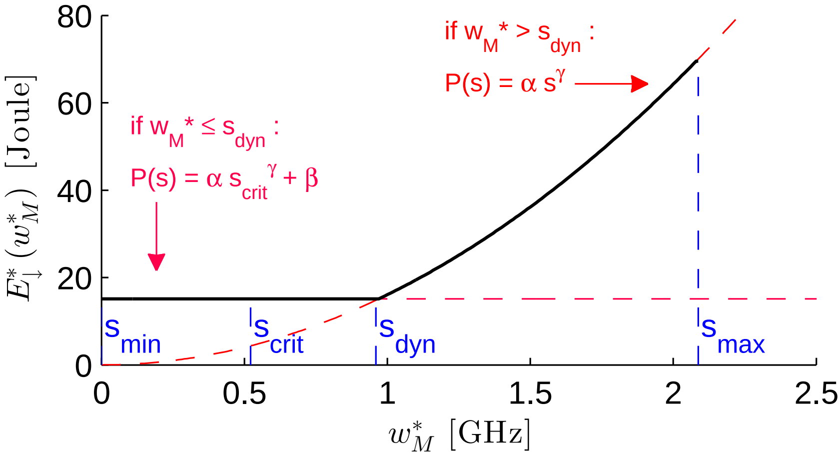
Outline

Karlsruhe Institute of Technology

- Introduction
- Motivation and Problem Definition
- Task Partitioning Scheme: Double-Largest-Task-First (DLTF)
- Approximation Factor Analysis (energy consumption): DLTF and SFA
  - Negligible Leakage Power Consumption
  - Non-negligible Leakage Power Consumption
  - Non-negligible Sleeping Overhead
- Simulations
- Conclusions

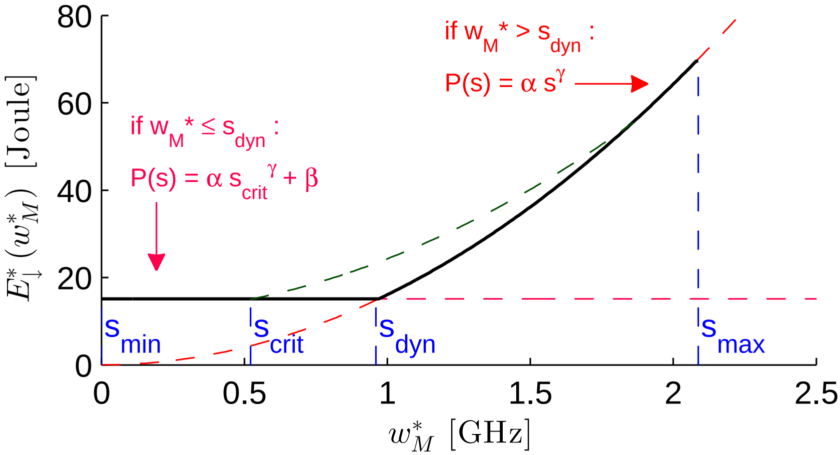
Non-negligible Leakage Power Consumption

We approximate the Lower Bound Energy Consumption:



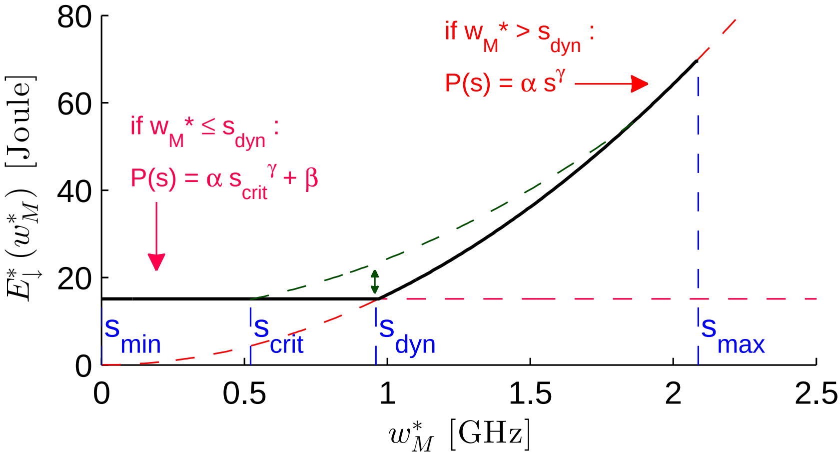
Non-negligible Leakage Power Consumption

We approximate the Lower Bound Energy Consumption:



Non-negligible Leakage Power Consumption

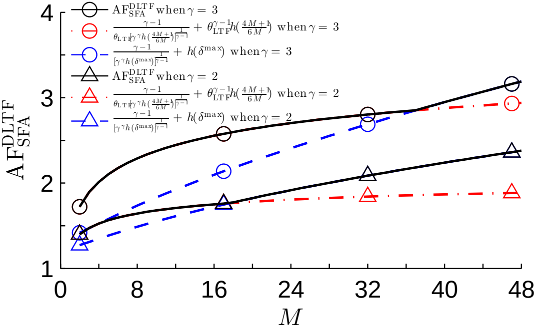
We approximate the Lower Bound Energy Consumption:



Negligible Leakage Power Consumption

Worst-case Approximation Factor for DLTF and SFA (when $\beta \neq 0$):

$$\mathrm{AF}_{\mathrm{SFA}}^{\mathrm{DLTF}} \le \max\left\{ \frac{\gamma-1}{\left[\gamma^{\gamma} h\left(\delta^{\max}\right)\right]^{\frac{1}{\gamma-1}}} + h\left(\delta^{\max}\right), \frac{\gamma-1}{\theta_{\mathrm{LTF}}\left[\gamma^{\gamma} h\left(\frac{4M+1}{6M}\right)\right]^{\frac{1}{\gamma-1}}} + \theta_{\mathrm{LTF}}^{\gamma-1} h\left(\frac{4M+1}{6M}\right) \right\}$$



Note: AF_{SFA}^{DLTF} only depends on the values of γ and M.

Outline

Karlsruhe Institute of Technology

- Introduction
- Motivation and Problem Definition
- Task Partitioning Scheme: Double-Largest-Task-First (DLTF)
- Approximation Factor Analysis (energy consumption): DLTF and SFA
  - Negligible Leakage Power Consumption
  - Non-negligible Leakage Power Consumption
  - Non-negligible Sleeping Overhead
- Simulations
- Conclusions

Non-negligible Sleeping Overhead

- Our strategy can be combined with any DPM schemes.
- Two cases:
  - $\sum_{i=1}^{M} w_i^* < s_{\text{crit}}$:
    - DLTF assigns all tasks on one core and SFA executes at s_{crit}.
    - Uniprocessor scheduling problem.
    - For example, using *Left-To-Right* (LTR) algorithm [4] $\rightarrow$ 2-approximation.
  - $\sum_{i=1}^{M} w_i^* \geq s_{\text{crit}}$:
    - Using any DPM scheme, against the optimal solution:

$$\text{AF}_{\text{SFA-DPM}}^{\text{DLTF}} \leq \text{AF}_{\text{SFA}}^{\text{DLTF}} + \frac{\gamma - 1}{\gamma}$$

[4] Sandy Irani, Sandeep Shukla, and Rajesh Gupta. “Algorithms for power savings”. In: *Symposium on Discrete Algorithms (SODA).* Baltimore, Maryland, 2003, pp. 37–46

Non-negligible Sleeping Overhead

Karlsruhe Institute of Technology

- Our strategy can be combined with any DPM schemes.
- Two cases:
  - $\sum_{i=1}^{M} w_i^* < s_{\text{crit}}$:
    - DLTF assigns all tasks on one core and SFA executes at s_{crit}.
    - Uniprocessor scheduling problem.
    - For example, using *Left-To-Right* (LTR) algorithm [4] $\rightarrow$ 2-approximation.
  - $\sum_{i=1}^{M} w_i^* \geq s_{\text{crit}}$:
    - Using any DPM scheme, against the optimal solution:

$$\text{AF}_{\text{SFA-DPM}}^{\text{DLTF}} \leq \text{AF}_{\text{SFA}}^{\text{DLTF}} + \frac{\gamma - 1}{\gamma}$$

[4] Sandy Irani, Sandeep Shukla, and Rajesh Gupta. "Algorithms for power savings". In: *Symposium on Discrete Algorithms (SODA)*. Baltimore, Maryland, 2003, pp. 37–46

Non-negligible Sleeping Overhead

- Our strategy can be combined with any DPM schemes.
- Two cases:
  - $\sum_{i=1}^{M} w_i^* < s_{\text{crit}}$:
    - DLTF assigns all tasks on one core and SFA executes at s_{crit}.
    - Uniprocessor scheduling problem.
    - For example, using *Left-To-Right* (LTR) algorithm [4] $\rightarrow$ 2-approximation.
  - $\sum_{i=1}^{M} w_i^* \geq s_{\text{crit}}$:
    - Using any DPM scheme, against the optimal solution:

$$\text{AF}_{\text{SFA-DPM}}^{\text{DLTF}} \leq \text{AF}_{\text{SFA}}^{\text{DLTF}} + \frac{\gamma - 1}{\gamma}$$

[4] Sandy Irani, Sandeep Shukla, and Rajesh Gupta. “Algorithms for power savings”. In: *Symposium on Discrete Algorithms (SODA)*. Baltimore, Maryland, 2003, pp. 37–46

Outline

- Introduction
- Motivation and Problem Definition
- Task Partitioning Scheme: Double-Largest-Task-First (DLTF)
- Approximation Factor Analysis (energy consumption): DLTF and SFA
  - Negligible Leakage Power Consumption
  - Non-negligible Leakage Power Consumption
  - Non-negligible Sleeping Overhead
- Simulations
- Conclusions

Simulation Results

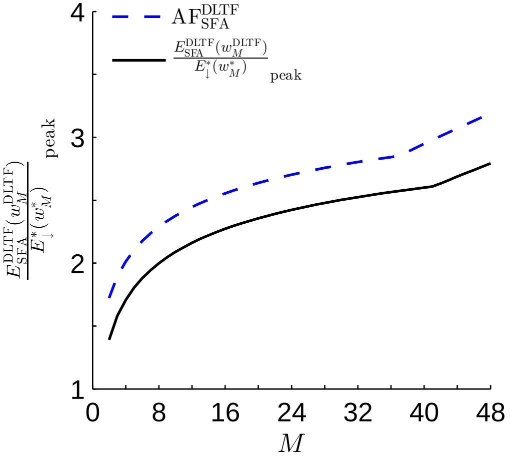
For negligible overhead for sleeping:

- Power parameters modelled from SCC.
- 150 cases of synthetic tasks for every *M*, with different:
  - Amount of tasks.
  - Cycle utilizations.
  - Resulting hyper-periods.
- Task Partitioning:
  - DLTF for SFA.
  - Lower bound for the optimal solution.

Simulation Results

For negligible overhead for sleeping:

- Power parameters modelled from SCC.
- 150 cases of synthetic tasks for every *M*, with different:
  - Amount of tasks.
  - Cycle utilizations.
  - Resulting hyper-periods.
- Task Partitioning:
  - DLTF for SFA.
  - Lower bound for the optimal solution.



Outline

- Introduction
- Motivation and Problem Definition
- Task Partitioning Scheme: Double-Largest-Task-First (DLTF)
- Approximation Factor Analysis (energy consumption): DLTF and SFA
  - Negligible Leakage Power Consumption
  - Non-negligible Leakage Power Consumption
  - Non-negligible Sleeping Overhead
- Simulations
- Conclusions

Conclusions

Karlsruhe Institute of Technology

- For periodic tasks, combining DLTF and SFA is a practical solution for energy minimization.
- Approximation factor of DLTF and SFA for energy efficiency:
  - Considered cases: negligible leakage, non-negligible leakage, and combinations with DPM.
  - Bounded by γ and M (for all cases).
  - Simulations show a *small* gap compared with our analysis (for the worst-case).
- Combining DLTF and SFA is an acceptable scheme based on the worst-case analysis.

Thank you!

Questions?

Thank you!

Questions?