Energy Efficiency Analysis for the Single Frequency Approximation (SFA) Scheme

Santiago Pagani and Jian-Jia Chen

LS 12, TU Dortmund

RTCSA - August 2013

Outline

- Introduction
- Motivation and Problem Definition
- Approximation Factor Analysis (energy consumption) of SFA
    - Negligible Leakage Power Consumption
    - Non-negligible Leakage Power Consumption
    - Balanced Task Sets and Non-negligible Overhead for Sleeping
- Simulations
- Conclusions

Outline

- Introduction

Introduction

Importance of Energy Efficiency:

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

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

Dynamic Power Management (DPM):

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

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.

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.

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

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.

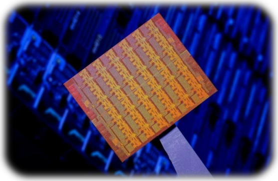


Figure: Intel's SCC snapshot

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 Power Model

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

Introduction

CMOS Power Model

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

Considering that:

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

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

Introduction

CMOS 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 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:

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

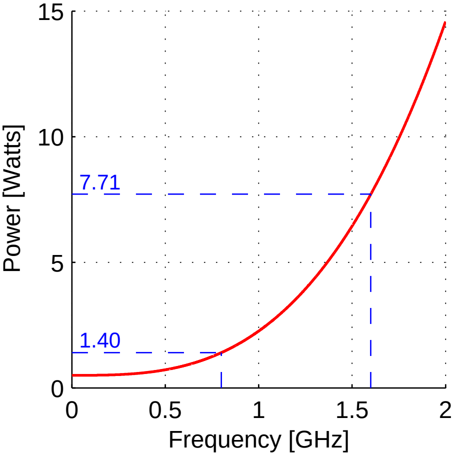


Figure: $\alpha = 1.76 \frac{\text{Watts}}{\text{GHz}^3}$, $\gamma = 3$ and $\beta = 0.5$ Watts

Introduction

Energy Consumption

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

Critical Frequency:

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

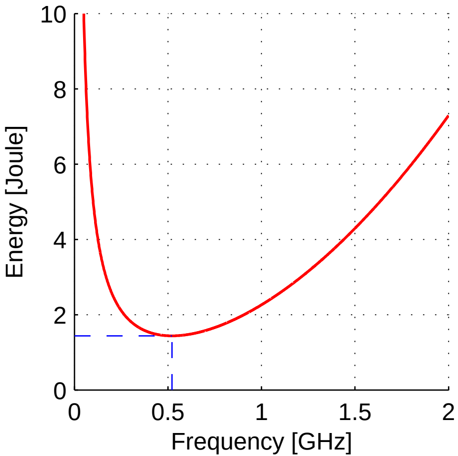


Figure: $\alpha = 1.76\frac{\text{Watts}}{\text{GHz}^3}$, $\gamma = 3$, $\beta = 0.5$ Watts and $\Delta c = 10^9$ cycles

Outline

- Motivation and Problem Definition

Motivation

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

What voltage/frequency policy should be used?

Motivation

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

What voltage/frequency policy should be used?

Single Frequency Approximation (SFA) Scheme:

- Use the lowest voltage/frequency, satisfying the timing constraints.
- Is the *simplest* and *most intuitive* strategy.

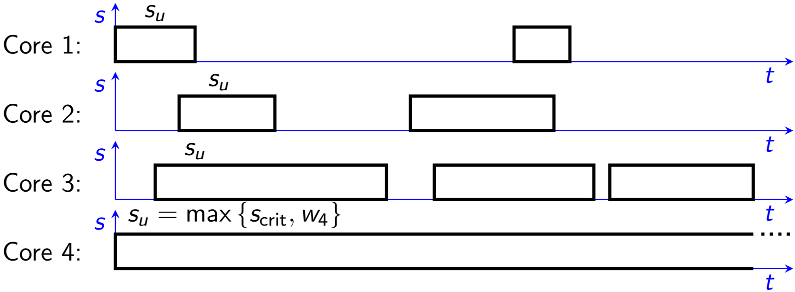
Motivation

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

What voltage/frequency policy should be used?

Single Frequency Approximation (SFA) Scheme:

- Use the lowest voltage/frequency, satisfying the timing constraints.
- Is the *simplest* and *most intuitive* strategy.



Motivation

PROs of SFA:

- Linear time complexity.
- Significantly reduces the management overhead.
- No frequency alignment between cores $\Longrightarrow$ Any uni-core DPM technique can be adopted individually in each core.

Motivation

PROs of SFA:

- Linear time complexity.
- Significantly reduces the management overhead.
- No frequency alignment between cores $\Longrightarrow$ Any uni-core DPM technique can be adopted individually in each core.

CONs of SFA:

- SFA might consume more energy than another DVFS schedule.

How much more?

Problem Definition

- For real-time tasks, already partitioned into task sets $\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_M$.
- Task sets ordered by their cycle utilizations: $w_1 \leq w_2 \leq \cdots \leq w_M$.
- Considering partitioned scheduling.
- Using *Earliest-Deadline-First* (EDF) algorithm.

Objective: Provide *theoretical analysis* to show the effectiveness of SFA for energy minimization.

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

Outline

- Approximation Factor Analysis (energy consumption) of SFA
  - Negligible Leakage Power Consumption

Negligible Leakage Power Consumption

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

- We execute at (single frequency) $s_u = w_M$.
- The cycle utilization distribution does not matter.

$$E_{\text{SFA}}^{\beta=0}(w_M) = \alpha L \left({w_M}^{\gamma-1}\right) \sum_{i=1}^{M} w_i$$

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

Negligible Leakage Power Consumption

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

- We execute at (single frequency) $s_u = w_M$.
- The cycle utilization distribution does not matter.

$$E_{\mathrm{SFA}}^{\beta=0}(w_M) = \alpha L \left({w_M}^{\gamma-1} \right) \sum_{i=1}^{M} w_i$$

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

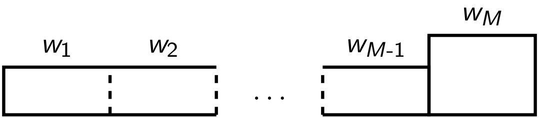
- Unroll periodic tasks in a hyper-period $\Rightarrow$ frame-based tasks.
- Use the results from Yang et al. [1]:

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

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

Negligible Leakage Power Consumption

Critical Cycle Utilization Distribution: Minimizes the lower bound of energy consumption, for a fixed w_M and $\sum_{i=1}^{M} w_i$.



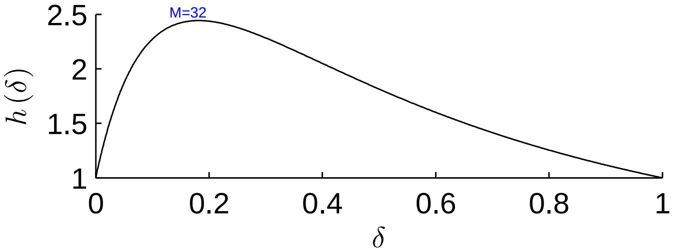
- $w_1 = w_2 = \cdots = w_{M-1} = \text{Average}\left(w_1, w_2, \ldots, w_{M-1}\right)$
- Utilization Ratio: $0 \leq \delta = \frac{\text{Average}(w_1, w_2, \ldots, w_{M-1})}{w_M} \leq 1$

Negligible Leakage Power Consumption

Approximation factor of SFA when $\beta = 0$:

$$\mathrm{AF}_{\mathrm{SFA}}^{\beta=0} \leq h(\delta) = \frac{1 - \delta + \delta M}{\left(1 - \delta + \delta \sqrt[\gamma]{M}\right)^{\gamma}} \leq h(\delta^*)$$

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

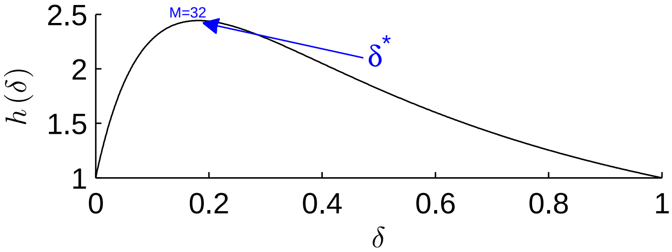


Negligible Leakage Power Consumption

Approximation factor of SFA when $\beta = 0$:

$$\mathrm{AF}_{\mathrm{SFA}}^{\beta=0} \leq h(\delta) = \frac{1 - \delta + \delta M}{\left(1 - \delta + \delta \sqrt[\gamma]{M}\right)^{\gamma}} \leq h(\delta^*)$$

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

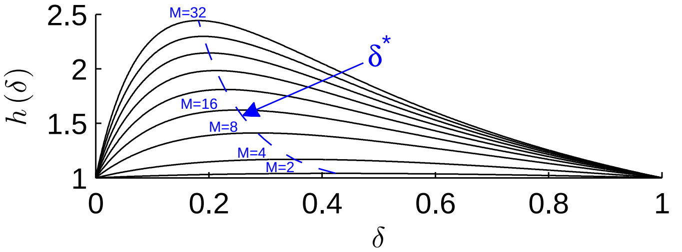


Negligible Leakage Power Consumption

Approximation factor of SFA when $\beta = 0$:

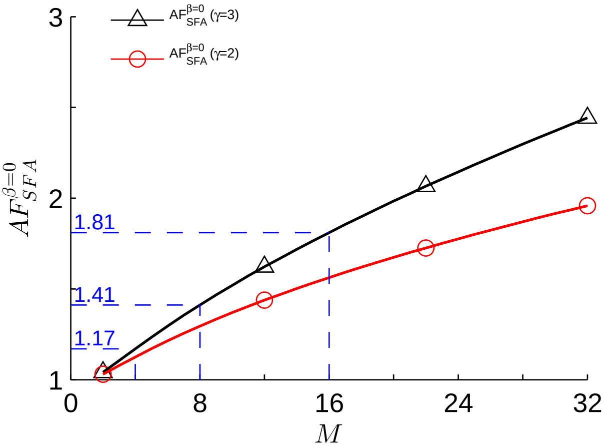
$$\mathsf{AF}_{\mathsf{SFA}}^{\beta=0} \leq h(\delta) = \frac{1-\delta+\delta M}{\left(1-\delta+\delta\sqrt[\gamma]{M}\right)^{\gamma}} \leq h(\delta^*)$$

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



Negligible Leakage Power Consumption

Approximation factor of SFA when $\beta = 0$ (function of M):



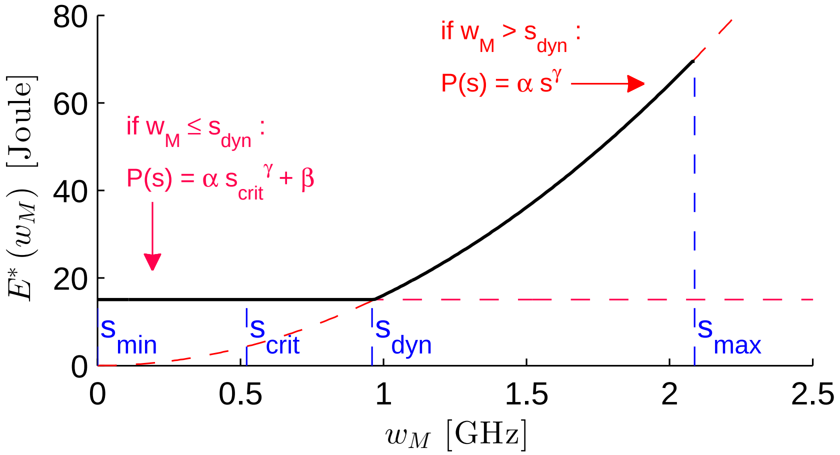
Note: $AF_{SFA}^{\beta=0}$ *only depends on the values of* γ *and* M*.*

Outline

- Approximation Factor Analysis (energy consumption) of SFA
  - Non-negligible Leakage Power Consumption

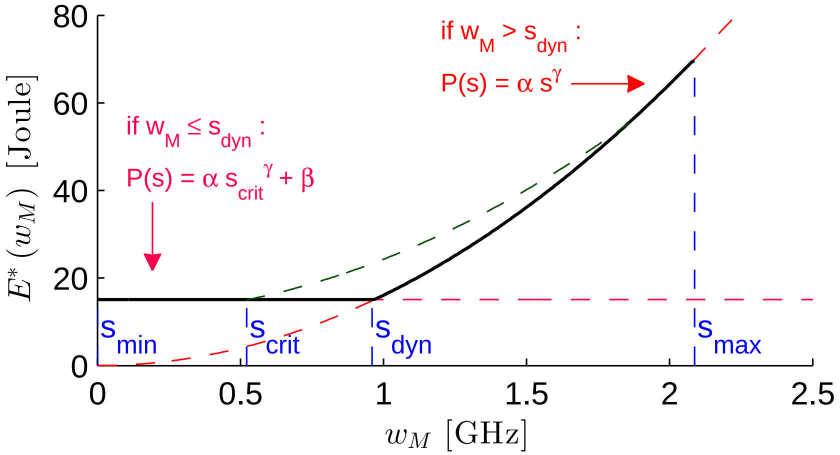
Non-negligible Leakage Power Consumption

We approximate the Lower Bound Energy Consumption:



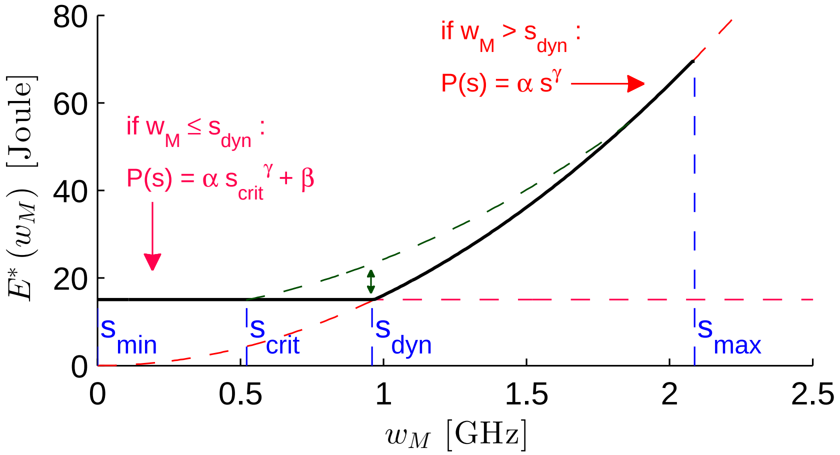
Non-negligible Leakage Power Consumption

We approximate the Lower Bound Energy Consumption:



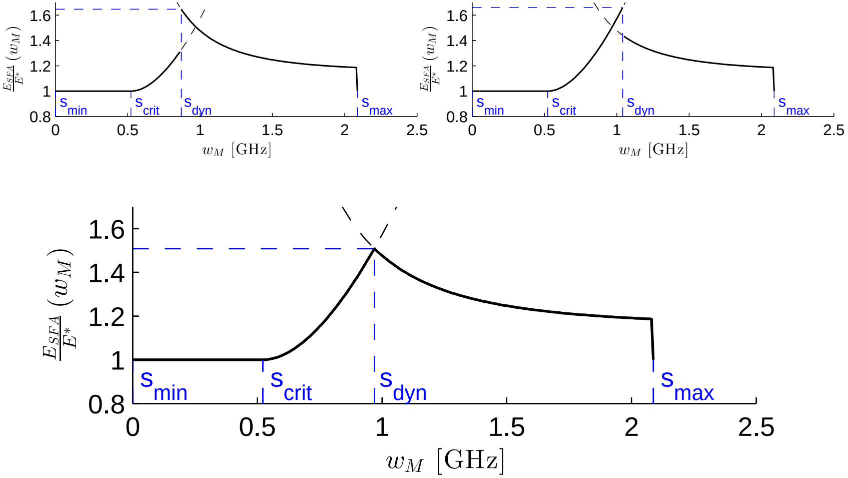
Non-negligible Leakage Power Consumption

We approximate the Lower Bound Energy Consumption:



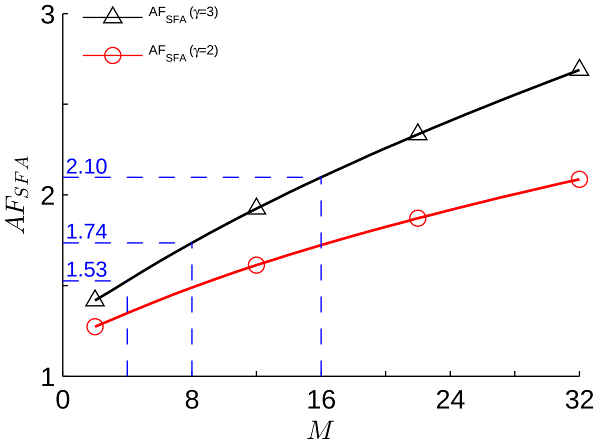
Non-negligible Leakage Power Consumption

The approximation factor depends on how we choose s_{dyn}:



Non-negligible Leakage Power Consumption

Approximation factor of SFA when $\beta \neq 0 \Rightarrow \mathrm{AF_{SFA}} \leq \frac{\gamma - 1}{[\gamma^{\gamma} h(\delta^*)]^{\frac{1}{\gamma-1}}} + h(\delta^*)$



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

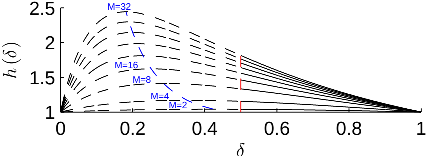
Outline

- Approximation Factor Analysis (energy consumption) of SFA
    - Balanced Task Sets and Non-negligible Overhead for Sleeping

Balanced Task Sets & Non-negligible Sleeping Overhead

Balanced Task Sets:

If $\delta \geq 0.5$ (e.g., using LTF) $\Rightarrow \text{AF}_{\text{SFA}}\,(\delta \geq 0.5) < \text{AF}_{\text{SFA}}$

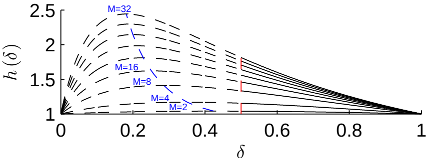


[2] Sandy Irani, Sandeep Shukla, and Rajesh Gupta. "Algorithms for power savings". In: *the 14th Symposium on Discrete Algorithms (SODA)*. 2003, pp. 37–46

Balanced Task Sets & Non-negligible Sleeping Overhead

Balanced Task Sets:

If $\delta \geq 0.5$ (e.g., using LTF) $\Rightarrow$ $\text{AF}_{\text{SFA}}\,(\delta \geq 0.5) < \text{AF}_{\text{SFA}}$



Non-negligible Overhead for Sleeping:

- SFA can be combined with any uni-core DPM solution.
- For example, with Left-To-Right (LTR)[2] algorithm :

$$\text{AF}_{\text{SFA-LTR}} = \text{AF}_{\text{SFA}} + 1$$

[2] Sandy Irani, Sandeep Shukla, and Rajesh Gupta. "Algorithms for power savings". In: *the 14th Symposium on Discrete Algorithms (SODA)*. 2003, pp. 37–46

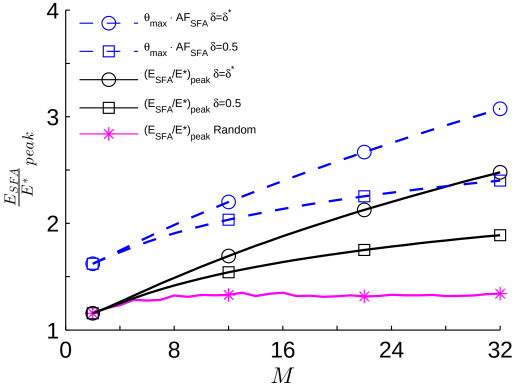
Outline

- Simulations

Simulation Results

For negligible overhead for sleeping:

- Power parameters modelled from SCC.
- Discrete Frequencies:
  - [0.1 GHz; 3.0 GHz]
  - Steps of 0.1 GHz
- w_M:
  - [0.2 GHz; 3.0 GHz]
  - Steps of 20 MHz
- $L = 1, 2, \ldots, 5$ seconds.



Outline

- Conclusions

Conclusions

- SFA: state-of-the-art energy efficient scheduling for periodic tasks.
- Approximation factor of SFA for energy efficiency:
  - Considered cases: negligible leakage, non-negligible leakage, balanced task sets, 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).
- SFA is an acceptable scheme based on the worst-case analysis.
- The analysis for SFA for fixed task sets is a cornerstone for task partitioning. Further work considering SFA and task partitioning will be published in RTSS 2013[3].

[3] Santiago Pagani and Jian-Jia Chen. "Energy Efficient Task Partitioning based on the Single Frequency Approximation Scheme". In: *Proceedings of the 34th IEEE Real-Time Systems Symposium (RTSS)*. Vancouver, Canada, 2013

Thank you!

Thank you!

Questions?

Thank you!

Questions?

Extensions for Practical Systems

Systems with Discrete Frequencies:

- Available frequencies $\{f_1, f_2, \ldots, f_F\}$.

Approximation factor of SFA for discrete frequencies $\Rightarrow$ $\text{AF}_{\text{SFA}} \cdot \theta_{\max}$

$$\theta_{\max} = \max_{1<i\leq F} \frac{P(f_i)\cdot f_{i-1}}{P(f_{i-1})\cdot f_i}$$

Extensions for Practical Systems

Systems with Discrete Frequencies:

- Available frequencies $\{f_1, f_2, \ldots, f_F\}$.

Approximation factor of SFA for discrete frequencies $\Rightarrow$
$AF_{SFA} \cdot \theta_{max}$

$$\theta_{max} = \max_{1<i\leq F} \frac{P(f_i)\cdot f_{i-1}}{P(f_{i-1})\cdot f_i}$$

For example:

- If $\alpha = 1.76 \frac{\text{Watts}}{\text{GHz}^3}$, $\beta = 0.5$ Watts, $\gamma = 3$
- Available frequencies $\{0.1 \text{ GHz}, 0.2 \text{ GHz}, \ldots, 3.0 \text{ GHz}\}$

$$\Rightarrow \quad \theta_{max} = 1.14$$

Extensions for Practical Systems

Systems with Multiple Voltage Islands:

- Given mapping of task partitions in every island.
- Using SFA in each individual island.

$$\Rightarrow \quad \mathrm{AF}_{\mathrm{SFA}}^{\text{V-islands}} = \frac{\sum_{j=1}^{V} E_{\mathrm{SFA}\,j}}{\sum_{j=1}^{V} E_{\mathrm{OPT}\,j}} \leq \frac{\sum_{j=1}^{V} \mathrm{AF}_{\mathrm{SFA}} \cdot E_j^*}{\sum_{j=1}^{V} E_j^*} = \mathrm{AF}_{\mathrm{SFA}}$$

Simulation Setup

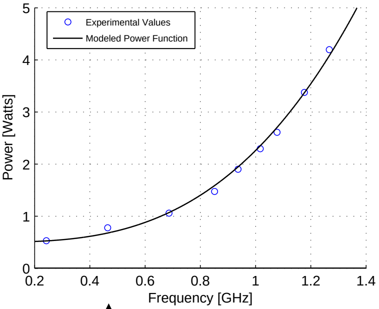
Experimental results on SCC[4]:

Frequency vs. voltage

Voltage [Volts]	Frequency [MHz]
0.73	301.48
0.75	368.82
0.85	569.45
0.94	742.96
1.04	908.92
1.14	1077.11
1.23	1223.37
1.32	1303.79

Power vs. voltage

Voltage [Volts]	Total Power [Watts]
0.70	25.38
0.80	37.26
0.91	50.76
1.00	70.73
1.05	91.25
1.10	110.15
1.14	125.27
1.21	161.99
1.28	201.40



Hardware parameters modelled from SCC:

- $\alpha = 1.76 \frac{\text{Watts}}{\text{GHz}^3}$, $\beta = 0.5$ Watts, $\gamma = 3$ and $s_{\text{crit}} = 0.52$ GHz.
- Available frequencies: $\{0.1 \text{ GHz}, 0.2 \text{ GHz}, \ldots, 3.0 \text{ GHz}\}$.

[4] Jason Howard and others. "A 48-Core IA-32 Processor in 45 nm CMOS Using On-Die Message-Passing and DVFS for Performance and Power Scaling". In: *J. Solid-State Circuits* 46.1 (2011), pp. 173–183

Simulation Setup

Maximum Cycle Utilization w_M (stepped by 20 MHz):

- (a) From 0.2 GHz to 1.3 GHz.
- (b) From 0.2 GHz to 3.0 GHz.

Hyper-periods (for every w_M):

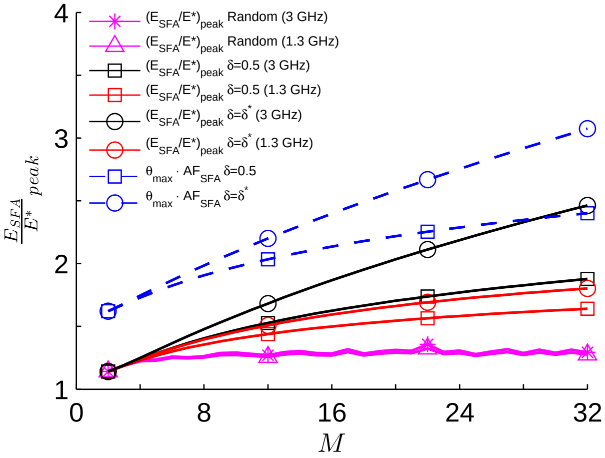
- $L = 1, 2, \ldots, 5$ seconds.

Cycle Utilization Distribution:

- (1) *Critical Utilization Distribution* with $\delta = \delta^*$ (worst-case).
- (2) *Critical Utilization Distribution* with $\delta = 0.5$ (balanced task sets).
- (3) 100 different random utilization distributions.

Detailed Simulation Results

For negligible overhead for sleeping:



Detailed Simulation Results

For non-negligible overhead for sleeping: