Optimal Speed Assignment for Probabilistic Execution Times

2nd Power-Aware Real-Time Computing Workshop

Claudio Scordino

University of Pisa, Italy
scordino@di.unipi.it

Enrico Bini

Scuola Superiore Sant'Anna, Italy
e.bini@sssup.it

Energy-constrained real-time systems

- Embedded systems
  - Special purpose computers, part of larger systems which may not be of electronic kind
  - Often powered by rechargeable batteries
  - Battery influences autonomy, size, weight and cost
  - **Goal**: extend the lifetime of the system
- Servers
  - The power consumed by the processors is increasing
  - Dissipating the heat generated is becoming very difficult
  - Cooling devices can consume up to 50% of total energy
  - **Goal**: reduce the thermal dissipation

Energy-constrained real-time systems

- Embedded systems
  - Special purpose computers, part of larger systems which may not be of electronic kind
  - Often powered by rechargeable batteries
  - Battery influences autonomy, size, weight and cost
  - **Goal**: extend the lifetime of the system
- Servers
  - The power consumed by the processors is increasing
  - Dissipating the heat generated is becoming very difficult
  - Cooling devices can consume up to 50% of total energy
  - **Goal**: reduce the thermal dissipation

Dynamic Voltage Scaling

- Dynamic Voltage Scaling (DVS):
  - Technique to reduce the power consumed by the processor
  - Change of processor voltage and frequency at runtime
    $\Rightarrow$ Tasks take more time to be executed
  - Allows to balance computational speed vs. energy consumption
- Power-aware scheduling algorithm
  - Selects both the task to be scheduled and the processor speed
  - Must provide the worst-case computational requirement

Dynamic Voltage Scaling

- Dynamic Voltage Scaling (DVS):
  - Technique to reduce the power consumed by the processor
  - Change of processor voltage and frequency at runtime
    $\Rightarrow$ Tasks take more time to be executed
  - Allows to balance computational speed vs. energy consumption

- Power-aware scheduling algorithm
  - Selects both the task to be scheduled and the processor speed
  - Must provide the worst-case computational requirement

Our problem

- The design of a new algorithm from scratch is complex
  - The problem with many tasks is complex
  - We preferred a bottom-up procedure
  - Provide the basis for the design of new algorithms
- Simple case:
  - A certain amount of "work" to be finished in $[0, T]$
    - One task τ
    - One hard deadline T
  - Processor with continuous speed scaling
- General model:
  - No specific power functions
  - Consider energy and time overheads during voltage transition

Our problem

- The design of a new algorithm from scratch is complex
  - The problem with many tasks is complex
  - We preferred a bottom-up procedure
  - Provide the basis for the design of new algorithms

- Simple case:
  - A certain amount of "work" to be finished in $[0, T]$
    - One task τ
    - One hard deadline T
  - Processor with continuous speed scaling

- General model:
  - No specific power functions
  - Consider energy and time overheads during voltage transition

Our problem

- The design of a new algorithm from scratch is complex
  - The problem with many tasks is complex
  - We preferred a bottom-up procedure
  - Provide the basis for the design of new algorithms
- Simple case:
  - A certain amount of "work" to be finished in $[0, T]$
    - One task τ
    - One hard deadline T
  - Processor with continuous speed scaling
- General model:
  - No specific power functions
  - Consider energy and time overheads during voltage transition

Probabilistic execution times

- The probability of a task executing for its WCEC is very low
  - Embedded systems: variations up to 87% wrt WCEC
  - Servers: average processor use between 10% and 50% of peak capacity
- Many algorithms try to predict the actual execution cycles
  - Typically, only the average value is used
- Better reduction can be achieved using a more detailed information on the required workload
  - **Idea**: exploit probabilistic information about execution times
  - **Goal**: minimize expected energy consumption while meeting the deadline

Probabilistic execution times

- The probability of a task executing for its WCEC is very low
  - Embedded systems: variations up to 87% wrt WCEC
  - Servers: average processor use between 10% and 50% of peak capacity
- Many algorithms try to predict the actual execution cycles
  - Typically, only the average value is used
- Better reduction can be achieved using a more detailed information on the required workload
  - **Idea**: exploit probabilistic information about execution times
  - **Goal**: minimize expected energy consumption while meeting the deadline

Probabilistic execution times

- The probability of a task executing for its WCEC is very low
  - Embedded systems: variations up to 87% wrt WCEC
  - Servers: average processor use between 10% and 50% of peak capacity
- Many algorithms try to predict the actual execution cycles
  - Typically, only the average value is used
- Better reduction can be achieved using a more detailed information on the required workload
  - **Idea**: exploit probabilistic information about execution times
  - **Goal**: minimize expected energy consumption while meeting the deadline

Deferring the work

- Since we don't know actual execution cycles, we can't compute the optimal constant speed
- **Idea**: defer some work
  - We expect that the task will request less than its WCEC
  - Technique widely adopted in many power-aware algorithms
    - Examples: DRA, RTDVS, EDF-DVS, PACE, PPACE
- We use only 2 levels of speed in $[0, T]$
  - **Goal**: find the optimal speed assignments and the optimal instant for speed change
  - We want the analytical expression in a closed form

Deferring the work

- Since we don't know actual execution cycles, we can't compute the optimal constant speed
- **Idea**: defer some work
  - We expect that the task will request less than its WCEC
  - Technique widely adopted in many power-aware algorithms
    - Examples: DRA, RTDVS, EDF-DVS, PACE, PPACE
- We use only 2 levels of speed in $[0, T]$
  - **Goal**: find the optimal speed assignments and the optimal instant for speed change
  - We want the analytical expression in a closed form

Deferring the work

- Since we don't know actual execution cycles, we can't compute the optimal constant speed
- **Idea**: defer some work
  - We expect that the task will request less than its WCEC
  - Technique widely adopted in many power-aware algorithms
    - Examples: DRA, RTDVS, EDF-DVS, PACE, PPACE
- We use only 2 levels of speed in $[0, T]$
  - **Goal**: find the optimal speed assignments and the optimal instant for speed change
  - We want the analytical expression in a closed form

System model

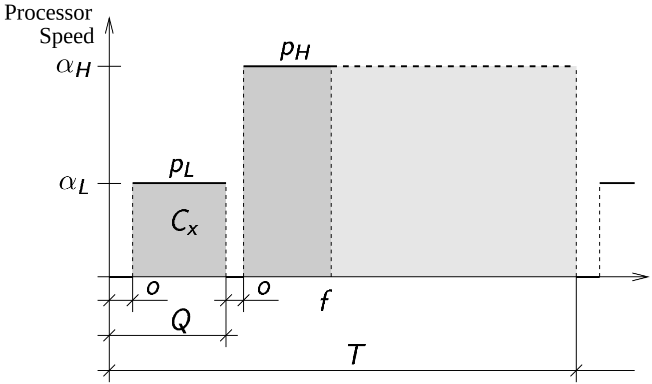
- Processor model:
  - Speed: $0 \leq \alpha \leq \alpha_{\text{max}}$
  - Generic function for power consumption: $p(\alpha)$
  - Overhead of voltage transition
    - o: time overhead
    - e: energy consumed
- Task model:
  - Hard deadline: T
  - Actual execution cycles c unknown
  - $f_c(c)$: p.d.f. of number of cycles required in $[0, T]$
  - C_{max}: worst case execution cycles in $[0, T]$

System model

- Processor model:
  - Speed: $0 \leq \alpha \leq \alpha_{max}$
  - Generic function for power consumption: $p(\alpha)$
  - Overhead of voltage transition
    - o: time overhead
    - e: energy consumed

- Task model:
  - Hard deadline: T
  - Actual execution cycles c unknown
  - $f_c(c)$: p.d.f. of number of cycles required in $[0, T]$
  - C_{max}: worst case execution cycles in $[0, T]$

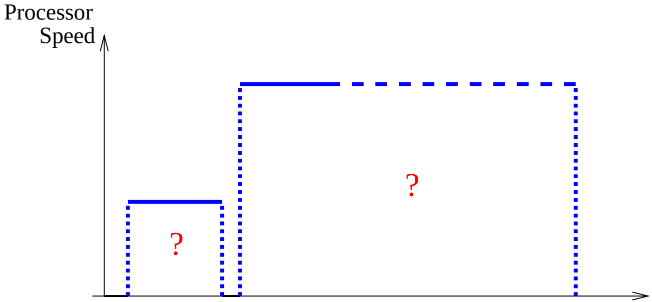
Energy management scheme (1)



Q: instant when to switch

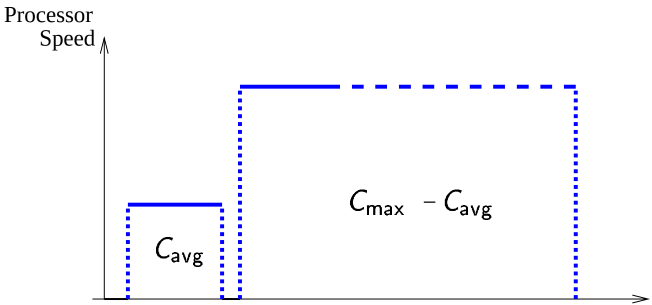
C_x: number of cycles provided at α_L

Energy management scheme (2)



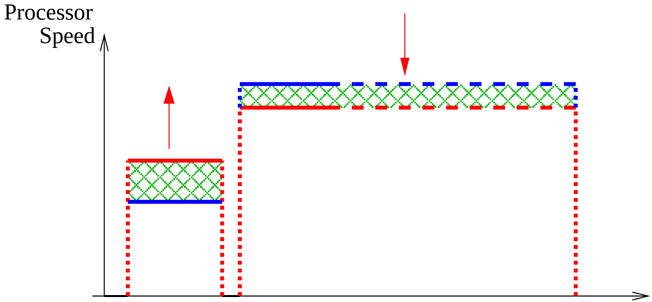
- How much should we execute in the first part ??

Energy management scheme (2)



- On average 50% of times we need only the first part
- Sub-optimal solution

Energy management scheme (2)



- We increase the number of times that we need only the first part

Average energy consumed

- From the worst-case constraint we have

$$\alpha_L(C_x, Q) = \frac{C_x}{Q - o} \qquad \alpha_H(C_x, Q) = \frac{C_{\max} - C_x}{T - Q - o}$$

- $E = \begin{cases} e + \frac{p_L}{\alpha_L} c & \text{if } f \leq Q \\ 2e + \frac{p_L}{\alpha_L} C_x + \frac{p_H}{\alpha_H}(c - C_x) & \text{if } f > Q + o \end{cases}$

- $E_{\text{avg}} = \int_0^{C_{\max}} E\, f_C(c)\, \mathrm{d}c$ Expectation of E

Average energy consumed

- From the worst-case constraint we have

$$\alpha_L(C_x, Q) = \frac{C_x}{Q - o} \qquad \alpha_H(C_x, Q) = \frac{C_{\max} - C_x}{T - Q - o}$$

- $E = \begin{cases} e + \frac{p_L}{\alpha_L} c & \text{if } f \leq Q \\ 2e + \frac{p_L}{\alpha_L} C_x + \frac{p_H}{\alpha_H}(c - C_x) & \text{if } f > Q + o \end{cases}$

- $E_{\text{avg}} = \int_0^{C_{\max}} E \, f_C(c) \, \mathrm{d}c$ Expectation of E

Average energy consumed

- From the worst-case constraint we have

$$\alpha_L(C_x, Q) = \frac{C_x}{Q - o} \qquad \alpha_H(C_x, Q) = \frac{C_{\max} - C_x}{T - Q - o}$$

- $E = \begin{cases} e + \frac{p_L}{\alpha_L} c & \text{if } f \leq Q \\ 2e + \frac{p_L}{\alpha_L} C_x + \frac{p_H}{\alpha_H}(c - C_x) & \text{if } f > Q + o \end{cases}$

- $E_{\text{avg}} = \int_0^{C_{\max}} E\, f_C(c)\, \mathrm{d}c$ Expectation of E

Average energy consumed (2)

$$E_{\text{avg}} = e\,(2 - F_C(C_x)) + \frac{p_H}{\alpha_H}(C_{\text{avg}} - \gamma(C_x)) + \frac{p_L}{\alpha_L}\,\gamma(C_x)$$

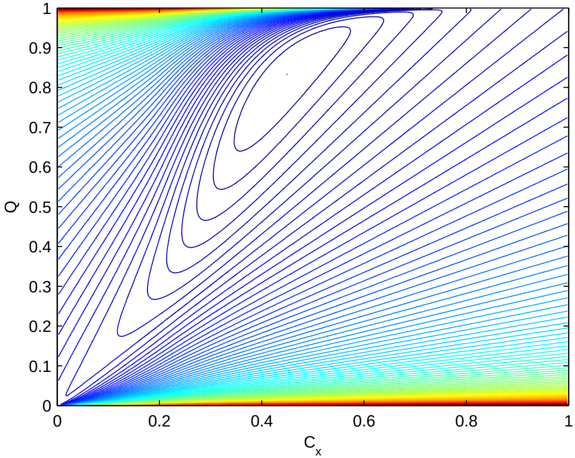
where

- $\gamma(x) = G_C(x) + x(1 - F_C(x))$ $\qquad 0 \leq \gamma(x) \leq C_{\text{avg}} \quad \forall x$
- $F_C(x) = \int_0^x f_C(c)\,\mathrm{d}c$
- $G_C(x) = \int_0^x c\, f_C(c)\,\mathrm{d}c$

Level curves of E_{avg}

- Plot of level curves of E_{avg} on a plane (C_x, Q)
- Particular case:
  - Exponential p.d.f.
  - Polynomial power function: $p(\alpha) = k\,\alpha^3$
  - $T = 1$
  - $C_{avg} = 0.2929$

Level curves of E_{avg} (2)



$$C_x > C_{avg} = 0.2929$$

Minimum of E_{avg}

- The minimum satisfies $\nabla E_{\text{avg}} = 0$
- For the general model the components of ∇E_{avg} are:

$$\begin{cases} \frac{\partial E_{\text{avg}}}{\partial C_x} = -e\, f_C(C_x) - \left(p'_H - \frac{p_H}{\alpha_H}\right) \frac{C_{\text{avg}} - \gamma(C_x)}{C_{\max} - C_x} \\ \qquad\qquad + \left(p'_L - \frac{p_L}{\alpha_L}\right) \frac{\gamma(C_x)}{C_x} - \left(\frac{p_H}{\alpha_H} - \frac{p_L}{\alpha_L}\right) \gamma'(C_x) \\ \frac{\partial E_{\text{avg}}}{\partial Q} = (p'_H \alpha_H - p_H) \frac{C_{\text{avg}} - \gamma(C_x)}{C_{\max} - C_x} - (p'_L \alpha_L - p_L) \frac{\gamma(C_x)}{C_x} \end{cases}$$

Polynomial power function

- Significant example:
  - Time and energy overheads equal to zero (i.e. $e = o = 0$)
  - Polynomial power function $p(\alpha) = k\,\alpha^{\boxed{n}}$

- $\nabla E_{avg} = 0$ can be simplified:

$$\begin{cases} \dfrac{(n-1)\,\gamma(C_x) + C_x\gamma'(C_x)}{(n-1)\,(C_{avg} - \gamma(C_x)) + (C_{max} - C_x)\gamma'(C_x)} \left(\dfrac{C_{max}}{C_x} - 1\right)\left(\dfrac{C_{avg}}{\gamma(C_x)} - 1\right) = \dfrac{T}{Q} - 1 \\ \left(\dfrac{C_{max}}{C_x} - 1\right)^{n-1}\left(\dfrac{C_{avg}}{\gamma(C_x)} - 1\right) = \left(\dfrac{T}{Q} - 1\right)^{n} \end{cases}$$

Case study 1 - Uniform density

- Uniform density function:

$$f_C(c) = \begin{cases} \frac{1}{C_{\max} - C_{\min}} & \text{if } C_{\min} \le c \le C_{\max} \\ 0 & \text{otherwise} \end{cases}$$

- $x = \frac{C_x}{C_{\max}}$ and $a = \frac{C_{\min}}{C_{\max}}$.
- When n=2

$$x_{\text{opt}} = \frac{1 + \sqrt{1 + 3\,a^2}}{3}.$$

- When n=3

$$x_{\text{opt}} = \frac{5 - \sqrt{5} + \sqrt{2}\sqrt{5(3 - \sqrt{5}) - 8(1 - \sqrt{5})\,a^2}}{8}$$

Case study 1 - Uniform density

- Uniform density function:

$$f_C(c) = \begin{cases} \frac{1}{C_{\max} - C_{\min}} & \text{if } C_{\min} \leq c \leq C_{\max} \\ 0 & \text{otherwise} \end{cases}$$

- $x = \frac{C_x}{C_{\max}}$ and $a = \frac{C_{\min}}{C_{\max}}$.

- When n=2

$$x_{\text{opt}} = \frac{1 + \sqrt{1 + 3\,a^2}}{3}.$$

- When n=3

$$x_{\text{opt}} = \frac{5 - \sqrt{5} + \sqrt{2}\sqrt{5(3 - \sqrt{5}) - 8(1 - \sqrt{5})\,a^2}}{8}$$

Case study 1 - Uniform density

- Uniform density function:

$$f_C(c) = \begin{cases} \frac{1}{C_{\max} - C_{\min}} & \text{if } C_{\min} \le c \le C_{\max} \\ 0 & \text{otherwise} \end{cases}$$

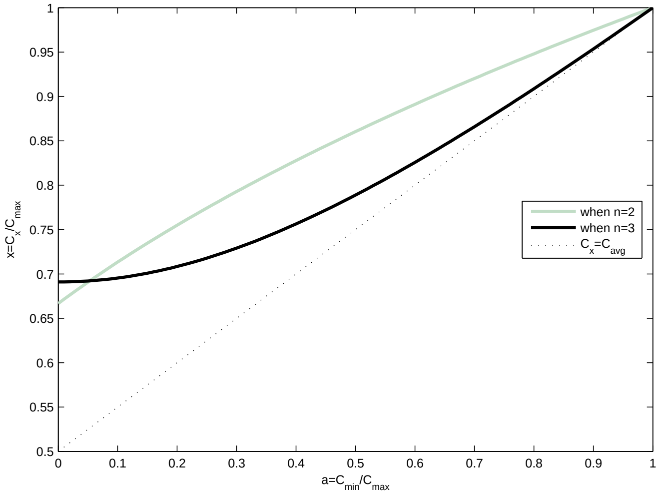
- $x = \frac{C_x}{C_{\max}}$ and $a = \frac{C_{\min}}{C_{\max}}$.
- When n=2

$$x_{\text{opt}} = \frac{1 + \sqrt{1 + 3\,a^2}}{3}.$$

- When n=3

$$x_{\text{opt}} = \frac{5 - \sqrt{5} + \sqrt{2}\sqrt{5(3 - \sqrt{5}) - 8(1 - \sqrt{5})\,a^2}}{8}$$

Case study 1 - Uniform density (2)



Case study 2 - Exponential density

- What happens for asymmetric densities ?

- Exponential density:

$$f_C(c) = \begin{cases} \frac{1}{K} e^{\beta c}(1-c)(c-a) & \text{if } c \in [a,1] \\ 0 & \text{otherwise} \end{cases}$$

K such that $\int_a^1 f_C(c)\mathrm{d}c = 1$.

- β allows to alter the symmetry of the density
  - $\beta < 0$: values close to $C_{\min}$ are more likely to happen
  - $\beta > 0$: values close to $C_{\max}$ are more likely to happen

Case study 2 - Exponential density

- What happens for asymmetric densities ?
- Exponential density:

$$f_C(c) = \begin{cases} \frac{1}{K} e^{\beta c}(1-c)(c-a) & \text{if } c \in [a,1] \\ 0 & \text{otherwise} \end{cases}$$

K such that $\int_a^1 f_C(c)\mathrm{d}c = 1$.

- β allows to alter the symmetry of the density
  - $\beta < 0$: values close to $C_{\min}$ are more likely to happen
  - $\beta > 0$: values close to $C_{\max}$ are more likely to happen

Case study 2 - Exponential density

- What happens for asymmetric densities ?

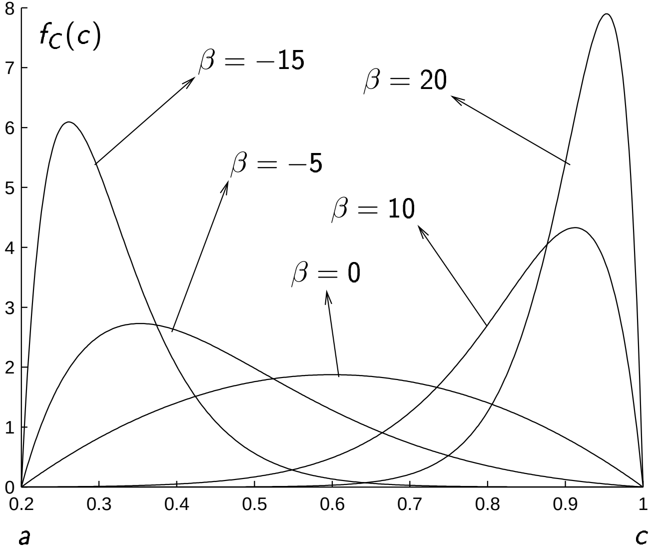
- Exponential density:

$$f_C(c) = \begin{cases} \frac{1}{K} e^{\beta c}(1-c)(c-a) & \text{if } c \in [a,1] \\ 0 & \text{otherwise} \end{cases}$$

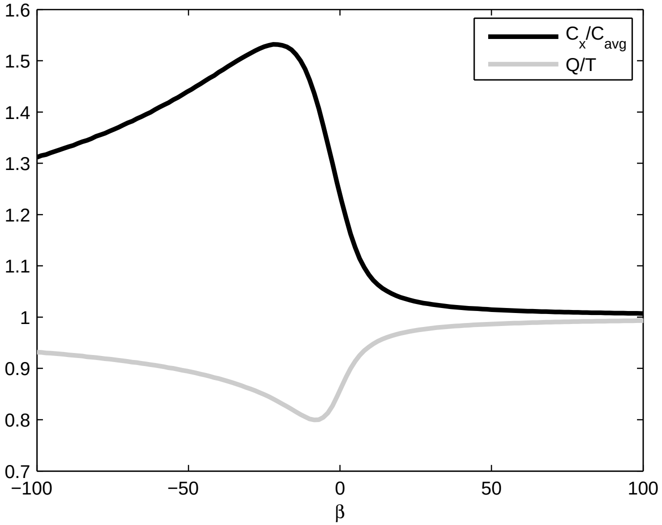
K such that $\int_a^1 f_C(c)\mathrm{d}c = 1$.

- β allows to alter the symmetry of the density
  - $\beta < 0$: values close to C_{min} are more likely to happen
  - $\beta > 0$: values close to C_{max} are more likely to happen

Case study 2 - Exponential density (2)



Case study 2 - Exponential density (3)



Conclusions

- Found the optimal solution with 2 speeds
  - Analytical expression
  - Very general model
    - No specific power functions
    - Time and energy overheads for voltage transition
- Applied to 2 case studies: uniform and exponential densities
- Still a work-in-progress
  - Basis for the design of new power-aware algorithms
  - **Future work**: see how this result can be extended to the case of *n* concurrent tasks