Power Aware Scheduling

Motivations

- PDAs and Pocket PCs
  - Display, processor, hard disk, and wireless LAN card are main sources of power consumption
  - Processor may consume up to 25% of the system power for laptop computers
  - Technology advance in processor architecture is much faster than battery technology
    - Powerful processors
    - Limited battery power
- Idea: Trade spare time against power saving with **DVS (Dynamic Voltage Scaling)**

Dynamic Voltage Scaling Processors

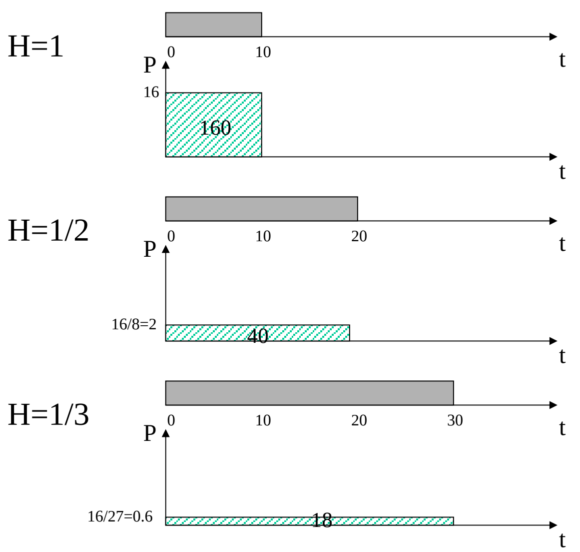
- Crusoe processor of Transmeta, Intel StrongARM processor
- Possible clock frequency (processor speed H) is proportional to supply voltage

$$f \propto V$$

- DVS can dynamically adjust supply voltage and thus processor speed
- Power consumption rate (P) vs. supply voltage (V) and clock frequency (f)

$$P = C \cdot f \cdot V^2 \propto V^3$$

Optimal Voltage? (0<V≈H ≤ 1)



How to determine optimal speed in real-time systems?

- Minimize the energy consumption while guaranteeing the schedulability
- Theorem: n periodic tasks $\{(p_i, e_i, NPS_i) \mid 1 \le i \le n\}$ can be feasibly scheduled by EDF at processor speed H (0<H≤1) if

$$\forall i, 1 \le i \le n, \sum_{j=1}^{n} \frac{e_j}{p_j} + \frac{\max\{NPS_j \mid p_j > p_i\}}{p_i} \le H$$

- Proof:

$$\forall i, 1 \le i \le n, \sum_{j=1}^{n} \frac{e_j / H}{p_j} + \frac{\max\{NPS_j \mid p_j > p_i\} / H}{p_i} \le 1$$

Optimization Problem

Minimize H

Subject to

$$\sum_{j=1}^{n} \frac{e_j}{p_j} + \frac{\max\{NPS_j \mid p_j > p_i\}}{p_i} \le H \quad \text{for all } 1 \le i \le n$$

We change H only when a existing task leaves or a new task is admitted

Further Energy Saving

- First, while the blocking is not ongoing, the total utilization is lower than 1. So, we can further reduce the speed for spare times
- Second, the speed H was determined assuming WCET. If jobs execute much less than WCET, we can reclaim the unused time and the unused time can be used for further reduction of speed

Dual-Speed Switching Algorithm

- If the tasks are fully preemptible, they can be feasibly scheduled with a minimum speed L

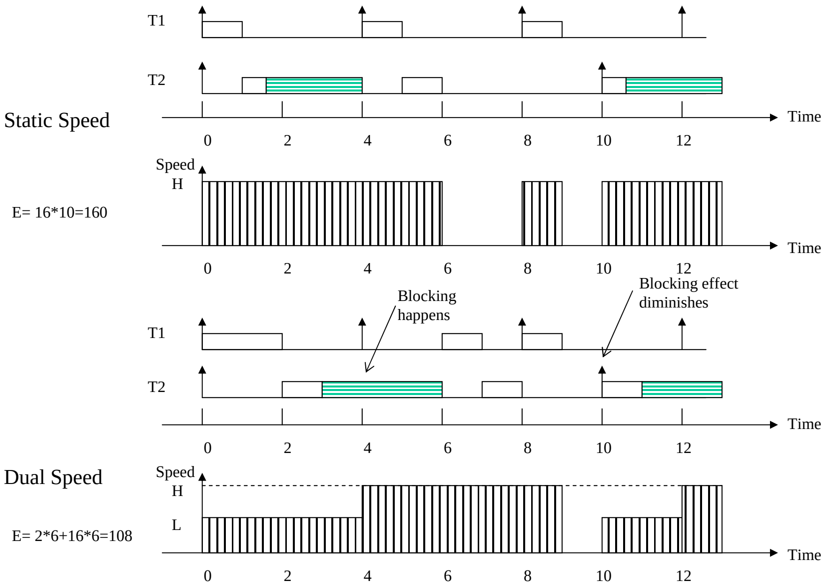
$$\sum_{j=1}^{n} \frac{e_j}{p_j} \leq L$$

- If a high priority job is blocked by a low priority job, the processor runs in <u>High speed</u> (H) mode until the blocking effect completely diminishes.

$$\forall i, 1 \leq i \leq n, \sum_{j=1}^{n} \frac{e_j}{p_j} + \frac{\max\{NPS_j \mid p_j > p_i\}}{p_i} \leq H$$

- In all other situations, the processor run in <u>Low speed</u> (L) mode

Dual-Speed Switching Algorithm



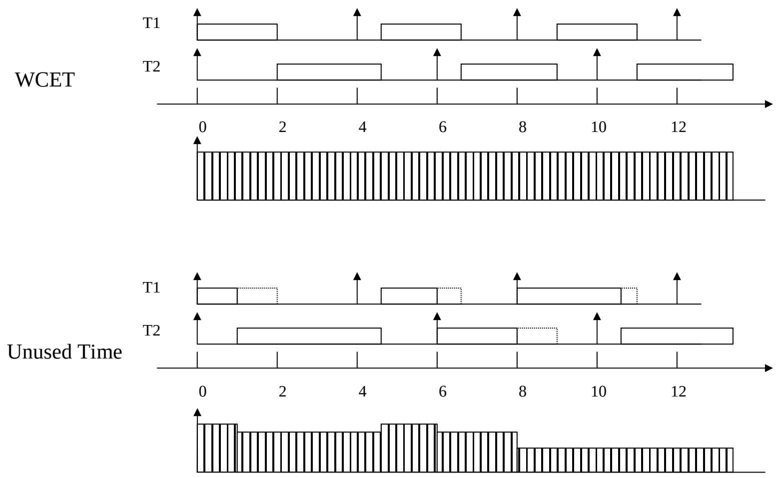
Reclaiming unused time

- When a job (whose deadline is *d*) finishes, the unused time (WCET – actualExeTime) is spare time for jobs whose deadline is later than *d* (Why NOT for jobs with deadlines earlier than *d*?)
- If we allocate such unused time *r* to a job *J* with remaining execution time of *e* (and deadline later than *d*),

$$\frac{e}{L} = e + r,$$

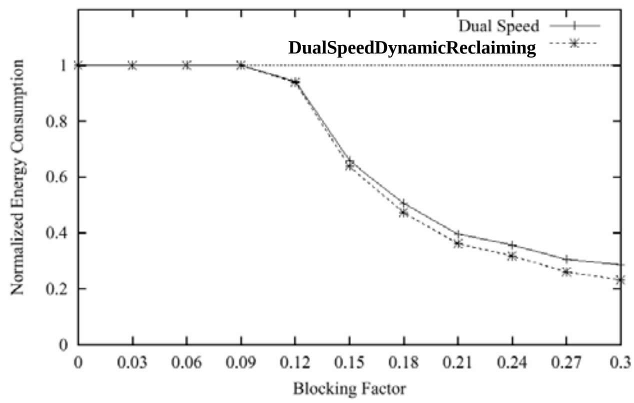
$$L = \frac{e}{e + r}$$

Reclaiming unused time

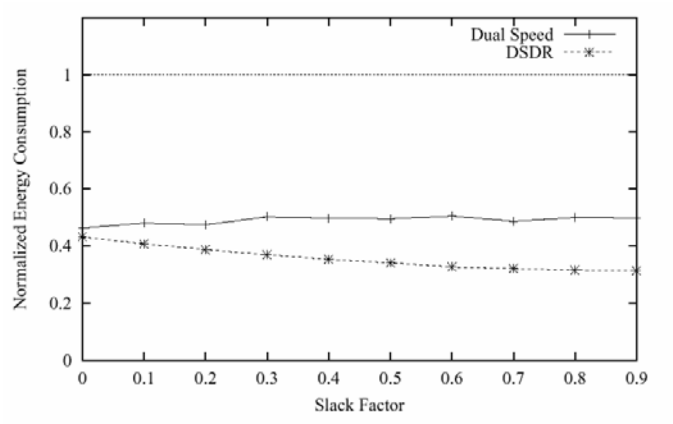


Normalized Energy Saving Relative to Static Speed

(- blocking factor effect -)



Normalized Energy Saving Relative to Static Speed
(- slack factor effect -)



Summary

- Power aware scheduling is a hot research topic
  - *Embedded processors with DVS*
  - *Radar scheduling with Energy budget constraints*
  - *Sensor network – power aware processing and routing*
- We saw an example where simple RT-theory can be effectively used for solving a very important and practical problem
- References
  - Hakan Aydin et al., Dynamic and Aggressive Scheduling Techniques for Power-Aware Real-Time Systems, RTSS 2001
  - Fan Zhang et al., Processor Voltage Scheduling for Real-Time Tasks with Non-Preemptible Sections, RTSS 2002