NEWS Lab

Flow Time Minimization under Energy Constraints

Jian-Jia Chen[*#], Kazuo Iwama[$], Tei-Wei Kuo[#], and Hseuh-I Lu[#]

[#]Department of Computer Science and Information Engineering,
National Taiwan University
*Fisheries Agency, Council of Agriculture,
Executive Yuan, Taiwan
[$]School of Informatics, Kyoto University,
YoshidaHonmachi, Kyoto, Japan

Agenda

- Introduction
- System Models and Preliminary Results
- Our Algorithms
  - Flow Time Minimization
    - Ideal Processors
    - Non-Ideal Processors
  - Weighted Flow Time Minimization
- Performance Evaluation
- Conclusion

Introduction

- Energy-efficiency is important in system designs

Hardware Methodology for Power Saving

- Dynamic power management (DPM)
  - The operation mode of the system
  - ACPI
- Micro-architecture technique
  - Adaptive architecture
  - Cache management
- Dynamic voltage scaling (DVS)
  - Supply voltage scaling
    - Intel Xscale, StrongARM; Transmeta Crusoe
    - Intel SpeedStep, AMD PowerNow!
  - Threshold voltage scaling

Dynamic Voltage Scaling

- A higher supply voltage usually results in a higher frequency (or higher execution speed)
  - $s = k * (V_{dd}-V_t)^2/(V_{dd})$, where
    - s is the corresponding speed of the supply voltage V_{dd} and
    - V_t is the threshold voltage
- The dynamic power consumption function $P_d()$ of the execution speeds of a processor is a convex function:
  - $P_d(s) = C_{ef}\, V_{dd}^{\,2}\, s$, in which C_{ef} is the switch capacitance related to tasks under executions
  - $P_d(s) = C_{ef}\, s^3/k^2$, when $V_t = 0$
- The static power consumption comes from the leakage current
  - A constant or
  - A linear function of the supply voltage

Energy-Efficient Scheduling versus Energy-Constrained Scheduling

- *Energy-efficient* scheduling is to minimize the energy consumption while the performance index or the timing constraint is guaranteed

- *Energy-constrained* scheduling is to maximize the performance or system rewards under a specified energy constraint

Performance Maximization

- Reward Maximization
  - Rusu et al. (RTSS'02)
  - Kang et al. (RTSS'02)
  - Rusu et al. (ECRTS'03)
  - Chen et al. (SAC'04)
  - AlEnawy and Aydin (ECRTS'04)
  - Chen and Kuo (RTSS'05)
- Flow Time Minimization
  - Albers and Fujiwara (STACS'06)
  - Pruhs et al. (SWAT'04)
- Completion Time Minimization

Agenda

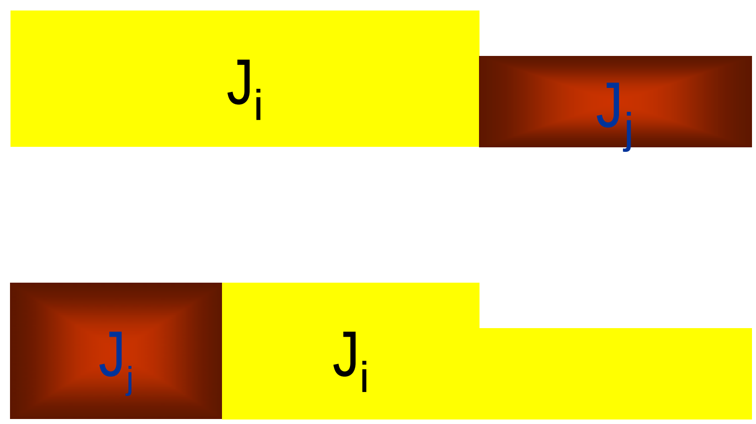
- Introduction
- System Models and Preliminary Results
- Our Algorithms
  - Flow Time Minimization
    - Ideal Processors
    - Non-Ideal Processors
  - Weighted Flow Time Minimization
- Performance Evaluation
- Conclusion

System Models

- Processor model
  - Ideal processors: $s_{min} \sim s_{max}$
  - Non-ideal processors: $(s_{min} = s_1, s_2, \ldots, s_M = s_{max})$
  - Power consumption function: P(s)
    - P(s) is a convex and increasing function
    - P(s)/s is an increasing function
- Job model
  - Each job J_j is associated with its computation requirement in CPU cycles: c_j
  - The flow time of a job: the interval length during the release time and the completion time of the job
  - Each job arrives at the same time: 0
- Our objective
  - Find a schedule for a given job set $J=(J_1, J_2, \ldots, J_N)$ such that the energy consumption is no more than the energy constraint E_b, and the average flow time of these N jobs is minimized

Execution Behavior for Optimal Solutions

There exists an optimal schedule which executes jobs in J in a non-decreasing order of their CPU execution cycles for both ideal and non-ideal processors



We index jobs so that the execution cycles of the jobs in J are in a non-decreasing order, i.e., shorted job first

A Motivational Example

$c_1 = 5$

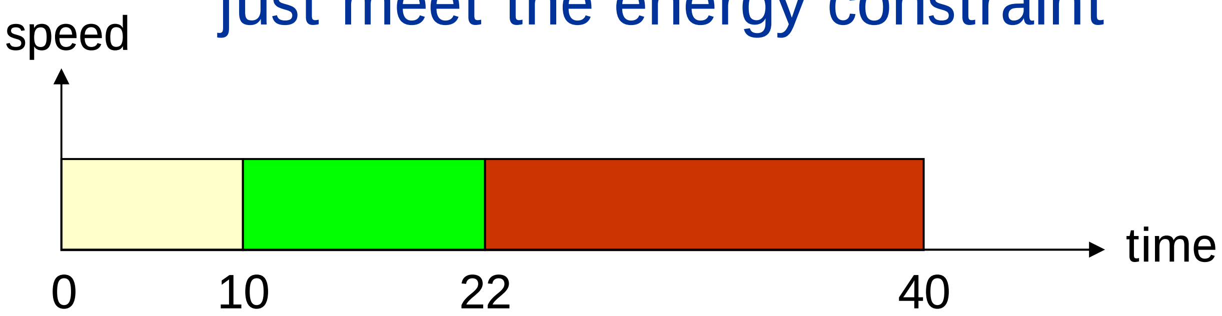
$c_2 = 6$

$c_3 = 9$

$P(s) = s^3$

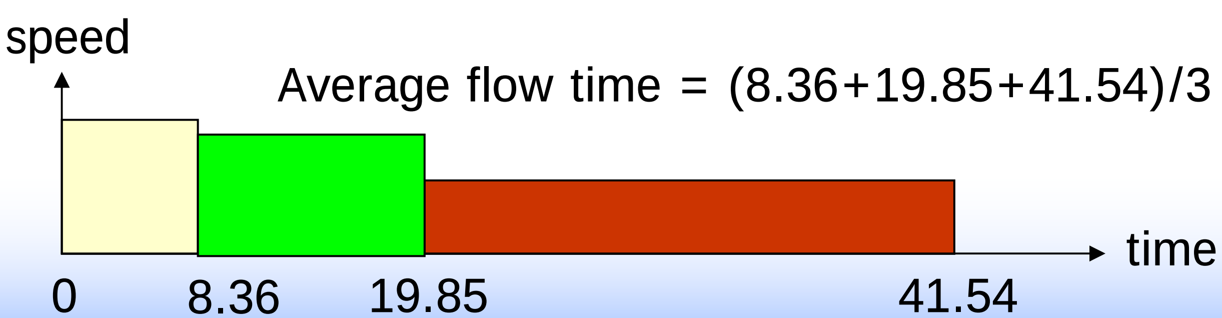
$E_b = 5$

Execute jobs at the speed that just meet the energy constraint



Average flow time = (10+22+40)/3 = 24

Average flow time = (8.36+19.85+41.54)/3 = 23.25



Agenda

- Introduction
- System Models and Preliminary Results
- Our Algorithms
  - Flow Time Minimization
    - Ideal Processors
    - Non-Ideal Processors
  - Weighted Flow Time Minimization
- Performance Evaluation
- Conclusion

Scheduling for Ideal Processors

- An optimal solution executes job J_i at a speed: r_i
- The energy consumption is $\sum_{j=1}^{N} P(r_j)\frac{c_j}{r_j}$
- Executing jobs from J_1 to J_N consecutively leads to a solution
  - The flow time of job J_j is $\sum_{i=1}^{j} c_i/r_i$
- The average flow time is $\sum_{j=1}^{N}\sum_{i=1}^{j}\frac{c_i}{r_i} = \sum_{j=1}^{N}(N-j+1)\frac{c_j}{r_j}$.
- There exists an optimal schedule with $r_1 \geq r_2 \geq \ldots \geq r_N$

Algorithm LM: Optimal Solutions

$O(N^2 \log N)$

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{N} (N - j + 1)\frac{c_j}{r_j} \\ \text{subject to} \quad & \sum_{j=1}^{N} P(r_j)\frac{c_j}{r_j} = E_b, \text{ and} \\ & s_{\min} \le r_j \le s_{\max}, \text{ for } j = 1, 2, \ldots, N. \end{aligned}$$

Relax the speed constraint

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{N} (N - j + 1)\frac{c_j}{r_j} \\ \text{subject to} \quad & \sum_{j=1}^{N} P(r_j)\frac{c_j}{r_j} = E_b, \end{aligned}$$

Apply Lagrange Multiplier Method

Check if the solution violates the speed constraint

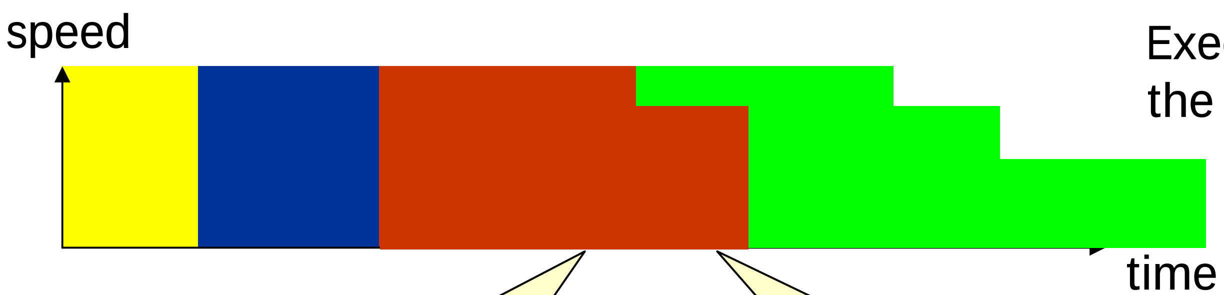
Yes! Form a sub-problem

No!

Return solution

Algorithm Greedy for Non-Ideal Processors

O(NM log NM)



Execute all the jobs at the highest speeds

Repeated until the energy constraint is satisfied

Slow down the job J_j that is most energy-flow-time efficient

Weighted Flow-Time Minimization

- If the execution order is determined
  - Apply simple revisions of Algorithm LM or Algorithm Greedy to determine the execution speeds
  - The weighted flow time is

$$\sum_{j=1}^{N} \left(\sum_{i=j}^{N} w_i \right) \frac{c_j}{r_j}$$

- Otherwise,
  - Applying the well-known weighted shortest-job-first strategy as the execution order, i.e., c_i/w_i in an increasing order

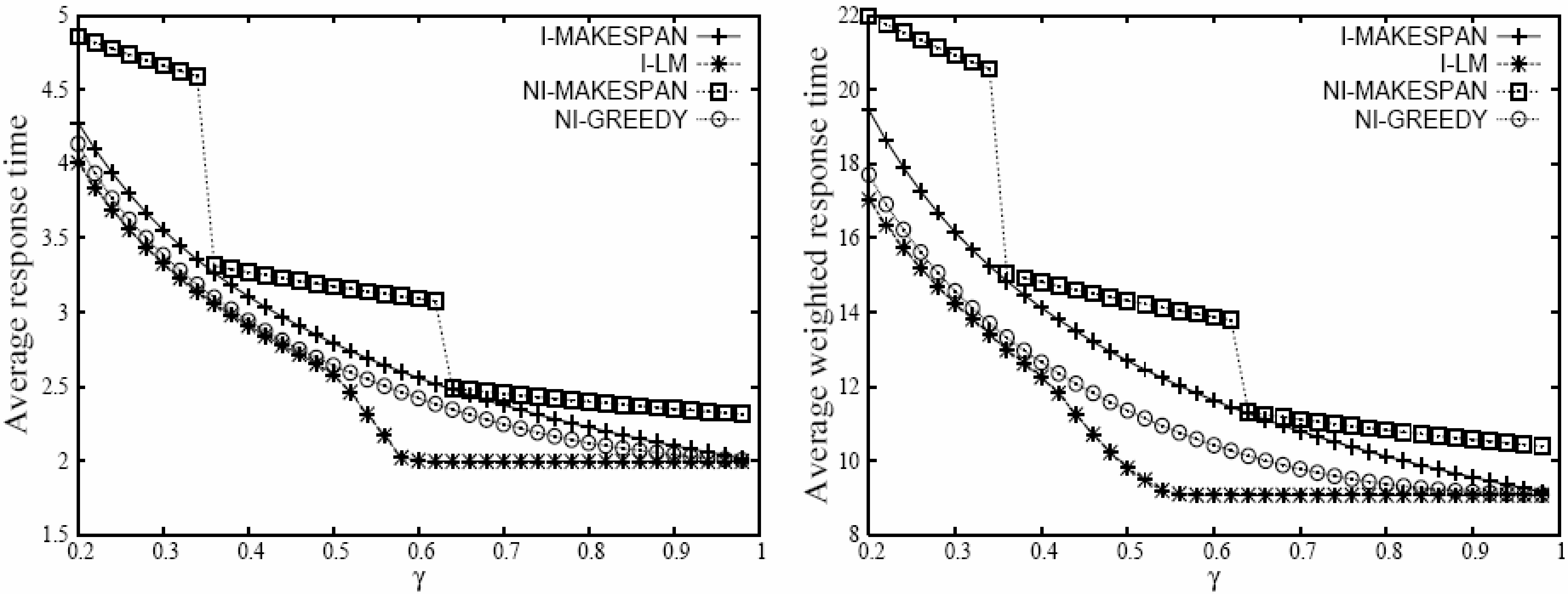
Agenda

- Introduction
- System Models and Preliminary Results
- Our Algorithms
  - Flow Time Minimization
    - Ideal Processors
    - Non-Ideal Processors
  - Weighted Flow Time Minimization
- Performance Evaluation
- Conclusion

Evaluation Setup

- Processor:
  - Intel XScale: 150, 400, 600, 800, 1000 MHz with 80, 170, 400, 900, 1600 mWatt
    - Normalized so that the highest speed is 1
  - Ideal processor approximation: s_{min}=0.15, s_{max}=1 with $P(s)= 0.08 + 1.52\ s^3$
- Jobs:
  - Execution cycle is a random variable in (0, 1]
  - The weight of a job is a random variable in (0.1, 10.1]

Evaluation Results(1)

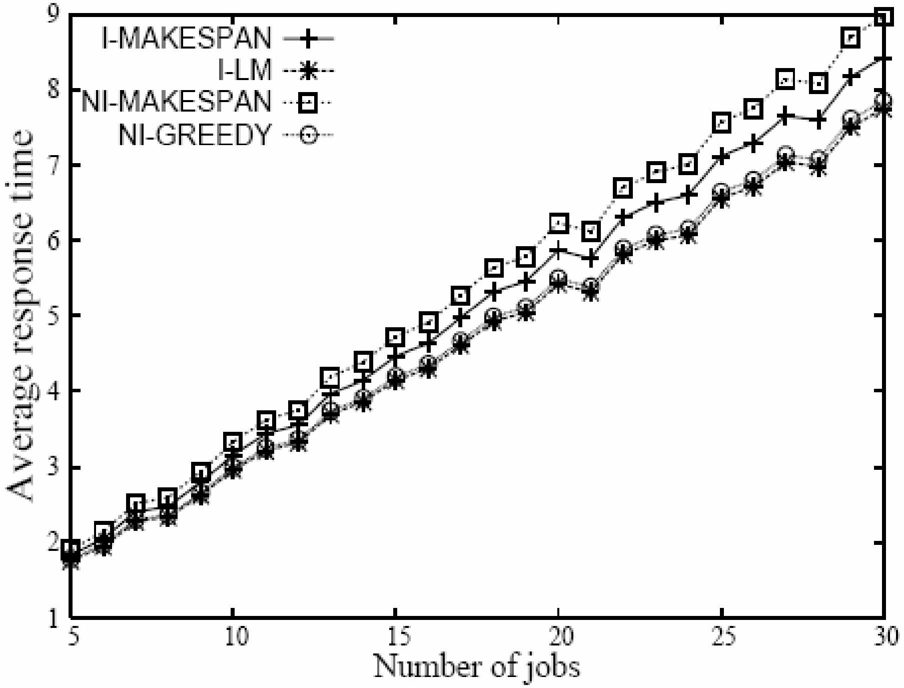


(a) unweighted cases (b) weighted cases

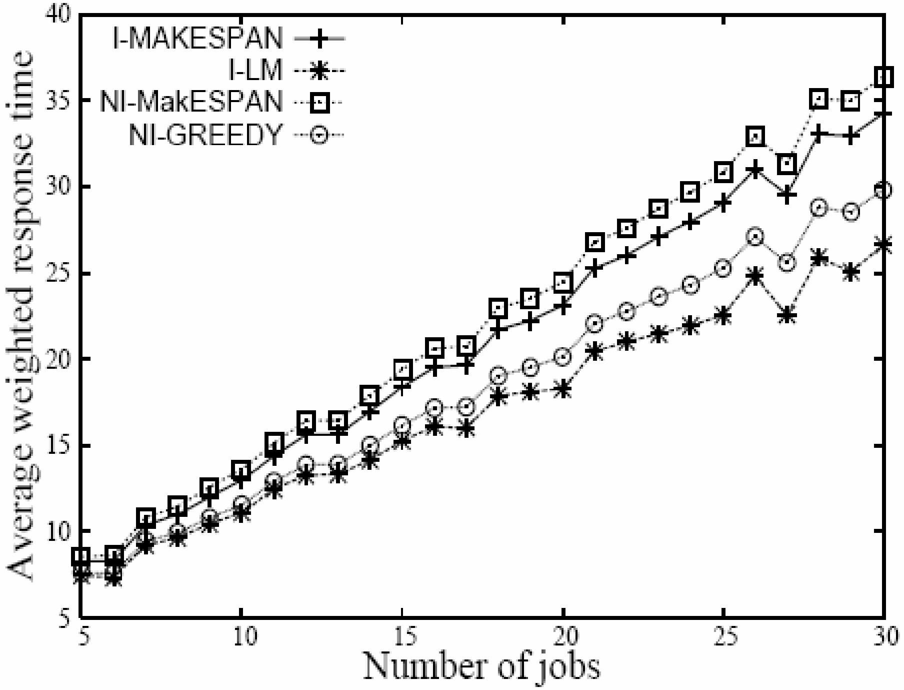
Emax (Emin, respectively) is the energy consumption by executing all the jobs at speed s_{max} (s_{min}, respectively).

Energy consumption constraint E_b is set as Emin + γ (Emax − Emin).

Evaluation Results(2)



(a) unweighted cases



(b) weighted cases

γ is 0.4

Conclusion and Future Work

- Conclusion
  - Optimal scheduling algorithms
    - Flow time minimization under energy constraints in ideal and non-ideal processors
  - Heuristic scheduling algorithms
    - Weight flow-time minimization under energy constraints
- Future work
  - Minimization of the average flow time for jobs with different arrival times
  - Worst-case analysis for the minimization of the average weighted flow time under a given energy constraint

Questions and Suggestions?