香 港 中 文 大 學
The Chinese University of Hong Kong

CENG5030

Part 1-3: DVFS

Bei Yu

(Latest update: March 25, 2019)

Spring 2019

These slides contain/adapt materials developed by

- Frances F Yao (2007). “Algorithmic problems in scheduling jobs on variable-speed processors”. In: *Proceedings of Combinatorial Pattern Matching*, pp. 3–3

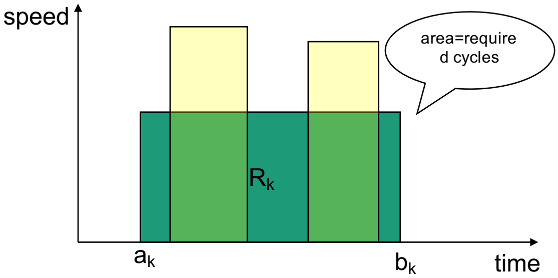
DVS Scheduling Model

- A set of n jobs

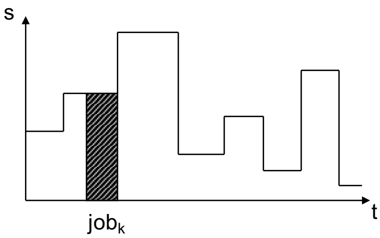
  a_k: arrival time

  b_k: deadline

  R_k: required CPU cycles
- Preemptive execution



- Schedule S specifies:
  - $0 \le s(t) < \infty$
  - which job is executed at time t
- Cost $E(S) = \int_0^1 s(t)^2 \, dt$



- What's the optimal (Min-Energy) schedule
  - Good characterization
    → efficient computation
  - Benchmark for heuristics

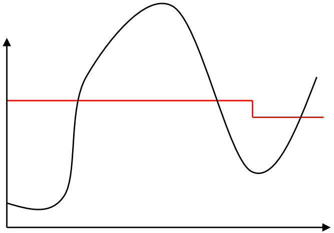
The Basics

- Each job will be executed at one uniform speed in optimal schedule
  - Convexity: $$s^2 + s'^2 > (\frac{s+s'}{2})^2 + (\frac{s+s'}{2})^2$$
- Optimal schedule needs at most n different speeds
  - the flatter the better
- Strategy:

  Determine peak speed s^*,

  apply iterative procedure

  to find 2nd peak speed etc.



Naive Algorithms

Created two algorithms that guarantee a deadline feasible solution with little regard for energy consumption:

Naive 1

Run each job, j, at a speed such that the completion time of job j is $\min[a_{j+1}, b_j]$.

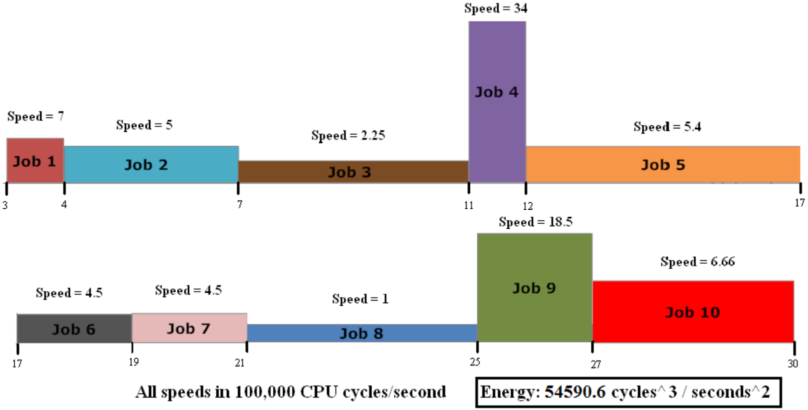
Naive 2

Find the minimum speed necessary to complete every job before its deadline. Run every job at that speed.

Case Study

Table 1: Data Set 1 – Medium Length Jobs

Job	1	2	3	4	5	6	7	8	9	10
a_j	3	4	7	11	12	17	19	21	25	27
b_j	12	11	20	18	19	30	39	48	30	30
R_j	7	15	11	34	27	9	9	4	37	20

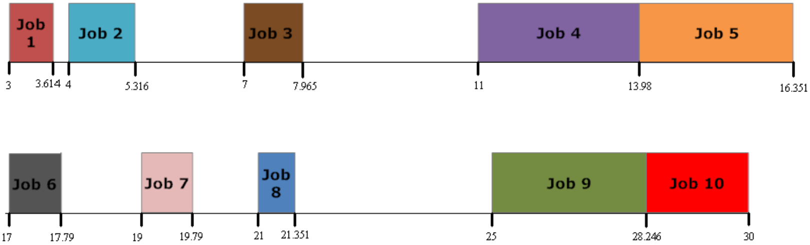


Naïve 1 Schedule

Case Study

Table 1: Data Set 1 – Medium Length Jobs

Job	1	2	3	4	5	6	7	8	9	10
a_j	3	4	7	11	12	17	19	21	25	27
b_j	12	11	20	18	19	30	39	48	30	30
R_j	7	15	11	34	27	9	9	4	37	20



All jobs are run at a speed of 11.4
(1,140,000 CPU cycles/second)

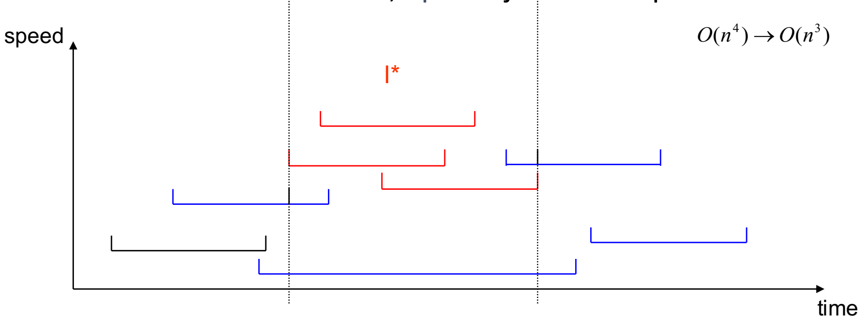
Energy: 22095.8 cycles^3 / second^2

Naïve 2 Schedule

Optimal Scheduling[1]

- What's the peak speed in the optimal schedule?
- $g(I) = \frac{\sum R_j}{|I|}$ defines the speed lower bound over any I
- s* $= \max_I g(I)$ defines peak speed and critical interval I*
- s* over critical interval is feasible
- Extract critical interval, update jobs and repeat

$O(n^4) \rightarrow O(n^3)$

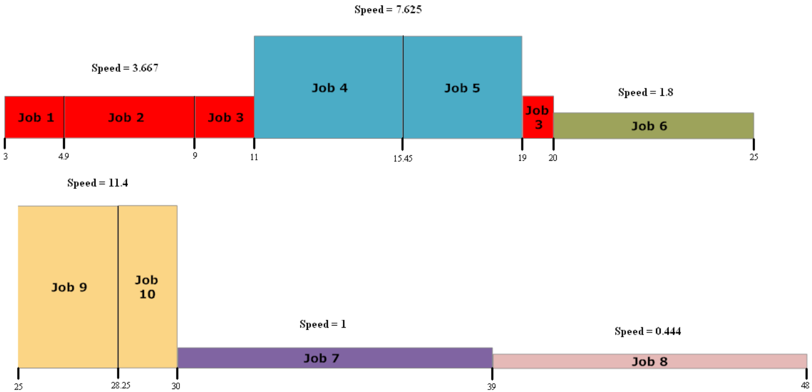


[1] Frances Yao, Alan Demers, and Scott Shenker (1995). "A scheduling model for reduced CPU energy". In: *Proc. FOCS*, pp. 374–382.

Case Study

Table 1: Data Set 1 – Medium Length Jobs

Job	1	2	3	4	5	6	7	8	9	10
a_j	3	4	7	11	12	17	19	21	25	27
b_j	12	11	20	18	19	30	39	48	30	30
R_j	7	15	11	34	27	9	9	4	37	20



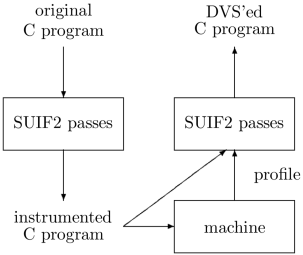
All speeds in 100,000 CPU cycles/second

Energy: 11302.5 cycles^3 / seconds^2

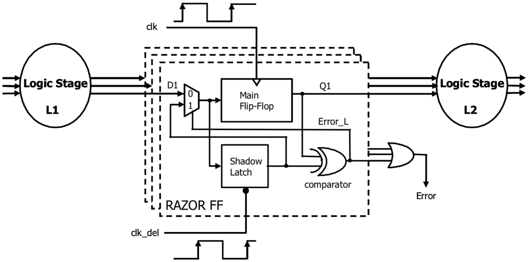
Optimal YDS Schedule

DVFS at Different Levels

- System Level DVFS[2]
- Program Level DVFS[3]
- Hardware Level DVFS[4]



(a) PLDI2003



(b) MICRO2003

[2] Mark Weiser et al. (1994). "Scheduling for reduced CPU energy". In: *Proc. OSDI*.

[3] Chung-Hsing Hsu and Ulrich Kremer (2003). "The design, implementation, and evaluation of a compiler algorithm for CPU energy reduction". In: *Proc. PLDI*. vol. 38. 5, pp. 38–48.

[4] Dan Ernst et al. (2003). "Razor: A low-power pipeline based on circuit-level timing speculation". In: *Proc. MICRO*, p. 7.