ASPDAC 2009

Analyzing and Optimizing Energy Efficiency of Algorithms on DVS Systems:

--A First Step towards Algorithmic Energy Minimization--

TETSUO YOKOYAMA

Nagoya University

Joint work with Gang Zeng,
Hiroyuki Tomiyama and Hiroaki Takada

Optimizing Energy Consumption at Algorithmic Level

Optimizing energy consumption:

- Many HW mechanisms available (DVFS, DPM, ...)
- Important in system level design
- Used in earlier stages of SW development phase

However, fundamental concepts have not yet been completed.

- Differences from performance optimization?
- Metrics?
- Programming logics and structure?
- Dataflow?

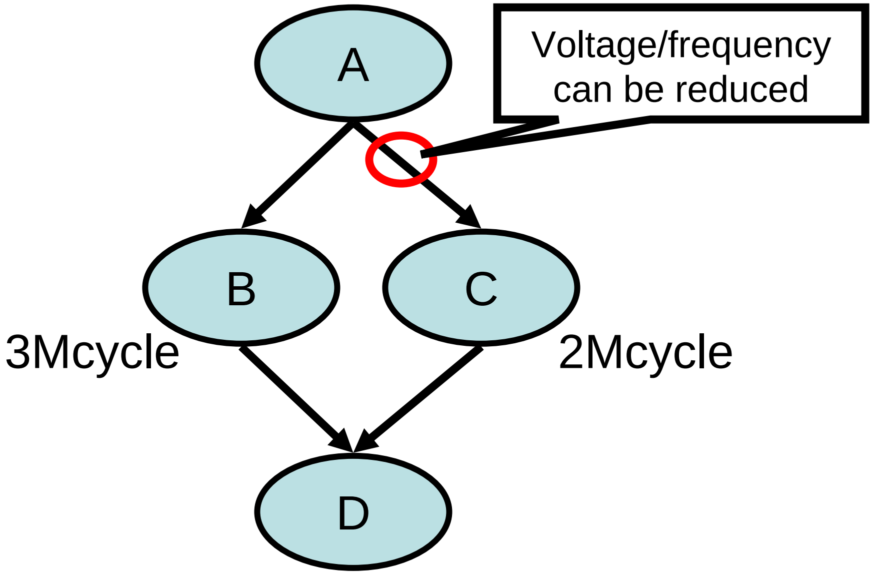
More precisely, we cannot answer:
Which, either Quicksort or Heapsort, is more energy optimal?

Target and Objective

- Target
  - DVS systems
  - Deadline constraints
  - Algorithmic level
- Objective
  - Clarify the difference between energy optimization and performance optimization
  - Propose a measure for energy consumption
  - Study a case of algorithmic energy optimization
  - Answer "Quicksort vs Heapsort. Which is more energy optimal?"

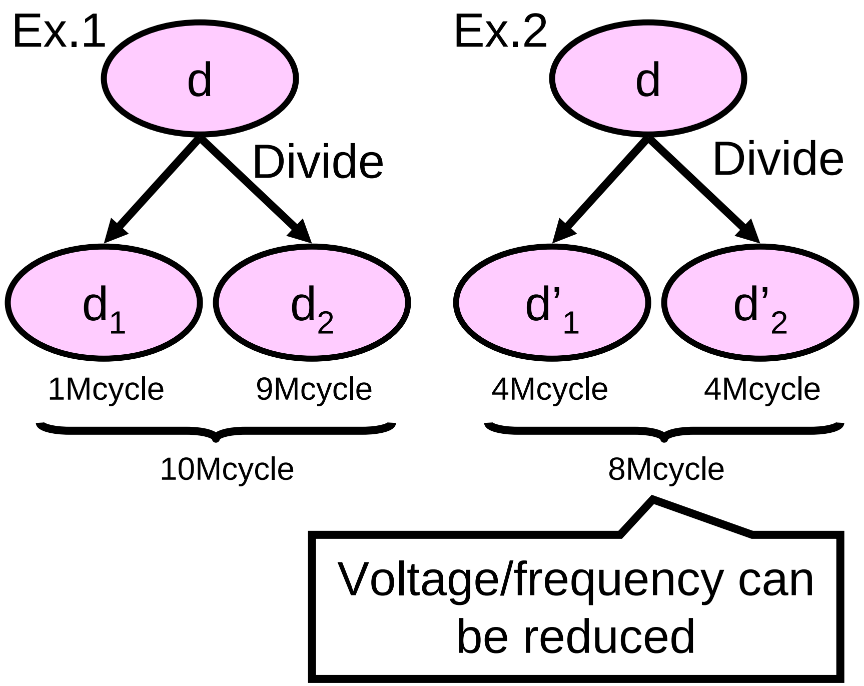
IntraDVS: Basic Concepts

Related work: Control flow graph



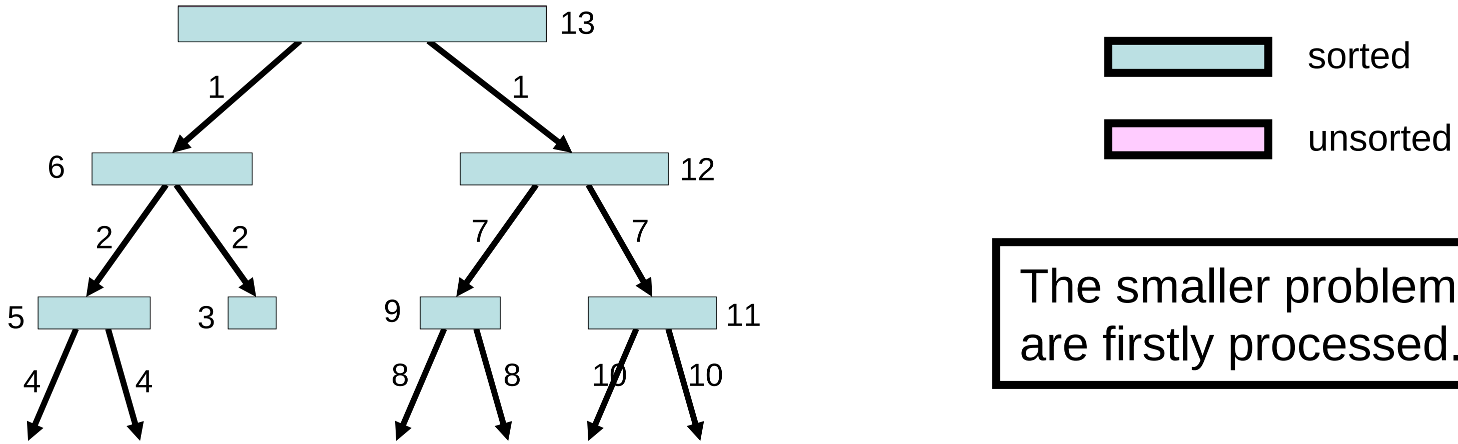
- The selected branch decides the remaining cycles
  - Ex. Either block B or block C is executed

Our approach: *Data* flow graph



- The sizes of divided subproblems decide the remaining cycles

Review of Quicksort



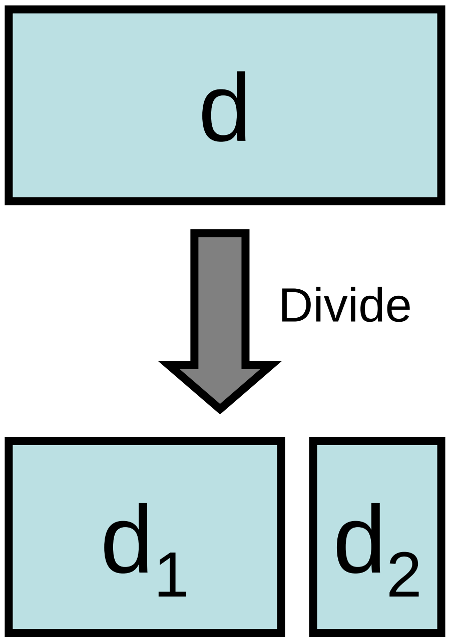
The smaller problems are firstly processed.

- Workload variation → slack time
  - WCEC: $\delta_w(n) \propto n^2$
  - ACEC: $\delta_a(n) \propto n \log n$
- Problem: WCEC is too big!
  → Heapsort does not have much workload variance.

Remaining Predicted Execution Cycles

- <u>WCEC</u>

$$\delta_{\mathsf{W}}(d) = c(d) + \max_{d_1, d_2 \in \mathsf{div}(d)} (\delta_{\mathsf{W}}(d_1) + \delta_{\mathsf{W}}(d_2))$$



$\delta_w(d)$: WCEC of processing d

c(d) + max($\delta_w(d_1)$ + $\delta_w(d_2)$)

Dividing time + Worst of sum of WCECs

Comparison of Remaining WCET of qsort

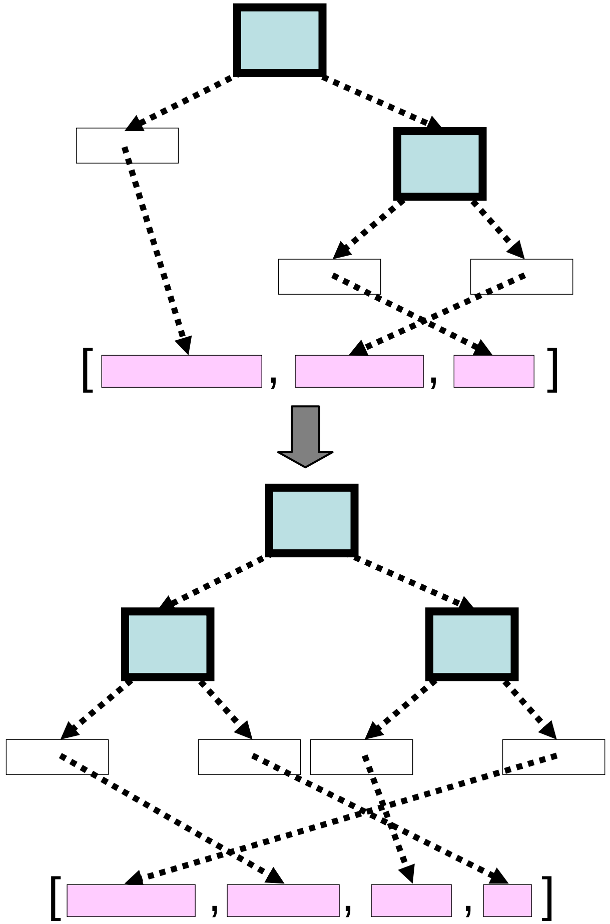
- Remaining WCET w/o concatenation:

$$\sum_{i=1}^{m} \delta_{\mathrm{wc}}(length(\mathrm{xs}_i))$$ (Approximation)

- In ordinary libraries, the smaller problem is firstly processed
  - **PRO**: Memory consumption is bound by O(log n)
  - **CON**: Remaining WCET does not rapidly decrease (early division technique)

	Decrease of RET	Memory Usage
Smallest	Slow	Low
Largest	Moderate	Middle
Larger	Rapid	High

Our approach:
Processing larger problems



Energy-efficient Hybrid Sorting Algorithm

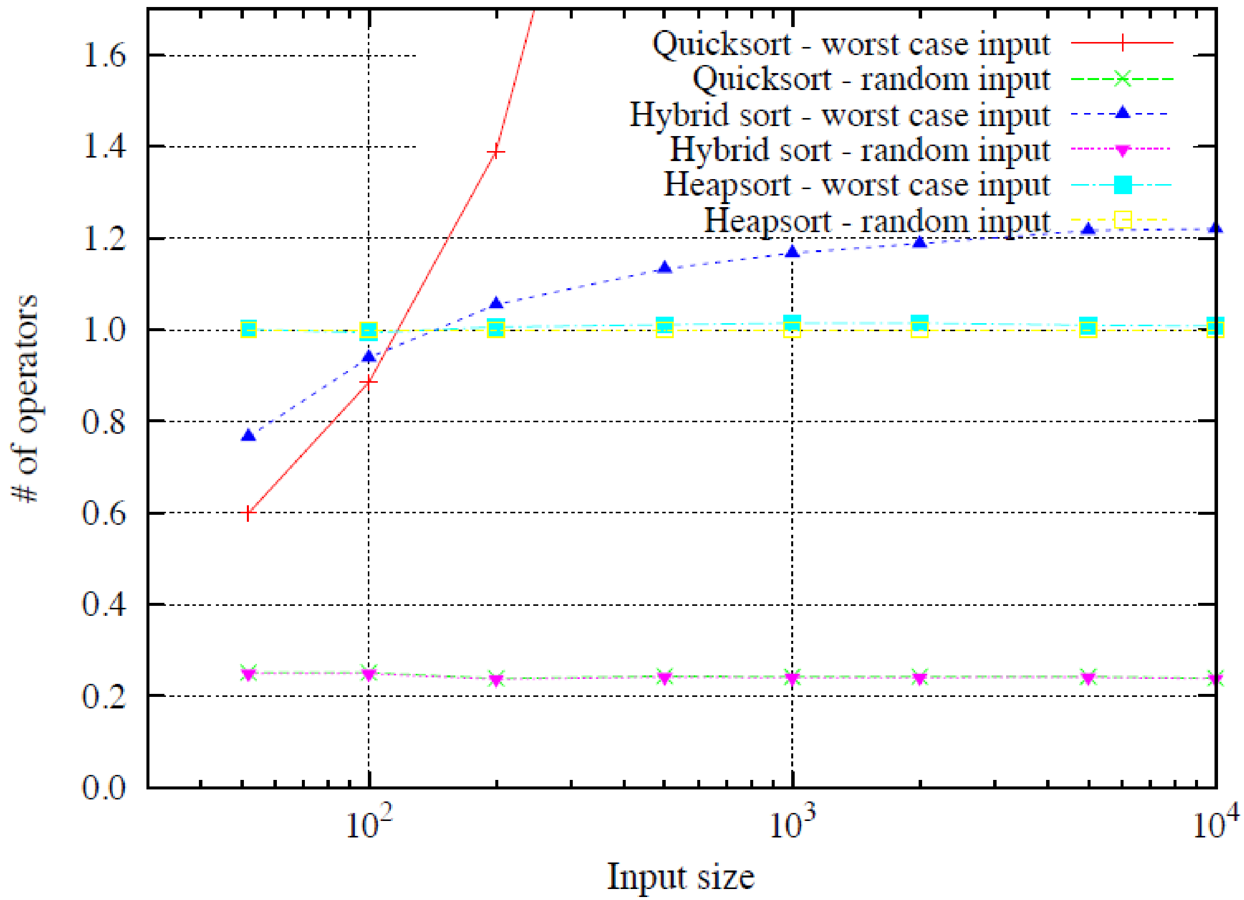
- Energy Efficiency = Original Algo. + Optimization

	Original Algo.	Energy Optimization
Quicksort	$O(n^2)$	Very good
Heapsort	$\Theta(n \log n)$	Bad
Hybrid	$\Theta(n \log n)$	Very good

- Our solution: Hybrid sort = Quicksort + Heapsort
  - First, Quicksort: fast on average
    - Early division technique is used.
    - Performance is very high for almost all of the input.
  - At worst case, changed into Heapsort; WCEC is bounded.

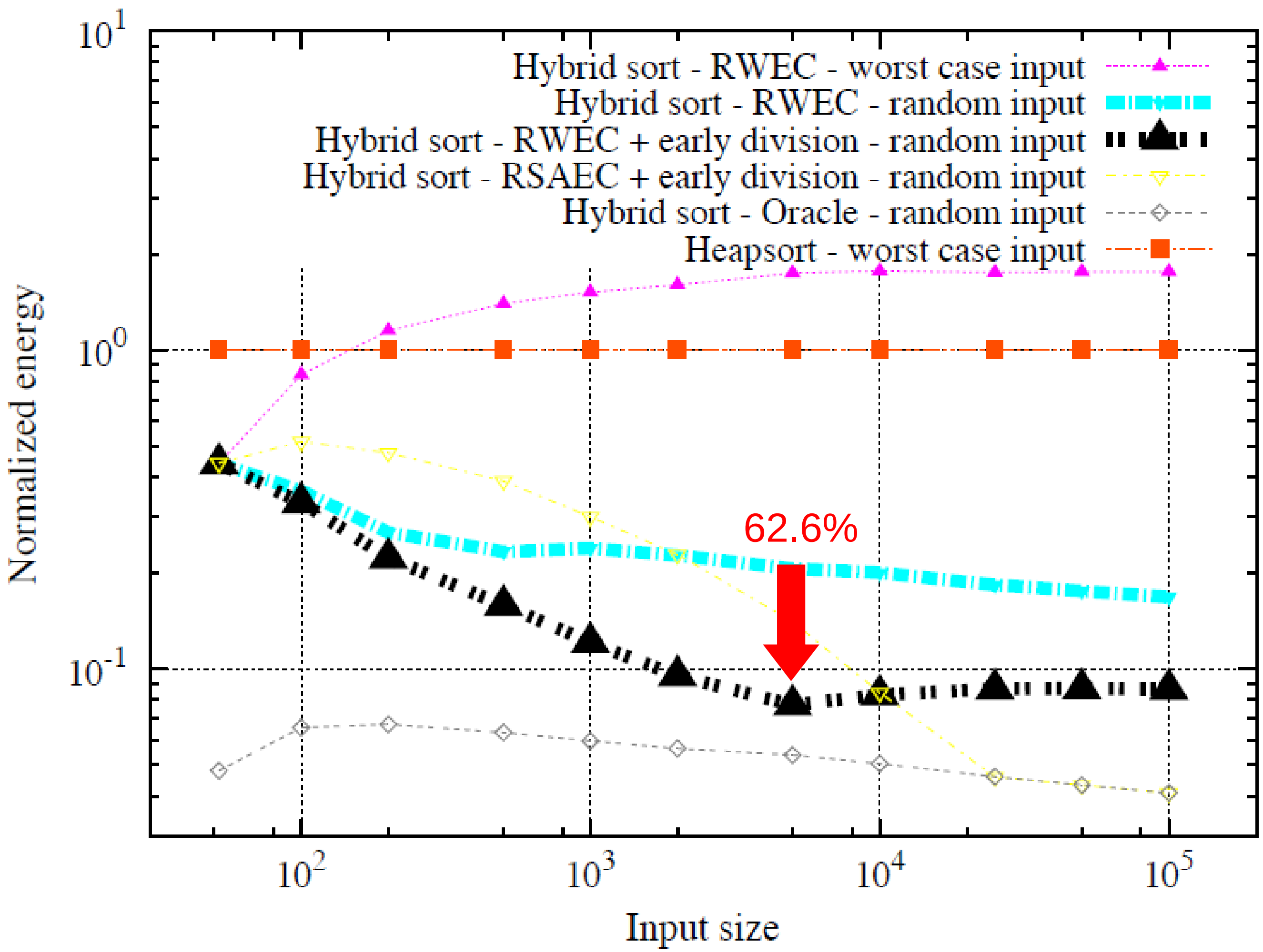
$\Rightarrow$ The energy efficiency of the sorting algorithm can be optimized on DVS systems.

Normalized Operator numbers of Sorting Programs

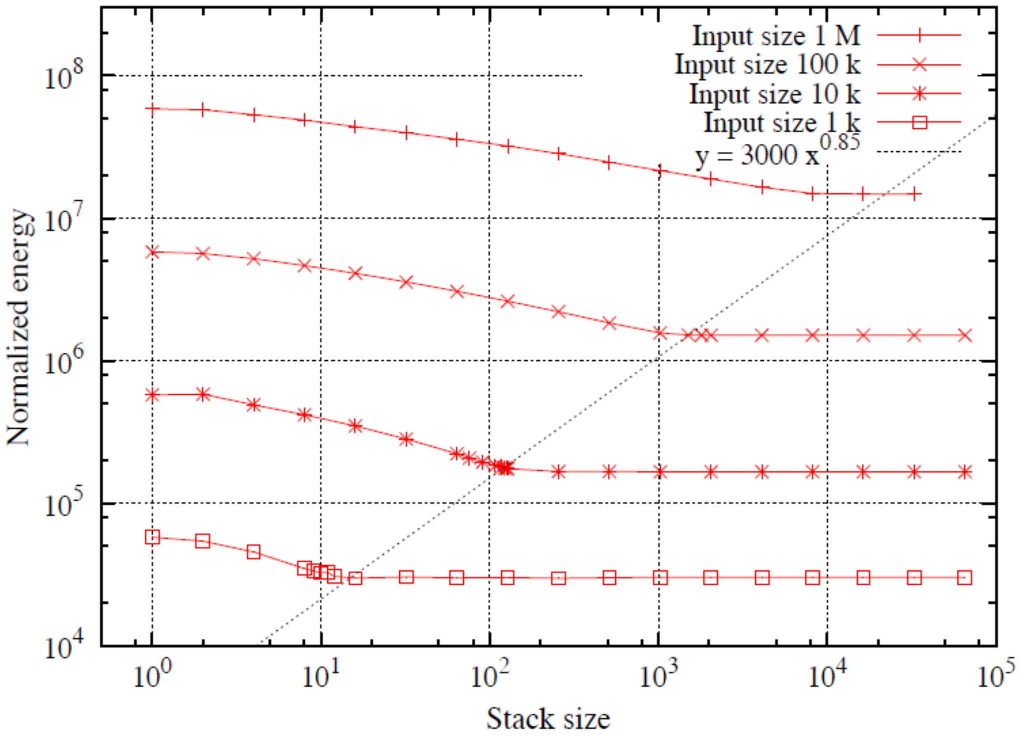


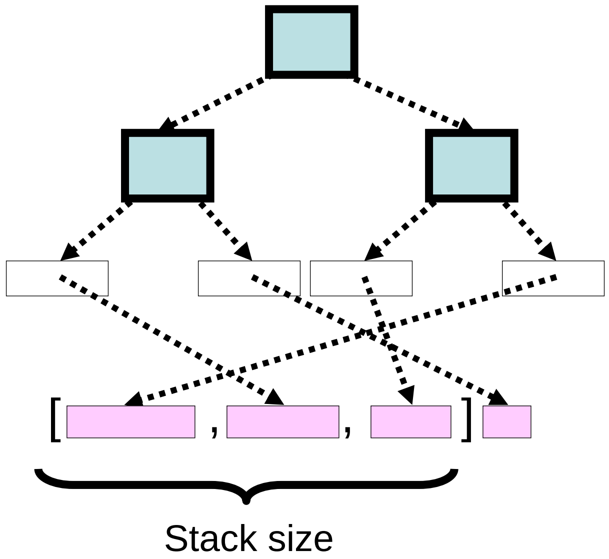
The data was obtained on a MIPS R5000 processor with 512 KB of secondary cache and 64 MB of main memory, using version 7.2.1 of the Silicon Graphics MIPSpro C++ compiler.

Normalized Energy of Sorting Programs Using DVS



Effects of the Stack Size on Energy Consumption Using SVS





Effects of the stack size on energy consumption

- A tradeoff between energy and memory
- The greater stack size it has, the more energy is saved.
- Energy savings saturated to about 25%
- This saturation occurs at line $y = 3000 x^{0.85}$

Measure for Evaluating Energy Consumption of Algorithm

- SVS: voltage scheduling before execution
- The optimal voltage of each task
  = The minimum voltage to finish the task execution exactly at the deadline

Lemma (Optimal SVS) For SVS without scaling bound, An algorithm of a given task is optimal on average iff

$$\delta_w^2 \delta_a$$

is minimum.

Implications:

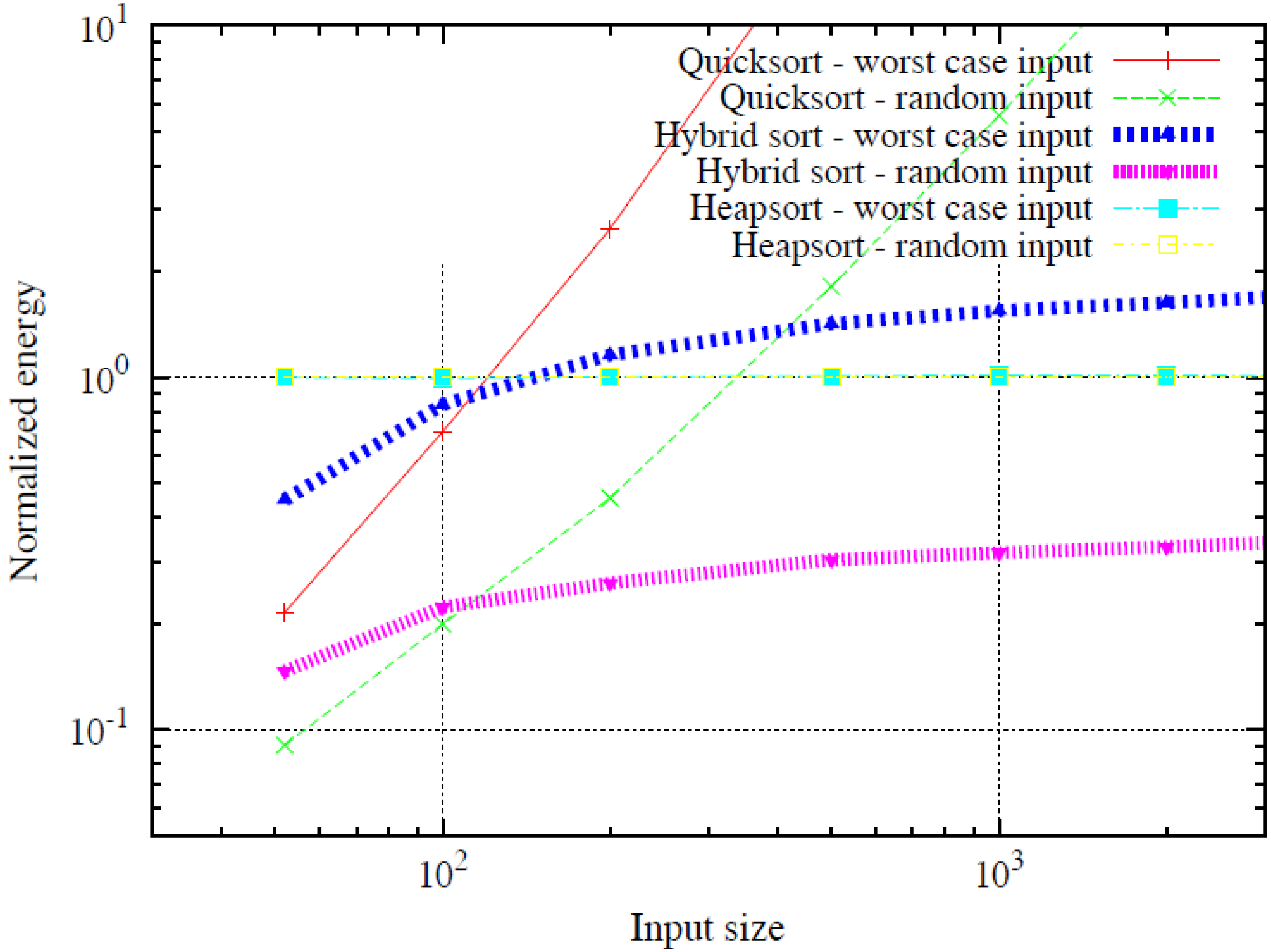
- Deadline does not affect the comparison of two algorithms
- Firstly, WCET δ_w should be reduced
- Secondly, ACET δ_a

Proof

1. Let D be deadline.
2. The execution of frequency δ_w/D finishes exactly on deadline.
3. The corresponding power consumption is $P(\delta_w/D)$, which is the smallest in the case that $f = \delta_w/D$ because of the monotonicity of *P*.
4. The execution time is the number of cycles *X* divided by frequency, *i.e.*, $X/(\delta_w/D)$.
5. Thus, energy consumption becomes $P(\delta_w/D) \cdot DX/\delta_w$
6. Now, we are comparing the different algorithms under the same deadline, and therefore *D* is constant.
7. The average energy consumption is proportional to the average of $(\delta_w)^2 X$.

CAVEAT: If frequencies range over $[f_{min},f_{max}]$,
the objective function becomes $(\delta_w)^2 \cdot \max(Df_{min}, \min(Df_{max}, \delta_a))$

Normalized Energy of Sorting Programs Using SVS



Comparison of Energy Consumption of Sorting: A Case Study

- Question:
  - What is an energy efficient sorting on SVS systems?
- Our Answer†:
  - When the input size is small,
    Energy optimization $\fallingdotseq$ Performance optimization
  - When the input size is large,

Average Energy Average Exec. Time	Hybrid << Heap << Quick Quick = Hybrid << Heap
Worst Energy Worst Exec. Time	Heap < Hybrid << Quick Heap < Hybrid <<< Quick

- Implications:
  - The proposed metrics enables this comparison
  - Energy optimization $\neq$ Performance optimization
  - Fastest on average $\neq$ Energy optimal on average

†Only consider Quicksort, Heapsort and Hybrid sort

Concluding Remarks

- *Algorithmic* energy efficiency is *meaningful*.
- We propose
  - Energy efficient sorting algorithm
  - A measure for evaluating the optimal energy of algorithms
  - IntraDVS strategies using data flow information

  $\Rightarrow$ We can discuss which, either Quicksort or Heapsort, is more energy efficient.

- Future work
  - How to write energy efficient programs (partially published)
  - Compare energy efficiency of other algorithms (ongoing)
  - Expose a tradeoff with energy (ongoing)