System-wide Power Management for Real-Time Systems

Roberto Medina and Liliana Cucu
roberto.medina-bonilla@inria.fr

RTSOPS

Context and motivation

- Reactive embedded systems
- Time constraints
- Battery powered

Context and motivation

- Reactive embedded systems
- Time constraints
- Battery powered

Challenges

- ↗ functionalities
- ↗ autonomy
- ↘ SWaP
- COTS

Energy management for RT systems

Dynamic Voltage and Frequency Scaling (DVFS)

- $\searrow$ voltage $\rightarrow$ $\searrow$ frequency $\rightarrow$ $\nearrow$ execution time.
- Derive a proper speed to meet deadlines.

Energy management for RT systems

Dynamic Voltage and Frequency Scaling (DVFS)

- $\searrow$ voltage $\rightarrow$ $\searrow$ frequency $\rightarrow$ $\nearrow$ execution time.
- Derive a proper speed to meet deadlines.

Dynamic Power Management (DPM)

- CPUs have low-power states, barely consuming energy.
- Know when system can be put into this low-power state.

Execution time

Classic real-time task model: τ_i, C_i, T_i, D_i.

Execution time

Classic real-time task model: τ_i, C_i, T_i, D_i.

How does C_i evolve w.r.t. frequency?

Execution time

Classic real-time task model: τ_i, C_i, T_i, D_i.

How does C_i evolve w.r.t. frequency?

- Proportional to frequency? i.e. C_i/s

Execution time

Classic real-time task model: τ_i, C_i, T_i, D_i.

How does C_i evolve w.r.t. frequency?

- Proportional to frequency? i.e. C_i/s
  - Linux GRUB-PA, DRA, AGR...

Execution time

Classic real-time task model: τ_i, C_i, T_i, D_i.

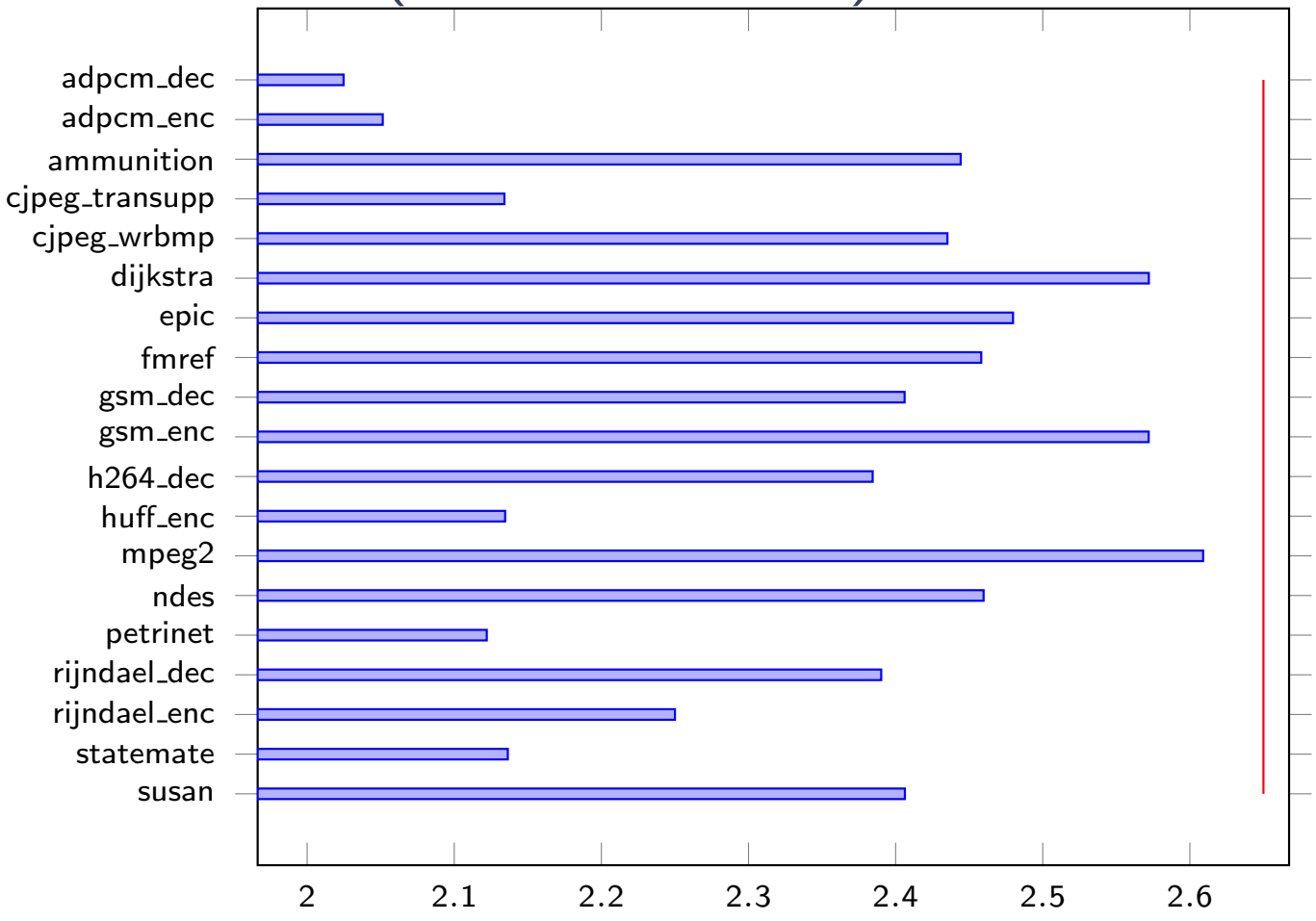
How does C_i evolve w.r.t. frequency?

- Proportional to frequency? i.e. C_i/s
  - Linux GRUB-PA, DRA, AGR...
- Fixed + frequency-dependent parts? i.e. $C_i^{fix} + C_i^{var}/s$
  - FAST, *Aydin et al. 2006*...

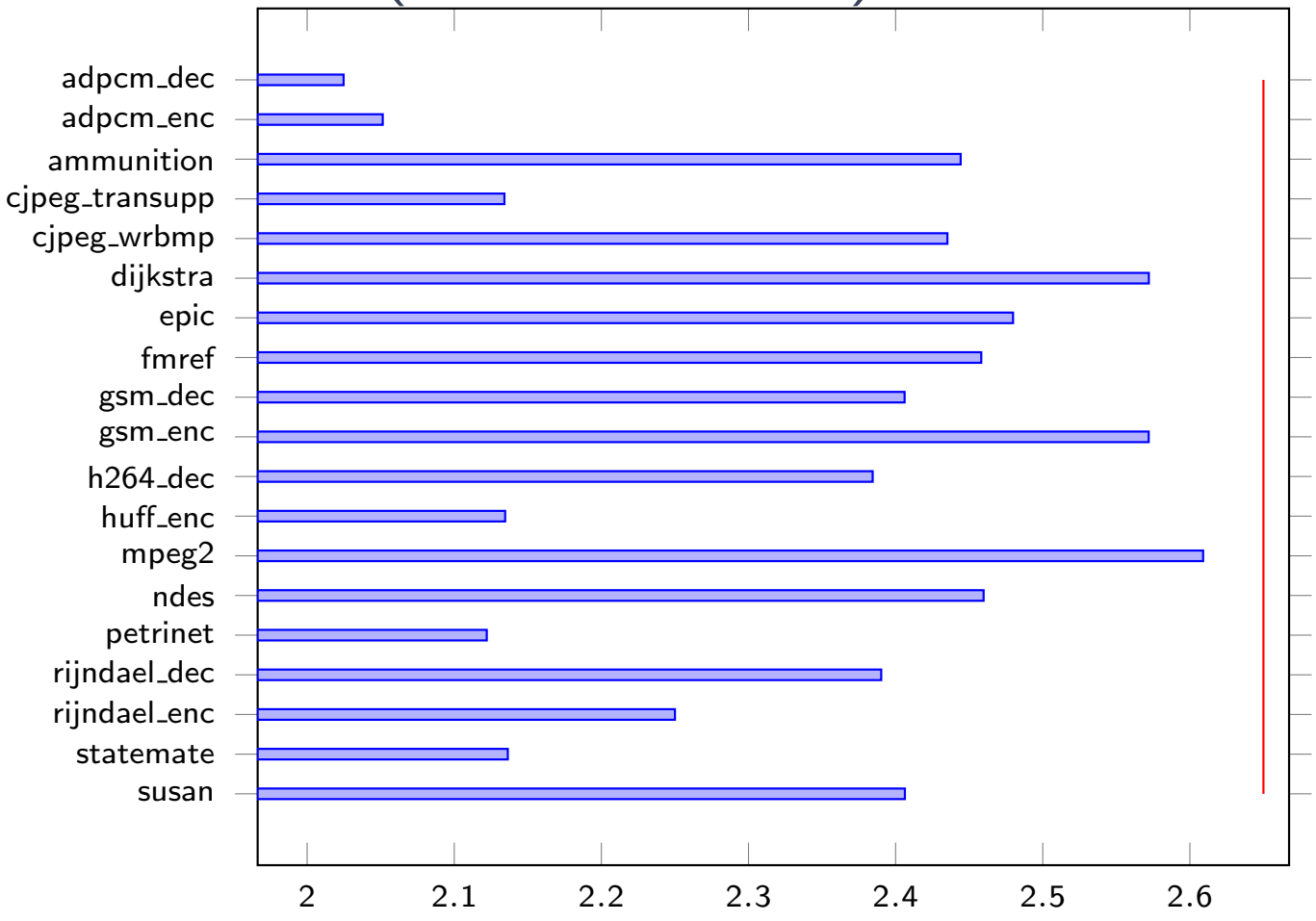
Experimental setup

- Run TACLeBench tests on isolation,
  i.e. one core = one benchmark.
- Perform 500 consecutive runs of the application.
- Processor speed set to *min* or *max*.
- Measure execution time and CPU cycles.
- Platforms:
  1. Intel i7-8650U, Linux 5.1.16, L1-L3 caches, 800MHz - 2100Mhz, DDR4 1200MHz.
  2. ARM Cortex A-53, Linux 4.19.56, L1-L2 caches, 600MHz - 1400MHz, LPDDR2 500MHz.

Execution time (Intel i7-8650U)

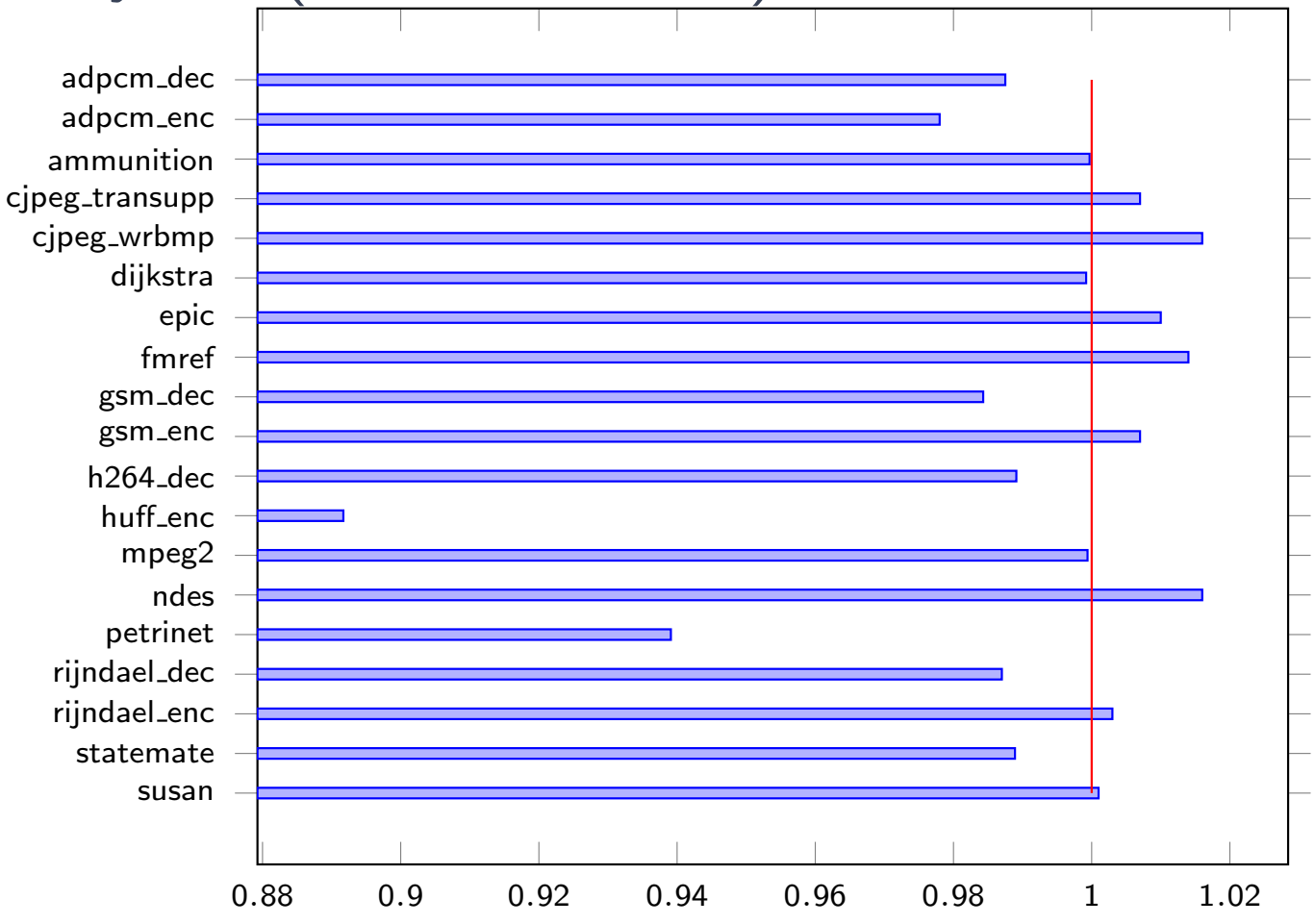


Execution time (Intel i7-8650U)

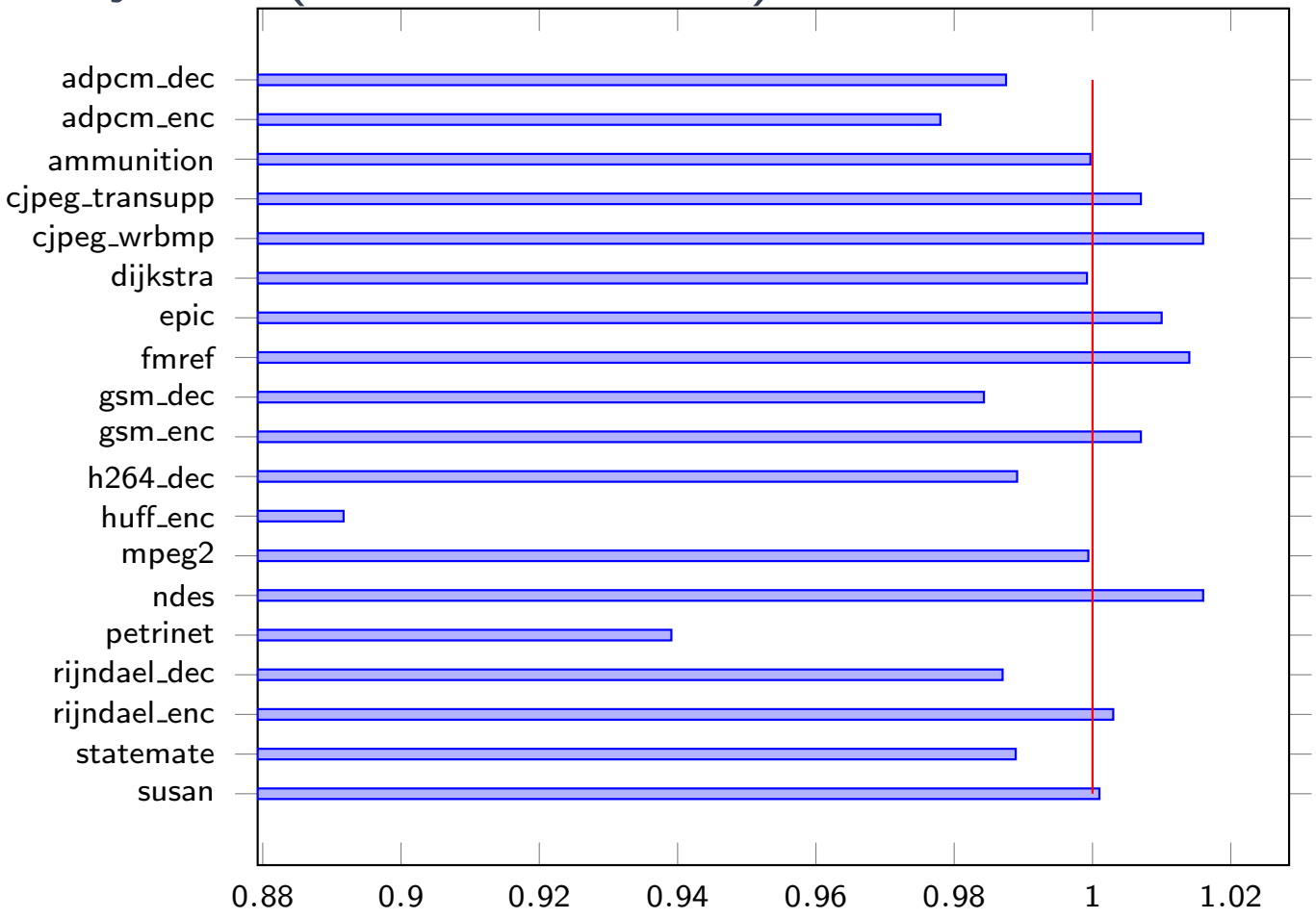


- Theoretical speedup never reached.
- C_i not proportional to CPU frequency.

CPU Cycles (Intel i7-8650U)

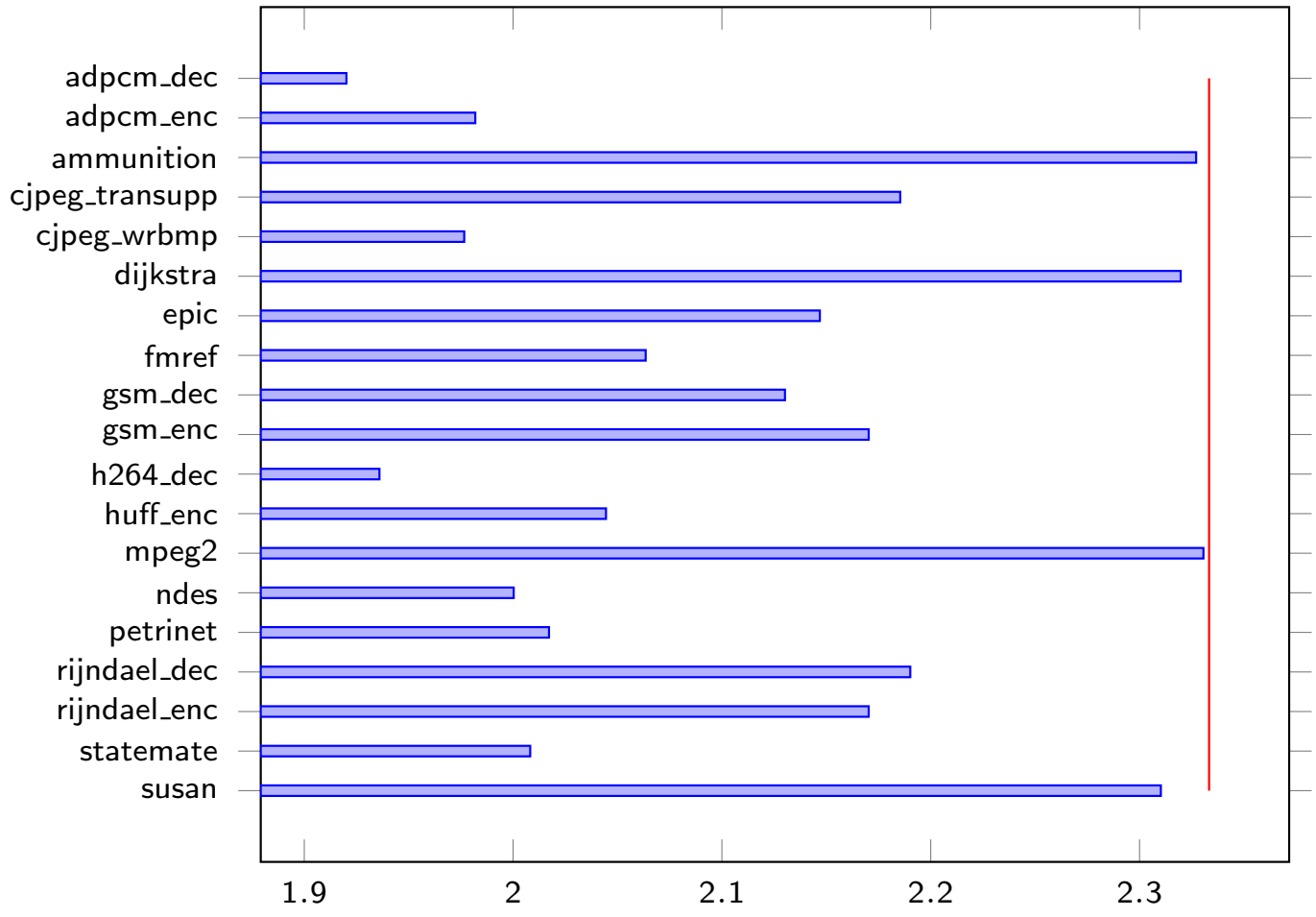


CPU Cycles (Intel i7-8650U)

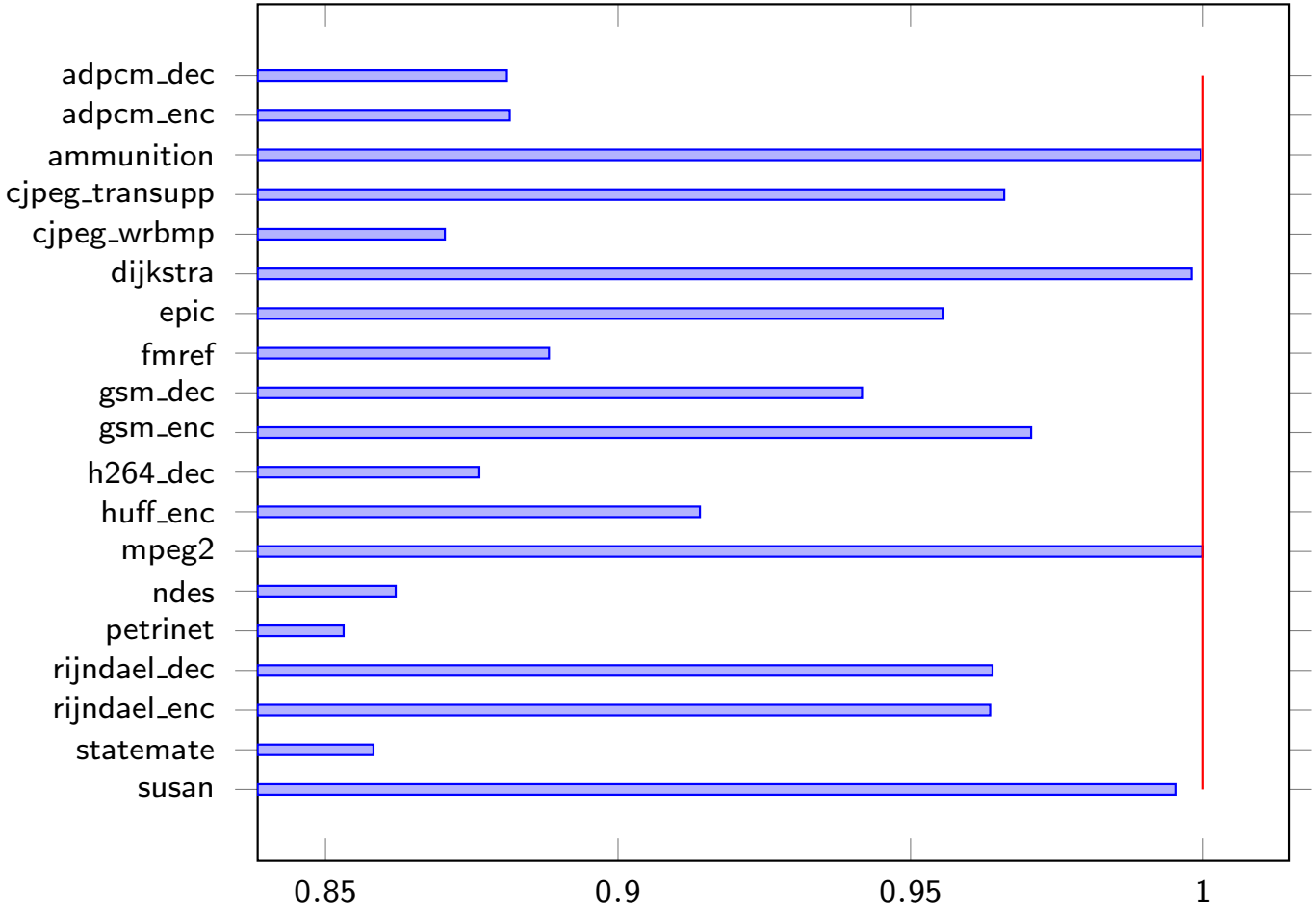


- $> 1 \rightarrow$ less cycles spent in the processor ✓
- Generally programs spend additional cycles waiting for the bus and memory (slower).

Execution time (ARM Cortex A-53)



CPU Cycles (ARM Cortex A-53)



Takeaway messages

- Tasks have different behaviors benefiting more or less on processor speed.
- C_i is not proportional to CPU frequency.
- Bus and memory need to be considered.
  - Model with a fixed part seems ok, but...
  - Requires a deep knowledge about the program and architecture.

Probabilistic RT systems

$$\tau_i = \left(\mathcal{C}_i = \begin{pmatrix} 2 & 3 & 8 \\ 0.5 & 0.3 & 0.2 \end{pmatrix}, T_i, D_i \right)$$

- RTA can guarantee a DMP for the system.
- Systems have a degree of schedulability with a high utilization rate ($U \geq 1$).
- $\nearrow$ functionalities.

Open problems: Frequency scaling and pET (1/2)

How does the pET change w.r.t. CPU frequency?

$$\mathcal{C}_i^f = \begin{pmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \end{pmatrix}$$

- Not proportional to processor speed.
- Obtain information about memory instructions (e.g. FAST) very time consuming.
- Difficult to predict on COTS.

Open problems: Frequency scaling and pET (1/2)

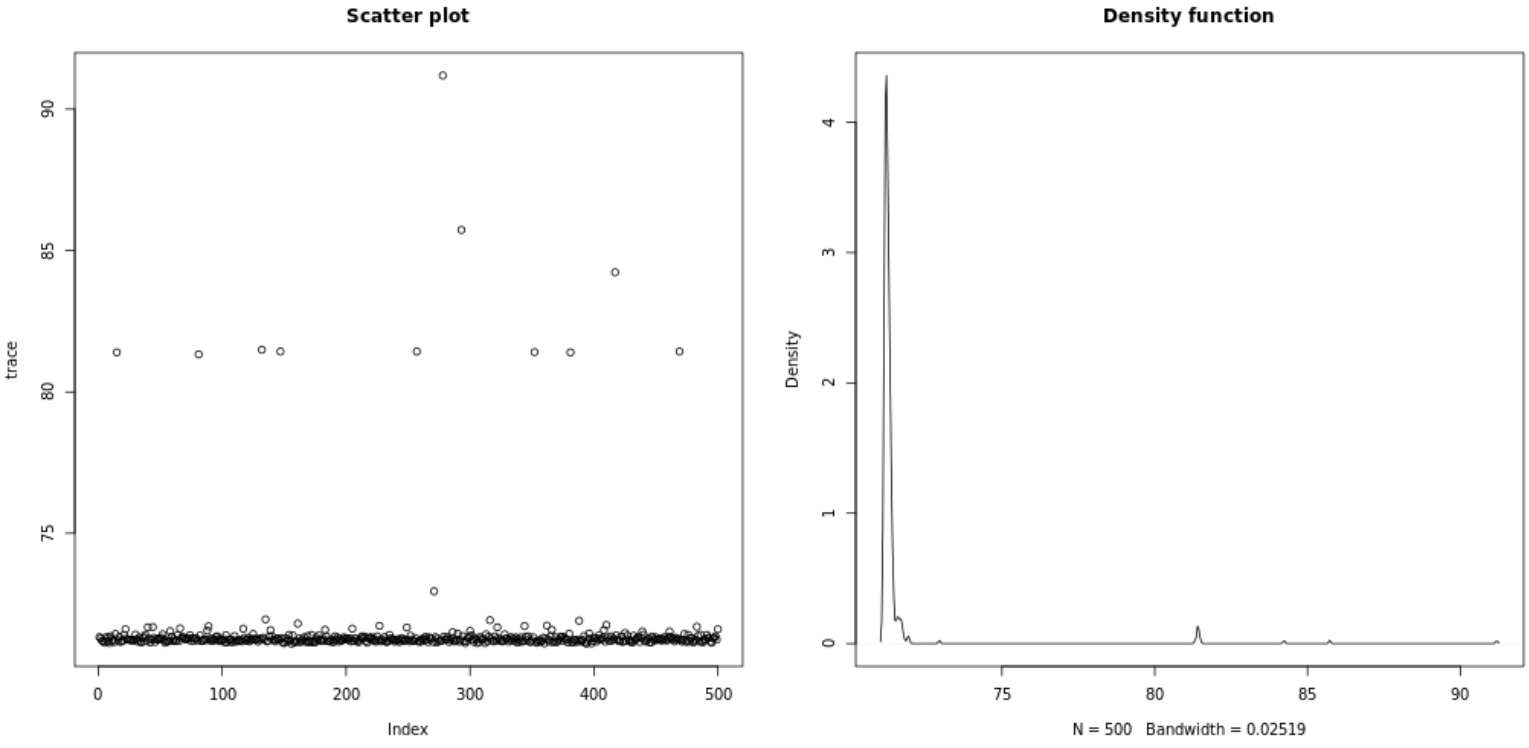


Figure: `dijkstra` @ 1400MHz

Open problems: Frequency scaling and pET (1/2)

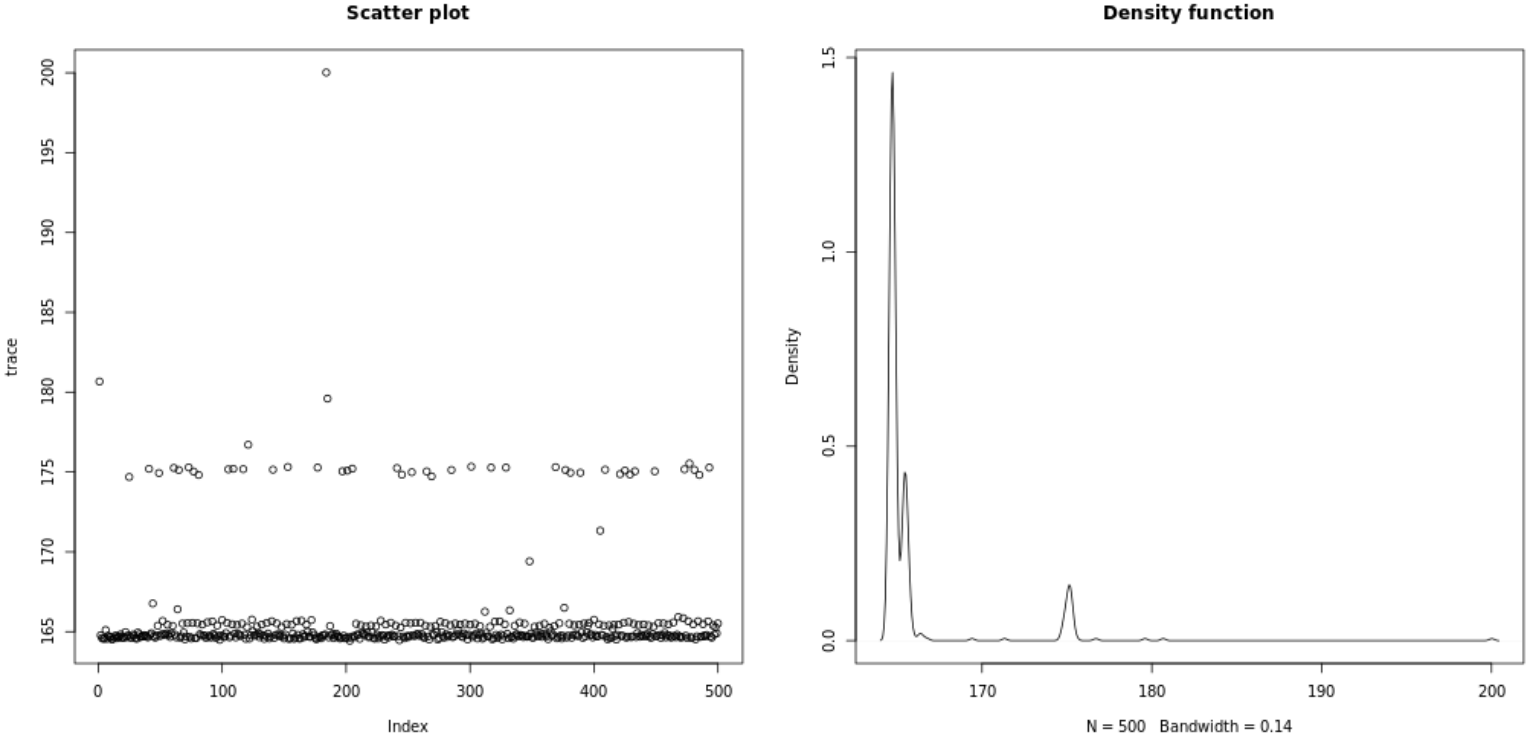


Figure: `dijkstra` @ 600MHz

Open problems: Frequency scaling and pET (2/2)

Can probabilistic RT system benefit from DVFS?

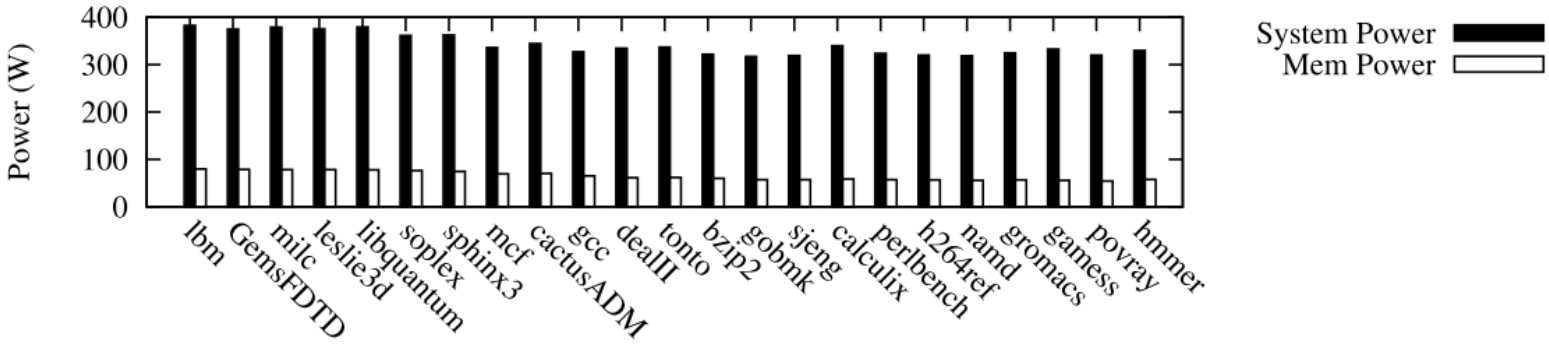
> *We ran experiments where the actual execution time follows a normal or uniform probability distribution function.*

"Power-Aware Scheduling for Periodic Real-Time Tasks"
Aydin et al., IEEE Transactions on Computers, 2004

- More precise distribution to know when to be aggressive.

Open problems: memory frequency scaling (1/2)

- Memory energy consumption is significant (∼20-25%)

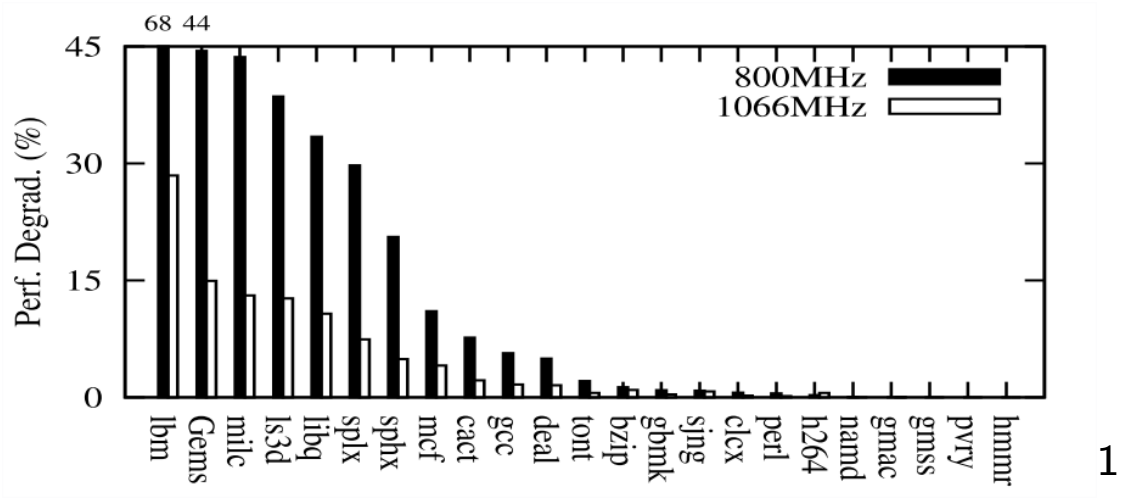


[1]

Can DVFS be exploited at a memory level?

[1] "Memory power management via dynamic voltage/frequency scaling", David et al, 8th ACM international conference on Autonomic computing. 2011

Open problems: memory frequency scaling (2/2)



[1]

How does the pET change w.r.t. memory frequency?

- Tasks less affected by memory accesses $\rightarrow$ scale down memory frequency.

[1] "Memory power management via dynamic voltage/frequency scaling", David et al, 8th ACM international conference on Autonomic computing. 2011

Thank you

Any question?