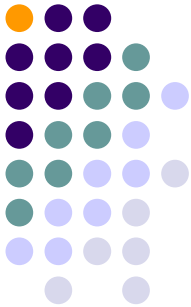


Energy-Aware Dense and Sparse Linear Algebra

Enrique S. Quintana-Ortí

HPC&A
High Performance Computing and Architectures
UNIVERSITAT JAUME·I

Energy-aware dense and sparse linear algebra

- Universidad Politécnica de Valencia
  - Pedro Alonso
- Universidad Jaime I
  - J. I. Aliaga
  - Manuel F. Dolz
  - Rafael Mayo
  - Enrique S. Quintana-Ortí
- CIMNE
  - Alberto F. Martín

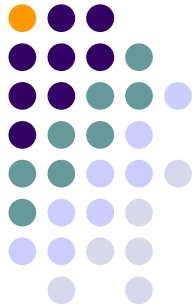
Concurrency and energy efficiency

2010 PFLOPS (10^{15} flops/sec.)

2010 JUGENE

- 10^9 core level
  (PowerPC 450, 850MHz → 3.4 GFLOPS)
- 10^1 node level
  (Quad-Core)
- 10^5 cluster level
  (73.728 nodes)

Concurrency and energy efficiency



2010 PFLOPS (10^{15} flops/sec.)

2010 JUGENE

- 10^9 core level
  (PowerPC 450, 850MHz → 3.4 GFLOPS)
- 10^1 node level
  (Quad-Core)
- 10^5 cluster level
  (73.728 nodes)

2020 EFLOPS (10^{18} flops/sec.)

- $10^{9.5}$ core level
- 10^3 node level!
- $10^{5.5}$ cluster level

Concurrency and energy efficiency

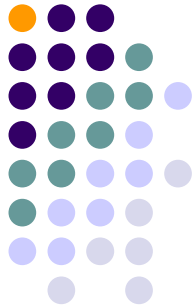
- Green500 (November 2011*)

Rank Green/Top	Site	#Cores	MFLOPS/W	LINPACK (TFLOPS)	MW to EXAFLOPS?
1/29	IBM Rochester – BlueGene/Q, Power BQC 16C 1.60 GHz	32.768	2.026.48	339,83	493,47
32/1	RIKEN AICS K Computer– Spar64 VIIIfx (8-core)	705.024	830,18	10.510,00	1.204,60

Most powerful reactor under construction in France
Flamanville (EDF, 2017 for US $9 billion):
1,630 MWe

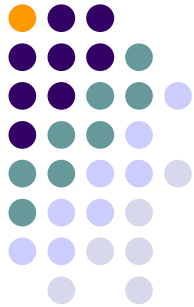
*Green500 June 2012 to be released today

Concurrency and energy efficiency



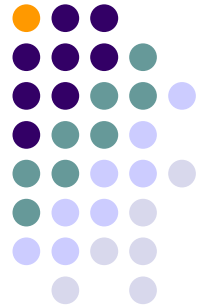
- Reduce energy consumption!
  - Costs over lifetime of an HPC facility often exceed acquisition costs
  - Carbon dioxide is a hazard for health and environment
  - Heat reduces hw reliability
- Personal view
  - Hardware features energy saving mechanism
  - Scientific apps are in general energy oblivious

Outline



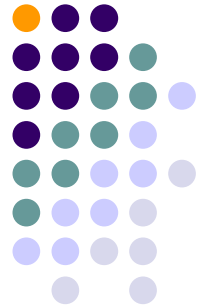
- ILUPACK
- Experimental setup
- Power model
  - Leveraging P-states
  - Leveraging C-states
- Dense linear algebra
- Conclusions

ILUPACK



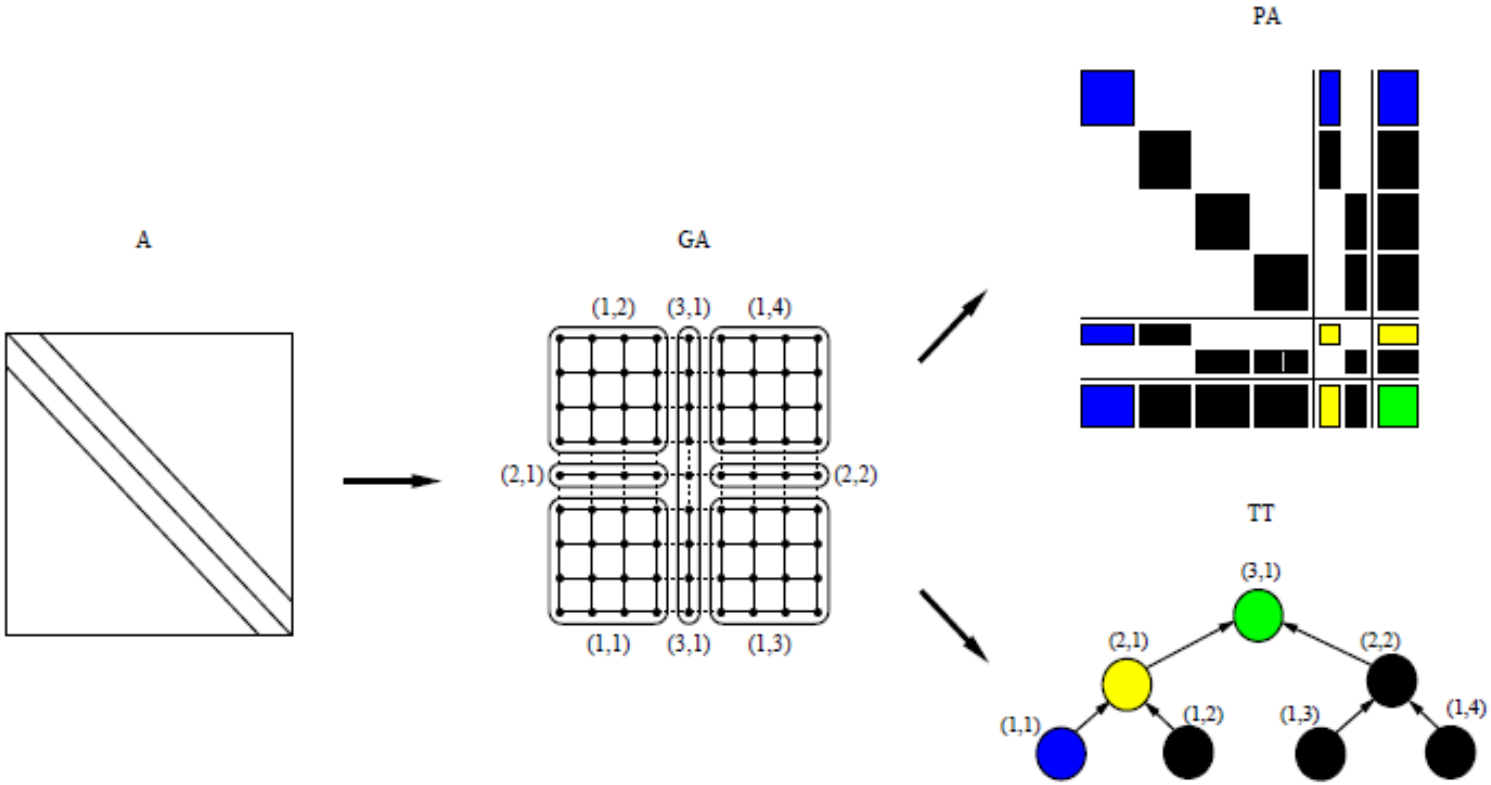
- Incomplete LU Package (http://ilupack.tu-bs.de)
  - Iterative Krylov subspace methods
  - Multilevel ILU preconditioners for general/symmetric/Hermitian positive definite systems
  - Based on inverse ILUs with control over growth of inverse triangular factors
  - Specially competitive for linear systems from 3D PDEs

ILUPACK
Multi-threaded version (task parallelism)



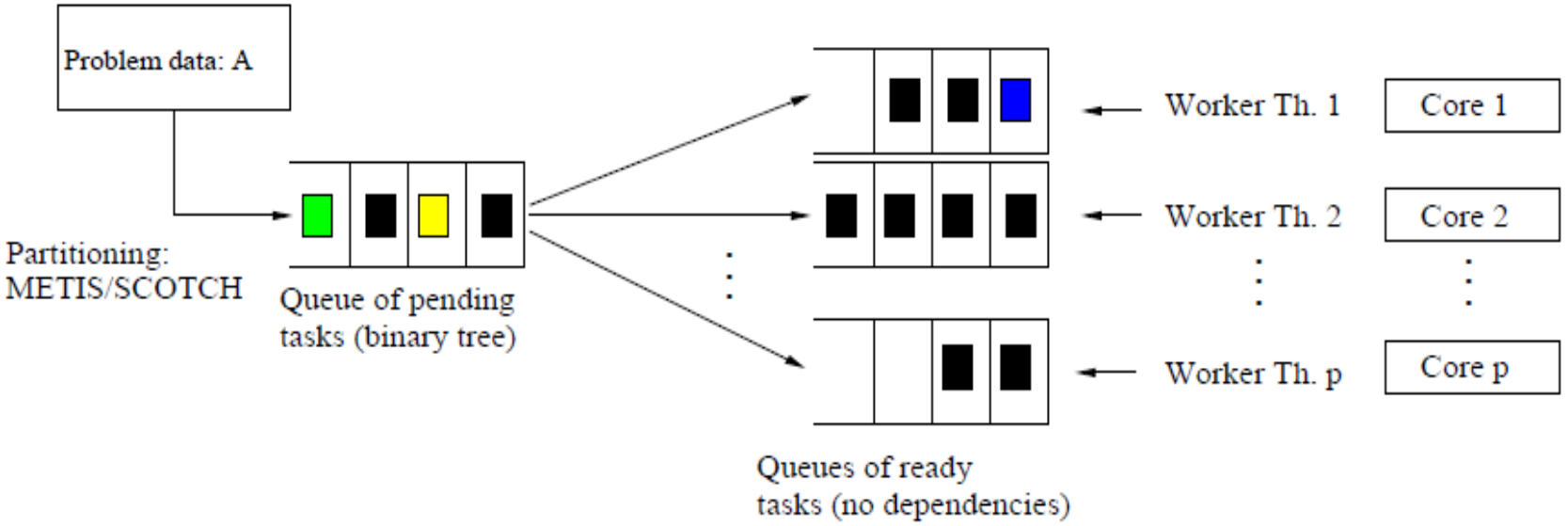
- Real s.p.d. systems
- Construction of preconditioner and PCG solver
- Algebraic parallelization based on a task tree
- Leverage task parallelism of the tree
- Dynamic scheduling via runtime (OpenMP)
  - "Exploiting thread-Level parallelism in the iterative solution of sparse linear systems". J. I. Aliaga, M. Bollhöfer, A. F. Martín, E. S. Quintana-Ortí. Parallel Computing, 2011

ILUPACK
Multi-threaded version (task parallelism)

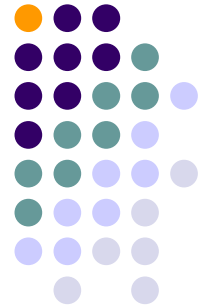


ILUPACK
Multi-threaded version (task parallelism)

- Run-time in charge of scheduling



Experimental setup



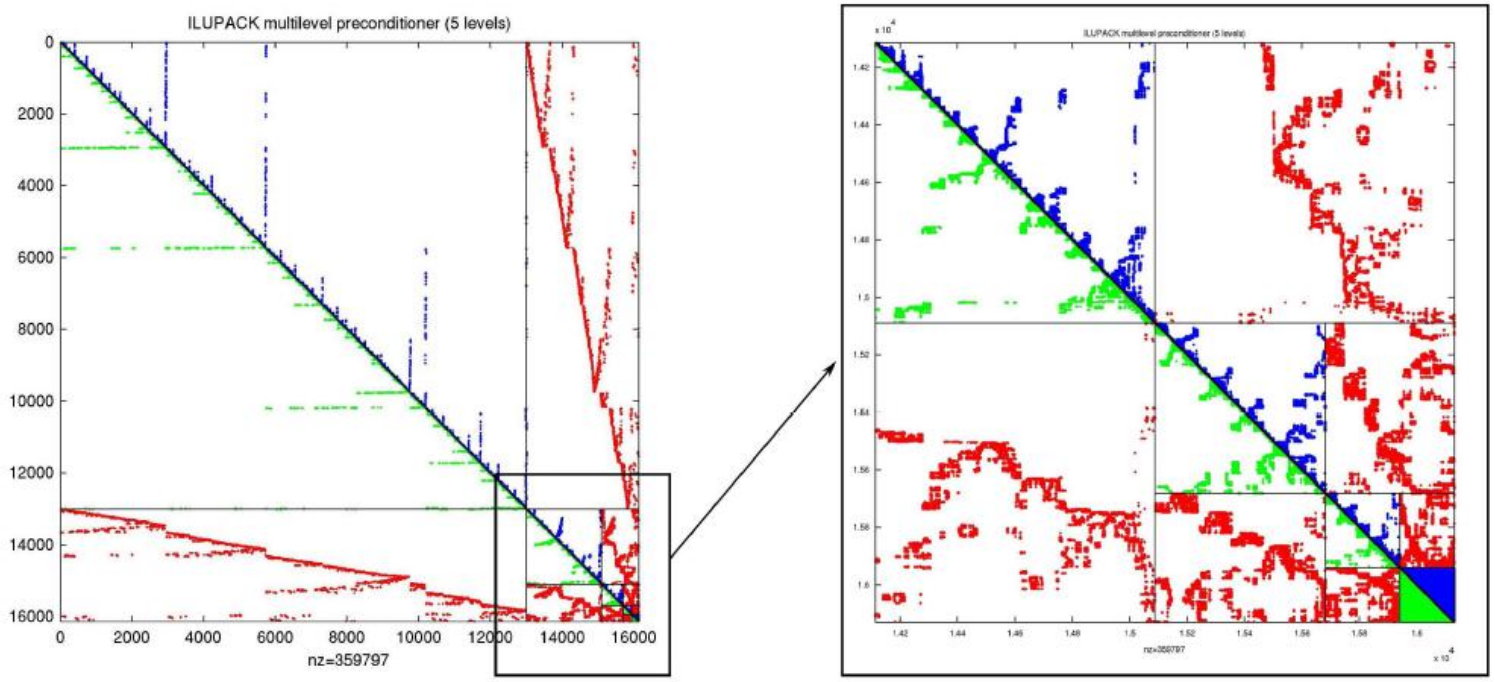
- 2 AMD Opteron 6128 processors
- 48 GB of RAM
- DVFS enabled per core (P-states)

P-state P_i	VCC_i	f_i
P_0	1.23	2.00
P_1	1.17	1.50
P_2	1.12	1.20
P_3	1.09	1.00
P_4	1.06	0.80

- C-states:
  - C0: normal operation mode
  - C1, C1E: disable core components (L1/L2 caches), clock signal, mem. controller,… increases energy savings at the expense of recovery time

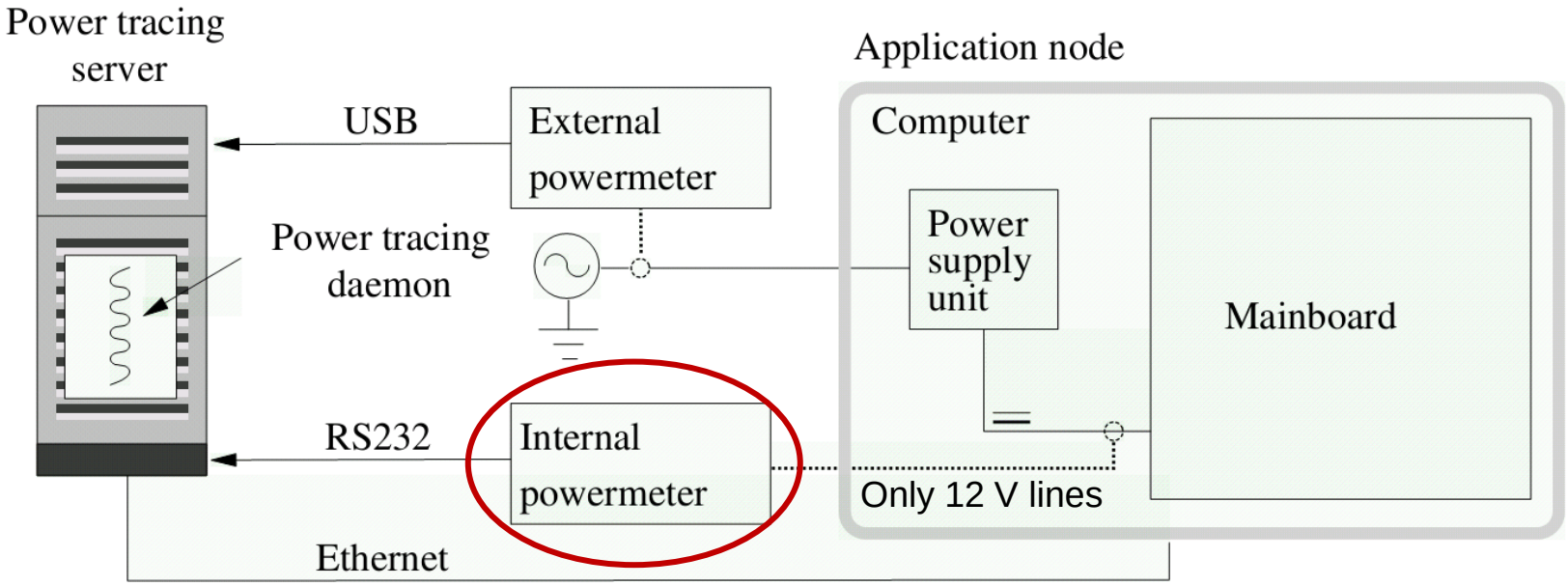
Experimental setup

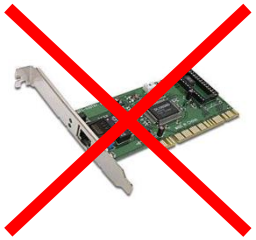
- Sparse linear system benchmark
  - Laplacian equation $-\Delta u = f$ in a 3D unit cube $\Omega = [0,1]^3$
  - Linear system $Au = b$ with $A \rightarrow n \times n$, $n = 252^3 \approx$ 16 million unknowns and 111 millions of nonzero entries



Cost of energy
Setup

- DC powermeter with sampling freq. $= 25$ Hz
  - LEM HXS 20-NP transductors with PIC microcontroller
  - RS232 serial port





Power model

$$P = P^{(S)Y(stem)} + P^{C(PU)} = P^{Y} + P^{S(tatic)} + P^{D(ynamic)}$$

- P^C is the power dissipated by CPU (socket): $P^S + P^D$
- P^S is the static power
- P^D is dynamic power
- P^Y is the power of remaining components (e.g., RAM)

Considerations:

- P^Y and P^S are constants (though P^S grows with temperature)
- Hot system

Power model

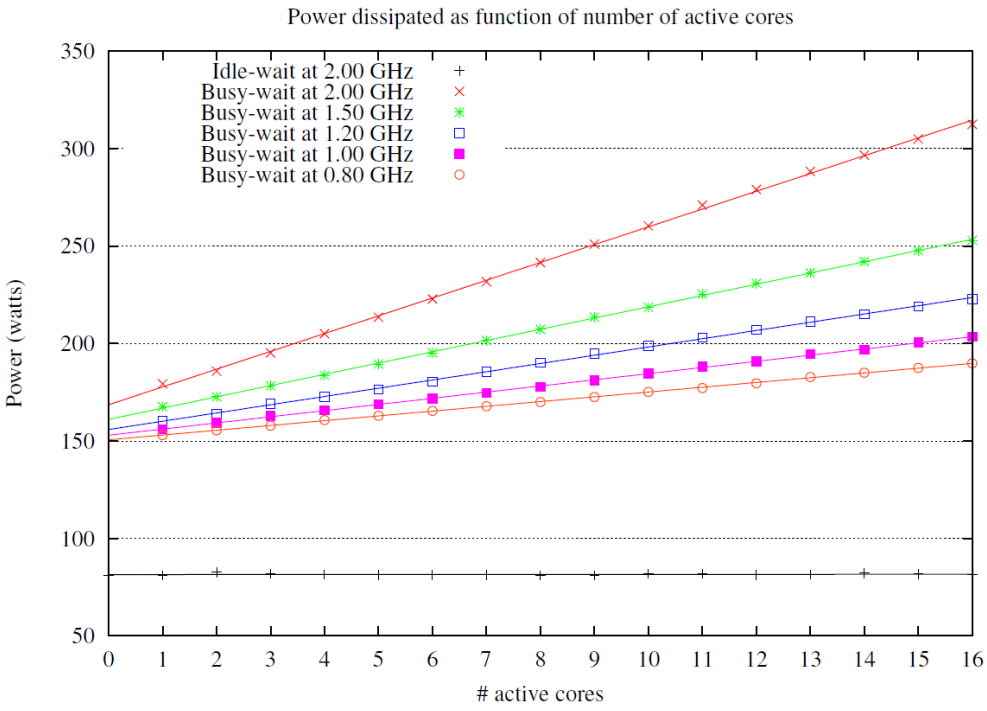
- System power:

$$P = P^Y + P^S + P^D$$

Estimated as *idle* power

Due to off-chip components:

e.g., RAM (only mainboard)

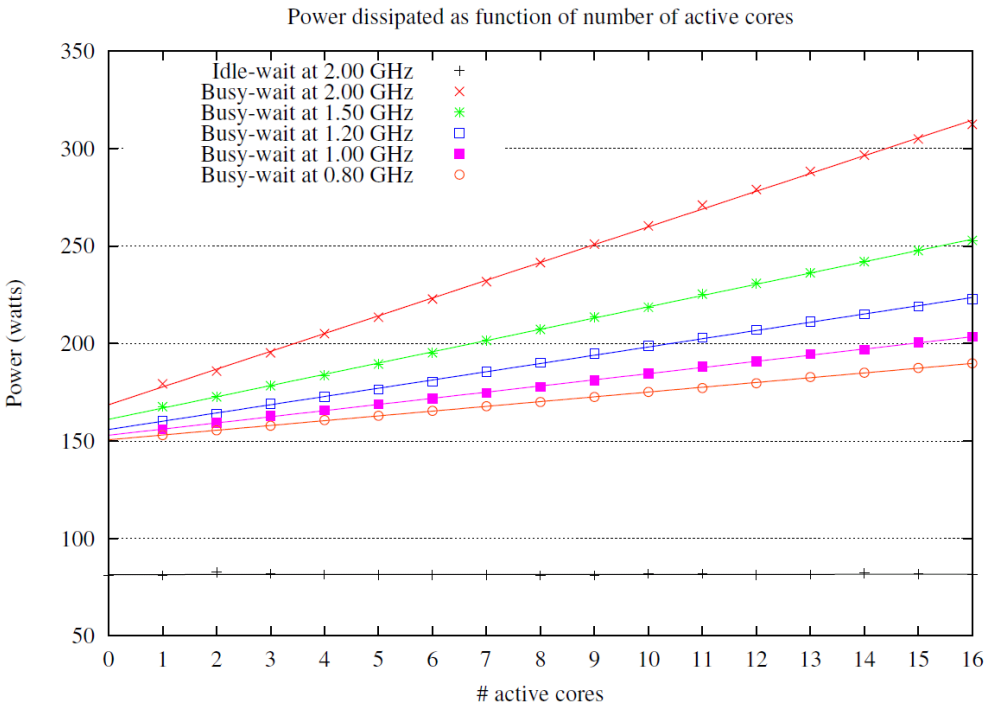


$$P^Y \approx P^I = 80.15\ \mathrm{W}$$

Power model

- Static power:

$$P = P^Y + P^S + P^D$$



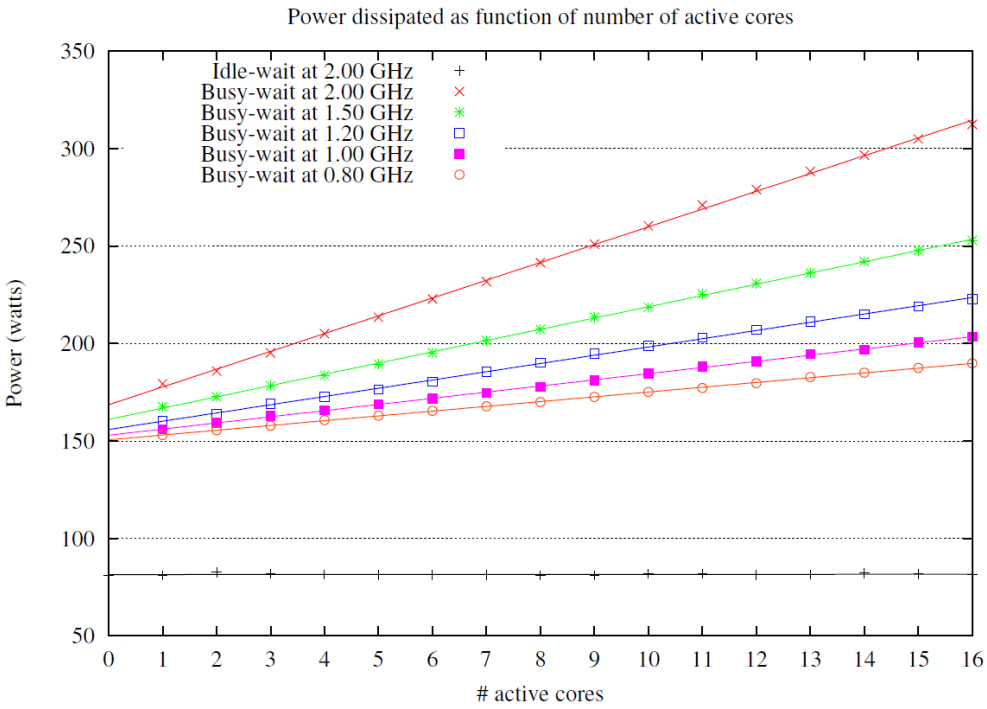
$$P^T{}_0(c) = a_0 + b_0 \cdot c = 168.59 + 9.12 \cdot c \text{ W}$$

$$P^S{}_0 \approx a_0 - P^Y = 88.44 \text{ W}$$

Power model

- Dynamic power:

$$P = P^Y + P^S + P^D$$



$$P^T{}_0(c) = a_0 + b_0 c = 168.59 + 9.12 \cdot c \text{ W}$$

Busy-wait: $P^D{}_0 \approx b_0 c = 9.12 \cdot c$ W

Power model

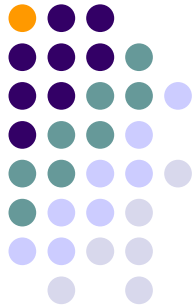
P-state P_i	Vcc_i	f_i	α_i	β_i	ΔP_i^S	ΔP_i^D
P_0	1.23	2.00	168.59	9.12	–	–
P_1	1.17	1.50	161.10	5.77	-9.52	-32.14
P_2	1.12	1.20	155.90	4.23	-17.09	-50.25
P_3	1.09	1.00	152.94	3.15	-21.47	-60.73
P_4	1.06	0.80	150.61	2.44	-25.73	-70.30

- Remember, $P^Y \approx P^I$ is constant
- Thus, e.g., moving all cores from P_0 to P_1

$$P^T{}_1(16) = P^S{}_0\,(1\text{-}0.0952) + P^D{}_0\,(16)(1\text{-}0.43214) + P^Y$$
$$= 259.19\,\text{W}$$

- These values agree withing 2.5% with the linear regression models

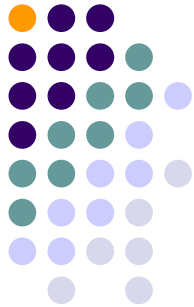
Power model
Leveraging P-states



P-state P_i	Vcc_i	f_i	α_i	β_i	ΔP_i^S	ΔP_i^D
P_0	1.23	2.00	168.59	9.12	–	–
P_1	1.17	1.50	161.10	5.77	-9.52	-32.14
P_2	1.12	1.20	155.90	4.23	-17.09	-50.25
P_3	1.09	1.00	152.94	3.15	-21.47	-60.73
P_4	1.06	0.80	150.61	2.44	-25.73	-70.30

- DVFS = P-states (see ACPI standard)
- Moving to a more power-friendly state results in ↓power
- ↓power = ↓energy?
- For a compute-bounded operation, f_i is linear to performance and time^{-1}
- In principle, for a memory-bounded operation (ILUPACK), decreasing f_i should not affect time!

Power model
Leveraging P-states



- 1st attempt: ~~Dynamic~~ Static voltage-frequency scaling

P-state P_i	T_i	$\bar{P}_i^T$	E_i	ΔT_i	$\Delta \bar{P}_i^T$	ΔE_i
P_0	34.06	282.87	9,634.78	–	–	–
P_1	43.57	235.64	10,267.72	21.88	-16.69	6.53
P_2	54.48	210.86	11.478.79	59.91	-25.45	19.20
P_3	61.58	197.01	12.132.79	80.73	-30.35	25.87
P_4	76.50	186.86	14,295.18	124.47	-33.94	48.28

Why?

Power model
Leveraging P-states

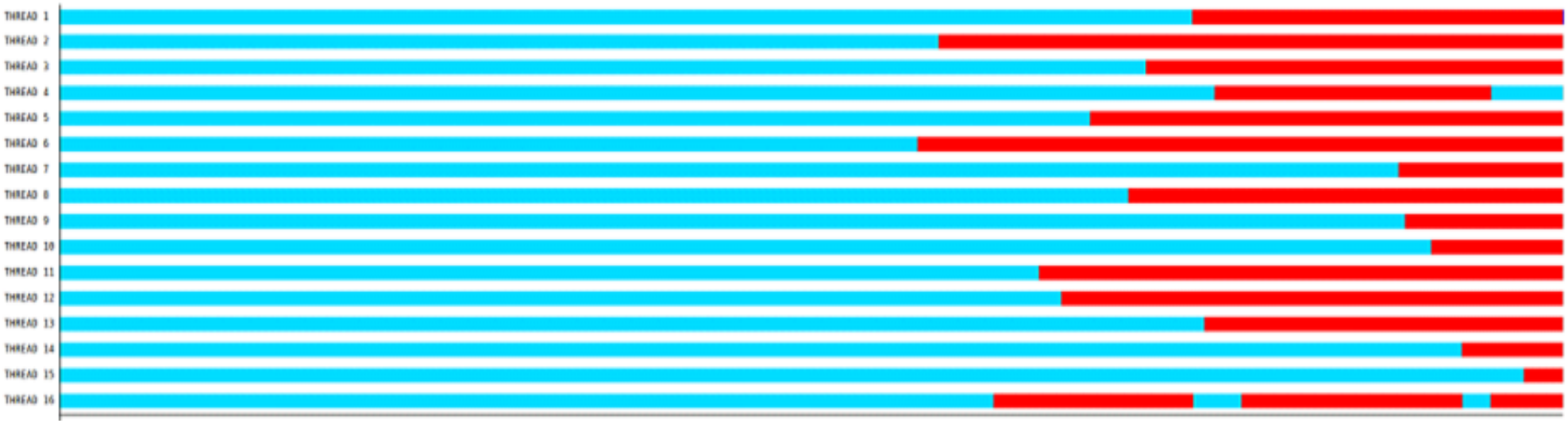
- 1st attempt: ~~Dynamic~~ Static voltage-frequency scaling

P-state P_i	Vcc_i	f_i	T_i	ΔT_i	BW_i	ΔBW_i
P_0	1.23	2.00	34.06	–	30.29	–
P_1	1.17	1.50	43.57	21.88	24.63	-18.67
P_2	1.12	1.20	54.48	59.91	20.46	-32.44
P_3	1.09	1.00	61.58	80.73	17.48	-42.30
P_4	1.06	0.80	76.50	124.47	14.00	-53.77

Combined effect of linear decrease of CPU performance and memory bandwidth!

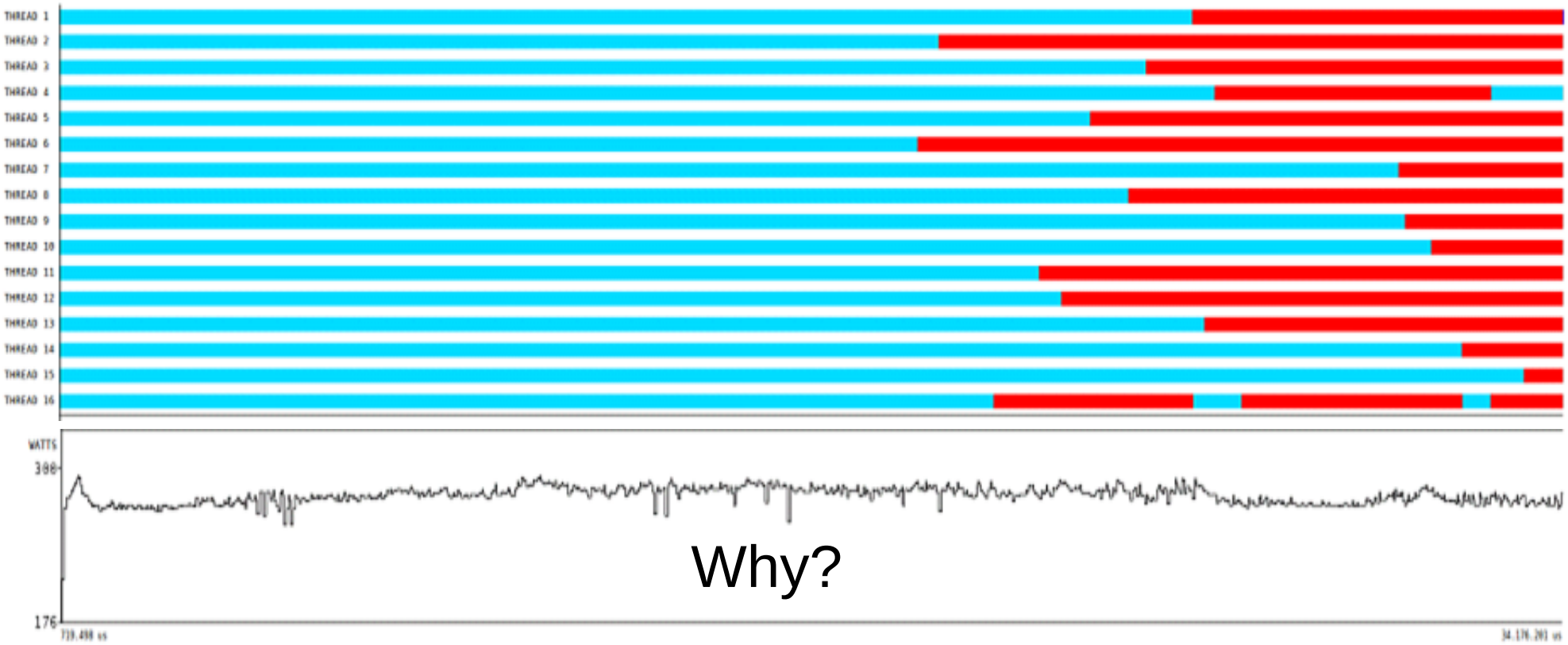
Power model
Leveraging P-states

- 2nd attempt: DVFS during idle periods



Power model
Leveraging P-states

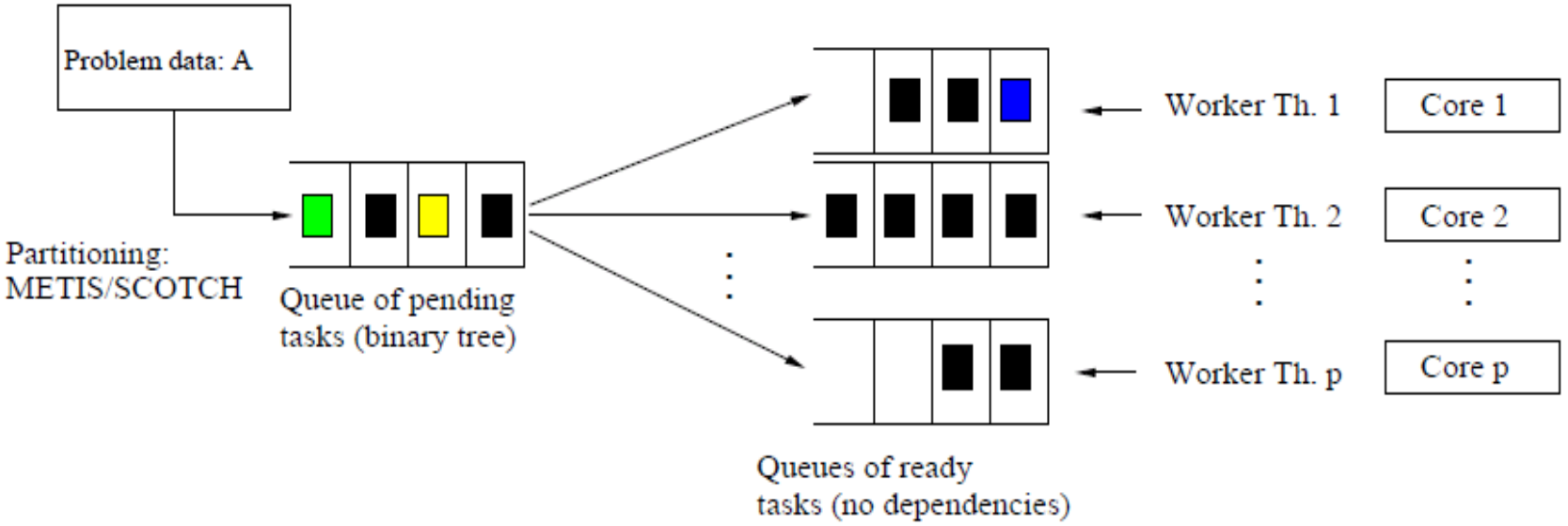
- 2nd attempt: DVFS during idle periods



Why?

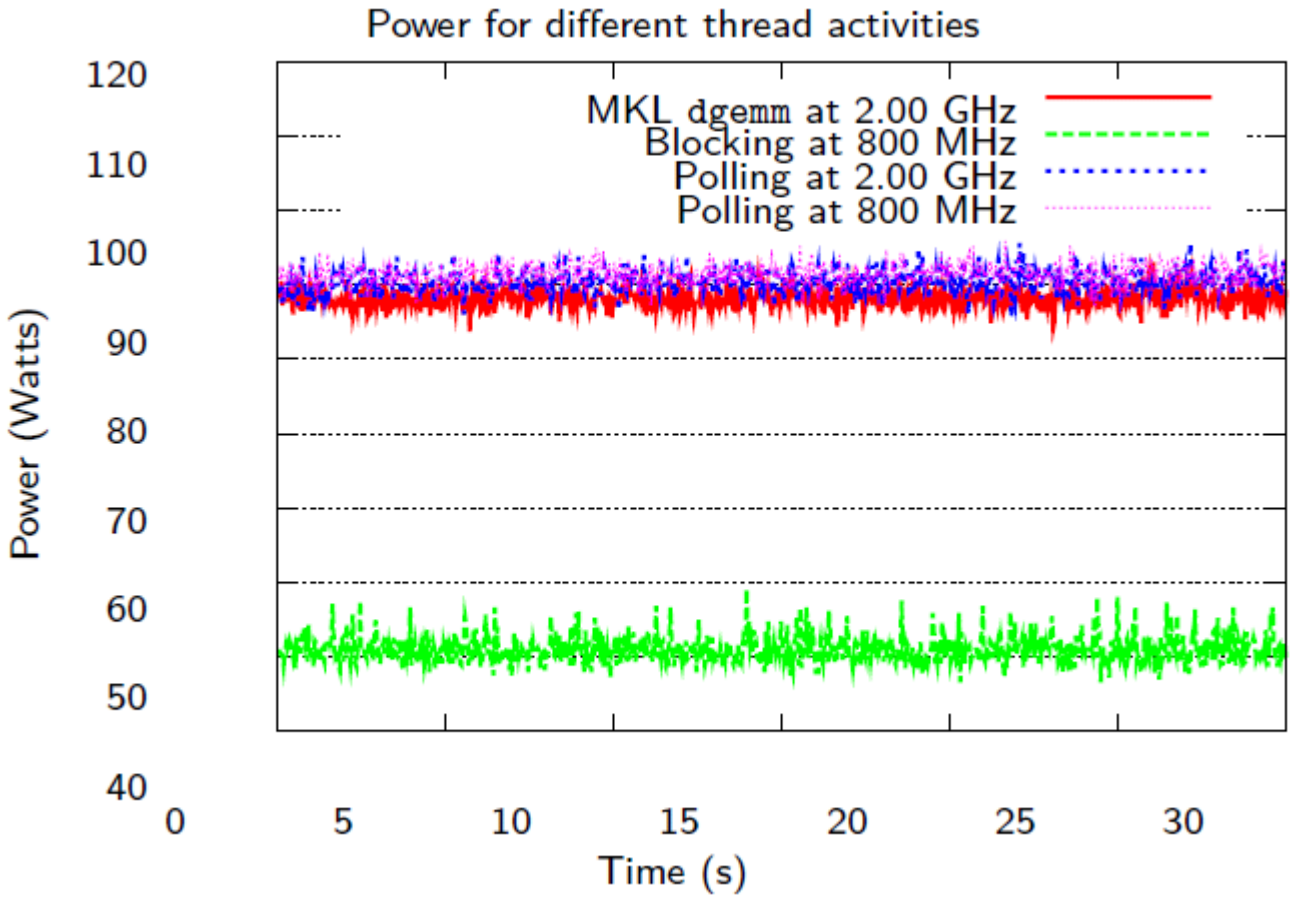
Power model
Leveraging P-states

- 2nd attempt: DVFS during idle periods



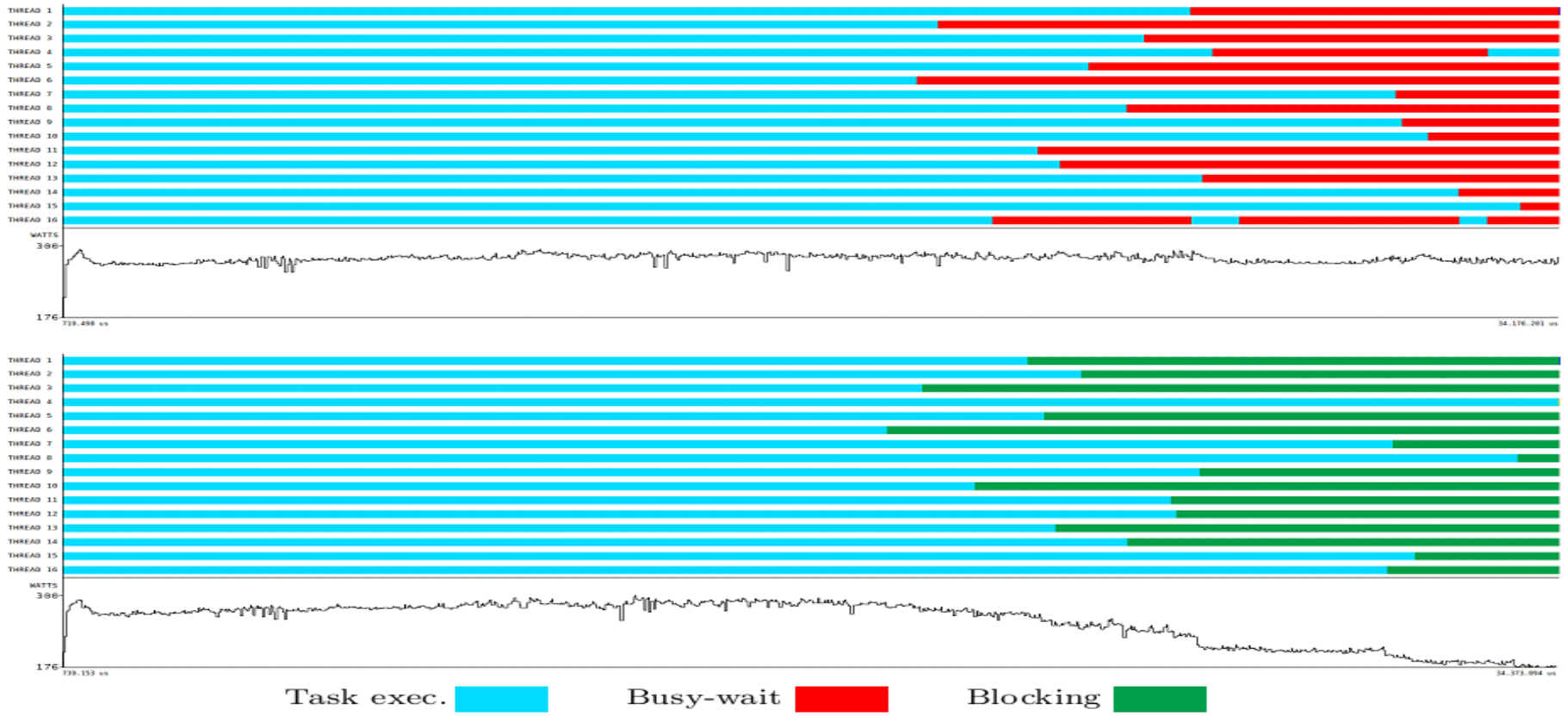
Power model
Leveraging P-states

- Active polling for work…

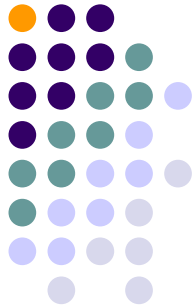


Power model
Leveraging P- and C-states

- 3rd attempt: DVFS and idle-wait



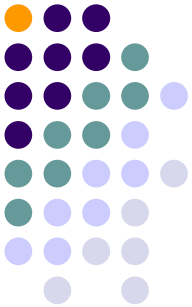
Power model
Leveraging P- and C-states



- 3rd attempt: DVFS and idle-wait:
  - Savings of 6.92% of total energy
  - Negligible impact on execution time

- …but take into account that
  - Idle time: 23.70%
  - Dynamic power: 32.32%
  - Upper bound of savings: $39.32 \cdot 0.2370 = 9.32\%$

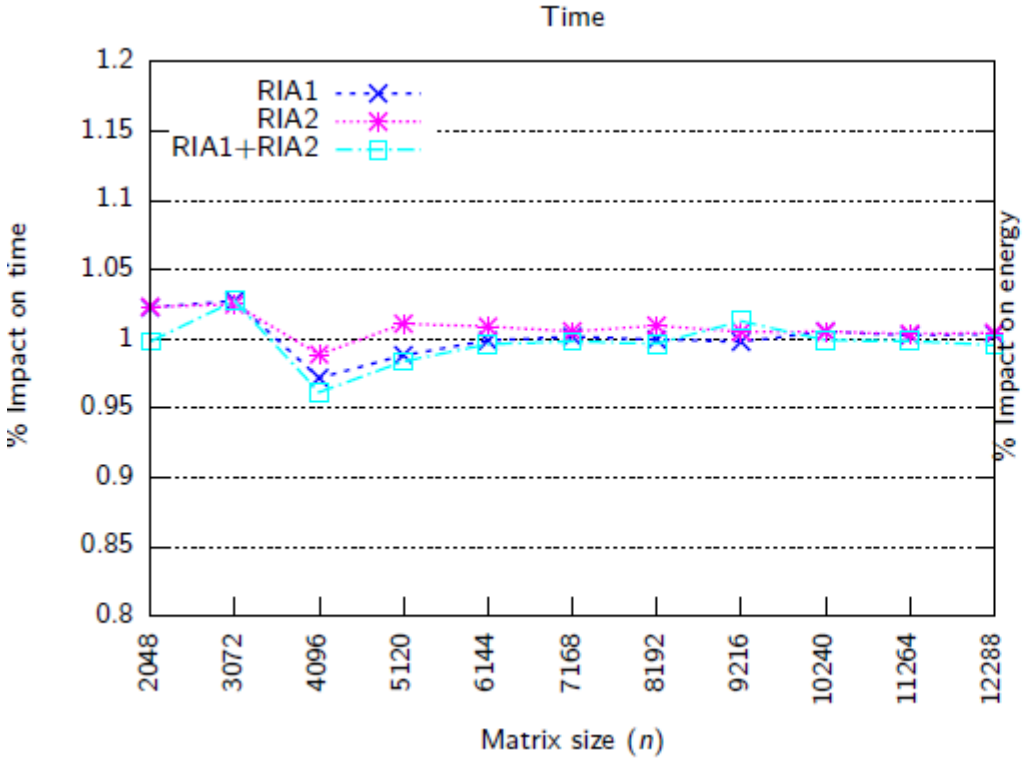
Power model
Leveraging P- and C-states

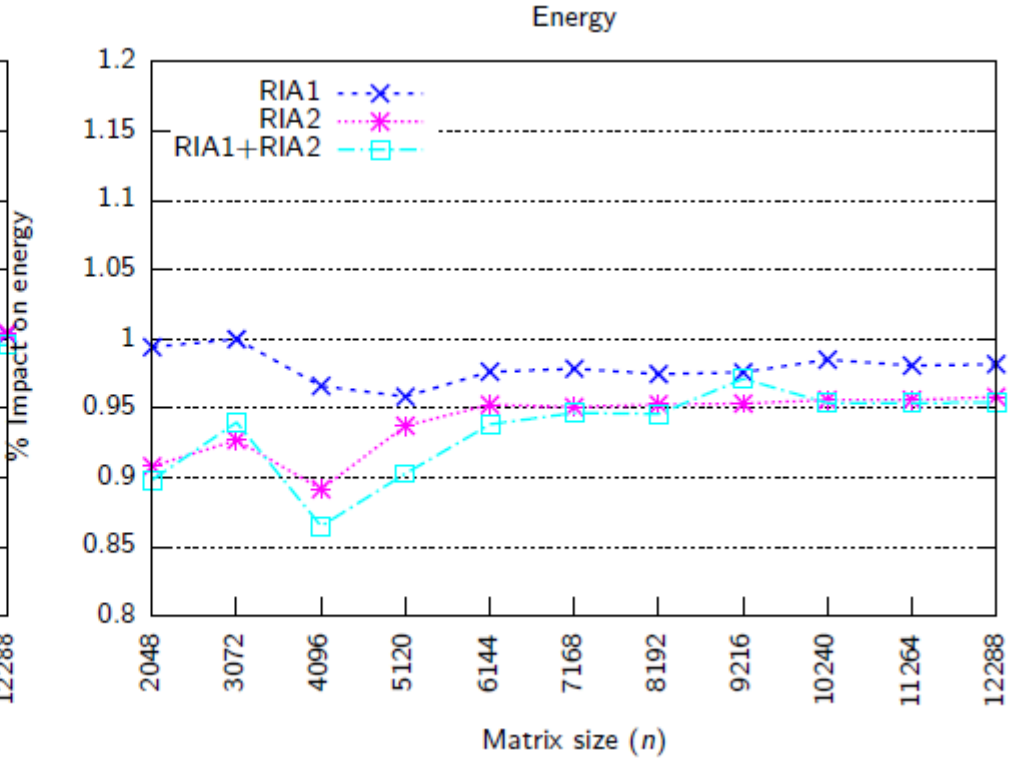


- Other opportunities to save energy
  - Dense linear algebra
  - Hybrid CPU-CPU computing

Power model
Leveraging P- and C-states multicore

- LUPP factorization

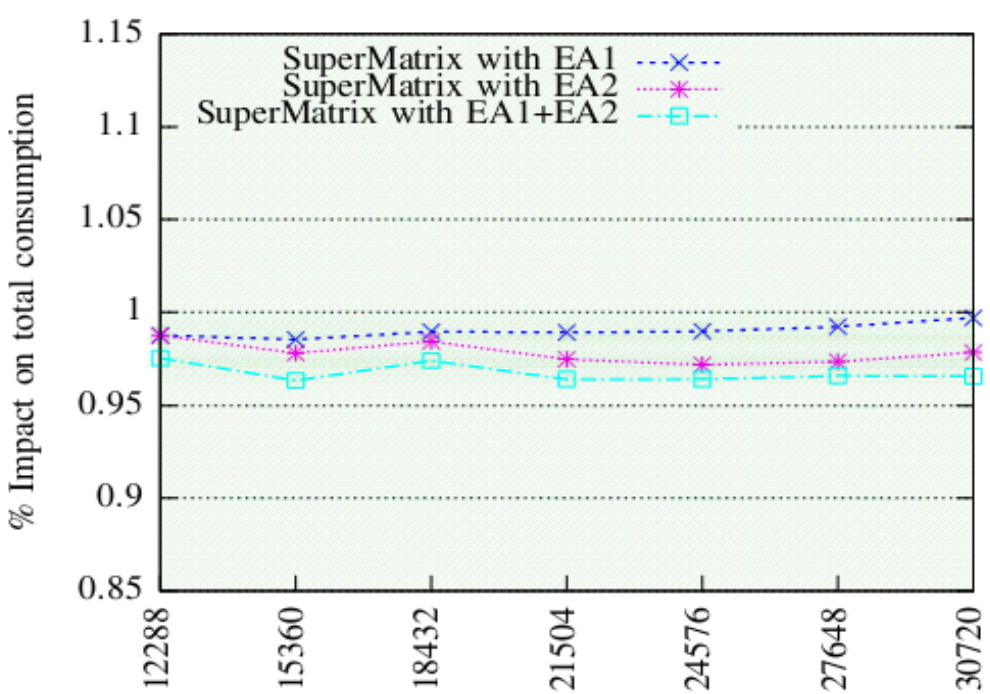


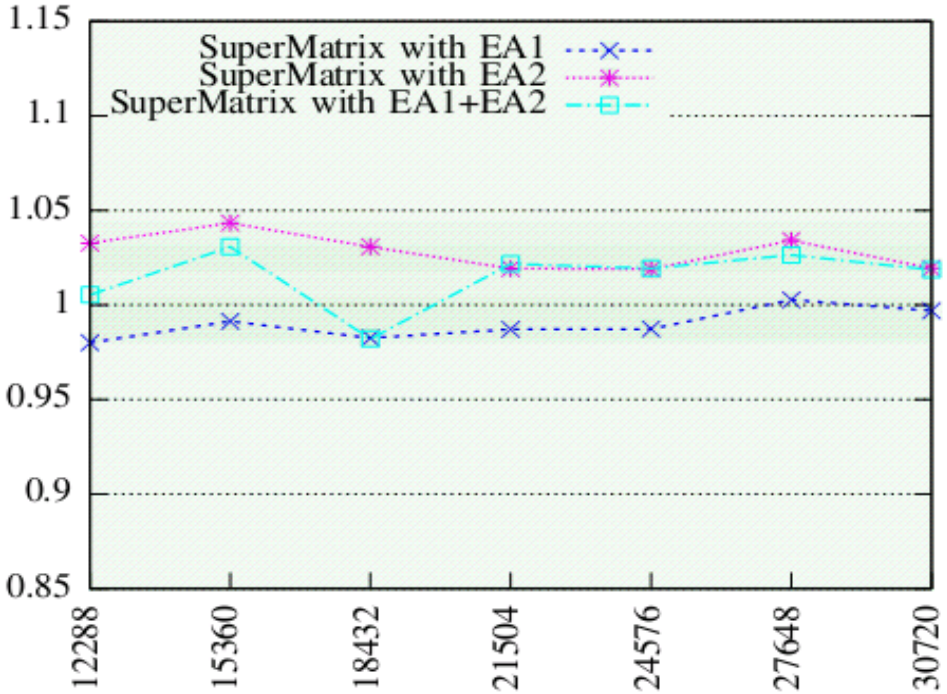


RIA1: leverage P-states
RIA2: idle-wait

Power model
Leveraging P- and C-states on CPU-GPU

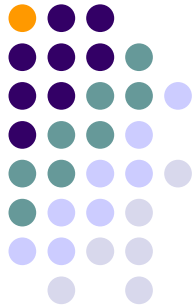
- Cholesky factorization





EA1: no polling when there is no work
EA2: no polling when work is in GPU

More information

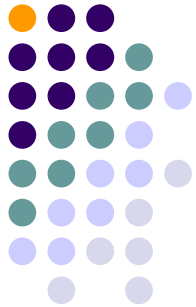


- “Leveraging task-parallelism in energy-efficient ILU preconditioners”, J. I. Aliaga, M. F. Dolz, A. F. Martín, E. S. Quintana-Ortí. ICT-GLOW 2012. Vienna (Austria)

- “Modeling power and energy of the task-parallel Cholesky factorization on multicore processors”, P. Alonso, M. F. Dolz, R. Mayo, E. S. Quintana-Ortí. EnaHPC 2012. Hamburg (Germany)

- “Energy-efficient execution of dense linear algebra algorithms on multicore processors”. P. Alonso, M. F. Dolz, R. Mayo, E. S. Quintana-Ortí. Cluster Computing, 2012

Performance and energy consumption Summary



- A battle to be won in the core arena
  - More concurrency
  - Heterogeneous designs
- A related battle to be won in the power arena
  - "*Do nothing, efficiently…*" (V. Pallipadi, A. Belay) or "*Doing nothing well*" (D. E. Culler)
  - Don't forget the cost of system+static power