Doing Nothing to Save Energy in Matrix Computations

Enrique S. Quintana-Ortí

quintana@icc.uji.es

eeClust Workshop

September 11, 2012, Hamburg, Germany

Energy efficiency
Motivation

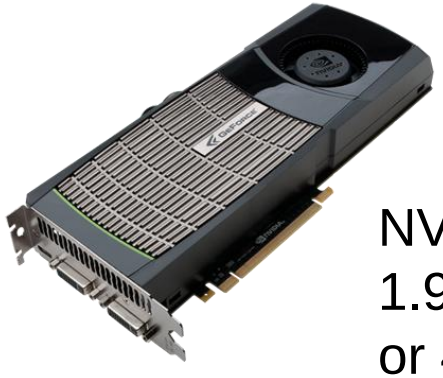
Doing nothing to save energy?

Why at Ena-HPC then?

Energy efficiency Motivation

- Green500/Top500 (June 2012)

Rank Green/Top	Site, Computer	#Cores	MFLOPS/W	LINPACK (TFLOPS)	MW to EXAFLOPS?
1/252	DOE/NNSA/LLNL BlueGene/Q, Power BQC 16C 1.60GHz	8,192	2,100.88	86.35	475.99
20/1	DOE/NNSA/LLNL BlueGene/Q, Power BQC 16C 1.60GHz	1,572,864	2,069.04	16,324.75	483.31



NVIDIA GTX 480 (250 W) (=1/4 low power hair dryer)
1.9 million GTXs ≈ 475.99 MW!
or 475.000 hair dryers

Energy efficiency Motivation

- Green500/Top500 (June 2012)

Rank Green/Top	Site, Computer	#Cores	MFLOPS/W	LINPACK (TFLOPS)	MW to EXAFLOPS?
1/252	DOE/NNSA/LLNL BlueGene/Q, Power BQC 16C 1.60GHz	8,192	2,100.88	86.35	475.99
20/1	DOE/NNSA/LLNL BlueGene/Q, Power BQC 16C 1.60GHz	1,572,864	2,069.04	16,324.75	483.31

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

30% !

Energy efficiency
Motivation

- 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 mechanisms:
    - P-states (DVFS), C-states
  - Scientific apps are in general energy oblivious

Energy efficiency
Motivation

- 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

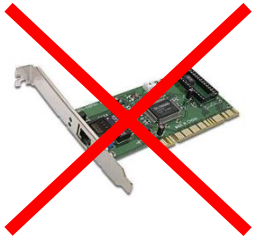
- Personal view
  - Hardware features energy saving mechanisms:
    - P-states (DVFS), C-states
  - Scientific apps are in general energy oblivious

Index

- Motivation
- Energy-aware hardware
  - Setup and tools
  - Energy-saving (processor) states
- Energy-aware software
- Conclusions

Energy-aware hardware

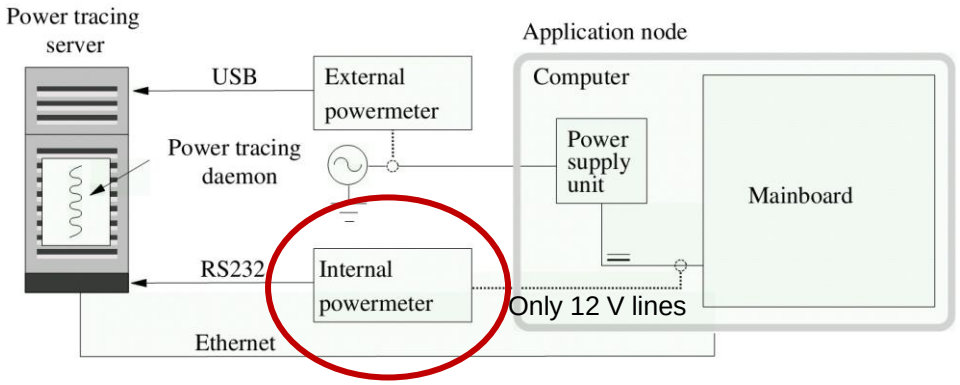
- Focus on the “processor”!

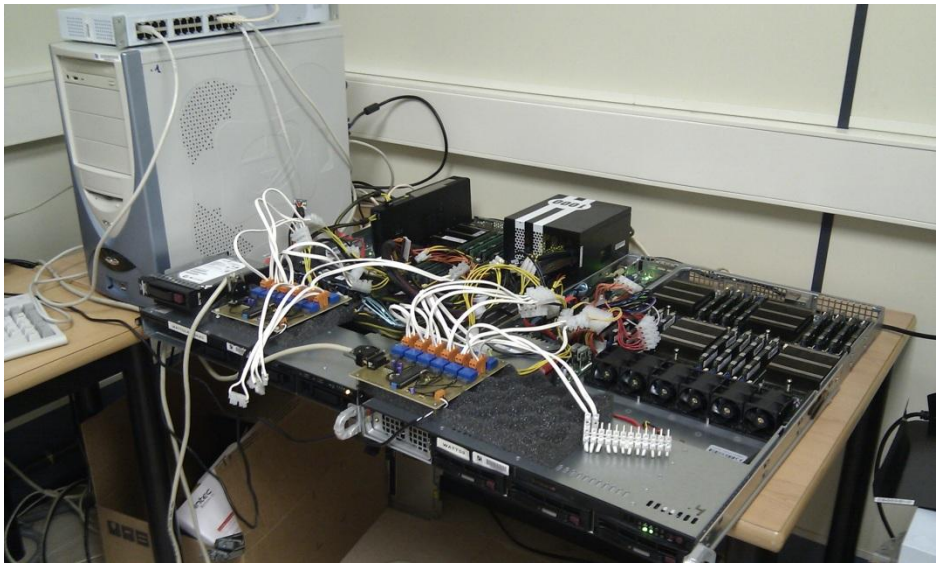



- Focus on single node performance

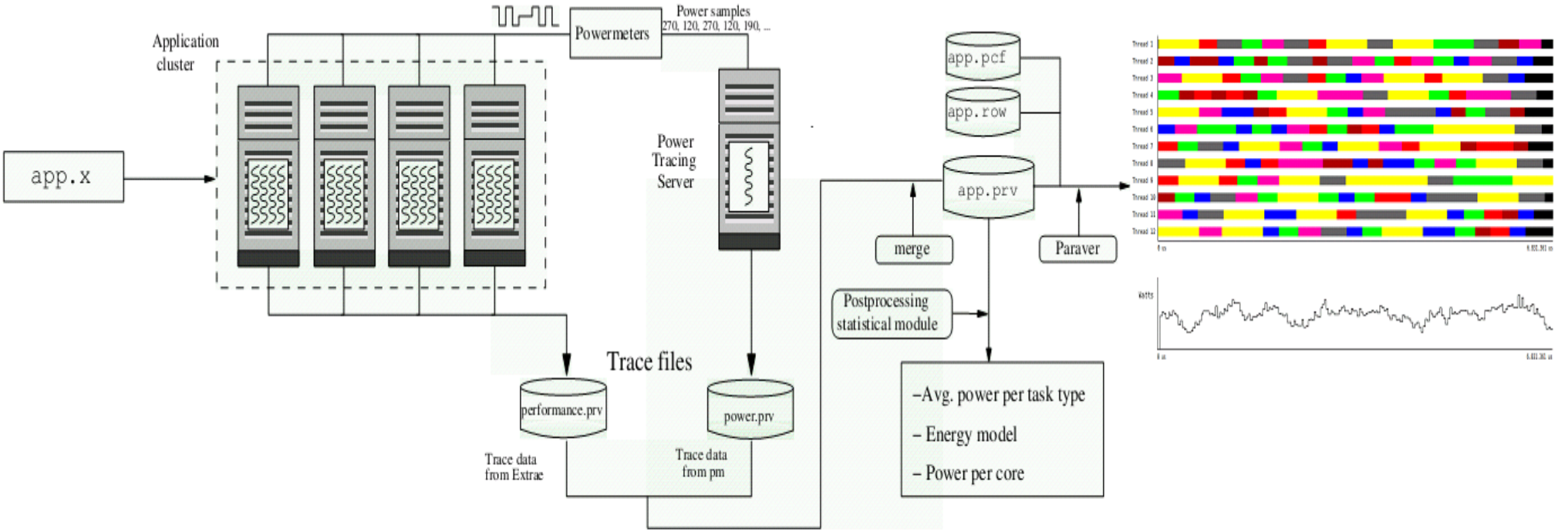
Energy-aware hardware
Setup and tools

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





Energy-aware hardware Setup and tools



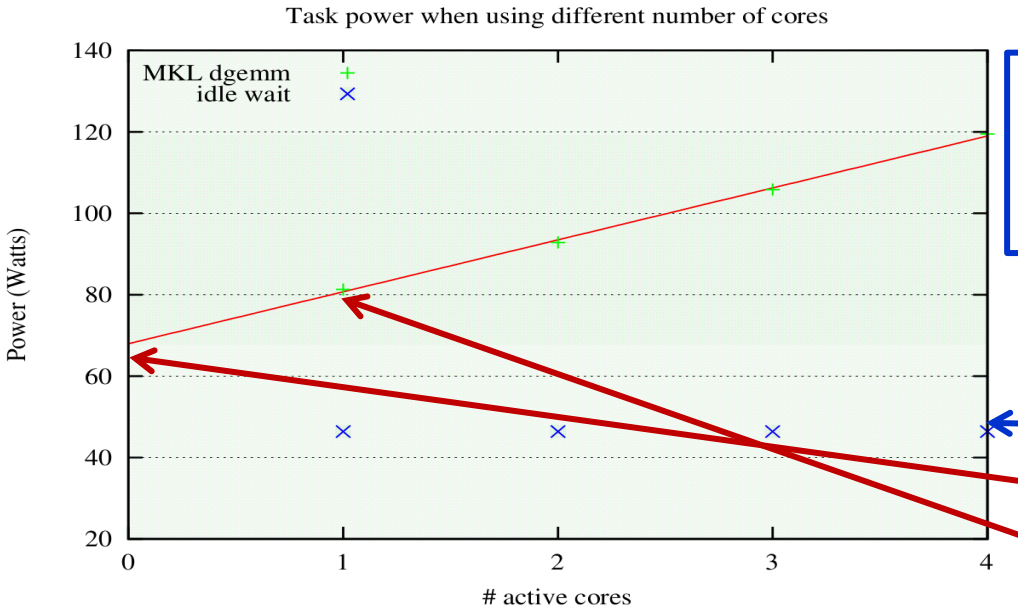
Energy-aware hardware
Setup and tools

- A simple model:

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

P^{C} is power dissipated by CPU (socket): $P^{S} + P^{D}$

P^{Y} is power of remaining components (e.g., RAM)



Server Intel:
Two Intel Xeon E5504 @ 2.0 GHz
(8 cores)

$P^{Y} \approx$ 46 W

$P^{S} \approx$ 21.5 W

$P^{D} \approx$ 12.75 W/core dgemm

Energy-aware hardware
Energy-saving states

- ACPI (*Advanced Configuration and Power Interface*): industry-standard interfaces enabling OS-directed configuration, power/thermal management of platforms





  - Revision 5.0 (Dec. 2011)
  - In the processor:
    - Performance states (P-states)
    - Power states (C-states)

Energy-aware hardware
Energy-saving states

- Performance states (P-states):
  - P0: Highest performance and power
  - Pi, i>0: As igrows, more savings but lower performance

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

Server AMD:
Two AMD Opteron 6128 cores @ 2.0 GHz (16 cores)

- $P = g\ (V^2\ f)$
- $E = \int_0^T P\ dt = g(V^2)$

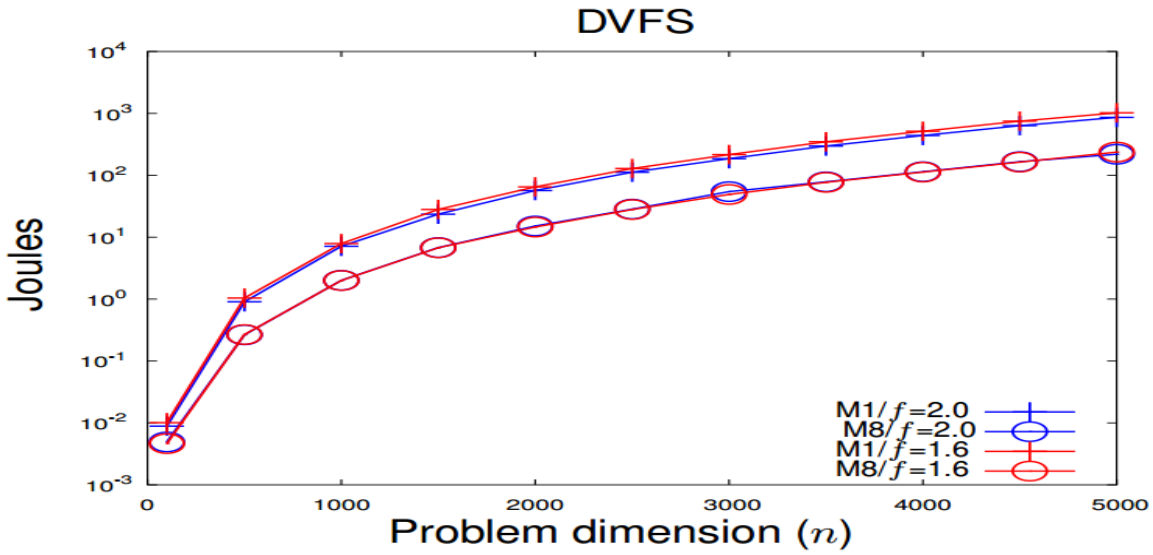
DVFS!

Energy-aware hardware
Energy-saving states

- Leveraging DVFS (transparent): Linux governors
  - **`Performance`**: Highest frequency
  - **`Powersave`**: Lowest frequency
  - **`Userspace`**: User's decision
  - **`Ondemand/conservative`**: Workload-sensitive

Energy-aware hardware
Energy-saving states

- To DVFS or not? General consensus:
  - No for compute-intensive apps.: reducing frequency increases execution time linearly



  - Yes for memory-bounded apps. as cores are idle a significant fraction of the time

Energy-aware hardware
Energy-saving states

- …but, in some platforms, reducing frequency via DVFS also reduces memory bandwidth proportionally!

P-state P_i	$V\text{CC}_i$	f_i	α_i	β_i	ΔP_i^S	ΔP_i^D	$\Delta P_i^T(16)$	BW_i	ΔBW_i
P_0	1.23	2.00	168.59	9.12	–	–	–	30.29	–
P_1	1.17	1.50	161.10	5.77	-9.52	-32.14	-17.58	24.63	-18.67
P_2	1.12	1.20	155.90	4.23	-17.09	-50.25	-28.34	20.46	-32.44
P_3	1.09	1.00	152.94	3.15	-21.47	-60.73	-33.26	17.48	-42.30
P_4	1.06	0.80	150.61	2.44	-25.73	-70.30	-39.85	14.00	-53.77

Server AMD

$$\Delta P_i^S \ (= \Delta V\text{CC}_i^2) \text{ and } \Delta P_i^D \ (= \Delta(V\text{CC}_i^2 \cdot f_i))$$

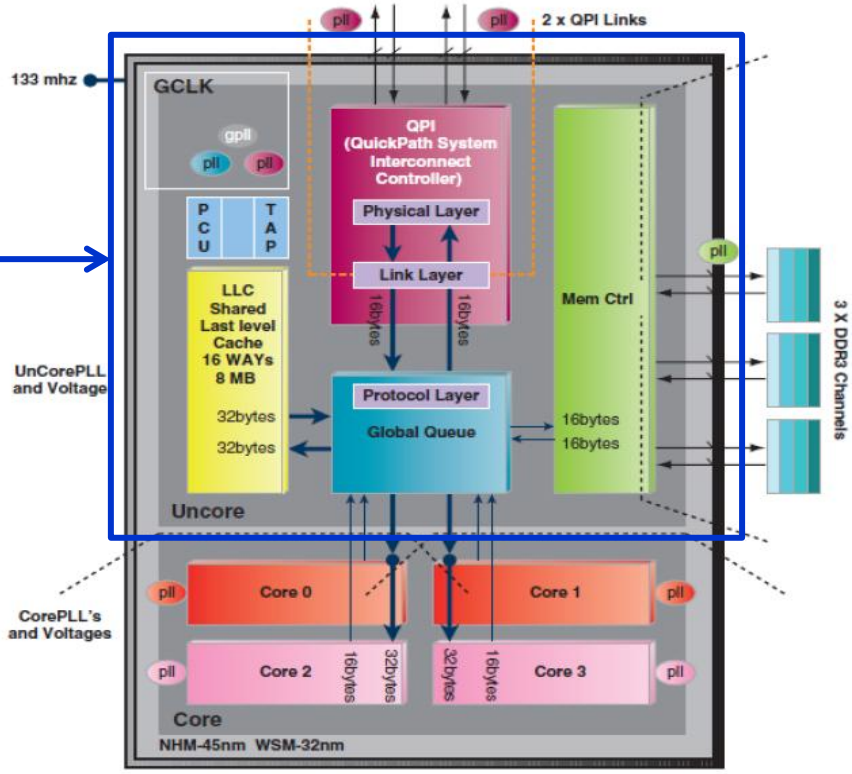
$$P_i^T(c) \ = \ P_i^S + P_i^D(c) + P^Y$$

Energy-aware hardware
Energy-saving states

- Separate power plans (Intel)

  Uncore:
  - LLC
  - Mem. controller
  - Interconnect controller
  - Power control logic

Intel Xeon 5500 (4 cores)



The Uncore: A Modular Approach to Feeding the High-performance Cores.
D. L. Hill et al. Intel Technology Journal, Vol. 14(3), 2010

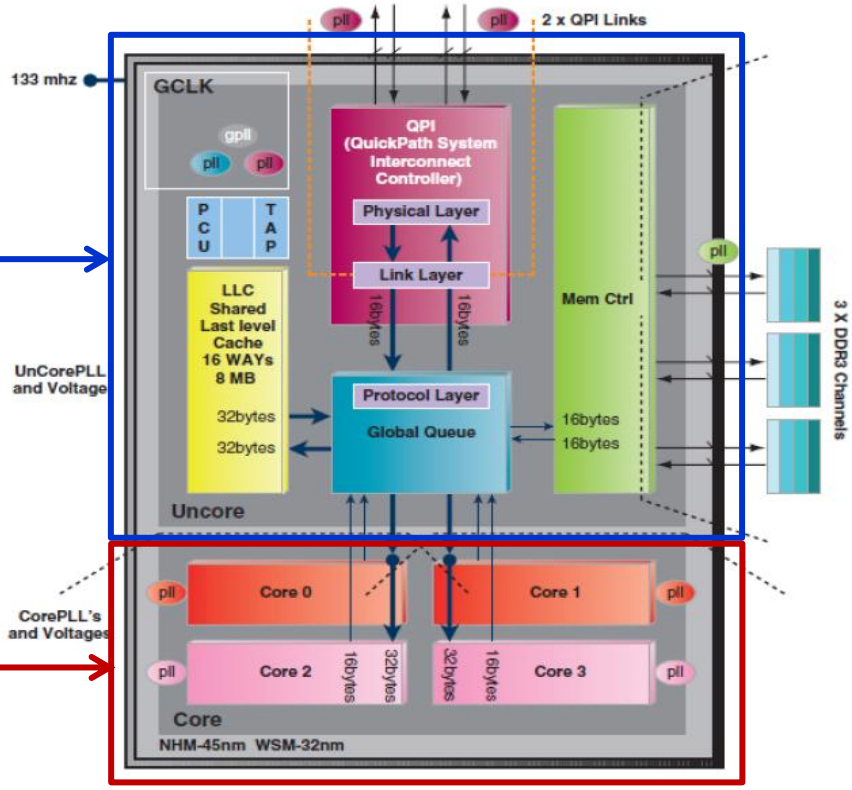
Energy-aware hardware
Energy-saving states

- Separate power plans (Intel)

  Uncore:
  - LLC
  - Mem. controller
  - Interconnect controller
  - Power control logic

  Core:
  - Execution units
  - L1 and L2 cache
  - Branch prediction logic

Intel Xeon 5500 (4 cores)



The Uncore: A Modular Approach to Feeding the High-performance Cores.
D. L. Hill et al. Intel Technology Journal, Vol. 14(3), 2010

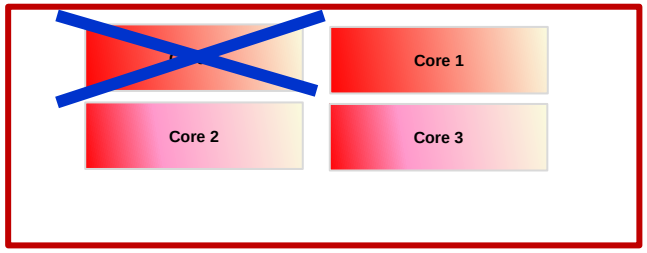
Energy-aware hardware
Energy-saving states

- Power states (C-states):
  - C0: normal execution (also a P-state)
  - Cx, x>0: no instructions being executed. As x grows, more savings but longer latency to reach C0
    - Stop clock signal
    - Flush and shutdown cache (L1 and L2 flushed to LLC)
    - Turn off core(s)

For Intel processors:

P-states at socket level but

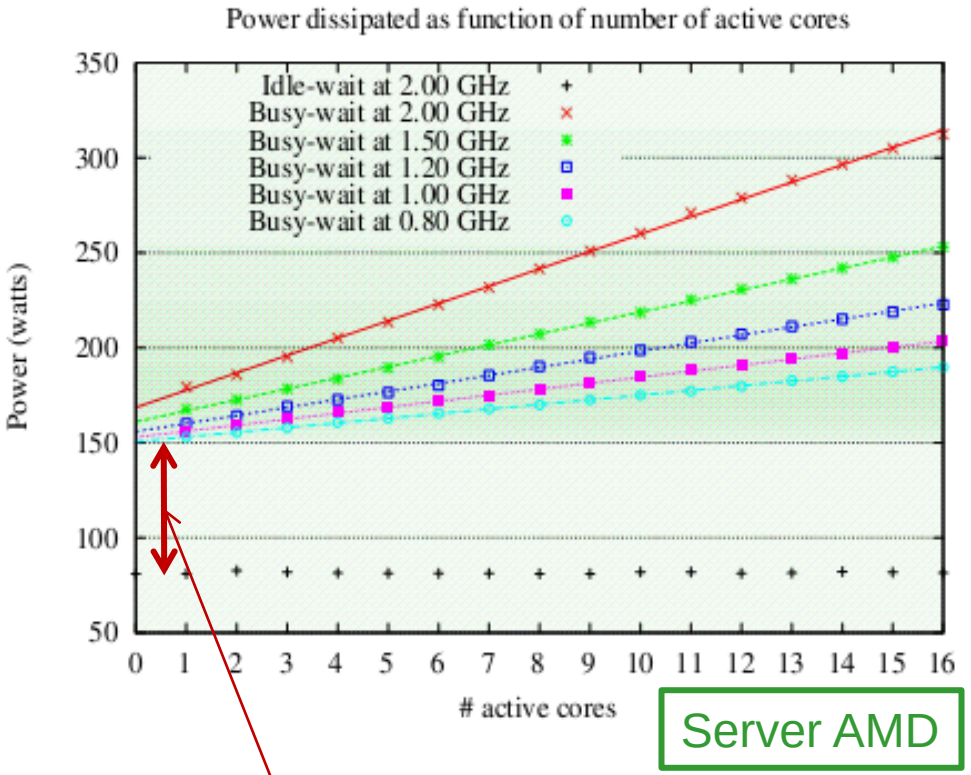
C-states at core level!



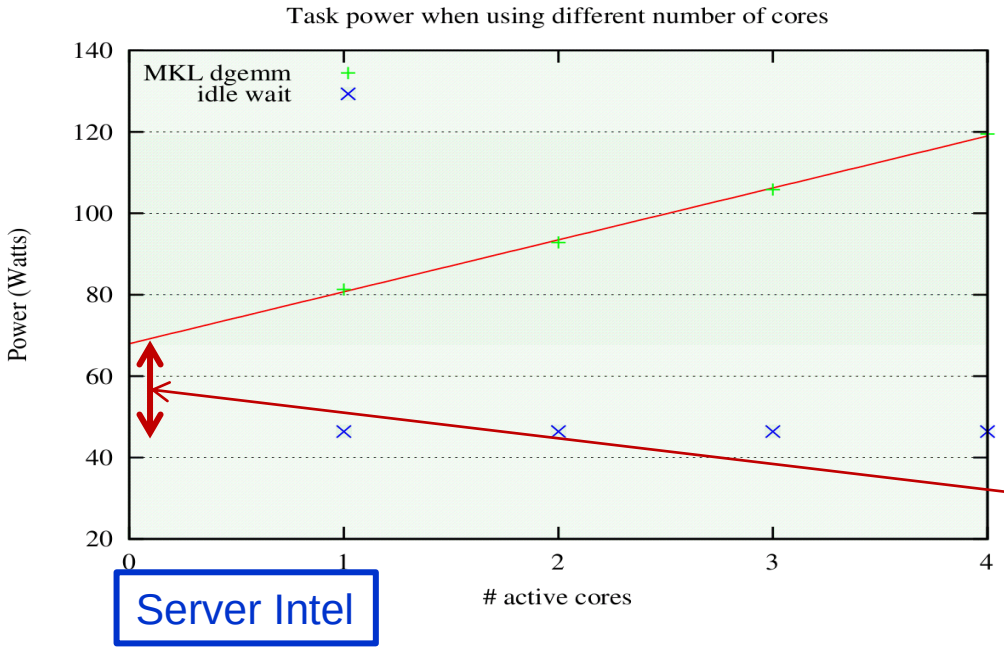
Energy-aware hardware
Energy-saving states

- Intel Core i7 processor:
  - **Core C0 State**
    - The normal operating state of a core where code is being executed
  - **Core C1/C1E State**
    - The core halts; it processes cache coherence snoops
  - **Core C3 State**
    - The core flushes the contents of its L1 instruction cache, L1 data cache, and L2 cache to the shared L3 cache, while maintaining its architectural state. All core clocks are stopped at this point. No snoops
  - **Core C6 State**
    - Before entering core C6, the core will save its architectural state to a dedicated SRAM on chip. Once complete, a core will have its voltage reduced to zero volts

Energy-aware hardware
Energy-saving states



Server AMD



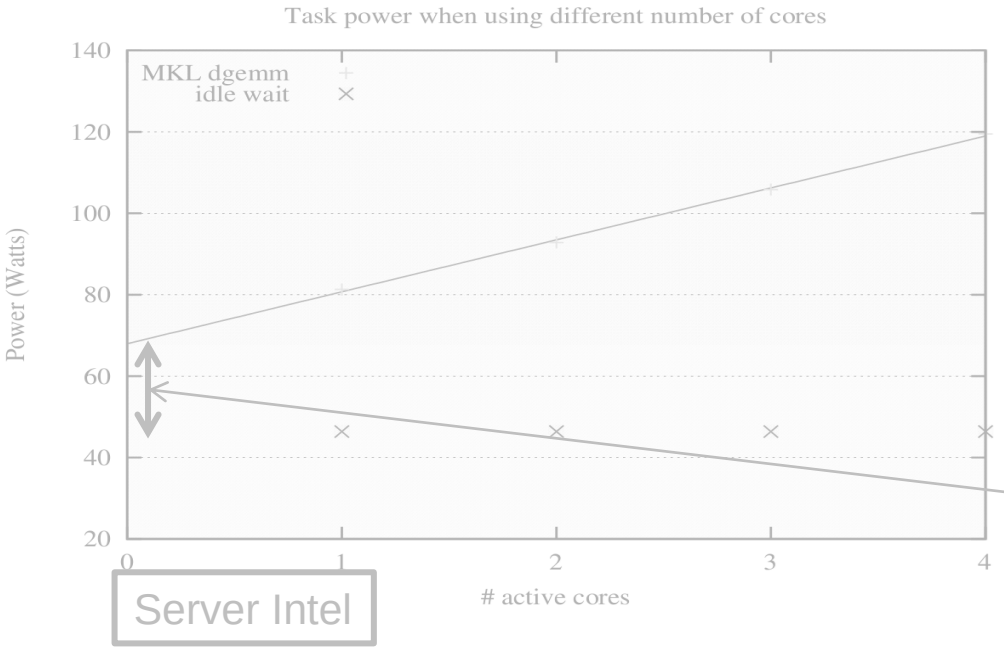
Server Intel

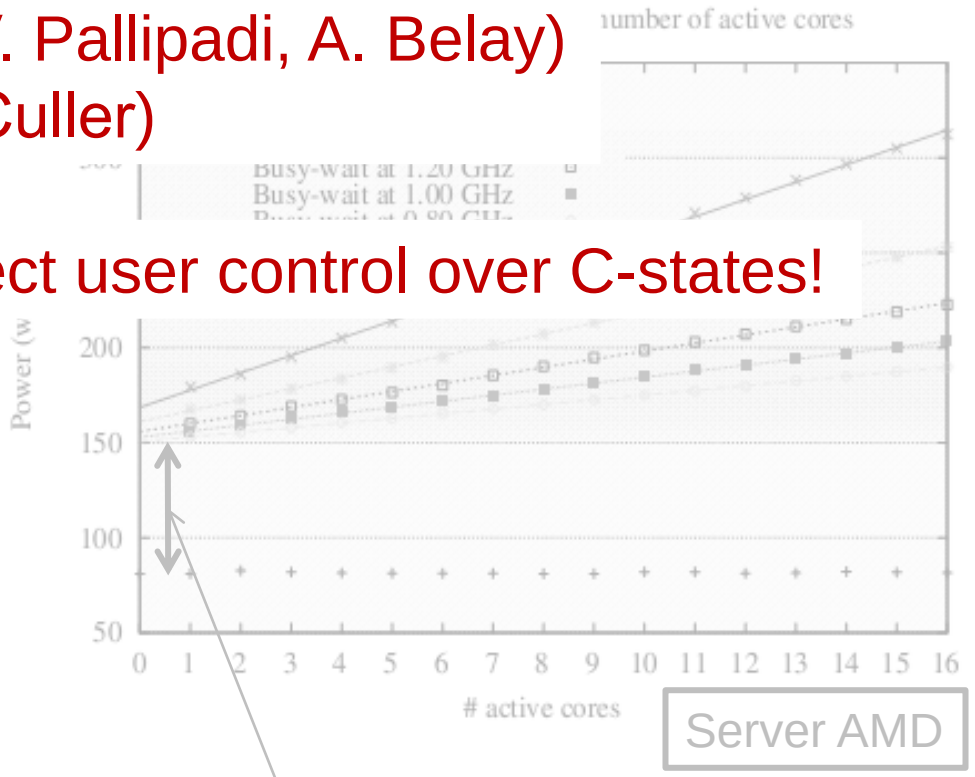
Opportunities
to save energy
via C-states!

Energy-aware hardware
Energy-saving states

"*Do nothing, efficiently…*" (V. Pallipadi, A. Belay)
"*Doing nothing well*" (D. E. Culler)

Not straight-forward. No direct user control over C-states!





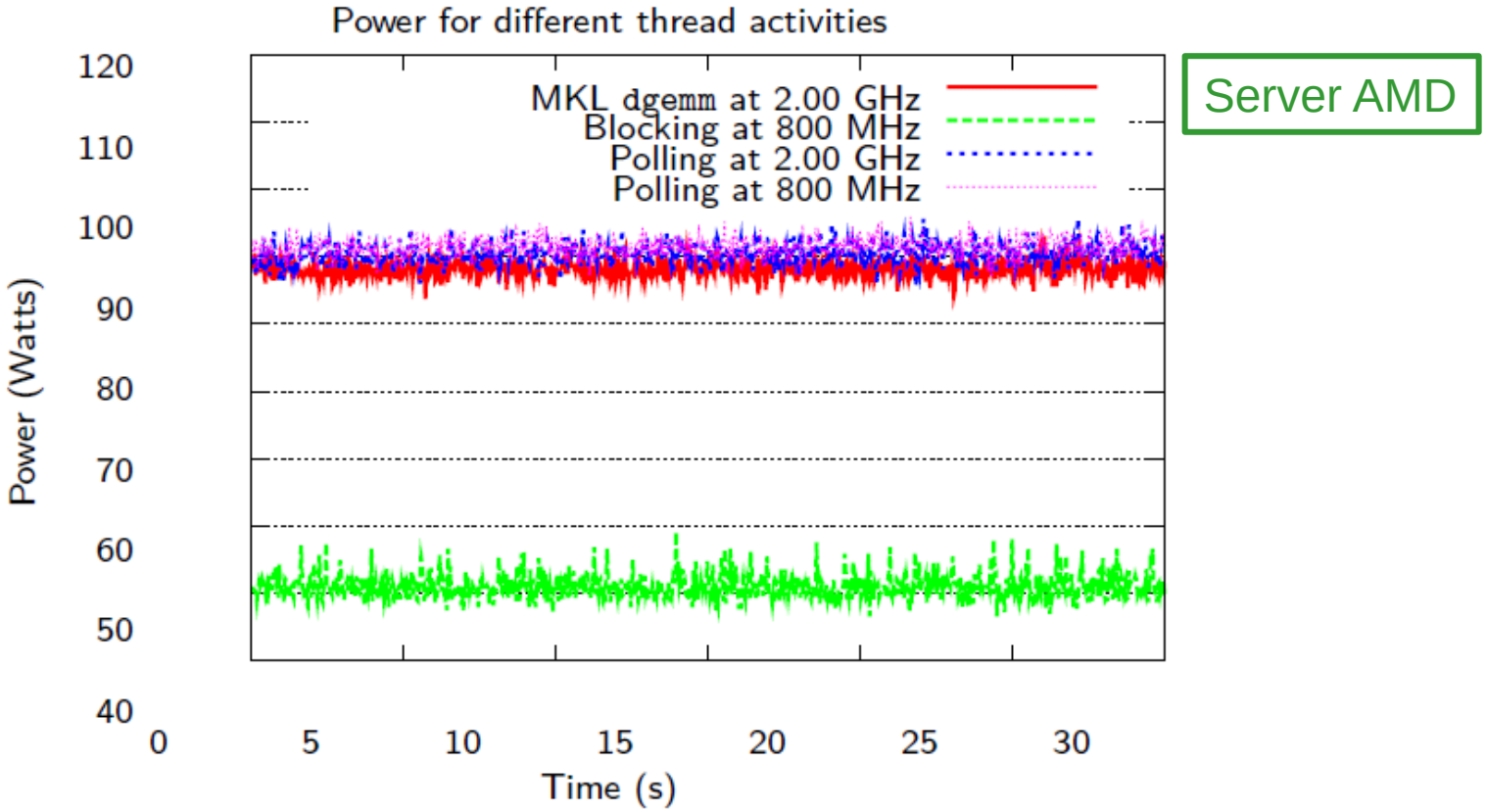
Opportunities
to save energy
via C-states!

Index

- Motivation
- Energy-aware hardware
- Energy-aware software
  - Opportunities
  - Task-parallel apps. for multicore
  - Hybrid CPU-GPU
  - MPI apps.
- Conclusions

Energy-aware software Opportunities

- Cost of core “inactivity”:



Server AMD

“*Do nothing, efficiently…*” (V. Pallipadi, A. Belay)

“*Doing nothing well*” (D. E. Culler)

Energy-aware software Opportunities

- Set necessary conditions so that hw promotes cores to energy-saving C-states: avoid idle processors doing polling!

- Scenarios, for compute-intensive or memory-bound apps.:
  - Task-parallel apps. for multicore CPUs
  - Hybrid CPU-GPU
  - MPI apps.

Energy-aware software
Task parallel apps. for multicore CPUs

- Principles of operation:
  - Exploitation of task parallelism
  - Dynamic detection of data dependencies (data-flow parallelism)
  - Scheduling tasks to resources on-the-fly

  - Surely not a new idea!

> "An Efficient Algorithm for Exploiting Multiple Arithmetic Units".
> R. M. Tomasulo.
> IBM J. of R&D, Vol. 11(1), 1967

Energy-aware software
Task parallel apps. for multicore CPUs

- "Taxonomy"

	CPU (multicore)	CPU-GPU
Linear algebra	libflame+SuperMatrix - UT PLASMA - UTK	libflame+SuperMatrix - UT MAGMA - UTK
Generic	SMPSs (OmpSs) - BSC	GPUSs (OmpSs) – BSC StarPU - INRIA Bordeaux

Energy-aware software
Task parallel apps. for multicore CPUs

- Generic runtime operation

- Automatic identification of tasks/dependencies

Application code

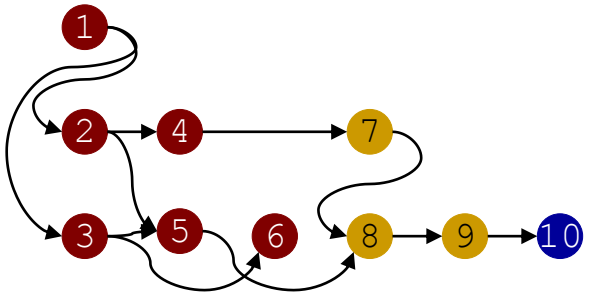
```
void task_function1( oper1... )
{
   ...
}

void task_function2( oper1... )
{
   ...
}

void task_function2( oper1... )
{
   ...
}
```

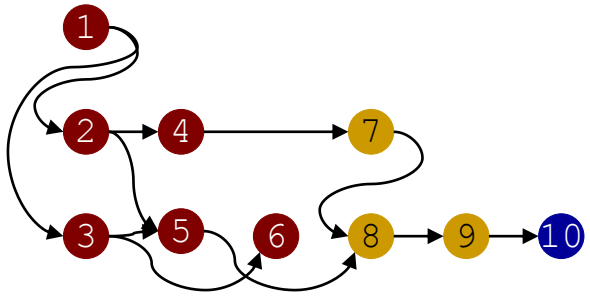
Runtime
ANALYSIS

How?
Strict order of invocations to operations (tasks) and directionality of operands (input, output, inout) identify dependencies

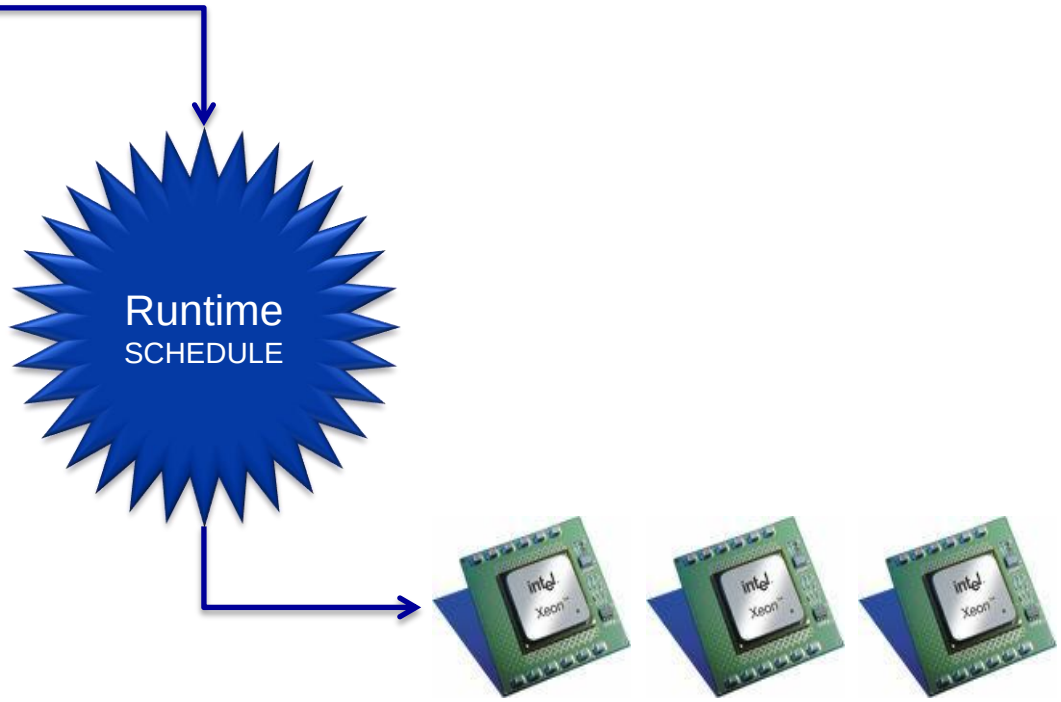


Energy-aware software
Task parallel apps. for multicore CPUs

- Generic runtime operation

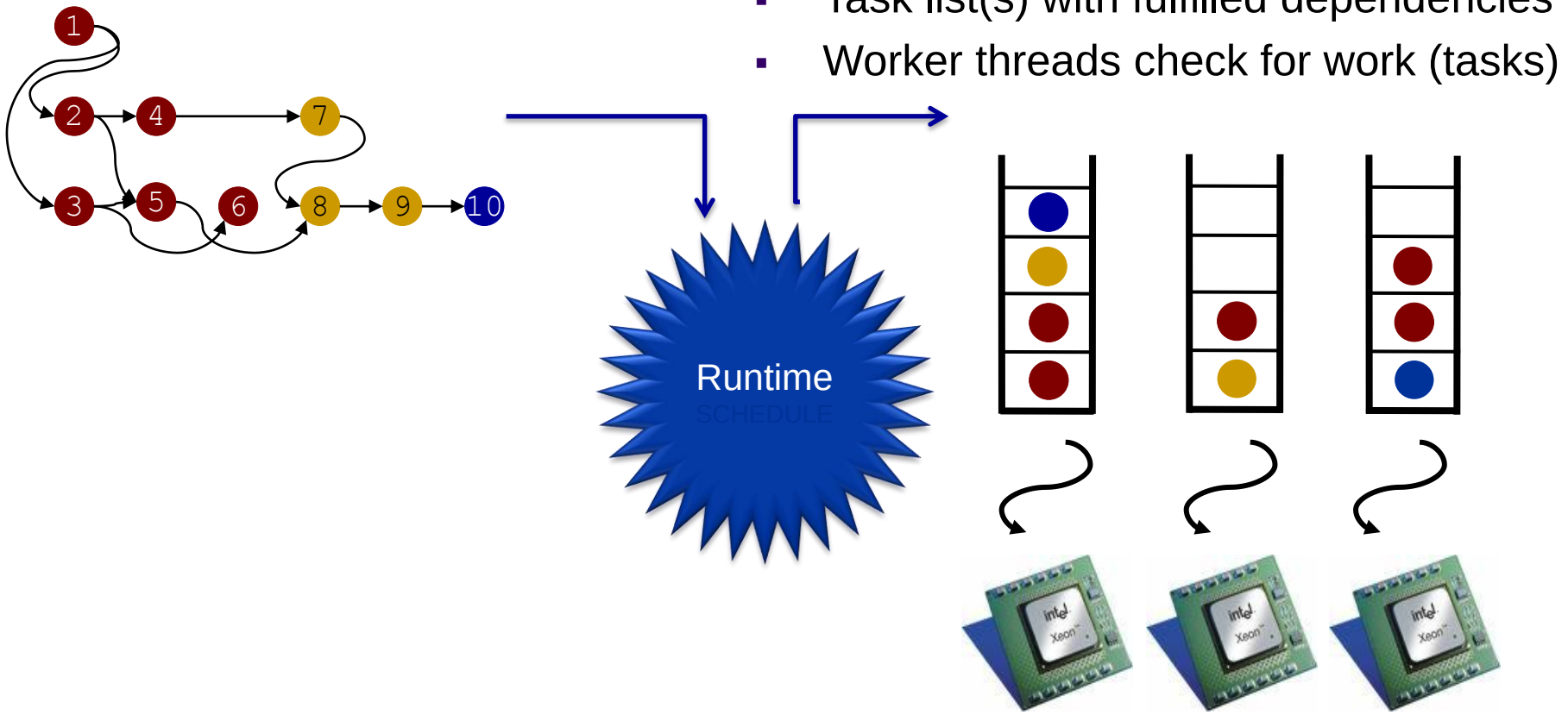


- Scheduling of tasks to computational resources (cores)



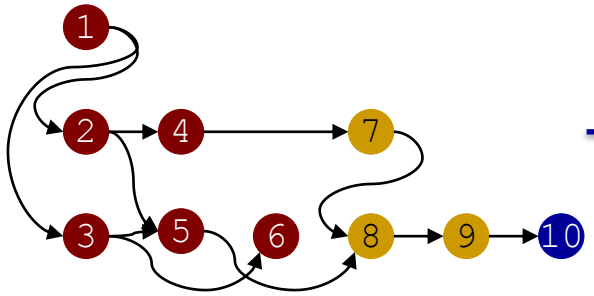
Energy-aware software
Task parallel apps. for multicore CPUs

- Generic runtime operation
  - Task list(s) with fulfilled dependencies
  - Worker threads check for work (tasks)

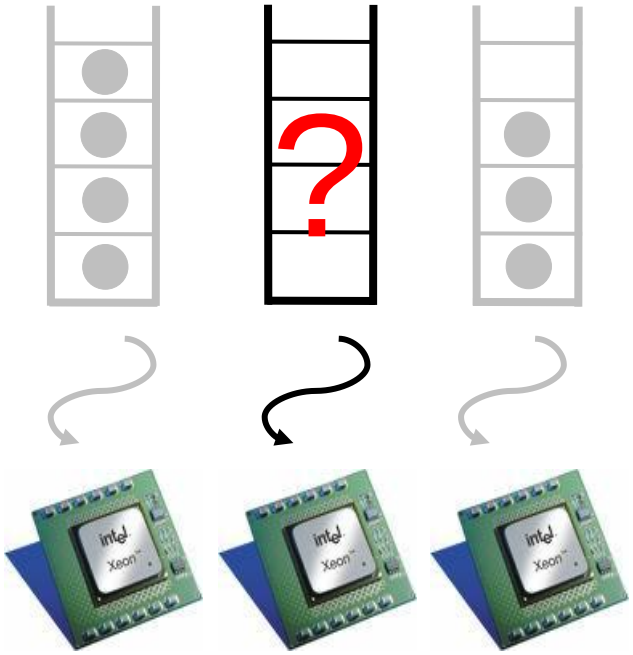


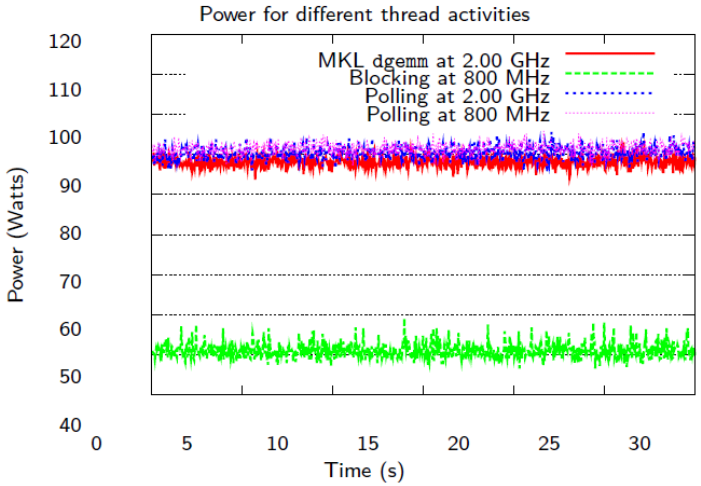
Energy-aware software
Task parallel apps. for multicore CPUs

- Generic runtime operation



- DVFS?
- Polling or blocking idle threads?

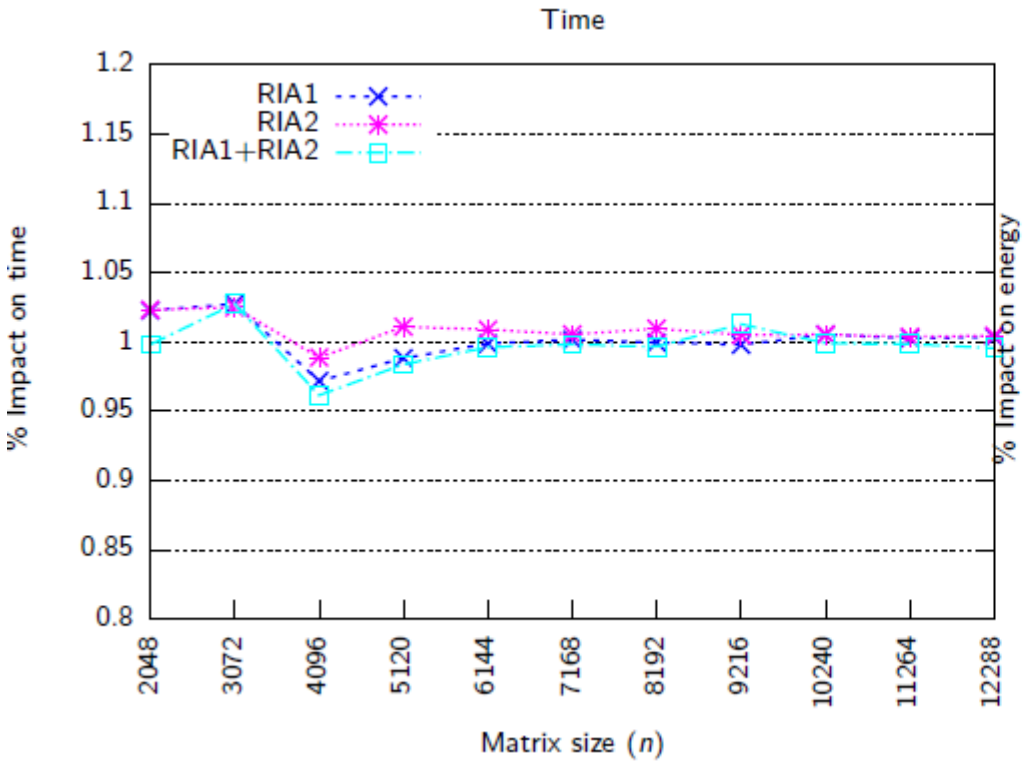


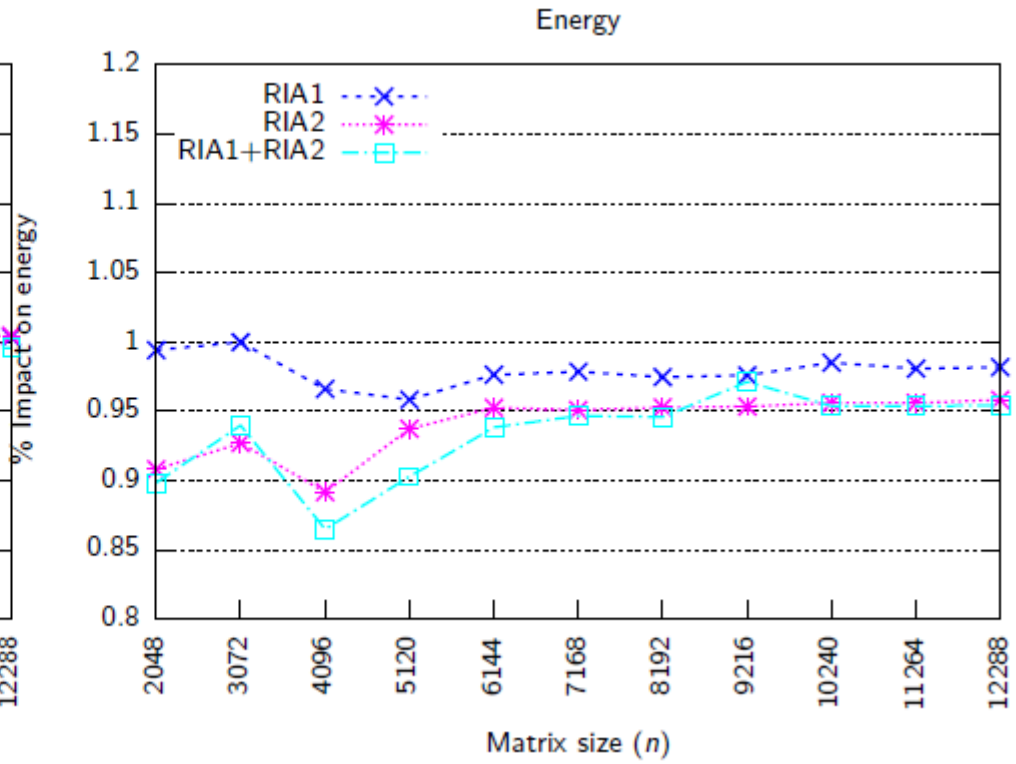


Server AMD

Energy-aware software
Task parallel apps. for multicore CPUs

- **`FLA_LU`** (LUpp fact.) from libflame + SuperMatrix runtime



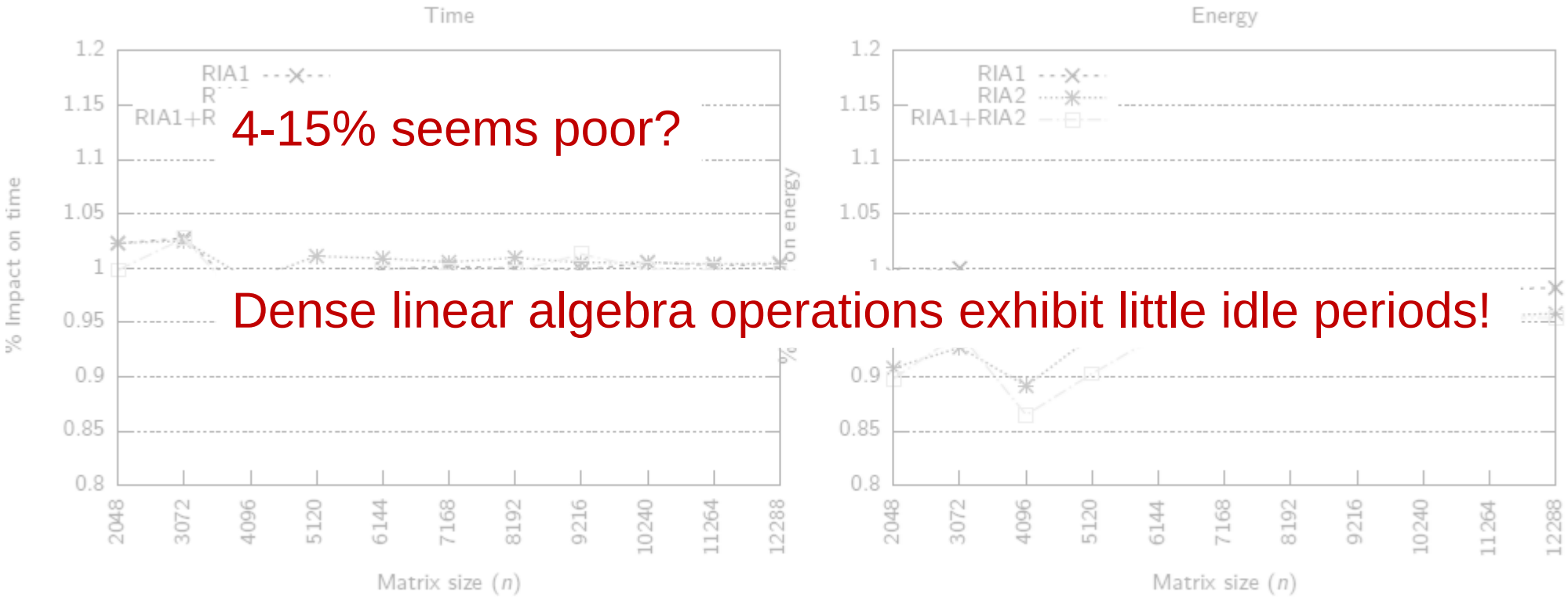


RIA1: DVFS (P-states) and polling for idle threads
RIA2: Blocking for idle threads

Server AMD

Energy-aware software
Task parallel apps. for multicore CPUs

- FLA_LU (LUpp fact.) from libflame + SuperMatrix runtime



4-15% seems poor?

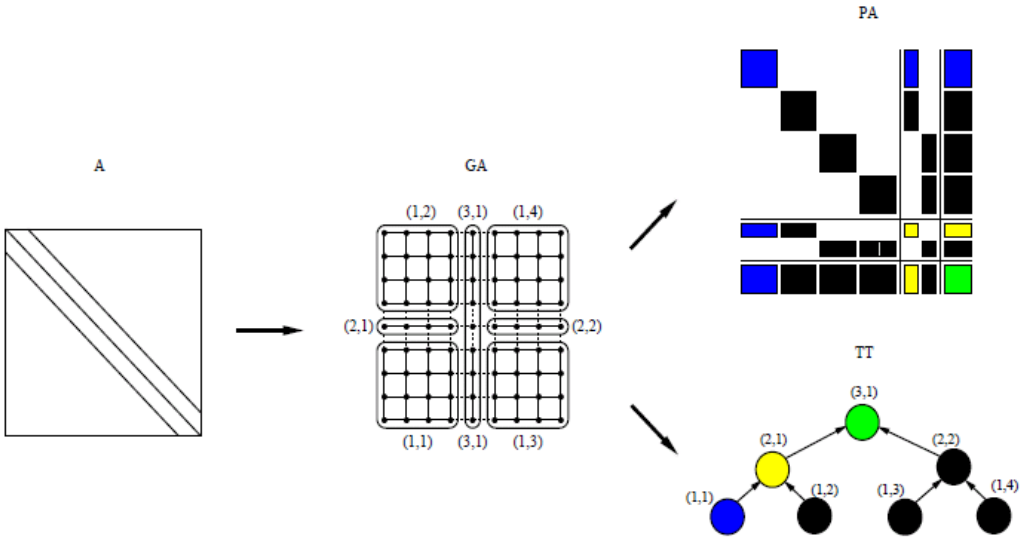
Dense linear algebra operations exhibit little idle periods!

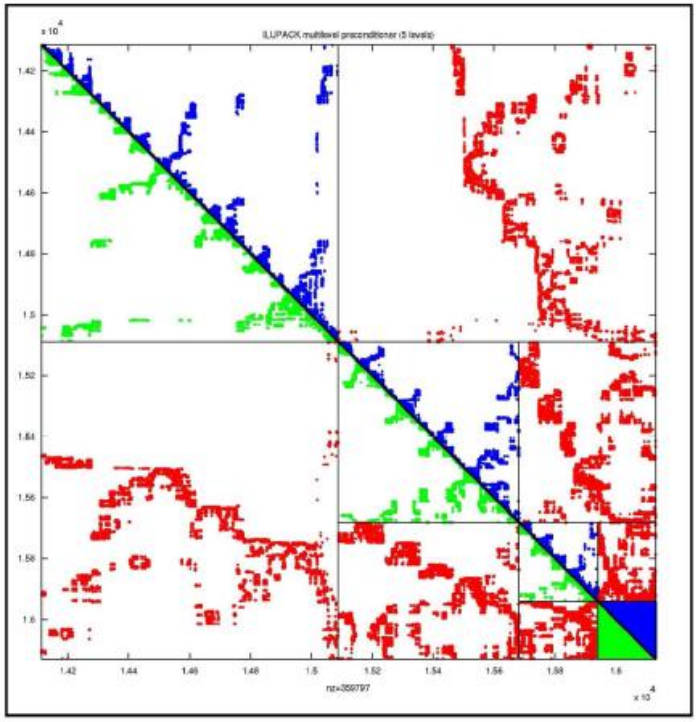
Server AMD

RIA1: DVFS (P-states) and polling for idle threads
RIA2: Blocking for idle threads

Energy-aware software
Task parallel apps. for multicore CPUs

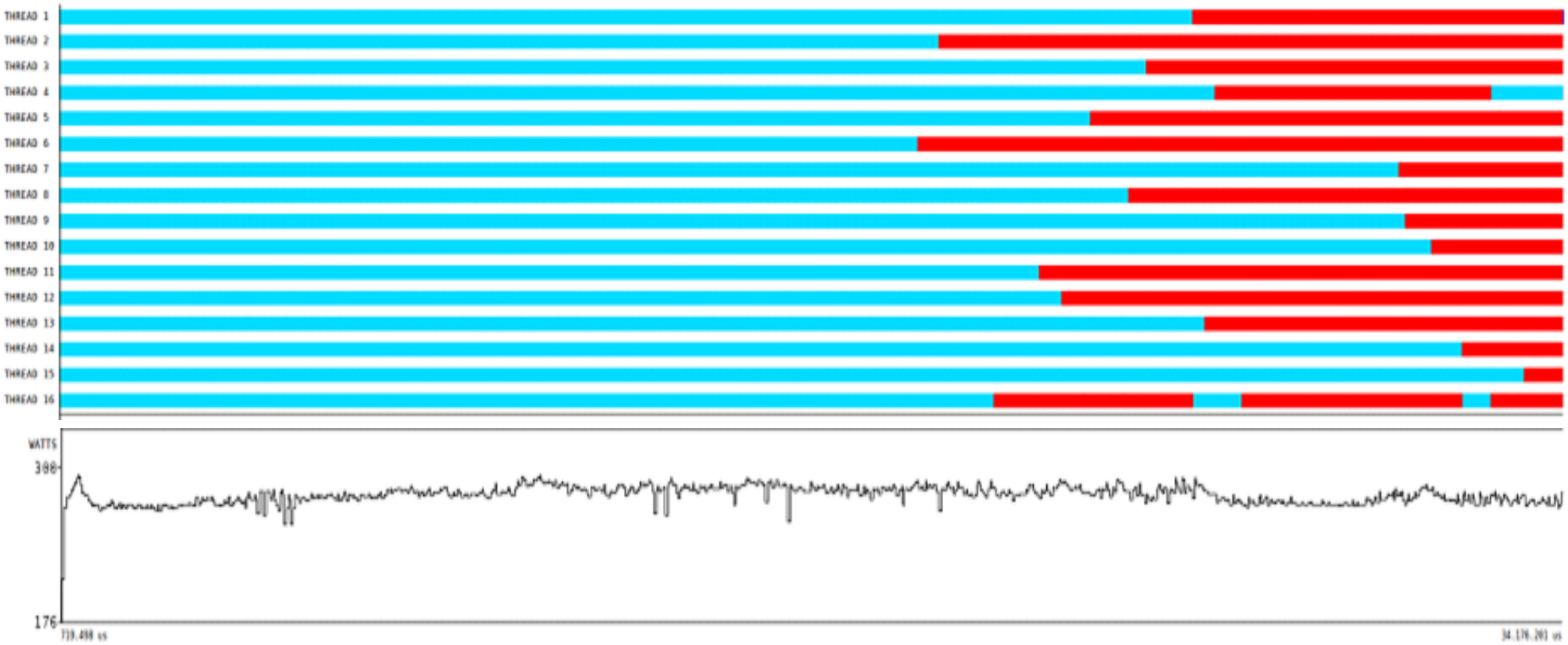
- Task-parallel implementation of ILUPACK (http://ilupack.tu-bs.de) for multicore processors with *ad-hoc runtime*
- Sparse linear system from Laplacian eqn. in a 3D unit cube





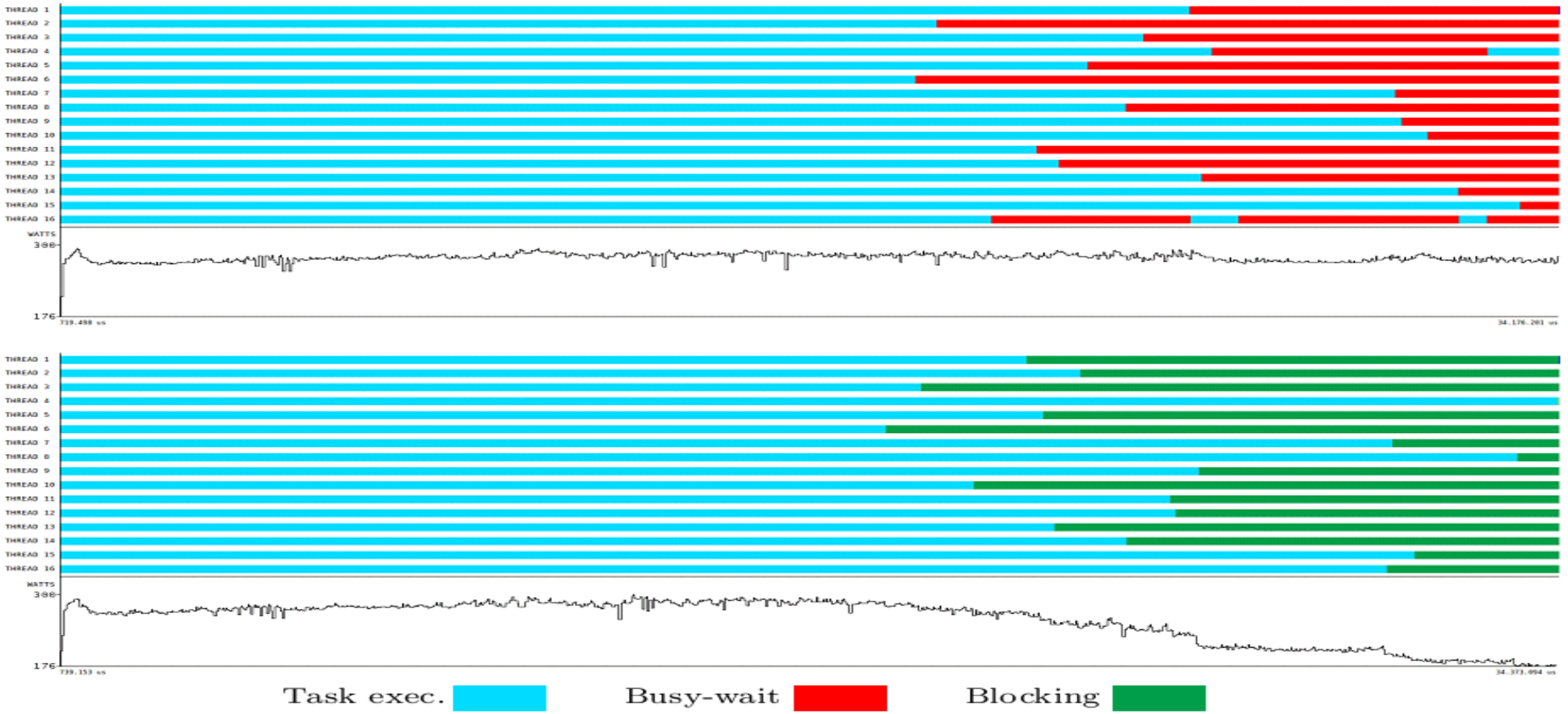
Energy-aware software
Task parallel apps. for multicore CPUs

- DVFS (P-states) and polling for idle threads



Energy-aware software
Task parallel apps. for multicore CPUs

- DVFS (P-states) vs blocking for idle threads



Energy-aware software
Task parallel apps. for multicore CPUs

- DVFS (P-states) vs polling for idle threads
  - Savings around 7% of total energy
  - Negligible impact on execution time

Energy-aware software
Task parallel apps. for multicore CPUs

- DVFS (P-states) vs polling for idle threads
  - Savings around 7% of total energy
  - Negligible impact on execution time

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

Energy-aware software Hybrid CPU-GPU

- Why CPU+GPU (for some compute-intensive apps.)?
  1. High computational power
  2. Affordable price

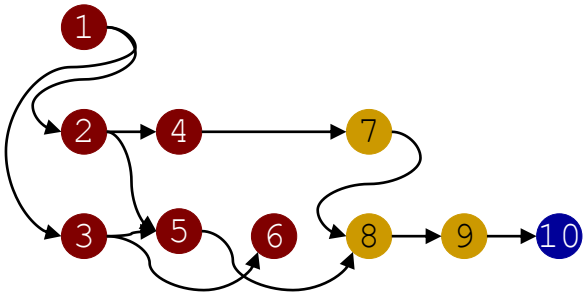
Energy-aware software Hybrid CPU-GPU

- Why CPU+GPU (for some compute-intensive apps.)?
  1. High computational power
  2. Affordable price
  3. High FLOPS per watts ratio!

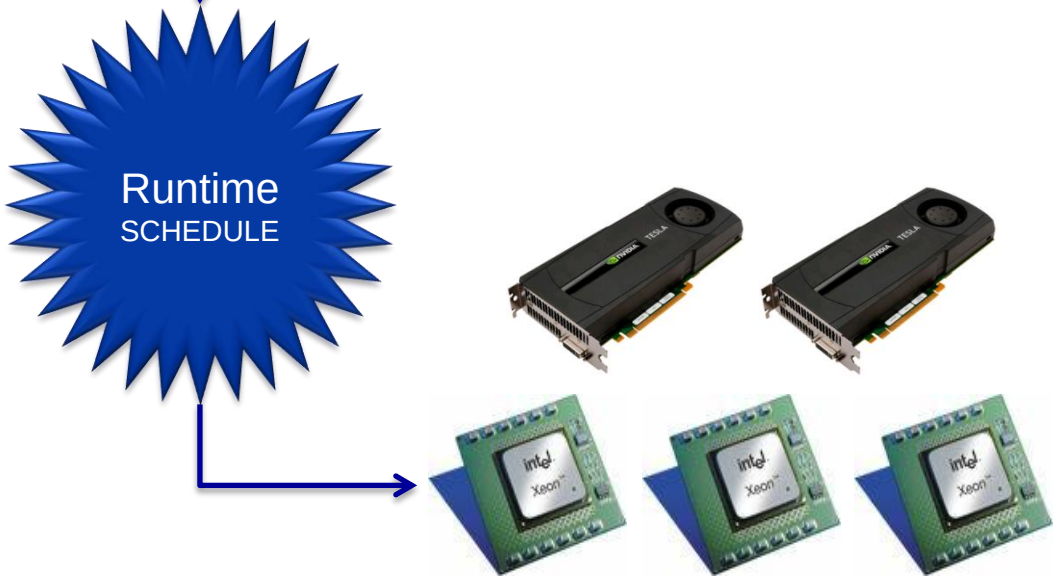
Rank Green/Top	Site, Computer	#Cores	MFLOPS/W	LINPACK (TFLOPS)	MW to EXAFLOPS?
1/252	DOE/NNSA/LLNL BlueGene/Q, Power BQC 16C 1.60GHz	8,192	2,100.88	86.35	475.99
22/--	Nagasaki University, DEGIMA Cluster, Intel i5, ATI Radeon GPU, Infiniband QDR	--	1,379.79	--	--

Energy-aware software Hybrid CPU-GPU

- Task-parallel apps. for hybrid CPU-GPU?

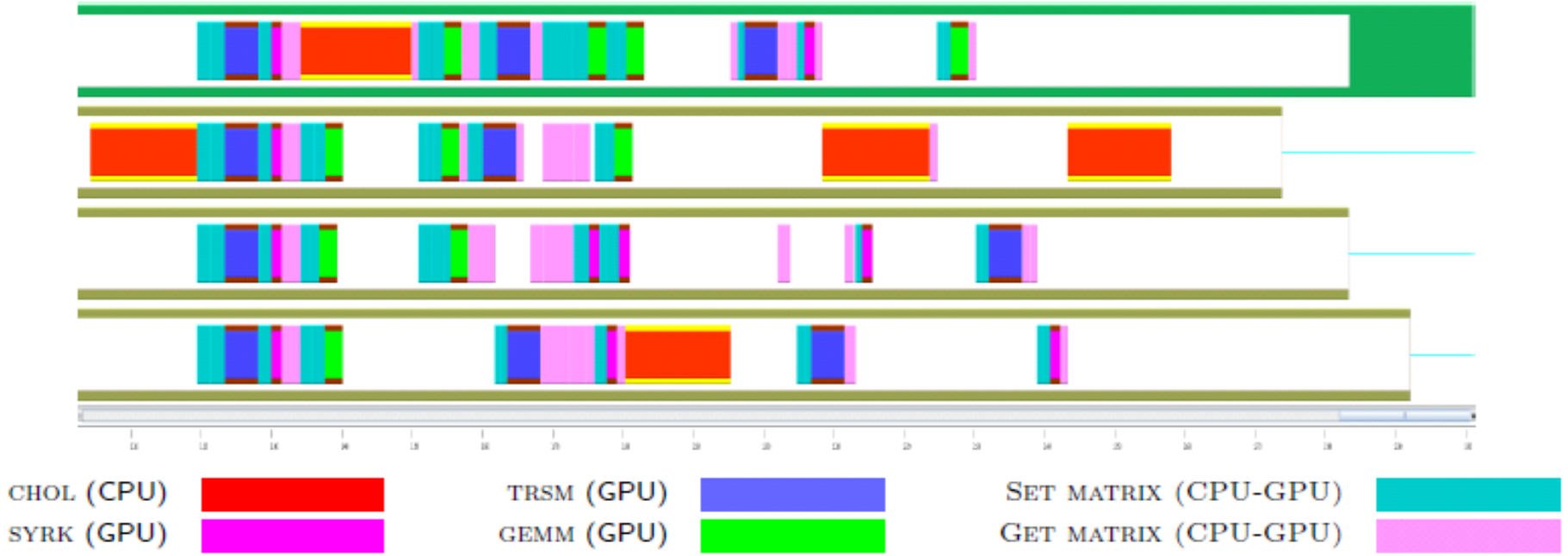


- Scheduling of tasks to heterogeneous computational resources
  - Choose the appropriate resource, CPU or GPU?
  - Reduce PCI-e communication



Energy-aware software Hybrid CPU-GPU

- **`FLA_Chol`** (Cholesky fact.) from `libflame+SuperMatrix` on 7,680x7,680 s.p.d. matrix



Server Intel2:
Intel Xeon E5540 @ 2.83 GHz (4 cores) + NVIDIA Tesla S2050 (4 GPUs)

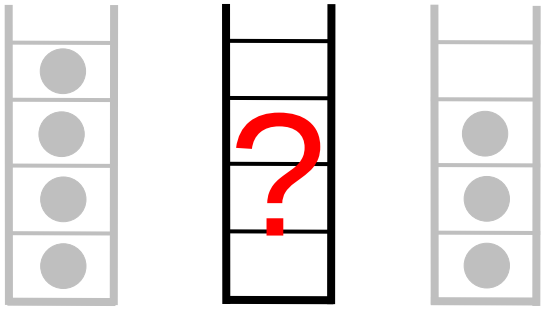
Energy-aware software Hybrid CPU-GPU

- **`FLA_Chol`** (Cholesky fact.) from `libflame+SuperMatrix` on 7,680x7,680 s.p.d. matrix

Server Intel2:
Intel Xeon E5540 @ 2.83 GHz (4 cores) +
NVIDIA Tesla S2050 (4 GPUs)

Energy-aware software Hybrid CPU-GPU

- “Sources” of idle CPU threads?
  - Case 1. No tasks with fulfilled dependencies available



→ Modified runtime to block idle CPU threads (same as multicore case)

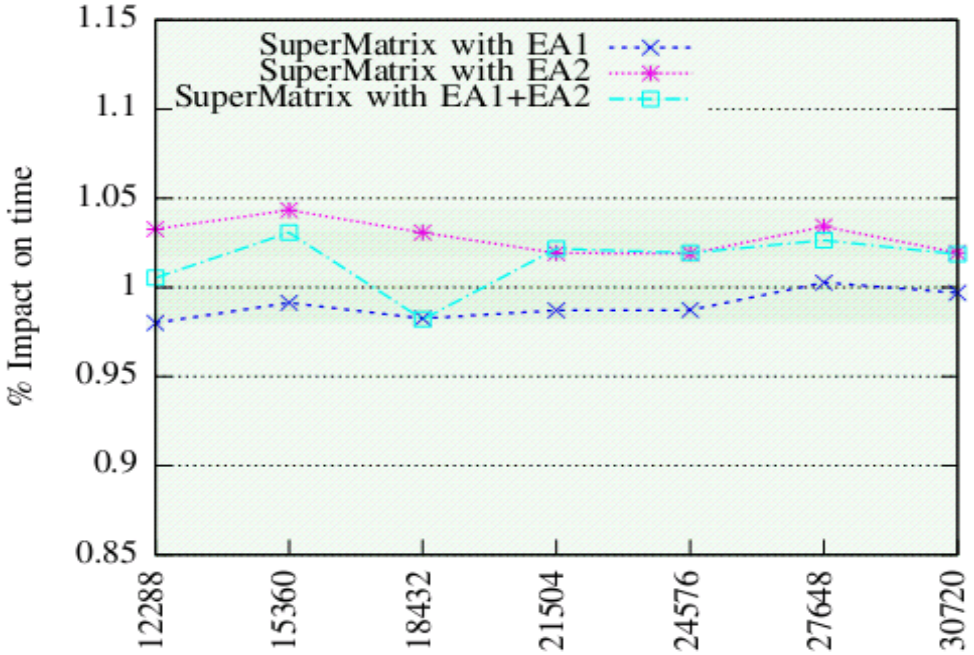
Energy-aware software Hybrid CPU-GPU

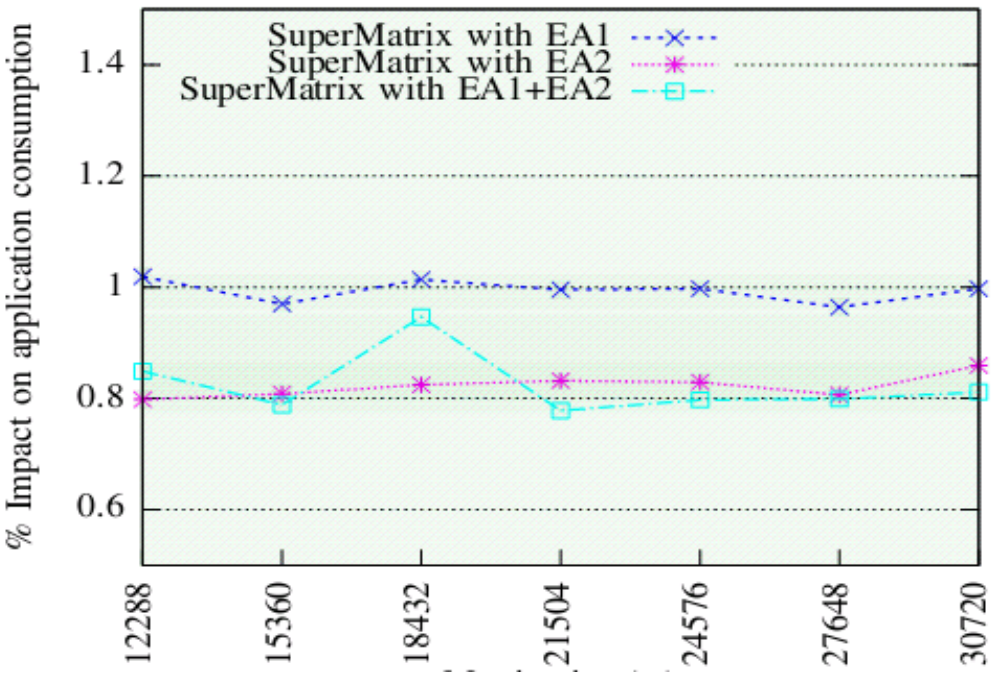
- “Sources” of idle CPU threads?
  - Case 2. CPU thread waiting for task being executed on GPU

→ Set blocking operation mode (synchronous) for CUDA kernels

Energy-aware software Hybrid CPU-GPU

- **`FLA_Chol`** (Cholesky fact.) from `libflame+SuperMatrix` on 7,680x7,680 s.p.d. matrix

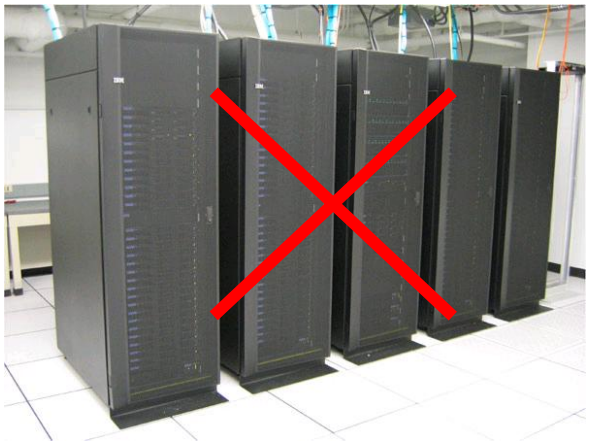




EA1: blocking for idle threads without task
EA2: blocking for idle threads waiting for GPU

Energy-aware software MPI apps.

- Focus on the processor and single node performance

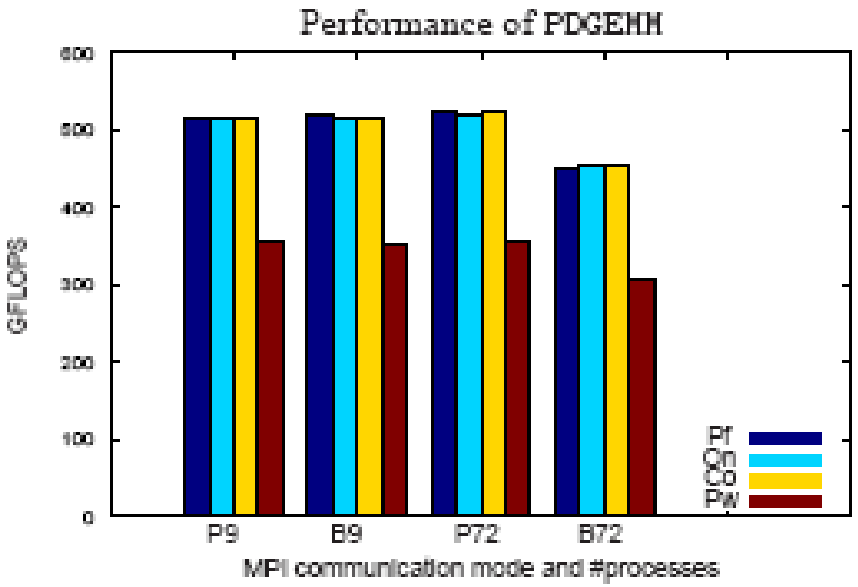


Energy-aware software MPI apps.

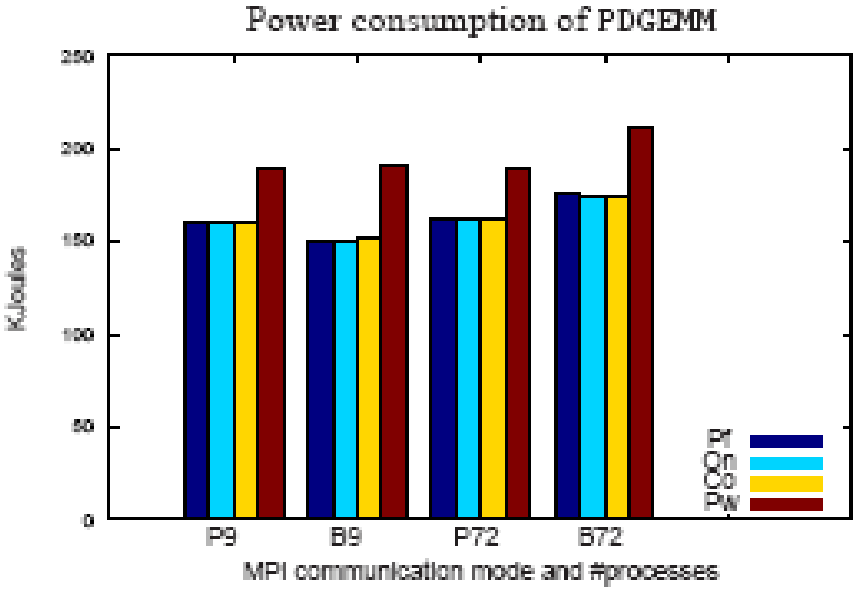
- Some implementations of MPI feature blocking/polling operation modes (e.g., MVAPICH2 1.5.1)
  - Communication thread blocks/polls for completion of data transfer
- These can be combined with:
  - Linux governor modes
  - Leverage node concurrency via MPI processes or threads

Energy-aware software MPI apps.

- **PDGEMM** (Matrix multiplication) from ScaLAPACK on matrices with 45,000 rows/columns



	Performance of PDGEMM (GFLOPS)			
	P9	B9	P72	B72
Pf	517.1	518.3	524.9	451.6
On	517.2	517.2	521.8	456.3
Co	517.3	517.2	522.6	453.9
Pw	354.7	354.1	356.2	308.1



	Power consumption of PDGEMM (KJoules)			
	P9	B9	P72	B72
Pf	160.5	149.7	161.7	175.7
On	160.1	150.4	162.6	174.1
Co	160.8	151.2	162.5	174.4
Pw	189.4	190.6	189.6	210.6

Conclusions

- 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

Thanks to…

UJI:	J. I. Aliaga, M. F. Dolz, R. Mayo
U. Politécnica de Valencia:	P. Alonso
KIT (Germany):	H. Anzt
BSC (Spain):	R. M. Badia, J. Planas
U. Complutense Madrid (Spain):	F. D. Igual
The University of Texas at Austin:	R. van de Geijn

More information

- "Tools for power and energy analysis of parallel scientific applications". P. Alonso, R. Badia, J. Labarta, M. Barreda, M. F. Dolz, R. Mayo, E. S. Quintana-Ortí, R. Reyes. ICPP 2012

  → Tools for power/energy analysis

- "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

  → Power model for dense linear algebra (L.A.) on multicore

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

  → Energy-aware schedules of dense L.A. on muticore

- "Leveraging task-parallelism in energy-efficient ILU preconditioners". J. I. Aliaga, M. F. Dolz, A. F. Martín, E. S. Quintana-Ortí. ICT-GLOW 2012

  → Power model for sparse L.A. + energy-aware runtime on multicore

- "Reducing energy consumption of dense linear algebra operations on hybrid CPU-GPU platforms". P. Alonso, M. F. Dolz, F. D. Igual, R. Mayo, E. S. Quintana. ISPA 2012

  → Energy-aware runtime on multicore + GPU

- "Analysis of strategies to save energy for message-passing dense linear algebra kernels". M. Castillo, J. C. Fdez., R. Mayo, E. S. Quintana, V. Roca. PDP 2012

  → Energy-aware for message-passing dense linear algebra

Conclusions

Questions?