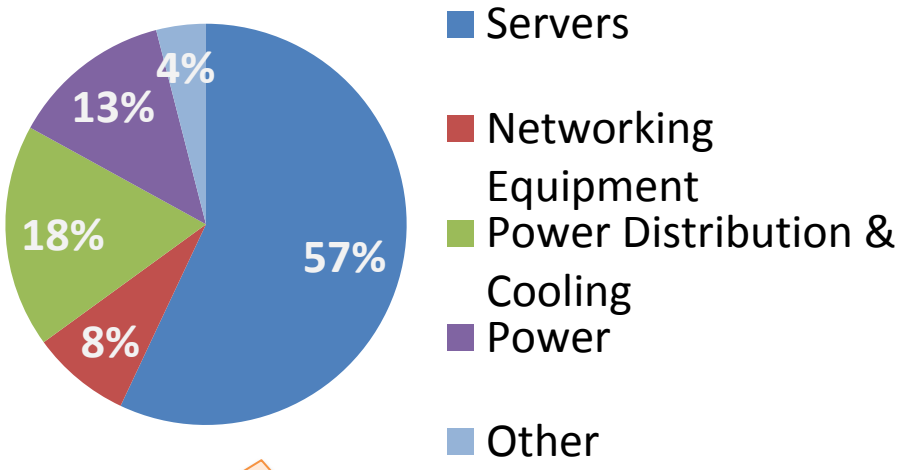
Dynamic Fine-Grained Scheduling for Energy-Efficient Main-Memory Queries

Iraklis Psaroudakis (EPFL, SAP AG), Thomas Kissinger (TU Dresden), Danica Porobic (EPFL), Thomas Ilsche (TU Dresden), Erietta Liarou (EPFL), Pinar Tözün (EPFL), Anastasia Ailamaki (EPFL), Wolfgang Lehner (TU Dresden)

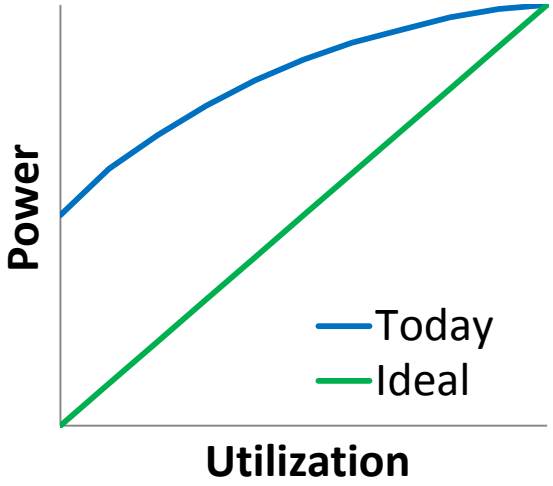
Why care about power?

Monthly datacenter costs [J. R. Hamilton]



30% power-related
Dynamic fraction increasing

Energy proportionality



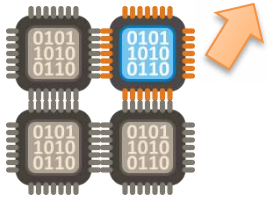
Getting there:

- Power management features
- Power-aware software

We need to make DBMS power-aware

Power management features

- Dynamic voltage and frequency scaling (DVFS)
- Turbo boost



> 2.9GHz

1.2GHz 2.9GHz

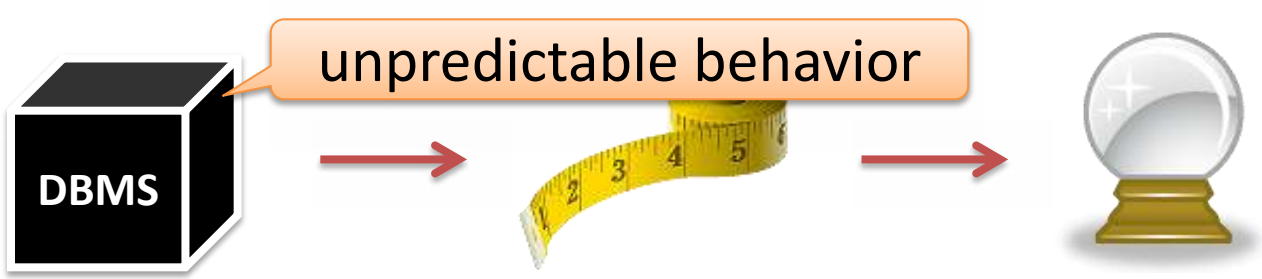
- Idle states (C-states)

- Power-related H/W counters

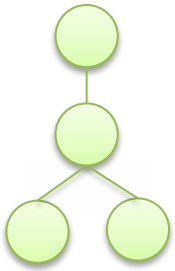
We can exploit these to improve energy efficiency

Current approaches

- Black box
  - e.g. dynamic concurrency throttling [TPDS13]



- Query optimizer [ICDE10]



+ power costs

coarse-grained, without low-level tuning

We need fine-grained energy-awareness in the database

Fine-grained energy-aware scheduling

How do you schedule this query plan?

- parameters:
  - parallelism
  - thread placement
  - data placement
  - dynamic voltage and frequency scaling (DVFS)

Calibration of operators under different parameters

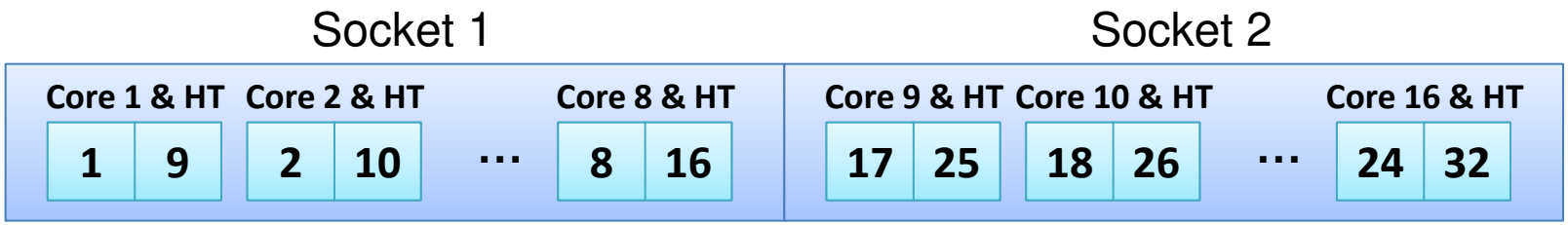
Concurrent partitioned scans

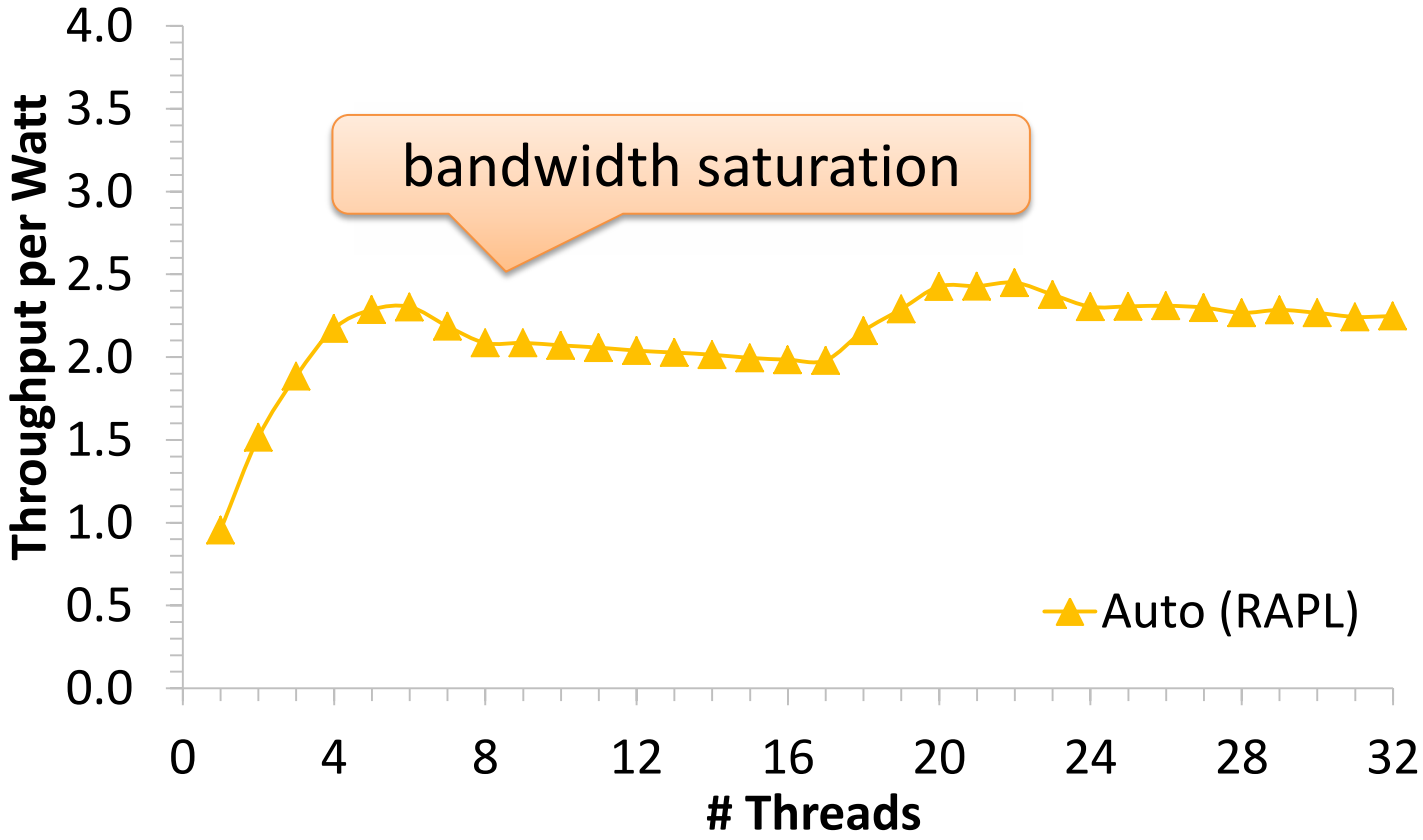
- Each thread scans 128MB of integers for 5 secs
- Maximize

$$performance\ per\ power = \frac{throughput}{power}$$

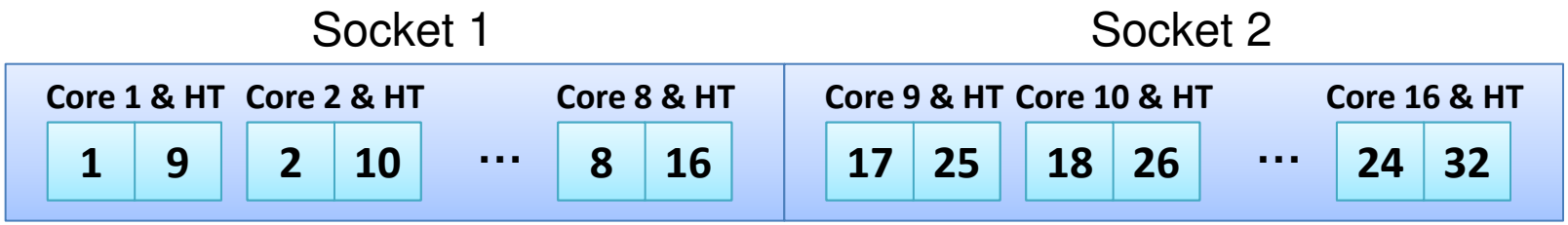
  - under different parallelism, scheduling, and frequency settings
- Machine
  - Two 8-core Intel Xeon E5-2690, HT enabled, 64GB RAM, frequencies from 1.2GHz to 2.9GHz
- Power measurements
  - Hardware performance counters RAPL (CPU & DRAM)
  - External equipment

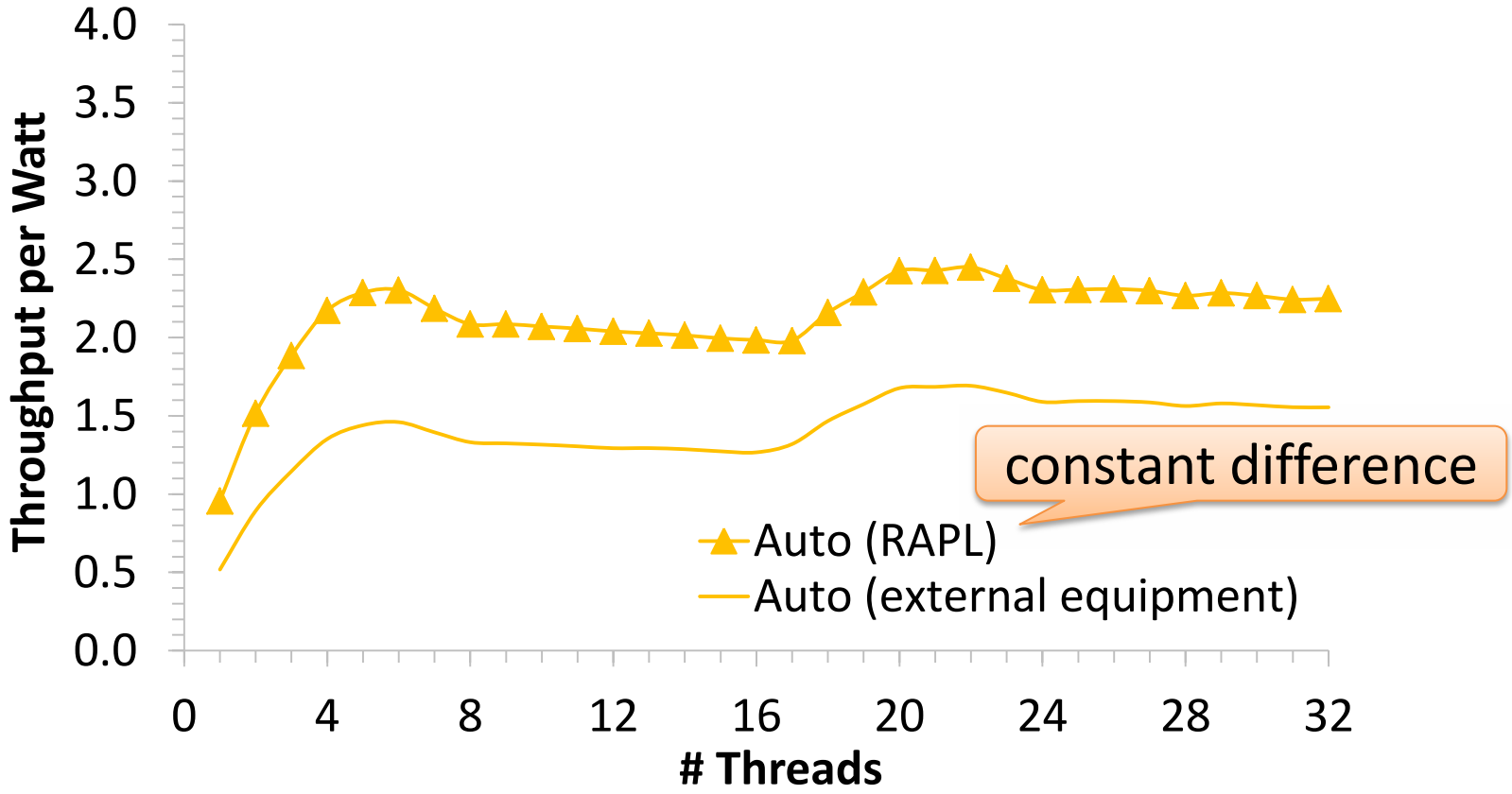
Socket-fill scheduling



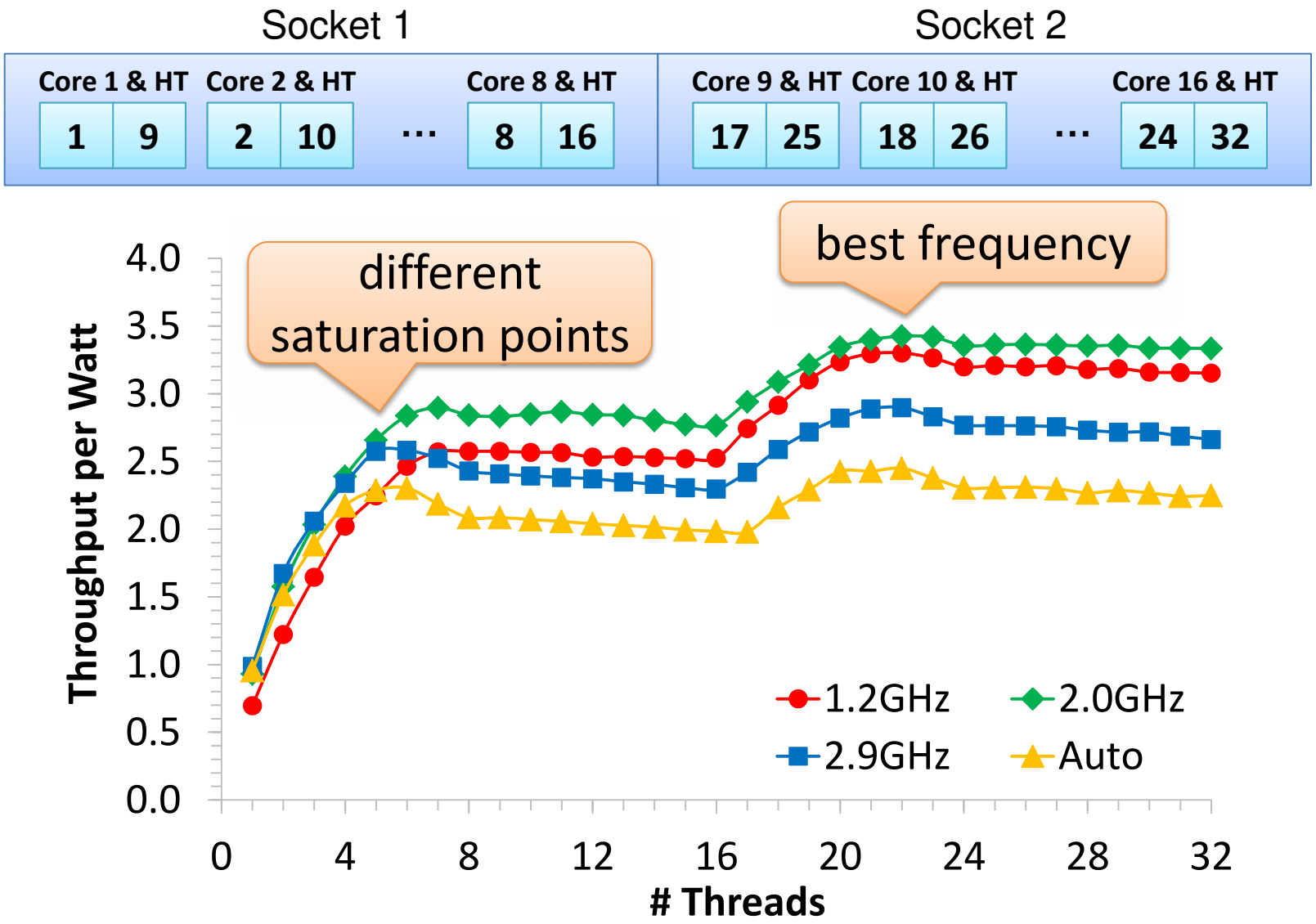


Socket-fill scheduling

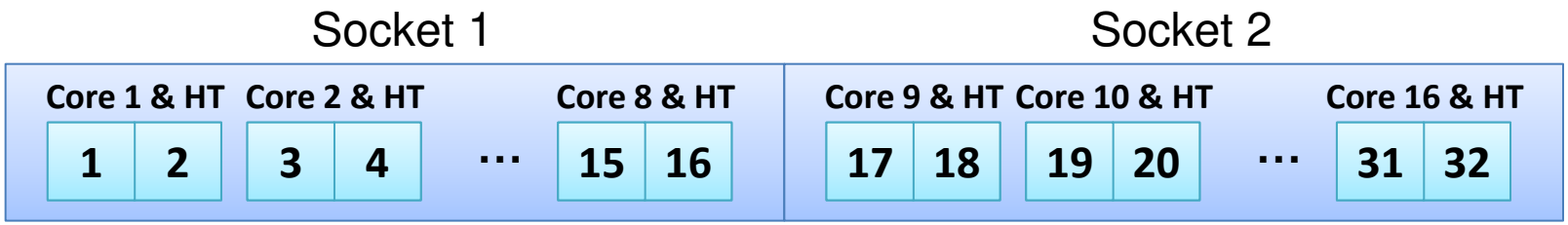


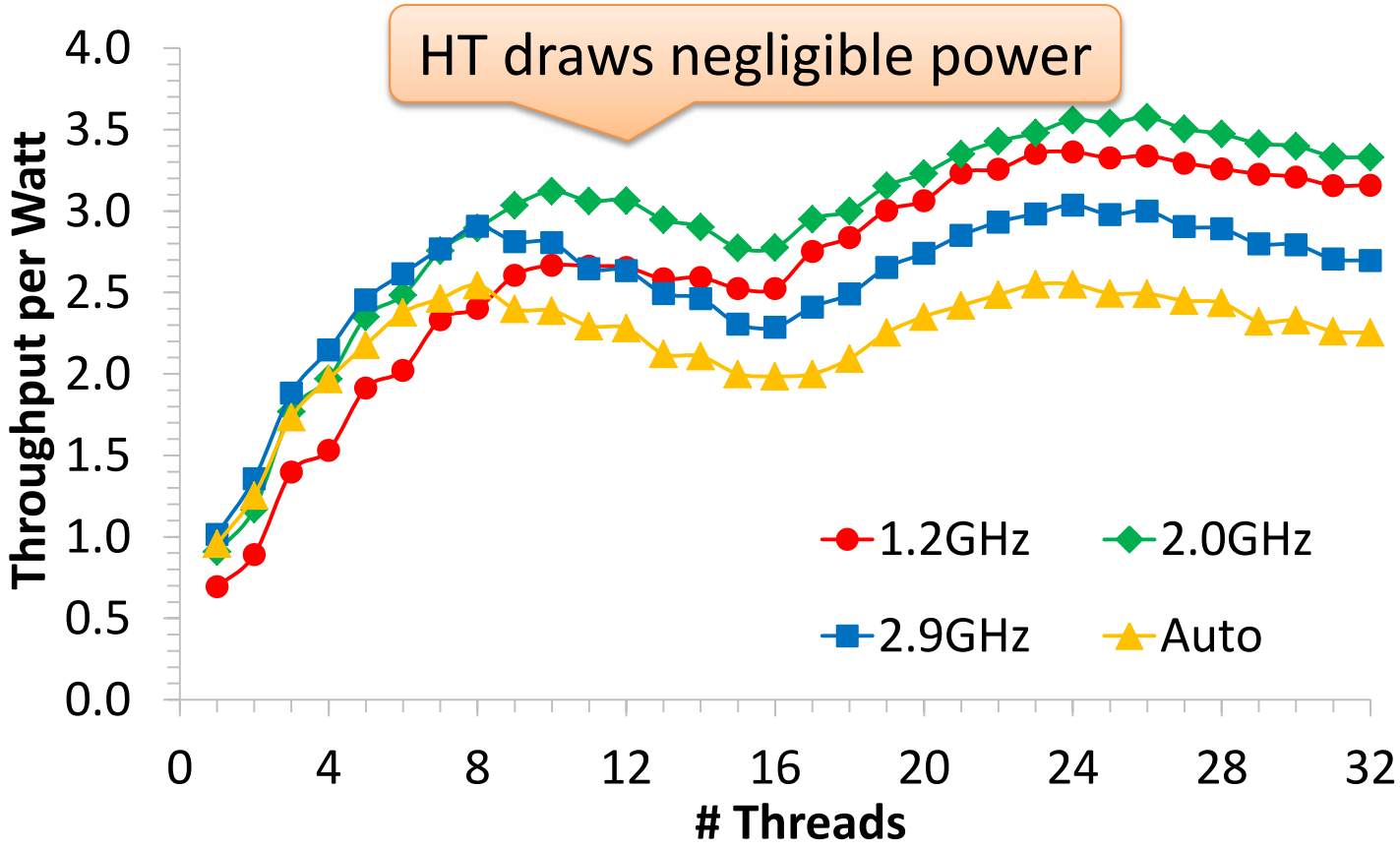


Socket-fill scheduling

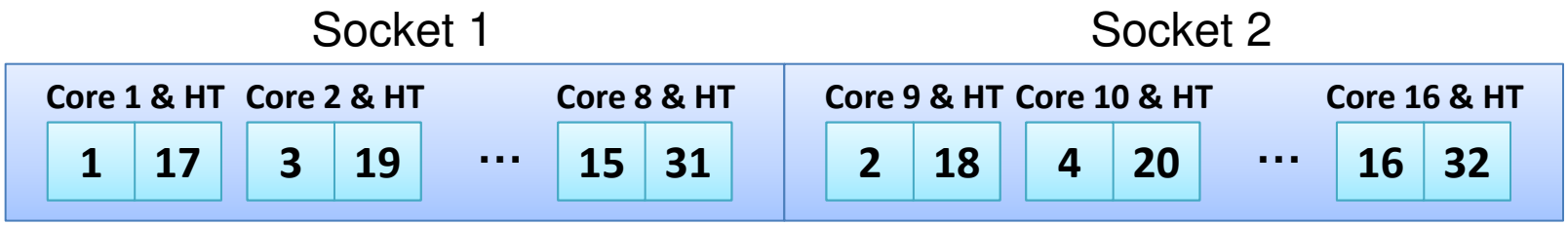


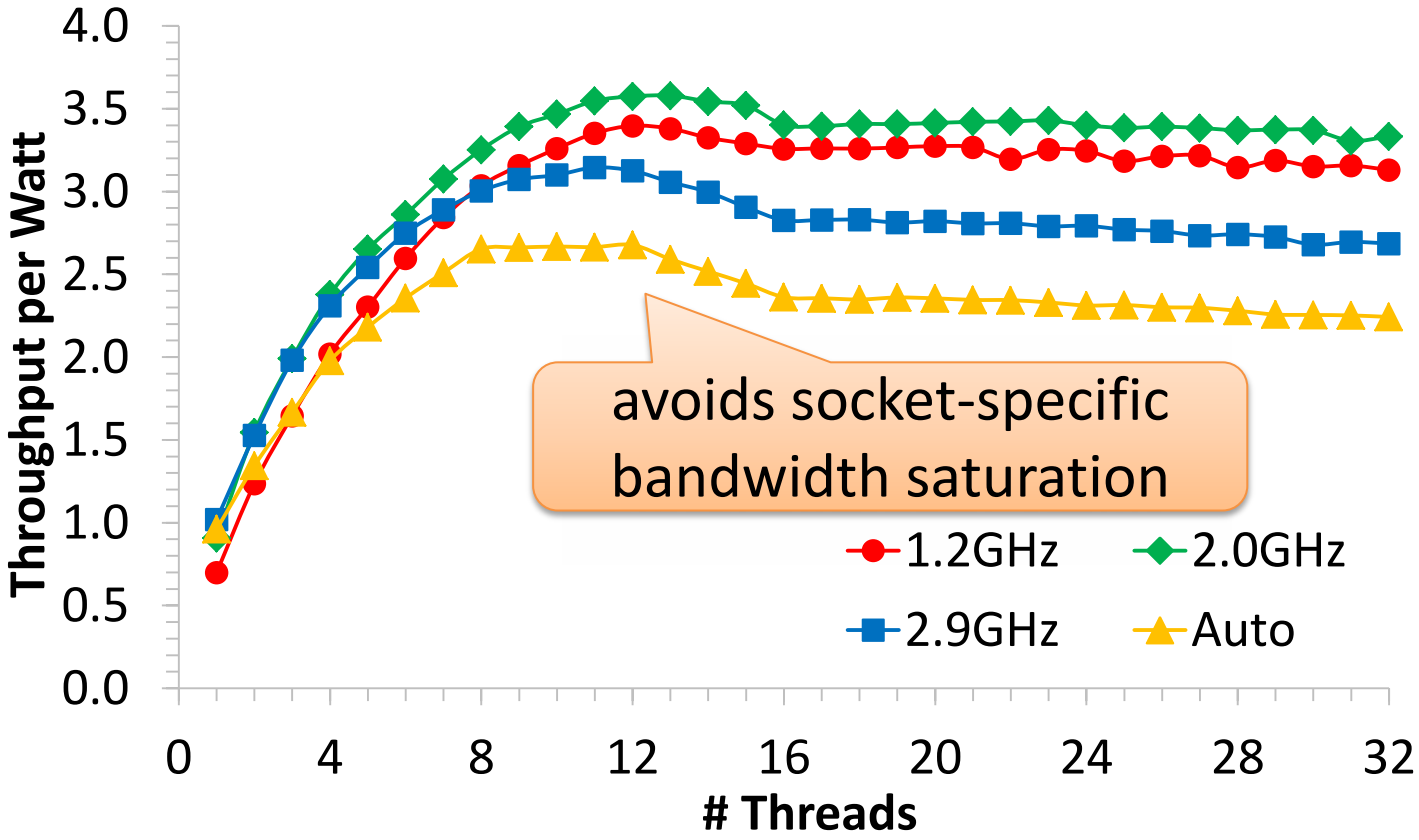
Socket-fill HT scheduling



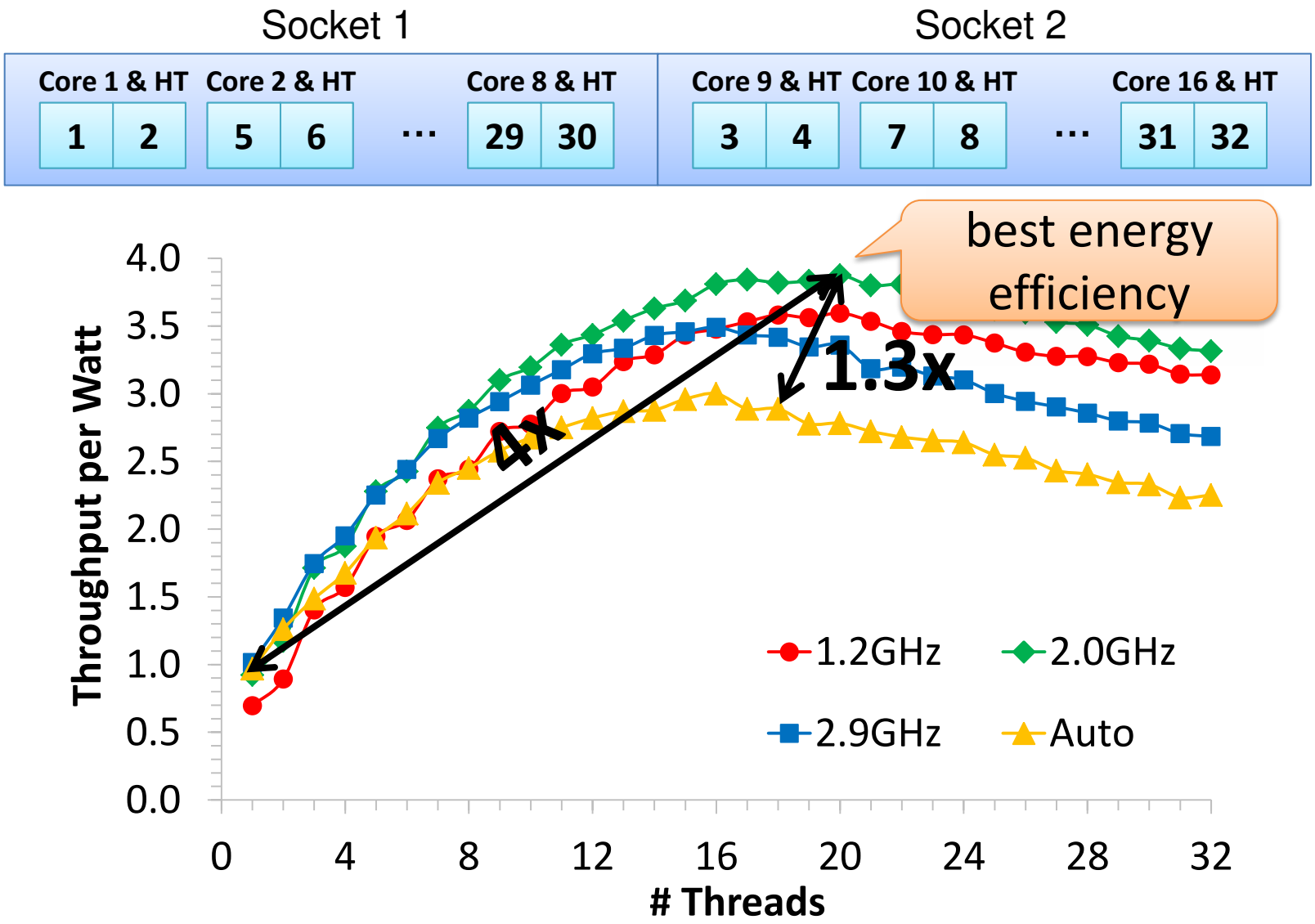


Socket-wise scheduling





Socket-wise HT scheduling



Parallel aggregation

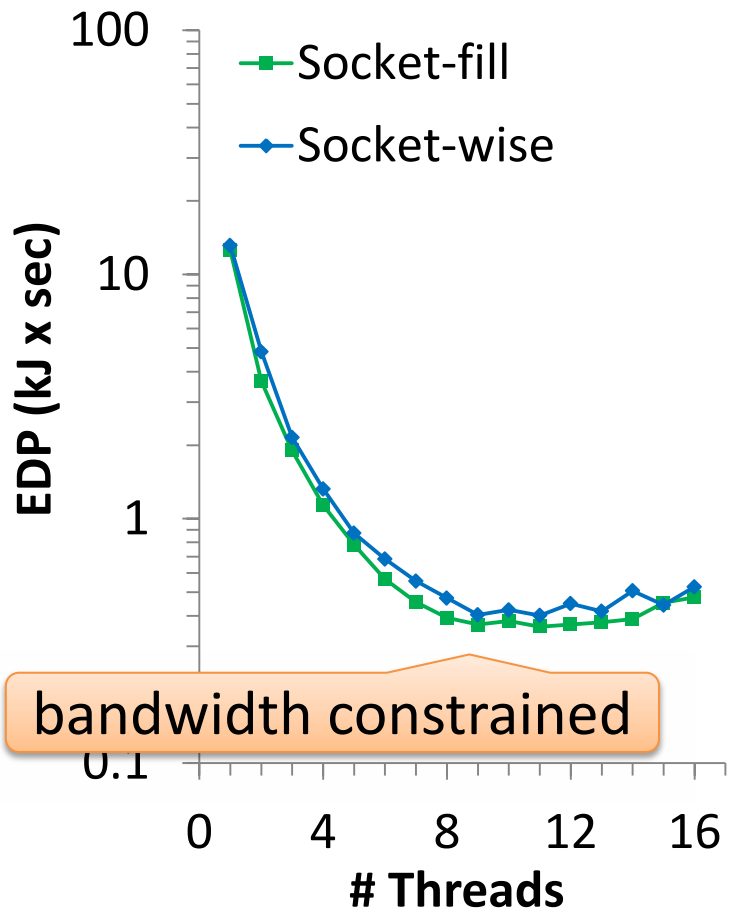
- $a = \sum\big(b(i) + c(i)\big)$, 4GB arrays
- Minimize

  $energy\ delay\ product\ (EDP) = response\ time\ (sec) * energy(\ J)$

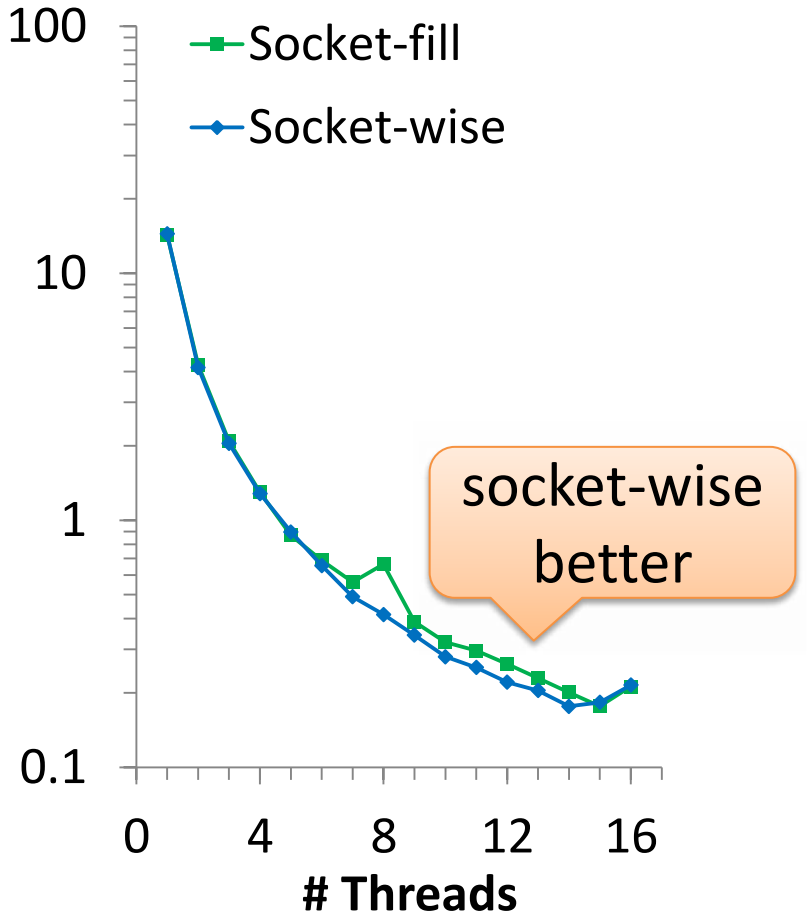
  – under different parallelism, scheduling, and memory placement
- Machine
  – Two 8-core Intel Xeon E5-2640, HT disabled, 256GB of RAM
- Memory placement
  – On first socket
  – Interleaved

Parallel aggregation

Memory on first socket



Memory interleaved



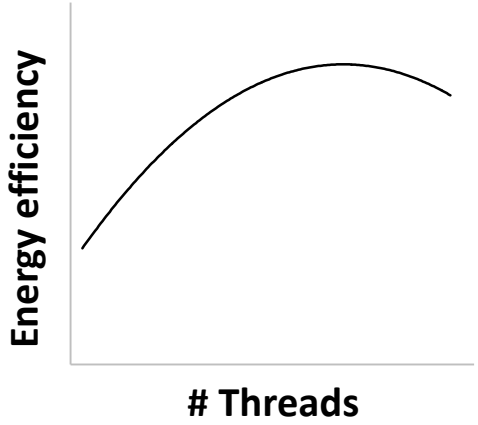
Main-memory memory-bound operations

- Intermediate frequency has best efficiency
  - Different saturation points
- Avoid memory bandwidth saturation
  - by data and thread placement
- Up to 4x energy efficiency

Fine-grained energy awareness

Calibration analysis

of operators and parameters

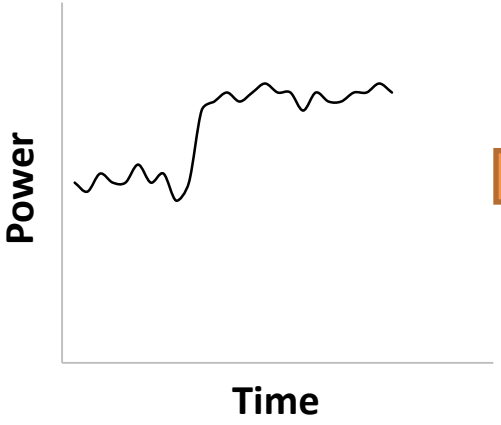


parallelism
data & thread placement
DVFS

THIS PAPER

Measurements

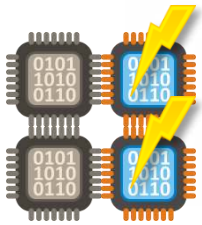
hardware counters and/or external equipment



power
CPU utilization
memory utilization

Runtime decisions

scheduling, resource allocation, power management



Thank you!

References

- [J. R. Hamilton] Internet-Scale Datacenter Economics: Where the Costs And Opportunities Lie. HPTS, 2011.
- [TPDS13] D. Li, B. R. de Supinski, M. Schulz, D. S. Nikolopoulos, and K. W. Cameron. Strategies for energy-ecient resource management of hybrid programming models. IEEE TPDS, 24(1):144-157, 2013.
- [ICDE10] Z. Xu, Y.-C. Tu, and X. Wang. Exploring power-performance tradeos in database systems. In ICDE, pages 485-496, 2010.