Thermal-Aware Scheduling in Multicore Systems Using Chaotic Attractor Predictors

Adam Wade Lewis and **Nian-Feng Tzeng**

Center for Advanced Computer Studies
University of Louisiana at Lafayette

The Center for Advanced Computer Studies

UNIVERSITY OF LOUISIANA Lafayette

OS Thread Scheduling

- Scheduling ...
  - in time: who runs next
  - in space: who runs where
- Optimization problem
  - Who runs next: least use of energy with best performance quality of service
  - Who runs where: best utilization of resources with least increase in processor and/or ambient temperature

Thread Scheduling & Power Management

Featuring...
AMD PowerNow!™ TECHNOLOGY

DVFS:

$$P = CV^2 f$$

SpeedStep

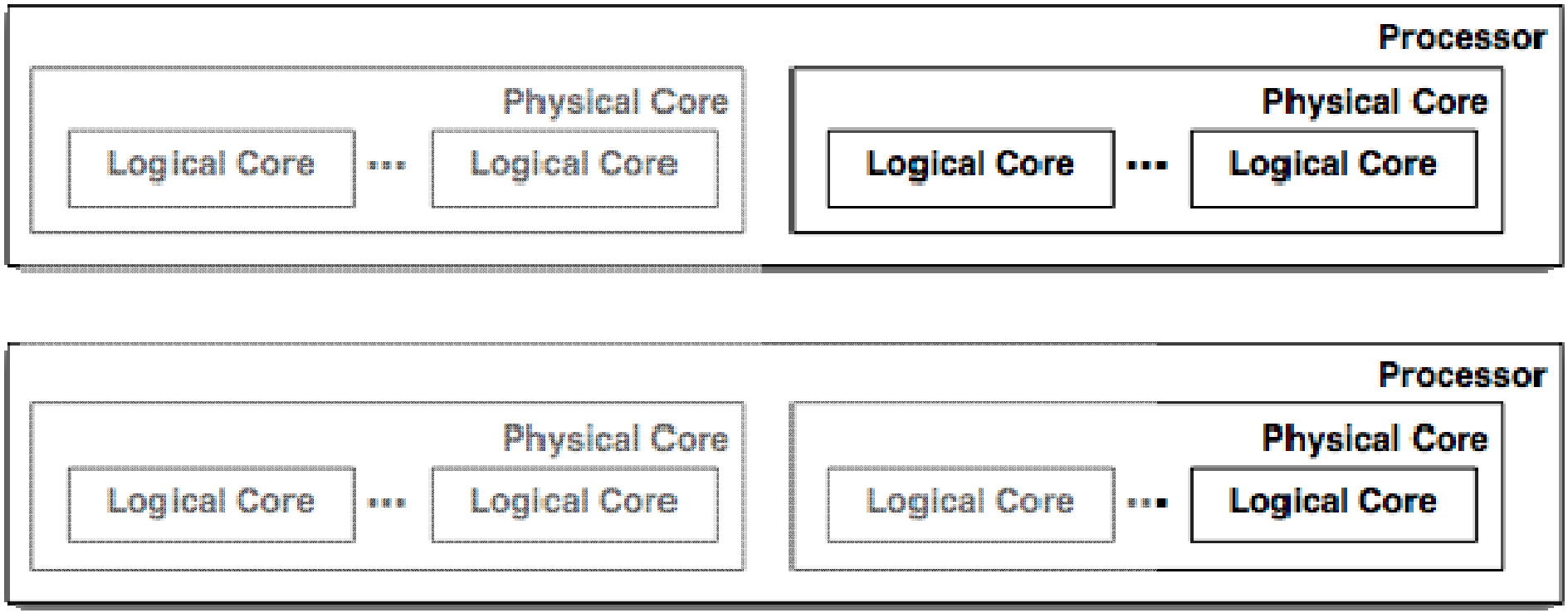
- Multi-core/Many-core
  - Cache affinity
  - Load balancing
  - To turn off the lights?

- Performance issues [LLBL 2007]
  - Lack of slack
  - High load → No gain
- Reliability issues [Bircher 2008]
  - Under-clocking & MTBF
- Proactive vs. reactive

The Center for Advanced Computer Studies

UNIVERSITY OF LOUISIANA Lafayette

Current Practice: Multicore Servers



- Find the set of processors maximizing performance on fewest possible logical cores

Prediction with Chaotic Time Series

Chaotic behavior

Core Die Temp.	Hurst Parameter (*H*)	Lyapunov Exponent (λ)
Core 0	0.99	0.051
Core 1	0.98	0.019
Core 2	0.97	0.034
Core 3	0.95	0.040

H close to 1: high self-similarity that preserves structure dynamics

λ > 0: system is chaotic

[Lewis & Tzeng 2010],

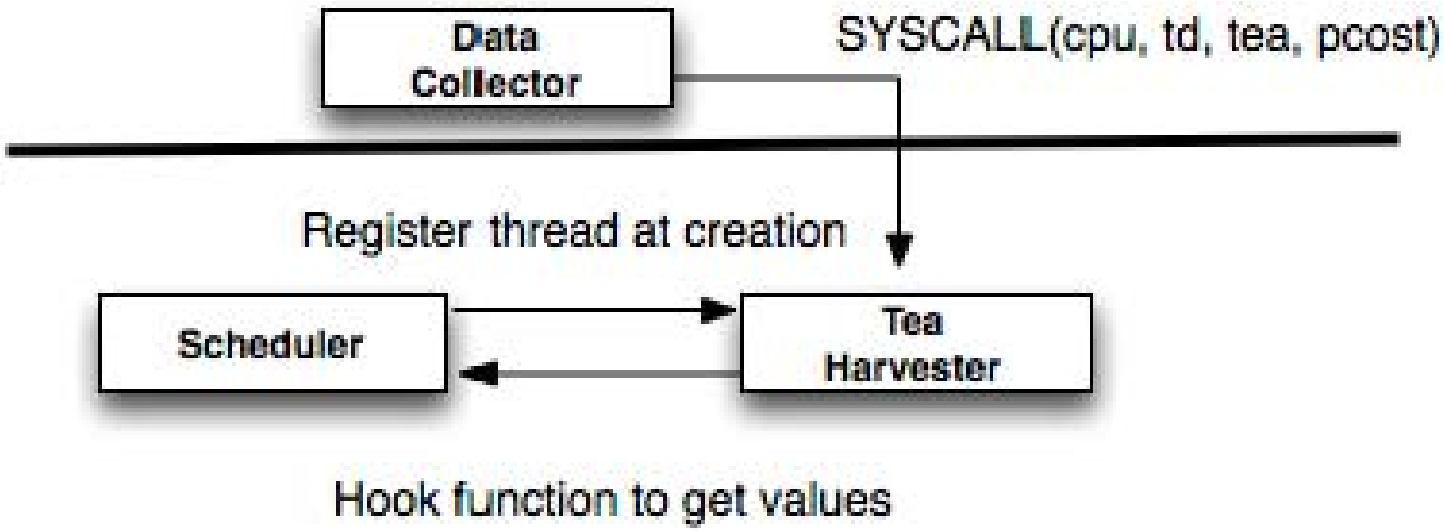
Chaotic Time Series

- Time-delay reconstructed state space
  - Uses Takens' Delay Embedding Theorem: Partitioning observations to build function that preserves behavioral and dynamic properties of the original chaotic system
- Finding nearest neighbors on attractor to those observations in a partition
- Perform least-square regression fit to find a polynomial that approximates the attractor

[**Lewis & Tzeng 2012**]

[**Lewis & Tzeng 2012**] A. Lewis, N.-F. Tzeng, and S. Ghosh, "Run-Time Energy Consumption Estimation for Server Workloads Based on Chaotic Time-Series Approximation," accepted in May 2012 for publication in *ACM Transactions on Architecture and Code Optimization* (*TACO*).

Thermal Aware Scheduler built on FreeBSD



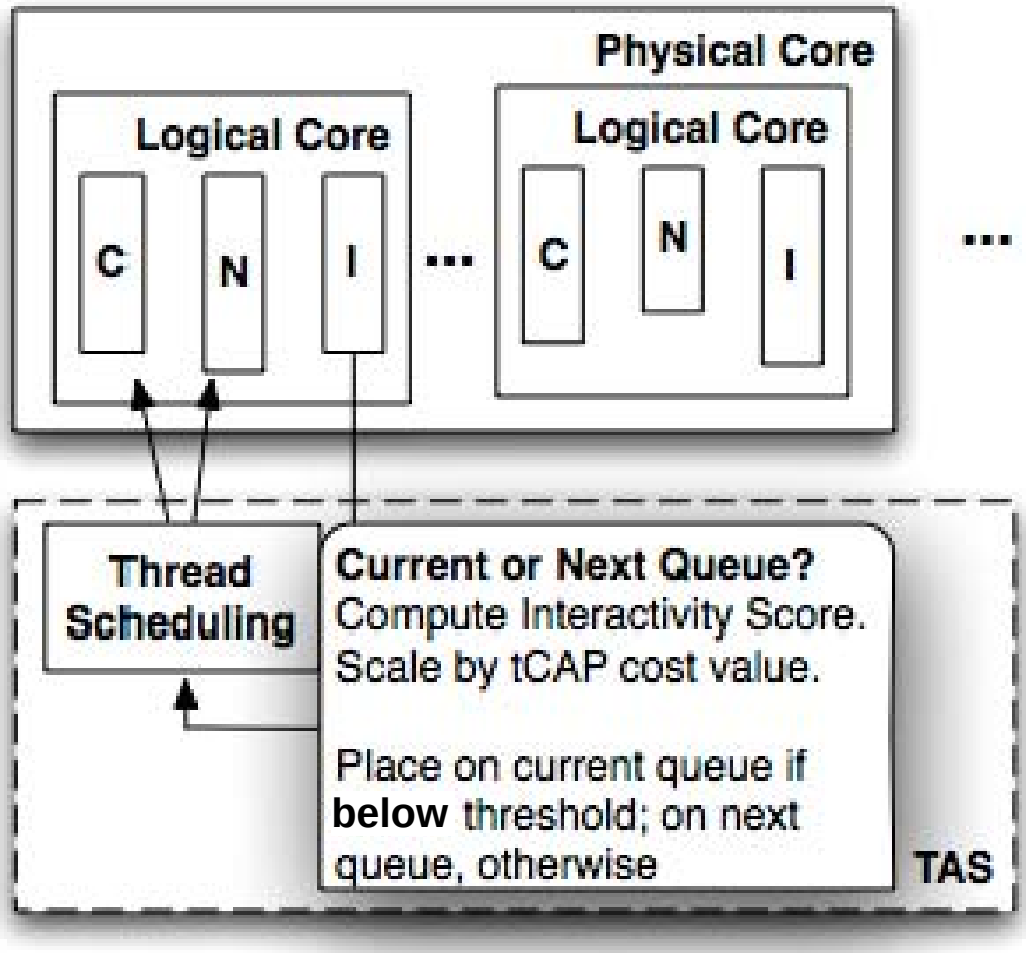
TEA: thermal equivalent of application

- Separate infrastructure for data collection and for computing *tCAP* prediction

Collected observations

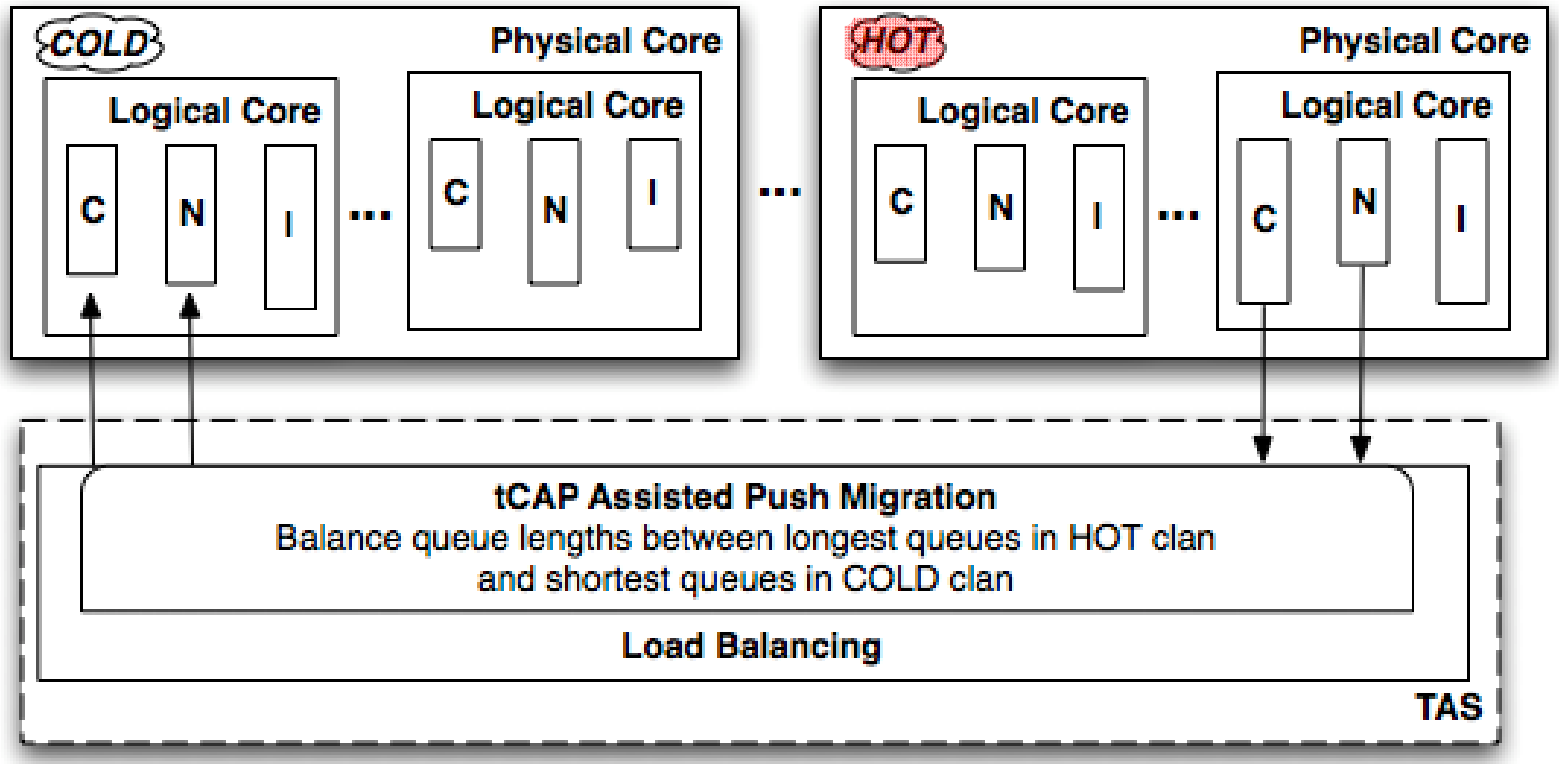
- PeCs for targeted processor
  - transactions between pairs of cores
  - transactions on the I/O handler
  - misses on last level cache
  - bytes of disk reads & writes
- OS kernel VM statistics
  - no. of instructions retired
- On-die and ambient temperature readings
- speeds of cooling fans

TAS Thread Scheduling



- Start with conventional decisions about "interactivity"
- Penalize threads with higher "thermal costs"

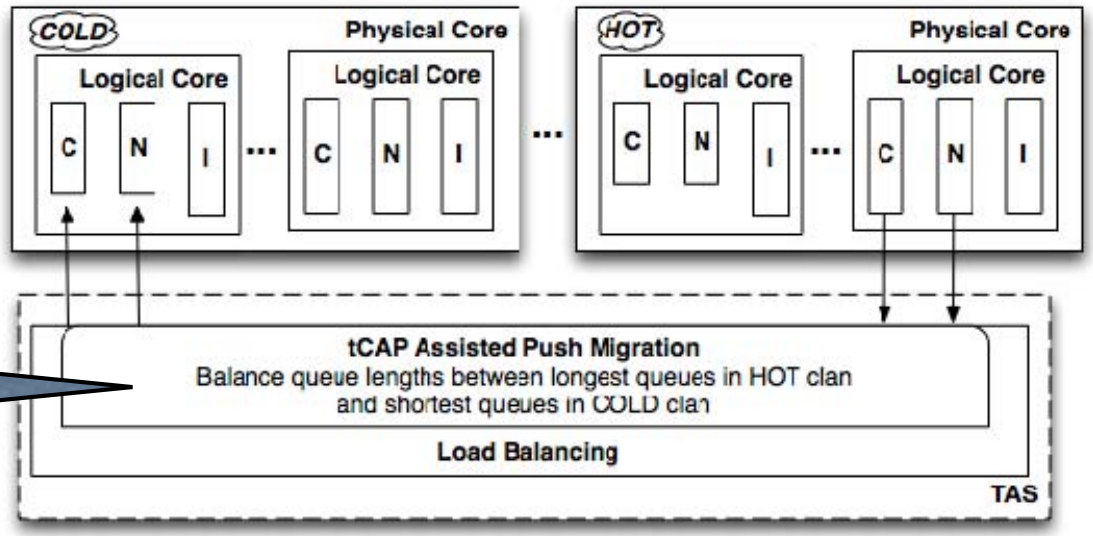
TAS Load Balancing



- Move threads from C/N queues in HOT core to those in COLD core
- Possibly lower performance for overtaxed cores to recover

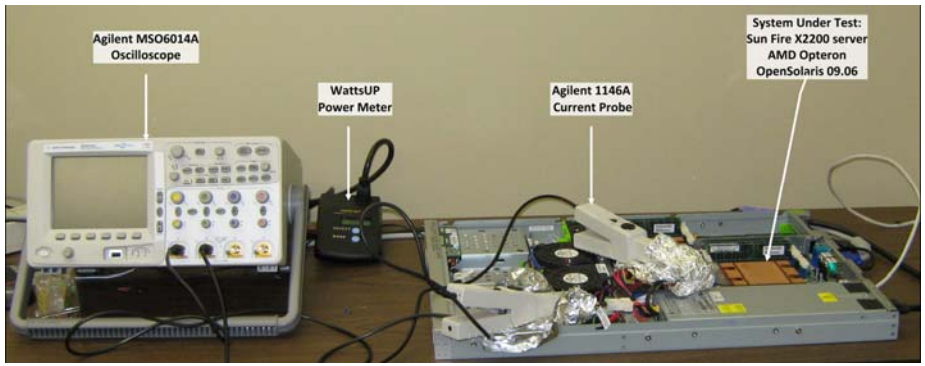
TAS Load Balancing

```
BEGIN
    Determine if the logical core
      is in the Hot, Warm, or Cold clan.
    For each thread in the run queue
      BEGIN
        Use tCAP to estimate resulting
        temperature change if the thread
        is executed.
      END
    Determine least loaded core in the
      Cold clan.
    Migrate the thread with worst impact
      on temperature to logical core in
      the Cold clan most suitable from
      the speed standpoint.
END
```



- Most suitable from speed standpoint?
  - Shared last level cache
  - Same processor only
  - Other processors

Evaluation Environment



	Dell Precision 490
CPU	Intel Xeon 5300 (Woodcrest)
CPU L2 cache	4MB
Memory	8GB, DDR2 667Mhz with ECC
Internal disk	500GB
Network	1x1000Mbps
Video	NVIDA Quadro FX3400

Training Benchmarks

Two extremes

- Idle scenario
- Most stressed scenario

Evaluation Benchmarks

PARSEC

['pär-"sek] A unit of measure

SPEC CPU2006

The Center for Advanced Computer Studies

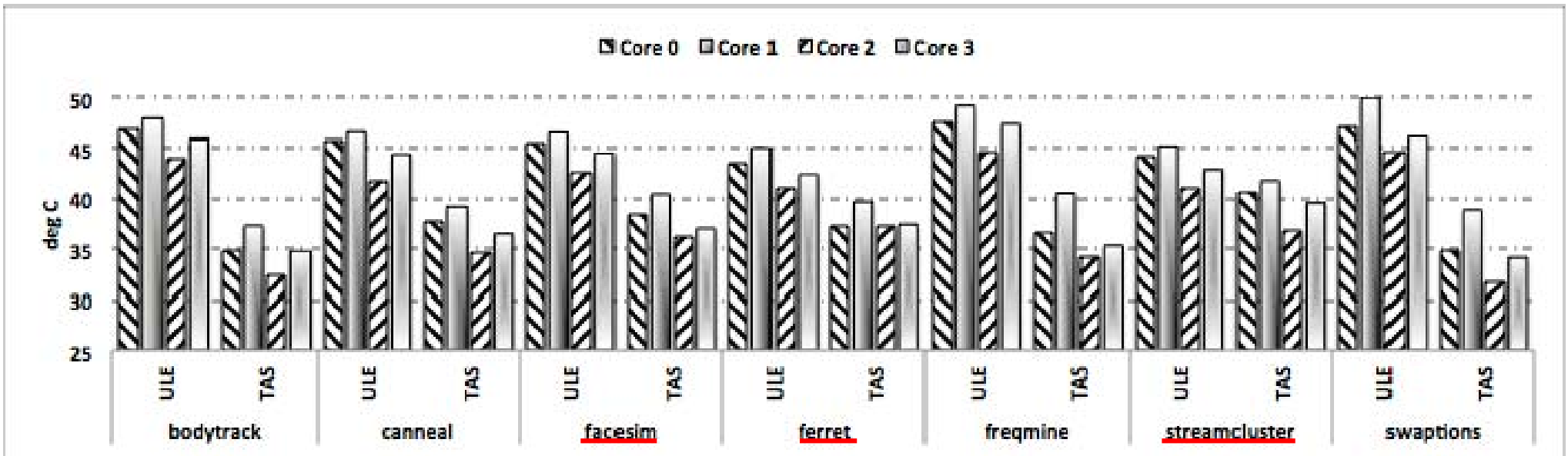
UNIVERSITY OF LOUISIANA Lafayette

PARSEC Benchmarks

PARSEC BENCHMARKS USED IN EVALUATION

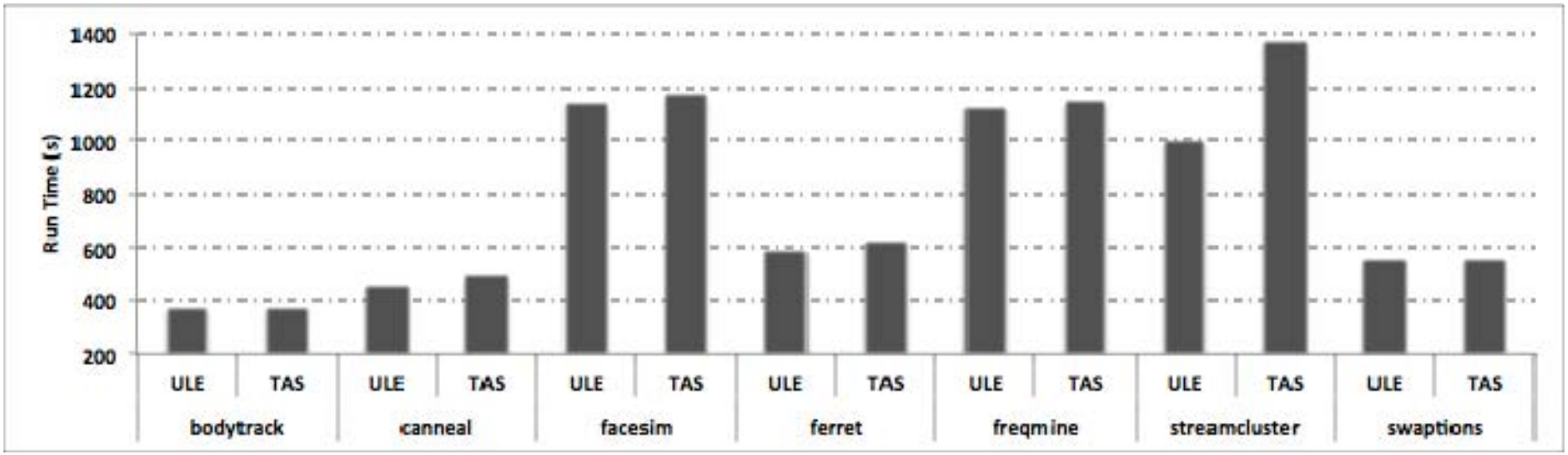
bodytrack	Computer vision image tracking application
canneal	Simulated annealing chip routing cost computation
facesim	Physical simulation of facial behavior
ferret	Content-based similarity search
freqmine	Simulate FP-growth method for Frequent Itemset Mining
streamcluster	Online clustering algorithm for data mining
swaptions	Simulated pricing of portfolio options

PARSEC Benchmarks: Temperature Reduction



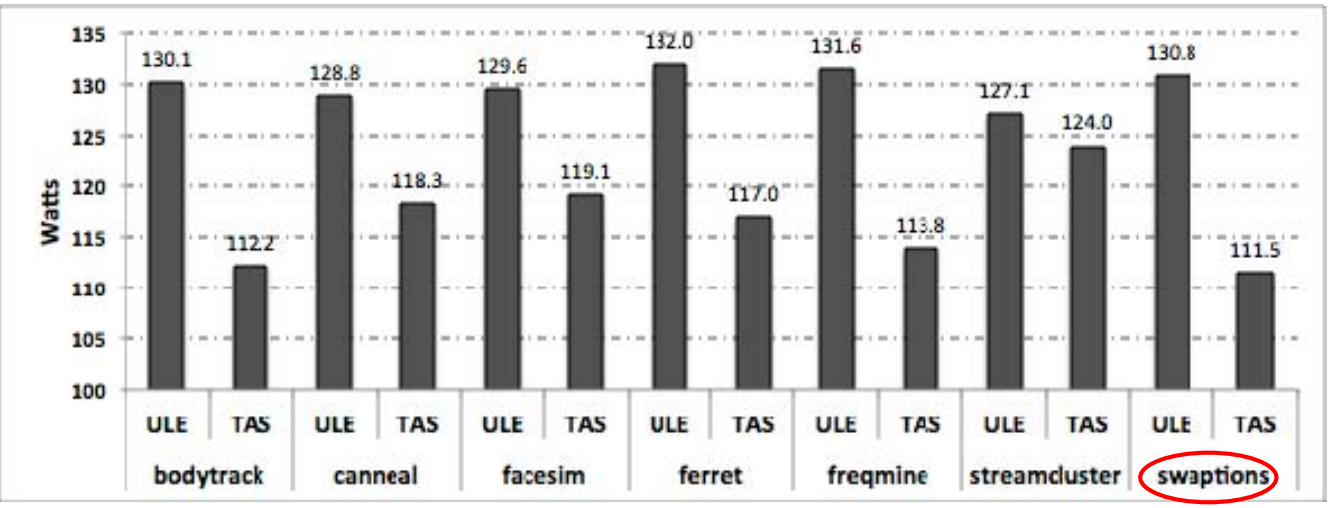
- Better results for benchmarks with smaller working sets
  - Lower cache requirements
  - Up to 12.8˚C reduction
- Other benchmarks
  - Smaller temperature benefits
  - Only 3 to 6˚C reduction vs. ULE

PARSEC Benchmarks: Execution Performance

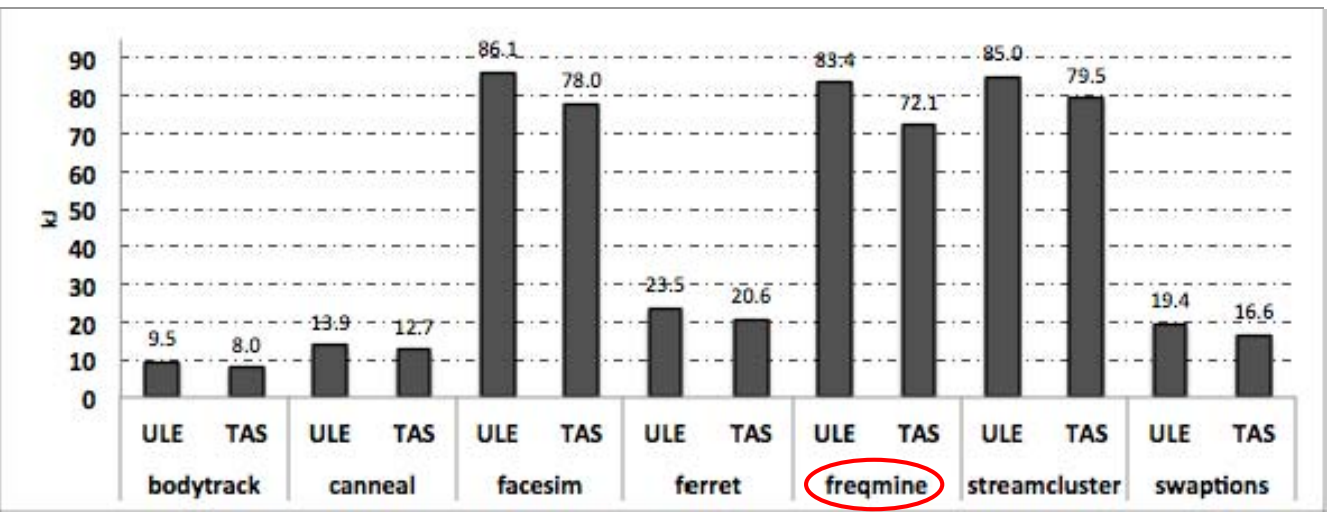


- Negligible performance degradation compared to ULE (mostly no more than 3.3%)
- Behavior of "streamcluster" benchmark
- Load balancing and cache affinity

PARSEC Benchmarks: Power and Energy



- Average power reduction by ~3 *W* to 19 *W* (14.7%)



- Energy savings by 1.2 *KJ* to 11.3 *KJ* (13.5%)

Questions?

Please Ask!