Balancing Power Consumption in Multiprocessor Systems

by A. Merkel, F. Bellosa | EuroSys06

presented by
Andreas Naderlinger

Software Systems Seminar
University of Salzburg, Austria
June 2007

Power Awareness

- Relation between power consumption and heat

- Noise level
- Costs
- Limited resources
- …

- **Performance**

Main Goal

- Boost performance in a multiprocessor system by balancing the power consumption among all CPUs.

Thermal Design

- Increasing power dissipation → Heat
- Increasing cooling costs

- 2 Alternatives:
  - Worst case thermal design
    - Over-dimensioned
    - High cooling costs
  - Moderate thermal design
    - Performance penalties (throttling)

Balanced Power Consumption

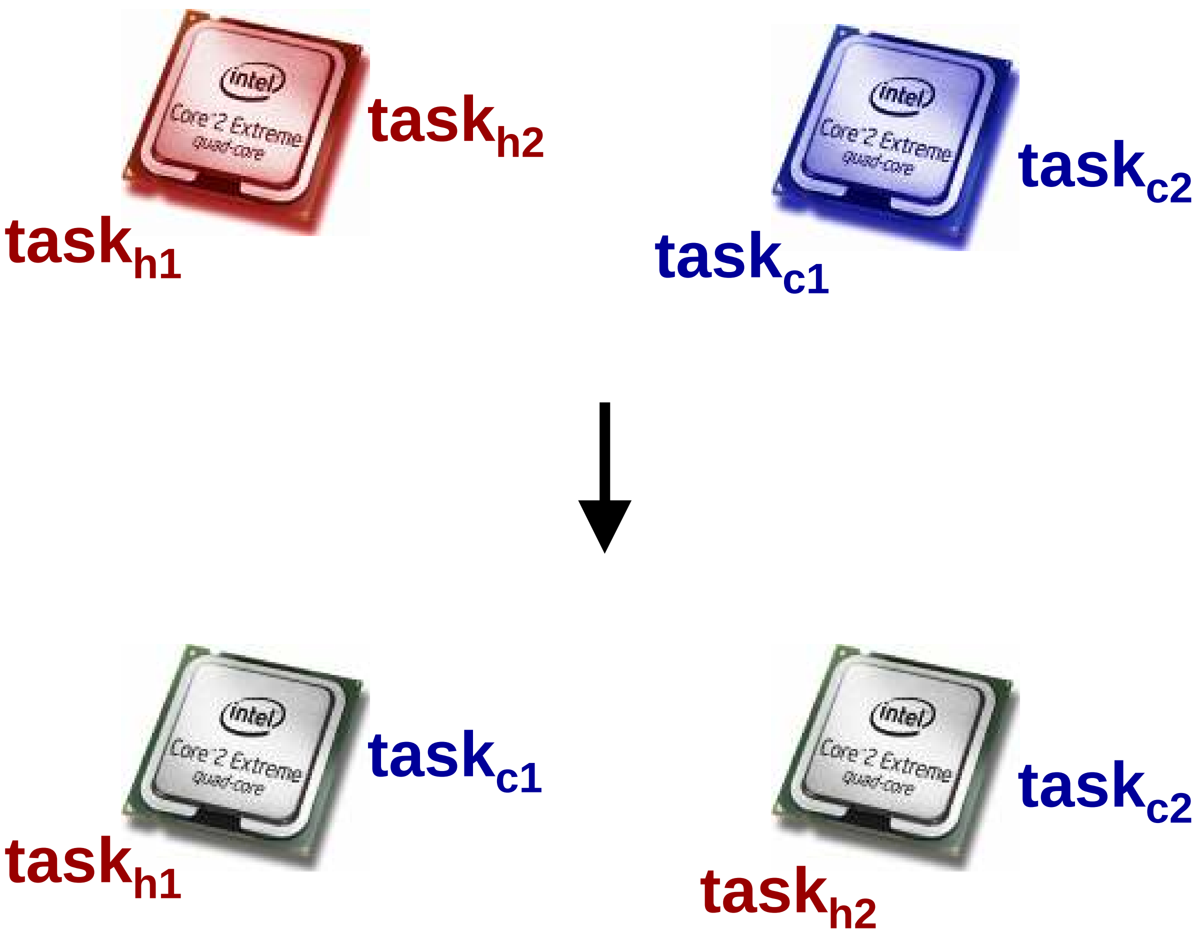
- Moderate design
- Minimizing performance penalties
- Throttling as last resort

- → Increase system’s throughput

Premise

- Power consumption depends on the kind of instruction:
- “Hot Tasks” cause high power consumption and high temperature
- “Cool Tasks” consume less power and lead to lower temperature

Basic Idea



task_{h2}
task_{h1}
task_{c2}
task_{c1}
task_{c1}
task_{h1}
task_{c2}
task_{h2}

Contribution

- Task Energy Profiles
- Energy aware scheduler

Task – Energy – Estimation

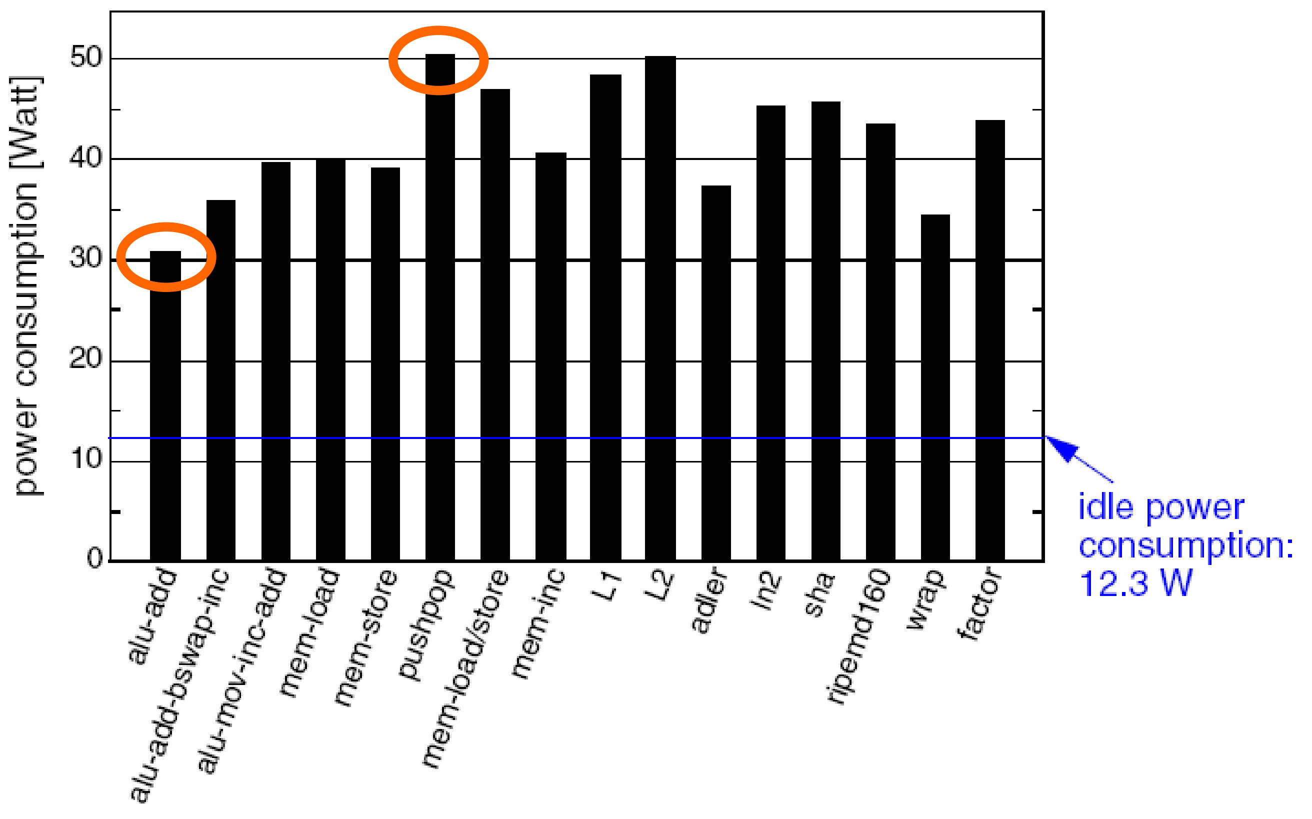
- Connection: task - temperature
- Correlation between temperature and power consumption
- Online estimation
  - depending on input data
  - depending on other tasks

Energy Estimation Approaches

- Temperature sensors
  - low temporal resolution of thermal diodes
  - significant overhead (via SMBus: ~5ms)
  - → no identification of responsible task
- Counting CPU cycles
  - wide variation
  - → inaccuracy of impact

Power consumption variance

- 100% CPU load



Event Monitoring Counters

- Intended for performance analysis
- Correspondence to processor activities

- Estimation of energy consumption

$$E = \sum_{i=1}^{n} a_i \cdot c_i$$

- Error: < 10% < 1℃
- No suitable counters for floating point applications

event	weight [nJ]
time stamp counter	6.17
unhalted cycles	7.12
μop queue writes	4.75
mispred branches	340.46
mem retired	1.73
ld miss 1L retired	13.55

Energy Profiles

- Prediction on energy consumption
- Per Tasks: → timeslice granularity

- Last value : Good Guess
- Rare significant changes

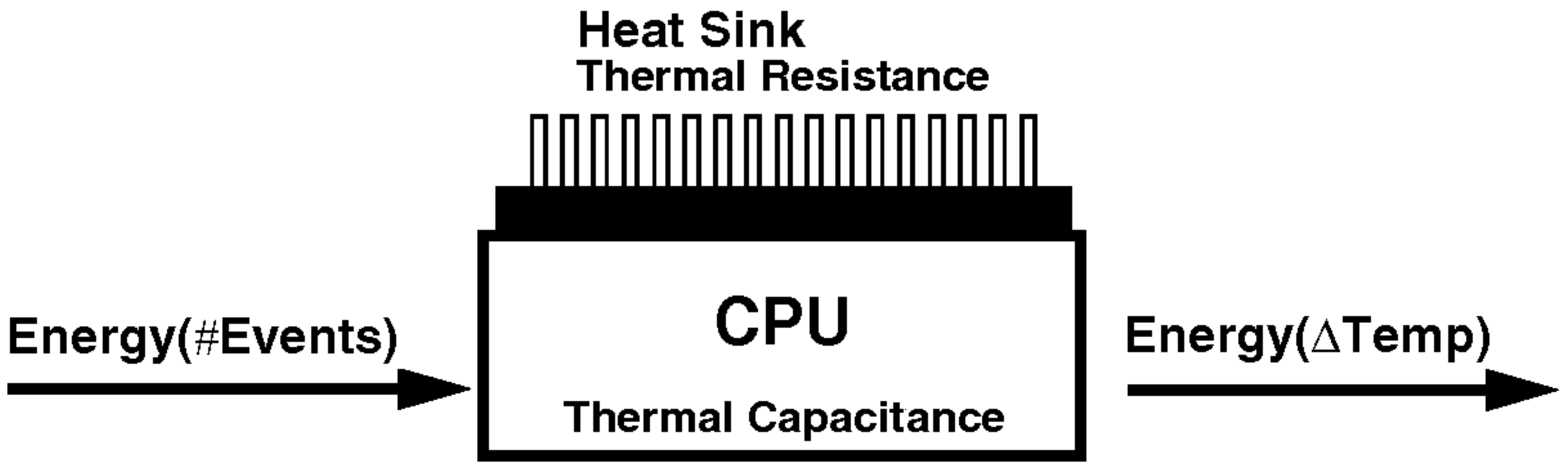
program	maximum	average
bash	19.0%	2.05%
bzip2	88.8%	5.45%
grep	84.3%	1.06%
sshd	18.3%	1.38%
openssl	63.2%	2.48%

Change in power consumption

- Exponential average

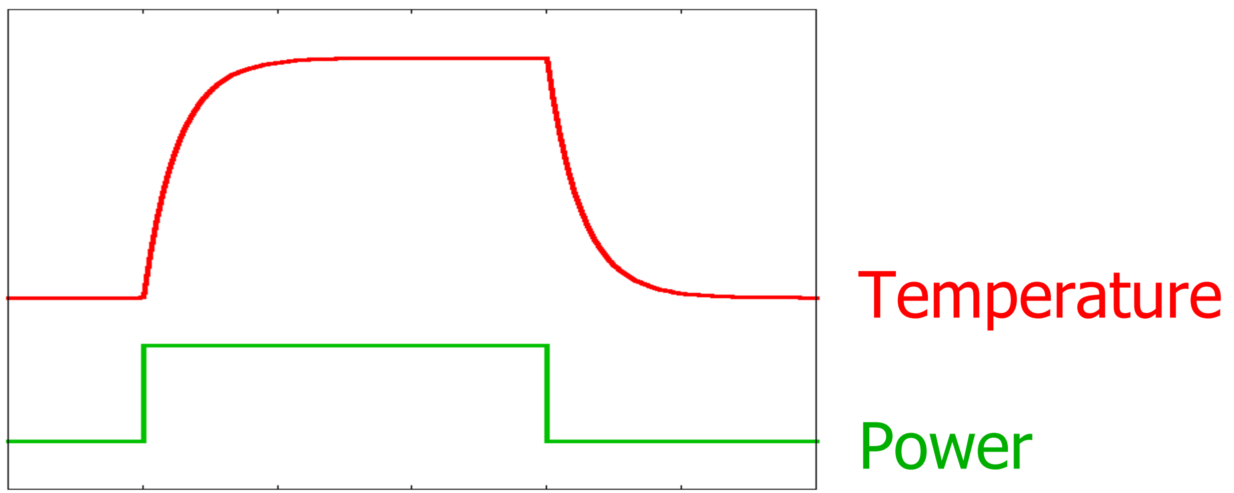
Thermal Model

- Power consumption ~ Temperature



Thermal Model

- Power consumption ~ Temperature
  - Strong relation
  - Very different characteristics
  - Both values are considered



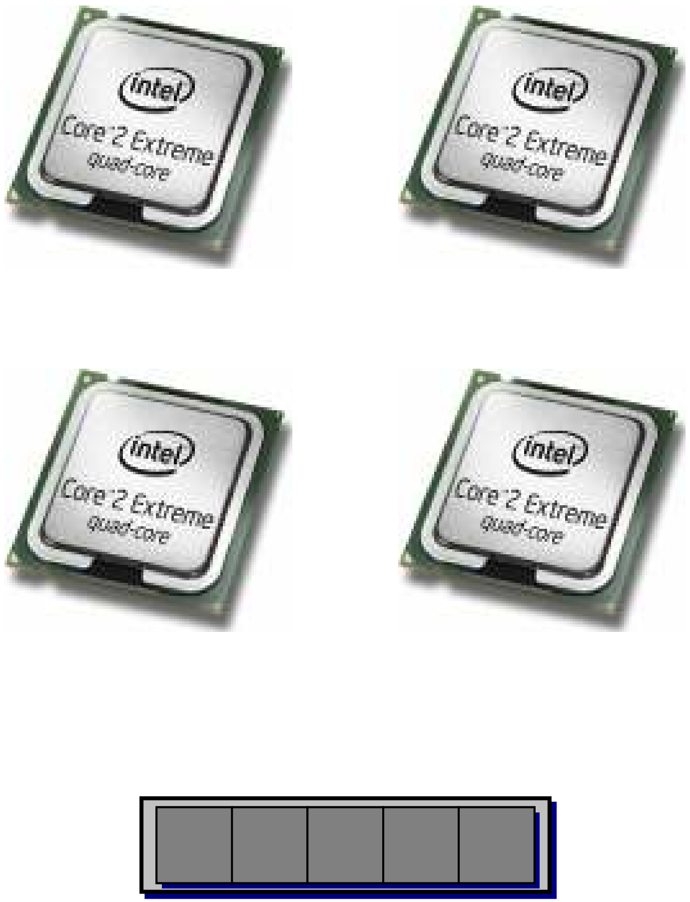
Energy Aware Scheduling

- Goal: avoid throttling
- Assigning hotter tasks to cooler CPUs and vice versa

- Limit CPU changes (cache affinity)

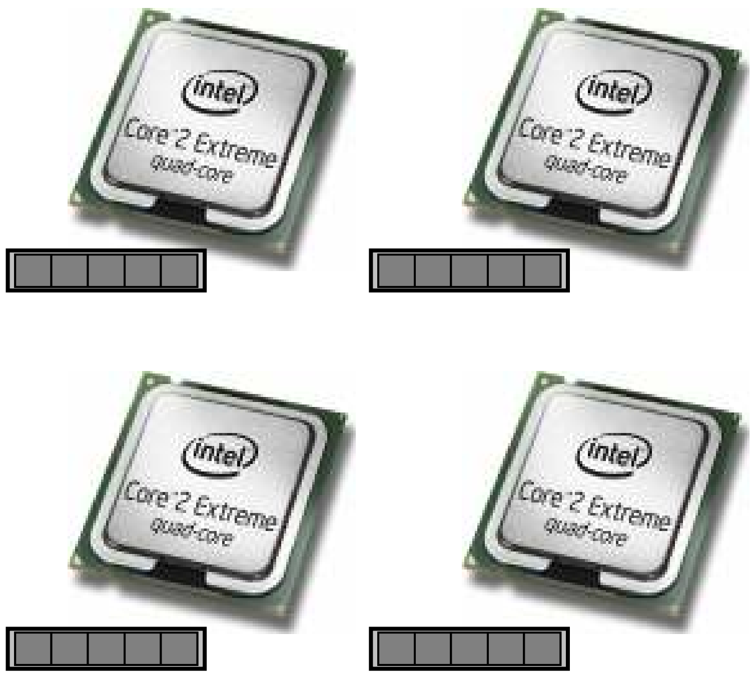
- Optimal scheduling requires topology knowledge (node affinity in NUMA systems)

Energy Aware Scheduling

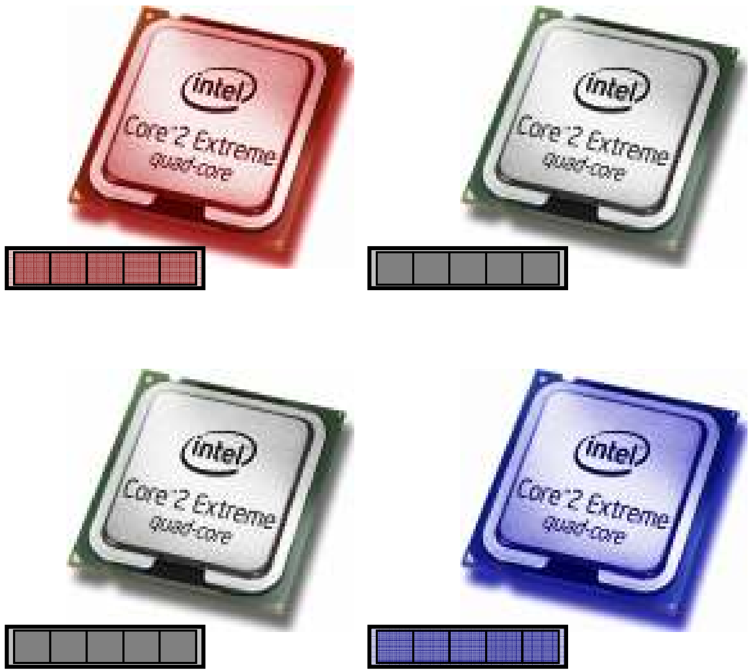


runqueue

Energy Aware Scheduling

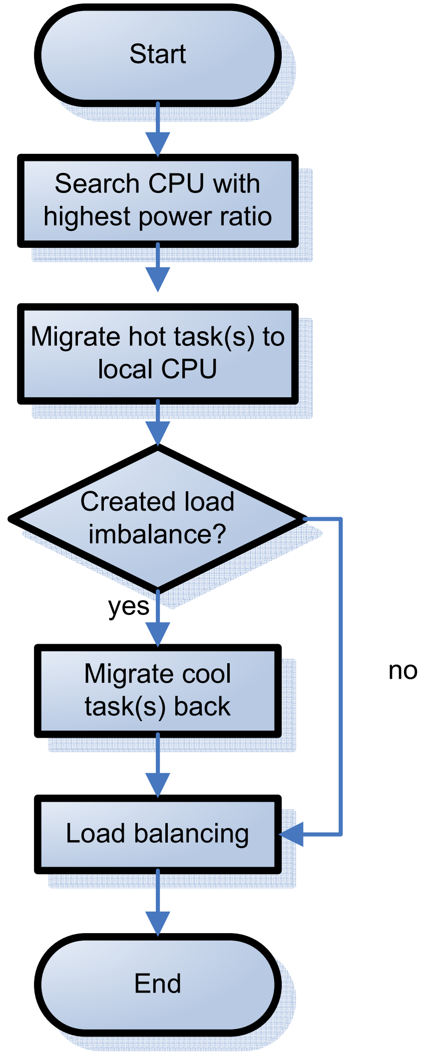


Energy Aware Scheduling

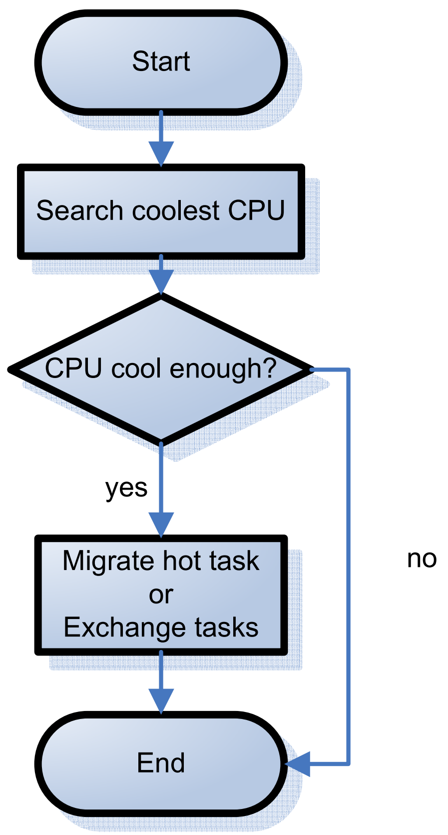


Energy Balancing

- ~ Linux load balancing
  - Distributed algorithm
  - Unidirectional migration
- Integrated into Linux 2.6.10
- Consistency between energy- and load balancing
- Multiple hierarchies



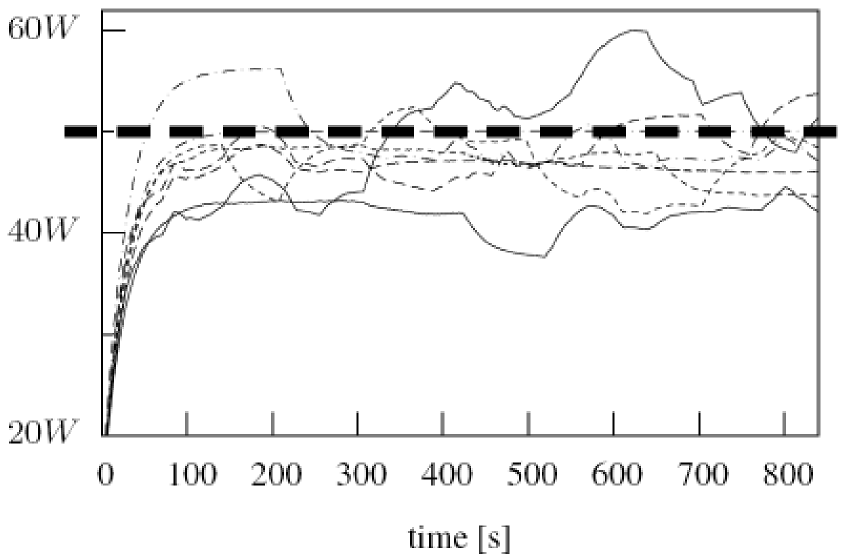
Hot Task Migration



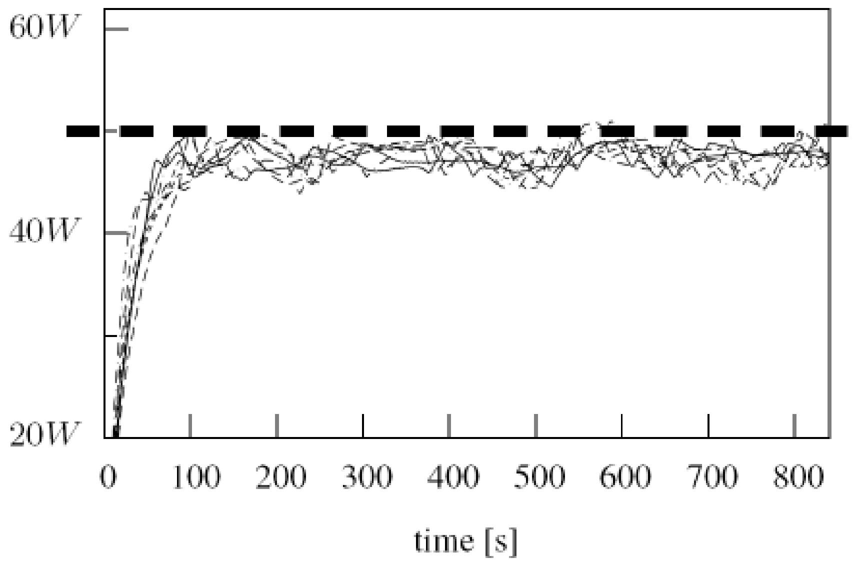
Evaluation: Energy Balancing

- IBM xSeries 445: 8 Pentium4 Xeon

disabled



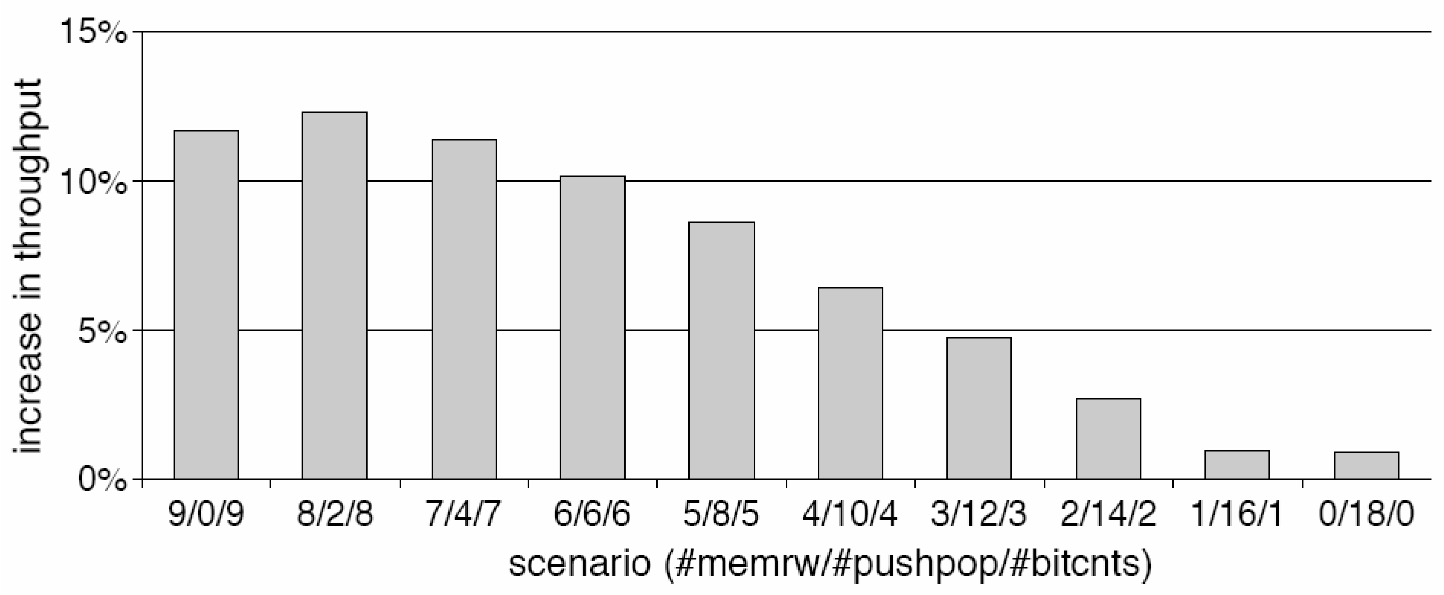
enabled



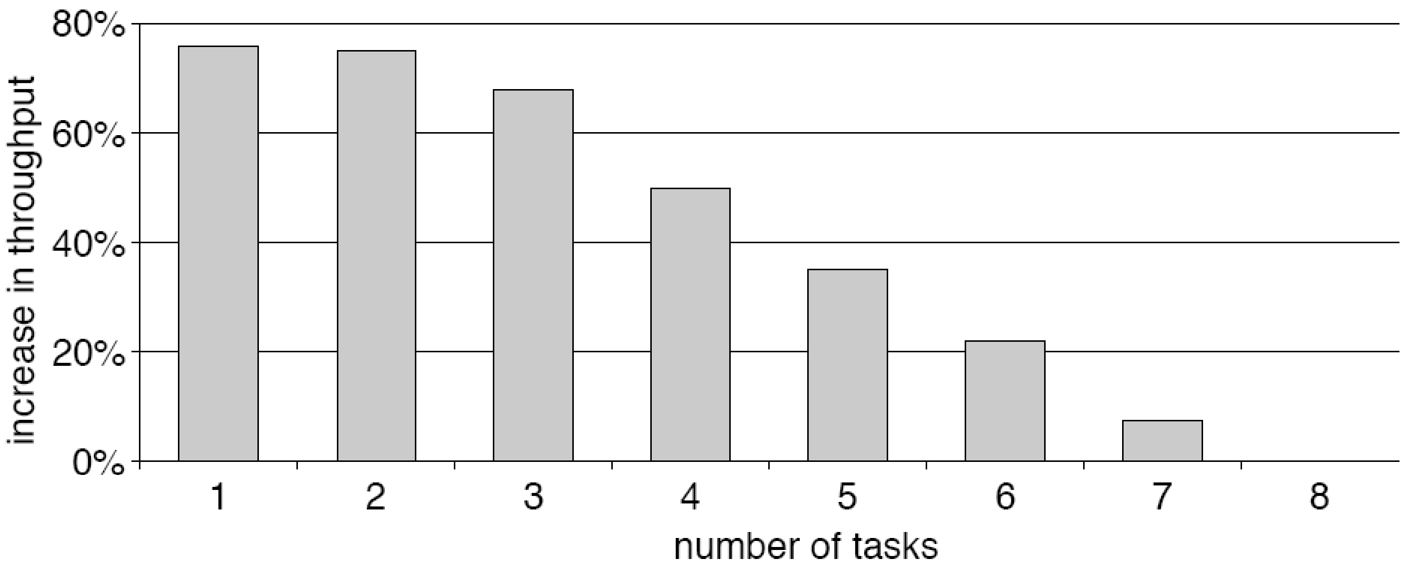
- ~ 5% less CPU throttling
- ~ 4.7% increased throughput

Evaluation: Energy Balancing

- Dependence on energy characteristics diversity



Evaluation: Hot Task Migration



Conclusion

- Event monitoring counters
- Task energy profiles for characterizing individual tasks by their power consumption
- Energy-aware scheduling to avoid thermal imbalances and reduce throttling
- ~ 5% higher throughput, but depends on workload and task characteristics

Thank you !