Event-Driven Thermal Management in SMP Systems

Andreas Merkel ● Frank Bellosa ● Andreas Weissel

System Architecture Group
University of Karlsruhe

Second Workshop on
Temperature-Aware Computer
Systems

June 5, 2005

State of the Art in Thermal Design

- Worst case thermal design
  - Overprovisioning
  - ➔ High cost
- More moderate thermal design power
  - Throttling to handle “hot” tasks
  - ➔ Performance penalties

Thermal Imbalances in SMP Systems

- Difference in power consumption of tasks
- Hot and cold processors
- Our Approach:
  - Migrate hot tasks away from a hot processor
  - Combine hot tasks with cool tasks on a processor
  - ➔ Reduce need for throttling

Contributions

- Migrate hot tasks away from a hot processor
- Combine hot tasks with cool tasks on a processor

- Prerequisites:
  - Characterization of tasks
    - → Task Energy Profiles
  - Policy for assigning tasks to CPUs
    - → Energy-Aware Scheduling

Contributions

- Migrate hot tasks away from a hot processor
- Combine hot tasks with cool tasks on a processor

- Prerequisites:
  - Characterization of tasks
    - → Task Energy Profiles
  - Policy for assigning tasks to CPUs
    - → Energy-Aware Scheduling

Contributions

- Migrate hot tasks away from a hot processor
- Combine hot tasks with cool tasks on a processor

- Prerequisites:
  - Characterization of tasks
    - → Task Energy Profiles
  - Policy for assigning tasks to CPUs
    - → Energy-Aware Scheduling

Outline

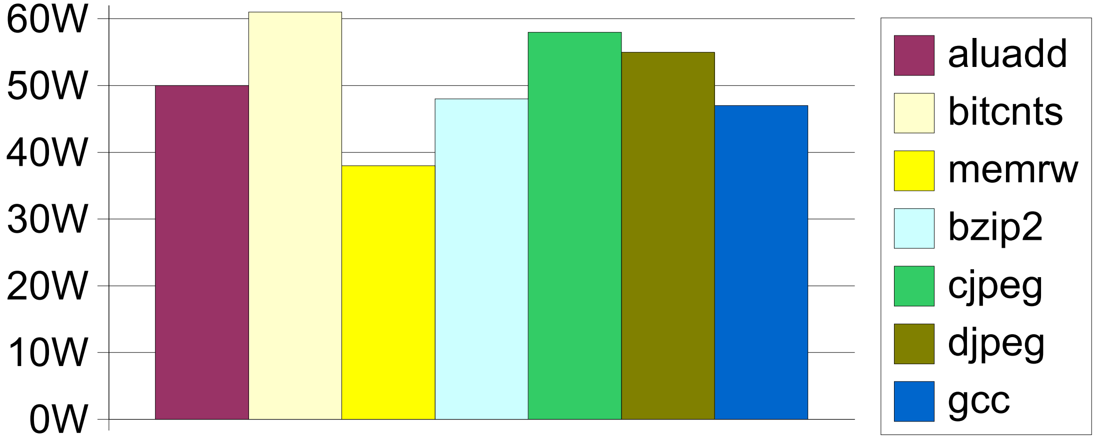
- Task Energy Profiles
- Energy-Aware Scheduling
  - Energy Balancing
  - Hot Task Migration
- Evaluation
- Conclusion

Task Energy Profiles

Characterizing Tasks

- Thermal diode
  - High thermal capacitance of chip and heat sink
  - Short scheduling intervals
  - → CPU temperature: mix of multiple tasks' characteristics
- Power consumption
  - 37W to 61W on Pentium 4 Xeon (2.2 GHz) for compute-intensive tasks
  - → Characterize tasks by their individual power consumption

Characterizing Tasks



- Power consumption
  - 37W to 61W on Pentium 4 Xeon (2.2 GHz) for compute-intensive tasks
  - ➔ Characterize tasks by their individual power consumption

Task Energy Profiles

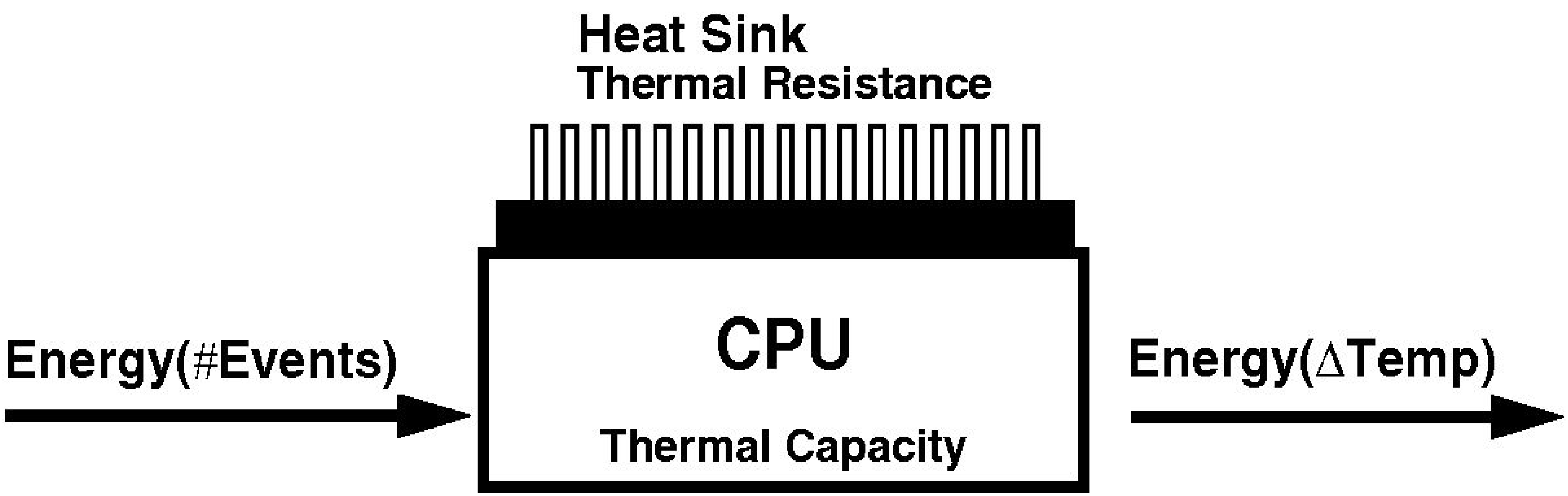
- Definition: Energy consumption for one timeslice
- Behavior of tasks depends on input data
  - → Online energy estimation required
- Tasks show phases of constant power consumption
  - → Exponential average of energy consumed during past timeslices
- Requirement:
  - Determine the amount of energy the CPU consumes during one timeslice

Energy Estimation using Event Monitoring Counters

- Estimate energy using event monitoring counters
- Count processor internal events
- Assign amount of energy to each event
- Calculate linear combination of counter values:
  - $\text{Energy} = \sum_i \#\text{event}_i \cdot \text{weight}_i$
- Error
  - < 10% for real-world integer applications
  - Higher for multimedia and floating point applications

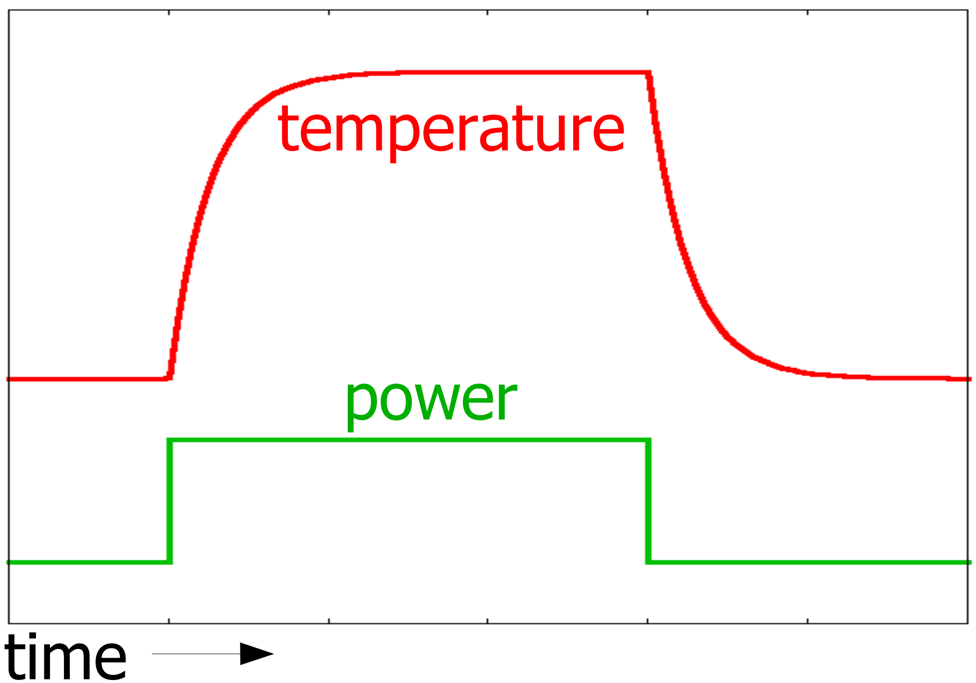
Thermal Model

- What is the processor temperature after a task with power consumption P ran for one timeslice?
- Thermal model of processor and heat sink
- Models temperature with exponential function



Thermal Model

- What is the processor temperature after a task with power consumption P ran for one timeslice?
- Thermal model of processor and heat sink
- Models temperature with exponential function



Energy-Aware Scheduling

Energy Aware Scheduling

- Objectives:
  - Minimize the need for throttling processors
  - Avoid unnecessary migrations (cache affinity)
- Best policy depends on number of tasks per runqueue
- More than one task
  - Balance power consumption between CPUs
- One task
  - Migrate task before CPU overheats

Linear Energy Balancing

- Goal: Balance CPU temperatures
- Intuitive approach: Balance CPU power
  - Equalize the average of task energy profiles for all runqueues
  - → Calculated power consumption rate
  - Mirrors future energy consumption

Linear Energy Balancing

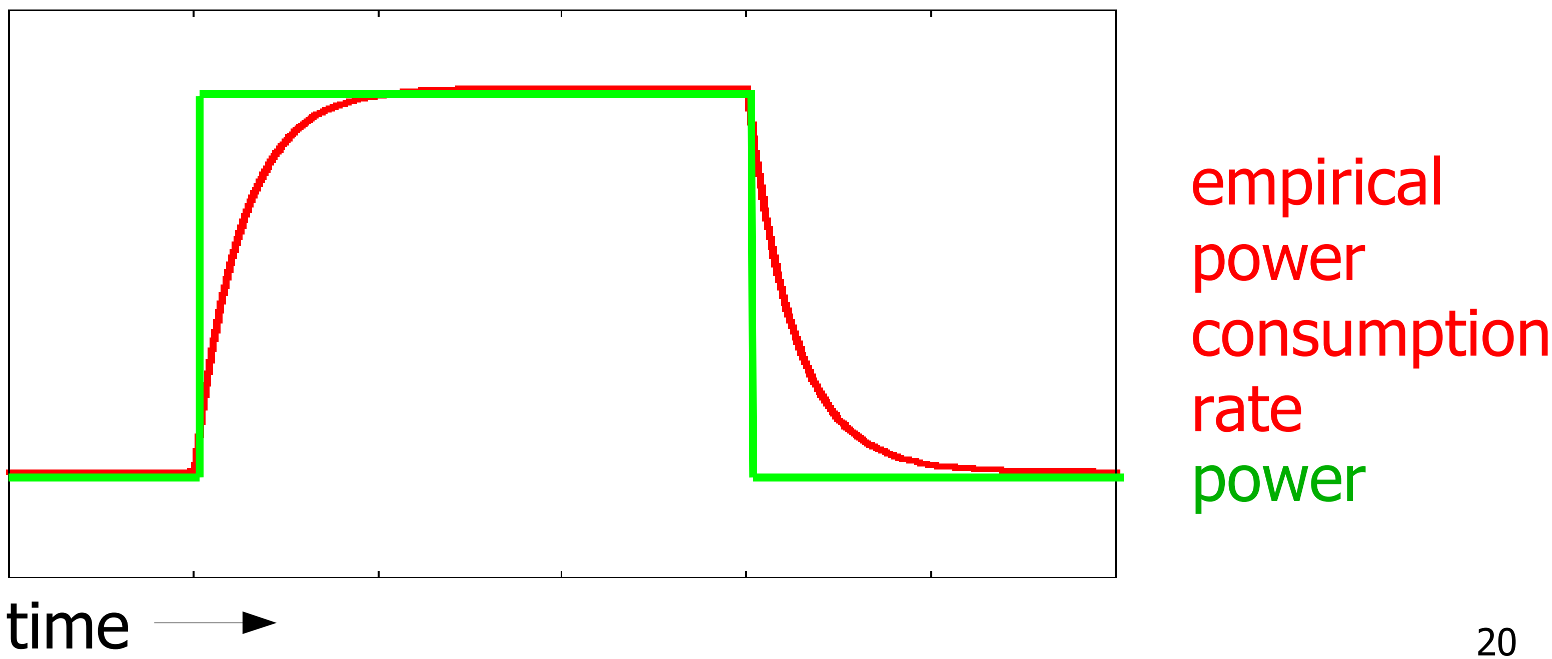
- Goal: Balance CPU temperatures
- Intuitive approach: Balance CPU power
  - Equalize the average of task energy profiles for all runqueues
  - → Calculated power consumption rate
  - Mirrors future energy consumption
- Problems:
  - Does not consider tasks that are blocked or have terminated
  - Heat produced by those tasks is still stored in the chip
  - Need to distinguish between hot and cool CPUs

Exponential Energy Balancing

- Fit averaging function to thermal model
- Exponential average of CPU's power consumption
  - ➔ Empirical power consumption rate
  - Mirrors past energy consumption ➔ temperature
  - Calibrate parameters to thermal model

Exponential Energy Balancing

- Fit averaging function to thermal model
- Exponential average of CPU's power consumption
  - ➔ Empirical power consumption rate
  - Mirrors past energy consumption ➔ temperature
  - Calibrate parameters to thermal model

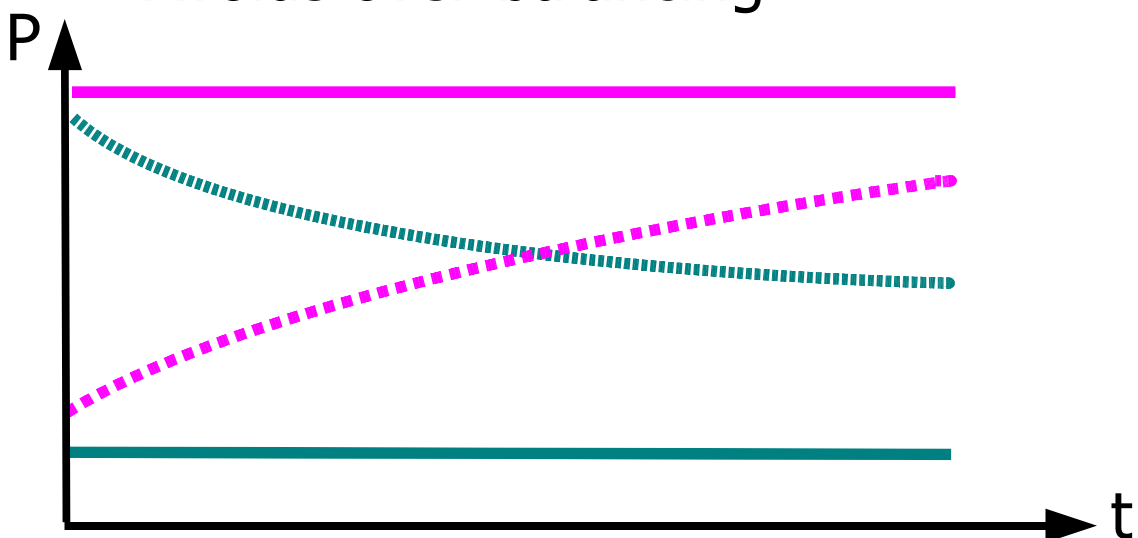


Energy Balancing

- Use both rates for energy balancing
- Migrate a hot task from CPU A to CPU B if both rates for A are greater than both rates for B
  - ➔ Hysteresis
  - ➔ Avoids ping-pong effects
  - ➔ Avoids over-balancing

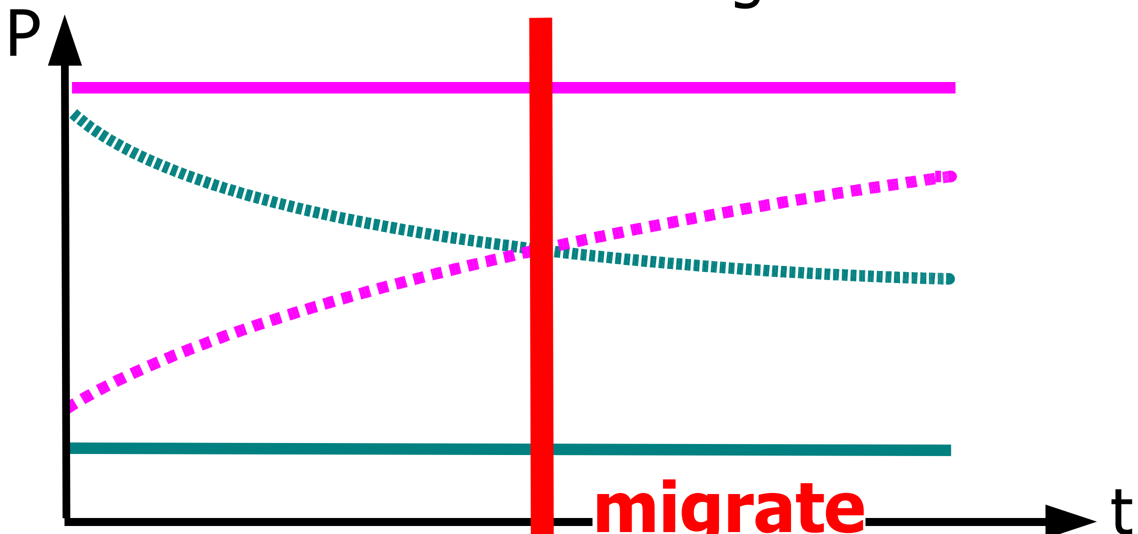
Energy Balancing

- Use both rates for energy balancing
- Migrate a hot task from CPU A to CPU B if both rates for A are greater than both rates for B
  - → Hysteresis
  - → Avoids ping-pong effects
  - → Avoids over-balancing



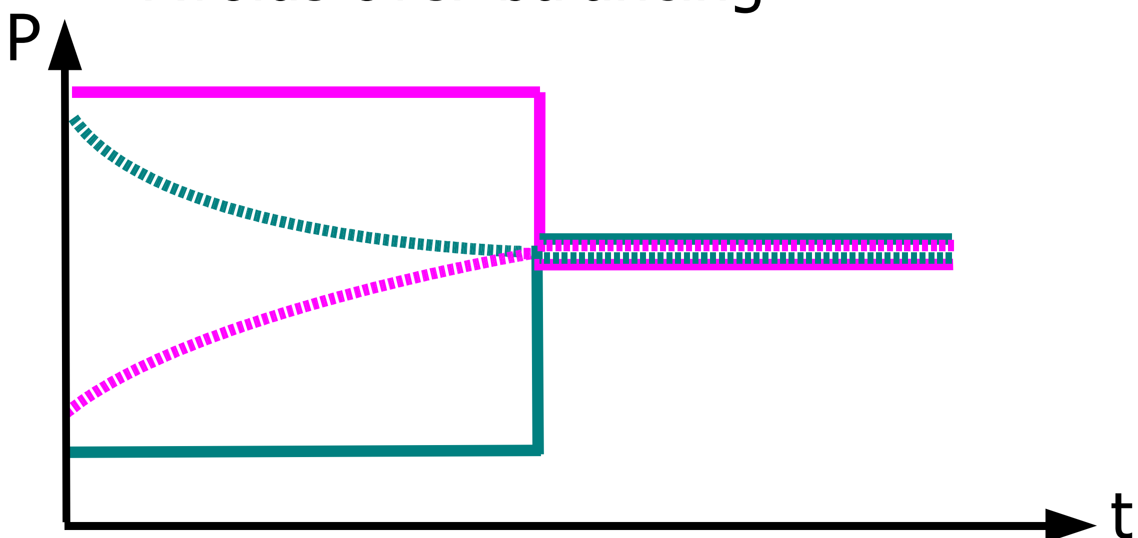
Energy Balancing

- Use both rates for energy balancing
- Migrate a hot task from CPU A to CPU B if both rates for A are greater than both rates for B
  - → Hysteresis
  - → Avoids ping-pong effects
  - → Avoids over-balancing



Energy Balancing

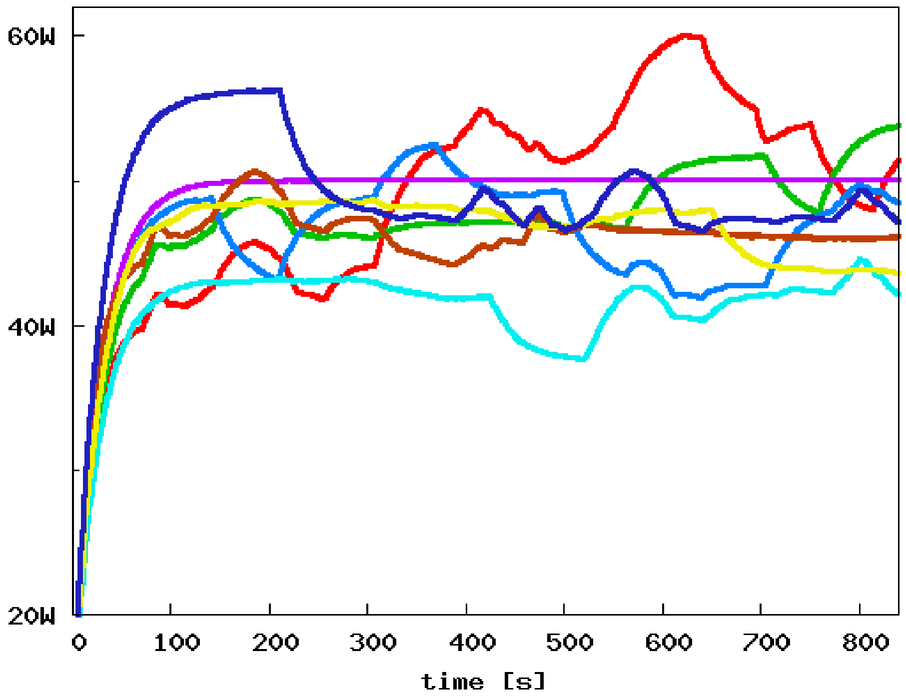
- Use both rates for energy balancing
- Migrate a hot task from CPU A to CPU B if both rates for A are greater than both rates for B
  - ➔ Hysteresis
  - ➔ Avoids ping-pong effects
  - ➔ Avoids over-balancing



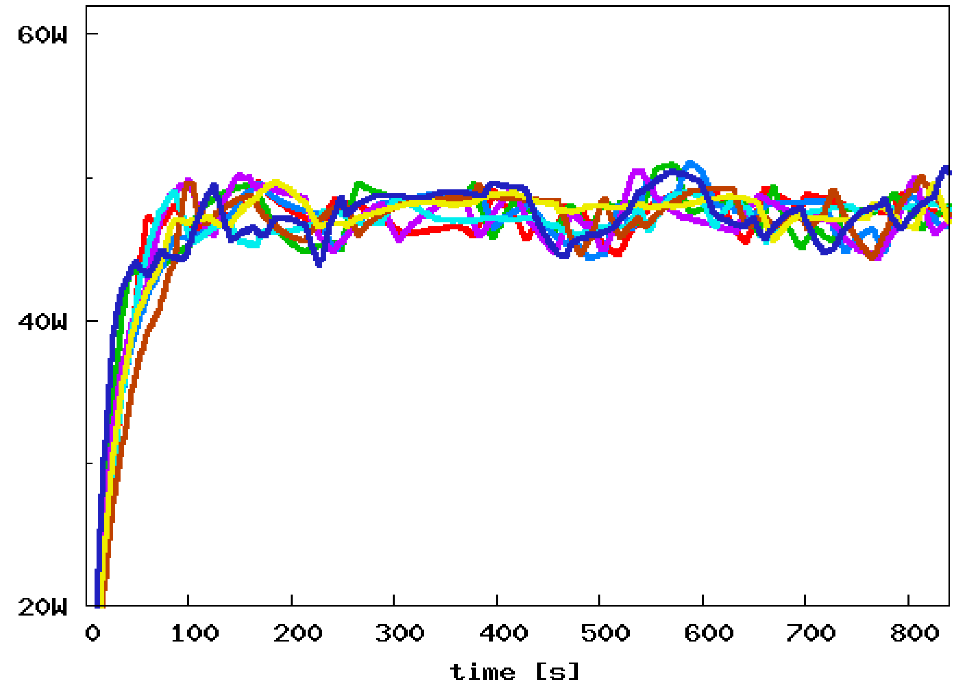
Energy Balancing

- Scenario: 8 CPUs executing different tasks

- Disabled



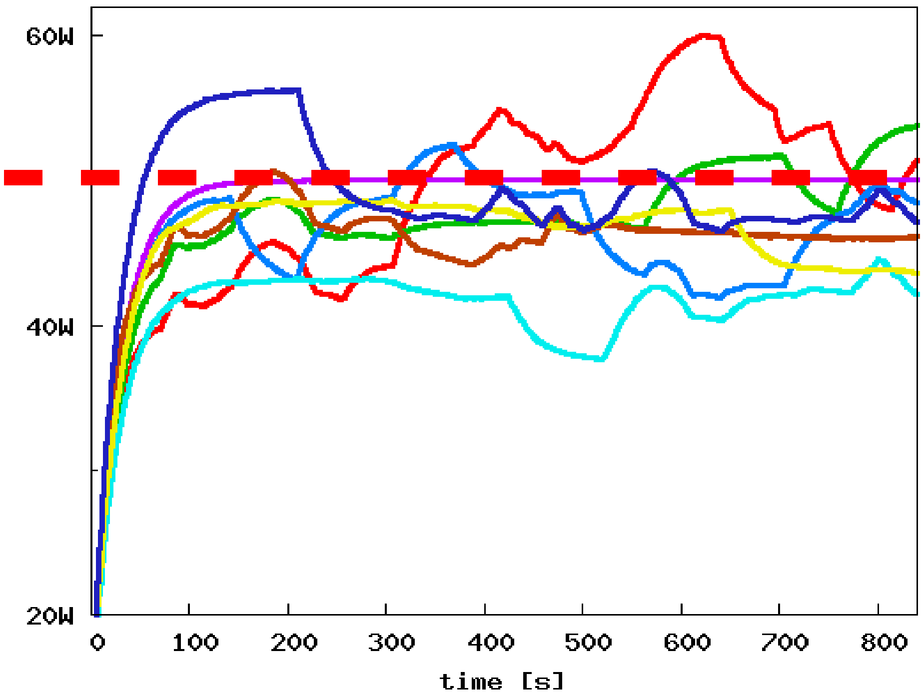
- Enabled



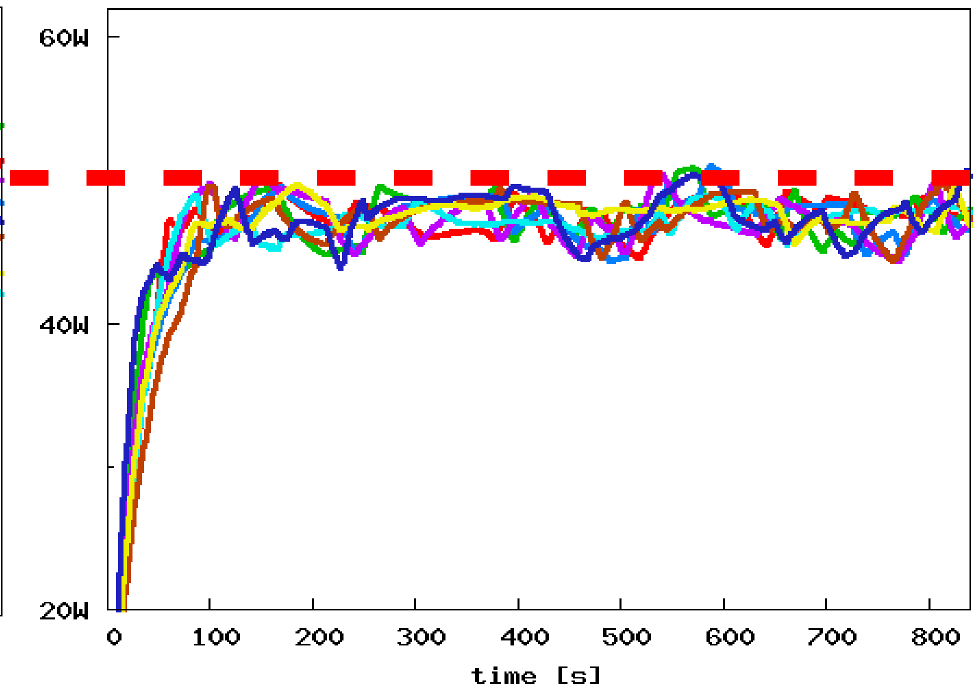
Energy Balancing

- Scenario: 8 CPUs executing different tasks

- Disabled



- Enabled



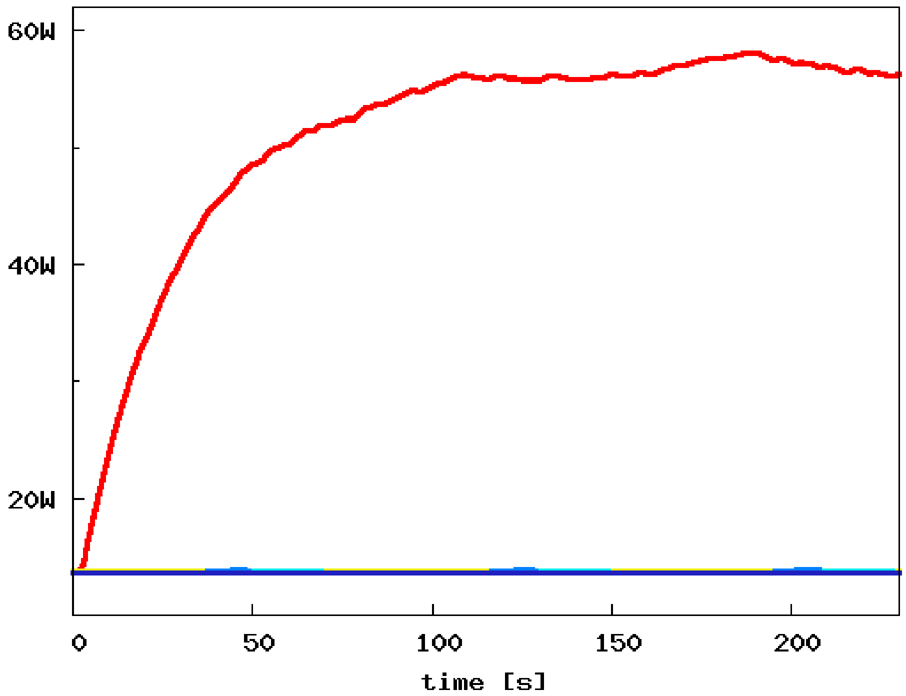
Hot Task Migration

- Only one task in a runqueue
  - ➔ Balancing not possible
- Migrate task to cooler CPU if CPU temperature comes close to maximum
- Search for cool target CPU
  - Idle CPU
    - ➔ Migrate task
  - CPU executing cool task
    - ➔ Swap tasks

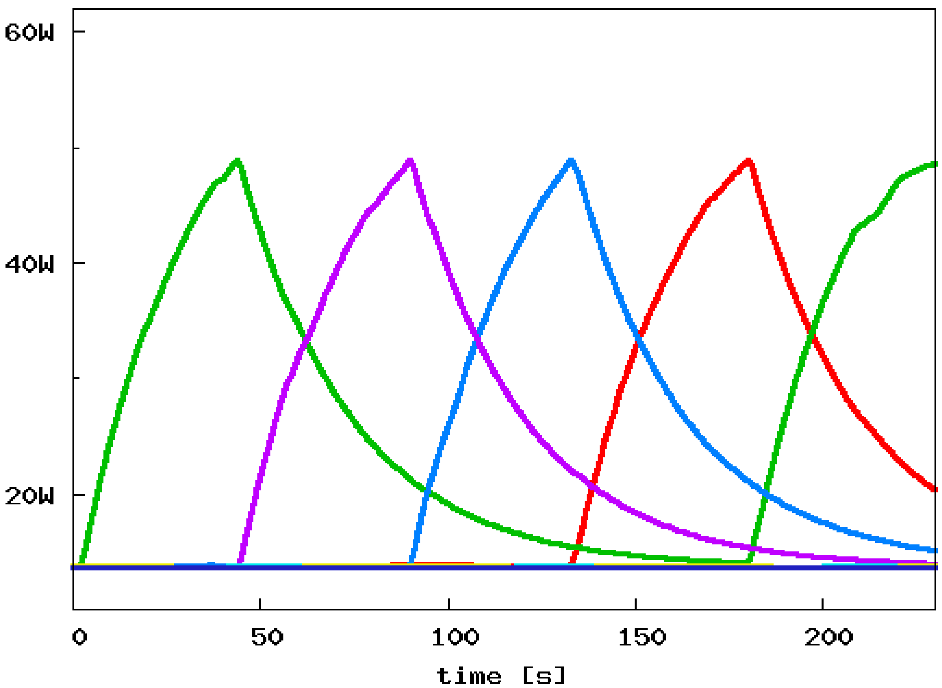
Hot Task Migration

- Scenario: 8 CPUs, 1 hot task

- Disabled



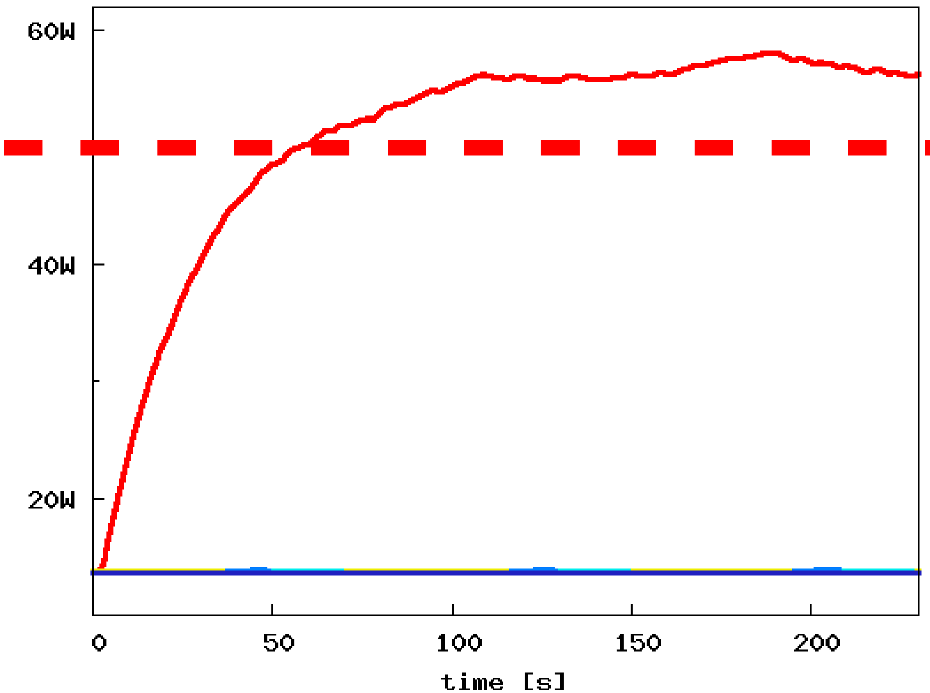
- Enabled



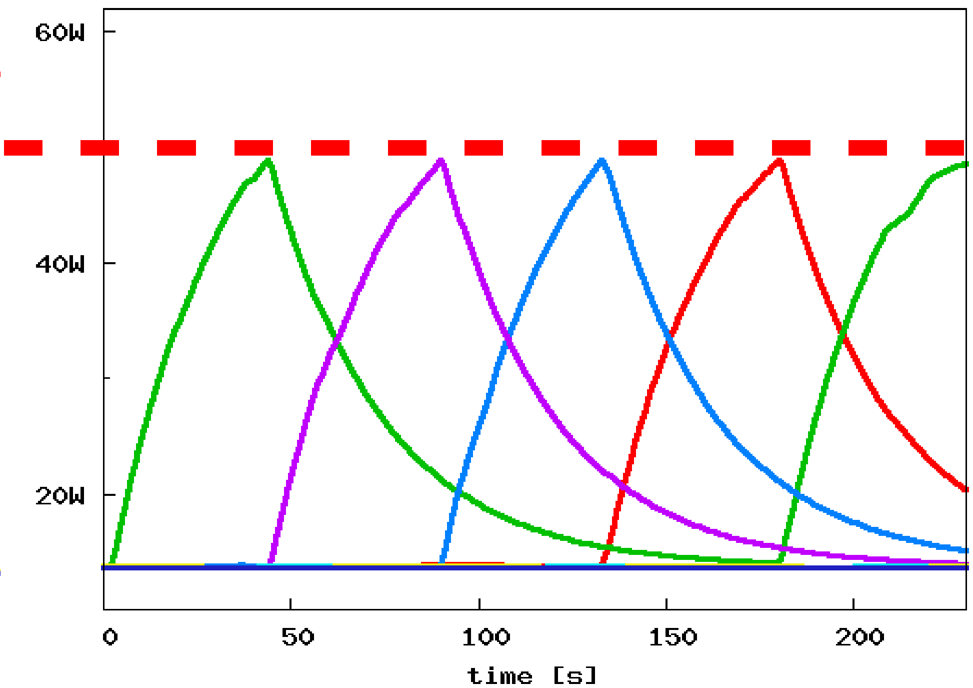
Hot Task Migration

- Scenario: 8 CPUs, 1 hot task

- Disabled



- Enabled



Evaluation

Evaluation

- Implementation of energy aware scheduling for the Linux kernel
- Test system:
  - 8-way Pentium 4 Xeon, 2.2 GHz
- Mixed workload:
  - 18 tasks
  - Power consumption ranging from 37W to 61W
- Temperature control:
  - Throttle a processor if temperature exceeds 38°C
  - Without temperature control highest temperature is 45°C

Results

- Energy-aware scheduling reduces need for throttling
- Throttling percentages in our example:
  - With energy-aware scheduling disabled: 15.2%
  - With energy-aware scheduling enabled: 10.2%
- Gain in duty cycles exceeds overhead for migrations
  - ➔ Increase in throughput
- In our example:
  - Number of tasks finished per time unit increases by 4.7%

Conclusion

- Characterize tasks by power consumption
- Determine energy profiles using event counters
- Use task energy profiles for energy-aware scheduling
  - Energy balancing
  - Hot task migration

- **Reduce thermal imbalances in SMP systems**
- **Minimize throttling → increase duty cycles**