Event-Driven Energy Accounting for Dynamic Thermal Management

Frank Bellosa • Simon Kellner • Martin Waitz • Andreas Weissel

Department of Computer Science 4 (Operating Systems)
University of Erlangen-Nuremberg
Martensstr. 1, 91058 Erlangen, Germany
{bellosa,kellner,waitz,weissel}@cs.fau.de

Benefits of Dynamic Thermal Management

- Cooling laptops, desktop PCs
  - reduced power needed for cooling
  - reduced fan speed & noise
- Cooling servers, server clusters
  - no need for overprovisioning, reduced costs (space, maintenance, ...)
  - safe operation in case of cooling unit failure

➜ Temperature sensors

Drawbacks of Existing Approaches

- If critical temperature is reached
  - throttle the CPU (e.g. halt cycles, stop clock mechanism) or
  - stop the processor execution
- But: neglect of application-, user- or service-specific requirements due to missing online information about
  - the originator of a specific hardware activation and
  - the amount of energy consumed by that activity

➜ Throttling penalizes all tasks

Outline

- From events to energy
  - event-monitoring counters
  - on-line estimation of energy consumption
- Energy containers
  - accounting of energy consumption
- From energy to temperature
  - temperature model
  - implementation of temperature estimation
  - task-specific temperature management

Approaches to Energy Characterization

- Reading of thermal diode embedded in modern CPUs
  - low temporal resolution
  - significant overhead: 5.5 ms (system management bus)
  - ➜ no information about originator of power consumption

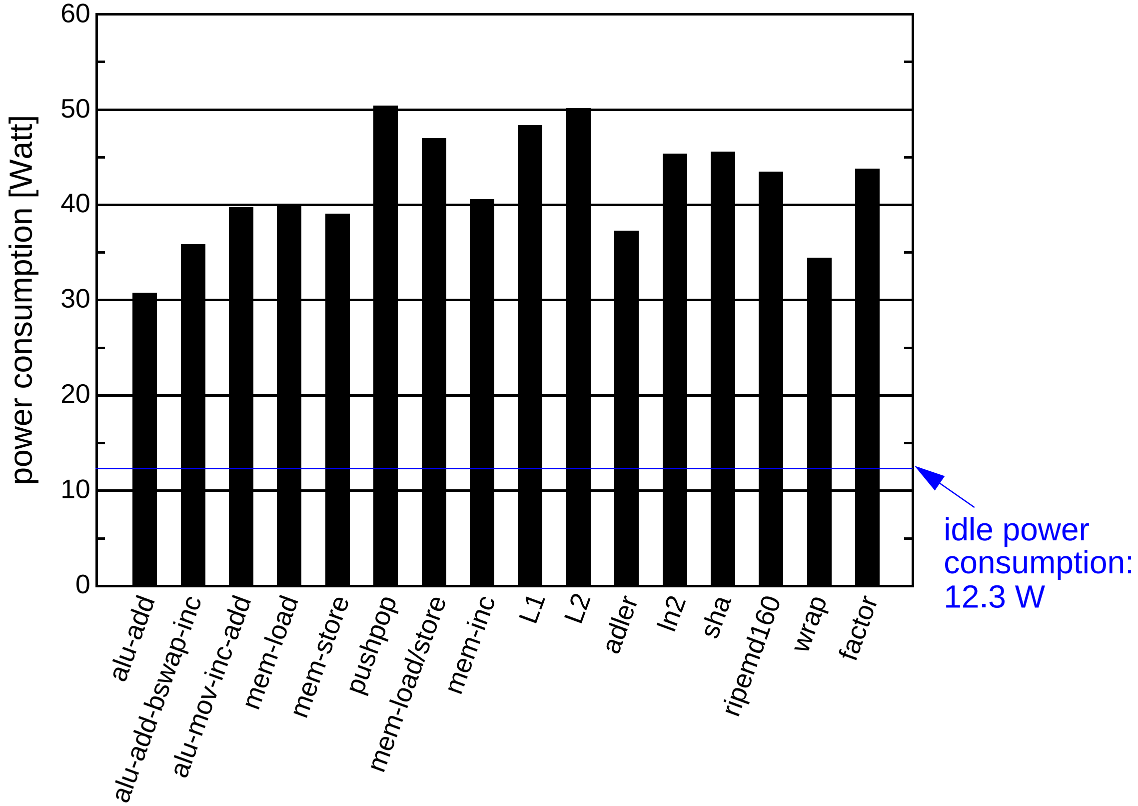
Approaches to Energy Characterization

- Reading of thermal diode embedded in modern CPUs
  - low temporal resolution
  - significant overhead: 5.5 ms (system management bus)
  - ➜ no information about originator of power consumption

- Counting CPU cycles
  - time as an indicator for energy consumption
  - time as an indicator for contribution to temperature level
  - throttling according to runtime
  - ➜ but: wide variation of the active power consumption

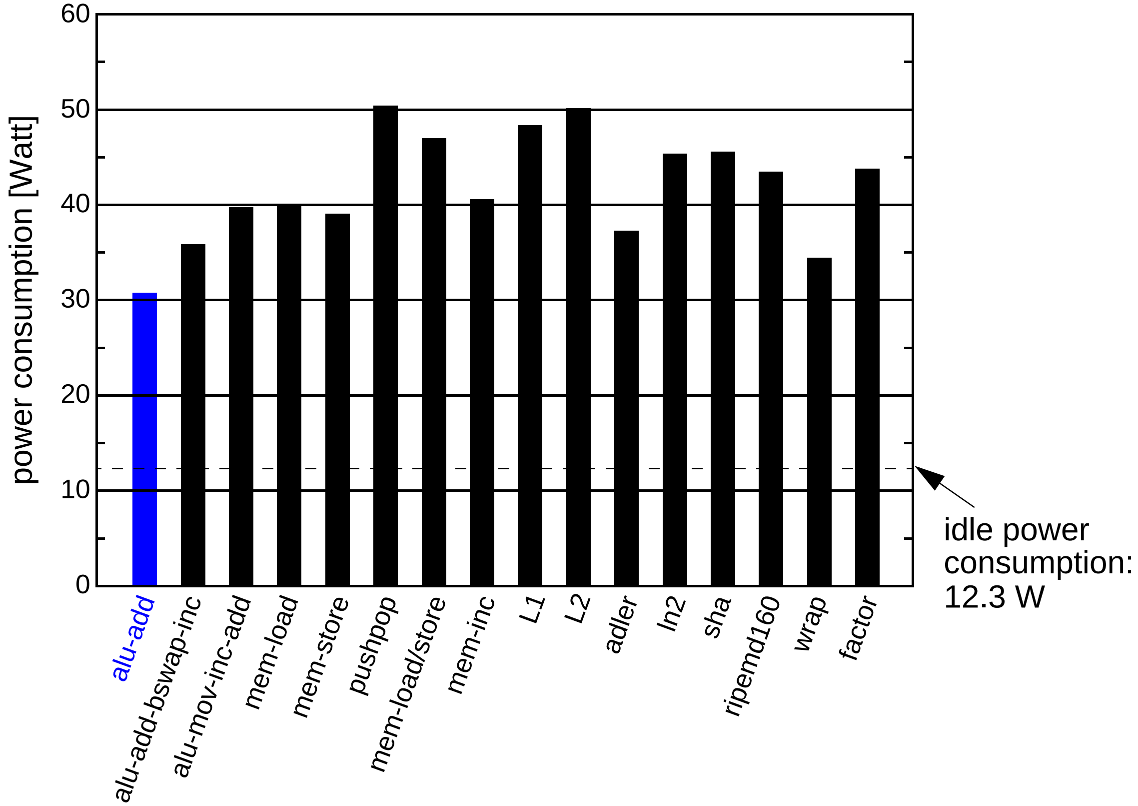
Approaches to Energy Characterization

- P4 architecture (2 GHz) running compute intensive tasks: 30–51 W



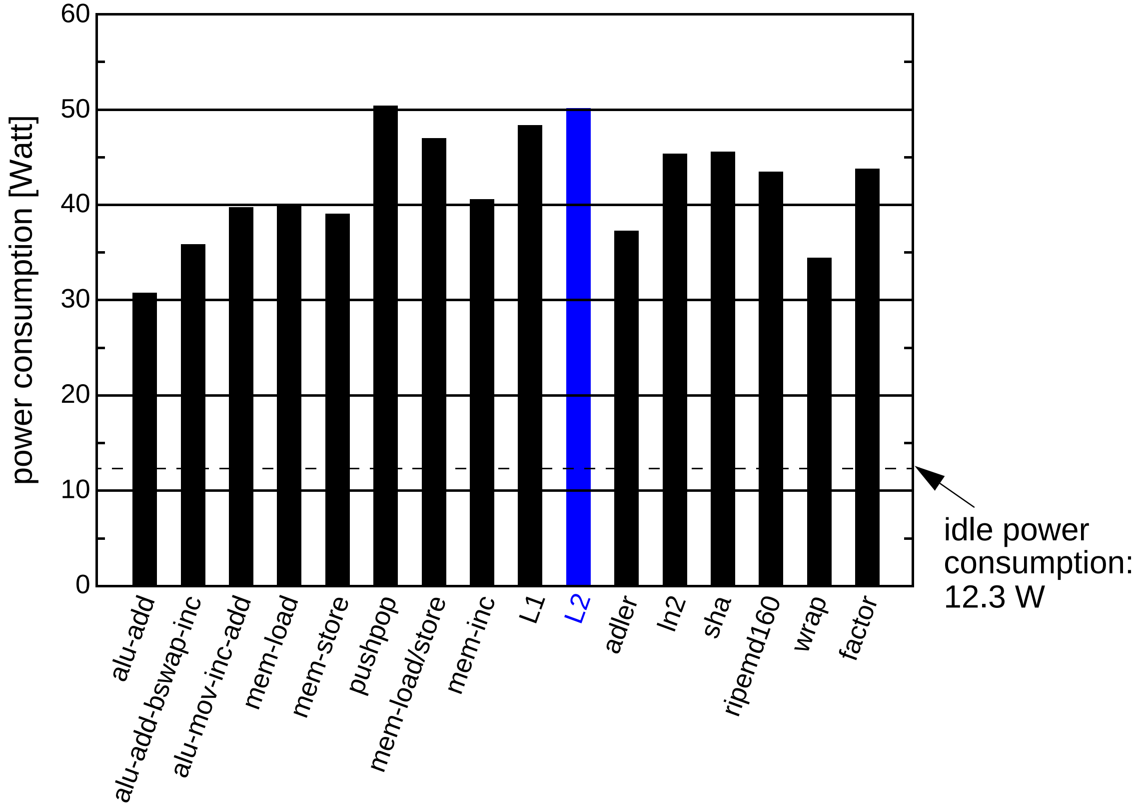
Approaches to Energy Characterization

- P4 architecture (2 GHz) running compute intensive tasks: 30–51 W



Approaches to Energy Characterization

- P4 architecture (2 GHz) running compute intensive tasks: 30–51 W



Approaches to Energy Characterization

- Reading of thermal diode embedded in modern CPUs
  - low temporal resolution
  - significant overhead: 5.5 ms (system management bus)
  - → no information about originator of power consumption

- Counting CPU cycles
  - time as an indicator for energy consumption
  - time as an indicator for contribution to temperature level
  - throttling according to runtime
  - → but: wide variation of the active power consumption
  - → CPU cycles are no clear indicator for energy consumption

→ The case for high-resolution energy estimation

From Events to Energy: Event-Monitoring Counters

- Performance counters register energy-critical events in the complete system architecture.
  - several events can be counted simultaneously
  - low algorithmic overhead
  - low memory overhead
  - high temporal resolution
  - fast response

- Energy estimation
  - ➜ correlate a processor-internal event to an amount of energy
  - select several events and use a linear combination of these event counts
  - estimate the processor's energy consumption

From Events to Energy: Methodology

- Measure the energy consumption of training applications
- Find the events with the highest correlation to energy consumption
  - only certain combinations of events can be counted simultaneously
- Find the linear combination of these events that produce the minimum estimation error
- Avoid underestimation of energy consumption

event	weight [nJ]	max. rate (events per cycle)	power contribution @ 2GHz
time stamp counter	6.17	1.0000	12.33 W
unhalted cycles	7.12	1.0000	14.23 W
µop queue writes	4.75	2.8430	26.99 W
retired branches	0.56	0.4738	0.53 W
mispred branches	340.46	0.0024	1.62 W
mem retired	1.73	1.1083	3.84 W
ld miss 1L retired	13.55	0.2548	6.91 W

From Events to Energy: Accuracy

- Training programs: relative error of 0–29%
- Average relative error: 5.96%
- “Real-world” applications: relative error < 6%

Application	Estimation error
Mozilla 1.0.0	-0.56%
Linux 2.5.64 kernel-build	4.16%
jvm98 1.03	2.20%
caffeine 2.5	6.09%
perl	4.95%
MiBench	-1.73%

- Few cases with underestimation
  - MiBench: floating point operations
  - no events for MMX, SSE & floating point instructions

Outline

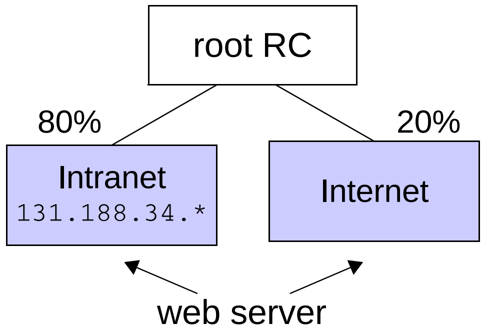
- From events to energy
  - event-monitoring counters
  - on-line estimation of energy consumption
- Energy containers
  - accounting of energy consumption
- From energy to temperature
  - temperature model
  - implementation of temperature estimation
  - task-specific temperature management

Drawbacks of Process Accounting

- Accounting to the execution domain (processes or threads)
- Example: web server serving requests from different client classes
  - e.g. internet/intranet, different service contracts
  - cannot be distinguished on the process level
  - ➜ accounting to different tasks/activities/clients not possible
  - ➜ “resource principal” can change dynamically

Energy Containers

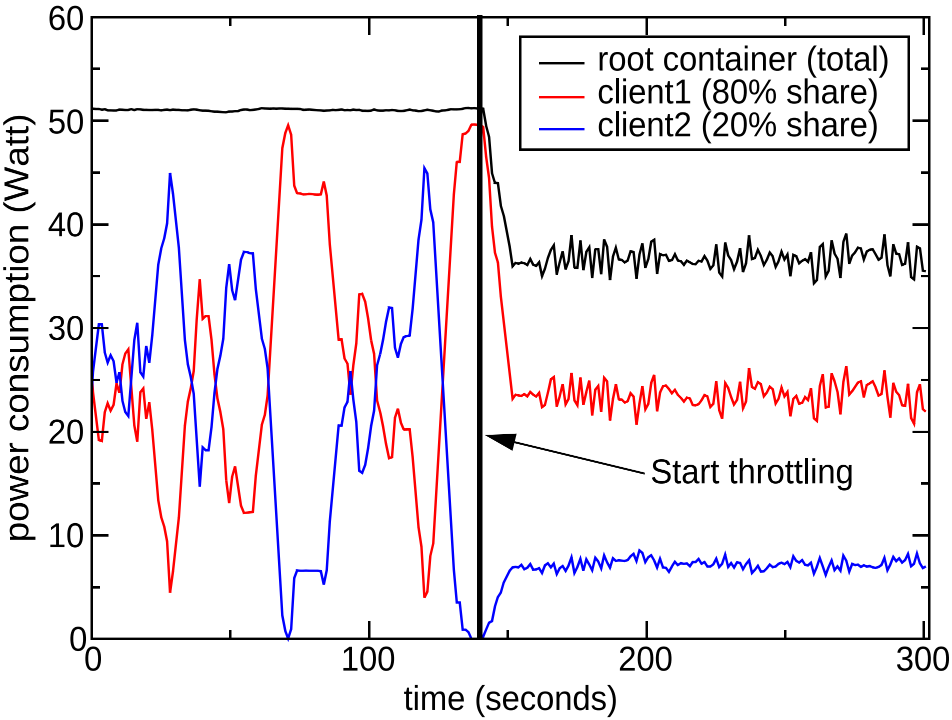
- Resource Containers [OSDI '99] → Energy Containers
  - separation of protection domain and "resource principal"
- Container Hierarchy
  - root Container (whole system)
  - processes are attached to containers



- Energy shares
  - amount of energy available (depending on energy limit
  - periodically refreshed
  - if a container runs out of energy, its processes are stopped
- Packet filter attaches server to the appropriate energy container
  - depending on source address of the packet
  - ➜ energy is automatically accounted to the activity responsible for it

Energy Containers

- Example:
  server application working for two clients with different shares

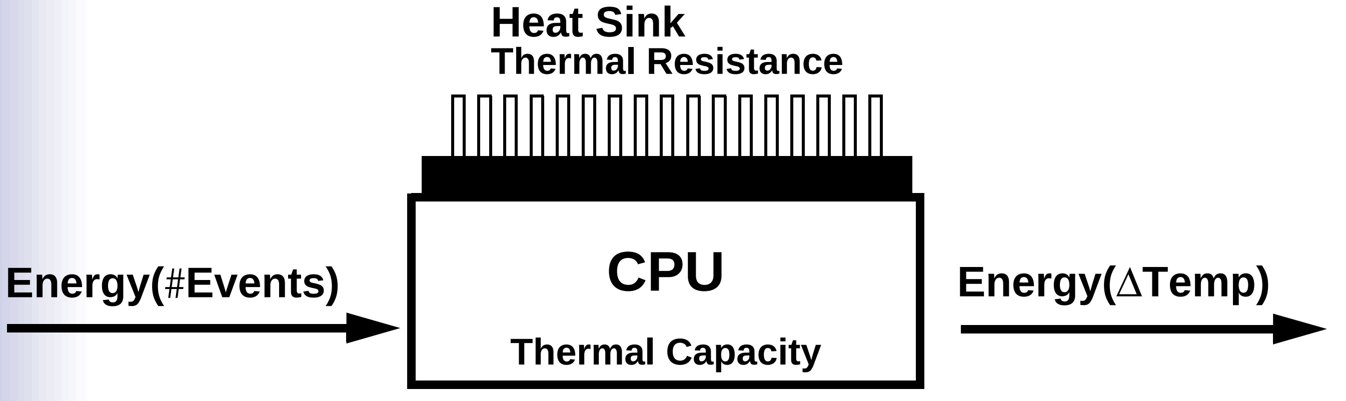


Outline

- From events to energy
  - event-monitoring counters
  - on-line estimation of energy consumption
- Energy containers
  - accounting of energy consumption
- From energy to temperature
  - temperature model
  - implementation of temperature estimation
  - task-specific temperature management

From Energy to Temperature: Thermal Model

- CPU and heat sink treated as a black box with energy in- and output



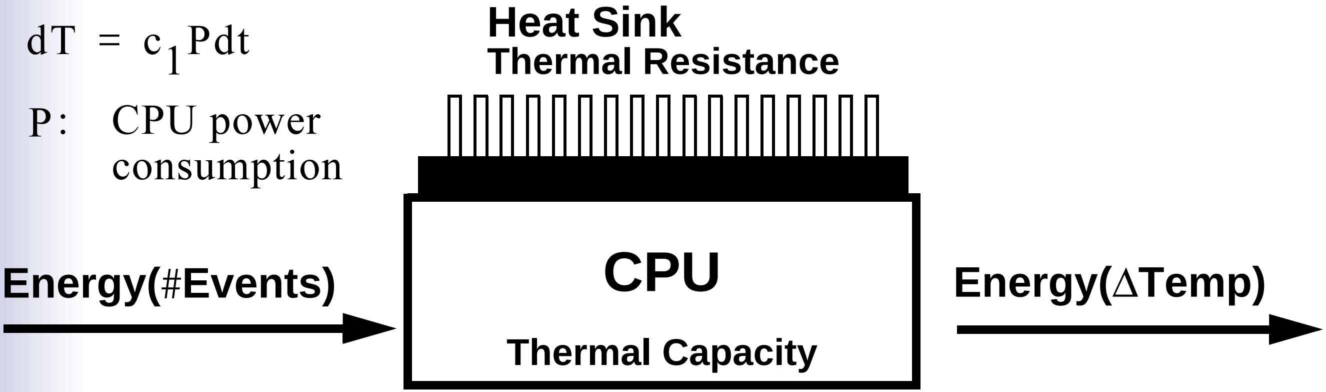
- ◆ energy input: electrical energy being consumed
- ◆ energy output: heat radiation and convection

From Energy to Temperature: Thermal Model

- **Energy input**: energy consumed by the processor

$$dT = c_1 P dt$$

P: CPU power consumption



From Energy to Temperature: Thermal Model

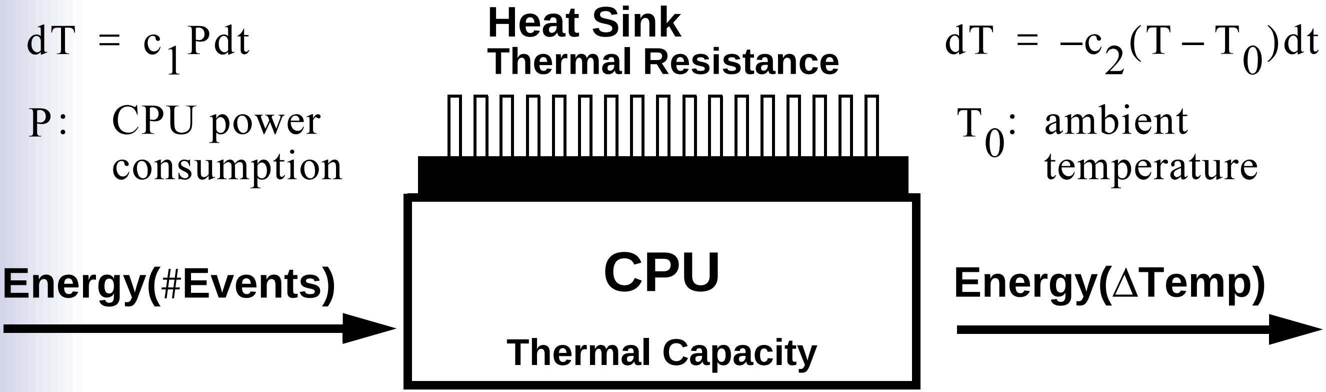
- **Energy output:** primarily due to convection

$$dT = c_1 P dt$$

P: CPU power consumption

$$dT = -c_2 (T - T_0) dt$$

T_0: ambient temperature



From Energy to Temperature: Thermal Model

- Altogether:

$$dT = [c_1 P - c_2 (T - T_0)] dt$$

  - energy estimator → power consumption P
  - time stamp counter → time interval dt
  - the constants c_1, c_2 and T_0 have to be determined

From Energy to Temperature: Thermal Model

- Altogether:

$$dT = [c_1 P - c_2 (T - T_0)] dt$$

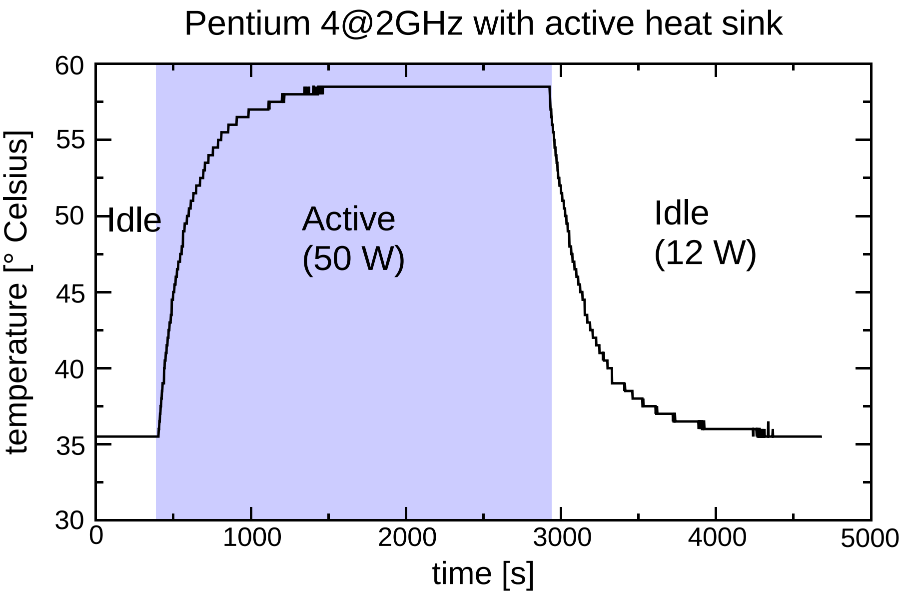
  - energy estimator → power consumption P
  - time stamp counter → time interval dt
  - the constants c_1, c_2 and T_0 have to be determined

- Solving this differential equation yields

$$T(t) = \underbrace{\frac{-c_0}{c_2} \cdot e^{-c_2 t}}_{\text{dynamic part}} + \underbrace{\frac{c_1}{c_2} \cdot P + T_0}_{\text{static part}}$$

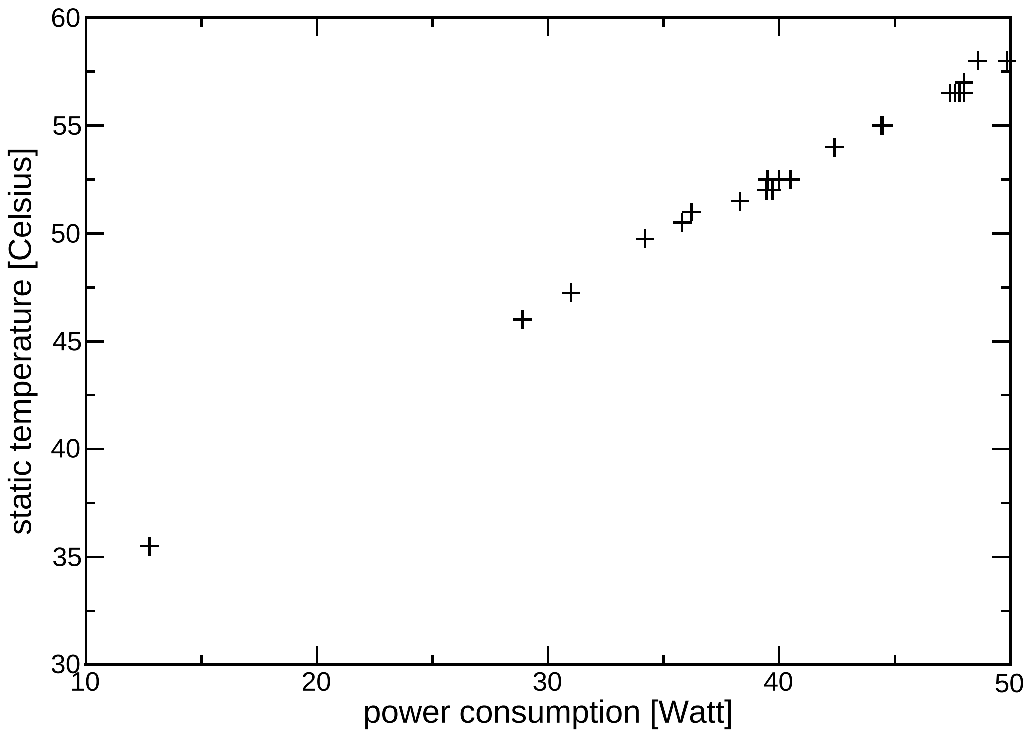
Thermal Model: Dynamic Part

- Measurements of the processor temperature
  - ◆ on a sudden constant power consumption and
  - ◆ a sudden power reduction to HLT power.
  - ➜ fit an exponential function to the data: coefficient = c_2



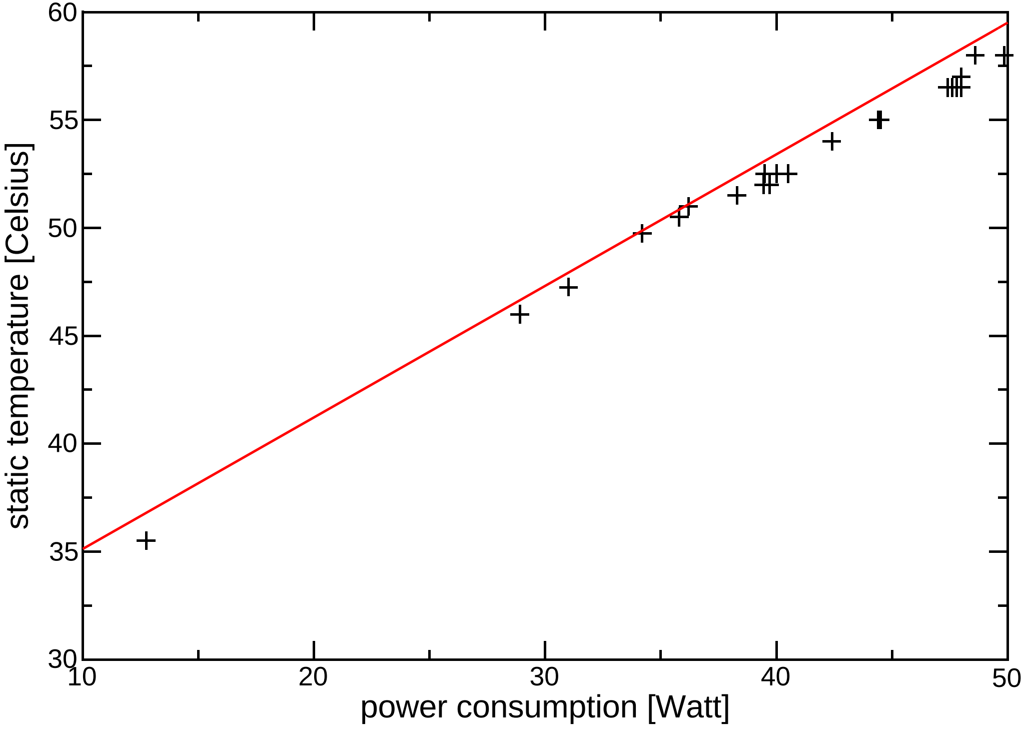
Thermal Model: Static Part

- Static temperatures and power consumption of the test programs



Thermal Model: Static Part

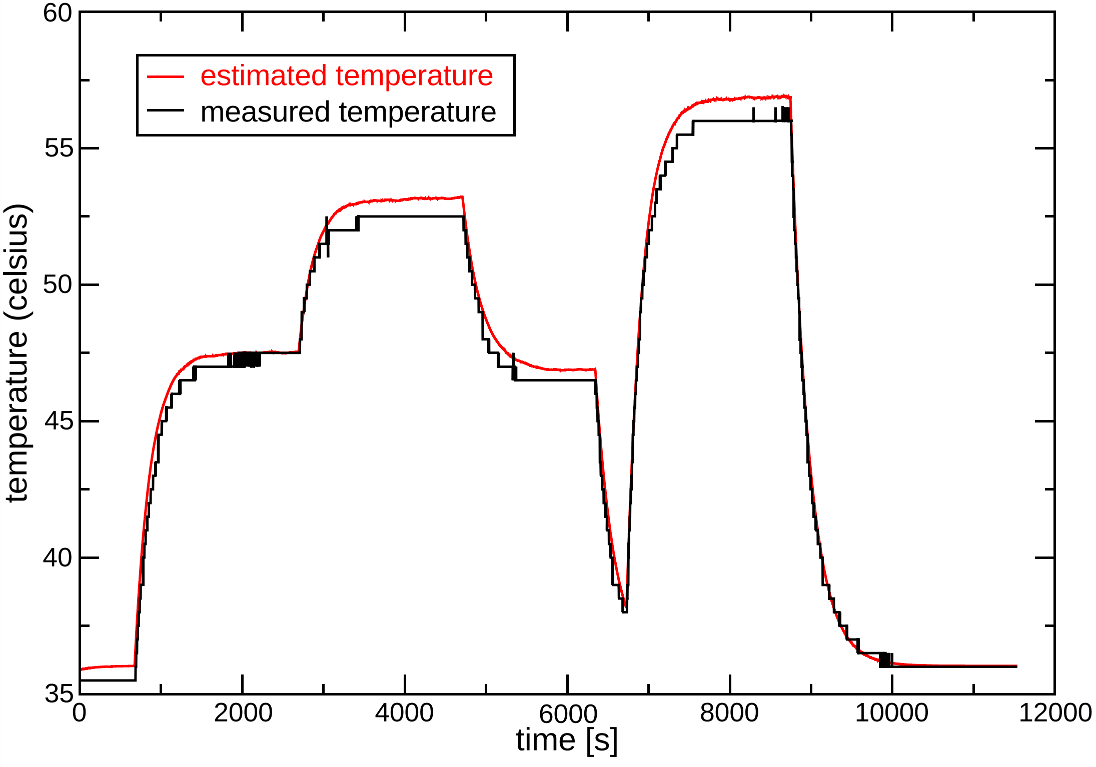
- Linear function to determine c_1 and T_0



Thermal Model: Implementation

- Linux 2.5 kernel
- Periodically read the energy consumption from the root container and compute a temperature estimation
- Deviation of a few degrees celsius over 24 hours
- Re-calibration with measured temperature every 10 to 20 minutes

Thermal Model: Accuracy



Temperature Management

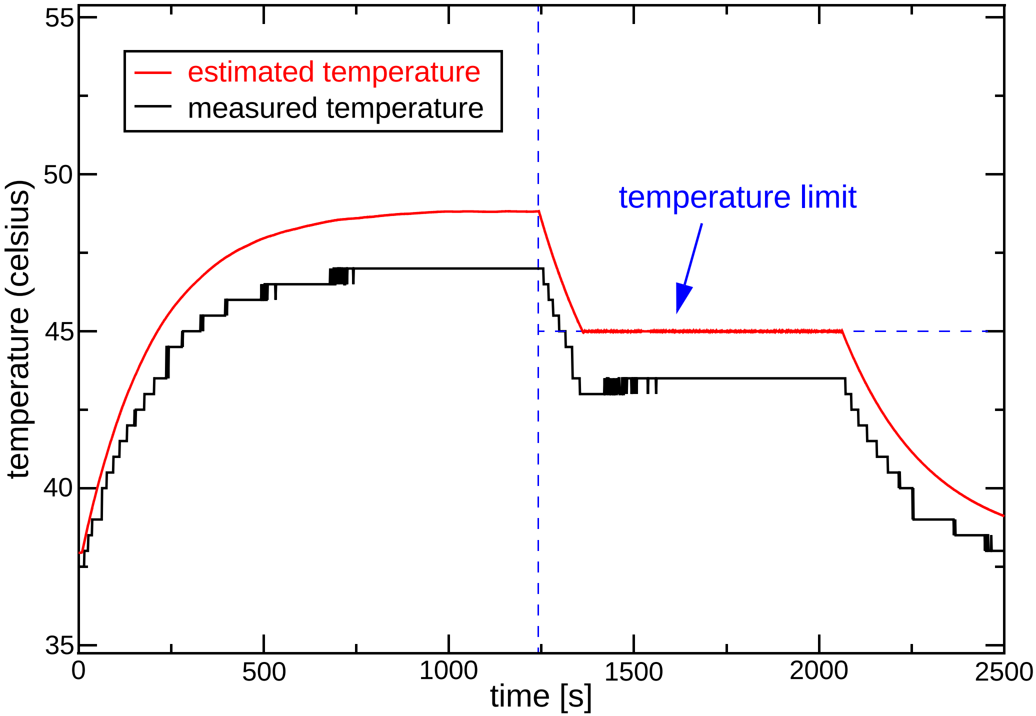
- Periodically compute an energy limit for the root container (depending on the temperature limit T_{limit})

$$dT = [c_1 P + c_2 (T - T_0)] dt \leq T_{limit} - T$$

- Dissolve to $P \rightarrow P_{limit}$
- Energy budgets of all containers are limited according to their shares
- Tasks are automatically throttled according to their contribution to the current temperature

Temperature Management

- Example: Enforcing a temperature limit of 45º



Temperature Management: Accuracy & Overhead

- Real-world applications
  - estimation within accuracy of temperature measurement (< 1º celsius)
  - errors in temperature estimations always due to energy model
  - ➜ limitations of performance counters
- Temperature limits are kept with negligible deviation
- Overhead
  - energy containers, estimation & temperature management: < 1%

Conclusion

- Event-monitoring counters enable
  - on-line energy accounting
  - task-specific temperature management
  - low overhead

- Future directions
  - examine more sophisticated energy models
  - task-specific frequency scaling to adjust the thermal load