nano-tera.ch
swiss scientific initiative in health / security / environment systems

YINS RTD 2013

FNSNF

COOLIP: Simple yet Effective Job allocation For Distributed Thermally-Throttled Processors

ETH zürich

Pratyush Kumar, Hoeseok Yang, Iuliana Bacivarov, Lothar Thiele

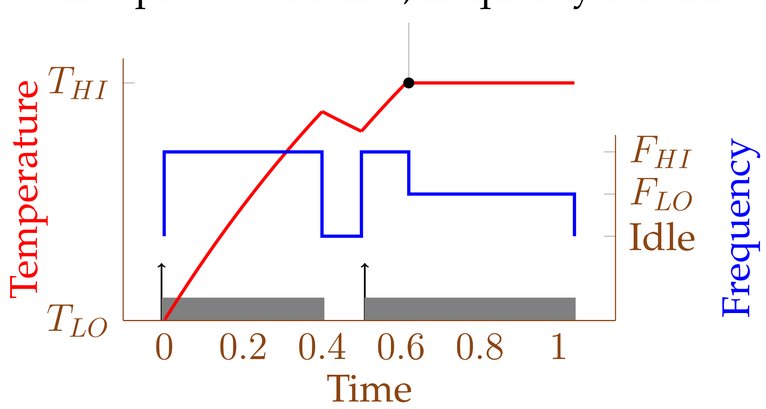
Computer Engineering and Networks Laboratory, ETH Zurich

Big Goal

In a data-center with multiple processors, where to execute the next job? Processors often run hot and need throttling.

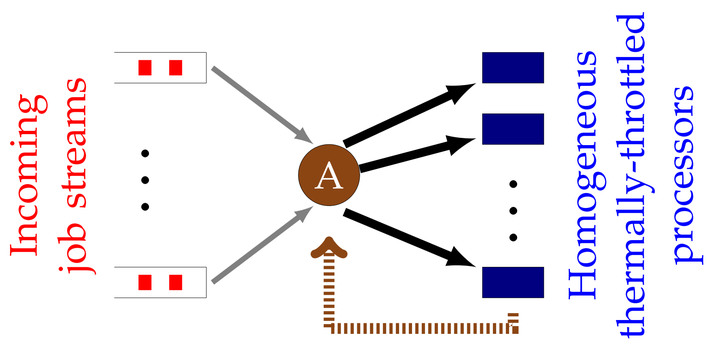
Thermally Throttled Processors

Once processor is hot, frequency is reduced



Most processors have in-built speed scaling. Usage of the higher speeds is throttled by temperature. For example, Turbo Boost in Intel Sandy Bridge processors can run the processor at up to 1.2-1.3x the Thermal Design Power (TDP) for up to 30-60s.

The Allocation Problem



Map incoming job to one of the processors to minimize response-times. Processors are identical and thermally isolated. Can sense if a processor is active or idle, and its temperature.

Idea: **Thermally-multiplex** jobs amongst the processors, to operate often in high frequencies.

Minimize Response-Time of the Next Job

Question: Is there a simple strategy which optimally minimizes the response-time of the next job?
Answer: Only true if the processors are in one of the **good states**.

State = $\big((t_1, T_1), (t_2, T_2), \ldots, (t_n, T_n)\big)$

- t_i: Time when processor will be idle
- T_i: All-idle temperature: Temperature when *all* processors will be idle

Good state: Earliest finishing processor also has the lowest all-idle temperature, i.e., for some i,

$$\arg\min\{t_1, t_2, \ldots, t_n\} = i = \arg\min\{T_1, T_2, \ldots, T_n\}$$

Simple strategy in a good state: Route next job to ith processor
This is **optimal**, independent of all parameters.

The COOLIP policy

COOLIP = COOLest among Idle Processors

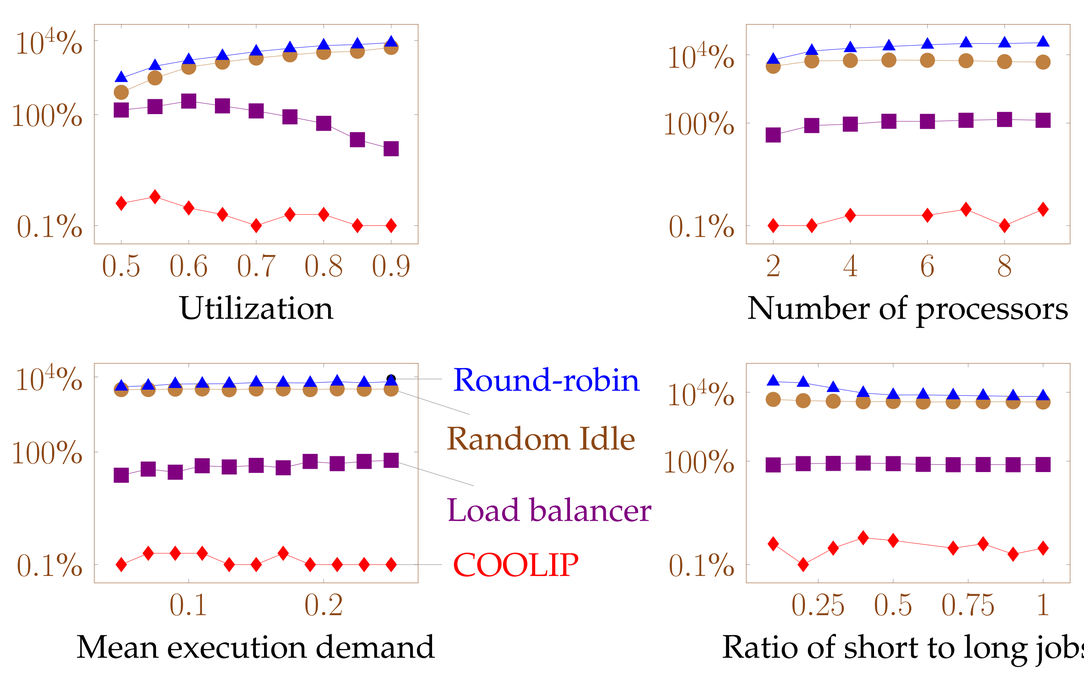
Route the next job to the earliest available processor. If there are several available, choose the coolest one.

In a good state, COOLIP optimally minimizes response-time. Else, no guarantee.

Experimental results

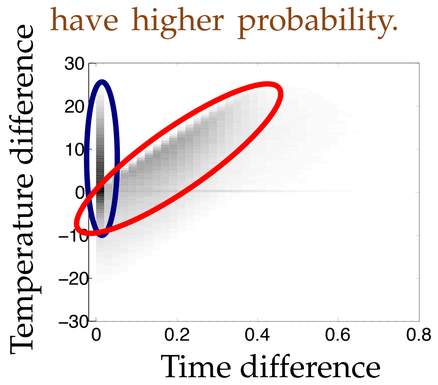
- Processors have Fourier heat dissipation.
- Jobs have Poisson inter-arrival times.
- Execution demand of jobs have a Gaussian distribution.

Deviation of 95-percentile response-time compared to optimal clairvoyant solution



- COOLIP performs almost identical to optimal solution.
- Other algorithms perform significantly worse.
- Even additional information such as execution demand of all pending jobs (load balancer) does not improve over COOLIP.

Why COOLIP works well?



Stationary distribution of states. Darker points have higher probability.

Processor states with COOLIP define a **Markov process**. Can identify the stationary distribution of the states.
Idea: Show that frequent states are good states.

Example: A setup with two processors. **Both sets of frequent states are good states.**

Conclusions and Outlook

COOLIP is a simple strategy that allocates most of the packets optimally, and has a near-optimal 95-percentile response time.
Open challenges: (a) Thermal coupling between processors, (b) Heterogeneous processors, (c) Bursty job arrival patterns.