Heat-aware Loadbalancing - Is it a thing?

Lukas Ifländler, Norbert Schmitt, Andreas Knapp, and Samuel Kounev

Chair of Software Engineering – Universität Würzburg

November 13, 2020

11th Symposium on Software Performance 2020, 12.–13. November 2020, Leipzig, Germany

https://se.informatik.uni-wuerzburg.de/
• Increasing amount of cloud users and services
Motivation

- Increasing amount of cloud users and services
- Increasing requirements on computing power by more traffic
  - Increasing computing power needed

[1] Motivation


[3]
Increasing amount of cloud users and services

Increasing requirements on computing power by more traffic

Increasing computing power needed

Solution 1: More servers?
Increasing amount of cloud users and services

Increasing requirements on computing power by more traffic

Increasing computing power needed

Solution 1: More servers? ✗

Solution 2: Better/Full usage of existing computing power ✓

make use of unallocated CPU boost power
1. Is it possible to exceed permanently and feasibly the clock rate with host relay?
1. Is it possible to exceed permanently and feasibly the clock rate with host relay?

2. What is the effect on the computing power compared to normal behavior?
Goals

1. Is it possible to exceed permanently and feasibly the clock rate with host relay?

2. What is the effect on the computing power compared to normal behavior?

3. What is the effect on the energy consumption compared to normal behavior?
1. A first approach to heat-aware load-balancing
1. A first approach to heat-aware load-balancing

2. A prototype implementation with SDN and an initial evaluation
Approach - Boosting

Intel Turbo Boost:
- CPU-Boost ≠ Overclocking in general
- CPU Boost respects specifications of hardware
- Max. n° of boosted cores and max. frequency is limited by headroom
- OS requests more computing power ➔ Boost
- Example Intel:
  - Stepwise increasing clock with 133.33 MHz
  - Time in boost state depends on:
    - Temperature
    - Energy

Other common used boost: AMD Turbo Core / AMD Precision Boost

Boosts are usually disabled in related work and HPC environments due to unpredictability
## Approach

<table>
<thead>
<tr>
<th>Foundations</th>
<th>Research Question</th>
<th>Approach</th>
<th>Evaluation</th>
<th>Conclusion</th>
</tr>
</thead>
</table>

### Motivation

### Goals & Contributions

### Approach

### Evaluation

### Conclusion
Approach
Approach
Approach

Motivation    > Goals & Contributions > Approach > Evaluation > Conclusion
Approach

Foundations ➤ Research Question ➤ Approach ➤ Evaluation ➤ Conclusion
Approach
Motivation  Goals & Contributions  Approach  Evaluation  Conclusion
Approach
Approach
Definition:
Not max. boosted anymore
⇒ time to switch
Definition:
Not max. boosted anymore
\[ \Rightarrow \text{time to switch} \]

Assumptions:

- The workload puts enough stress on a server that it will boost.
- The server is under stress that the migrated workload will only be adequately serviced when boosted.
Prototype Implementation

- Client: HTTP Load Generator

[4]
Prototype Implementation

- Client: **HTTP Load Generator**
- Worker:
  - Workload: **Bungee-LU-Servelet**
  - Data collector: **Telegraf**

Motivation > Goals & Contributions > Approach > Evaluation > Conclusion
Prototype Implementation

Client: HTTP Load Generator

Worker:
• Workload: Bungee-LU-Servelet
• Data collector: Telegraf

Monitor
• Database: InfluxDB
• Dashboard: Chronograf
• SDN-Controller framework: Ryu
Prototype Implementation

- **Client:** HTTP Load Generator
- **Worker:**
  - Workload: Bungee-LU-Servelet
  - Data collector: Telegraf
- **Monitor**
  - Database: InfluxDB
  - Dashboard: Chronograf
  - SDN-Controller framework: Ryu
- **Switch:** Software-Switch with Open vSwitch
Prototype Implementation

- **Client:** HTTP Load Generator
  - **Worker:**
    - Workload: Bungee-LU-Servelet
    - Data collector: Telegraf
- **Monitor:**
  - Database: InfluxDB
  - Dashboard: Chronograf
  - SDN-Controller framework: Ryu
- **Switch:** Software-Switch with Open vSwitch
- **Testbed with three servers, SDN, and controller**
Three evaluated scenarios:

A. Highly overbooked
B. Medium overbooked
C. Not overbooked
Three evaluated scenarios:

A. Highly overbooked
B. Medium overbooked
C. Not overbooked
Three evaluated scenarios:
A. Highly overbooked
B. Medium overbooked
C. Not overbooked
Three evaluated scenarios:
A. Highly overbooked
B. Medium overbooked
C. Not overbooked
## Preliminary Measurements

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Switched</th>
<th>Unswitched</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average response time in seconds</td>
<td>0.85</td>
<td>1.83</td>
</tr>
<tr>
<td>Number of requests sent/received/lost</td>
<td>1310 / 1310 / 0</td>
<td>1310 / 1304 / 6</td>
</tr>
<tr>
<td>Temperature max. r04/r05/r06 in °C</td>
<td>61.99 / 61.13 / 61.35</td>
<td>69.76 / * / *</td>
</tr>
<tr>
<td>Temperature average r04/r05/r06 in °C</td>
<td>59.29 / 58.81 / 59.13</td>
<td>65.96 / * / *</td>
</tr>
<tr>
<td>Average CPU frequency in MHz r04/r05/r06</td>
<td>2459.86 / 2147.86 / 2333.61</td>
<td>3518.44 / * / *</td>
</tr>
<tr>
<td>Average CPU maximum in MHz</td>
<td>3670.24</td>
<td>3580.32</td>
</tr>
<tr>
<td>Average power consumption (idle: 98.19W) in Watt</td>
<td>135.54</td>
<td>147.98</td>
</tr>
<tr>
<td>Additional power consumption in total and percent</td>
<td>37.35 / 38.03%</td>
<td>49.79 / 50.71%</td>
</tr>
</tbody>
</table>
Achievements

• Working heat-aware and boost-oriented load rotation

• Higher average of max. CPU frequency
Achievements

• Working heat-aware and boost-oriented load rotation

• Higher average of max. CPU frequency

Open Tasks

• Include performance metrics
  • Throughput
  • Latency

• Eliminate simplifications

• Stateful migration

• Tests with additional workloads

• Extend to heterogenous systems

• Power and temperature evaluation
Thank You!

https://se.informatik.uni-wuerzburg.de/
References

[4] https://se.informatik.uni-wuerzburg.de/fileadmin/10030200/user_upload/logos/HTTPlogo.png
[5] https://se.informatik.uni-wuerzburg.de/fileadmin/_processed_/9/9/csm_csm_BUNGEE_02_837dac3f07_ddddb202d42.png