HotGauge: A Methodology for Characterizing Advanced Hotspots in Modern and Next Generation Processors

Alexander Hankin\textsuperscript{1*}, David Werner\textsuperscript{1*}, Julien Sebot\textsuperscript{2}, Kaushik Vaidyanathan\textsuperscript{3}, Maziar Amiraski\textsuperscript{1}, Mark Hempstead\textsuperscript{1}

\textsuperscript{1}Tufts University, \textsuperscript{2}Intel, \textsuperscript{3}Google \textsuperscript{*co-authors}

2021 IEEE International Symposium on Workload Characterization

November 7, 2021
Introduction

• System design trends have exacerbated thermal hotspots and have created “advanced” hotspots
  – Exponential power density
  – Cramming additional logic

• Advanced hotspots are a critical design concern and will continue to get worse without proper mitigation

• We have built the necessary tools for studying advanced hotspots and developing mitigation techniques
Advanced hotspot in a 7nm client processor similar to Intel Skylake

Hot functional units surpass \(120^{\circ}\mathrm{C}\) while nearby units within 200 \(\mu\mathrm{m}\) remain 30 degrees cooler.

How bad are they?
How bad are they?

Distribution of temperature deltas over 200 μs intervals

Higher temperature deltas in 7nm.
How bad are they?

(a) 14nm

(b) 7nm

Distribution of temperature deltas over 200 μs intervals

Wider distribution of temperature deltas within 200μs.
Background and Motivation

• Power density is localized, application-dependant, and rapidly increasing
  – standard thermal controllers won’t work
• Lack of a hotspot definition
• Existing simulation methodologies are limited
Project Goal

TODO: sell the project
- Novel metrics and analysis techniques
- End-to-end simulation infrastructure
- Demonstrate using case studies
HotGauge
HotGauge takes as input the system models and workloads to be evaluated for hotspots.

First major task, Perf-power-therm co-simulation, integrates performance, power, and thermal models to perform rapid end-to-end thermal simulation.
Hotspot Characterization

- Rigorous definition of a hotspot
- Automated method for detecting them
- Quantitative metric to compare hotspot severity between floorplans, or other model changes
Hotspot Definition and Detection

- **Goal**: capture all phenomena which will cause either performance loss or reliability problems
  - absolute temperature
  - maximum localized temperature differential (MLTD)
Hotspot Severity Metric

- Hotspot Severity accounts for both MLTD and temperature
- Comprised of multiple parameterized sigmoid functions, each dealing with a different concern

\[ \sigma(x_0, y_0, s, a) = \frac{a}{1 + e^{-s(x-x_0)}} + y_0 \]

\[ \text{sev}(x, y) = \sigma_{df}(T_{x,y}) + \sigma_{M}(\text{MLTD}_{x,y}) \times \sigma_{T}(T_{x,y}) \]

where:

\[ \sigma_{df} = \sigma(115, 0, 0.2, 2) \]
\[ \sigma_{M} = \sigma(15, -0.25, 0.2, 1.25) \]
\[ \sigma_{T} = \sigma(60, 0.35, 0.05, 0.65) \]

1 → crash or permanent damage imminent
0.5 → immediate mitigation required
0 → no concern
Case Study - Skylake Proxy

CPU Microarchitecture Parameters

- Process node [nm]: 14, 10, 7
- Cores: 7
- Core area [mm²]: 5, 2.5, 1.25
- Frequency: 5 GHz
- SMT: 2
- ROB entries: 224
- LQ entries: 72
- SQ entries: 56
- Scheduler entries: 97
- L1I $: Private, 32 KiB
- L1D $: Private, 32 KiB
- L2 $: Private, 512 KiB
- L3 $: Shared ring, 16 MiB

--- | --- | --- | ---
Thermal grease | 0.04e-4 | 3.376e-12 | 30
Copper (heat spreader) | 3.9e-4 | 3.376e-12 | 3e3
Solder TIM | 0.25e-4 | 1.628e-12 | 200
Silicon (IC wafer) | 1.20e-4 | 1.651e-12 | 380

Tufts University
School of Engineering
Electrical and Computer Engineering
Evaluation and Case Study

Using *HotGauge*, we aim to...

- Quantify how bad the problem is
- How much worse will the problem be in next generation processors
- Evaluating mitigations
Results - Temperature Distribution vs Time

Initial conditions dramatically affect how quickly hotspots can arise and distribution of IC temperatures.
MLTD vs. Tech Node

MLTD varies more from core to core

Higher MLTD in newer 7nm node
Results -- TUH vs tech node
Results -- TUH vs warmup
Hotspot behaviour across benchmarks varies considerably. Some benchmarks are more affected by core-placement.
Hotspot Locations

Some areas are prone to hotspots

Some areas have infrequent hotspots
Case Study

Single Unit Mitigation

Goal : decrease hotspots caused by *problematic* units

Baseline : make 7nm unit no worse than 14nm unit
            *(using hotspot-severity)*

Method : scale area of problematic units
          *(thereby decreasing power-density)*
Unit Scaling

Floating Point Instruction Window

Effectiveness of unit scaling varies from benchmark to benchmark
Effectiveness of unit scaling varies from benchmark to benchmark.
Scaling Register Access Tables

- Hotspot severity of 1.0:
  - 14nm: few workloads
  - 7nm: most workloads
- Varied impact of scaling RATs across workloads
Other Contributions

- Model validation
  - Thermal stack
  - Power Model
- Modified Sniper/MCPAT for new nodes
- Case Study: Scaling the entire IC
Conclusion

• In this work we introduce HotGauge, a holistic methodology for characterizing hotspots in modern and next generation processors.

• HotGauge details new methods and metrics for characterizing and comparing hotspot severity across any next generation processors.

• This will allow the architecture community to develop architecture level mitigations to work alongside traditional thermal regulation techniques to solve the advanced thermal hotspots which are occurring in modern and next generation processors.
HotGauge FrameWork

Publicly available on GitHub! [GitHub Link]

https://github.com/TuftsCompArchLab/HotGauge

Includes Docker container to run thermal-simulations and perform analysis

Tufts University
School of Engineering
Electrical and Computer Engineering
HotGauge: A Methodology for Characterizing Advanced Hotspots in Modern and Next Generation Processors

Alexander Hankin\textsuperscript{1*}, David Werner\textsuperscript{1*}, Julien Sebot\textsuperscript{2}, Kaushik Vaidyanathan\textsuperscript{3}, Maziar Amiraski\textsuperscript{1}, Mark Hempstead\textsuperscript{1}

\textsuperscript{1}Tufts University, \textsuperscript{2}Intel, \textsuperscript{3}Google \*co-authors

https://github.com/TuftsCompArchLab/HotGauge
# Validation

<table>
<thead>
<tr>
<th></th>
<th>14nm Si</th>
<th>model</th>
<th>$C_{\text{dyn}}$ [nF] error</th>
<th>10nm Si</th>
<th>model</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td>bzip2</td>
<td>1.33</td>
<td>1.36</td>
<td>+2%</td>
<td>1.32</td>
<td>1.17</td>
<td>-11%</td>
</tr>
<tr>
<td>gcc</td>
<td>1.51</td>
<td>1.30</td>
<td>-14%</td>
<td>1.80</td>
<td>1.13</td>
<td>-37%</td>
</tr>
<tr>
<td>omnetpp</td>
<td>1.16</td>
<td>1.33</td>
<td>+15%</td>
<td>0.99</td>
<td>1.16</td>
<td>+17%</td>
</tr>
<tr>
<td>povray</td>
<td>1.87</td>
<td>1.62</td>
<td>-13%</td>
<td>1.87</td>
<td>1.36</td>
<td>-27%</td>
</tr>
<tr>
<td>hmem</td>
<td>1.52</td>
<td>1.65</td>
<td>+9%</td>
<td>1.49</td>
<td>1.38</td>
<td>-7%</td>
</tr>
<tr>
<td>abs. avg.</td>
<td>-</td>
<td>-</td>
<td>11%</td>
<td>-</td>
<td>-</td>
<td>20%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Process Node [nm]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>14</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$\Psi \ [^\circ \text{C/W}]$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>povray</td>
<td>0.96</td>
<td>1.13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>TDP [W]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>povray</td>
<td>63</td>
<td>53</td>
</tr>
</tbody>
</table>
There isn’t a core scaling figure, just text..