

Automatic Core Specialization for AVX-512 Applications

Mathias Gottschlag, Peter Brantsch, Frank Bellosa | October 13, 2020

KARLSRUHE INSTITUTE OF TECHNOLOGY (KIT) - OPERATING SYSTEMS GROUP



Effects of AVX-512

- AVX-512: SIMD instructions for data parallelism
- AVX-512 speeds up Poly1305 MAC
 - ⇒ web server slowed down by 10% if AVX-512 is used
- AVX-512 can speed up machine learning by up to 2.2x
 - ⇒ Applications running in parallel run 10% slower
- **This talk: How to prevent this slowdown?**

$$\begin{array}{|c|} \hline a_0 \\ \hline a_1 \\ \hline a_2 \\ \hline a_3 \\ \hline \end{array} + \begin{array}{|c|} \hline b_0 \\ \hline b_1 \\ \hline b_2 \\ \hline b_3 \\ \hline \end{array} = \begin{array}{|c|} \hline c_0 \\ \hline c_1 \\ \hline c_2 \\ \hline c_3 \\ \hline \end{array}$$

Vlad Krasnov: *On the dangers of Intel's frequency scaling*. Cloudflare, Nov. 2017

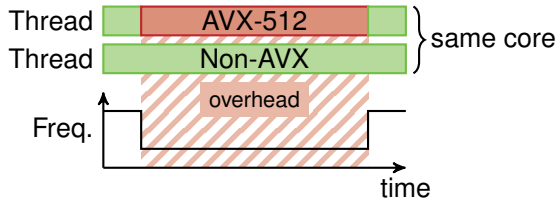
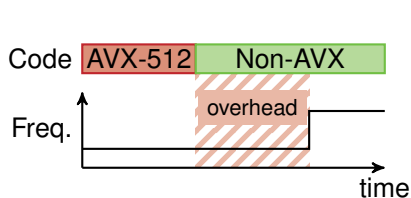
Aubrey Li: *Core scheduling: Fixing when fast instructions go slow*. LPC'19, Sep. 2019

- Complex SIMD instructions cause momentary high power consumption
 - ⇒ Result: High power *variability*
- Power is limited (heat, voltage drops)
 - ⇒ Different max. frequencies possible depending on instructions
- **Intel: Particularly low frequency for AVX-512 code!**

Intel Xeon Processor Scalable Family – Specification Update. June 2020

AVX-512 Overhead

- Frequency reduction affects non-AVX-512 code



- Local speedup, global slowdown?

⇒ **Solution to prevent slowdown at runtime**

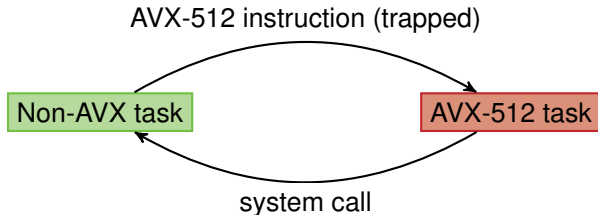
Core Specialization

- Idea: Spatial separation



- Result: Only “AVX-512 cores” slowed down
⇒ **Reduced frequency impact**

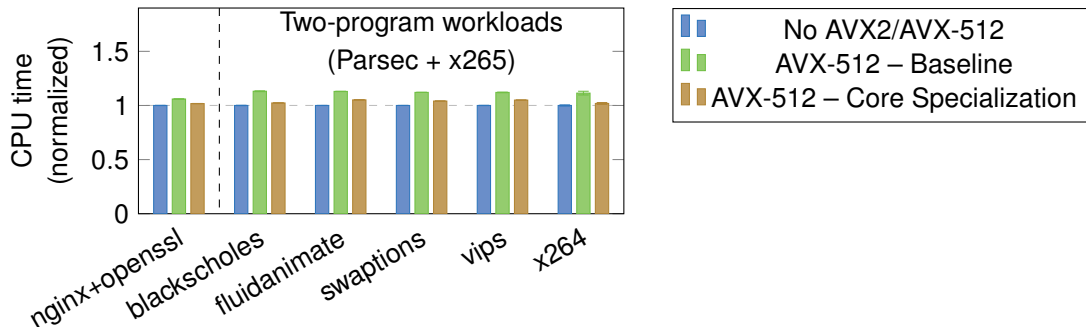
- Categorization of tasks



- Non-AVX tasks allowed on AVX-512 cores
 - But: Prioritize AVX-512 tasks
- More details in the paper.

Evaluation

- CPU time for heterogeneous workloads
- Usage of AVX-512 configurable



- Original: 11.3% overhead due to AVX-512
- **Our approach: 3.4%**

- Traps detect *all* 512-bit register accesses
 - Impossible to detect “energy-intensive” instructions
 - Better hardware/software interface?
 - Missing: NUMA support
 - Prevent migration between NUMA domains
 - Missing: Automatic allocation of AVX-512 cores
 - Number of cores based on load
- ⇒ Future work

- AVX-512 slows other code down
 - 10% overhead reported for several scenarios
 - Impact hard to predict
- Contribution: Scheduler modification to reduce slowdown
 - Core specialization
 - Intercept AVX-512 instructions
 - Restrict AVX-512 code to *AVX-512 cores*
- Evaluation: Slowdown reduced to 3.4% (was: 11.3%)