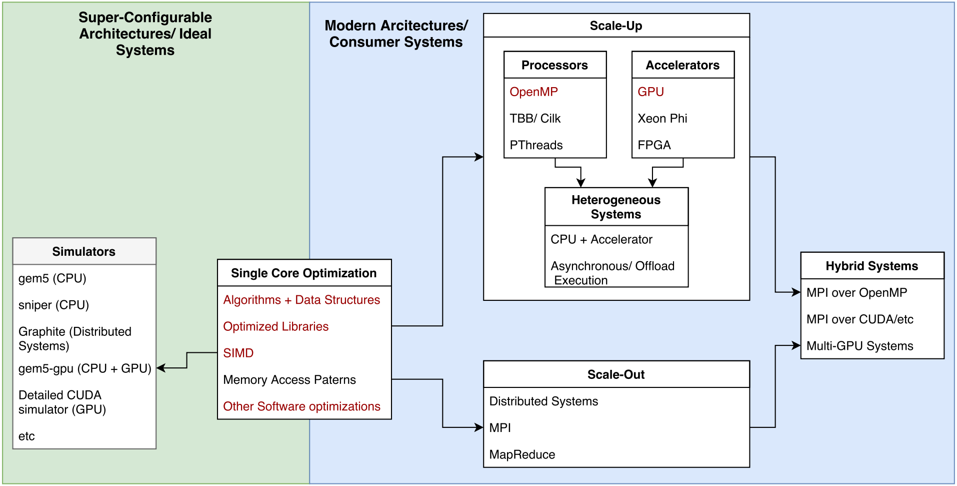
NTUA/CERN PhD Meeting

Konstantinos Iliakis

December 4, 2017

Table of Contents

PhD Work-plan



Go to the proposed PhD plan.

Top-Down Analysis

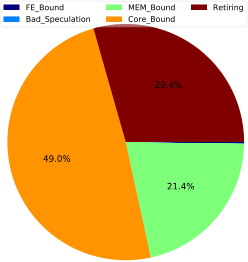
A Top-Down Method for Performance Analysis and Counters Architecture [1]

- A practical method to quickly identify bottlenecks.
- Divides the total cycles into 4 main categories:
  1. Front End
  2. Bad Speculation
  3. Back End ($\rightarrow$ Memory or Core)
  4. Retiring
- The method has been adopted by VTune ('general-exploration' analysis)

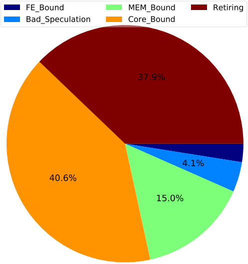
---

[1] Yasin, Ahmad. "A top-down method for performance analysis and counters architecture." Performance Analysis of Systems and Software (ISPASS), 2014 IEEE International Symposium on. IEEE, 2014.

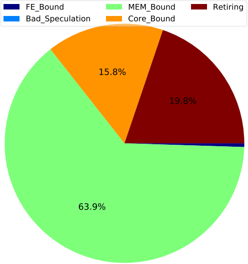
Top-Down Analysis for BLonD (I)



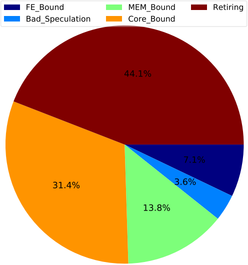
kick CPI:0.663



histogram CPI:0.547

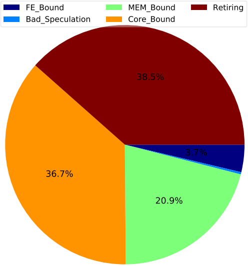


drift CPI:0.888

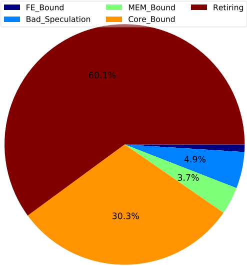


interp-kick CPI:0.424

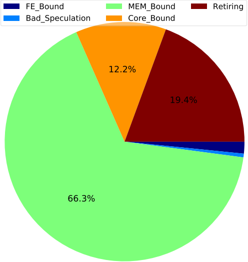
Top-Down Analysis for BLonD (II)



convolution CPI:0.664



synchrotron-radiation CPI:0.322



fft-convolution CPI:1.054

Top-Down Analysis for BLonD (III)

Remarks

- 5 core bound benchmarks, 2 memory bound
- 3 with a bad speculation portion (due to branch misprediction)
- Core bound:
  1. Pressure on an execution port that serves a specif uop [1]
  2. Data dependencies
- System: Intel i7-6700 (Due to a problem with the Haswell platform)

[1] https://en.wikichip.org/wiki/intel/microarchitectures/haswell

Roofline Model

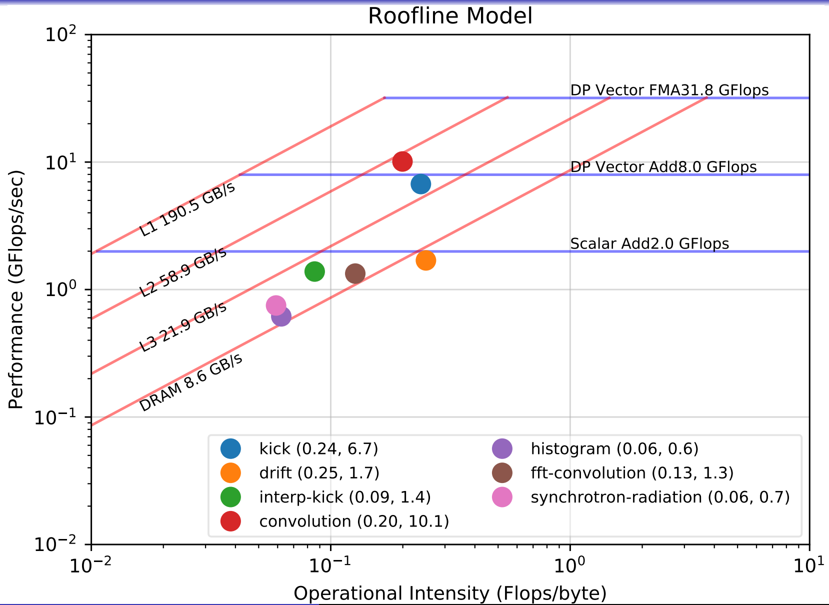
Roofline: An insightful Visual Performance Model for Multi-core Architectures [1]

- A simple visual model that offers insight on the programs performance and limitations.
- The system's peak performance is defined by the memory BW and peak FLOPS $\rightarrow$ ceilings.
- A follow-up publication [2] that also considers the BW of the multiple cache levels, has been embedded in the Intel Advisor.

[1] Williams, Samuel, Andrew Waterman, and David Patterson. "Roofline: an insightful visual performance model for multicore architectures." Communications of the ACM 52.4 (2009): 65-76.

[2] Ilic, Aleksandar, Frederico Pratas, and Leonel Sousa. "Cache-aware roofline model: Upgrading the loft." IEEE Computer Architecture Letters 13.1 (2014): 21-24.

Roofline Model for BLonD (I)



Roofline Model for BLonD (II)

Remarks

- System: Intel Xeon E5-2683 v3 (Haswell) 2x14cores
- Ideally all benchmarks should be at the L2 ceiling (dataset doesn't fit in the L2 cache) or DP Vector(Vectorized)/ Scalar (Not-vectorized) ceiling.
- Convolution almost reached the L2 ceiling.
- Combination of `kick()` or `interp_kick()` with `drift()` would increase OI and Performance.

Optimization Techniques Applied on BLonD

1. C++ Extensions (Main code in python).
2. Vectorized fast math library (for sin, cos, exp etc) [1].
3. Code restructuring to assist auto-vectorization.
4. Use of the Intel MKL library.
5. Multi-threading with OpenMP.
6. Loop tiling for vectorization, cache locality.
7. GPU versions of the core kernels, evaluation of multiple frameworks, e.g. OpenACC, CUDA, Thrust, PyCUDA.
8. Top-Down analysis to characterize the code → define bottlenecks.
9. Roofline model to evaluate the performance of the code.

[1] https://github.com/drbenmorgan/vdt

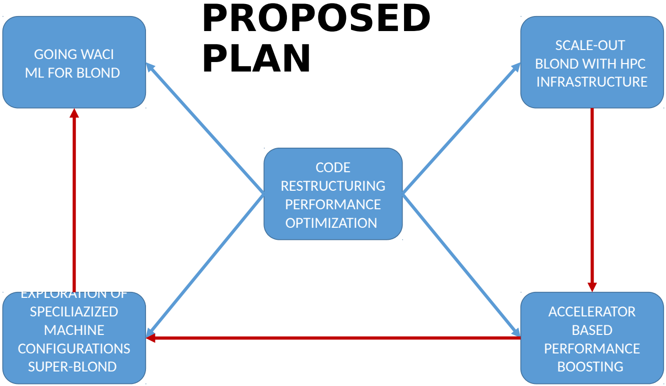
Other Issues

- ICAP18 (IC on Computational Accelerator Physics) [1].
- Set-back due to 'user permissions' problems on the available systems.
- Studying Papers (Micro-architecture simulators, performance profiling and modeling, scale-out/ accelerators/ HPC for scientific codes).

[1] http://www.icap18.org/

Thank you for your attention

Proposed PhD Plan



Back to presentation