A First-Principles Approach to Performance and Power Models for Contemporary Multi- and Many-Core Processors

Friedrich-Alexander-University Erlangen-Nuremberg, Computer Architecture
Johannes Hofmann <johannes.hofmann@fau.de>

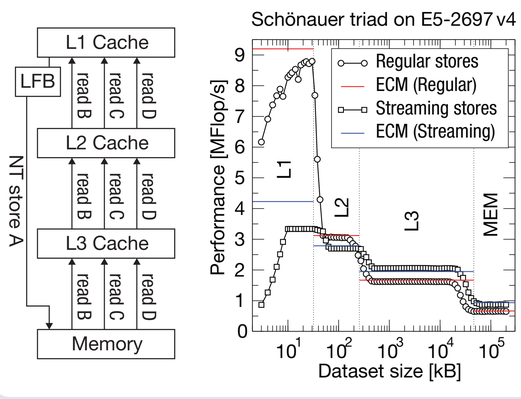
Vision: Understand performance, power, and energy properties of modern processors and recommend best-practices

Improve Execution-Cache-Memory (ECM) Performance Model [1-4]

- Separate general principles and microarchitecture-dependent behavior into *general ECM model* and *ECM machine models*
- Model additional hardware designs
  - Write-through caches
  - Streaming stores
  - Cluster on Die mode
  - Separate frequency domains
  - Victim caches
  - Partially and fully overlaping data transfers
  - Dynamic Uncore frequency
- Devise machine models for
  - Intel Haswell- and Broadwell-EP
  - IBM POWER8
  - Intel Skylake-SP
  - AMD Zen (Ryzen, Epyc)
- Reduce ECM model error near the saturation point by an order of magnitude

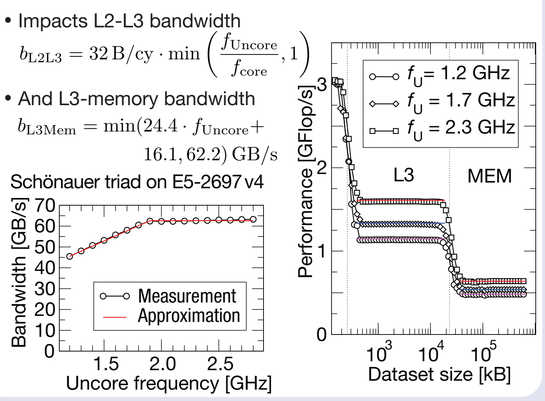
selected examples

Regular vs. streaming stores

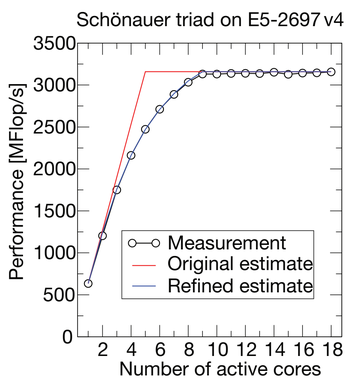


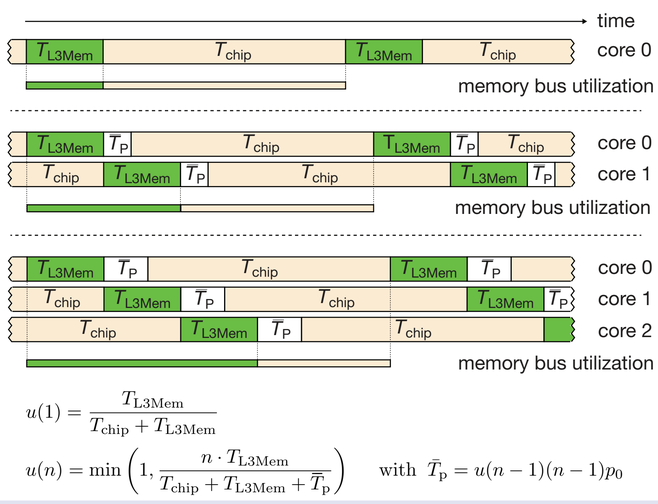
Variable Uncore frequency

- Impacts L2-L3 bandwidth
$$b_{\mathrm{L2L3}} = 32\,\mathrm{B/cy} \cdot \min\left(\frac{f_{\mathrm{Uncore}}}{f_{\mathrm{core}}}, 1\right)$$
- And L3-memory bandwidth
$$b_{\mathrm{L3Mem}} = \min(24.4 \cdot f_{\mathrm{Uncore}} + 16.1, 62.2)\,\mathrm{GB/s}$$



Refined multi-core estimate





$$u(1) = \frac{T_{\mathrm{L3Mem}}}{T_{\mathrm{chip}} + T_{\mathrm{L3Mem}}}$$

$$u(n) = \min\left(1, \frac{n \cdot T_{\mathrm{L3Mem}}}{T_{\mathrm{chip}} + T_{\mathrm{L3Mem}} + \bar{T}_{\mathrm{p}}}\right) \quad \text{with } \bar{T}_{\mathrm{p}} = u(n-1)(n-1)p_0$$

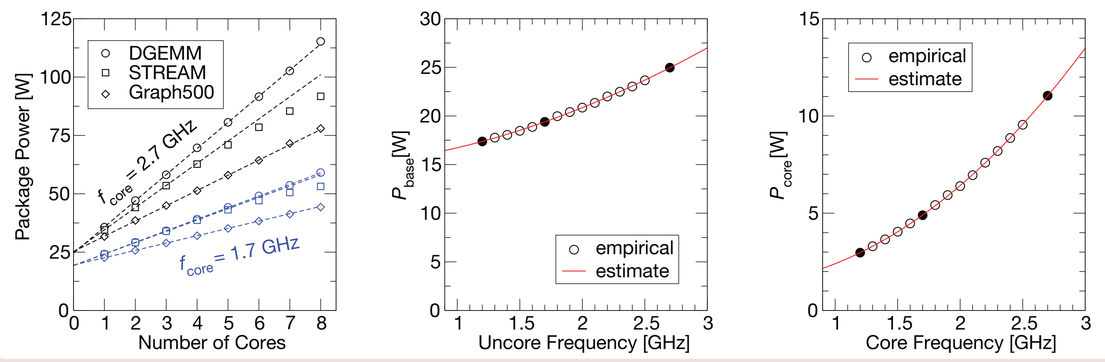
Quantitative Power Model [5]

Quantitative power estimate s.t. active core count, CPU core and Uncore frequencies for arbitrary scalable and saturating steady-state codes

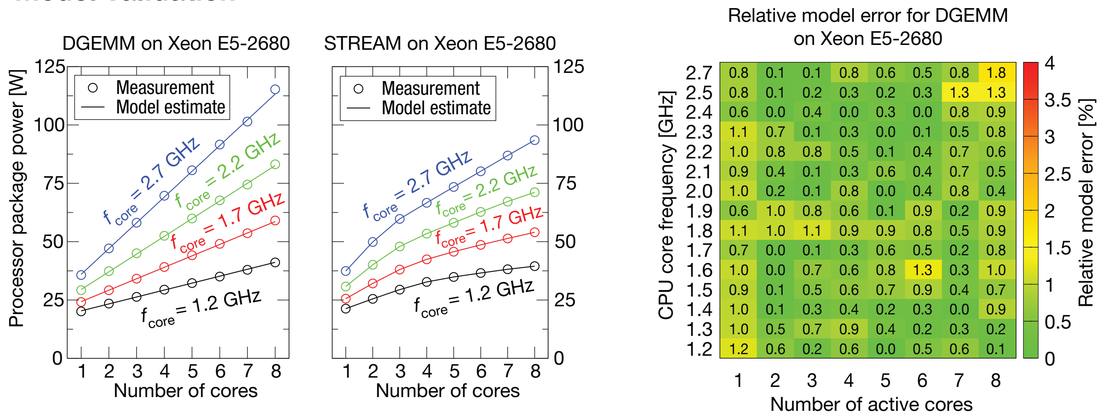
- Full chip power given by sum of baseline and aggregate core power
$$P_{\mathrm{chip}}(f_{\mathrm{Uncore}}, f_{\mathrm{core}}, \varepsilon_n) = P_{\mathrm{base}}(f_{\mathrm{Uncore}}) + nP_{\mathrm{core}}(f_{\mathrm{core}}, \varepsilon_n)$$
- Baseline power depends on processor's Uncore frequency
$$P_{\mathrm{base}}(f_{\mathrm{Uncore}}) = p_0^{\mathrm{base}} + p_1^{\mathrm{base}} f_{\mathrm{Uncore}} + p_2^{\mathrm{base}} f_{\mathrm{Uncore}}^2$$
- Per-core power depends on CPU core frequency, code scalability
$$P_{\mathrm{core}}(f_{\mathrm{core}}, \varepsilon_n) = \left(p_0^{\mathrm{core}} + p_1^{\mathrm{core}} f_{\mathrm{core}} + p_2^{\mathrm{core}} f_{\mathrm{core}}^2\right) \cdot \varepsilon_n^{\alpha}$$
- Validated on Intel Sandy and Ivy Bridge, Haswell, Broadwell, AMD Zen

Model setup

1. Derive baseline power s.t. Uncore freq. by linear extrapolation towards 0 cores
2. Fit baseline power parameters using the data obtained in (1)
3. Derive core power contribution from empirical data by subtracting baseline estimate (2), then fit core parameters



Model validation



Derived Energy Model [5]

An analytic energy model can be derived by combining the ECM model estimate Π_{ECM} and the power model estimate P_{chip}

$$E(f_{\mathrm{Uncore}}, f_{\mathrm{core}}, n) = \frac{P_{\mathrm{chip}}(f_{\mathrm{Uncore}}, f_{\mathrm{core}}, n, \varepsilon_n)}{\Pi_{\mathrm{ECM}}(f_{\mathrm{Uncore}}, f_{\mathrm{core}}, n)}$$

Analytic deductions

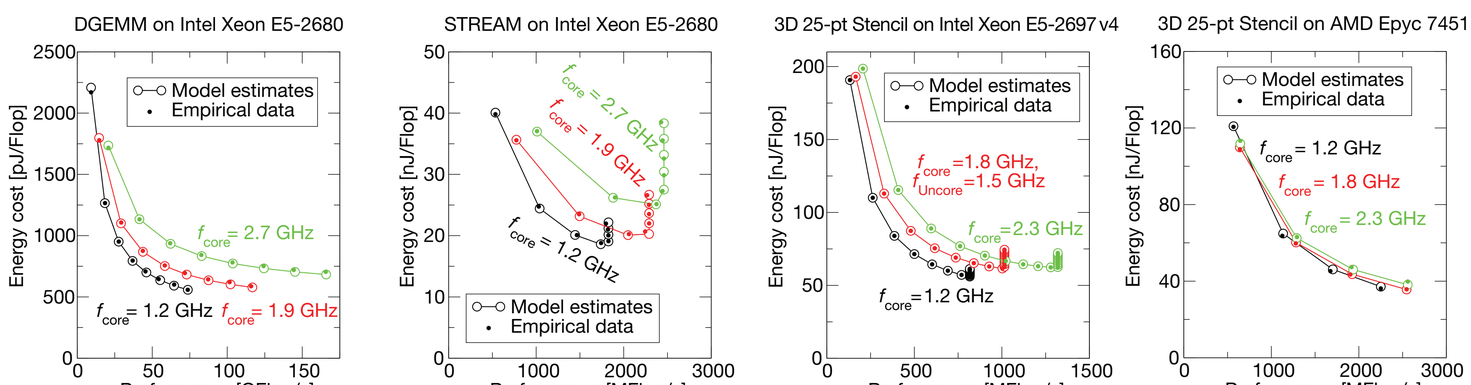
Minimum energy w.r.t. number of active cores *n*

- scalable codes: use all available cores
$$\frac{\partial E}{\partial n} = -\frac{P_{\mathrm{base}}(f_{\mathrm{Uncore}})}{-n^2 \cdot \Pi_{\mathrm{ECM}}(f_{\mathrm{core}}, f_{\mathrm{Uncore}})} < 0$$
- saturating codes: use number of cores n_s required to saturate bottleneck
$$E = \frac{P_{\mathrm{base}}(f_{\mathrm{Uncore}}) + n\varepsilon^{\alpha} \cdot P_{\mathrm{core}}(f_{\mathrm{core}}, 1)}{\Pi_{\mathrm{Sat}}}$$

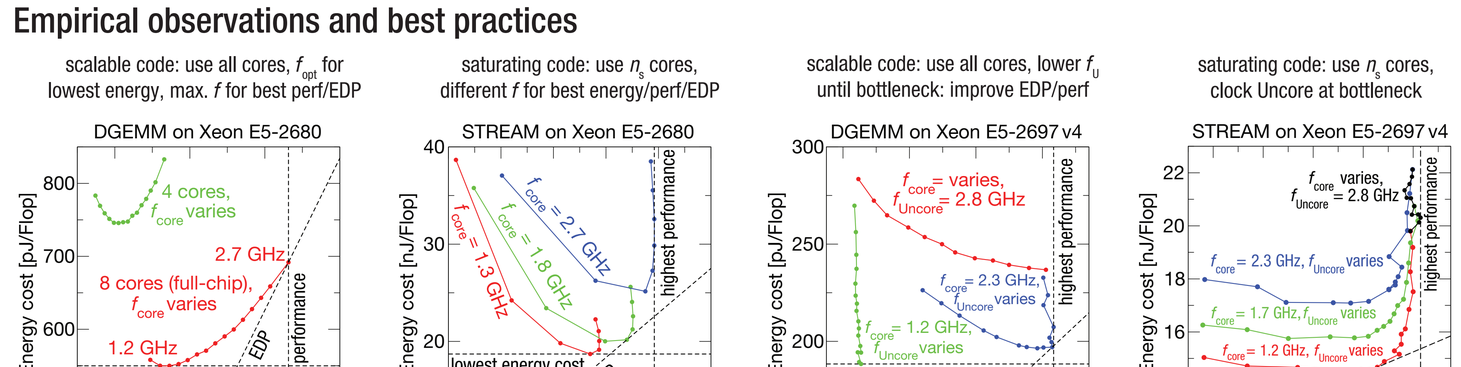
Minimum energy w.r.t. CPU core frequency f_{core}

- Optimum frequency subject to clock domains
$$f_{\mathrm{core}}^{\mathrm{opt}}(n) = \sqrt{\frac{p_0^{\mathrm{base}} + np_0^{\mathrm{core}}}{p_2^{\mathrm{base}} + np_2^{\mathrm{core}}}}$$

Model validation



Empirical observations and best practices



FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

References

[1] J. Hofmann, D. Fey, M. Riedmann, J. Eitzinger, G. Hager, and G. Wellein: Performance analysis of the Kahan-enhanced scalar product on current multi- and manycore processors. Concurrency and Computation: Practice and Experience, ISSN: 1532-0634

[2] J. Hofmann, D. Fey, J. Eitzinger, G. Hager, and G. Wellein: Analysis of Intel's Haswell Micro-architecture Using the ECM Model and Microbenchmarks Architecture of Computing Systems - ARCS 2016: 29th International Conference, Nuremberg, Germany, April 4-7, 2016

[3] J. Hofmann, D. Fey: An ECM-based energy-efficiency optimization approach for bandwidth-limited streaming kernels on recent Intel Xeon processors. 4th International Workshop on Energy Efficient Supercomputing, Salt Lake City, UT, USA, November 14, 2016

[4] J. Hofmann, G. Hager, G. Wellein, D. Fey: An analysis of core- and chip-level architectural features in four generations of Intel server processors. High Performance Computing: 32nd International Conference, ISC High Performance 2017, Frankfurt, Germany, June 18-22, 2017

[5] J. Hofmann, G. Hager, D. Fey: On the accuracy and usefulness of analytic energy models for contemporary multicore processors. Accepted for High Performance Computing: 33nd International Conference, ISC High Performance 2018, Frankfurt, Germany, June 24-28, 2018