Towards performance and power model for multi-core processors with DVFS

28th Workshop on Sustained Simulation Performance (WSSP 2018)

Dmitry Khabi (HLRS)

Content

Energy-Efficiency in High Performance Computing

Performance Models

Power Dissipation Models

Outlook

Energy-Efficiency in High Performance Computing

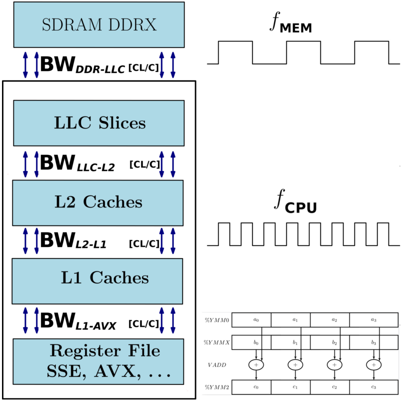
Reduction in consumed joules while increasing computational job throughput at acceptable additional invest costs

The optimization strategies are particularly refers to:

- HPC policies that govern the use of HPC resources.
- Job sceduling.
- Software optimization.
- HPC Infrastructure such as cooling, power supplies etc.
- Performance and power dissipation of hardware: compute nodes, processors, memory, and network.

This talk is about a developed method for evaluating and comparing hardware in terms of energy efficiency. In the foreground is the dependency between the power dissipation of the processors and the number of active cores and CPU frequencies.

State of the art: Execution-Cache-Memory (ECM) model (Hager et al. 2012)



- The bandwidths in CL/c between the levels of the memory hierarchy are the input parameters.
- The data transfers between cache levels occur in packets of one cache line (CL).
- In any clock cycle (C), the L1 cache can either load/evict CL from/to L2 or communicate with registers (same for all memory hierarchy levels).

Execution-Cache-Memory Performance Modell (ECM)

Eq. (1) describes the execution time on one CPU core with three cache levels and main memory (e.g. Haswell E5-2680v3)

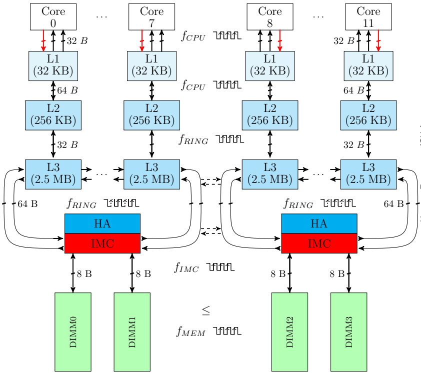
$$
\begin{aligned}
T_{ECM,MEM} \quad & = \max(T_{OL}, T_{nOL} + T_{L1\leftrightarrow L2} + T_{L2\leftrightarrow L3} + T_{L3\leftrightarrow MEM}); \\
\\
T_{ECM,MEM} \quad & \text{-Modeled execution time on one CPU core;} \\
T_{OL} \quad & \text{-Execution time in pipeline with overlapping to data transfer;} \\
T_{nOL} \quad & \text{-Execution time in pipeline without overlapping to data transfer;} \\
T_{L1\leftrightarrow L2} \quad & \text{-Data transer time between L1 und L2;} \\
T_{L2\leftrightarrow L3} \quad & \text{-Data transfer time between L2 and L3;} \\
T_{L3\leftrightarrow MEM} \quad & \text{-Data transfer time between L3 and MEM;}
\end{aligned}
\tag{1}
$$

The results show a very good agreement of the ECM model with the measurements if f_{CPU} is a base CPU frequency if data fits cache.

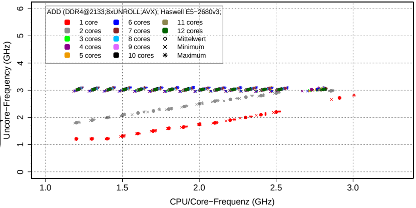
In case of holding the data in memory, the model has noticeable deviations from the performance measurements if the CPU frequency and the number of cores are varied.

The ECM model was refinement with additional components (Hofmann, Hager, and Fey 2018).

Haswell Hardware Architecture



- The cores, IMC, L3 cache and memory are clocked differently.



Dependencies between the core's and the L3 Ring interconnect's frequencies for a stream operation (memory bound), if the data fits in cache.

Evaluation Kernels for Haswell with 12 cores (E5-2680v3)

Listing 1: $A[i] = B[i] + C[i]$ (OpenMP).

```
#pragma omp parallel
{
  const long length=length_per_thread;
  const long tests = num_tests;
  double* __restrict loc_a=a[thread_num];
  ...
  for(jj=0; jj<length;jj+=32){
    __m256d C=_mm256_load_pd(loc_a+ii);
    __m256d B=_mm256_load_pd(loc_b+ii);
    __m256d A=_mm256_add_pd(A,B);
    ...
    __m256d C8=_mm256_load_pd(loc_c+ii+28);
    __m256d B8=_mm256_load_pd(loc_b+ii+28);
    __m256d A8=_mm256_add_pd(C8,B8);

    _mm_store_pd(loc_a+ii, A)
    ...
    _mm_store_pd(loc_a+ii+28, A8)
  }
  ...
}
```

Streaming data access, variable length
($L \in L1, L2, L3, RAM$)

Listing 2: MVM in CSR (PETsC[Balay et al. 2017]).

```
double mat_value, vec_value, tmp_value;
int64_t col_idx;
for(ii=0; ii<length; ii++)
{
   const int64_t first_col=xadj[ii];
   const int64_t next_first_col=xadj[ii+1];
   tmp_value=0.0;
   for(jj=first_col; jj<next_first_col; jj++)
   {
      col_idx=adjncy[jj];
      mat_value=aa[jj];
      vec_value=xx[col_idx];
      tmp_value+=mat_value*vec_value;
    }
    yy[ii]=tmp_value;
}
```

Indirect data access, 3D-Poisson equation
27-point stencil 128 × 128 × 128 (MEM)

Extension of ECM Model - CPU Data Transfer Model (DTM)

In contrast to the EMC model, the overlapping of the data transfer between the uncore components and execution in the core (including local cache) is allowed.
The memory and the L3 caches are connected over the Internal Memory Controller (IMC).

$$
\boxed{\begin{array}{ll}
T^{DTM} & = \max(T^{OL}_{L2\leftrightarrow AVX}, T^{nOL}_{L2\leftrightarrow AVX} + T_{L3\leftrightarrow L2} + T_{L3\leftrightarrow IMC} + T_{MEM\leftrightarrow IMC}); \\
T_{DTM,MEM} & \textit{- Modeled execution time on CPU;} \\
T^{OL}_{L2\leftrightarrow AVX} & \textit{- Execution time in the processor cores (pipeline, L1, L2)} \\
 & \qquad \textit{with overlapping data transfer between} \\
 & \qquad \textit{the Uncore components (main memory, IMC and L3);} \\
T^{nOL}_{L2\leftrightarrow AVX} & \textit{- Execution time in the processor cores without overlapping;} \\
T_{A\leftrightarrow B} & \textit{- Execution time of the data transfer between the components} \\
 & \qquad \textit{A und B without overlapping;}
\end{array}} \tag{2}
$$

Performance Metrics for CPU Data Transfer Model

Transition from time to performance:

$$T_{A\leftrightarrow B} = \frac{Q_{A\leftrightarrow B}}{Bw_{A\leftrightarrow B}} \quad [\mathrm{s}]$$

$Q_{A\leftrightarrow B}$-bytes to transfer between the components A and B (e.g. L1 and AVX)

$Bw_{A\leftrightarrow B}$-theoretical/measured bandwidth of the interface between A and B

Time to transfer between hierarchy levels of memory are to be summed:

$$T_{A\leftrightarrow C} = T_{A\leftrightarrow B} + T_{C\leftrightarrow B} \quad \Leftrightarrow \quad \frac{Q_{C\leftrightarrow B}}{Bw_{C\leftrightarrow B}} = \frac{Q_{A\leftrightarrow B}}{Bw_{A\leftrightarrow B}} + \frac{Q_{C\leftrightarrow B}}{Bw_{C\leftrightarrow B}};$$

Kernel ADD (per Flop/+, (Intel 2016)):

$$Q_{L1\leftrightarrow AVX} = 24\mathrm{B}\ ; Q_{L3\leftrightarrow L2} = 32\mathrm{B} \ldots \Leftrightarrow \quad \mathrm{Q}_{L1\leftrightarrow AVX} = 2/3 \times Q_{L2\leftrightarrow L1}$$

Performance model for bandwidth between L2 and AVX registers (ADD):

$$Bw_{L2\leftrightarrow AVX} = (\tfrac{2}{3} \times \frac{1}{Bw_{L1\leftrightarrow AVX}} + \frac{1}{Bw_{L2\leftrightarrow L1}})^{-1}[\mathrm{B/s}];$$

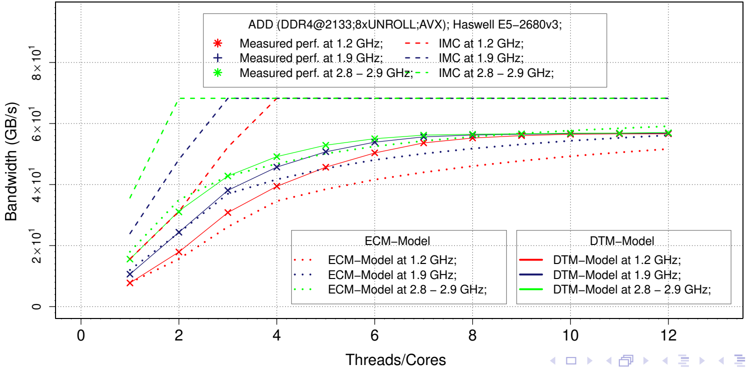
Data Transfer Model for Memory (DTM)

The DTM-Modell allows the overlapping, while transfer the data between IMC and memory:

$$
\begin{aligned}
\frac{1}{Bw_{DTMMEM\leftrightarrow AVX}} &= \frac{1}{Bw_{MEM\leftrightarrow IMC}} + \frac{1}{Bw_{IMC\leftrightarrow L3}} + \frac{1}{Bw_{L3\leftrightarrow L2}} + \frac{K_{Overlapping}}{Bw_{MESSL2\leftrightarrow AVX}};\\
Bw_{MEM\leftrightarrow IMC} &= 8 \times 4 \times f_{IMC};\\
f_{IMC} &= \min(N_{mem\_channel} \times f_{MEM}, Bw_{L3}[B/s]/(8[B] \times N_{mem\_channel}))\\
Bw_{IMC\leftrightarrow L3} &= 64 \times \frac{f_{RING}}{2} \times p;\\
Bw_{L3\leftrightarrow L2} &= 32 \times f_{RING} \times p;\\
&\quad \text{Data transfers between IMC, L3 and L2 are well balanced.}\\
Bw_{MESSL2\leftrightarrow AVX} &= 32 \times Perf(f_{CPU}, p); \quad Perf(f_{CPU}, p) = (\beta_{0,L2} \times p \times f_{CPU});\\
K_{Overlapping}: &\quad 0 \leq K_{Overlapping} \leq 1.0 \wedge min(Bw_{DTMMEM\leftrightarrow AVX} - Bw_{MESSMEM\leftrightarrow AVX});
\end{aligned}
\tag{3}
$$

Comparison between ECM model, DTM model and measurement (Haswell E5-26980 v3)

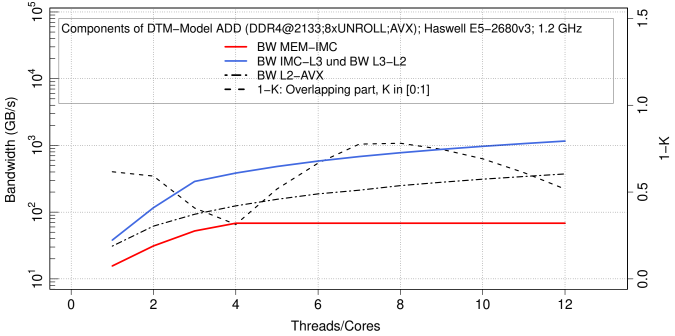
The dashed lines show the bandwidth between IMC and memory. The dots show the measurements. The solid lines show and the DTM model. The dots line shows the ECM model.



Overlapping coefficient and components of DTM-Model 1/3

$f_{CPU} = 1.2GHz$

$$\frac{1}{Bw_{DTMMEM\leftrightarrow AVX}} = \frac{1}{Bw_{MEM\leftrightarrow IMC}} + \frac{1}{Bw_{IMC\leftrightarrow L3}} + \frac{1}{Bw_{L3\leftrightarrow L2}} + \frac{K_{Overlapping}}{Bw_{MESSL2\leftrightarrow AVX}};$$

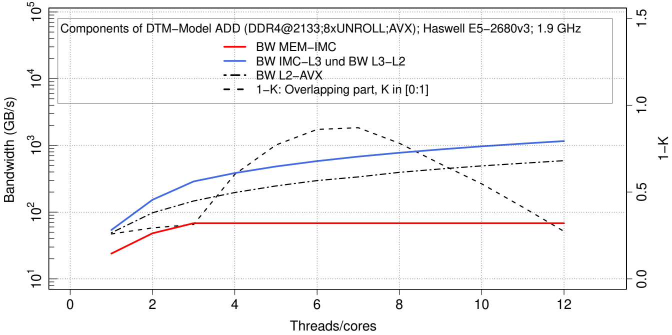


7-8 Cores:75% of T_{L2AVX} overlapped

Overlapping coefficient and components of DTM-Model 2/3

$f_{CPU} = 1.9GHz$

$$\frac{1}{Bw_{DTMMEM \leftrightarrow AVX}} = \frac{1}{Bw_{MEM \leftrightarrow IMC}} + \frac{1}{Bw_{IMC \leftrightarrow L3}} + \frac{1}{Bw_{L3 \leftrightarrow L2}} + \frac{K_{Overlapping}}{Bw_{MESSL2 \leftrightarrow AVX}};$$

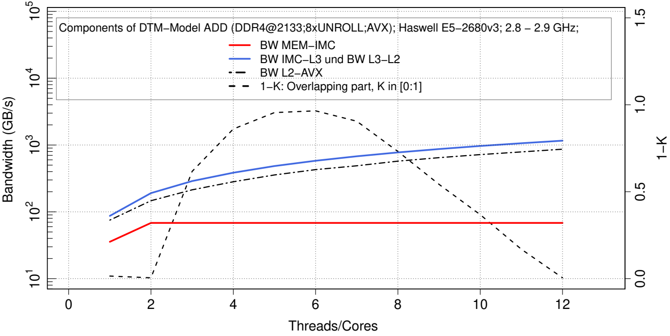


6-7 Cores:80% of T_{L2AVX} overlapped

Overlapping coefficient and components of DTM-Model 3/3

$f_{CPU} = 2.8 - 2.9GHz$

$$\frac{1}{Bw_{DTMMEM\leftrightarrow AVX}} = \frac{1}{Bw_{MEM\leftrightarrow IMC}} + \frac{1}{Bw_{IMC\leftrightarrow L3}} + \frac{1}{Bw_{L3\leftrightarrow L2}} + \frac{K_{Overlapping}}{Bw_{MESSL2\leftrightarrow AVX}};$$



4-5 Cores:95% of T_{L2AVX} overlapped („optimal“ CPU configuration)

State of the art: Approximation of Power Dissipation (CPU and RAM)

$P_{cmos} = P_{static} + P_{dynamic} + P_{shortcircuit}$ (Chandrakasan, Sheng, and Brodersen 1995);

- $P_{static} = I_{leakage} * V_{dd}$ - Transistors conduct a small amount of current even when they are turned off.
- $P_{dynamic} = C_L * V_{dd}^2 * f_{clk}$ - Is caused by charge and discharge of semiconductor devices.
- $P_{shortcircuit} = I_{sc} * V_{dd}$- Short circuit aries when both the NMOS and PMOS transistors simultaneously active.

V_{dd} - Supply voltage (in some cases known);
C_L - Loading capacitance (is unknown);
f_{clk} - Clock frequency (CPU is simultaneously clocked at multiple frequencies);

ECM: The dynamic power dissipation is a quadratic polynomial in the clock frequency

- $P_{CPU} = \alpha_0 + (\alpha_1 \times f_{clk} + \alpha_2 \times {f_{clk}}^2) \times p$
- Components of CPU doesn't work with the same frequency f_{clk}.
- Is the quadratic polynomial suitable if one also want to consider the power dissipation of an additional hardware (memory or even an entire compute node)?

Approximation of Power Dissipation (Khabi and Küster 2013)

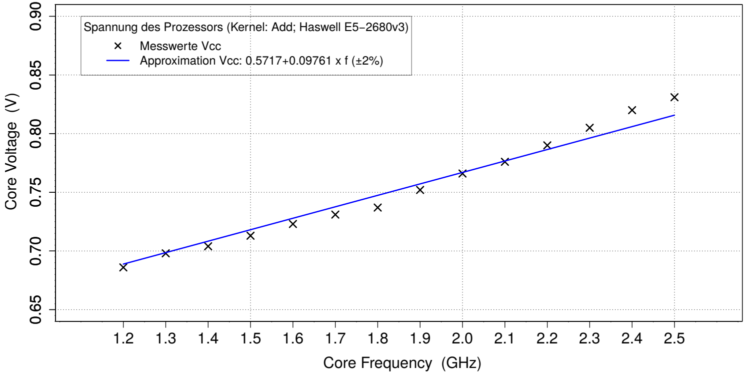
Power of a kernel operation depends on

- Hardware (CPU, SDRAM);
- Data sizes ($L \in L1, L2, L3, RAM$);
- Number of active threads (p);
- Core frequency (f_{CPU} : $P(f = 0) \neq 0$ - Static power);

QR algorithm is used to find the coefficients α, λ of best approximation for the power dissipation in the 2-norm.

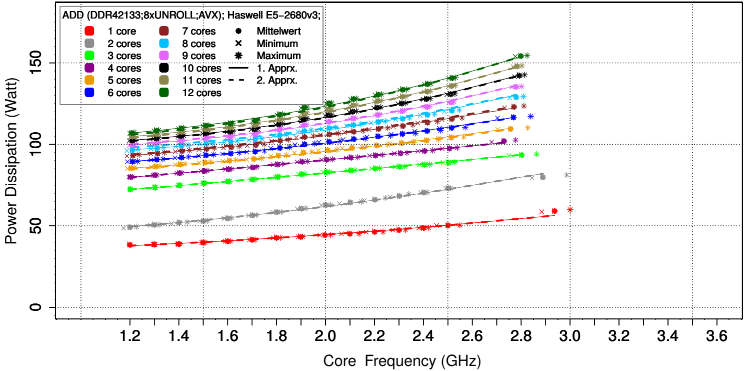
$$
\begin{aligned}
P(f_{CPU}, p)_l &= (\alpha_{0,0,l} + \alpha_{0,1,l} \times p) + (\alpha_{1,0,l} + \alpha_{1,1,l} \times p) \times {f_{CPU}}^{\lambda_l}; \\
\epsilon_{L_2} &= \sqrt{\sum_{p=1}^{n_p} \sum_{i=0}^{m_f-1} (P_{f_i,l,p_i} - P(f_i, p)_l)^2}; \\
p : &\quad \textit{Number of active threads}; \\
f_i : &\quad \textit{P-States (possible frequencies)}\ f_i \in \{f_0, f_1, .., f_{m_f-1}\}; \\
m_f : &\quad \textit{Number of P-States}; \\
n_p : &\quad \textit{Max. number of cores}\ ; \\
P_{f_i,l,p_i} : &\quad \textit{measured power dissipation}\ f = f_i \wedge p = p_i;
\end{aligned}
\tag{4}
$$

Dynamic voltage and frequency scaling: relationship between V_{cc} and f_{core}



- During the execution of the kernel operation Add on 12 cores of *Haswell* (E5-2680v3);

Power Dissipation of Kernel Operation ADD (SDRAM)



- During the execution of the kernel operation Add on *Haswell* (E5-2680v3);
- The data fits in main memory;

Power Approximation of Kernel Operation ADD (SDRAM)

Add on Haswell (E5-2680v3):

$$
\begin{aligned}
P(f, p \in (1, 2))_{ADD,RAM} &= (25.46588 + 8.26171 \times p) + (0.49423 + 2.39527 \times p) \times f^{1.91};\\
P(f, p \in (3.., 12))_{ADD,RAM} &= (65.52072 + 2.02131 \times p) + (2.02131 + 0.32525 \times p) \times f^{2.11};\\
&\epsilon_{rel} \leq 0.1(\text{with Turbo } 0.077);
\end{aligned}
$$

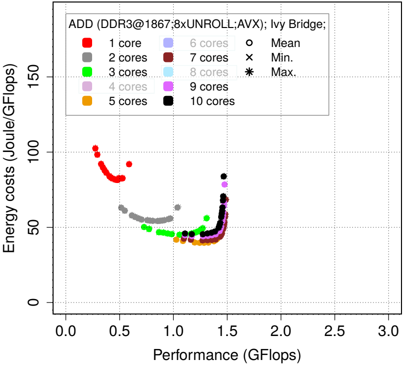
Add on Hazel Hen (2xHaswell):

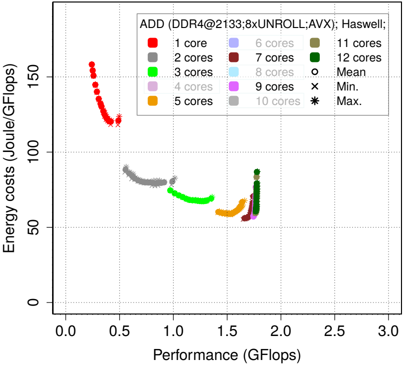
$$
\begin{aligned}
P(f, p \in (2 \times 1, 2 \times 2))^{ADD}_{RAM} &= (63.74772 + 7.19333 \times p) + (2.01264 + 3.51754 \times p) \times f^{1.77};\\
P(f, p \in (2 \times 3, .., 2 \times 12)^{ADD}_{RAM} &= (152.047746 + 3.19220 \times p) + (1.931612 + 0.242121 \times p) \times f^{2.65};\\
&\epsilon_{rel} \leq 0.057;
\end{aligned}
$$

PETsC-CG on Hazel Hen (2xHaswell):

$$
\begin{aligned}
P(f, p \in (2 \times 1, 2 \times 2))^{CG}_{128x128x128} &= (63.68315 + 18.3396 \times p) + (1.2453 + 1.74137 \times p) \times f^{2.77};\\
P(f, p \in (2 \times 3, ..., 2 \times 12))^{CG}_{128x128x128} &= (117.2081 + 9.939 \times p) + (5.52700 + 0.212965 \times p) \times f^{2.45};\\
&\epsilon_{rel} \leq 0.05;
\end{aligned}
\tag{5}
$$

Energy Costs of Kernel Operation ADD on Ivy-Bridge and Haswell (SDRAM)





$\forall p$: The higher frequencies are expensive and provides a minor performance benefit.

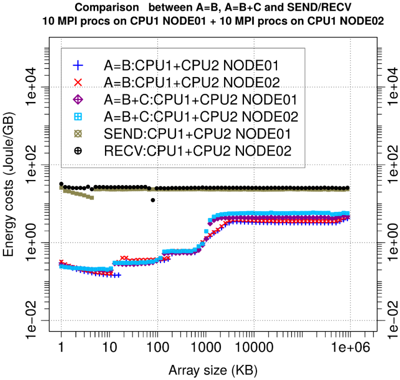
Ivy Bridge: The computation with less active cores is faster than with all cores.

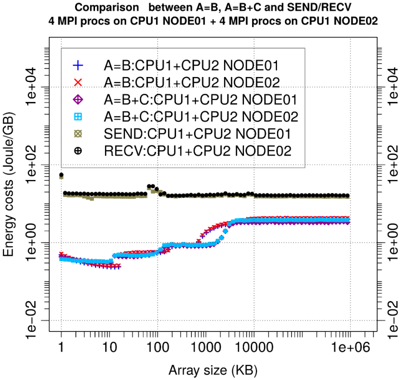
Haswell: consumes more power than Ivy Bridge.

Note: We consider CPU **and** memory power consumption.

Outlook: Comparison between A=B, A=B+C and MPI SEND/RECV

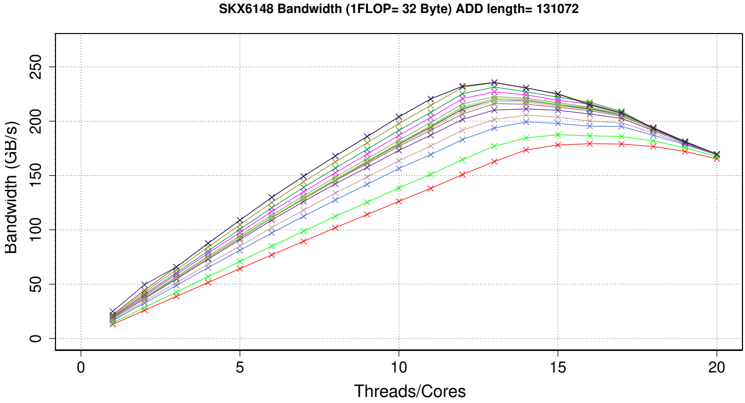
Results obtained on two compute node with 2 × *Ivy-Bridge* in Turbo Mode.
The log scale hides the differences between the energy efficiency of the processing with 20 and 8 processes. However, the diagrams show a significant gap between the local and distributed data-processing.





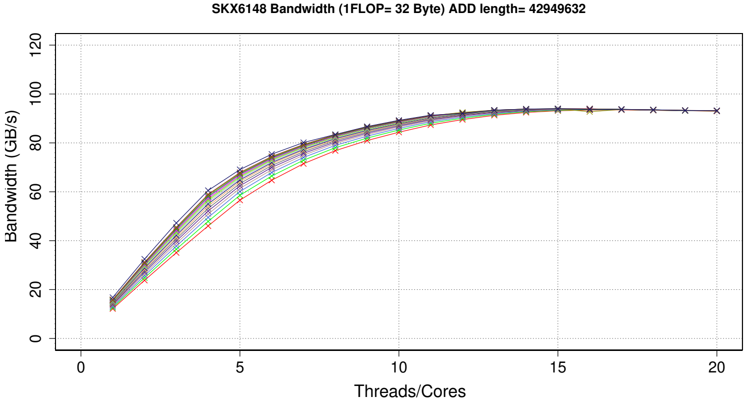
Outlook: Skylake

Kernel ADD, data fits in L3-Cache (mesh interconnect); Measured together with Holger Berger (HPC Division NEC Deutschland GmbH)



Outlook: Skylake

Kernel ADD, data fits in main memory (6 channels, SDRAM-DDR4); Measured together with Holger Berger (HPC Division NEC Deutschland GmbH)



- Balay, Satish et al. (2017). *PETSc Users Manual*. Tech. rep. ANL-95/11 - Revision 3.8. Argonne National Laboratory. URL: http://www.mcs.anl.gov/petsc.
- Chandrakasan, Anantha P., Samuel Sheng, and Robert W. Brodersen (1995). “Low Power CMOS Digital Design”. In: *IEEE JOURNAL OF SOLID STATE CIRCUITS* 27, pp. 473–484.
- Hager, Georg et al. (2012). “Exploring performance and power properties of modern multicore chips via simple machine models”. In: *CoRR* abs/1208.2908. URL: http://arxiv.org/abs/1208.2908.
- Hofmann, Johannes, Georg Hager, and Dietmar Fey (2018). “On the accuracy and usefulness of analytic energy models for contemporary multicore processors”. In: *CoRR* abs/1803.01618. arXiv: 1803.01618. URL: http://arxiv.org/abs/1803.01618.
- Intel (2016). “Intel 64 and IA-32 Architectures Optimization Reference Manual”. In: chap. Load and Store Operation Overview, pp. 2–24. URL: http://www.intel.com/content/www/us/en/architecture-and-technology/64-ia-32-architectures-optimization-manual.html.
- Khabi, Dmitry and Uwe Küster (2013). “Power Consumption of Kernel Operations”. In: *Sustained Simulation Performance 2013*. Springer, pp. 27–45.

The described research activities are partly funded by the Federal Ministry for Economic Affairs and Energy within the BEAM-ME project (ID: 03ET4023A-F).

The End