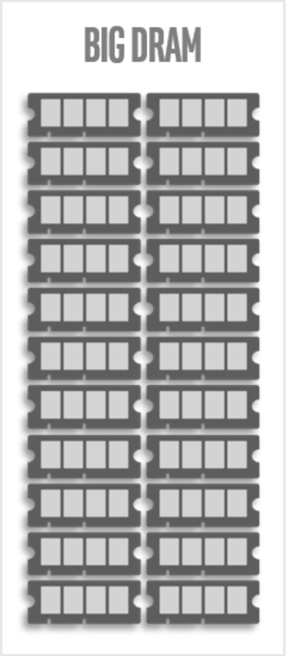
Evaluation of Intel Memory Drive Technology Performance for Scientific Applications

Vladimir Mironov, Andrey Kudryavtsev, Yuri Alexeev, Alexander Moskovsky

Introducing Intel® Memory Drive Technology

- Use Intel® Optane™ SSD DC P4800X **transparently as memory**
- Grow **beyond system DRAM capacity**, or **replace high-capacity DIMMs** for lower-cost alternative, with **similar performance**
- **Leverage storage**-class memory today!
  - **No change to software** stack: unmodified Linux* OS, applications, and programming
  - **No change to hardware**: runs bare-metal, loaded before OS from BIOS or UEFI
- **Aggregated single volatile memory pool**

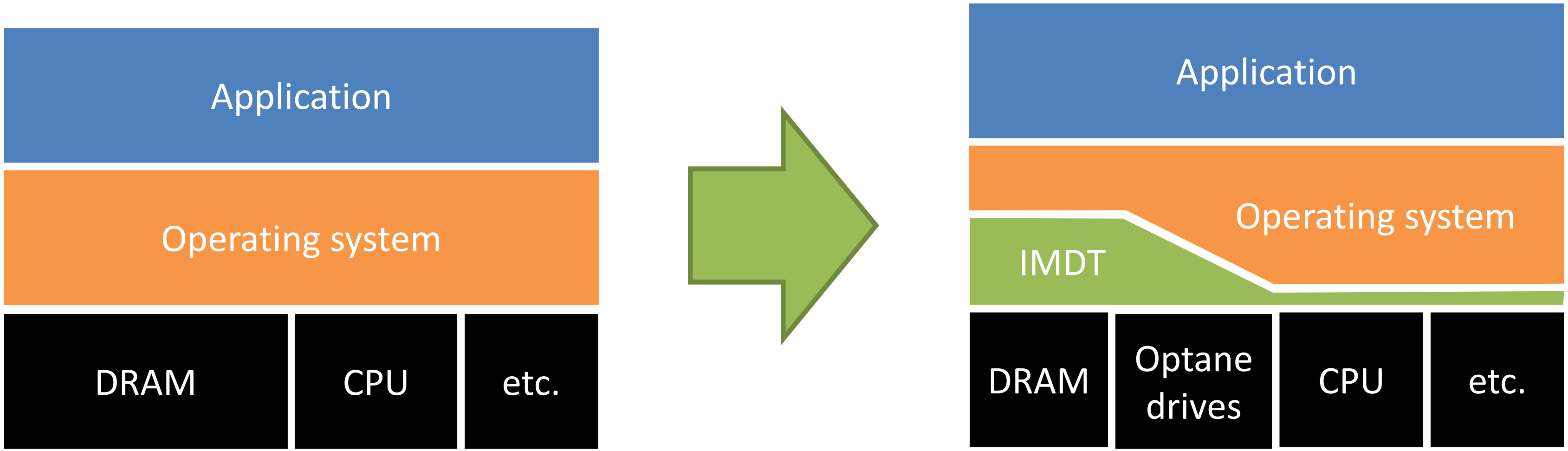


OLD

NEW

*Other names and brands may be claimed as the property of others

How Intel® Memory Drive Technology works



- IMDT operates on hypervisor level
- Requires CPU with virtualization support
- Supports any Linux distribution

- Store unused memory pages to Optane drives and load to DRAM when needed
- **Analyzes memory access patterns to prefetch data on local to CPU DRAM slot**

When to use and when not to use Intel® Memory Drive Technology

✓ Your application is designed to use very large amount of memory

- Benefits from the large memory pool
- Virtually no performance decrease on benchmarks with high arithmetic intensity

✓ Your application does not handle memory-locality/NUMA well

- Benefits from the intelligent control of NUMA memory access

✗ Your application is bound by the memory bandwidth

- The memory-bandwidth of Xeon is >50GB/s; Optane is 2GB/s per SSD
- Up to ~50% efficiency is expected, not more

What is important for Intel® Memory Drive Technology?

- Predictable accesses
  - If there is a pattern to the memory access, be it simple such as “sequential”, mid-complex like “fetch 1K every 72K”, or entirely complex like “if going to an ID field in a record in a table, fetch the whole record”
- High arithmetic intensity (FLOPs/byte ratio)
  - For every fetch from memory (in average) many compute cycles done
- High concurrency
  - Using at least 50% of the cores in a server platform concurrently, preferably more and even over-subscribed

IMDT BENCHMARKS

Hardware description

- Dual-socket Intel® Xeon® E5-2699 v4 (2x22 cores, 2.2 GHz)
  - First configuration (MDT):
    - 256 GB ECC DDR4
    - 4x320 GB Intel® Optane™ SSD (≈10 GB/s aggregated bandwidth)
  - Second configuration (lot of DRAM):
    - 1536 GB ECC DDR4
- Dual-socket Intel Xeon Gold 6154 (2x18 cores, 3.0 GHz)
  - First configuration:
    - 192 GB ECC DDR4
    - 8x Intel® Optane™ SSD
  - Second configuration
    - 1536 GB ECC DDR4
  - Only few benchmarks have been run yet

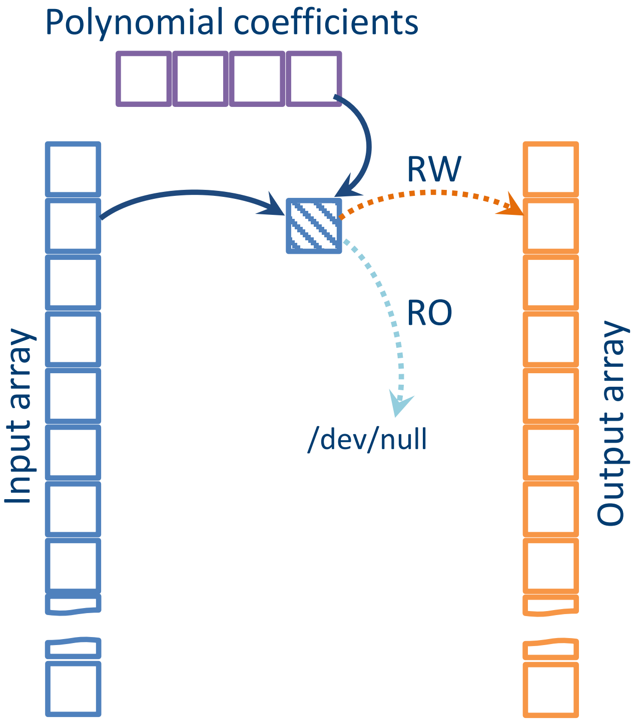
Polynomial benchmark

- Sequential-memory access benchmark
  - Compute polynomial values over a large array of input data
- Types of memory access patterns:
  - Read only (RO)
  - Read and write to another array (RW)
- Adjustable degree of polynomials
- Polynomials are computed using Horner method:

$$P(x) = (\dots((a_n x + a_{n-1})x + a_{n-2})\dots)x + a_0$$

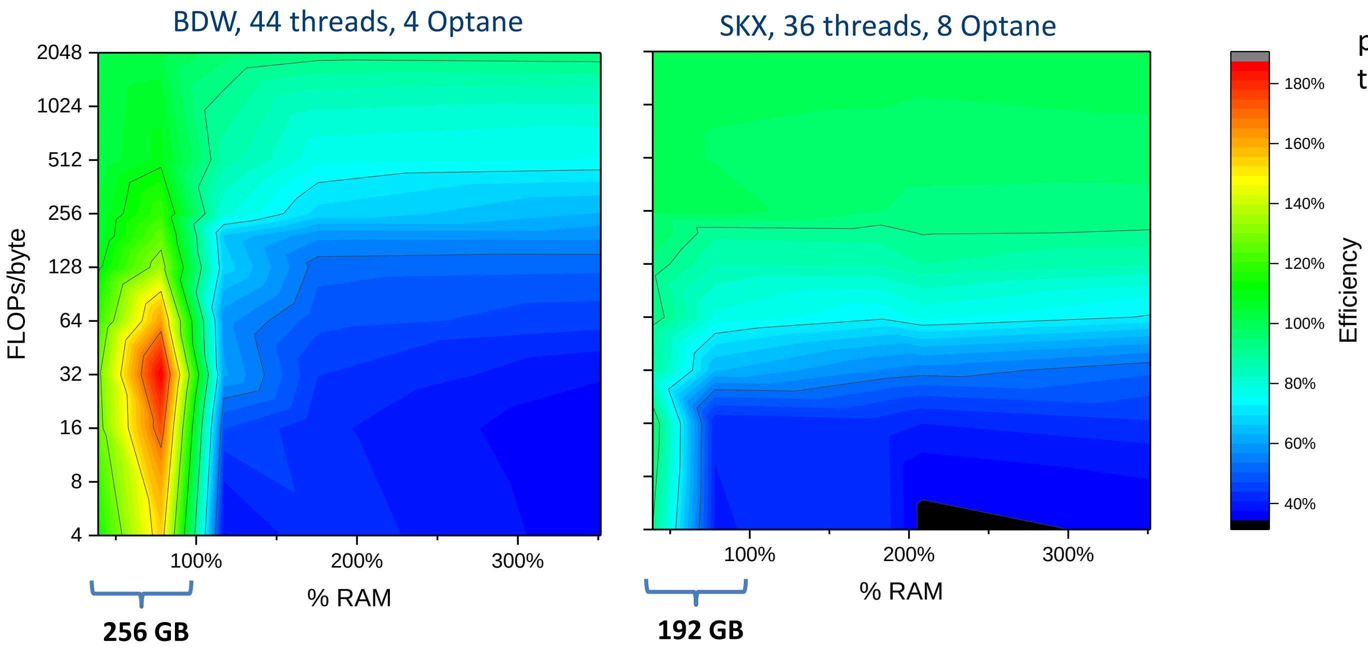
$$N_{FLOP} = (2 \cdot degree) \cdot N_{data}$$

$$\frac{FLOPs}{byte} = \frac{2 \cdot degree}{sizeof(real\_t)}$$



Polynomial benchmark (Read Only)

Efficiency: Intel® Memory Drive technology vs RAM

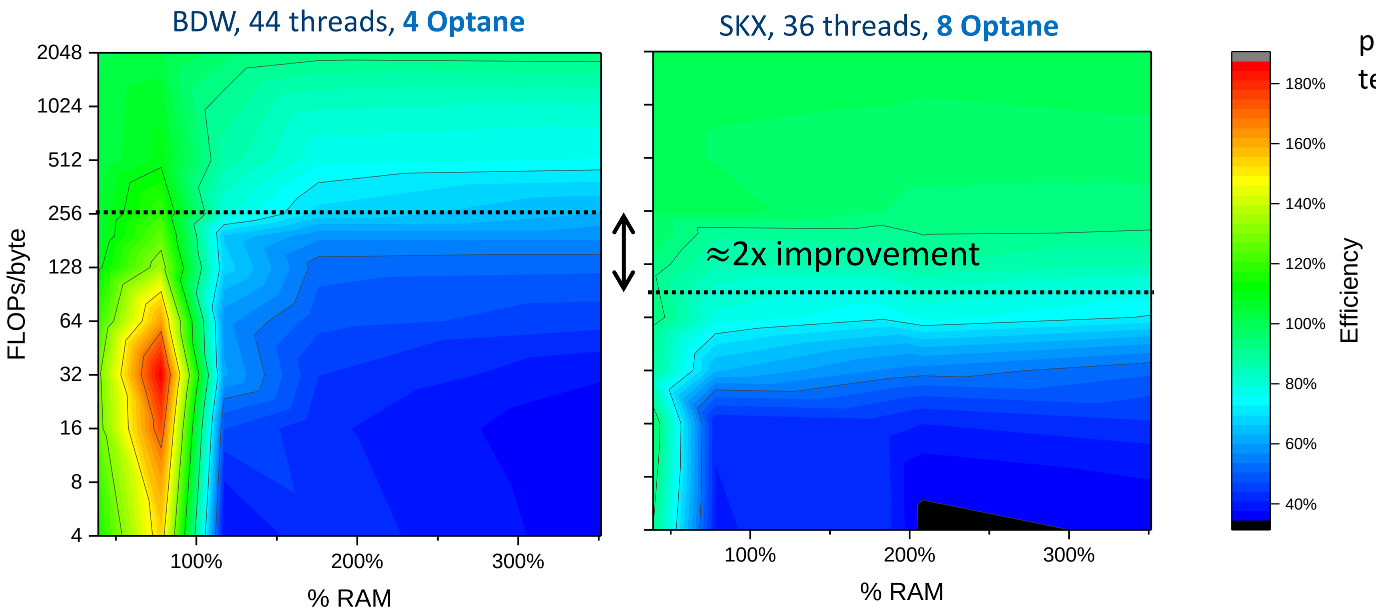


SKX results are preliminary – not all test cases have been sampled yet

% RAM – workload size, FLOPs/byte – workload complexity, color – efficiency

Polynomial benchmark (Read Only)

Efficiency: Intel® Memory Drive technology vs RAM

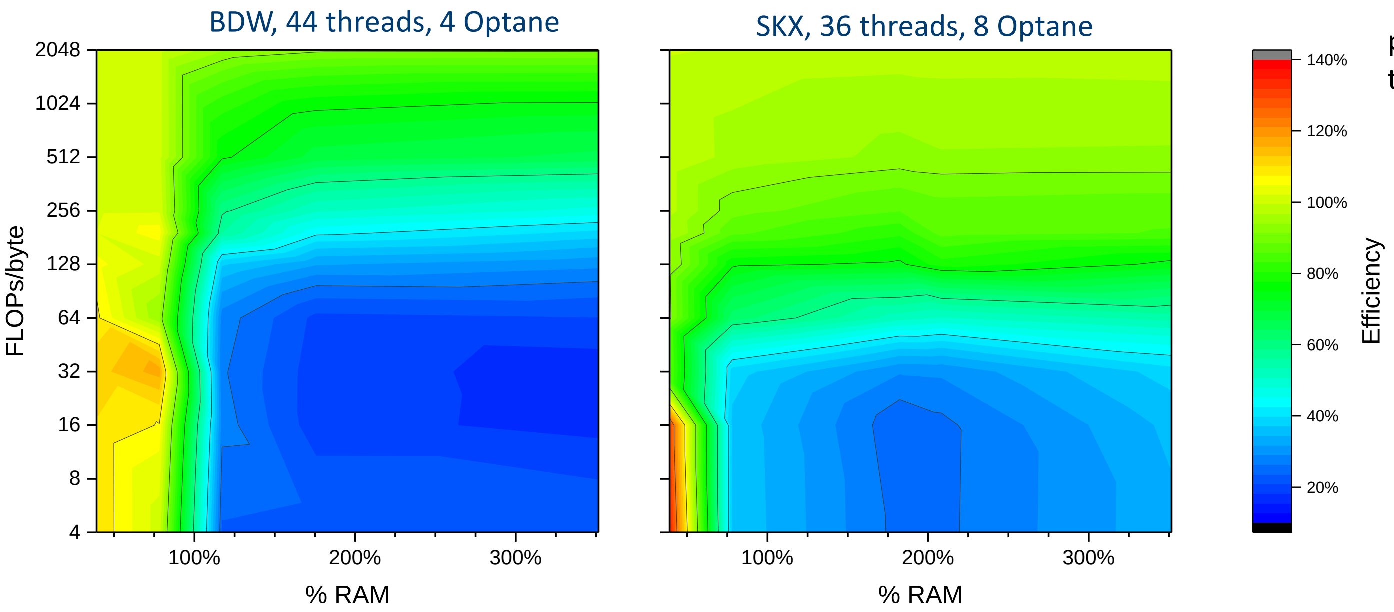


SKX results are preliminary – not all test cases have been sampled yet

% RAM – workload size, FLOPs/byte – workload complexity, color – efficiency

Polynomial benchmark (Read&Write)

Efficiency: Intel® Memory Drive technology vs RAM



SKX results are preliminary – not all test cases have been sampled yet

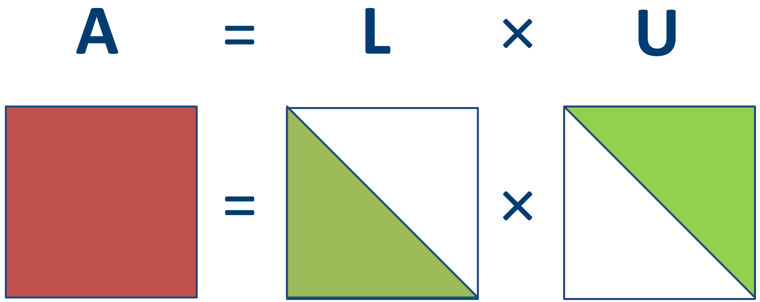
% RAM – workload size, FLOPs/byte – workload complexity, color – efficiency

Polynomial benchmark summary

- If data size is larger than DRAM:
  - Arithmetic intensity (AI) requirements to get efficiency >80% depends on the workload, number of drives and CPU:
    - RO: 128-256 FLOPs/byte
    - RW: 256-512 FLOPs/byte
  - AI should be measured on DRAM-LLC level
- If data fits in DRAM:
  - No performance degradation
  - MDT can be faster for NUMA non-aware applications
- Arithmetic intensity requirements decrease linearly with the number of Intel Optane drives

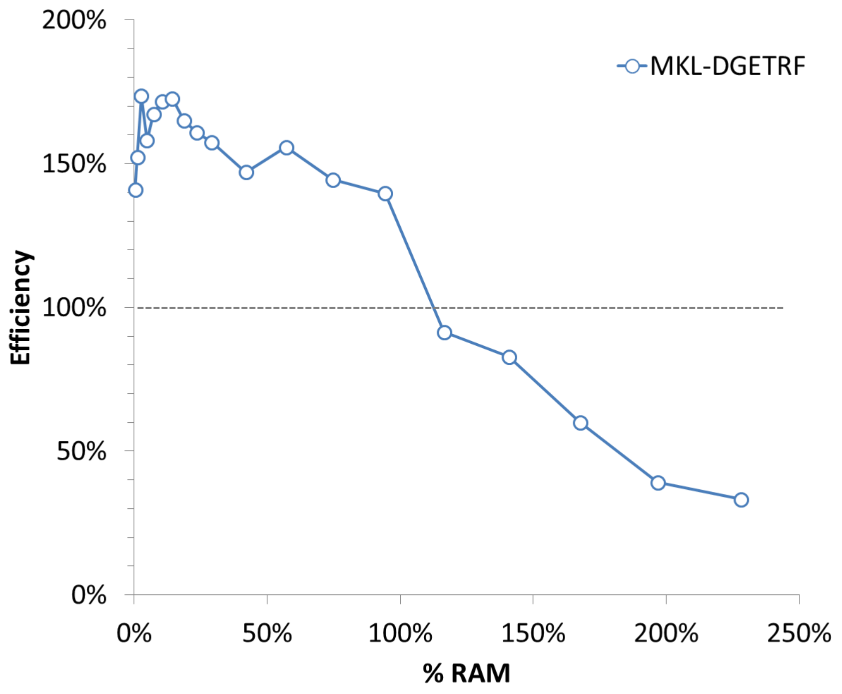
LU decomposition

- Factorization of matrix A into product of lower triangular (L) and upper triangular (U) matrices
- A commonly used kernel in many scientific codes:
  - Solving systems of linear equations
  - Matrix inversion
  - Computing determinants
- A kernel in LINPACK benchmark



LU decomposition

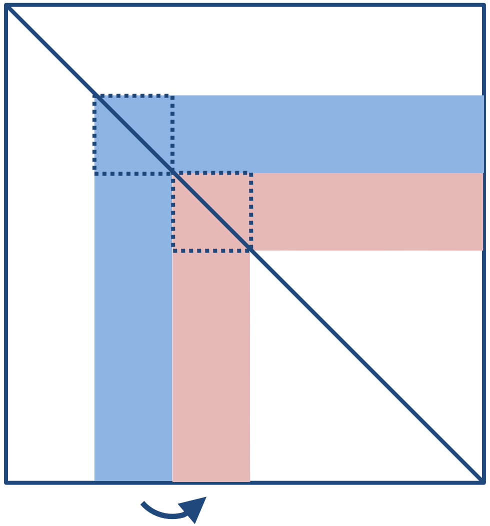
- Performance results
  - DRAM maximum performance: 850 GFLOPs/s
  - Intel® Memory Drive Technology maximum performance: 1,250 GFLOPs/s
  - A huge performance degradation beyond ≈150% RAM utilization
- Can we improve the results?



LU decomposition

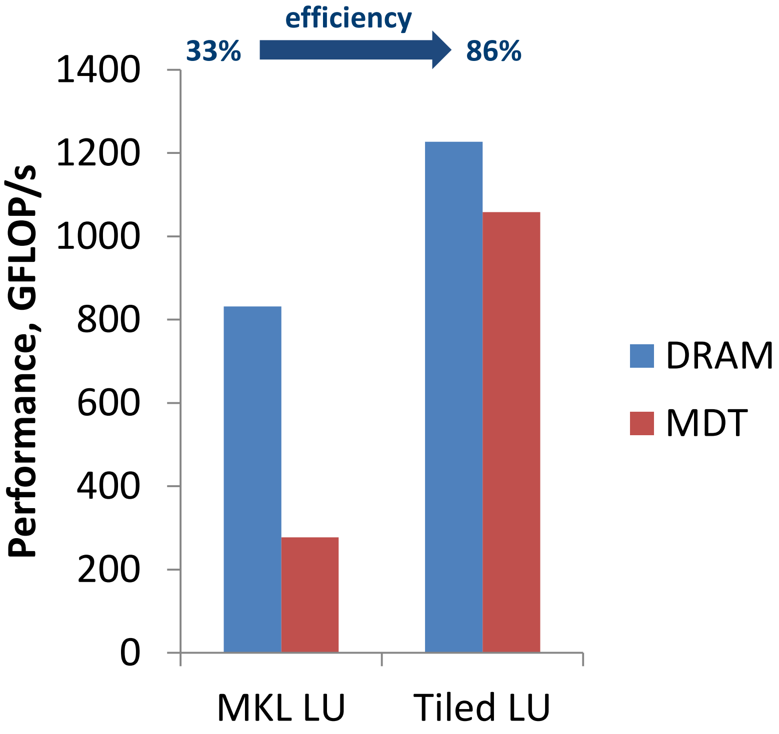
- Memory access pattern is by column blocks
- Nearby elements are scattered throughout different memory pages
  - 4KB page = 512 double precision numbers
  - A huge data traffic for large matrices ($2 \cdot 10^5$ and above)
- There are tiled LU algorithms (e.q. PLASMA)

LU memory access pattern



LU decomposition

- Memory access pattern is by column blocks
- Nearby elements are scattered throughout different memory pages
  - 4KB page = 512 double precision numbers
  - A huge data traffic for large matrices ($2 \cdot 10^5$and above)
- There are tiled LU algorithms (e.q. PLASMA)
- We used a simple implementation from *hetero-streams* code base
- Little performance degradation beyond 100% RAM usage



$N = 280000$, 230% RAM

Lessons learned from benchmarks with Intel® Memory Drive Technology

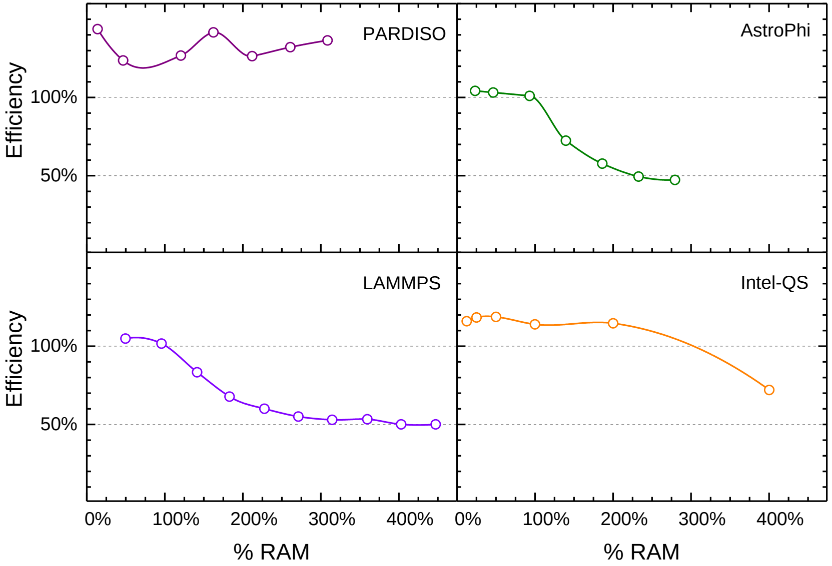
- Data moving between Intel® Optane™ SSDs and RAM is very expensive (10-20 GB/s max):
  - Reuse data as much as possible
    - Arithmetic intensity on DRAM↔MDT level should be ≥200-500 FLOPs/byte depending on the number of Optane
  - Redesign data structures in you program for locality
  - Work with large data chunks
  - Think about DRAM as a large L4 cache for MDT
- Same optimization principles as on NUMA architectures
- Data-oriented programming is a must
  - It benefits another modern hardware as well

Scientific applications

- Computational chemistry:
  - LAMMPS* (molecular dynamics)
  - GAMESS (two-electron integral kernel)
- Astrophysics:
  - AstroPhi* (hyperbolic partial differential equation solver)
- Sparse linear algebra problems:
  - Intel® Math Kernel Library PARDISO
- Quantum computing simulator:
  - Intel-QS, formerly known as qHipster

Scientific applications

- Results:
  - Efficiency is slightly higher than 100% within DRAM
  - Efficiency beyond DRAM varies from 50% up to >100%
  - LAMMPS, AstroPhi and Intel-QS are memory bound apps, efficiency tends to 50% when memory growth



Conclusions

- Efficiency of optimized applications is close to 100% with Intel® Memory Drive Technology
- Efficiency of non-optimized applications can vary from 20% to more than 100%. Typical efficiency of bandwidth-bound applications is up to 50%.
- Optimal performance is expected on next generation of Intel® Optane™ SSDs

Future work

– Scaling of IMDT performance vs number of Optane SSDs
– Comparing Intel Optane-powered fat-memory node with distributed memory on scientific applications
– Testing Intel® Optane™ DC Persistent memory

Our paper at SC’18 workshop

V. Mironov, A. Kudryavtsev, Y. Alexeev, A. Moskovsky, I. Kulikov, and I. Chernykh. 2018. Evaluation of Intel Memory Drive Technology Performance for Scientific Applications. In: *Proceedings of the Workshop on Memory Centric High Performance Computing* (MCHPC'18). ACM, New York, NY, USA, 14-21.

Acknowledgements

- We thank:
  - ScaleMP team for technical support
  - RSC Group and Siberian Supercomputer Center ICMMG SB RAS for providing access to certain hardware
  - Gennady Fedorov (Intel) for help with Intel PARDISO benchmark
  - Igor Chernykh and Igor Kulikov (ICMMG SB RAS) for AstroPhi code benchmarks
  - Justin Hogaboam (Intel) for Intel QS code
  - all of you for your attention!