Considerations for parallel optimization

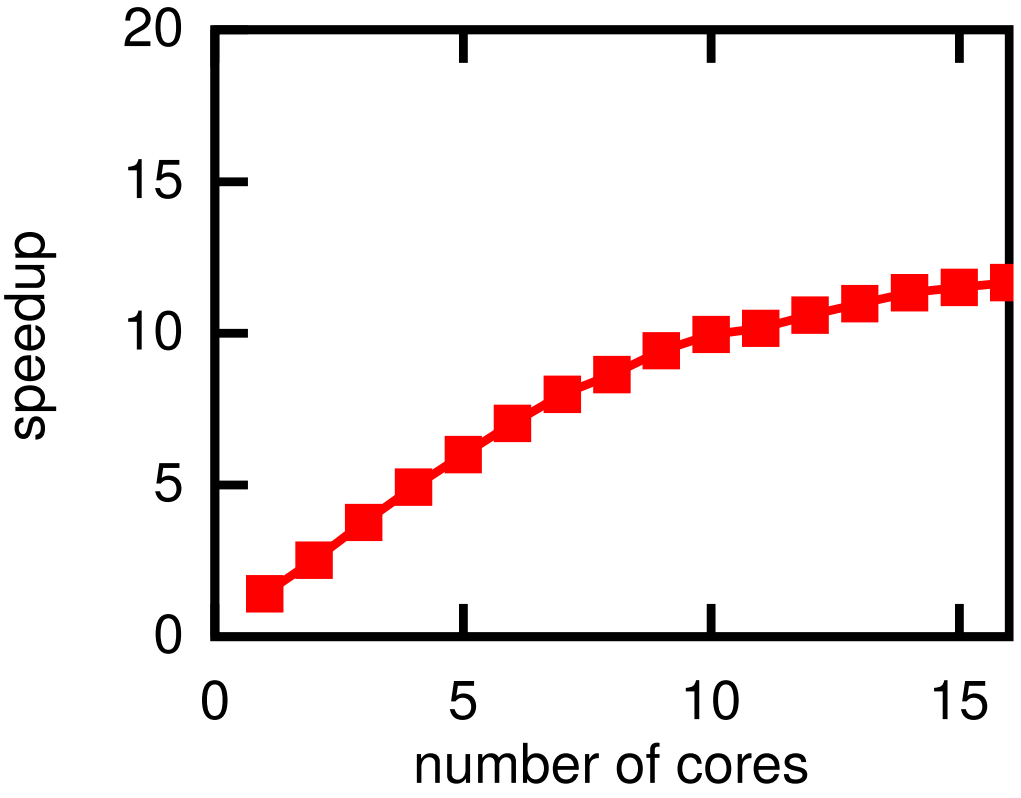
Tamás Fehér

High Level Support Team
Max Planck Institute for Plasma Physics, Garching, Germany

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

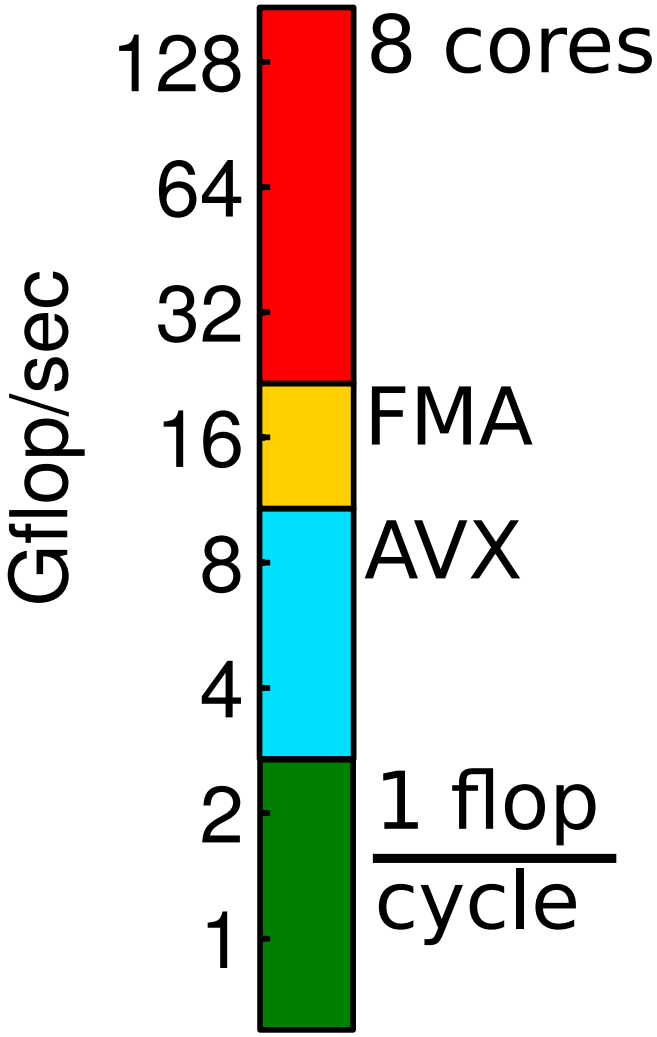
Outline

- Hardware bottlenecks
- Roofline model
- Amdahl's law
- OpenMP & MPI overhead

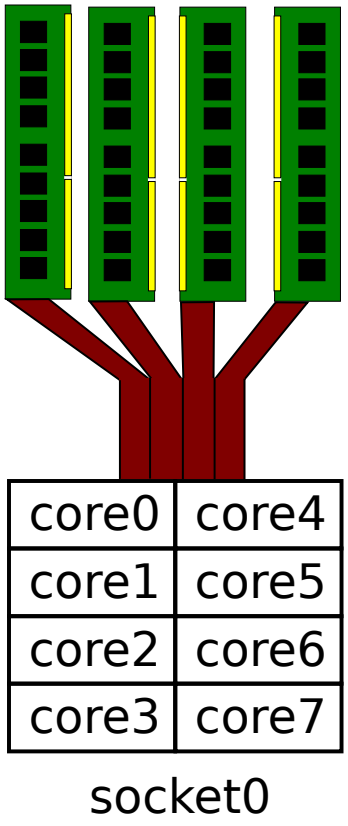


CPU bottleneck (HELIOS)

- Intel Xeon E5-2680
- 8 cores
- 2.7 GHz
- 21.6 Gflop/s (core)
- 345.6 Gflop/s (node)

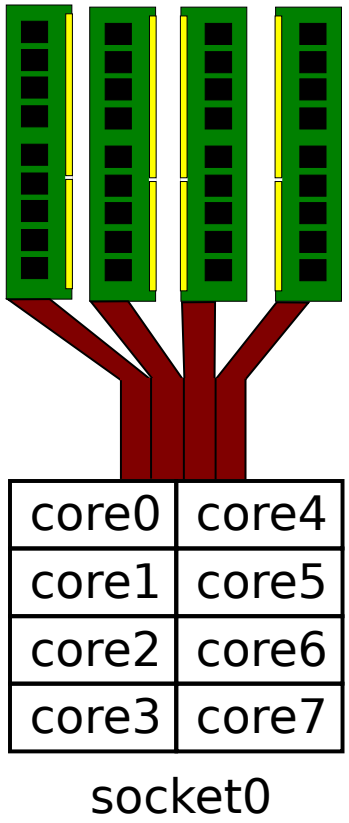


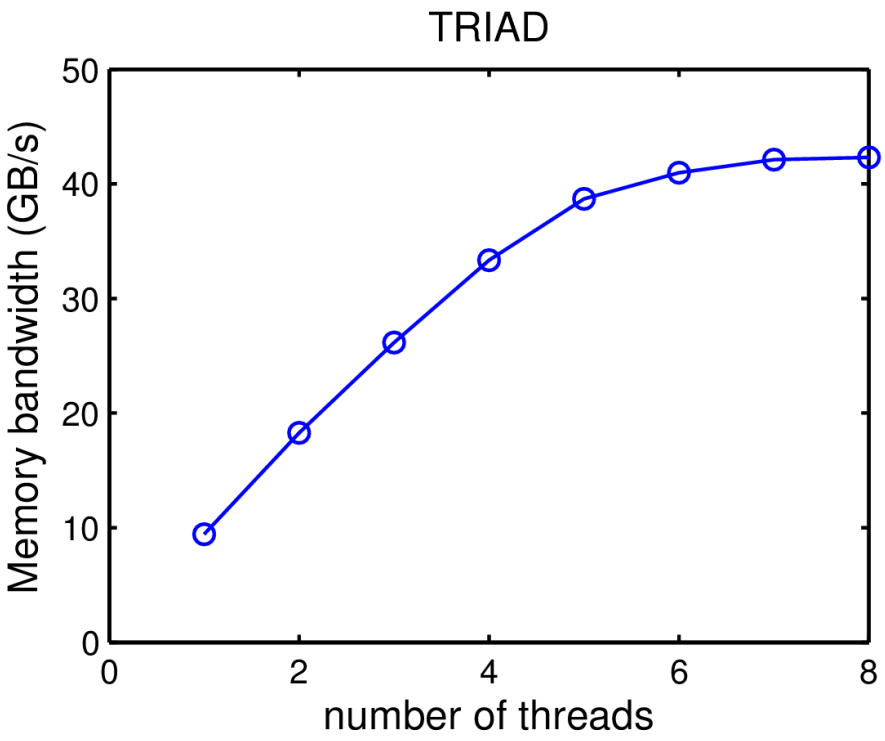
Memory bottleneck



- Memory: 4 x DDR3 1600
- Total Bandwidth: 51.2 GB/sec
- 4 channels 12.8 GB/sec

Memory bottleneck





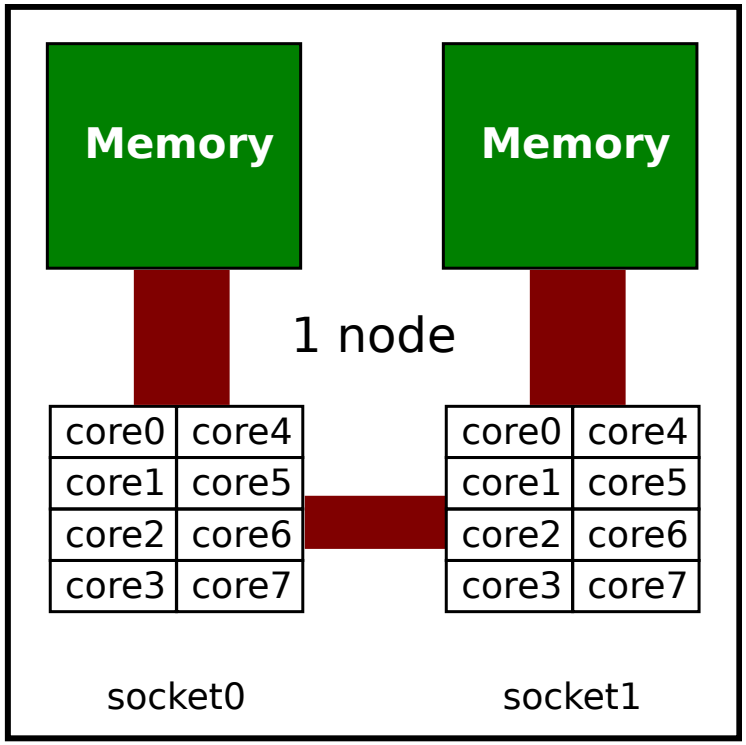
STREAM benchmark

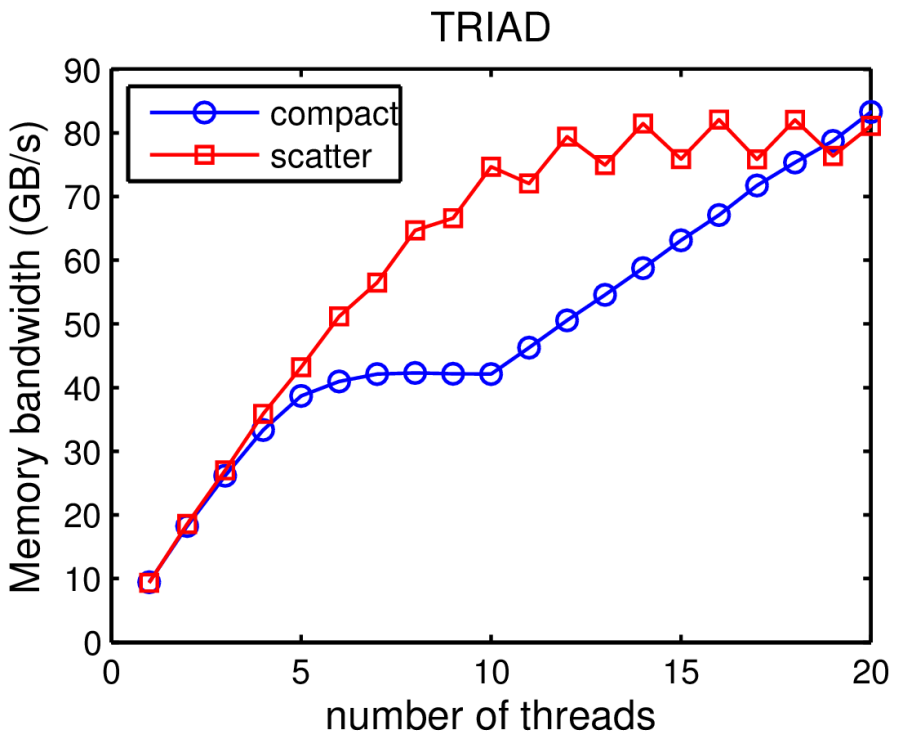
J. D. McCalpin https://www.cs.virginia.edu/stream/

```
!$OMP PARALLEL DO
DO j = 1,n
   b(j) = 0.5d0
   c(j) = 0.2d0
END DO
```

```
!$OMP PARALLEL DO
DO j = 1,n
   a(j) = b(j) + scalar*c(j)
END DO
```

Memory bottleneck





STREAM benchmark

J. D. McCalpin https://www.cs.virginia.edu/stream/

```
!$OMP PARALLEL DO
DO j = 1,n
   b(j) = 0.5d0
   c(j) = 0.2d0
END DO
```

```
!$OMP PARALLEL DO
DO j = 1,n
   a(j) = b(j) + scalar*c(j)
END DO
```

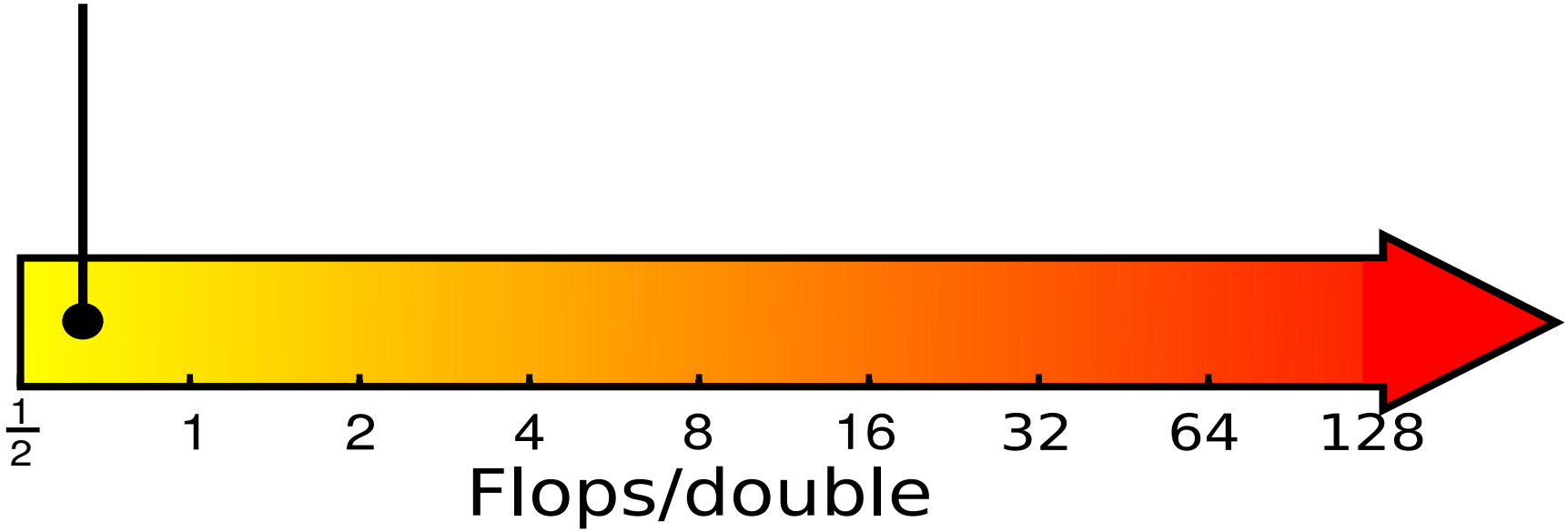
Arithmetic Intensity = Flops / bytes

$$AI\left[\frac{\text{Flops}}{\text{bytes}}\right] \cdot BW\left[\frac{\text{bytes}}{\text{sec}}\right] = \text{performance}\left[\frac{\text{Flops}}{\text{sec}}\right]$$

Arithmetic Intensity = Flops / bytes

$$AI\left[\frac{\text{Flops}}{\text{bytes}}\right] \cdot BW\left[\frac{\text{bytes}}{\text{sec}}\right] = \text{performance}\left[\frac{\text{Flops}}{\text{sec}}\right]$$

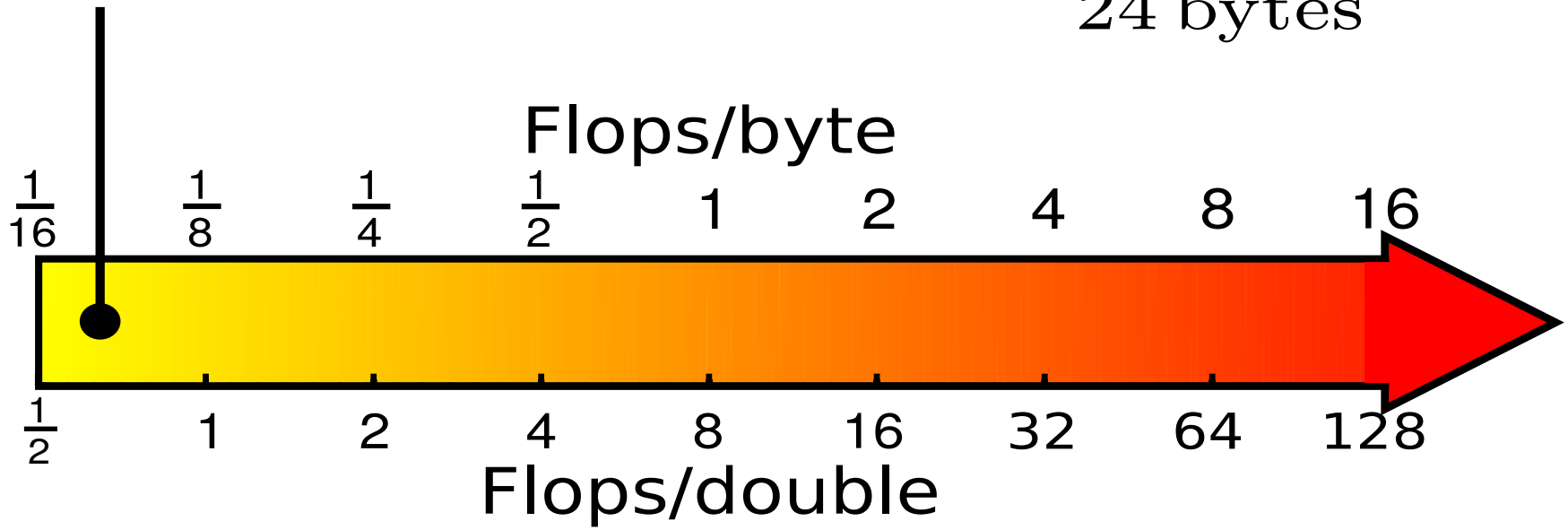
TRIAD: a(i) = b(i) + s*c(i)



Arithmetic Intensity = Flops / bytes

$$AI\left[\frac{\text{Flops}}{\text{bytes}}\right] \cdot BW\left[\frac{\text{bytes}}{\text{sec}}\right] = \text{performance}\left[\frac{\text{Flops}}{\text{sec}}\right]$$

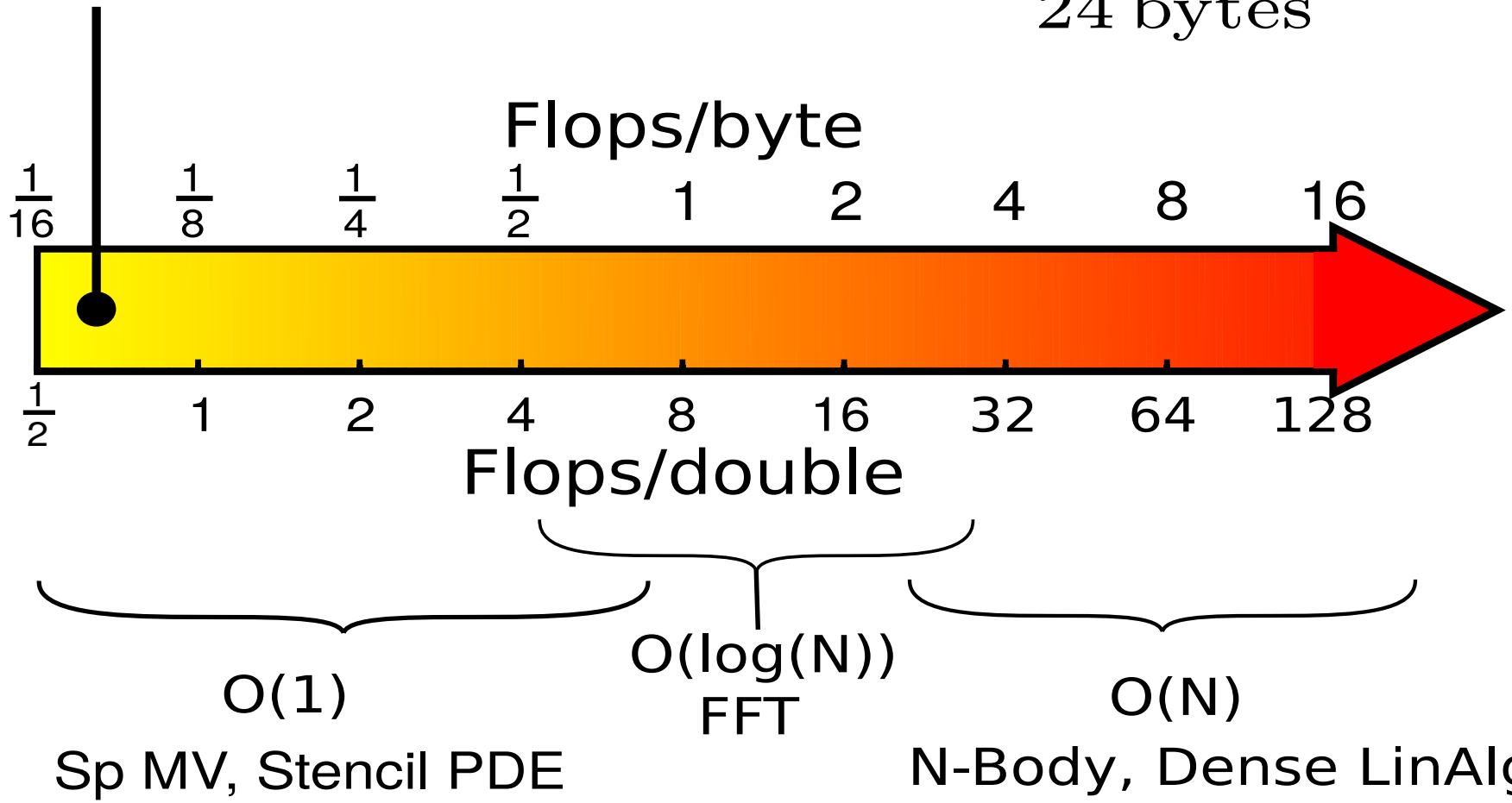
TRIAD: a(i) = b(i) + s*c(i) $\longrightarrow \frac{2\ \text{Flops}}{24\ \text{bytes}}$



Arithmetic Intensity = Flops / bytes

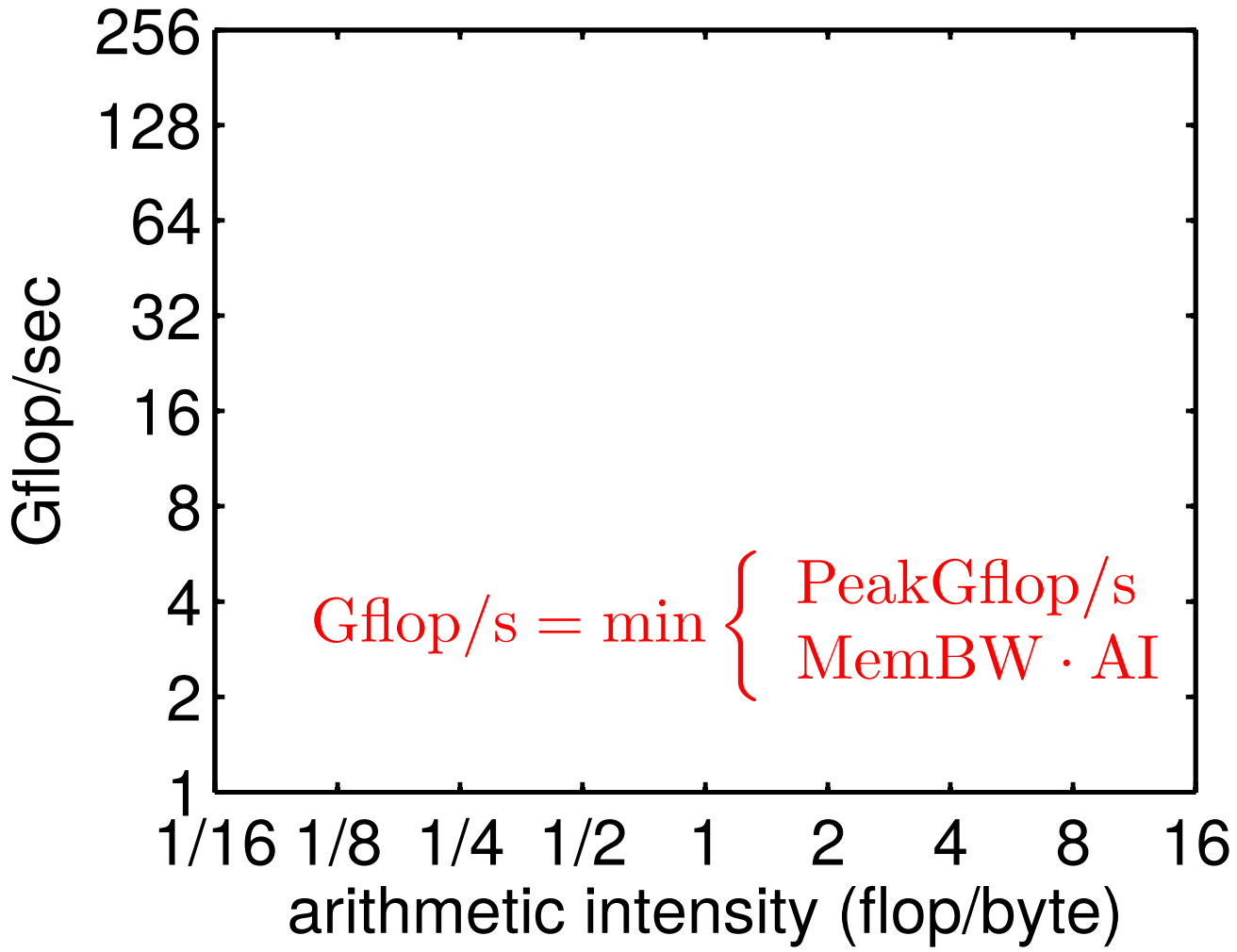
$$AI\left[\frac{\text{Flops}}{\text{bytes}}\right] \cdot BW\left[\frac{\text{bytes}}{\text{sec}}\right] = \text{performance}\left[\frac{\text{Flops}}{\text{sec}}\right]$$

TRIAD: a(i) = b(i) + s*c(i) $\longrightarrow \frac{2\ \text{Flops}}{24\ \text{bytes}}$



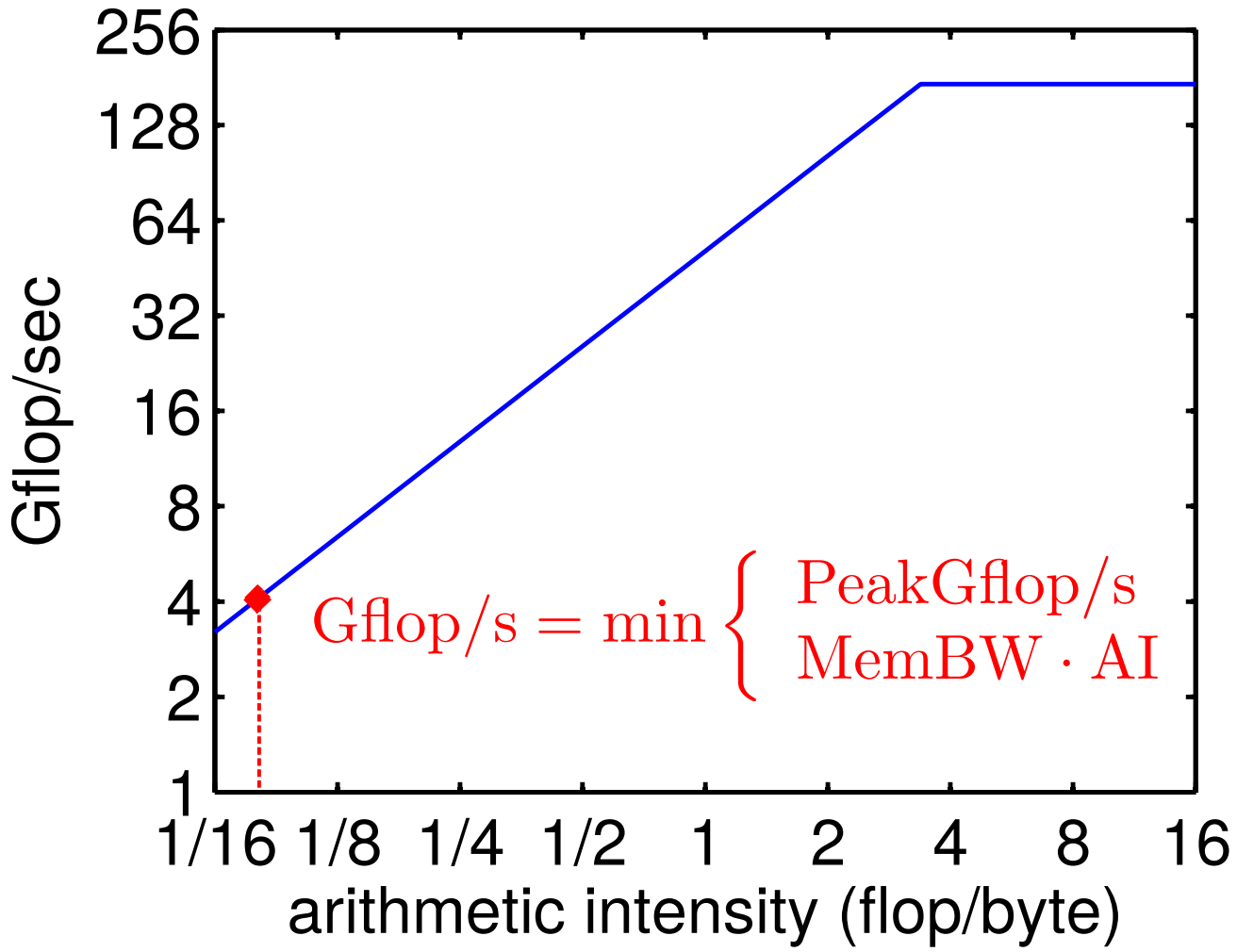
Roofline model

S. Williams et. al, Communications of the ACM 52, 65-72 (2009)

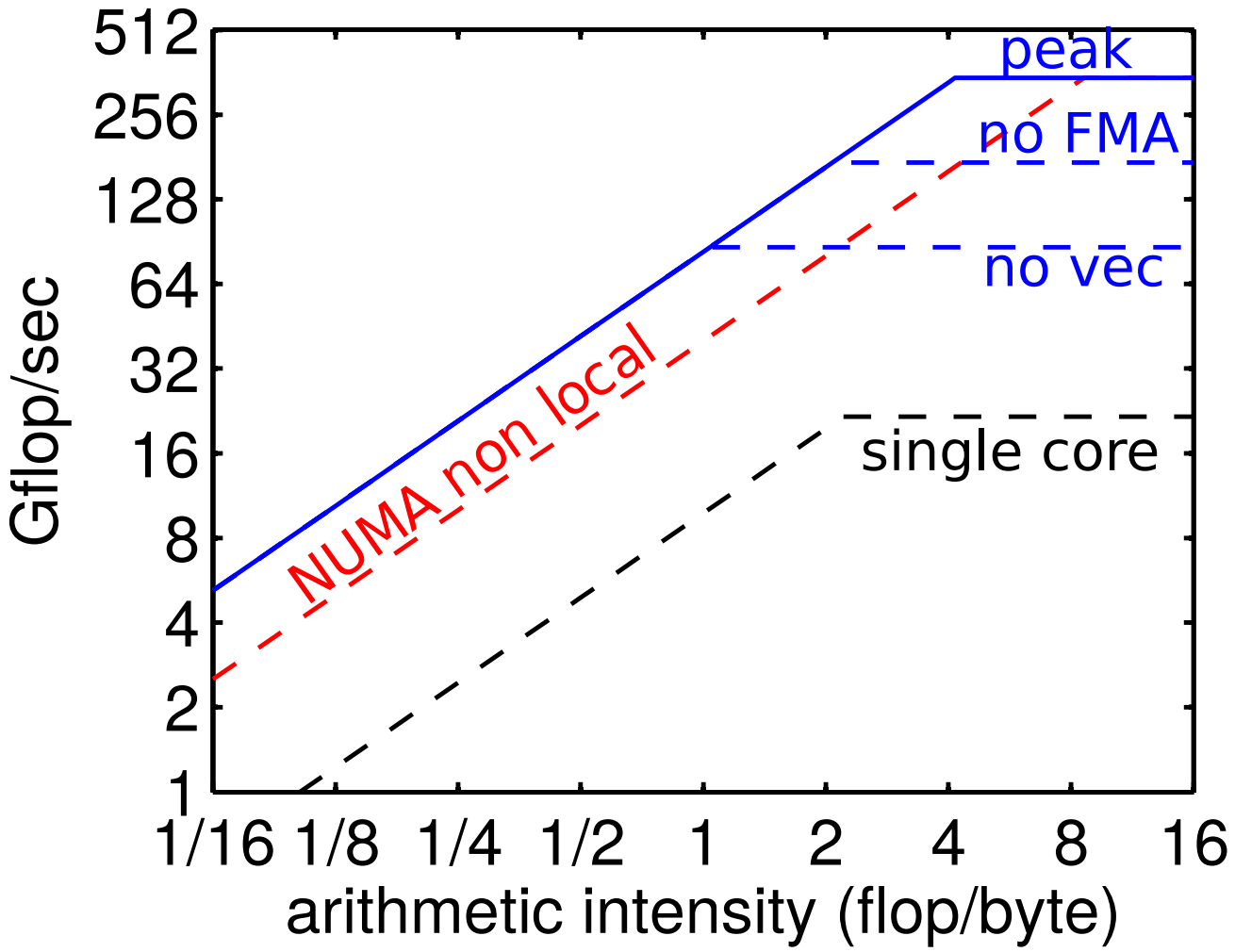


Roofline model

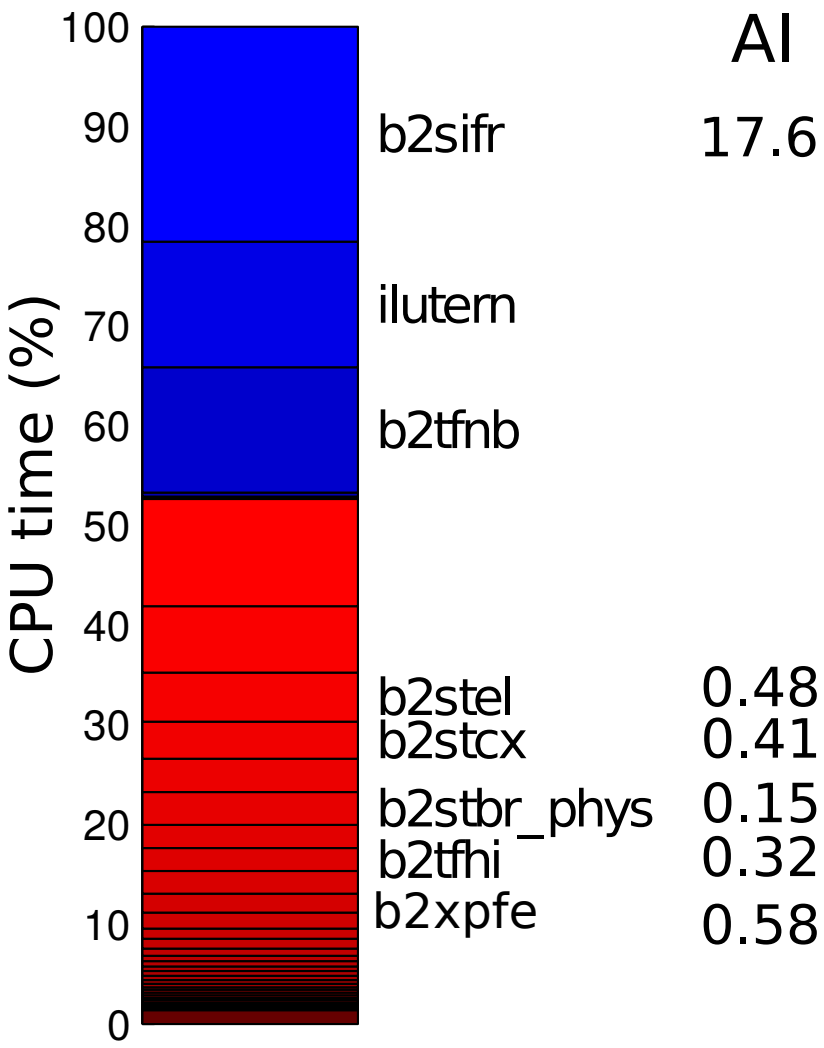
S. Williams et. al, Communications of the ACM 52, 65-72 (2009)



Roofline ceilings

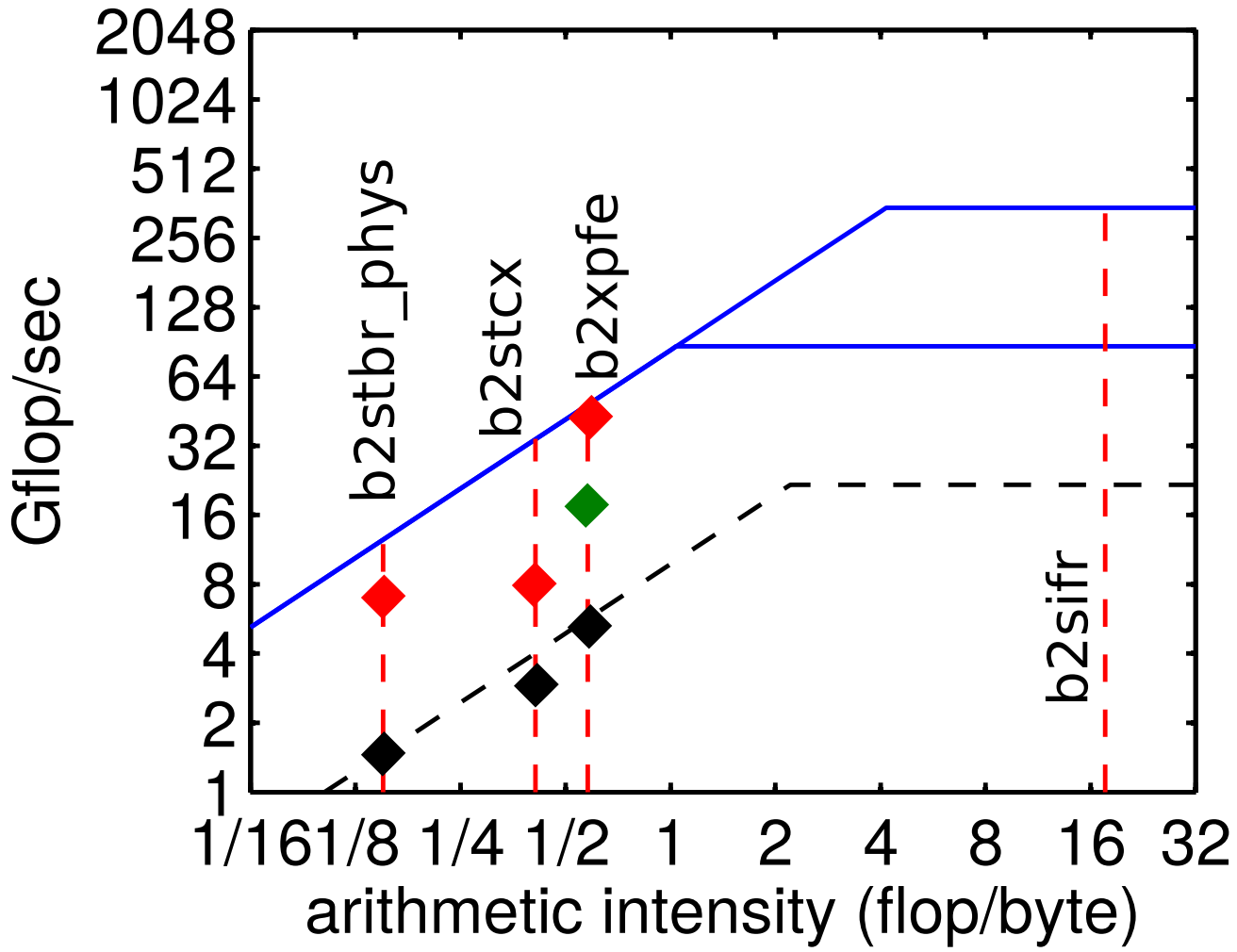


Examples from B2



- Profiled with FTIMINGS
- Memory usage calculated separately
- Ideal cache usage assumed
- AI between 0.1 – 0.6
- Exception: b2sifr (N-body)

B2 Roofline


2048
1024
512
256
128
64
32
16
8
4
2
1
Gflop/sec
b2stbr_phys
b2stcx
b2xpfe
b2sifr
1/16
1/8
1/4
1/2
1
2
4
8
16
32
arithmetic intensity (flop/byte)

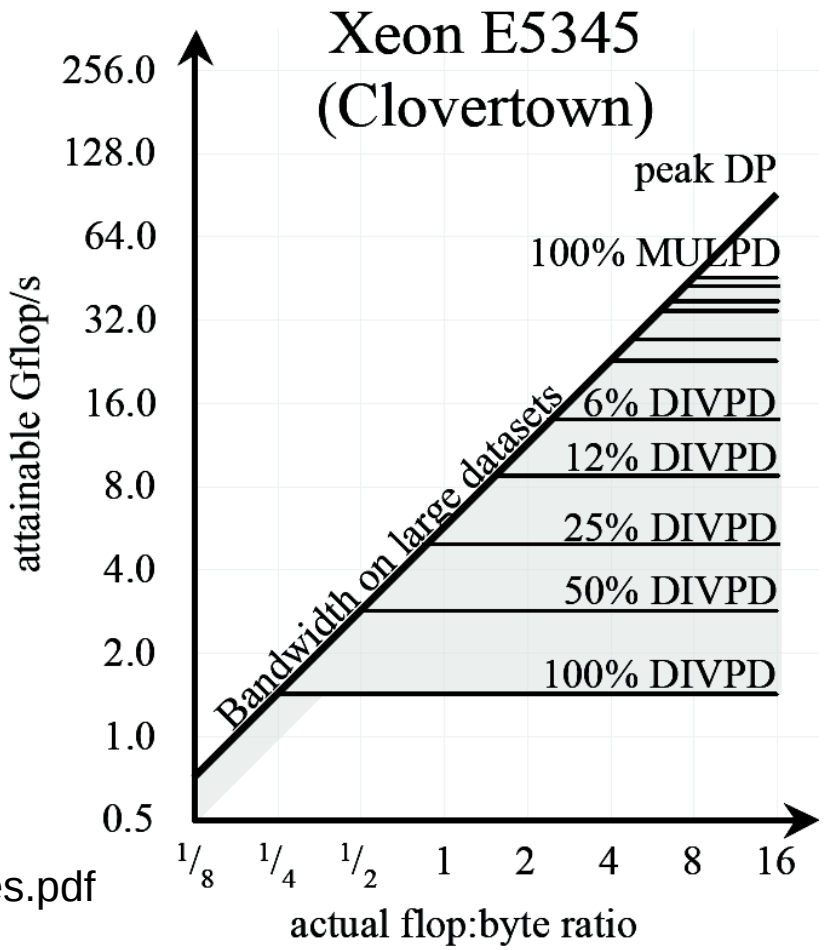
Flop ≠ Flop

- Division takes longer
- Not pipelined
- Can stall other instructions

instruction	+ VADDPD	/ VDIVPD
latency	3	21-29
Issue latency	1	20-28

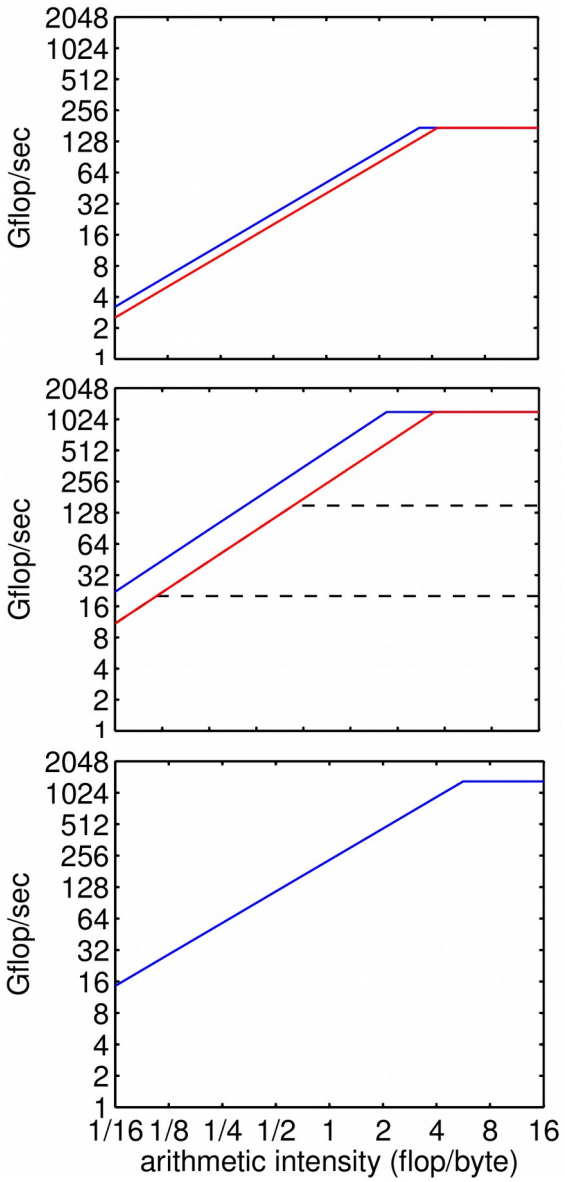
A. Fog: Instruction Tables, TU Denmark
http://www.agner.org/optimize/instruction_tables.pdf

S. Williams PhD Thesis (2008)



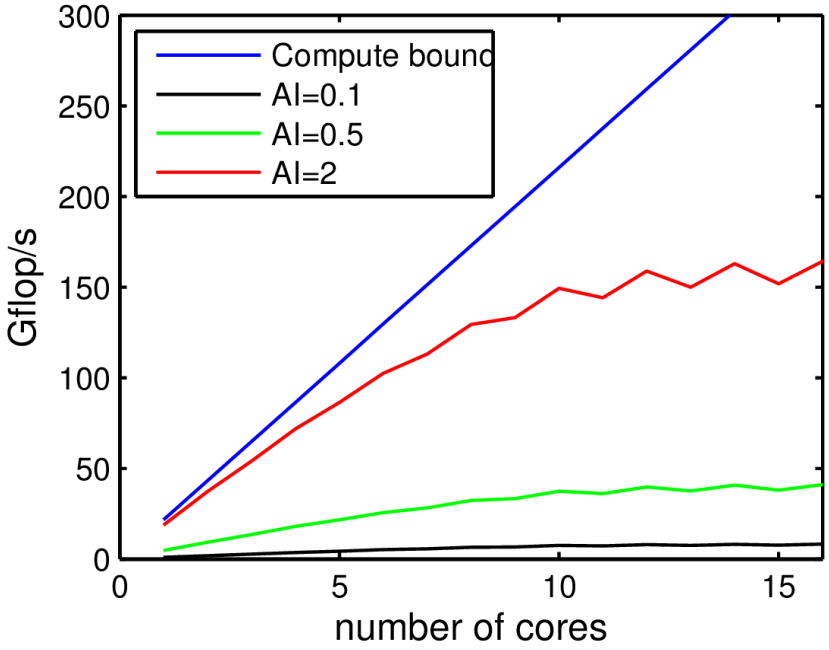
Different architectures

- Intel Xeon E5 2680
  - 172 Gflop/s
  - 51.2 GB/s
- Xeon Phi 7120
  - 1.2 Tflop/s
  - 174 GB/s
- Nvidia Tesla K20x
  - 1.3 Tflop/s
  - 250 GB/s

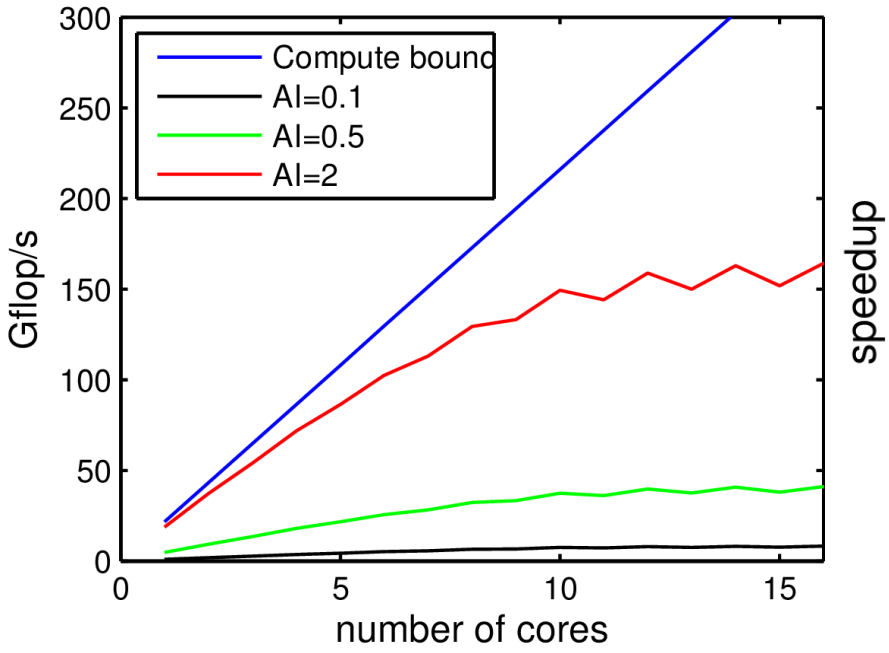


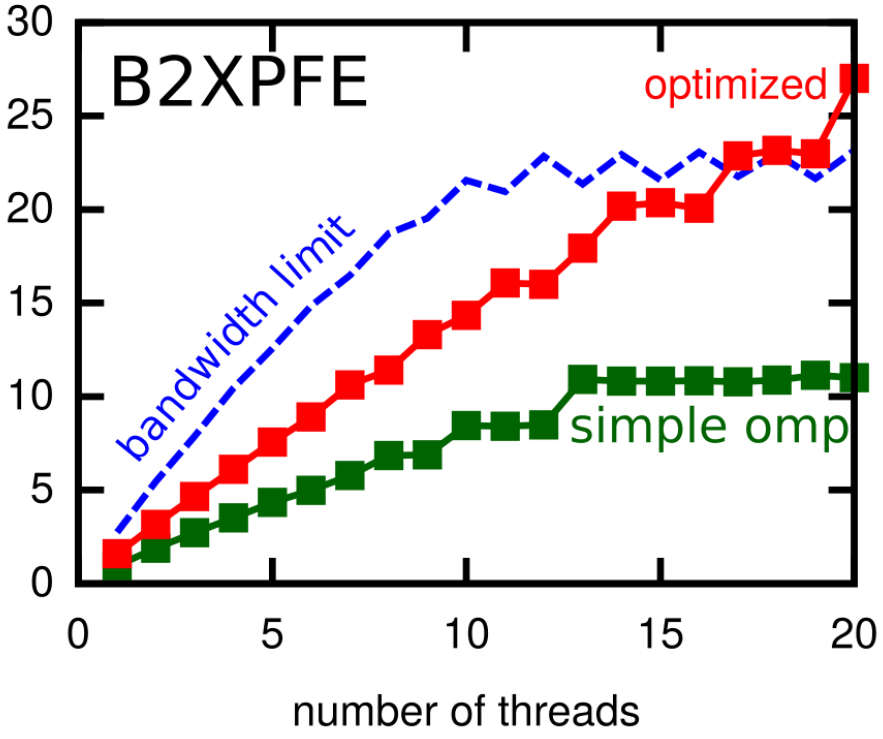
Colors: theoretical / actual bandwidth

Cores vs speedup



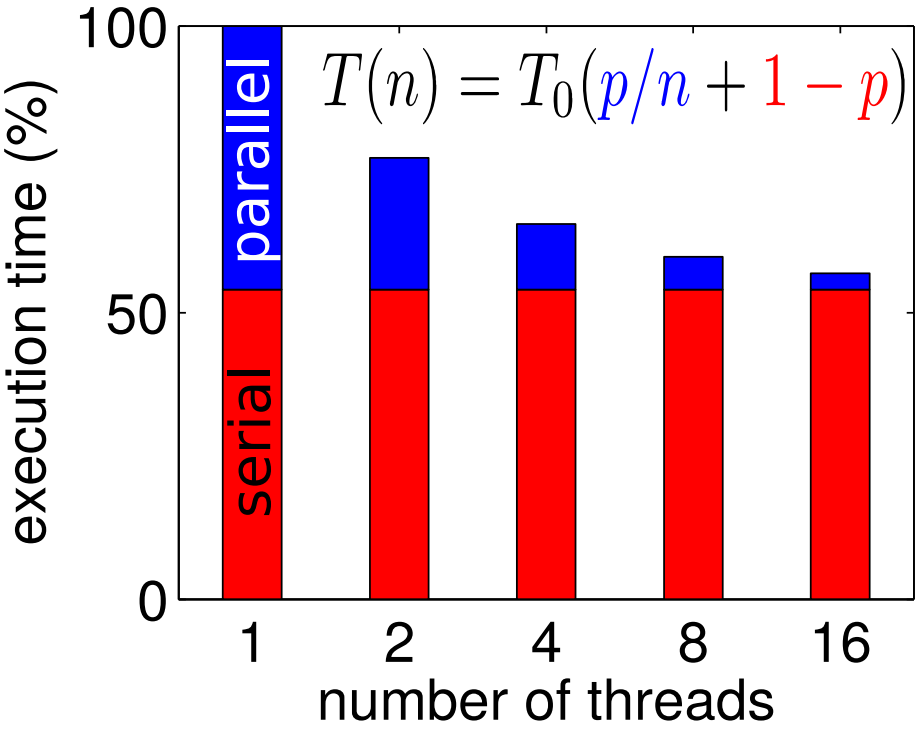
Cores vs speedup

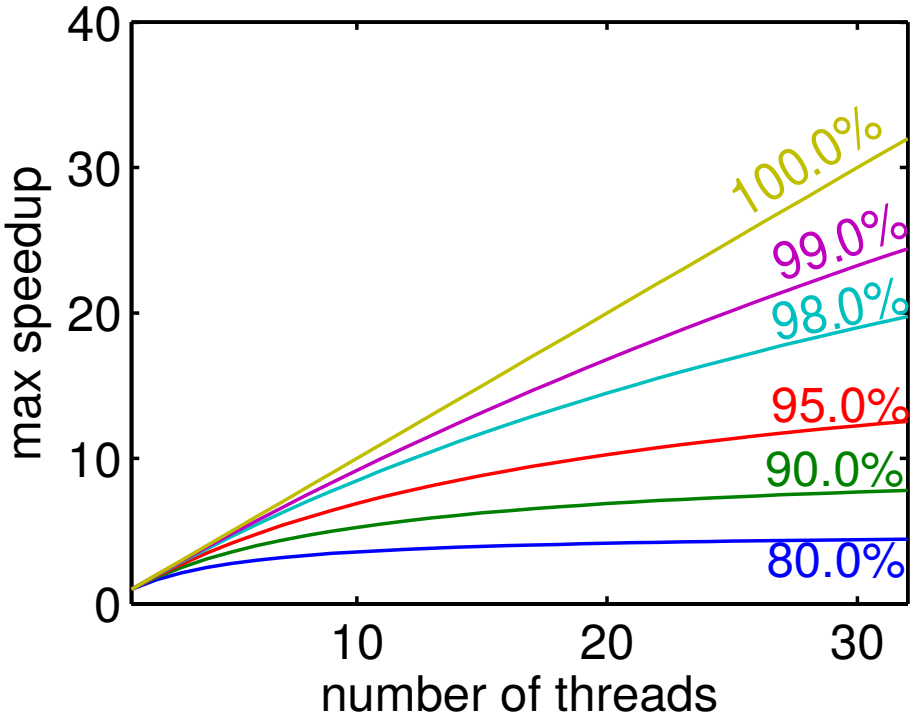




Amdahl's law

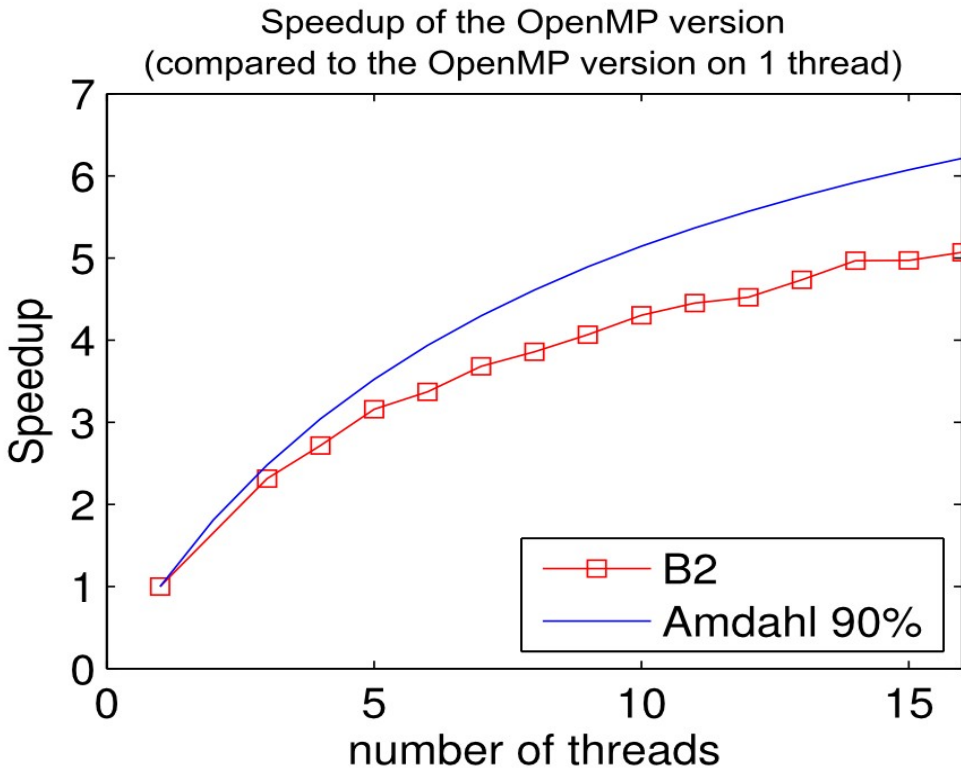
Speedup depends on the parallel fraction of the code

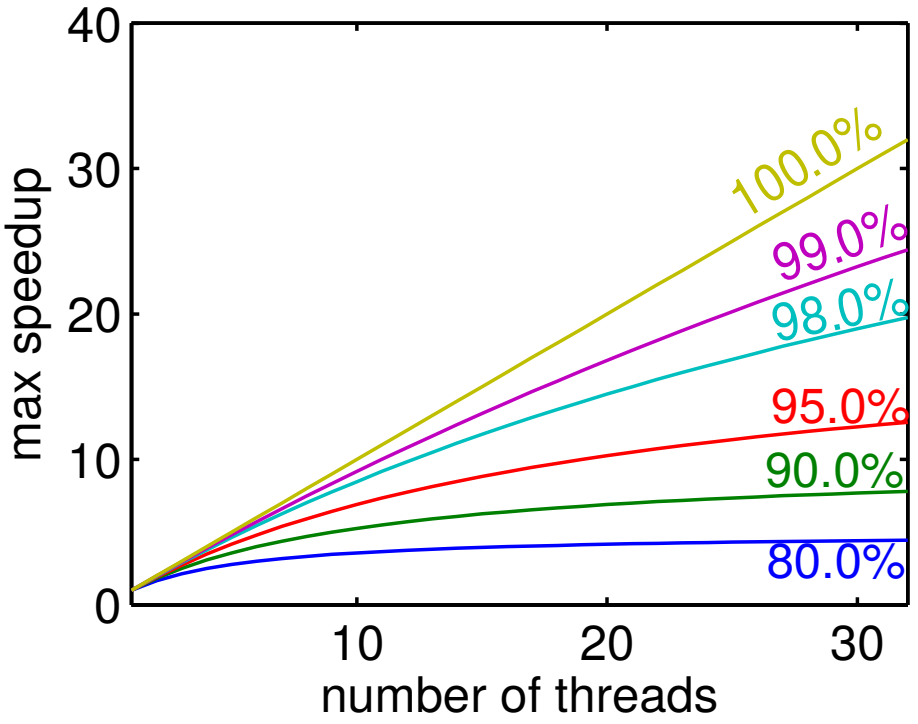




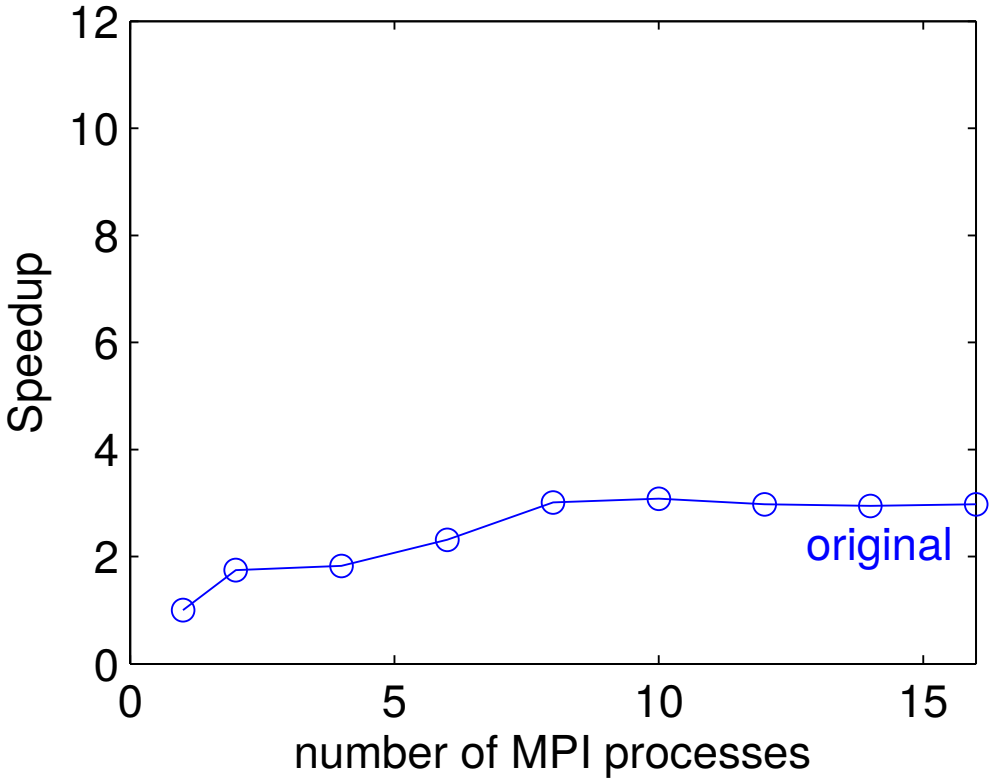
Amdahl's law

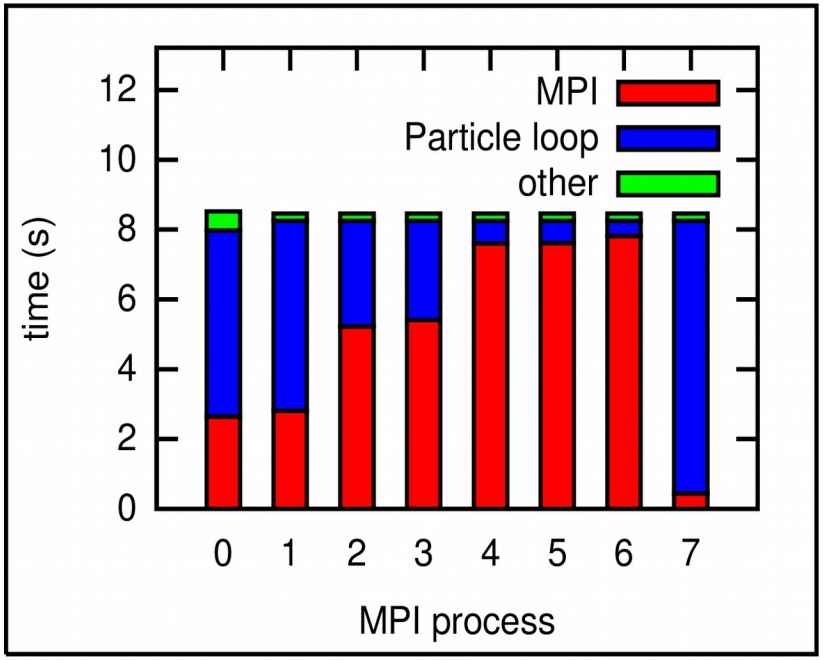
Speedup depends on the parallel fraction of the code



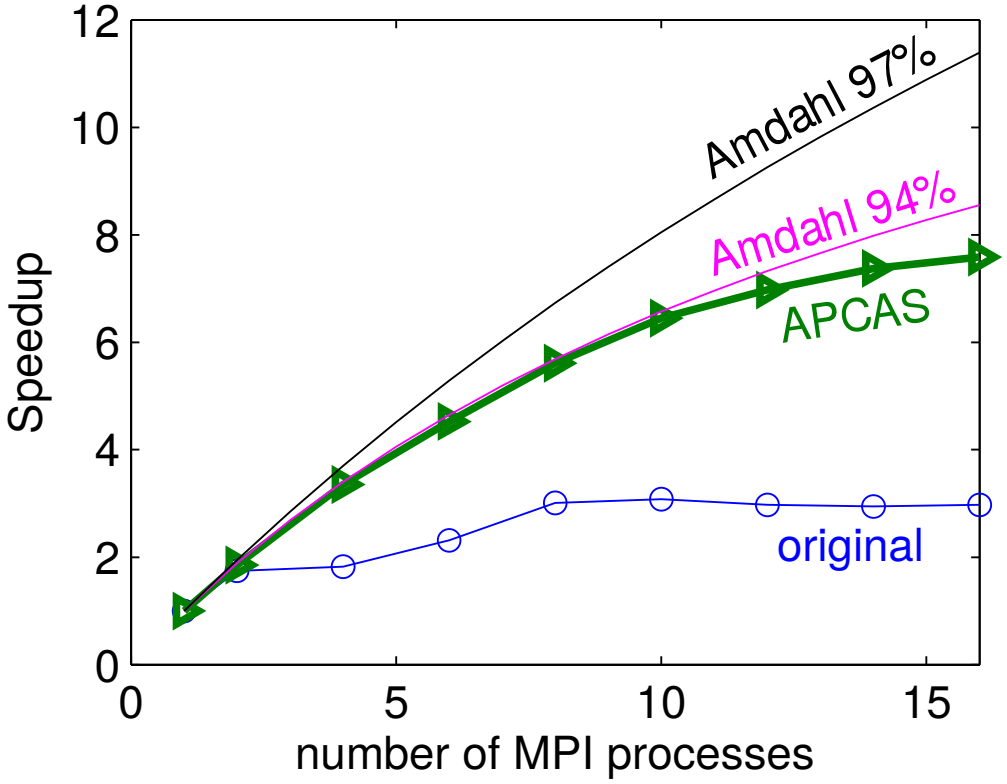


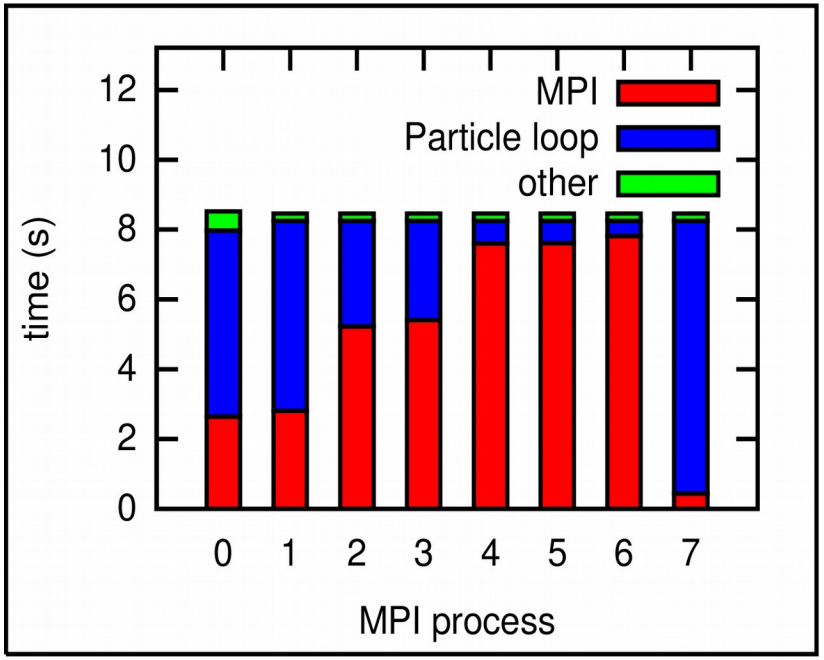
Load Balance





Load Balance





OpenMP parallelization

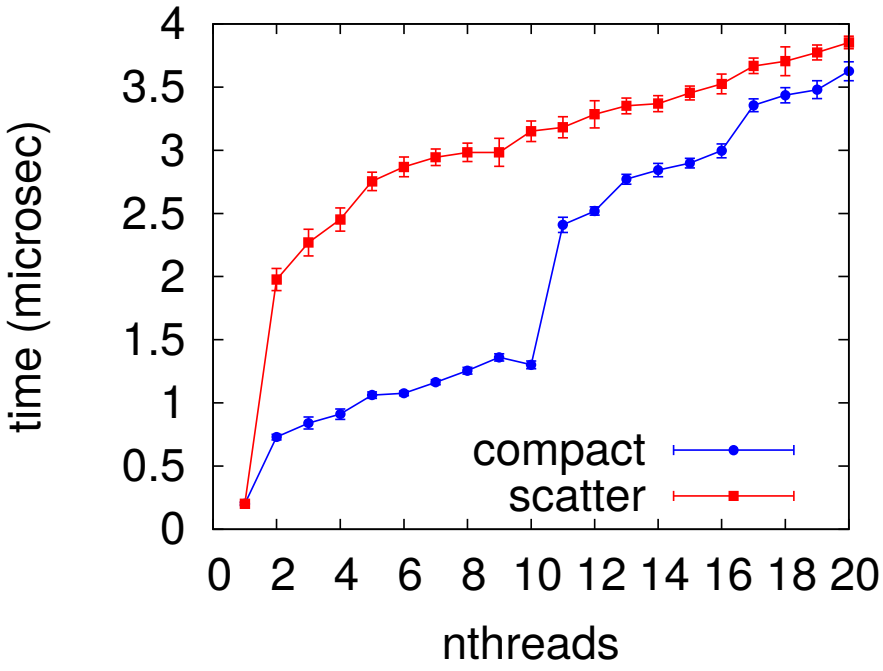
- Overhead for creating parallel region on Hydra Ivy Bridge
- 4 µs ~ 0.086 Mflop (1 core) ~ 41 kb
- MIC – more threads, more overhead
- Reduction over arrays can be costly

Sequential code

```
for ( j = 0 ; j < innerreps ; j ++) {
    delay ( delaylength ) ;
}
```

Parallel code

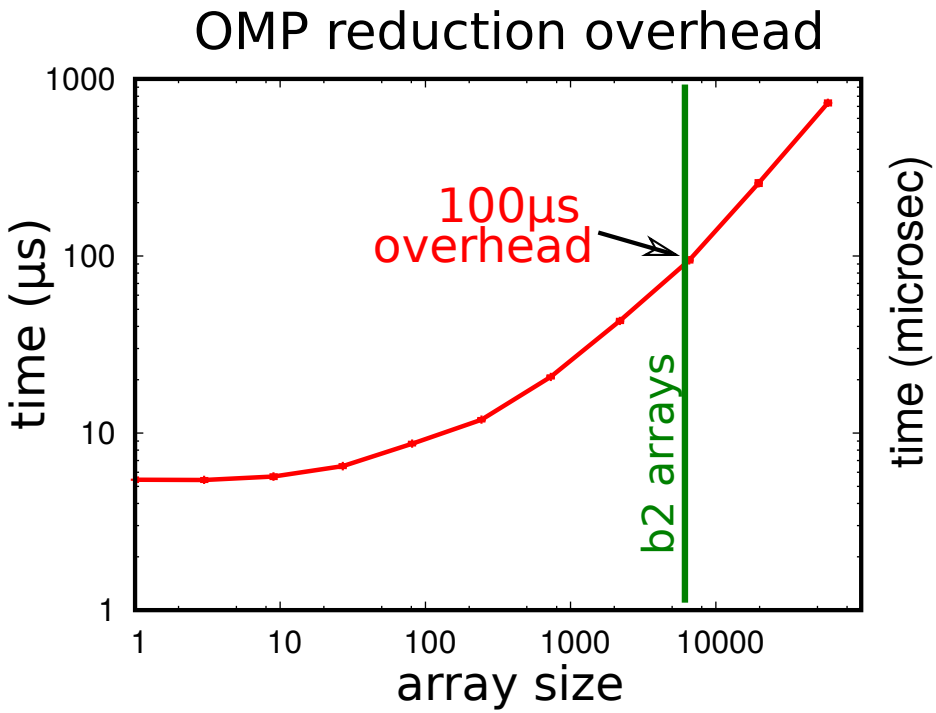
```
for ( j = 0 ; j < innerreps ; j ++) {
#pragma omp parallel for
    for ( i = 0 ; i < nthreads ; i ++) {
        delay ( delaylength ) ;
    }
}
```

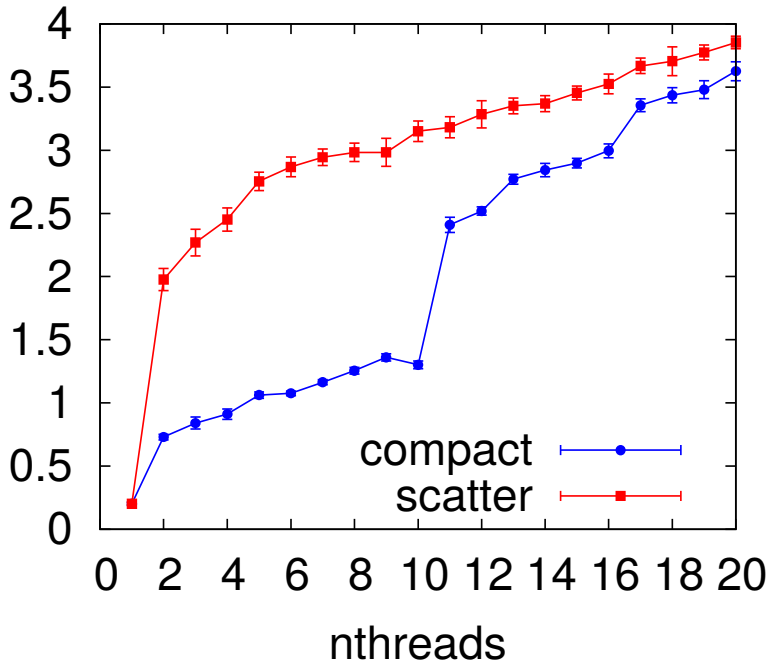


EPCC OpenMP Microbenchmarks J.M. Bull et. al

OpenMP parallelization

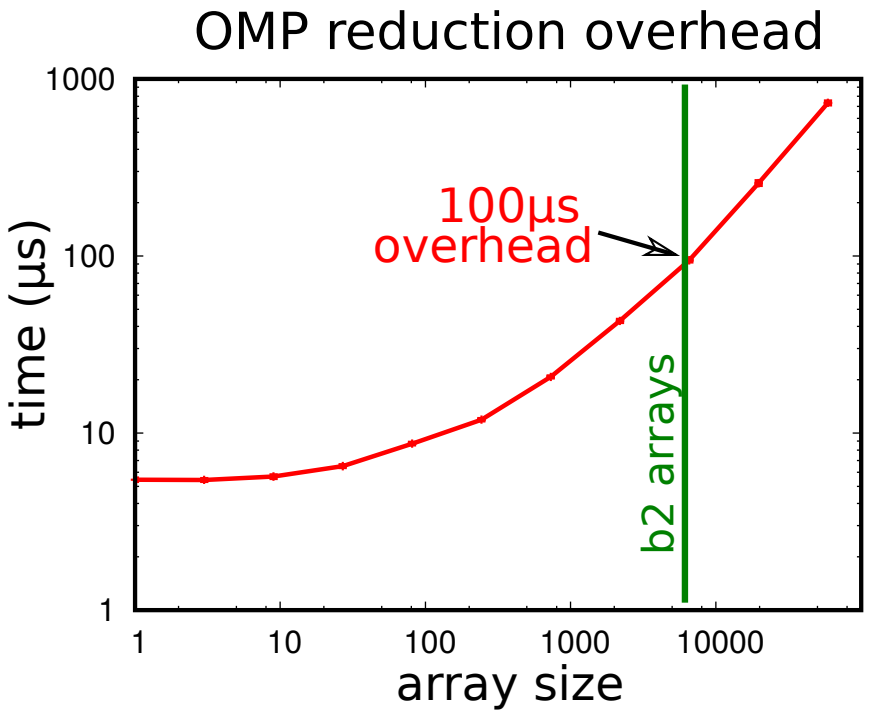
- Overhead for creating parallel region on Hydra Ivy Bridge
- 4 μs ~ 0.086 Mflop (1 core) ~ 41 kb
- MIC – more threads, more overhead
- Reduction over arrays can be costly

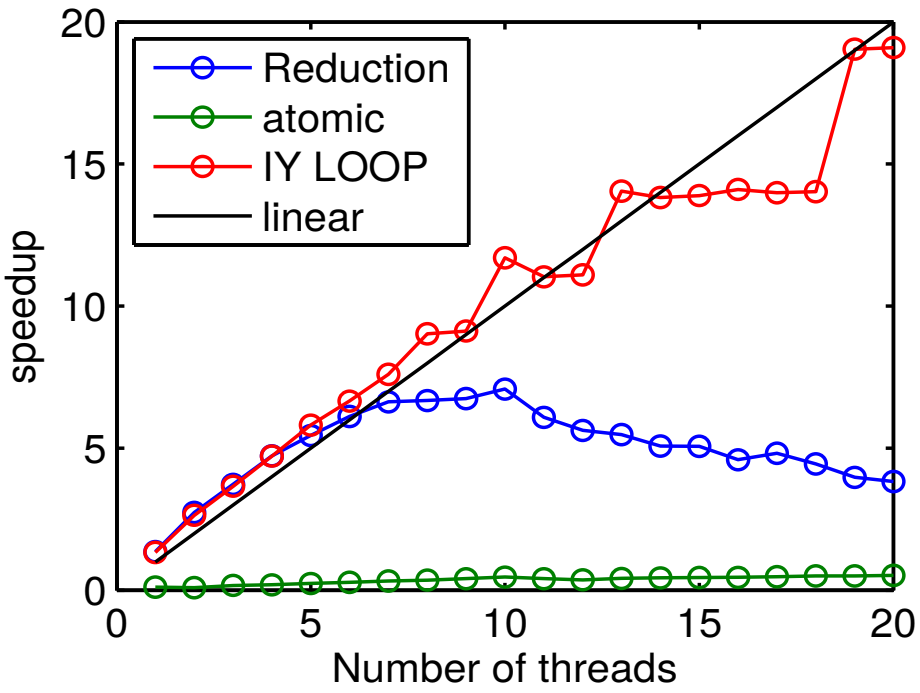




OpenMP parallelization

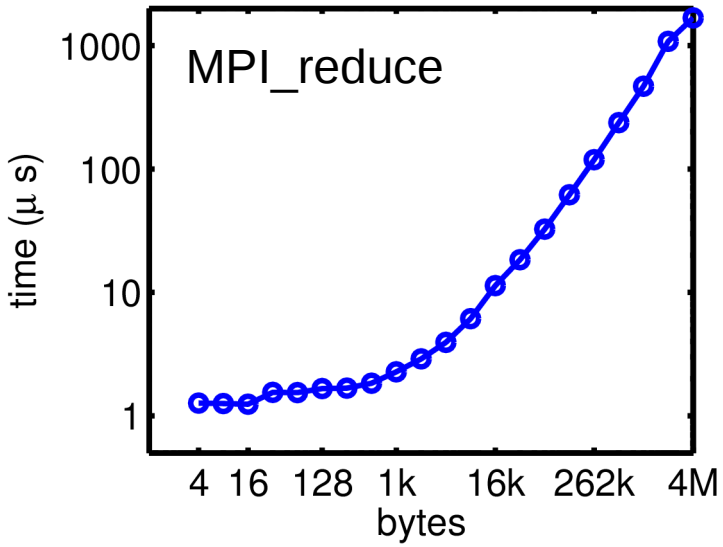
- Overhead for creating parallel region on Hydra Ivy Bridge
- 4 μs ~ 0.086 Mflop (1 core) ~ 41 kb
- MIC – more threads, more overhead
- Reduction over arrays can be costly



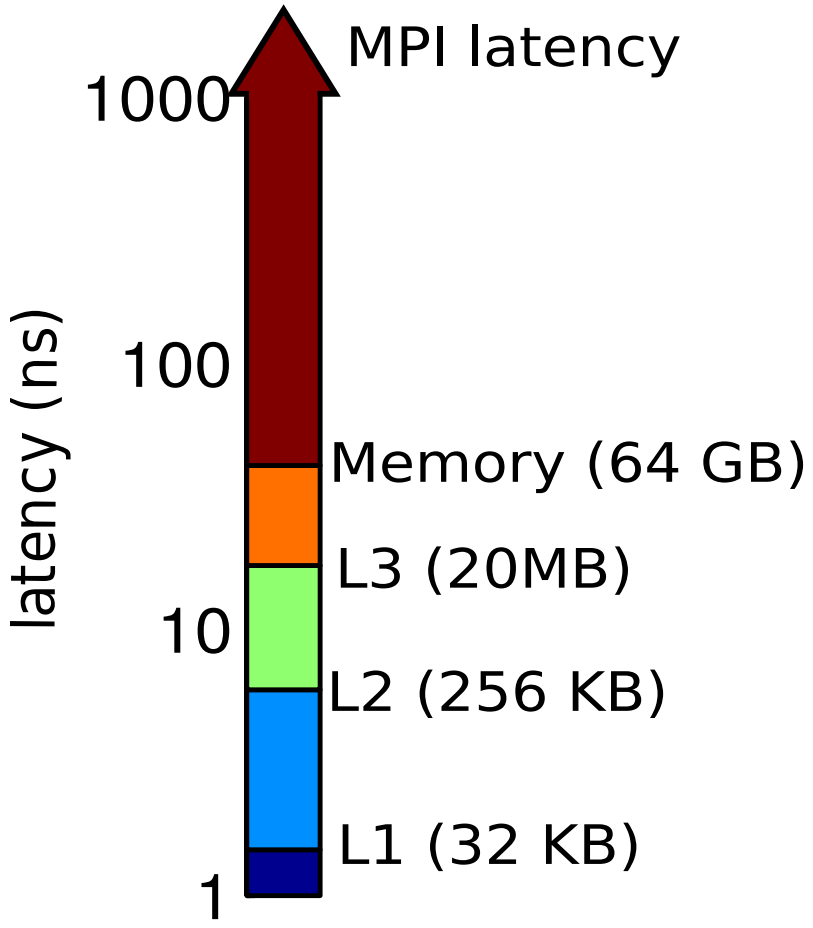


MPI bandwidth and overhead

- Infiniband
- Bandwidth ~ 5 GB/s
- Latency ~ µs
- Communication overhead depends on Nproc and message size



Intel MPI benchmark



Summary

- Speedup is limited by
  - CPU peak Gflop/s or memory bandwidth bottlenecks
  - Overhead of parallelization constructs
  - Amdahl's law
- Roofline model – visualization
  - Helpful for evaluating optimization efforts
  - Can guide further optimization