Performance & Roofline Model

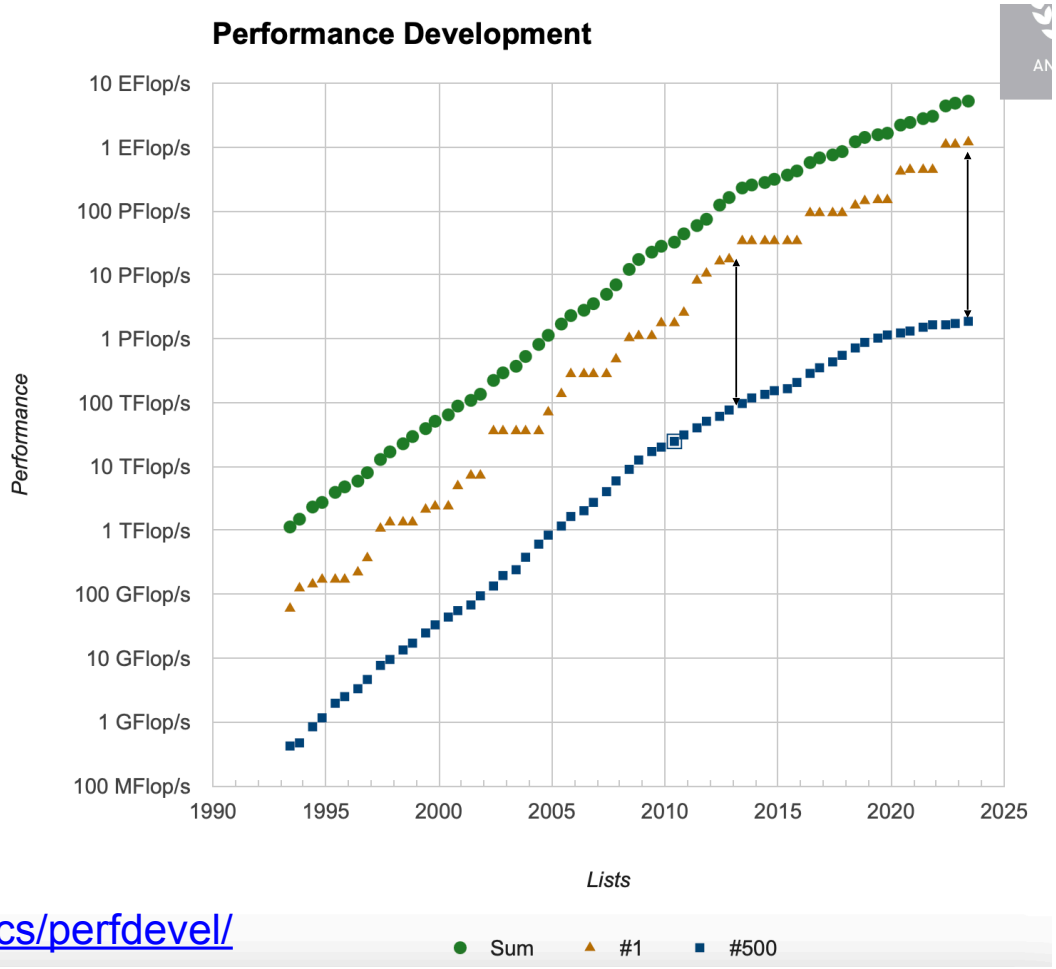
Andrea Bartolini <a.bartolini@unibo.it>

(Architettura dei) Calcolatori Elettronici, 2023/2024

What makes a computer a supercomputer?

What makes a computer a supercomputer?

- Top500:



https://top500.org/statistics/perfdevel/

Top500

Rpeak:
Theoretical Maximum Performance
*# DP Flops/cycle * # DP FPUs * Nominal Core's Frequency * # Cores [TFLOPs/s]*

Rmax:
Measured DP Flating point operation per second durign an HPL/Linpack run [TFLOPs /s]

Cores: # of cores

Power: Power consumpiton during the HPL run [KW]

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,730,112	1,102.00	1,685.65	21,100
2	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
3	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,220,288	309.10	428.70	6,016
4	**Leonardo** - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, Atos EuroHPC/CINECA Italy	1,463,616	174.70	255.75	5,610
5	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148.60	200.79	10,096
6	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94.64	125.71	7,438
7	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC National Supercomputing Center in Wuxi China	10,649,600	93.01	125.44	15,371
8	**Perlmutter** - HPE Cray EX235n, AMD EPYC 7763 64C 2.45GHz, NVIDIA A100 SXM4 40 GB, Slingshot-10, HPE DOE/SC/LBNL/NERSC United States	761,856	70.87	93.75	2,589
9	**Selene** - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, Nvidia NVIDIA Corporation United States	555,520	63.46	79.22	2,646
10	**Tianhe-2A** - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000, NUDT National Super Computer Center in Guangzhou China	4,981,760	61.44	100.68	18,482

Top500 – Green500

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,730,112	1,102.00	1,685.65	21,100
2	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
3	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,220,288	309.10	428.70	6,016
4	**Leonardo** - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, Atos EuroHPC/CINECA Italy	1,463,616	174.70	255.75	5,610
5	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148.60	200.79	10,096
6	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94.64	125.71	7,438
7	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC National Supercomputing Center in Wuxi China	10,649,600	93.01	125.44	15,371
8	**Perlmutter** - HPE Cray EX235n, AMD EPYC 7763 64C 2.45GHz, NVIDIA A100 SXM4 40 GB, Slingshot-10, HPE DOE/SC/LBNL/NERSC United States	761,856	70.87	93.75	2,589
9	**Selene** - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, Nvidia NVIDIA Corporation United States	555,520	63.46	79.22	2,646
10	**Tianhe-2A** - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000, NUDT National Super Computer Center in Guangzhou China	4,981,760	61.44	100.68	18,482

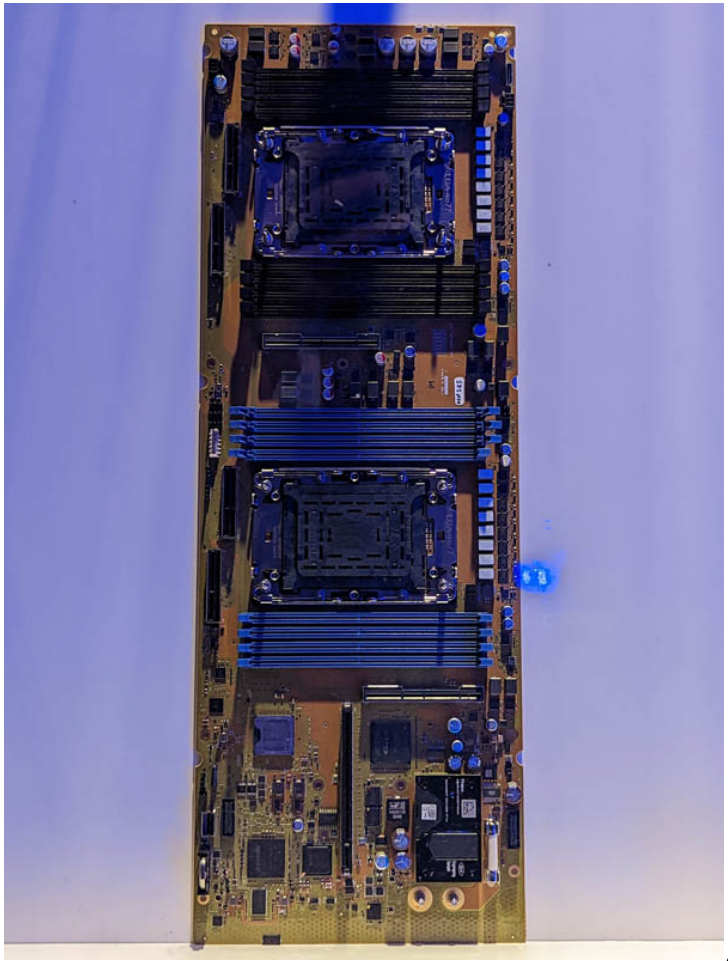
Green500 Data

Rank	TOP500 Rank	System	Cores	Rmax (PFlop/s)	Power (kW)	Energy Efficiency (GFlops/watts)
1	405	**Henri** - Lenovo ThinkSystem SR670 V2, Intel Xeon Platinum 8362 2800Mhz (32C), NVIDIA H100 80GB PCIe, Infiniband HDR, Lenovo Flatiron Institute United States	5,920	2.04	31	65.091
2	32	**Frontier TDS** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	120,832	19.20	309	62.684
3	11	**Adastra** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE Grand Equipement National de Calcul Intensif - Centre Informatique National de l'Enseignement Suprieur (GENCI-	319,072	46.10	921	58.021
6	1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,730,112	1,102.00	21,100	52.227
7	3	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,220,288	309.10	6,016	51.382
8	159	**ATOS THX.A.B** - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, Atos Atos France	25,056	3.50	86	41.411

What makes a computer an HPC node?

- Find the difference:

https://www.intel.com/content/www/us/en/developer/articles/technical/how-to-set-up-your-intel-nuc-kit.html

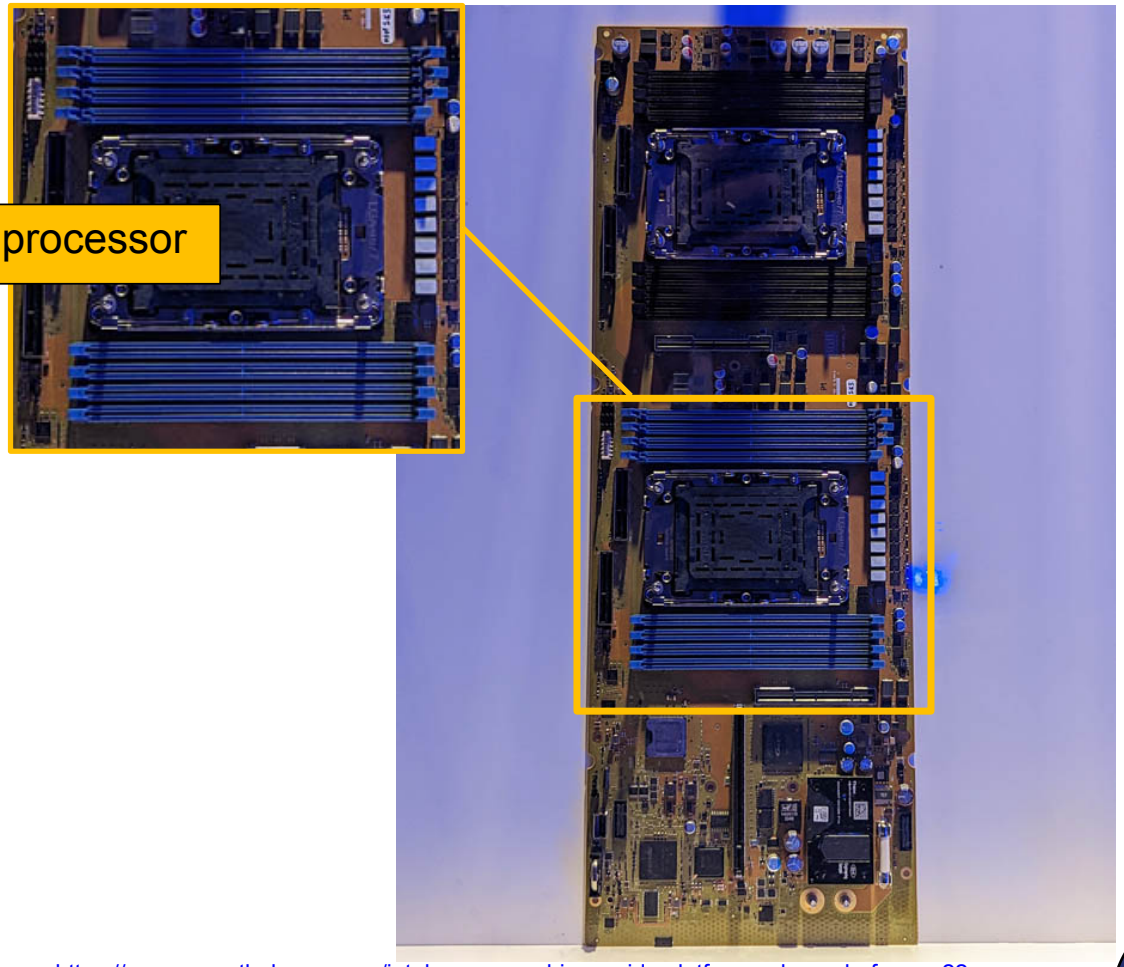


https://www.servethehome.com/intel-xeon-sapphire-rapids-platforms-shown-before-sc22-supermicro-asus-qct-lenovo-atos-coolit-hpe/atos-bullsequana-x3410-motherboard-at-sc22/

What makes a computer an HPC node?

- Find the difference:

https://www.intel.com/content/www/us/en/developer/articles/technical/how-to-set-up-your-intel-nuc-kit.html



https://www.servethehome.com/intel-xeon-sapphire-rapids-platforms-shown-before-sc22-supermicro-asus-qct-lenovo-atos-coolit-hpe/atos-bullsequana-x3410-motherboard-at-sc22/

What makes a computer an HPC node?

Intel® Core™ i3-1220P Processor
12M Cache, up to 4.40 GHz

Download Specif

Essentials

Product Collection	12th Generation Intel® Core™ i3 Processors
Code Name	Products formerly Alder Lake
Vertical Segment	Mobile
Processor Number	i3-1220P
Lithography	Intel 7
Recommended Customer Price	$309.00

CPU Specifications

Total Cores	10
# of Performance-cores	2
# of Efficient-cores	8
Total Threads	12
Max Turbo Frequency	4.40 GHz
Performance-core Max Turbo Frequency	4.40 GHz
Efficient-core Max Turbo Frequency	3.30 GHz
Cache	12 MB Intel® Smart Cache

https://www.intel.com/content/www/us/en/products/sku/226257/intel-core-i31220p-processor-12m-cache-up-to-4-40-ghz/specifications.html

Intel® Xeon® Platinum 8470Q Processor
105M Cache, 2.10 GHz

Export specifications

Essentials

Product Collection	4th Generation Intel® Xeon® Scalable Processors
Code Name	Products formerly Sapphire Rapids
Vertical Segment	Server
Processor Number	8470Q
Lithography	Intel 7
Use Conditions	Server/Enterprise
Recommended Customer Price	$9410.00

CPU Specifications

Total Cores	52
Total Threads	104
Max Turbo Frequency	3.80 GHz
Processor Base Frequency	2.10 GHz
Cache	105 MB
Intel® UPI Speed	16 GT/s
Max # of UPI Links	4
TDP	350 W

https://ark.intel.com/content/www/us/en/ark/products/231727/intel-xeon-platinum-8470q-processor-105m-cache-2-10-ghz.html

What makes a computer an HPC node?

Intel® Core™ i3-1220P Processor
12M Cache, up to 4.40 GHz

Download Specif

Essentials

Product Collection	12th Generation Intel® Core™ i3 Processors
Code Name	Products formerly Alder Lake
Vertical Segment	Mobile
Processor Number	i3-1220P
Lithography	Intel 7
Recommended Customer Price	$309.00

CPU Specifications

Total Cores	10
# of Performance-cores	2
# of Efficient-cores	8
Total Threads	12
Max Turbo Frequency	4.40 GHz
Performance-core Max Turbo Frequency	4.40 GHz
Efficient-core Max Turbo Frequency	3.30 GHz
Cache	12 MB Intel® Smart Cache
Instruction Set	64-bit
Instruction Set Extensions	Intel® SSE4.1, Intel® SSE4.2, Intel® AVX2

Intel® Xeon® Platinum 8470Q Processor
105M Cache, 2.10 GHz

Export specifications

Essentials

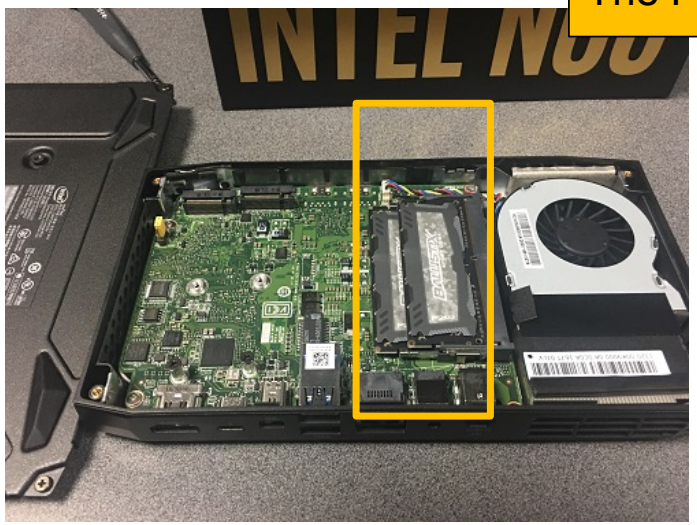
Product Collection	4th Generation Intel® Xeon® Scalable Processors
Code Name	Products formerly Sapphire Rapids
Vertical Segment	Server
Processor Number	8470Q
Lithography	Intel 7
Use Conditions	Server/Enterprise
Recommended Customer Price	$9410.00

CPU Specifications

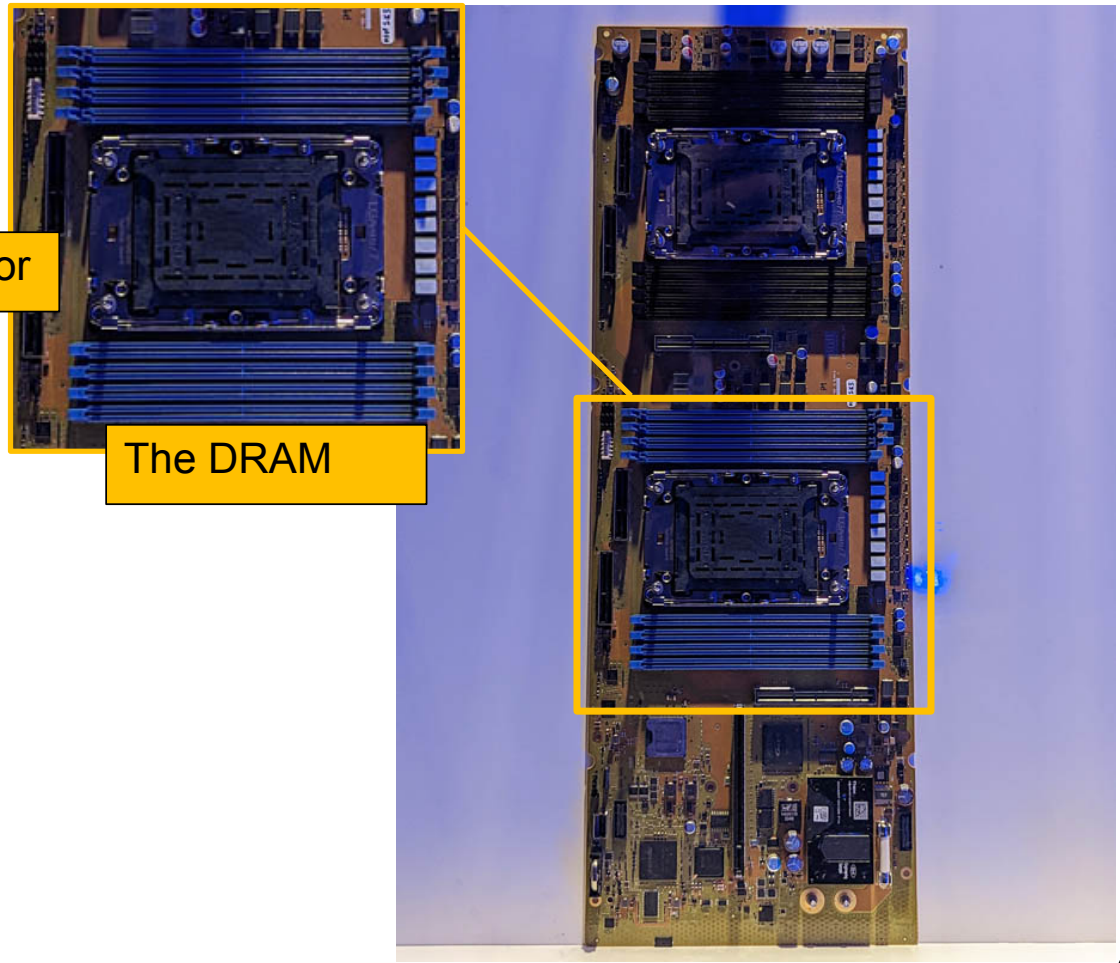
Total Cores	52
Total Threads	104
Max Turbo Frequency	3.80 GHz
Processor Base Frequency	2.10 GHz
Cache	105 MB
Intel® UPI Speed	16 GT/s
Max # of UPI Links	4
TDP	350 W
Instruction Set Extensions	Intel® AMX, Intel® SSE4.2, Intel® AVX, Intel® AVX2, Intel® AVX-512
# of AVX-512 FMA Units	2

What makes a computer an HPC node?

- Find the difference:



https://www.intel.com/content/www/us/en/developer/articles/technical/how-to-set-up-your-intel-nuc-kit.html



https://www.servethehome.com/intel-xeon-sapphire-rapids-platforms-shown-before-sc22-supermicro-asus-qct-lenovo-atos-coolit-hpe/atos-bullsequana-x3410-motherboard-at-sc22/

What makes a computer an HPC node?

Intel® Core™ i3-1220P Processor
12M Cache, up to 4.40 GHz

Download Specif

Essentials

Product Collection	12th Generation Intel® Core™ i3 Processors
Code Name	Products formerly Alder Lake
Vertical Segment	Mobile
Processor Number	i3-1220P
Lithography	Intel 7
Recommended Customer Price	$309.00

Memory Specifications

Max Memory Size (dependent on memory type)	64 GB
Memory Types	Up to DDR5 4800 MT/s Up to DDR4 3200 MT/s Up to LPDDR5 5200 MT/s Up to LPDDR4x 4267 MT/s
Max # of Memory Channels	2
ECC Memory Supported ‡	No

CPU

Total	
# of P	
# of E	
Total	
Max T	
Performance-core Max Turbo Frequency	4.40 GHz
Efficient-core Max Turbo Frequency	3.30 GHz
Cache	12 MB Intel® Smart Cache

Instruction Set	64-bit
Instruction Set Extensions	Intel® SSE4.1, Intel® SSE4.2, Intel® AVX2

Intel® Xeon® Platinum 8470Q Processor
105M Cache, 2.10 GHz

Export specifications

Essentials

Product Collection	4th Generation Intel® Xeon® Scalable Processors
Code Name	Products formerly Sapphire Rapids
Vertical Segment	Server
Processor Number	8470Q
Lithography	Intel 7

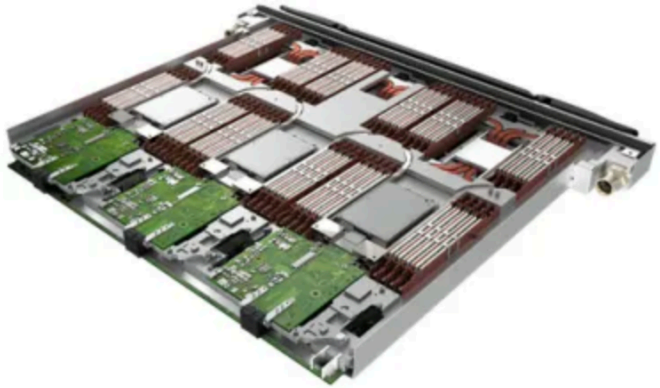
Memory Specifications

Max Memory Size (dependent on memory type)	4 TB
Memory Types	Up to DDR5 4800 MT/s 1DPC Up to DDR5 4400 MT/s 2DPC
Max # of Memory Channels	8
ECC Memory Supported ‡	Yes

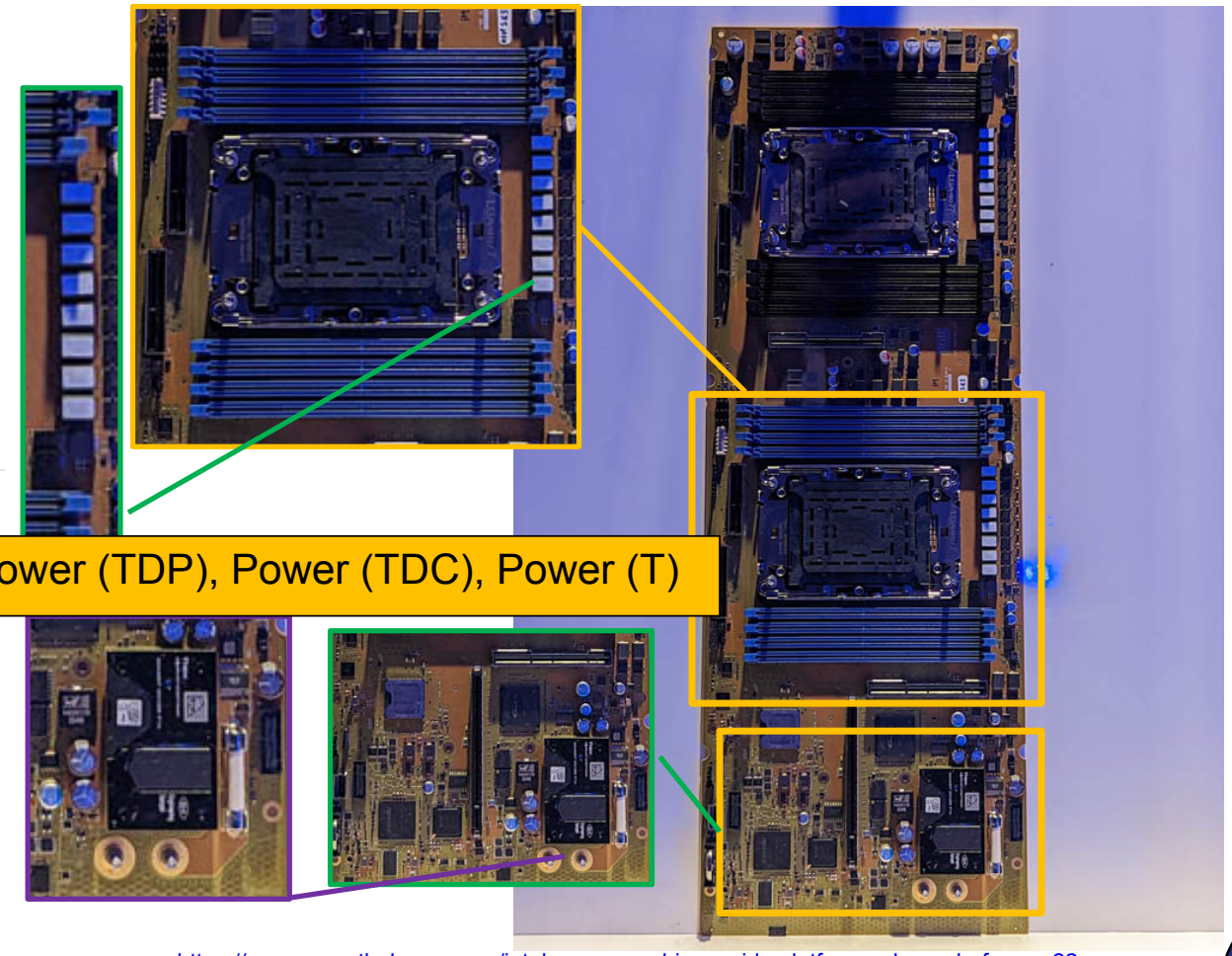
Cache	105 MB
Intel® UPI Speed	16 GT/s
Max # of UPI Links	4
TDP	350 W

Instruction Set Extensions	Intel® AMX, Intel® SSE4.2, Intel® AVX, Intel® AVX2, Intel® AVX-512
# of AVX-512 FMA Units	2

What makes a computer an HPC node?



https://www.intel.com/content/www/us/en/developer/articles/technical/how-to-set-up-your-intel-nuc-kit.html



Power (TDP), Power (TDC), Power (T)

https://www.servethehome.com/intel-xeon-sapphire-rapids-platforms-shown-before-sc22-supermicro-asus-qct-lenovo-atos-coolit-hpe/atos-bullsequana-x3410-motherboard-at-sc22/

What makes a computer an HPC node?

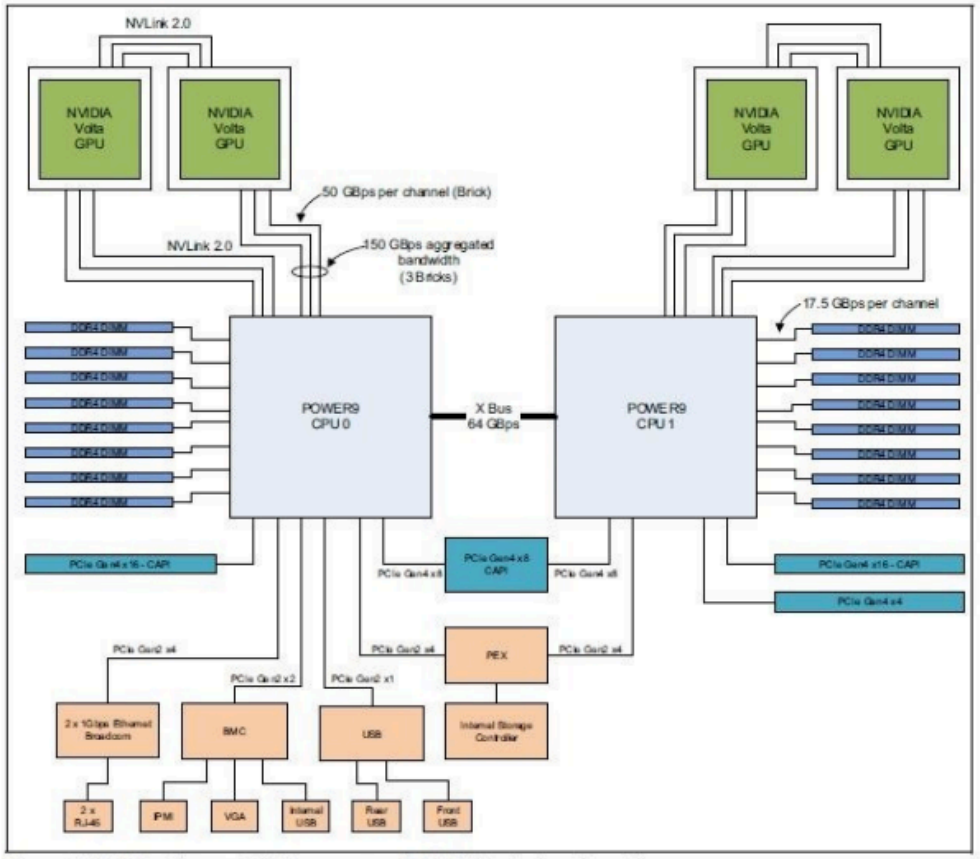
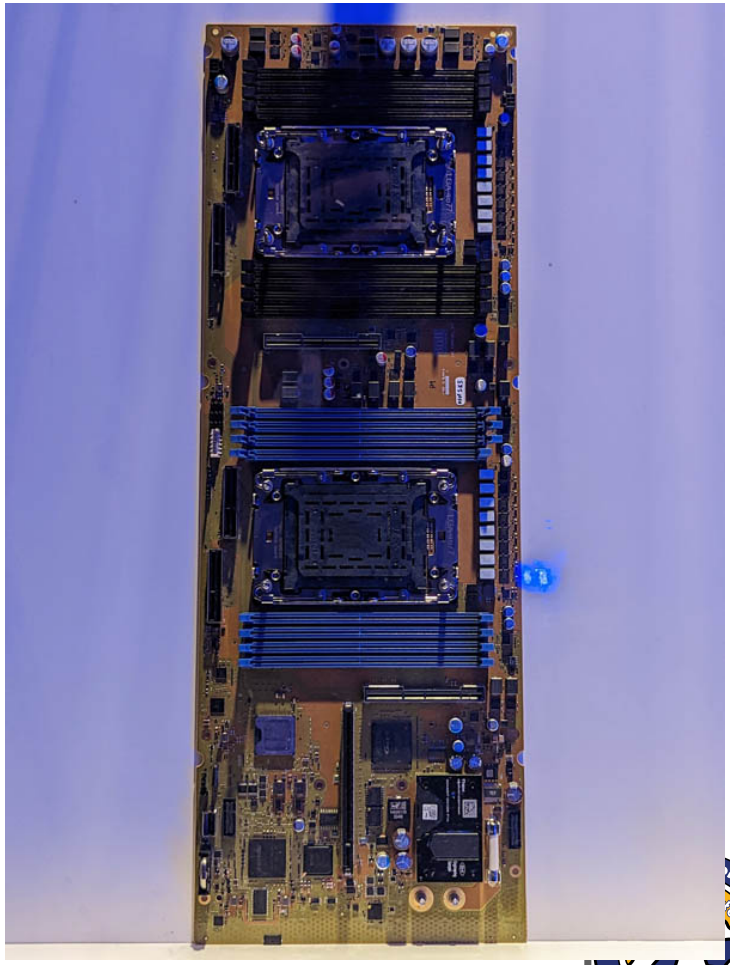


Figure 2-5 The Power AC922 server model GTH logical system diagram

[Marconi100]



How do we measure performance?

Top500 -> DPFLOP/s

Rpeak:
Theoretical Maximum Performance
*# DP Flops/cycle * # DP FPUs * Nominal Core's Frequency * # Cores [TFLOPs/s]*

Rmax:
Measured DP Flating point operation per second durign an HPL/Linpack run [TFLOPs /s]

Cores: # of cores

Power: Power consumpiton during the HPL run [KW]

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,730,112	1,102.00	1,685.65	21,100
2	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
3	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,220,288	309.10	428.70	6,016
4	**Leonardo** - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, Atos EuroHPC/CINECA Italy	1,463,616	174.70	255.75	5,610
5	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148.60	200.79	10,096
6	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94.64	125.71	7,438
7	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC National Supercomputing Center in Wuxi China	10,649,600	93.01	125.44	15,371
8	**Perlmutter** - HPE Cray EX235n, AMD EPYC 7763 64C 2.45GHz, NVIDIA A100 SXM4 40 GB, Slingshot-10, HPE DOE/SC/LBNL/NERSC United States	761,856	70.87	93.75	2,589
9	**Selene** - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, Nvidia NVIDIA Corporation United States	555,520	63.46	79.22	2,646
10	**Tianhe-2A** - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000, NUDT National Super Computer Center in Guangzhou China	4,981,760	61.44	100.68	18,482

HPL Algorithm

- Solution of a system of linear algebraic equation

$$Ax=b \quad ; \qquad A = NxN$$

- Method: Gaussian Elimination with partial pivoting

1. LU Factorization using GE:

$$A = LU \quad ; \; L=lower\ triangular,\ U=upper\ triangular$$

2. Substituting:

$$LUx=b$$

3. Solve (forward substitution, applied during the factorization)

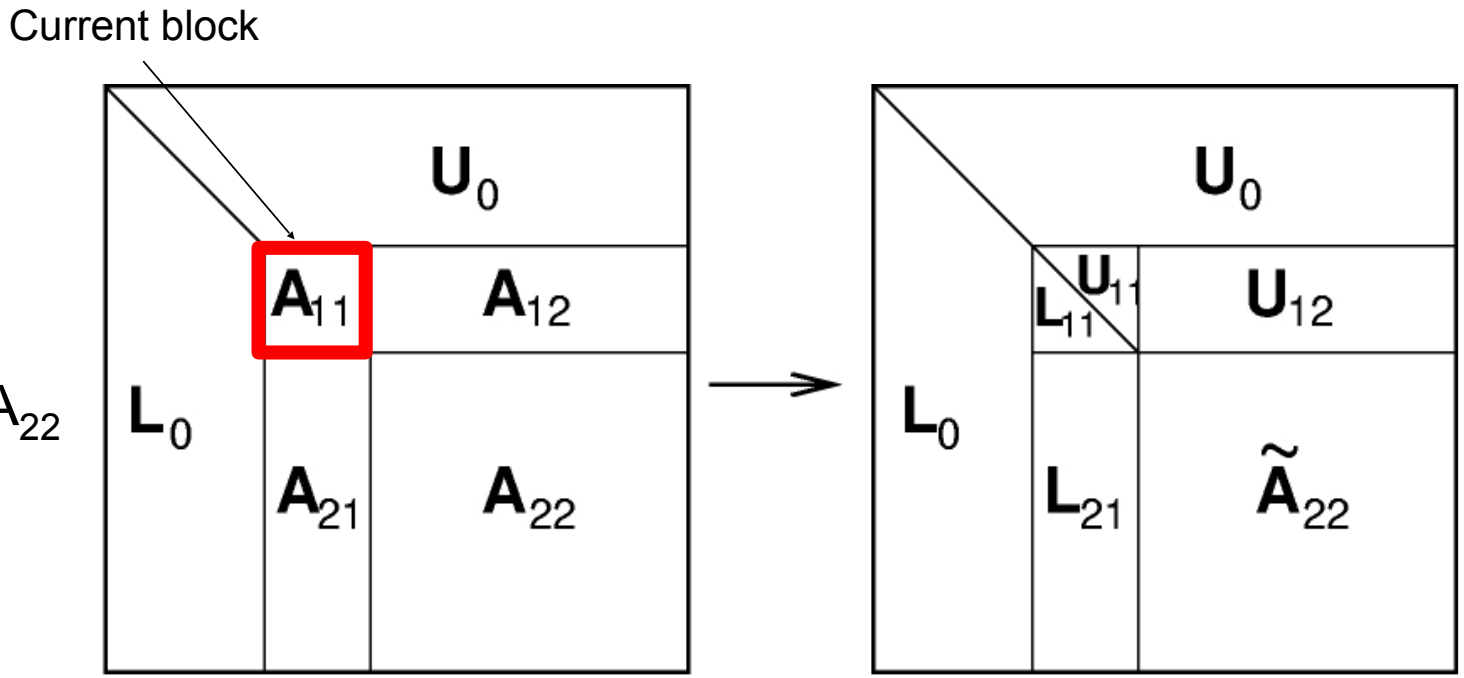
$$Ly = b \quad ; \quad -> y$$

4. Solve (backward substitution)

$$Ux = y \quad ; \quad -> x$$

LU Factorization

- Factorize the Current block
- Update A_{21}, A_{12}, and A_{22}



* The parallel implementation of the algorithm requires data communication between the processes that own the matrix blocks

HPL Cost

Given a problem size (N), the number of DP Floating Point Operations executed can be estimated as:

- LU decomposition: $2/3*N^3$
- Backward substitution: $2N^2$
- Backward substitution: $2N^2$
- Data used: $\sim N^2$

STREAM

- The STREAM benchmark is a simple synthetic benchmark program that measures sustainable memory bandwidth (in MB/s) and the corresponding computation rate for simple vector kernels.
- https://www.cs.virginia.edu/stream/

Exercise 1a: Execute

- Execute from command line, using default settings

```
$ ./stream


-------------------------------------------------------------
STREAM version $Revision: 5.10 $
-------------------------------------------------------------
This system uses 8 bytes per array element.
-------------------------------------------------------------
Array size = 10000000 (elements), Offset = 0 (elements)
Memory per array = 76.3 MiB (= 0.1 GiB).
Total memory required = 228.9 MiB (= 0.2 GiB).
Each kernel will be executed 20 times.
 The *best* time for each kernel (excluding the first iteration)
 will be used to compute the reported bandwidth.
-------------------------------------------------------------
Number of Threads requested = 4
Number of Threads counted = 4
-------------------------------------------------------------
Your clock granularity/precision appears to be 1 microseconds.
Each test below will take on the order of 104441 microseconds.
   (= 104441 clock ticks)
Increase the size of the arrays if this shows that
you are not getting at least 20 clock ticks per test.
-------------------------------------------------------------
WARNING -- The above is only a rough guideline.
For best results, please be sure you know the
precision of your system timer.
-------------------------------------------------------------
Function    Best Rate MB/s  Avg time     Min time     Max time
Copy:           1210.8     0.132802     0.132148     0.133741
Scale:          1038.0     0.155015     0.154146     0.156040
Add:            1139.4     0.212405     0.210631     0.215261
Triad:          1139.1     0.212451     0.210701     0.214228
-------------------------------------------------------------
Solution validates: avg error less than 1.000000e-13 on all three arrays
-------------------------------------------------------------
```

Results

STREAM

- The STREAM benchmark basically loads an array of numbers in memory and performs some basic operations on it
- The four benchmarks executed by STREAM are:

Name	Kernel	Bytes/Iteration	FLOPS/Iteration
COPY	a[i] = b[i]	16	0
SCALE	a[i] = q*b[i]	16	1
SUM	a[i] = b[i] + c[i]	24	1
TRIAD	a[i] = b[i] + q*c[i]	24	2

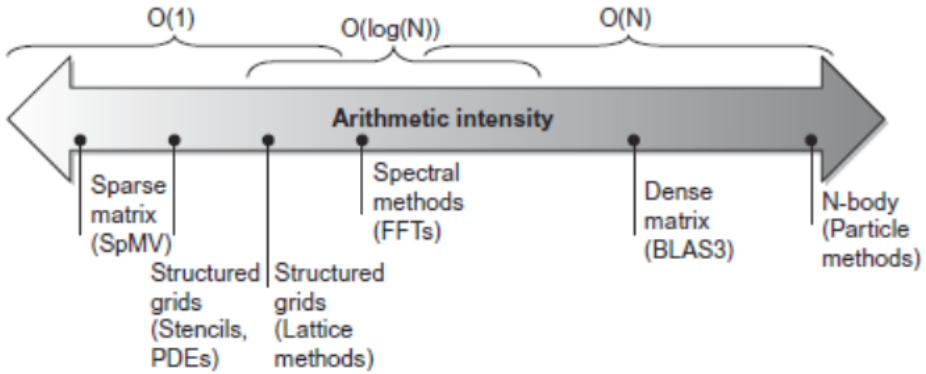
- COPY: measures the transfer rate with no arithmetic
- SCALE: adds a scalar operation
- SUM: sums two operands
- TRIAD: defines a multiply-add operation.
  - Very common in many basic computation kernels
  - Recent CPUs offer a dedicated extension to their instruction set called Fused Multiply-Add (FMA)

STREAM

- STREAM main parameters:
  - –STREAM_ARRAY_SIZE
    - define the maximum number of array elements
    - it is suggested to be four times the sum of all level caches
  - –NTIMES
    - the number of times each benchmark is run

Roofline Performance Model

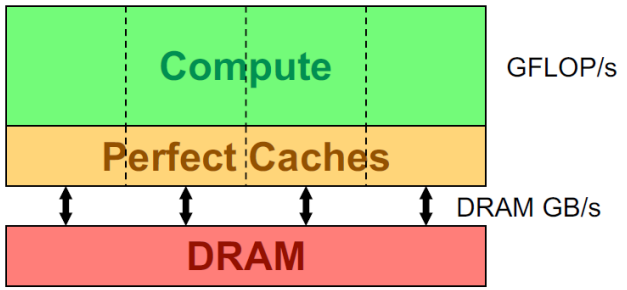
- Basic idea:
  - Plot peak floating-point throughput as a function of arithmetic intensity
  - Ties together floating-point performance and memory performance for a target machine
- Arithmetic intensity
  - Floating-point operations per byte read



Roofline (DRAM)

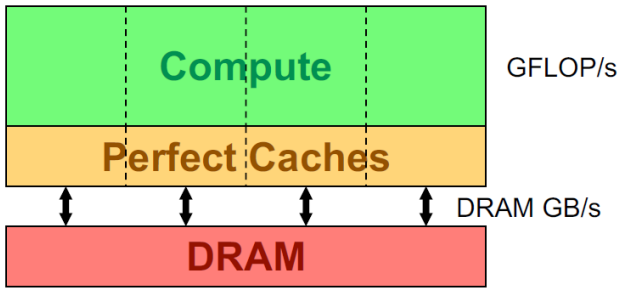
- Any given loop nest will perform:
  - Computation (e.g. FLOPs)
  - Communication (e.g. moving data to/from DRAM)
- With perfect overlap of communication and computation...
  - Run time is determined by whichever is greater

$$\text{Time} = \max \begin{cases} \#\text{FLOPs} / \text{Peak GFLOP/s} \\ \#\text{Bytes} / \text{Peak GB/s} \end{cases}$$



Roofline (DRAM)

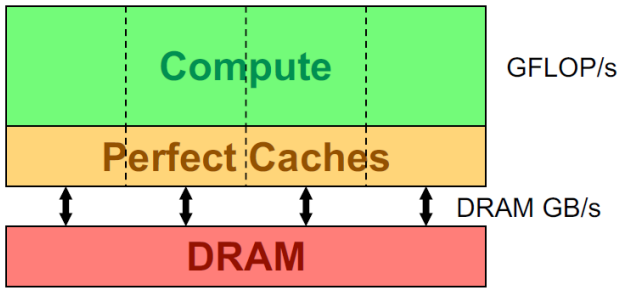
- Any given loop nest will perform:
  - Computation (e.g. FLOPs)
  - Communication (e.g. moving data to/from DRAM)
- With perfect overlap of communication and computation...
  - Run time is determined by whichever is greater



$$\frac{\text{Time}}{\#\text{FLOPs}} = \max \begin{cases} 1 \,/\, \text{Peak GFLOP/s} \\ \#\text{Bytes} \,/\, \#\text{FLOPs} \,/\, \text{Peak GB/s} \end{cases}$$

Roofline (DRAM)

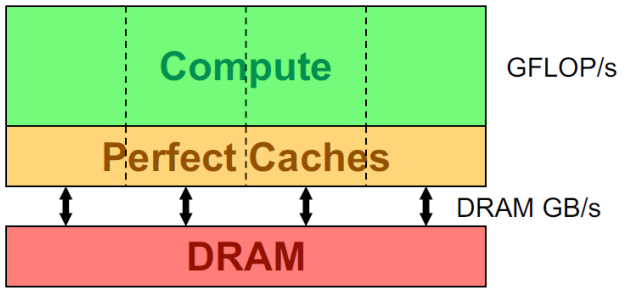
- Any given loop nest will perform:
  - Computation (e.g. FLOPs)
  - Communication (e.g. moving data to/from DRAM)
- With perfect overlap of communication and computation...
  - Run time is determined by whichever is greater



$$\frac{\#\text{FLOPs}}{\text{Time}} = \min \begin{cases} \text{Peak GFLOP/s} \\ (\#\text{FLOPs} / \#\text{Bytes}) * \text{Peak GB/s} \end{cases}$$

Roofline (DRAM)

- Any given loop nest will perform:
  - Computation (e.g. FLOPs)
  - Communication (e.g. moving data to/from DRAM)
- With perfect overlap of communication and computation...
  - Run time is determined by whichever is greater



$$\text{GFLOP/s} = \min \begin{cases} \text{Peak GFLOP/s} \\ \text{AI} * \text{Peak GB/s} \end{cases}$$

AI (Arithmetic Intensity) = FLOPs / Bytes (as presented to DRAM)

Aritmetic Intensity

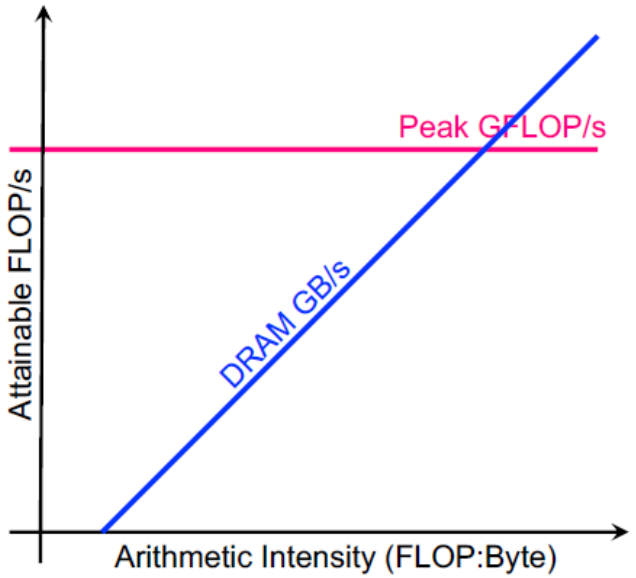
- Measure of data locality (data reuse)
- Ratio of **Total Flops** performed to **Total Bytes** moved
- For the DRAM Roofline…
  - Total Bytes to/from DRAM
  - Includes all cache and prefetcher effects
  - Can be very different from total loads/stores (bytes requested)
  - Equal to ratio of sustained GFLOP/s to sustained GB/s (time cancels)

Roofline Model

$$\text{GFLOP/s} = \min \begin{cases} \text{Peak GFLOP/s} \\ \text{AI} * \text{Peak GB/s} \end{cases}$$

AI (Arithmetic Intensity) = FLOPs / Bytes (moved to/from DRAM)

- Plot Roofline bound using Arithmetic Intensity as the x-axis
- **Log-log scale** makes it easy to doodle, extrapolate performance along Moore's Law, etc…

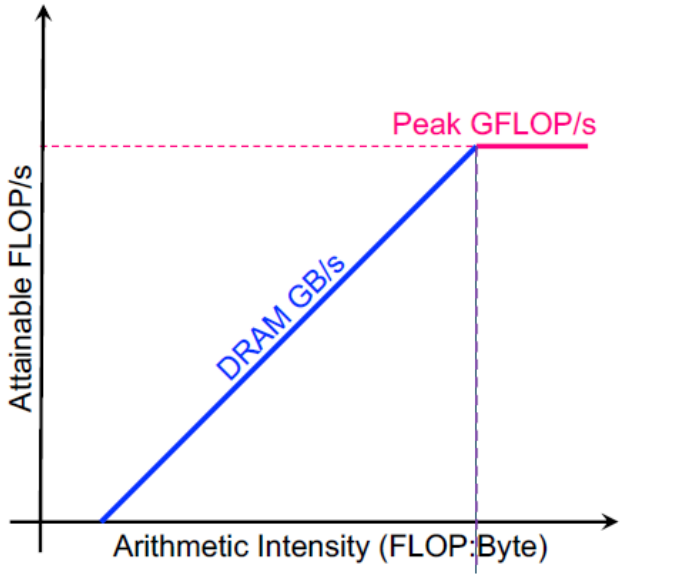


Roofline Model

$$\text{GFLOP/s} = \min \begin{cases} \text{Peak GFLOP/s} \\ \text{AI} * \text{Peak GB/s} \end{cases}$$

AI (Arithmetic Intensity) = FLOPs / Bytes (moved to/from DRAM)

- Plot Roofline bound using Arithmetic Intensity as the x-axis
- **Log-log scale** makes it easy to doodle, extrapolate performance along Moore's Law, etc…



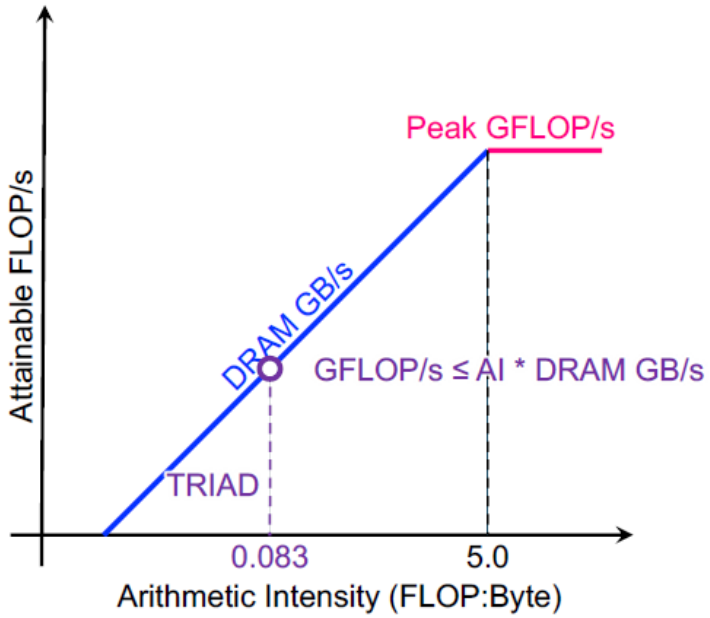
Transition @ AI == Peak GFLOP/s / Peak GB/s == 'Machine Balance'

Roofline Examples #1

- Typical machine balance is 5-10 FLOPs per byte…
  - 40-80 FLOPs per double to exploit compute capability
  - Artifact of technology and money
  - **Unlikely to improve**
- Consider STREAM Triad…

```
#pragma omp parallel for
for(i=0;i<N;i++){
  Z[i] = X[i] + alpha*Y[i];
}
```

  - 2 FLOPs per iteration
  - Transfer 24 bytes per iteration (read X[i], Y[i], write Z[i])
  - **AI = 0.083 FLOPs per byte == Memory bound**

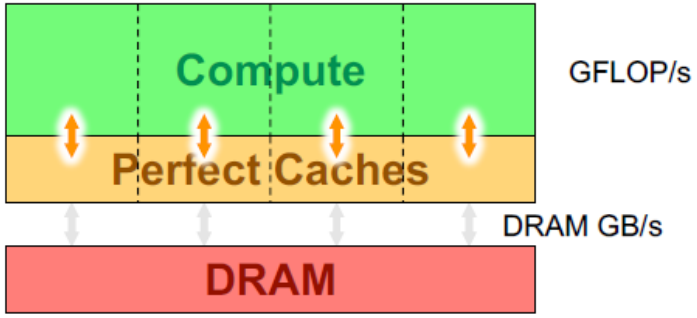


Roofline Examples #2

- Conversely, 7-point constant coefficient stencil…
  - 7 FLOPs
  - 8 memory references (7 reads, 1 store) per point
  - **AI = 7 / (8*8) = 0.11 FLOPs per byte**

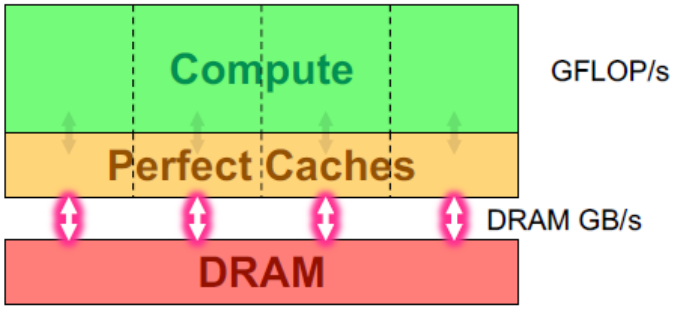
    (measured at the L1)

```
#pragma omp parallel for
for(k=1;k<dim+1;k++){
for(j=1;j<dim+1;j++){
for(i=1;i<dim+1;i++){
 new[k][j][i] = -6.0*old[k  ][j  ][i  ]
                + old[k  ][j  ][i-1]
                + old[k  ][j  ][i+1]
                + old[k  ][j-1][i  ]
                + old[k  ][j+1][i  ]
                + old[k-1][j  ][i  ]
                + old[k+1][j  ][i  ]
}}}
```



Roofline Examples #2

- Conversely, 7-point constant coefficient stencil…
  - 7 FLOPs
  - 8 memory references (7 reads, 1 store) per point
  - **Ideally, cache will filter all but 1 read and 1 write per point**

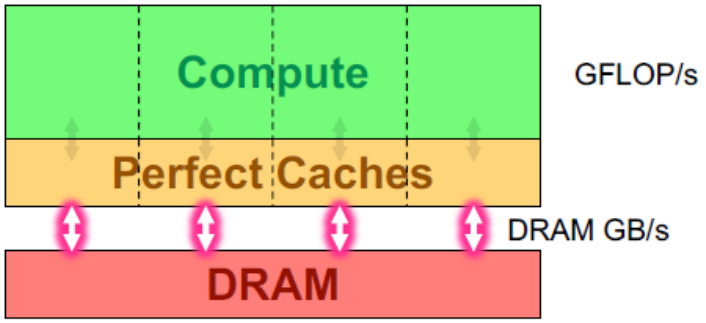



```
#pragma omp parallel for
for(k=1;k<dim+1;k++){
for(j=1;j<dim+1;j++){
for(i=1;i<dim+1;i++){
 new[k][j][i] = -6.0*old[k  ][j  ][i  ]
                + old[k  ][j  ][i-1]
                + old[k  ][j  ][i+1]
                + old[k  ][j-1][i  ]
                + old[k  ][j+1][i  ]
                + old[k-1][j  ][i  ]
                + old[k+1][j  ][i  ]
}}}
```

Roofline Examples #2

- Conversely, 7-point constant coefficient stencil…
  - 7 FLOPs
  - 8 memory references (7 reads, 1 store) per point
  - Ideally, cache will filter all but 1 read and 1 write per point
  - ➢ **7 / (8+8) = 0.44 FLOPs per byte (DRAM)**

```
#pragma omp parallel for
for(k=1;k<dim+1;k++){
for(j=1;j<dim+1;j++){
for(i=1;i<dim+1;i++){
  new[k][j][i] = -6.0*old[k  ][j  ][i  ]
                 + old[k  ][j  ][i-1]
                 + old[k  ][j  ][i+1]
                 + old[k  ][j-1][i  ]
                 + old[k  ][j+1][i  ]
                 + old[k-1][j  ][i  ]
                 + old[k+1][j  ][i  ];
}}}
```

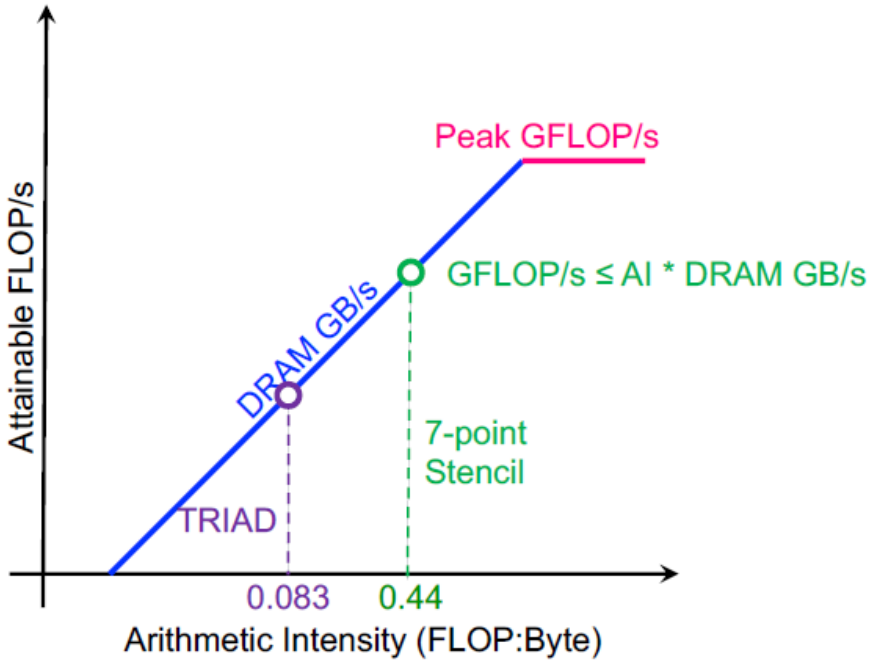


Roofline Examples #2

- Conversely, 7-point constant coefficient stencil…
  - 7 FLOPs
  - 8 memory references (7 reads, 1 store) per point
  - Ideally, cache will filter all but 1 read and 1 write per point
  - **7 / (8+8) = 0.44 FLOPs per byte (DRAM)**

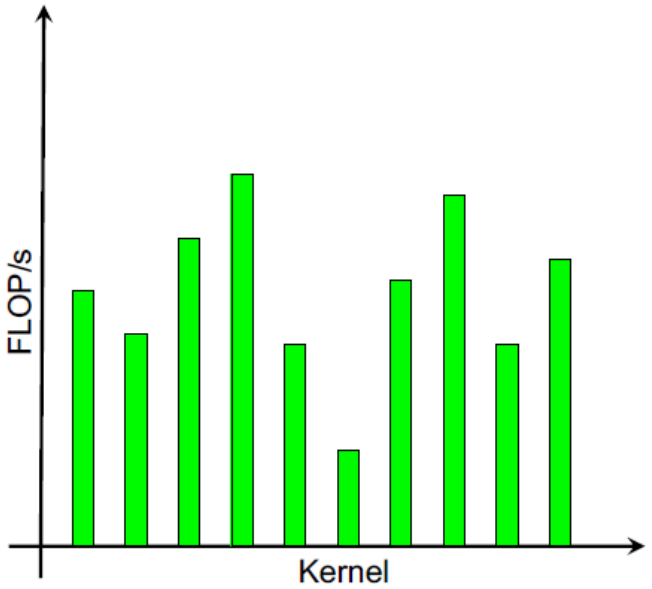
    == memory bound, but 5x the FLOP rate as TRIAD

```
#pragma omp parallel for
for(k=1;k<dim+1;k++){
for(j=1;j<dim+1;j++){
for(i=1;i<dim+1;i++){
  new[k][j][i] = -6.0*old[k  ][j  ][i  ]
                     + old[k  ][j  ][i-1]
                     + old[k  ][j  ][i+1]
                     + old[k  ][j-1][i  ]
                     + old[k  ][j+1][i  ]
                     + old[k-1][j  ][i  ]
                     + old[k+1][j  ][i  ];
}}}
```



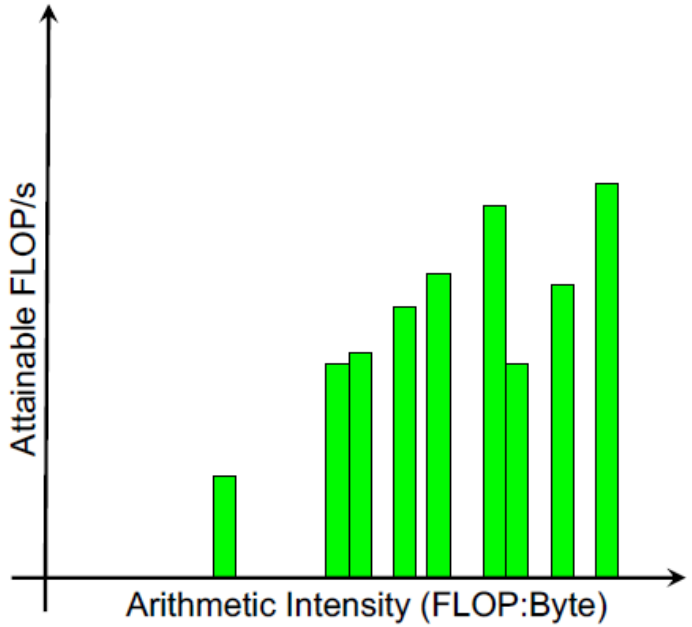
What is “Good” Performance?

- Think back to our mix of loop nests (benchmarks)…



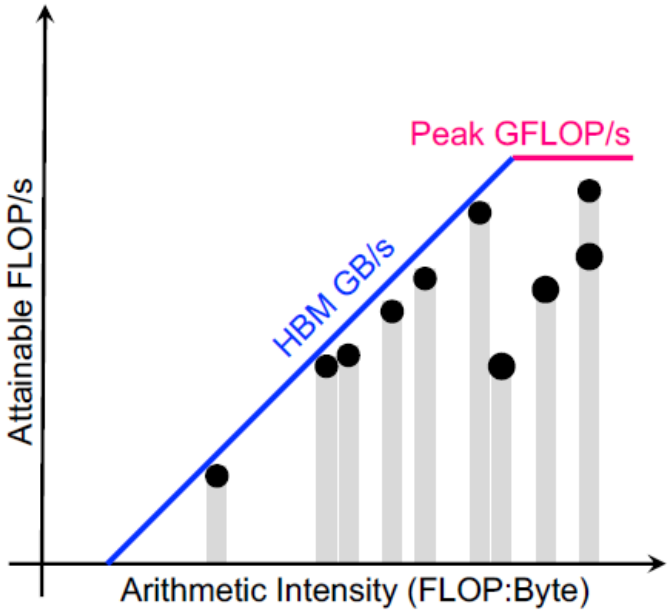
What is “Good” Performance?

- Think back to our mix of loop nests (benchmarks)
- We can sort kernels by their arithmetic intensity…



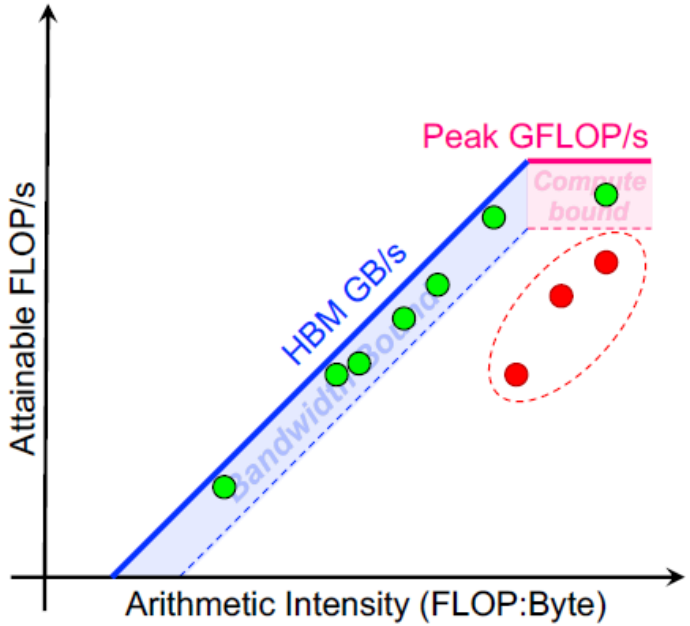
What is “Good” Performance?

- Think back to our mix of loop nests (benchmarks)
- We can sort kernels by their arithmetic intensity…
- … and compare performance relative to machine capabilities



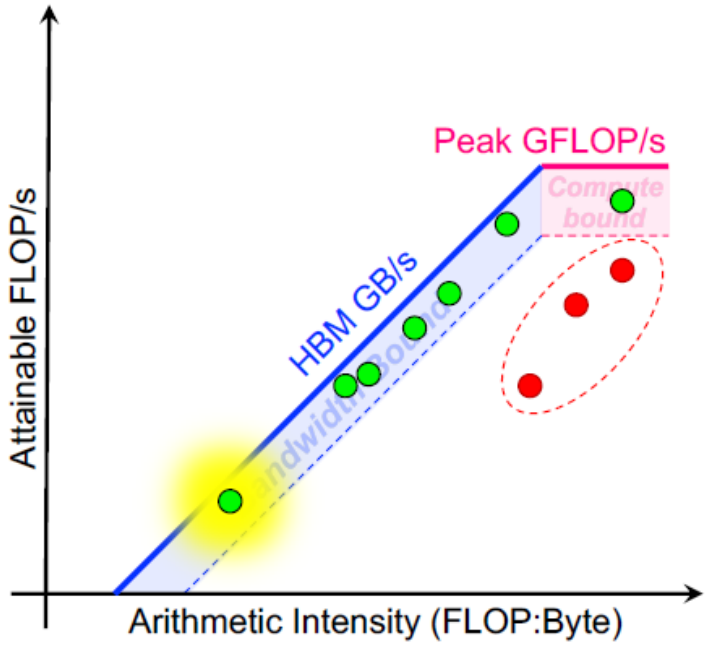
What is “Good” Performance?

- Kernels near the roofline are making **good use** of computational resources



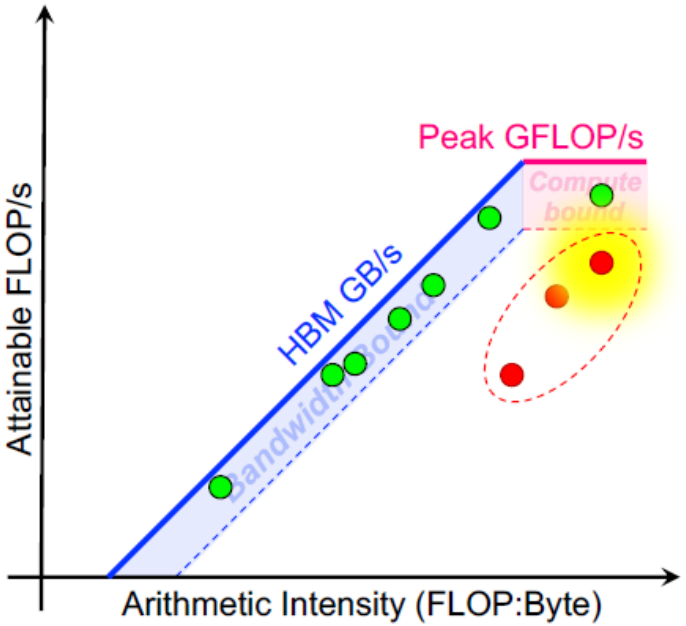
What is “Good” Performance?

- Kernels near the roofline are making **good use** of computational resources
  - kernels can have **low performance** (GFLOP/s), but make **good use** (%STREAM) of a machine



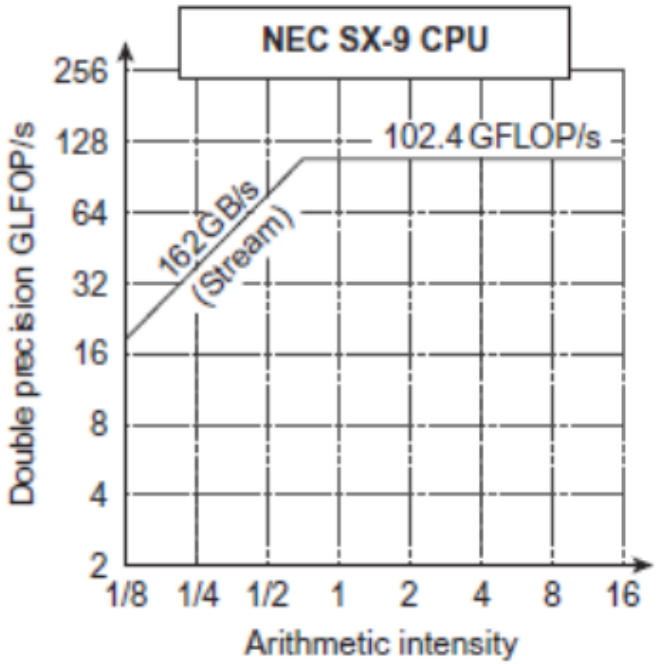
What is “Good” Performance?

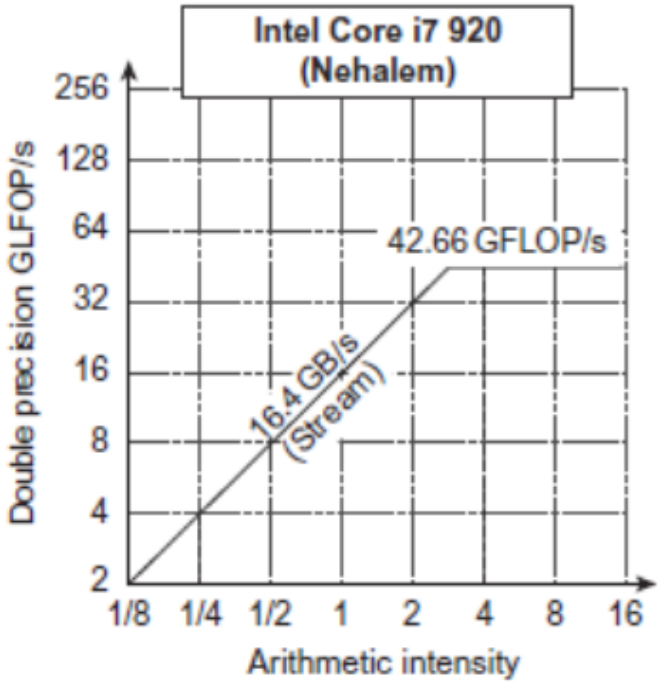
- Kernels near the roofline are making **good use** of computational resources
  - kernels can have **low performance** (GFLOP/s), but make **good use** (%STREAM) of a machine
  - kernels can have **high performance** (GFLOP/s), but still make **poor use** of a machine (%peak)



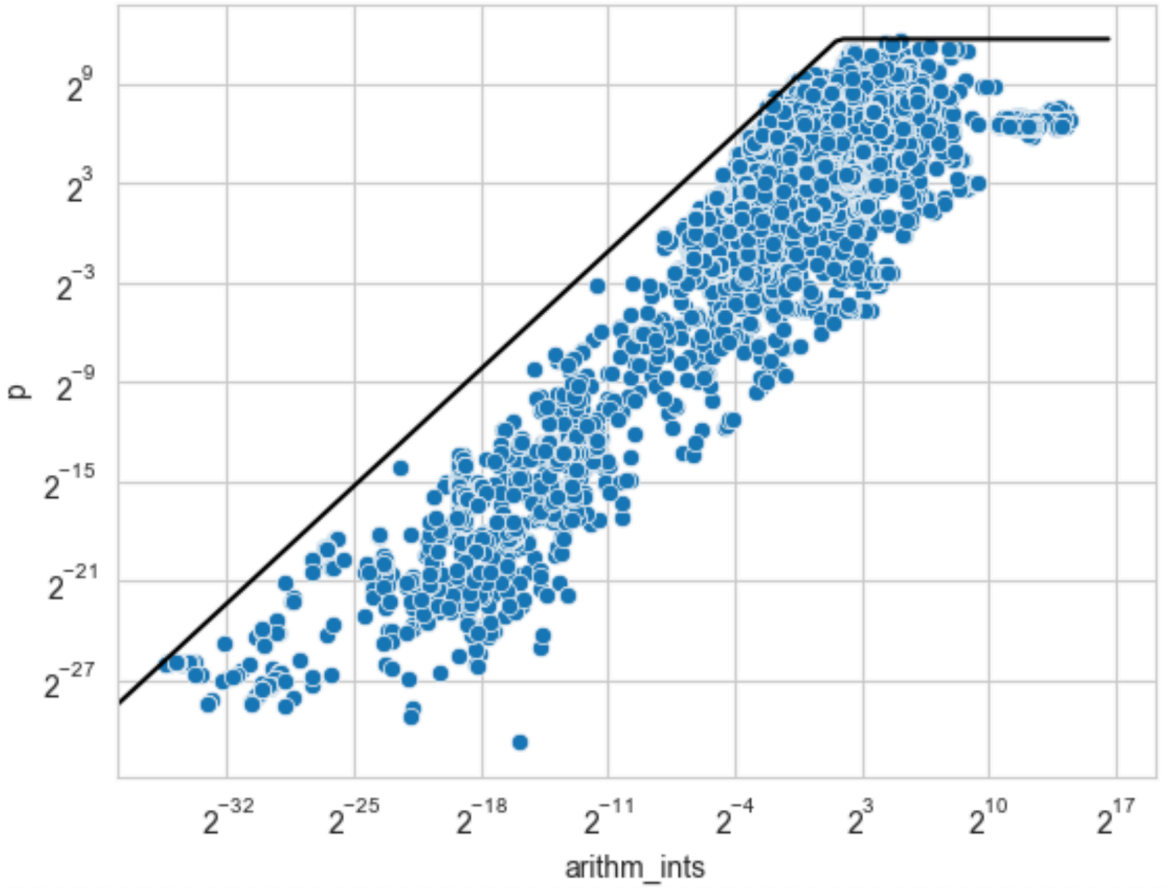
Examples

- Attainable GFLOPs/sec = (Peak Memory BW × Arithmetic Intensity, Peak Floating Point Perf.)





Intel core specs


p
2^{9}
2^{3}
2^{-3}
2^{-9}
2^{-15}
2^{-21}
2^{-27}
2^{-32}
2^{-25}
2^{-18}
2^{-11}
2^{-4}
2^{3}
2^{10}
2^{17}
arithm_ints