High performance computing (HPC) clusters

4. Top10 HPC systems*

*Note: The information presented in these slides is outdated.
Please check source material in the links for the latest information.

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	8,699,904	1,194.00	1,679.82	22,703
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	1,824,768	238.70	304.47	7,404
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	2,414,592	148.60	200.79	10,096

https://www.top500.org

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	4,981,760	61.44	100.68	18,482

https://www.top500.org

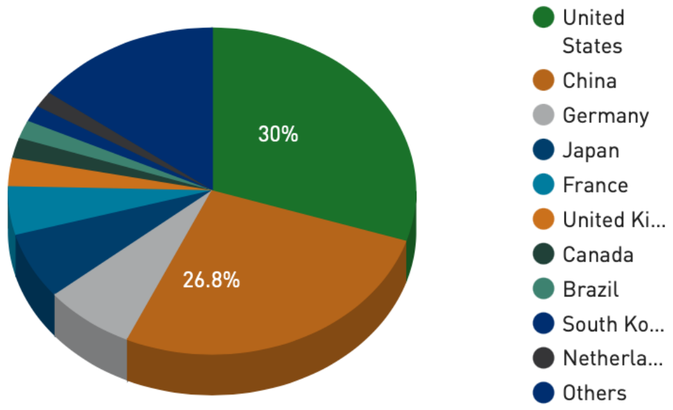
HPC in India

75	**AIRAWAT - PSAI** - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Infiniband HDR, Netweb Technologies Center for Development of Advanced Computing (C-DAC) India	81,344	8.50	13.17
131	**PARAM Siddhi-AI** - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, Atos Center for Development of Advanced Computing (C-DAC) India	41,664	4.62	5.27
169	**Pratyush** - Cray XC40, Xeon E5-2695v4 18C 2.1GHz, Aries interconnect , HPE India Institute of Tropical Meteorology India	119,232	3.76	4.01 1,353
316	**Mihir** - Cray XC40, Xeon E5-2695v4 18C 2.1GHz, Aries interconnect , HPE National Centre for Medium Range Weather Forecasting India	83,592	2.57	2.81 955

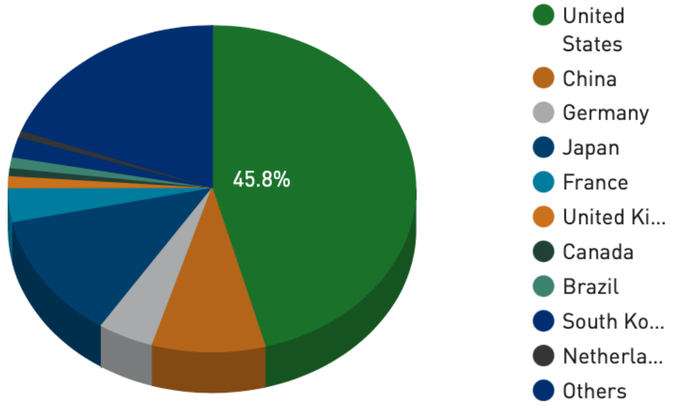
https://www.top500.org

Country-wise listing

Countries System Share

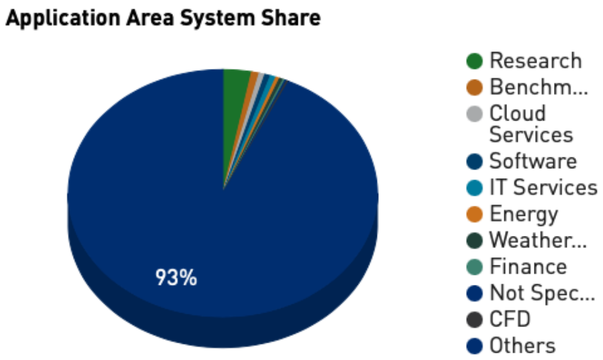


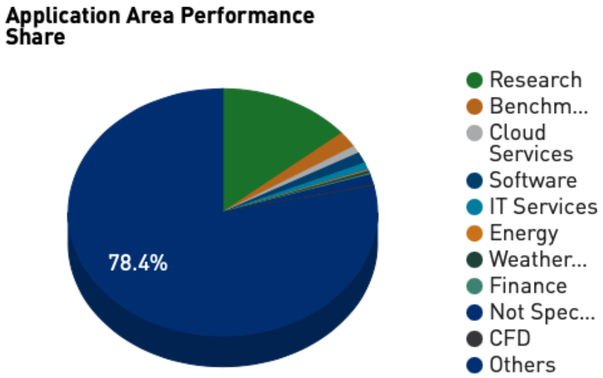
Countries Performance Share



	Countries	Count	System Share (%)	Rm
1	United States	150	30	2,
2	China	134	26.8	
3	Germany	36	7.2	
4	Japan	33	6.6	
5	France	24	4.8	
6	United Kingdom	14	2.8	
7	Canada	10	2	
8	Brazil	9	1.8	
9	South Korea	8	1.6	
10	Netherlands	8	1.6	

Application-wise listing





	Application Area	Count	System Share (%)
1	Research	15	3
2	Benchmarking	4	0.8
3	Cloud Services	3	0.6
4	Software	3	0.6
5	IT Services	3	0.6
6	Energy	2	0.4
7	Weather and Climate Research	2	0.4
8	Finance	1	0.2
9	Not Specified	1	0.2
10	CFD	1	0.2

Accelerator-wise listing

	Accelerator/Co-Processor	Count	Sy
1	NVIDIA Tesla V100	61	
2	NVIDIA A100	27	
3	NVIDIA A100 SXM4 40 GB	18	
4	NVIDIA Tesla A100 80G	10	
5	NVIDIA Tesla V100 SXM2	10	
6	AMD Instinct MI250X	10	
7	NVIDIA Tesla A100 40G	9	
8	NVIDIA A100 SXM4 80 GB	7	
9	NVIDIA H100	5	
10	NVIDIA Tesla P100	5	
11	NVIDIA Volta GV100	4	
12	NVIDIA Tesla K40	2	

	Interconnect	Count	System Share (%)
1	100G Ethernet	86	17.2
2	25G Ethernet	64	12.8
3	10G Ethernet	44	8.8
4	Infiniband HDR	37	7.4
5	Intel Omni-Path	33	6.6
6	Infiniband EDR	32	6.4
7	Mellanox HDR Infiniband	28	5.6
8	Aries interconnect	24	4.8
9	InfiniBand HDR100	20	4
10	Slingshot-10	17	3.4

	Operating System	Count	System Share (%
1	Linux	235	4
2	CentOS	62	12.
3	Cray Linux Environment	24	4.
4	HPE Cray OS	22	4.
5	Red Hat Enterprise Linux	12	2.
6	bullx SCS	8	1.
7	Ubuntu 20.04.1 LTS	8	1.
8	SLES15 SP2	6	1.
9	Ubuntu	6	1.
10	CentOS Linux 7	6	1.
11	RHEL 7.7	6	1.

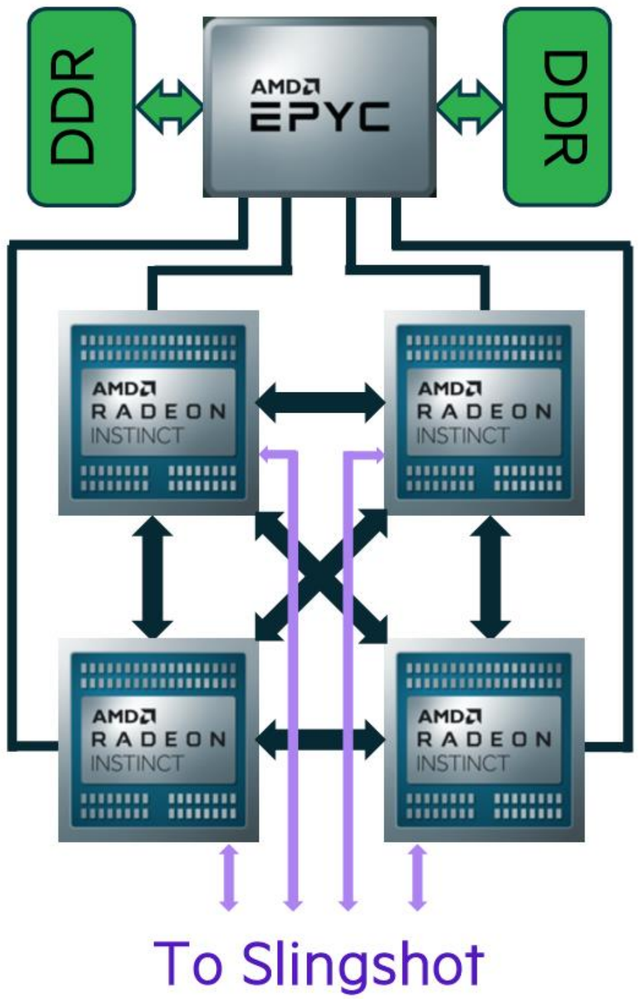
	Processor Generation	Count	
1	Xeon Gold (Skylake)	99	
2	Xeon Gold 62xx (Cascade Lake)	91	
3	AMD Zen-2 (Rome)	63	
4	AMD Zen-3 (Milan)	58	
5	Xeon Platinum 82xx (Cascade Lake)	44	
6	Xeon Platinum 83xx (Ice Lake)	35	
7	Intel Xeon E5 (Broadwell)	23	
8	Xeon Platinum (Skylake)	23	
9	Intel Xeon E5 (Haswell)	12	
10	IBM POWER9	7	

	Cores per Socket	Count	S
1	24	108	
2	64	94	
3	20	60	
4	16	36	
5	32	36	
6	18	35	
7	28	29	
8	12	21	
9	48	19	
10	14	10	
11	8	9	
12	36	8	
13	68	6	
14	38	5	
15	10	5	

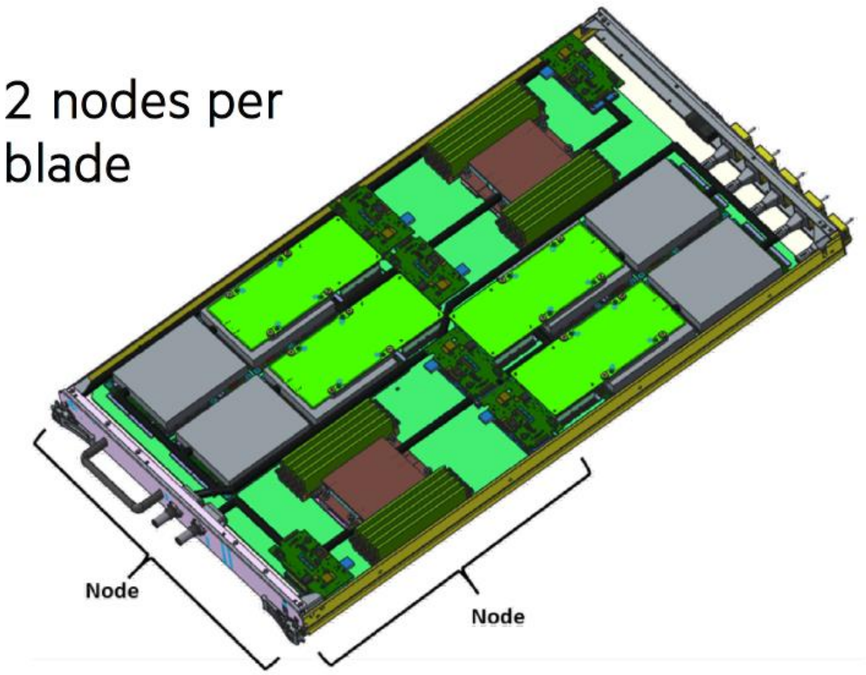
Frontier specs

Wikipedia

AMD GPU (ORNL)



2 nodes per blade



https://www.olcf.ornl.gov/frontier/#4

SYSTEM SPECS	TITAN	SUMMIT	FRONTIER
Peak Performance	27 PF	200 PF	1.6 EF
Cabinets	200	256	74
Node	1 AMD Opteron CPU 1 NVIDIA K20X Kepler GPU	2 IBM POWER9™ CPUs 6 NVIDIA Volta GPUs	1 HPC and AI Optimized 3rd Gen AMD EPYC CPU 4 Purpose Built AMD Instinct 250X GPUs
CPU-GPU Interconnect	PCI Gen2	NVLINK Coherent memory across the node	AMD Infinity Fabric

System Interconnect	Gemini	2x Mellanox EDR 100G InfiniBand Non-Blocking Fat-Tree	Multiple Slingshot NICs providing 100 GB/s network bandwidth. Slingshot network which provides adaptive routing, congestion management and quality of service.
Storage	32 PB, 1 TB/s, Lustre Filesystem	250 PB, 2.5 TB/s, GPFS™	2-4x performance and capacity of Summit's I/O subsystem. Frontier will have near node storage like Summit.

IITK's Param Sanganak

Processor model: 2*Intel Xeon R Platinum 8268
24 cores, 14nm, 2.2 GHz, 38.5 MB Cache

GPU: V100 Nvidia

	CPU	CPU (High Mem)	CPU (with GPU support)	GPU ready
Cores/Node	48	48	48	40
Frequency (GHz)	2.9	2.9	2.9	2.5
RAM (GB)	192	768	192	192
# nodes	150	78	64	20
RPeak	668.1	447.4	285.1	376
Total RAM (TB)	28.8	59.9	12.3	3.84

Graphical Processing Units (GPU)

https://medium.com/@smallfishbigsea/basic-concepts-in-gpu-computing-3388710e9239

https://jonathan-hui.medium.com/ai-chips-a100-gpu-with-nvidia-ampere-architecture-3034ed685e6e

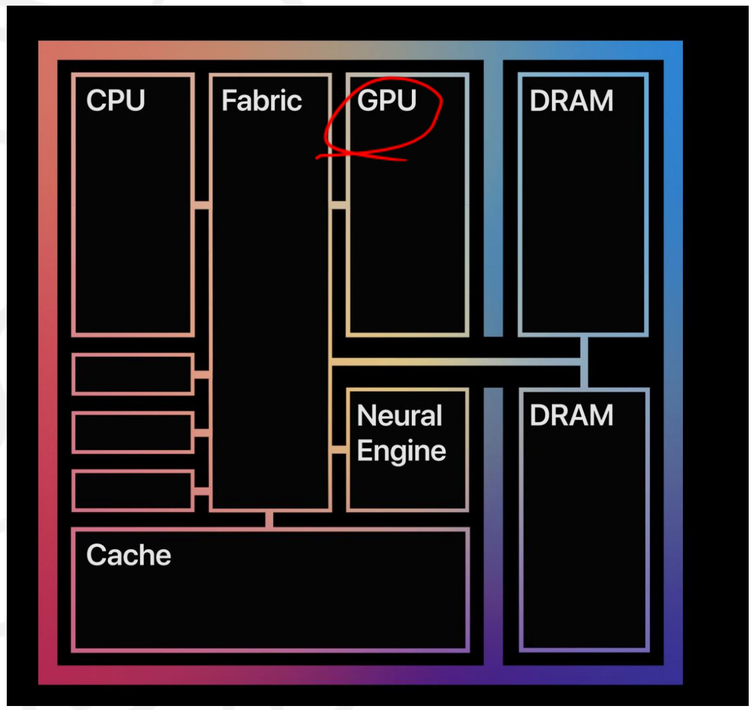
https://images.nvidia.com/aem-dam/en-zz/Solutions/data-center/nvidia-ampere-architecture-whitepaper.pdf

Graphics & Gaming

5000 cores

Nvidia quadro

https://www.nvidia.com/



https://www.apple.com

AMD's Radeon

https://www.amd.com

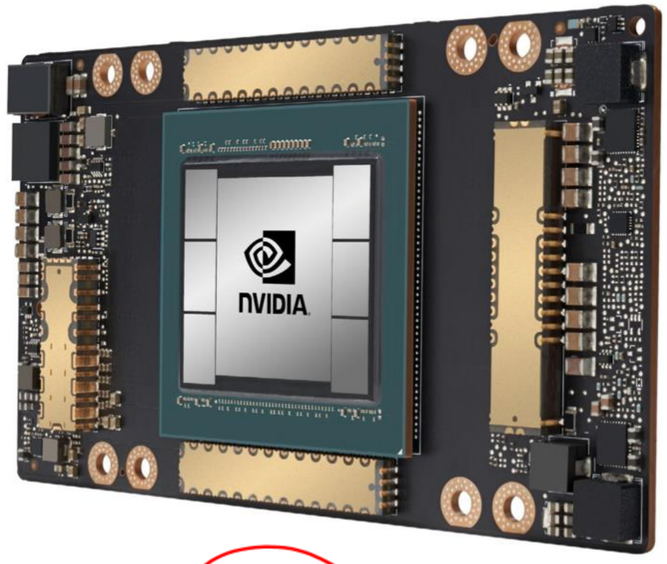
Using GPUs for HPC

V100, A100, H100

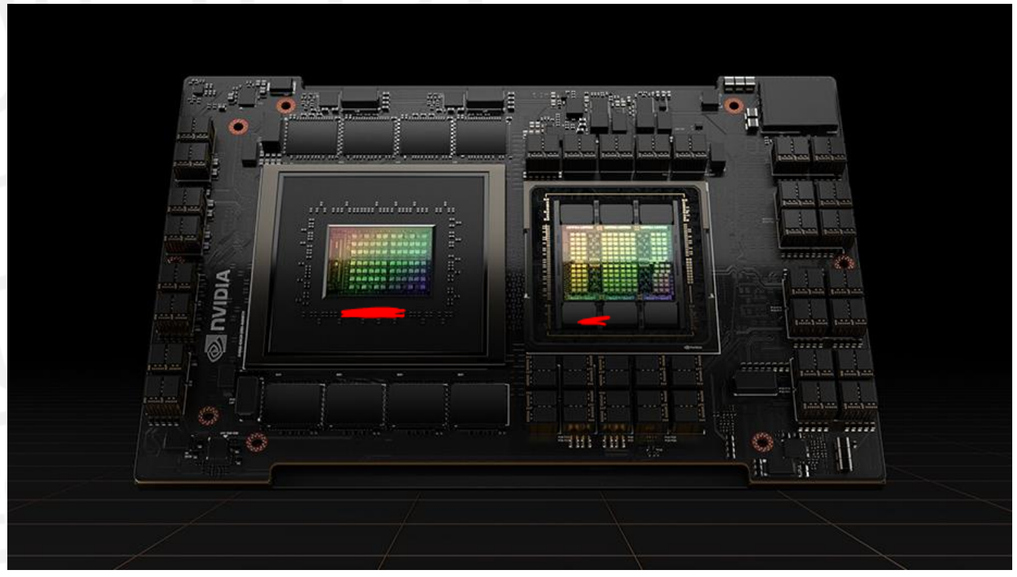
Volta, Ampere, Hopper architecture

Optimized for AI/ML

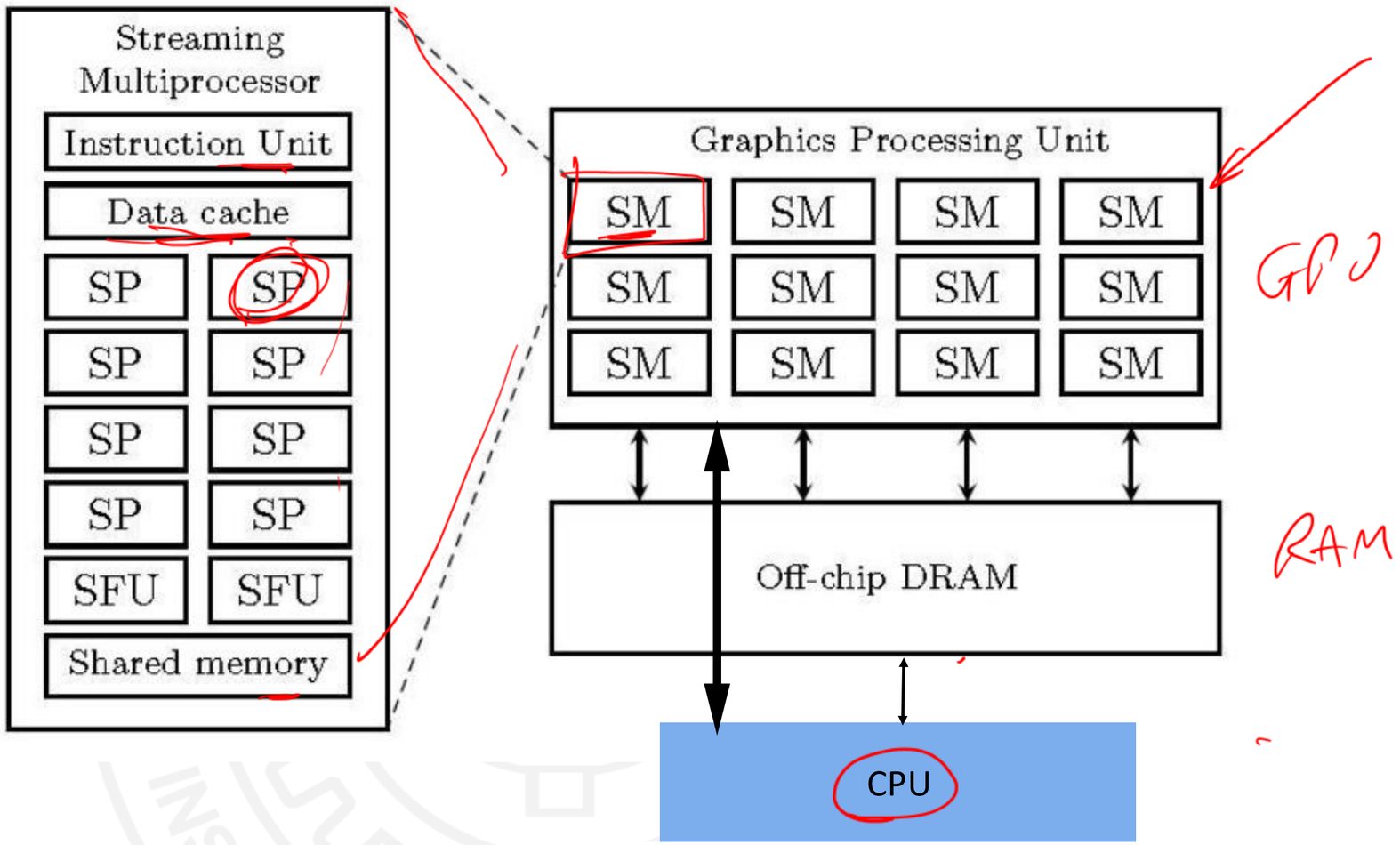
nvidia.com



A100



H100 + Grace CPU



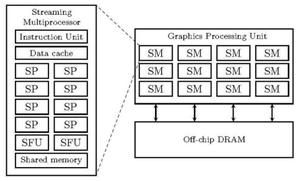
- Streaming Multiprocessor (SM)
- Scalar Processors (SP)

A Data-Parallel Algorithmic Modelica Extension for Efficient Execution on Multi-Core Platforms

Mahder Gebremedhin, Afshin Hemmati Moghadam, Peter Fritzson, Kristian Stavåker
Department of Computer and Information Science
Linköping University, SE-581 83 Linköping, Sweden
{mahder.gebremedin, peter.fritzson, Kristian.stavaker}@liu.se, afshe586@student.liu.se

Figure 6. GA100 Full GPU with 128 SMs (A100 Tensor Core GPU has 108 SMs)

https://images.nvidia.com/aem-dam/en-zz/Solutions/data-center/nvidia-ampere-architecture-whitepaper.pdf

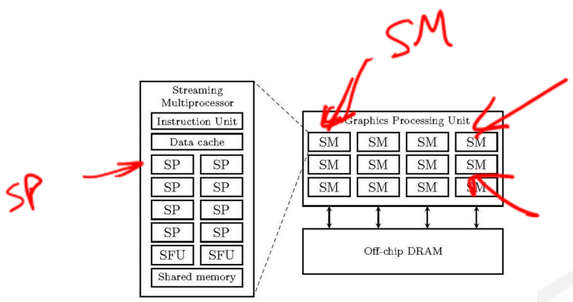


$C = A + B$

$A_i \; B_i$

$40^6 \quad 10^6$

- Vector processors: Single Instruction, Multiple Data (SIMD)
- GPUs: Single Instruction, Multiple Thread (SIMT)
- GPU runs a *Kernel*, which is massively multi-threaded.
- Kernel is divided into thread blocks, which is divided into number of threads.
- Thread blocks are automatically distributed among SPs.
- All threads execute one common instruction at a time.
- Each SP has a set of registers, which are shared among threads allocated to a SM.



A Data-Parallel Algorithmic Modelica Extension for Efficient Execution on Multi-Core Platforms

Mahder Gebremedhin, Afshin Hemmati Moghadam, Peter Fritzson, Kristian Stavåker
Department of Computer and Information Science
Linköping University, SE-581 83 Linköping, Sweden
{mahder.gebremedin, peter.fritzson, Kristian.stavaker}@liu.se, afshe586@student.liu.se

- Each SM can be treated as a separate processor.
- “Streaming Multiprocessors (SM) can work with different code, performing different operations with entirely different data (MIMD).”
- “All Scalar processors (SP) in one streaming multiprocessor execute the same instruction at the same time but work on different data (SIMT).”

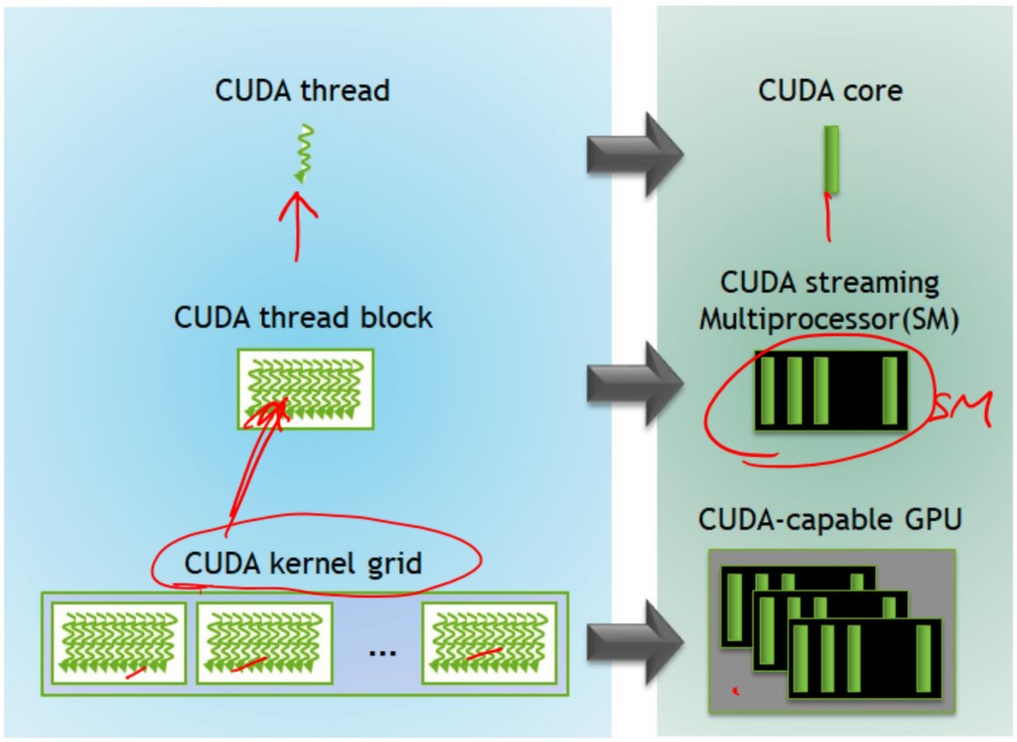


Figure 3. Kernel execution on GPU.

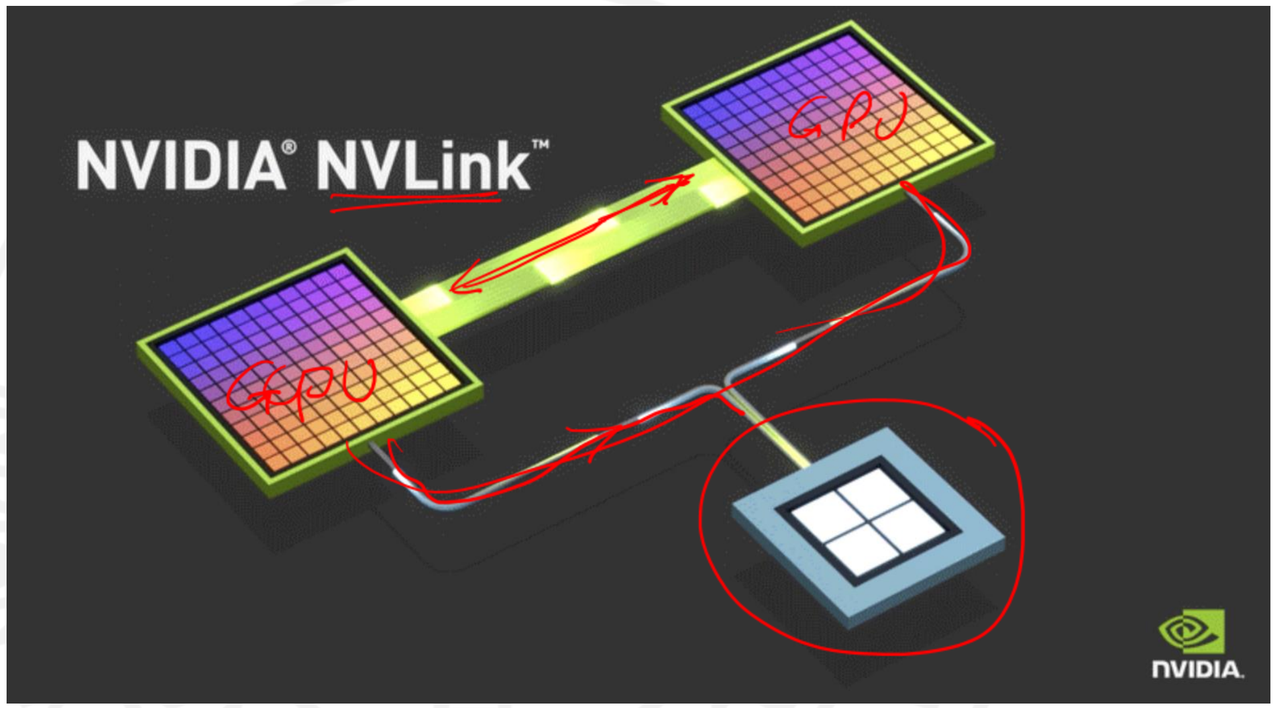
https://developer.nvidia.com/blog/cuda-refresher-cuda-programming-model/

Figure 7. GA100 Streaming Multiprocessor (SM)

https://images.nvidia.com/aem-dam/en-zz/Solutions/data-center/nvidia-ampere-architecture-whitepaper.pdf

Figure 7. GA100 Streaming Multiprocessor (SM)

- For each SM, Ampere has
- 4 processing block/SM, 1 Warp scheduler/processing block.
- 64 FP32 CUDA Cores/SM and 8192 FP32 CUDA Cores per full GPU.
- 64 INT32 CUDA Cores/SM, 32 FP64 CUDA Cores/SM.
- 192 KB of combined shared memory and L1 data cache
- 1 Tensor Cores/SM and 512 Tensor Cores per full GPU
- Common L2 cache: 40 MB
- Slower HBM (host has access to it): 40 or 80 GB



600 GB/sec between GPU pairs

PCIe 4: 31.5 GB/sec

Card	A100	V100
Transistors	54 billions	21.1 billions
CUDA cores	6912	5120
DP performance	9.7 TFLOPs	7.8 TFLOPs
SP performance	19.5 TFLOPs	15.7 TFLOPs
Tensor Performance	125 TF	125 TF
Node	7nm	12nm TSMC
Mem	40 GB HBM2e / 80 GB	16/32GB HBM2
Mem bus	6144-bit	4096-bit
Mem bandwidth	1.6 TB/sec	900 GB/sec
Tensor cores	1024 3rd gen	640 1st gen
Interface	PCIe 4.0x16	PCIe 3.0x16
TDP	400 W	250-300 W

H100

Graphics Processor

GPU Name:	GH100
Architectur	Hopper
Foundry:	TSMC
Process Size:	4 nm
Transistors:	80,000 million
Density:	98.3M / mm²
Die Size:	814 mm²

Graphics Features

DirectX:	N/A
OpenGL:	N/A
OpenCL:	3.0
Vulkan:	N/A
CUDA:	9.0

Graphics Card

Release Date:	Mar 22nd, 2022
Generation:	Tesla Hopper (Hxx)
Predecesso	Tesla Ada
Production:	Active
Bus Interface:	PCIe 5.0 x16

Board Design

Slot Width:	Dual-slot
Length:	268 mm 10.6 inches
Width:	111 mm 4.4 inches
TDP:	350 W
Suggested	750 W

Clock Speeds

Base Clock:	1095 MHz
Boost Clock:	1755 MHz
Memory Clock:	1593 MHz 3.2 Gbps effective

Render Config

Shading Units:	14592
TMUs:	456
ROPs:	24
SM Count:	114
Tensor Cores:	456
L1 Cache:	256 KB (per SM)
L2 Cache:	50 MB

Memory

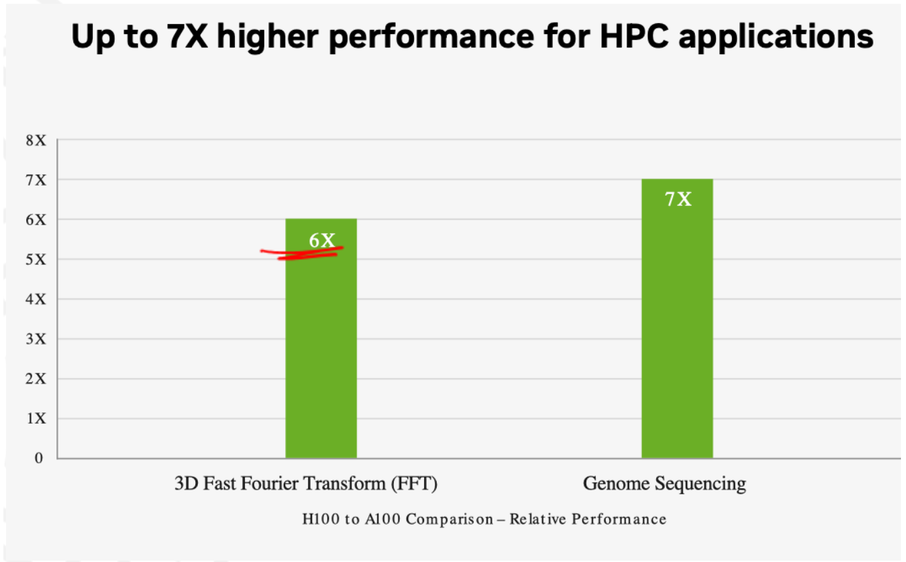
Memory Size:	80 GB
Memory Type:	HBM2e
Memory Bus:	5120 bit
Bandwidth:	2,039 GB/s

Theoretical Performance

Pixel Rate:	42.12 GPixel/s
Texture Rate:	800.3 GTexel/s
FP16 (half):	204.9 TFLOPS (4:1)
FP32 (float):	51.22 TFLOPS
FP64 (double):	25.61 TFLOPS (1:2)

https://www.techpowerup.com/gpu-specs/h100-pcie-80-gb.c3899

Form Factor	H100 SXM	H100 PCIe	H100 NVL
FP64	34 teraFLOPS	26 teraFLOPS	68 teraFLOPs
FP64 Tensor Core	67 teraFLOPS	51 teraFLOPS	134 teraFLOPs
FP32	67 teraFLOPS	51 teraFLOPS	134 teraFLOPs
TF32 Tensor Core	989 teraFLOPS[2]	756 teraFLOPS[2]	1,979 teraFLOPs[2]
BFLOAT16 Tensor Core	1,979 teraFLOPS[2]	1,513 teraFLOPS[2]	3,958 teraFLOPs[2]
FP16 Tensor Core	1,979 teraFLOPS[2]	1,513 teraFLOPS[2]	3,958 teraFLOPs[2]
FP8 Tensor Core	3,958 teraFLOPS[2]	3,026 teraFLOPS[2]	7,916 teraFLOPs[2]
INT8 Tensor Core	3,958 TOPS[2]	3,026 TOPS[2]	7,916 TOPS[2]
GPU memory	80GB	80GB	188GB
GPU memory bandwidth	3.35TB/s	2TB/s	7.8TB/s[3]
Interconnect	NVLink: 900GB/s PCIe Gen5: 128GB/s	NVLink: 600GB/s PCIe Gen5: 128GB/s	NVLink: 600GB/s PCIe Gen5: 128GB/s



https://www.nvidia.com/en-in/data-center/h100/

Programming

- CUDA
- OpenCL
- OpenACC
- Nvidia compiler with pragmas
- Python (CuPy)

AMD MI100

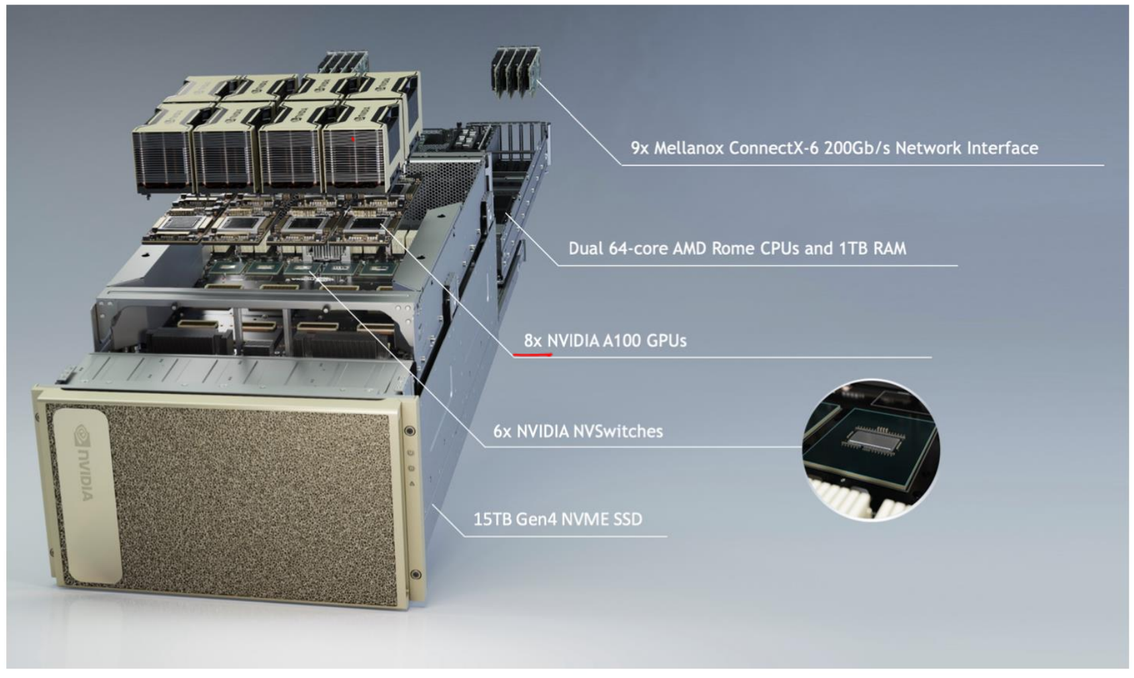
Card	A100	AMD MI100
Transistors	54 billions	21.1 billions
cores	6912	7680
Peak FB64	9.7 TFLOPs	11.5 TFLOPs
Peak FB32	19.5 TFLOPs	43.1 TFLOPs
Peak FB16	78 TF	184.6
Peak BF16	39 TF	92.3 TF
Peak INT8	624 TOPS	184.6 TOPS
Peak INT4	1248 TOPS	184.6 TOPS
Mem	40 GB HBM2e	32GB HBM2
Mem bandwidth	1.6 TB/sec	1.2 TB/sec
TDP	400 W	300 W

Accelerator	Architecture	Lithography	Compute Units	Memory	Memory Type	PCIe Support	Form Factor	FP16 Performance	BF16 Performance	FP32 Performance	FP32 Matrix Performance	FP64 Performance	FP64 Matrix Performance	INT8 Performance	
MI6	GCN 4	14 nm	36	16 GB	GDDR5	3.0	PCIe	5.7 TFLOPS	N/A	5.7 TFLOPS	N/A	358 GFLOPS	N/A	N/A	
MI8	GCN 3	28 nm	64	4 GB	HBM			8.2 TFLOPS		8.2 TFLOPS		512 GFLOPS			
MI25	GCN 5	14 nm	64	16 GB	HBM2			26.4 TFLOPS		12.3 TFLOPS		768 GFLOPS			
MI50	GCN 5	7 nm	60			4.0		26.5 TFLOPS		13.3 TFLOPS		6.6 TFLOPS		53 TOPS	
MI60	GCN 5		64	32 GB				29.5 TFLOPS		14.7 TFLOPS		7.4 TFLOPS		59 TOPS	
MI100	CDNA		120					184.6 TFLOPS	92.3 TFLOPS	23.1 TFLOPS	46.1 TFLOPS	11.5 TFLOPS		184.6 TOPS	
MI210	CDNA 2	6 nm	104	64 GB	HBM2e			181 TFLOPS		22.6 TFLOPS	45.3 TFLOPS	22.6 TFLOPS	45.3 TFLOPS	181 TOPS	
MI250	CDNA 2		208	128 GB			OAM	362.1 TFLOPS		45.3 TFLOPS	90.5 TFLOPS	45.3 TFLOPS	90.5 TFLOPS	362.1 TOPS	
MI250X	CDNA 2		220					383 TFLOPS		47.92 TFLOPS	95.7 TFLOPS	47.9 TFLOPS	95.7 TFLOPS	383 TOPS	

Wikipedia

GPU Boxes

DGX A100

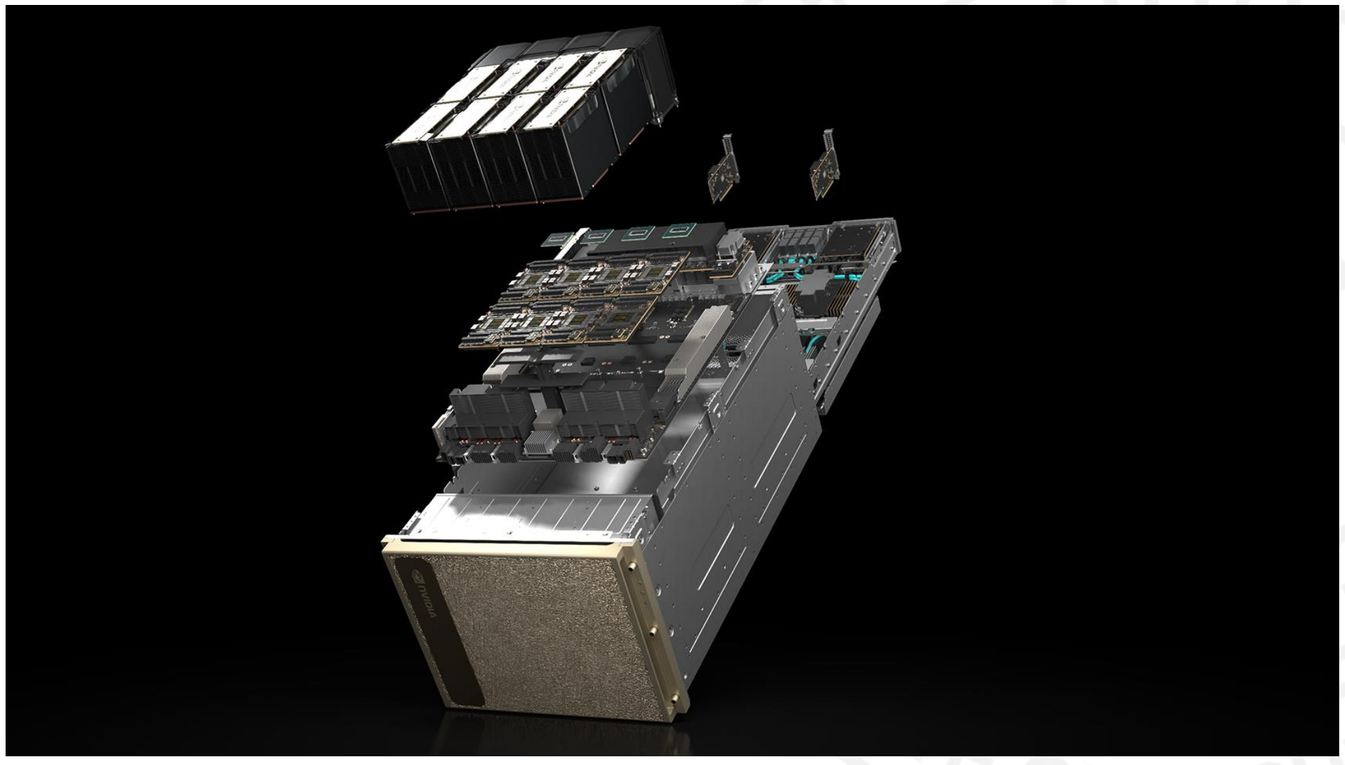


nvidia.com

SYSTEM SPECIFICATIONS

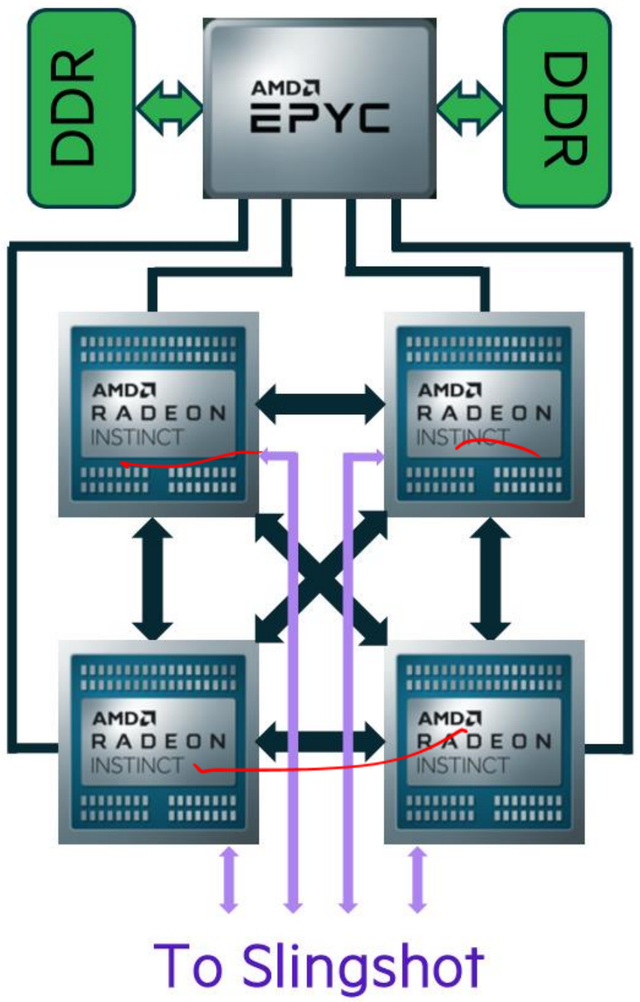
GPUs	**8x NVIDIA A100 Tensor Core GPUs**
GPU Memory	**320 GB total**
Performance	**5 petaFLOPS AI** **10 petaOPS INT8**
NVIDIA NVSwitches	**6**
System Power Usage	**6.5kW max**
CPU	**Dual AMD Rome 7742, 128 cores total, 2.25 GHz (base), 3.4 GHz (max boost)**
System Memory	**1TB**
Networking	**8x Single-Port Mellanox ConnectX-6 VPI** **200Gb/s HDR InfiniBand** **1x Dual-Port Mellanox ConnectX-6 VPI** **10/25/50/100/200Gb/s Ethernet**
Storage	**OS: 2x 1.92TB M.2 NVME drives** **Internal Storage: 15TB (4x 3.84TB) U.2 NVME drives**
Software	**Ubuntu Linux OS**
System Weight	**271 lbs (123 kgs)**
Packaged System Weight	**315 lbs (143kgs)**
System Dimensions	**Height: 10.4 in (264.0 mm)** **Width: 19.0 in (482.3 mm) MAX**

DGX H100

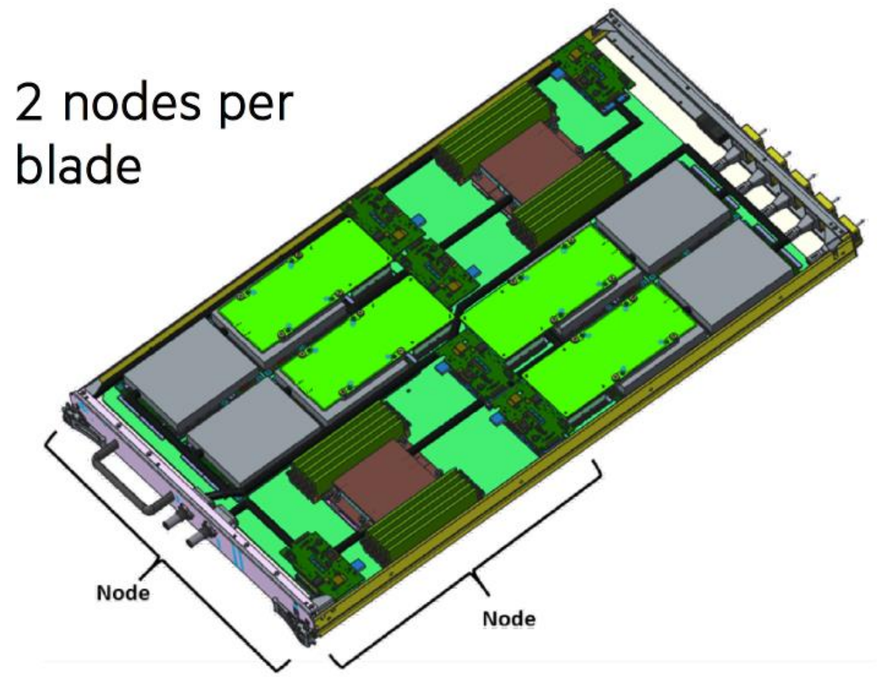


- **8x NVIDIA H100 GPUs With 640 Gigabytes of Total GPU Memory**
  18x NVIDIA NVLink connections per GPU, 900 gigabytes per second of bidirectional GPU-to-GPU bandwidth
- **4x NVIDIA NVSwitches**
  7.2 terabytes per second of bidirectional GPU-to-GPU bandwidth, 1.5X more than previous generation
- **10x NVIDIA ConnectX -7 400 Gigabits-Per-Second Network Interface**
  1 terabyte per second of peak bidirectional network bandwidth
- **Dual Intel Xeon Platinum 8480C processors, 112 cores total, and 2 TB System Memory**
  Powerful CPUs for the most intensive AI jobs
- **30 Terabytes NVMe SSD**
  High speed storage for maximum performance

AMD GPU (ORNL)

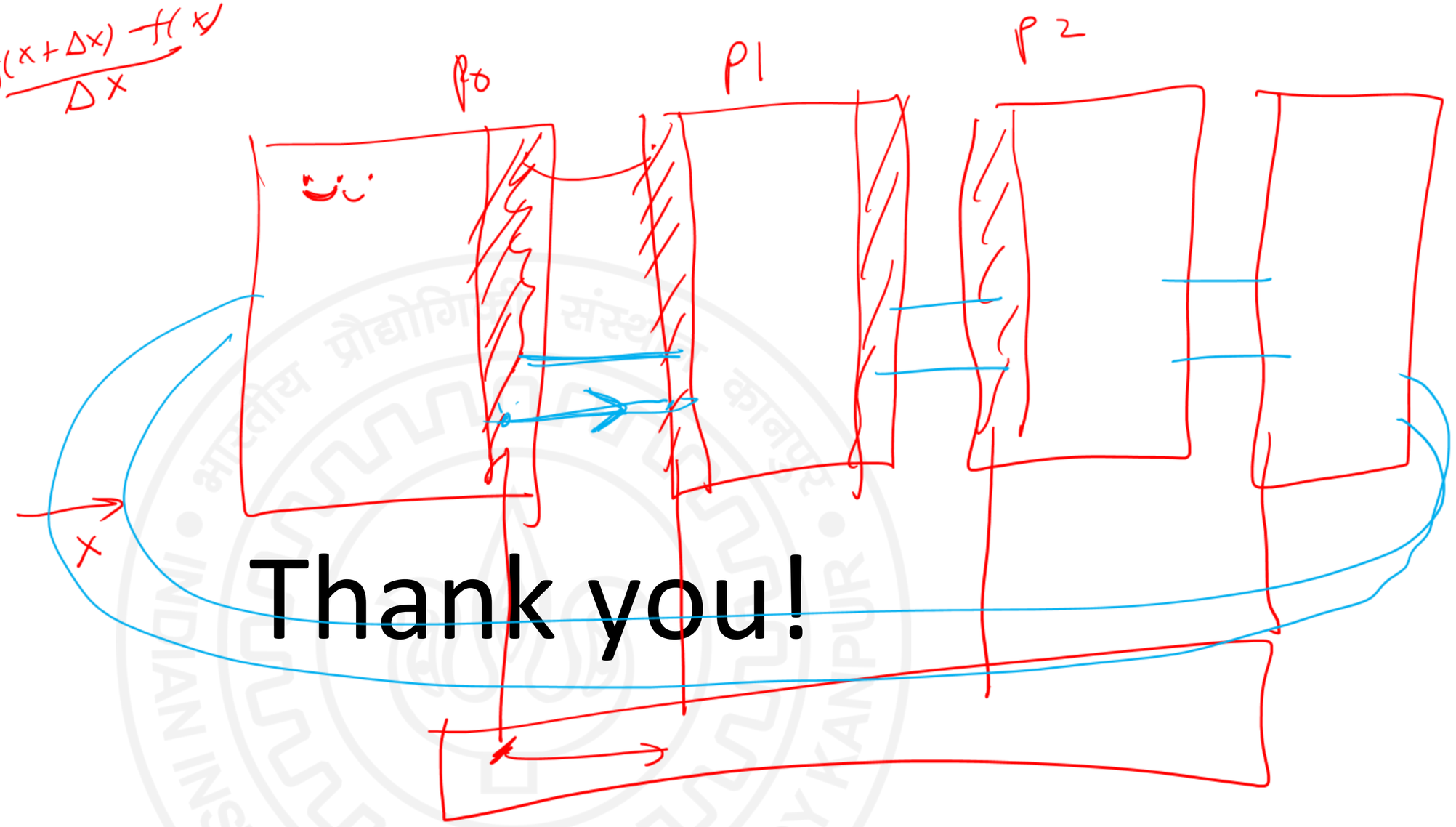


2 nodes per blade



https://www.olcf.ornl.gov/frontier/#4

$$\frac{df}{dx} = \frac{f(x+\Delta x) - f(x)}{\Delta x}$$



Thank you!

N

Earth simulator

2003

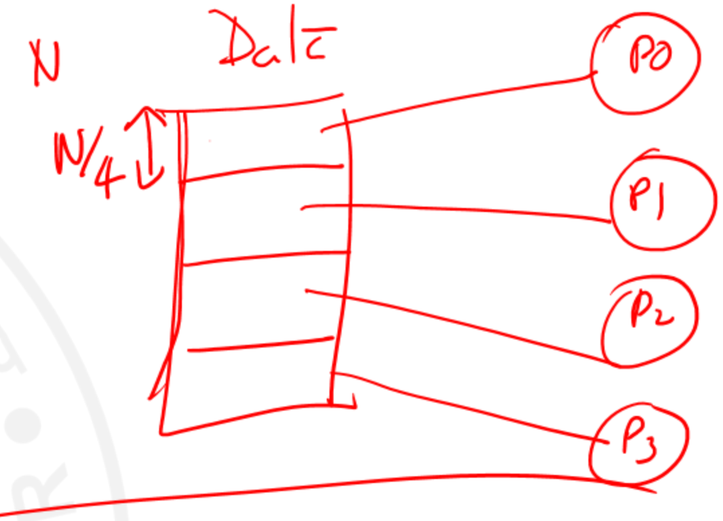
50%

2%

Programming paradigm

Single Program Multiple Data (SPMD)

- Single program runs on all the processors.
- Divide the work equally among processors: Load balance.
- May need to communicate among some processors.
- The result is collated by the *master processor.*



myid

Data = A[i*N/4 : (i+1)N/4 - 1]

for ← data

mysum = Sum the data

if myid == 0 :

get (local sum)

add localsum → SUM

Shared Memory: OPENMP, pthreads

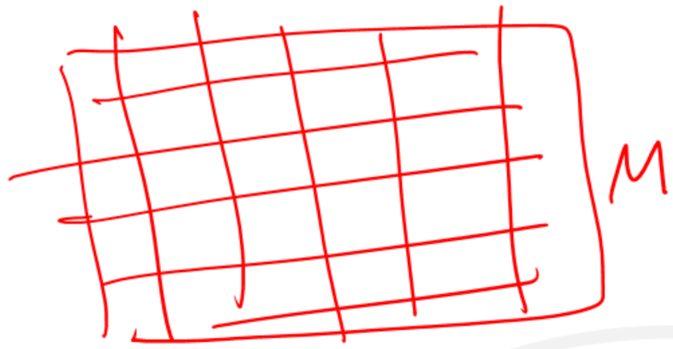
Distributed Memory:
Message passing interface (MPI)

Multiple Program Multiple Data (MPMD)

- For example, human body—brain some proc, heart some other proc, etc.
- Earth and atmospheric simulations.

- Not very popular, can convert MPMD to SPMD
- Coordination among different programs is difficult.

Speedup & Efficiency



- Speedup *S* = Tserial/Tparallel
  - Ideal: *p*
- Efficiency = *S*/*p*
- *Tparallel = Tserial/p + Toverhead*

Amdahl's law

Strong scaling

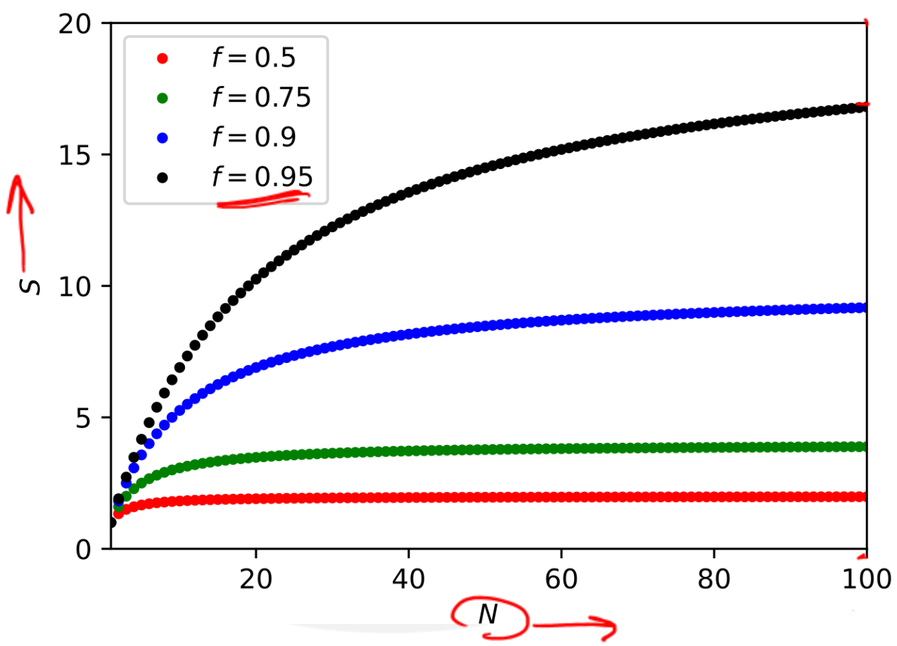
- Simulation with M data points (FIXED) with N processors.
- Total time for a program $T = t_s + t_p$ (+ t_{comm})
- After parallelisation, total time $T' = t_s + t_p/N$ (N=number of procs)
- Time fraction that is parallelized $f = t_p/T = t_p/(t_s + t_p)$
- Speedup of parallelized code $S' = N$ (assumption!)
- $T' = t_s + t_p - t_p + t_p/N$
- $T'/T = 1 - f + f/N$
- Overall speedup $S = T/T' = \frac{1}{1-f+f/N}$

- Overall speedup S = T/T' = $\frac{1}{1-f+f/N}$
- S < $\frac{1}{1-f}$ (bound)
- If a code can be parallelized up to 90%, the best efficiency we can get is S<1/0.1=10.
- If p=1, S = N

Strong scaling

$$\text{Speedup} = S = \frac{1}{1 - f + f/N}$$

N=procs



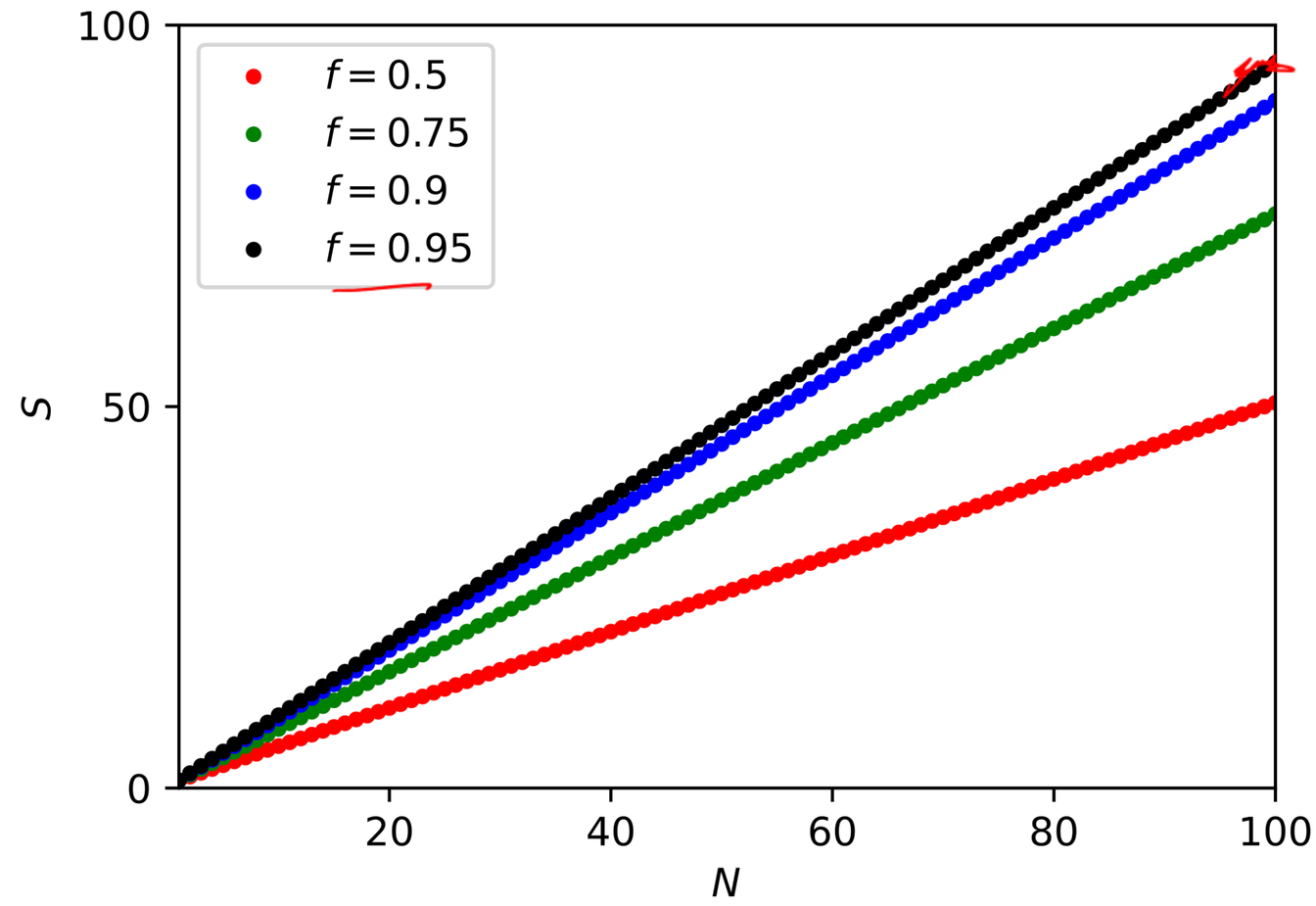
https://www.kth.se/blogs/pdc/2018/11/scalability-strong-and-weak-scaling/

Gustafson's law & weak scaling

- Simulation with M*N data points with N processors.
- Time for the sequential part = t_s; Time for the parallel part = t_p
- Total time = $t_s + t_p$
- Total # of ops with 1 processor = $(t_s + t_p)$FLOP
- Total # of ops with N processors = $(t_s + t_pN)$FLOP
  - $=(t_s + t_p - t_p + t_pN)$FLOP
- Weak scaling speedup = Ration of the two = S= $1 - f + Nf$

$$\frac{t_s + t_p - t_p + t_pN}{t_s + t_p} = 2$$

$$S = 1 - f + Nf$$



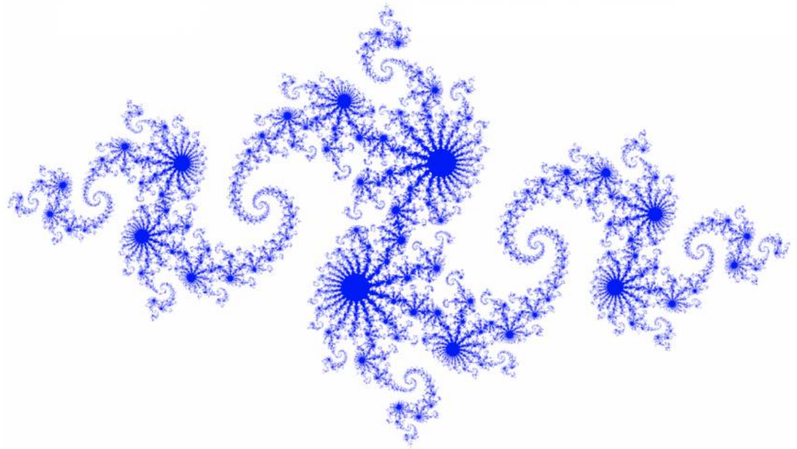
Example

Scalability: strong and weak scaling

Xin Li, 2018-11-09

https://www.kth.se/blogs/pdc/2018/11/scalability-strong-and-weak-scaling/

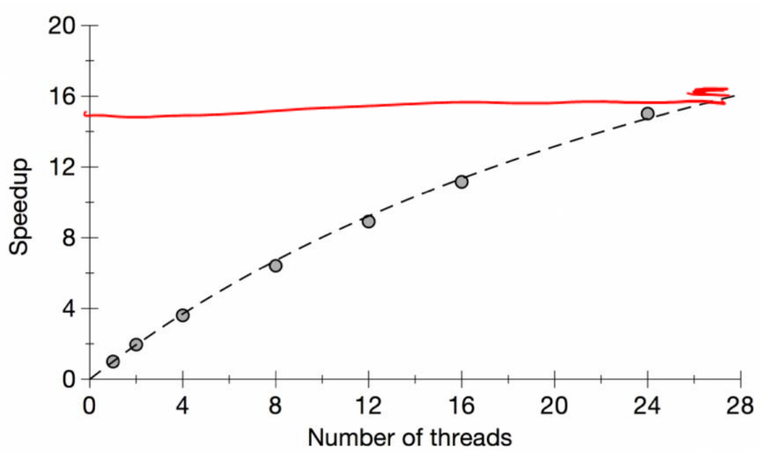
Julia set generator



Strong scaling

Table 1: Strong scaling for Julia set generator code

height	width	threads	time
10000	2000	1	3.932 sec
10000	2000	2	2.006 sec
10000	2000	4	1.088 sec
10000	2000	8	0.613 sec
10000	2000	12	0.441 sec
10000	2000	16	0.352 sec
10000	2000	24	0.262 sec

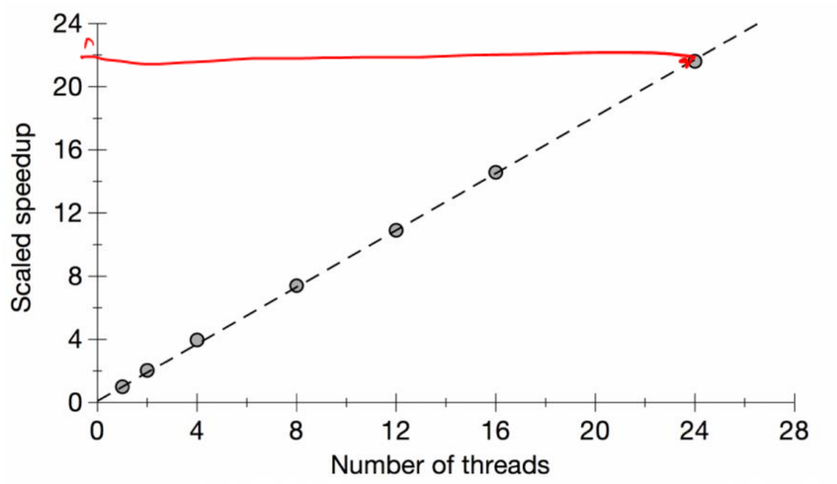


$$S = \frac{1}{1 - f + f/N}$$

f=0.97

Table 2: Weak scaling for Julia set generator code

height	width	threads	time
10000	2000	1	3.940 sec
20000	2000	2	3.874 sec
40000	2000	4	3.977 sec
80000	2000	8	4.258 sec
120000	2000	12	4.335 sec
160000	2000	16	4.324 sec
240000	2000	24	4.378 sec



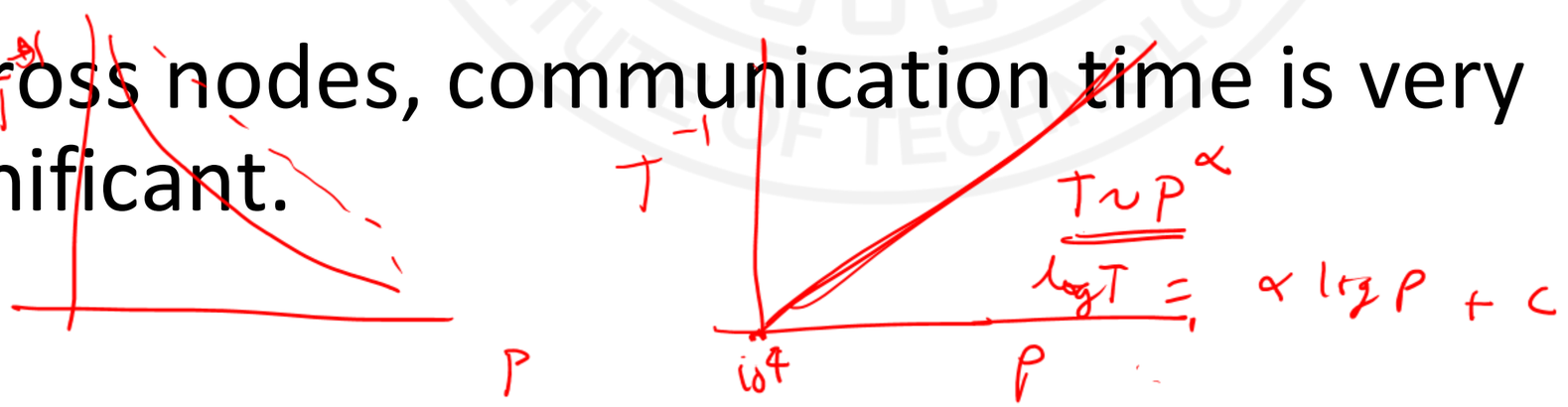
Fit with

$S = 1 - f + Nf$

$f=0.9$

Problems with the Above Formulas

- These formulas assume zero communication and mem-copy time.
- They do not work for large N and data
  - Such programs do not run on a single processor.
  - Data is spread over nodes.
  - Across nodes, communication time is very significant.



Scaling on CRAY XC40 (Shaheen II)

TARANG

Minimum number of cores to be used is large (2000 or more)
T_{serial} cannot be determined easily.
So we plot time taken vs. procs.

J. Parallel Distrib. Comput. 113 (2018) 77–91

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

Scaling of a Fast Fourier Transform and a pseudo-spectral fluid solver up to 196608 cores

Anando G. Chatterjee [a], Mahendra K. Verma [a], Abhishek Kumar [a,*], Ravi Samtaney [b], Bilel Hadri [c], Rooh Khurram [c]

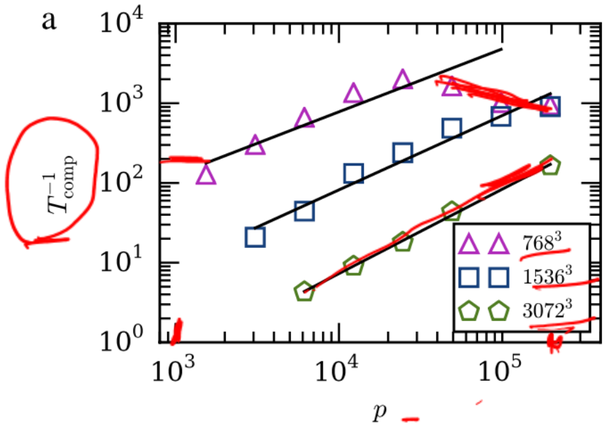
[a] Department of Physics, Indian Institute of Technology Kanpur, Kanpur 208016, India
[b] Mechanical Engineering, Division of Physical Science and Engineering, King Abdullah University of Science and Technology, Thuwal, 23955-6900, Saudi Arabia
[c] KAUST Supercomputing Laboratory, King Abdullah University of Science and Technology, Thuwal, 23955-6900, Saudi Arabia

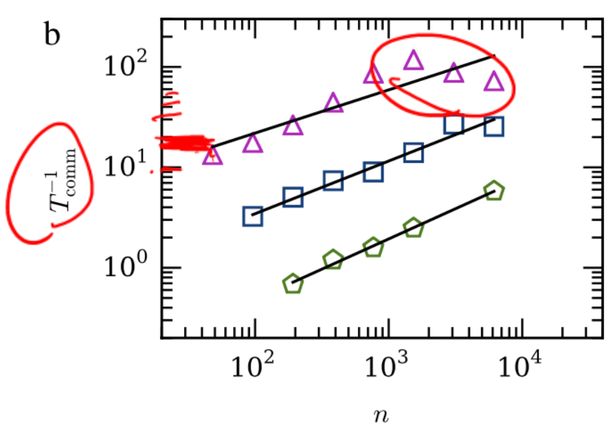
FFTK: Strong scaling

$$T_{\text{comp}} = \frac{1}{c_1} p^{-\gamma_1},$$

$$T_{\text{comm}} = \frac{1}{c_1} n^{-\gamma_2},$$

$$T^{-1} = \frac{1}{C} p^{\gamma},$$





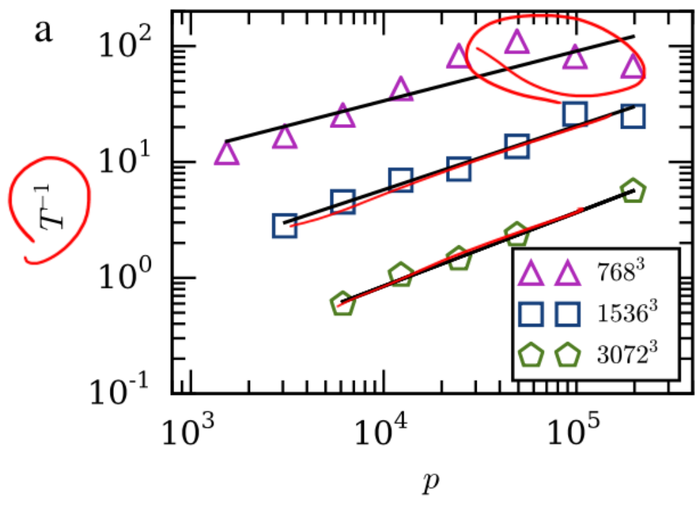


Fig. 5. Scalings of FFTK on Cray XC40 for the FFF basis: (a) Plots of T_{comp}^{-1} vs. p (number of cores) for 768^3, 1536^3, and 3072^3 grids. (b) Plots of T_{comm}^{-1} vs. n (number of nodes) using the above convention.

$$T_{\text{comp}} = \frac{1}{c_1} p^{-\gamma_1}, \tag{17a}$$

$$T_{\text{comm}} = \frac{1}{c_1} n^{-\gamma_2}, \tag{17b}$$

$$T^{-1} = \frac{1}{C} p^{\gamma}, \tag{17c}$$

Table 4
FFTK scaling on Cray XC40 for the FFF and SFF basis: The exponents γ_1 for the computation time (T_{comp}), γ_2 for the communication time (T_{comm}), and γ for the total time (T) [refer to Eq. (17) for definition]. Maximum cores used: 196 608.

Grid	γ_1	γ_2	γ
FFF			
768^3	0.79 ± 0.14	0.43 ± 0.09	0.43 ± 0.09
1536^3	0.93 ± 0.08	0.52 ± 0.04	0.55 ± 0.04
3072^3	1.08 ± 0.03	0.60 ± 0.02	0.64 ± 0.02
SFF			
768^3	0.82 ± 0.13	0.44 ± 0.03	0.46 ± 0.04
1536^3	0.97 ± 0.07	0.63 ± 0.02	0.66 ± 0.01
3072^3	0.99 ± 0.04	0.70 ± 0.05	0.73 ± 0.05

Table 5
FFTK on Cray XC40: Effective FLOP rating in Giga FLOP/s of Cray XC40 cores for various grid sizes and ppn. The efficiency E is the ratio of the effective per-core FLOP rating and the peak FLOP rating of each core (approximately 36 G FLOP/s).

Grid size	768^3	1536^3	3072^3
GFlop/s	0.45	0.53	0.64
E	0.013	0.015	0.018

$5 N \lg N$

$E = \frac{\#\ \text{FLOP offered}}{\#\ \text{MAX FLOP}}$

0.02

Chatterjee et al., JPDC 2018

FFT: Weak scaling

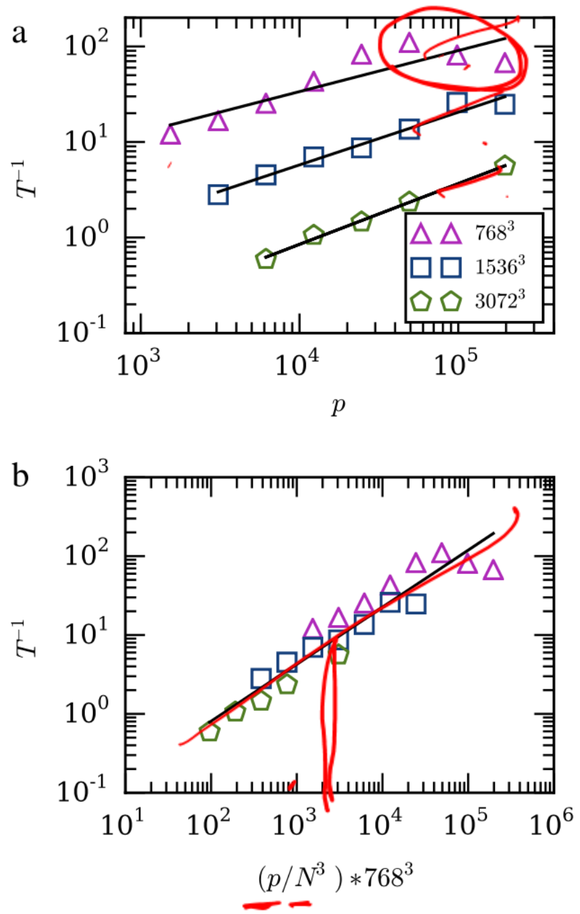


Fig. 6. Scalings of FFTK on Cray XC40 for the FFF basis: (a) plots of T^{-1} vs. p for 768^3, 1536^3, and 3072^3 grids. (b) plots of T^{-1} vs. p/N^3 exhibits weak scaling with an exponent of $\gamma = 0.72 \pm 0.03$.

Scaling on Selene

SN Computer Science (2023) 4:625
https://doi.org/10.1007/s42979-023-02109-0

ORIGINAL RESEARCH

Scalable Multi-node Fast Fourier Transform on GPUs

Manthan Verma[1] · Soumyadeep Chatterjee[1] · Gaurav Garg[4] · Bharatkumar Sharma[5] · Nishant Arya[3] · Sashi Kumar[3] · Anish Saxena[2] · Mahendra K. Verma[1]

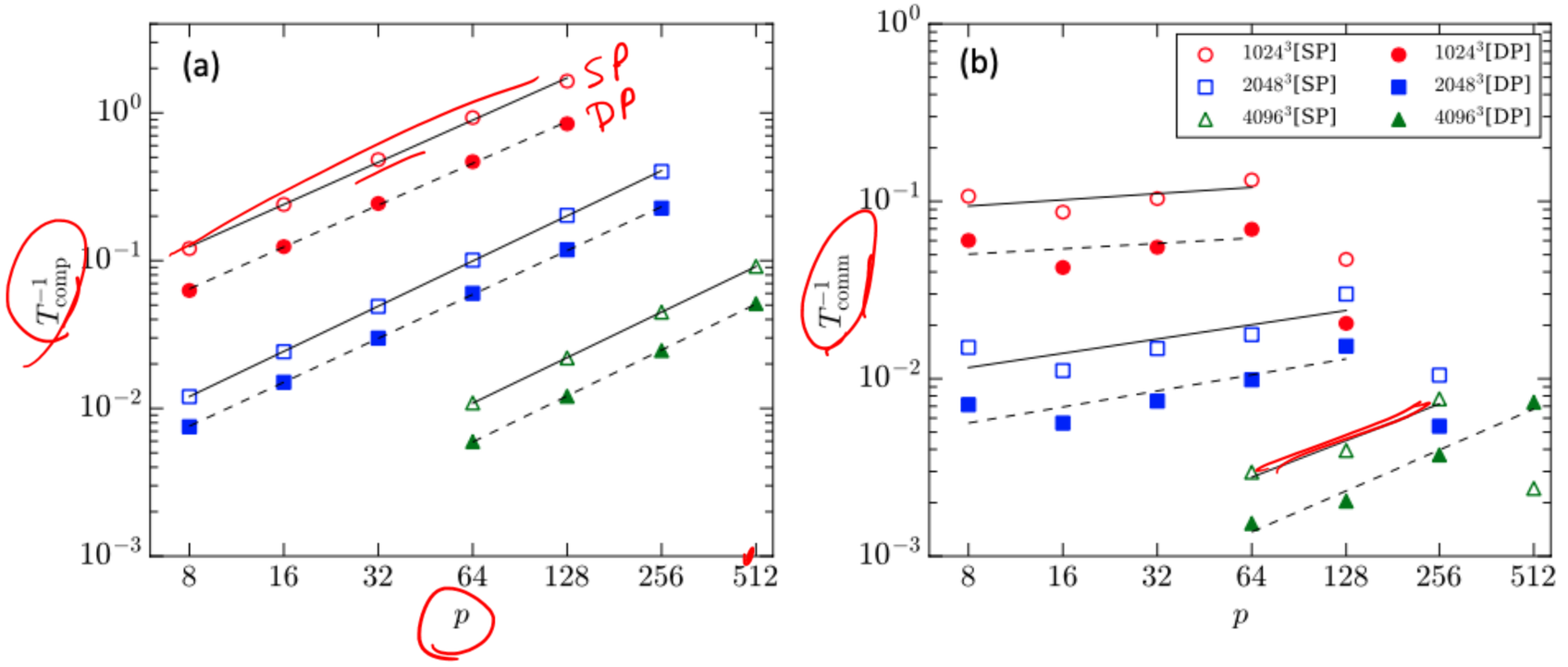


Fig. 2 **a** Plots of inverse computation time, T^{-1}_{comp}, vs. p (number of GPUs) for 1024^3, 2048^3, and 4096^3 grids. **b** Plot of inverse communication time, T^{-1}_{comm}, vs. p. For all the figures, we employ filled symbols for double precision FFTs and unfilled ones for single-precision FFTs. Note that T_{comp} and T_{comm} (in milliseconds) are for a forward-inverse FFT pair

Table 1 Scaling of our code on selene: the exponents γ_1 for the computation time (T_{comp}) scaling, γ_2 for the communication time (T_{comm}) scaling, and γ for the total time (T) scaling

Grid size	γ_1		γ_2		γ	
	SP	DP	SP	DP	SP	DP
1024^3	0.98 ± 0.01	0.97 ± 0.01	0.12 ± 0.11	0.11 ± 0.15	0.35 ± 0.05	0.34 ± 0.07
2048^3	1.00 ± 0.01	1.00 ± 0.002	0.27 ± 0.12	0.30 ± 0.10	0.49 ± 0.06	0.48 ± 0.05
4096^3	1.00 ± 0.02	1.00 ± 0.02	0.69 ± 0.16	0.64 ± 0.13	0.75 ± 0.13	0.71 ± 0.11

Here, we present single-precision (SP) and double-precision (DP) results

Weak scaling

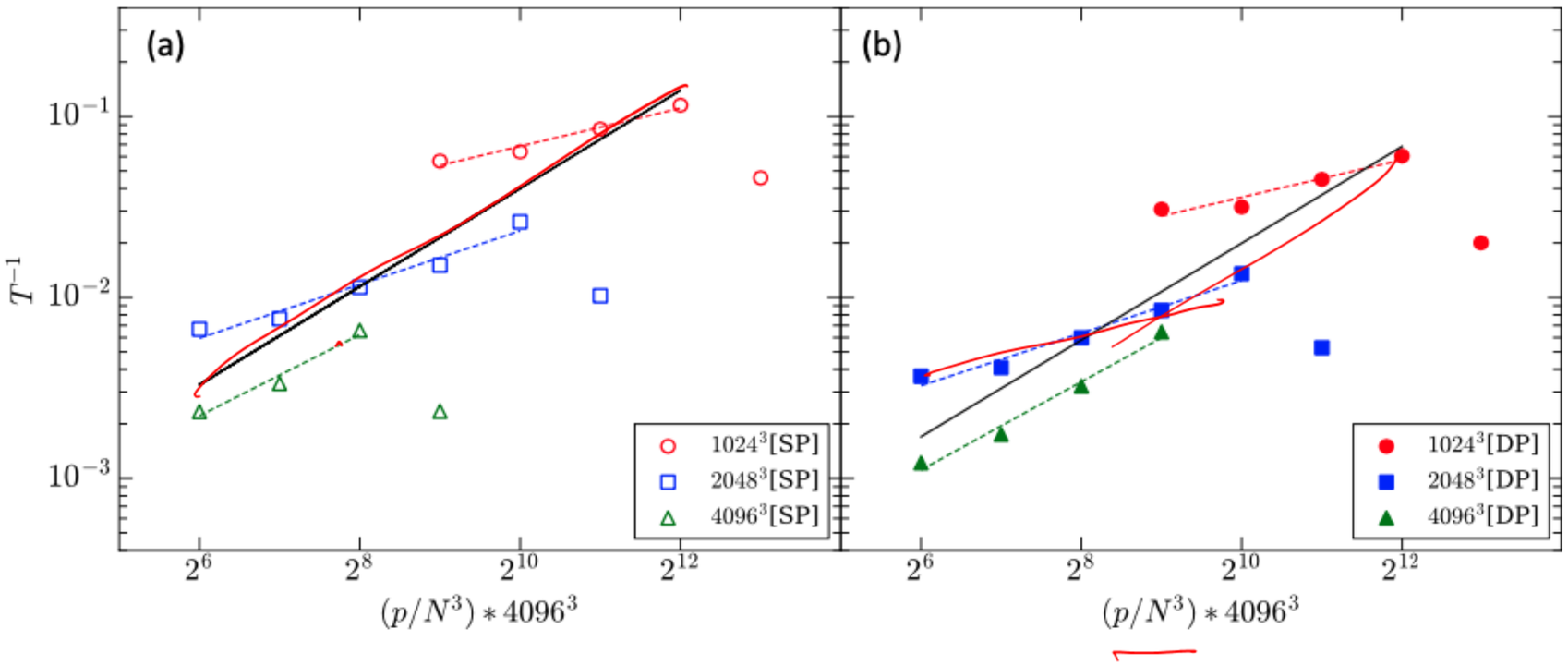


Fig. 4 Plots illustrating the weak scaling (T^{-1} vs. $(p/N^3) * 4096^3$) of our code on *Selene* for 1024^3, 2048^3, and 4096^3 grids

Good Programming Practices

Software Engg.

Planning

- State the problem statement, problem size
- Work out the solution (algorithm)
- Think very carefully and patiently!

C++ Stroup

Which hardware?

- PC
- Server
- cluster (your inst or elsewhere)
- GPU
- International accounts

Software

fftw MIT
fftk (A

- Decide the software tools
- If solution or library available, make use of them.
  - Blitz++, FFT, yaml, Cupy, …. eigen

Which computer language?

- Prototype using Python
- Scientific languages: Fortran, C, C++, Python
- I recommend Python & C++
- C++ because Oop (object-oriented programming)
- Writing large codes are easier in C++. Clear abstraction of tasks!
- Useful features like function overloading, virtual function, classes, template, etc.
- Python programming is fast. Portable to GPUs

- C++: The standard float operations are slower compared to Fortran.
- However, fast libraries are available. Use blitz++, Armadillo, Boost, Eigen
- OpenACC: std::vector

Python vs. C++

- Python, interpreter language, is *typically* slower than C++.
- However, using C libraries and clever programming can make Python code as fast as C++.
- Coding in Python is fast. That's why we prototype in Python first. — 03
- We could write a parallel code in Python.
- We can call efficient C, C++, and Fortran codes in Python code.

Compilers

- GNU ("GNU's Not Unix!")—Free software: gcc
- Clang, LLVM
- Intel compilers
- PGI compilers
- Nvidia compiler
- AMD AOCC for Rome processor

MPI & OpenMP

- OpenMP part of language now: gcc, Intel compilers…
- MPI—free ones: OpenMPI, MPICH3
- Other MPI's: Intel, Cray, …
- Please install MPICH3 & mpi4pi in your laptop/desktop.

Testing

- Test thoroughly!
- Test against exact results (e.g., energy conservation)!
- Test against the limiting cases (viscosity = 0)
- Keep testing frequently.

$$\int |\psi|^2 dx = 1$$

Versioning

- Management of different version of a code.
- Github: free for academic use

Building codes

- A package has 50 to 100 or even larger number of files.
- Use tools to build with dependencies
- CMake, Makefile

Open or closed?

- Opensource or
- Commercial code
- Academic code: keep it open source.
- Which license? GPL, BSD, …
- I recommend BSD.

Patience & perseverance

- TARANG as an example
- Developers: MKV, Anando, Manthan, Soumyadeep
- Versions: C++, Python, CUDA
- A general code for fluid, MHD, convection, …
- Planned for 1 year (2004); Learnt C++
- Googled heavily for FFT, MPI and sought help from friends…

- Make the code better and better..
- Tested thoroughly, organised workshop
- Hunt for hardware: EKA, PARAM YUVA, SHAHEEN
- Contributors: Many students and users
- At present, our code is one of the best spectral codes in the world

Dr. A. P. J. Abdul Kalam Cray HPC award to TARANG

http://mahendra-verma.blogspot.com/search?q=cray

Team work

- Core team
- Part-time developers
- Testers
- Users
- Documenters

Advertise!

- Make a good manual!
- Good website!
- Use social media!
- Conduct developer workshop!
- Conduct user workshop!
- Spread word around!

Exascale challenge

- Applications that scale well on exascale systems.
- Well-designed program
- Optimisation
- Great hardware
- Requires good knowledge of hardware, software

SN Computer Science (2020) 1:178
https://doi.org/10.1007/s42979-020-00184-1

Check for updates

ORIGINAL RESEARCH

SN

Challenges in Fluid Flow Simulations Using Exascale Computing

Mahendra K. Verma[1] · Roshan Samuel[2] · Soumyadeep Chatterjee[1] · Shashwat Bhattacharya[2] · Ali Asad[1]