September 23-24, 2019, Moscow

Russian Supercomputing Days is international supercomputing conference carried on the best traditions of Russian and international supercomputing events.

An Overview of High Performance Computing and Future Requirements

Jack Dongarra

University of Tennessee

Oak Ridge National Laboratory

University of Manchester

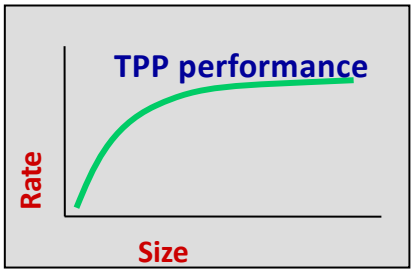
Foreign Member of the Russian Academy of Sciences

Since 1993

H. Meuer, H. Simon, E. Strohmaier, & JD

- Listing of the 500 most powerful Computers in the World
- Yardstick: Rmax from LINPACK MPP

  $Ax=b$, *dense problem*



- Updated twice a year
  - SC‘xy in the States in November
  - Meeting in Germany in June
- All data available from **www.top500.org**

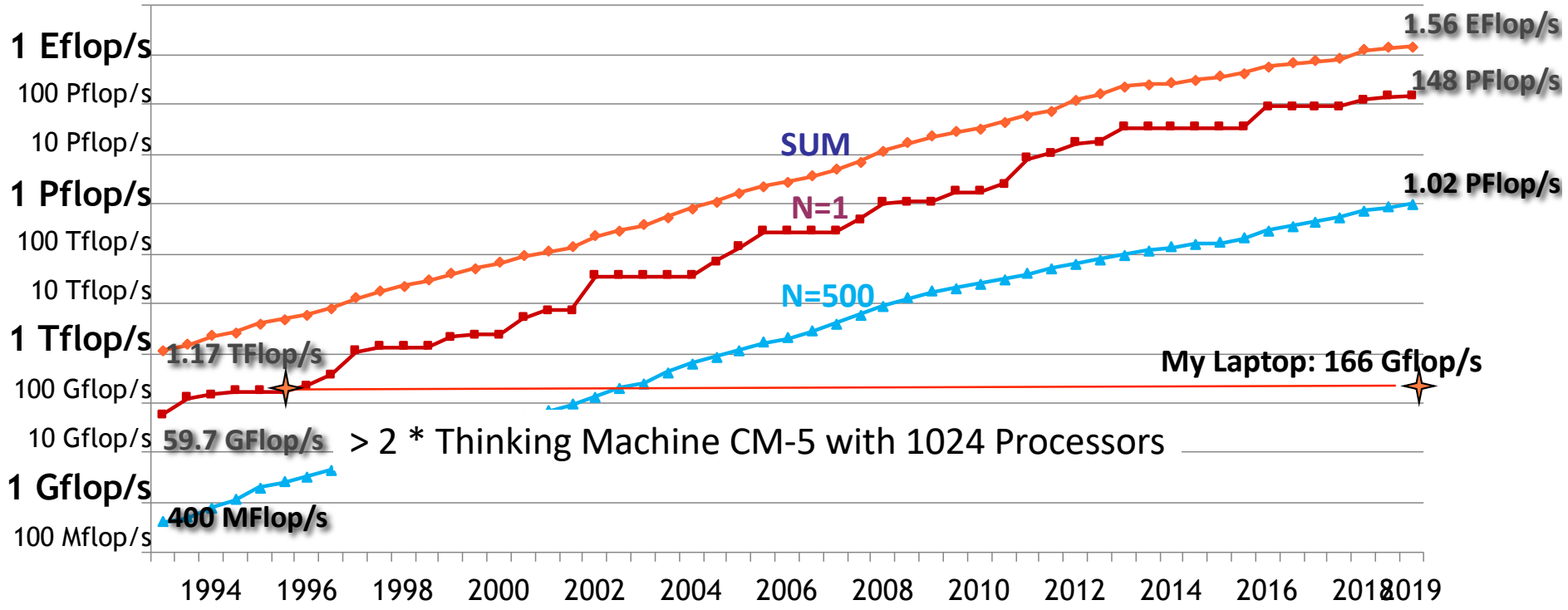
State of Supercomputing in 2019

- Pflops (> 10^{15} Flop/s) computing fully established with all 500 systems.
- Three technology architecture possibilities or "swim lanes" are thriving.
  - Commodity (e.g. Intel)
  - Commodity + accelerator (e.g. GPUs) (133 systems)
  - Special purpose lightweight cores (e.g. IBM BG, Knights Landing, TaihuLight, ARM (only 1 system))
- Interest in supercomputing is now worldwide, and growing in many new markets (~50% of Top500 computers are in industry).
- Intel processors largest share, 96% followed by AMD, .6%.
- Exascale (10^{18} Flop/s) projects exist in many countries and regions.

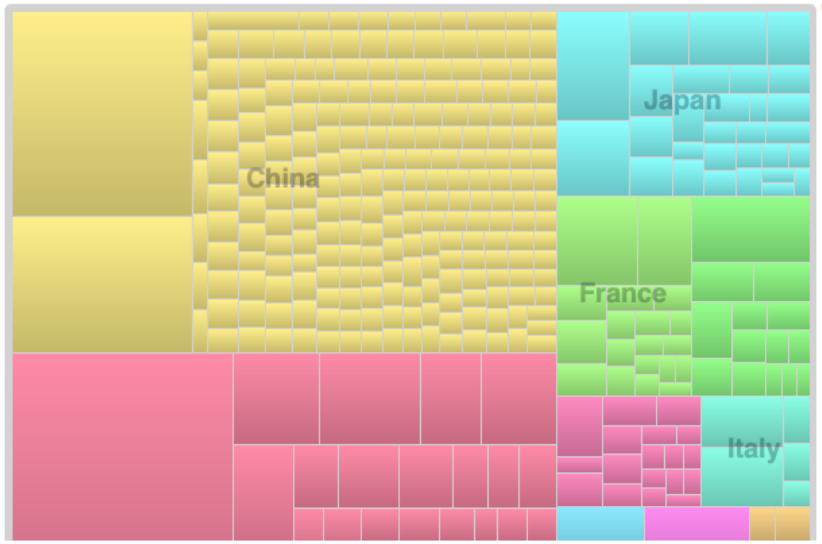
June 2019: The TOP 10 Systems (1/3 of the Total Performance)

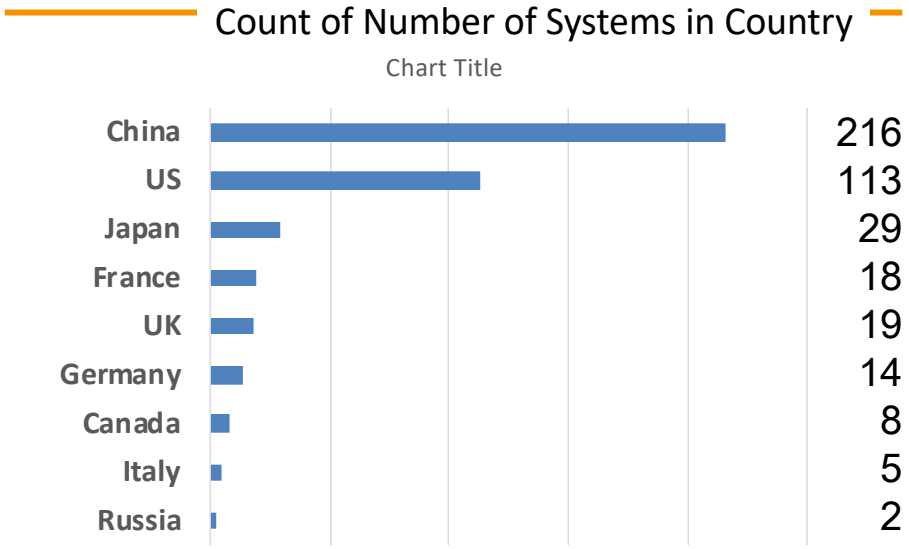
Rank	Site	Computer	Country	Cores	Rmax [Pflops]	% of Peak	Power [MW]	GFlops/ Watt
1	DOE / OS Oak Ridge Nat Lab	Summit, IBM Power 9 (22C, 3.0GHz), Nvidia GV100 (80C), Mellonox EDR	USA	2,397,824	149.	74	11.1	14.7
2	DOE / NNSA L Livermore Nat Lab	Sierra, IBM Power 9 (22C, 3.1GHz), Nvidia GV100 (80C), Mellonox EDR	USA	1,572,480	94.6	75	7.44	12.7
3	National Super Computer Center in Wuxi	Sunway TaihuLight, SW26010 (260C) + Custom	China	10,649,000	93.0	74	15.4	6.05
4	National Super Computer Center in Guangzhou	Tianhe-2A NUDT, Xeon (12C) + MATRIX-2000 + Custom	China	4,981,760	61.4	61	18.5	3.32
5	Texas Advanced Computing Center / U of Texas	Frontera, Dell C6420, Xeon Platinum, 8280 28C 2.7 GHz, Mellanox HDR	USA	448,448	23.5	61		
6	Swiss CSCS	Piz Daint, Cray XC50, Xeon (12C) + Nvidia P100 (56C) + Custom	Swiss	387,872	21.2	78	2.38	8.90
7	DOE / NNSA Los Alamos & Sandia	Trinity, Cray XC40,Xeon Phi (68C) + Custom	USA	979,968	20.2	49	7.58	2.66
8	Nat Inst of Advanced Indust Sci & Tech	AI Bridging Cloud Infrast (ABCI) Fujitsu Xeon (20C, 22.4GHz) Nvidia V100 (80C) IB-EDR	Japan	391,680	16.9	61	1.65	12.05
9	Leibniz Rechenzentrum	SuperMUC-NG, Lenovo,ThinkSystem SD530, Xeon Platinum 8174 24C 3.1GHz, Intel Omni-Path	Germany	311,040	19.5	72		
10	DOE / NNSA L Livermore Nat Lab	Lassen, IBM Power System p9 22C 3.1 GHz, Mellanox EDR, Nvdida V100 (80C)	USA	288,288	18.2	79		

Performance Development of HPC over the Last 26 Years from the Top500



COUNTRIES SHARE





Rank	Computer	Site	Manufacturer	Rmax [TFlop/s]	Rpeak [TFlop/s]
93	Lomonosov 2 T-Platform A-Class Cluster, Xeon E5-2697v3 14C 2.6GHz, Intel Xeon Gold 6126, Infiniband FDR, Nvidia K40m/P-100	Moscow State University - Research Computing Center	T-Platforms	2478	4947
365	Cray XC40, Xeon E5-2697v4 18C 2.3GHz, Aries interconnect	Main Computing Center of Roshydromet	Cray Inc./T-Platforms	1200	1293

Current #1 System Overview

System Performance

- Peak performance of 200 Pflop/s for modeling & simulation
- Peak performance of 3.3 Eflop/s for 16 bit floating point used in for data analytics, ML, and artificial intelligence

Each node has

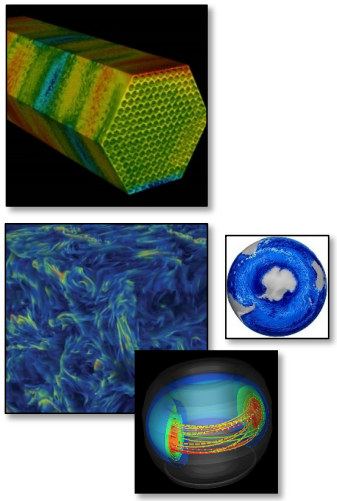
- 2 IBM POWER9 processors
  - Each w/22 cores
  - 2.3% performance of system
- 6 NVIDIA Tesla V100 GPUs
  - Each w/80 SMs
  - 97.7% performance of system
- 608 GB of fast memory
- 1.6 TB of NVMe memory

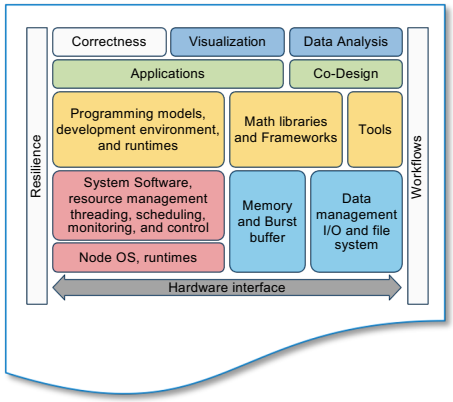
The system includes

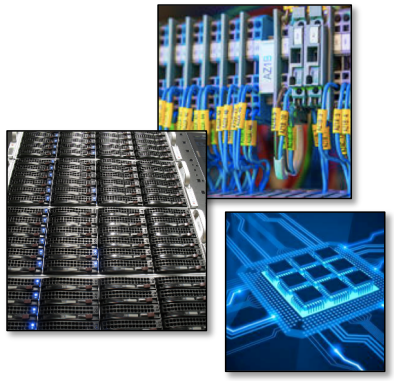
- 4608 nodes
  - 27,648 GPUs
  - Street value $15K each
- Dual-rail Mellanox EDR InfiniBand network
- 250 PB IBM Spectrum Scale file system transferring data at 2.5 TB/s

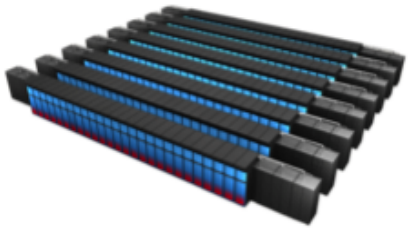
US Department of Energy Exascale Computing Program has formulated a holistic approach that uses co-design and integration to achieve capable exascale

Application Development	Software Technology	Hardware Technology	Exascale Systems
Science and mission applications	Scalable and productive software stack	Hardware technology elements	Integrated exascale supercomputers









ECP's work encompasses applications, system software, hardware technologies and architectures, and workforce development

US Department of Energy (DOE) Roadmap to Exascale Systems

An impressive, productive lineup of *accelerated node* systems supporting DOE's mission

Pre-Exascale Systems [Aggregate Linpack (Rmax) = 323 PF]

First U.S. Exascale Systems

2012	2016	2018	2020	2021-2023

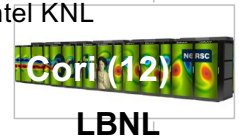
ORNL
Cray/AMD/NVIDIA

ORNL
IBM/NVIDIA

$1.8 B,
Just for 3 Hardware systems at Exascale

ANL
IBM BG/Q

ANL
Cray/Intel KNL



LBNL
Cray/Intel Xeon/KNL

LBNL
Cray/AMD/NVIDIA

FRONTIER

ORNL
AMD/Cray

ANL
Intel/Cray

LLNL
IBM BG/Q

LANL/SNL
Cray/Intel Xeon/KNL

LLNL
IBM/NVIDIA

CROSSROADS

LANL/SNL
TBD

LLNL
Cray/?

ECP EXASCALE COMPUTING PROJECT

HPCG Results; The Other Benchmark

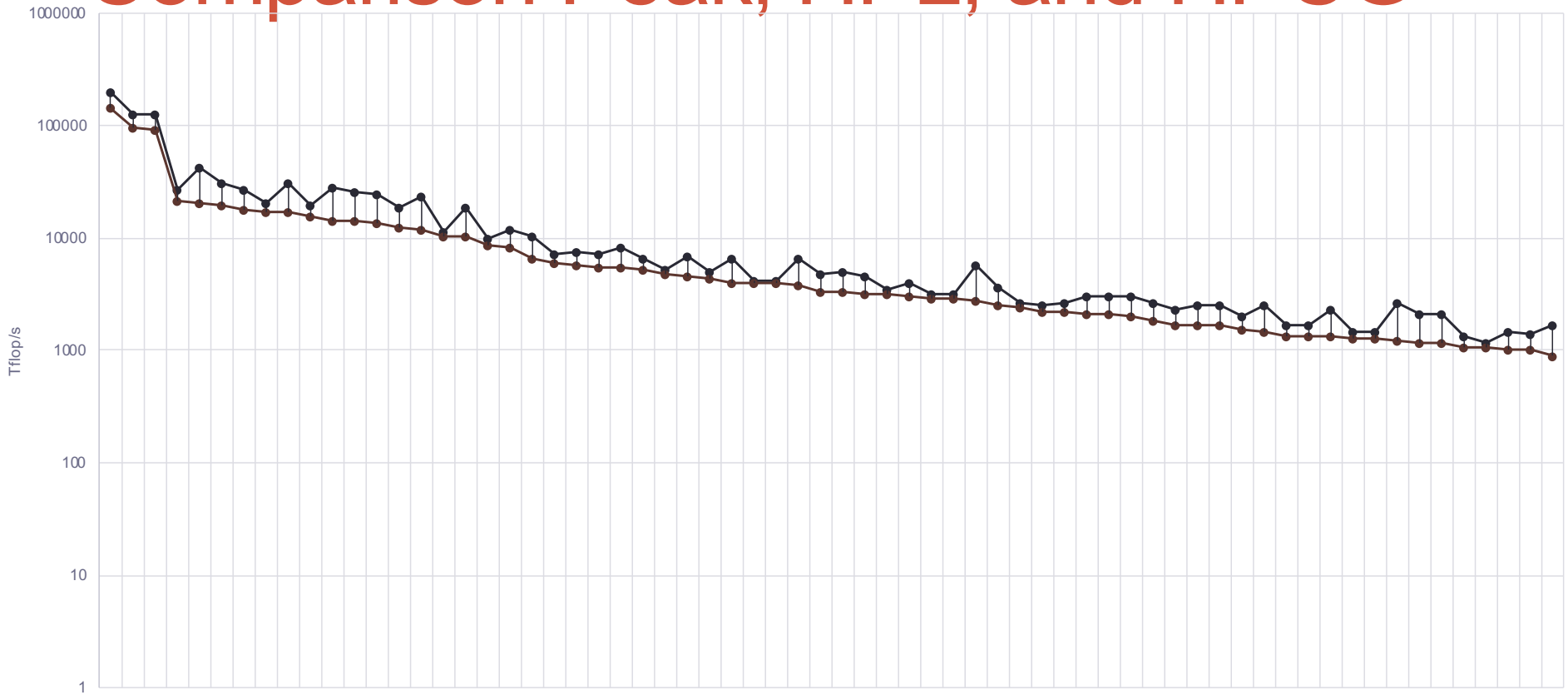
- High Performance Conjugate Gradients (HPCG).
- Solves $Ax=b$, A large, sparse, b known, x computed.
- An optimized implementation of PCG contains essential computational and communication patterns that are prevalent in a variety of methods for discretization and numerical solution of PDEs

- Patterns:
  - Dense and sparse computations.
  - Dense and sparse collectives.
  - Multi-scale execution of kernels via MG (truncated) V cycle.
  - Data-driven parallelism (unstructured sparse triangular solves).
- Strong verification (via spectral properties of PCG).

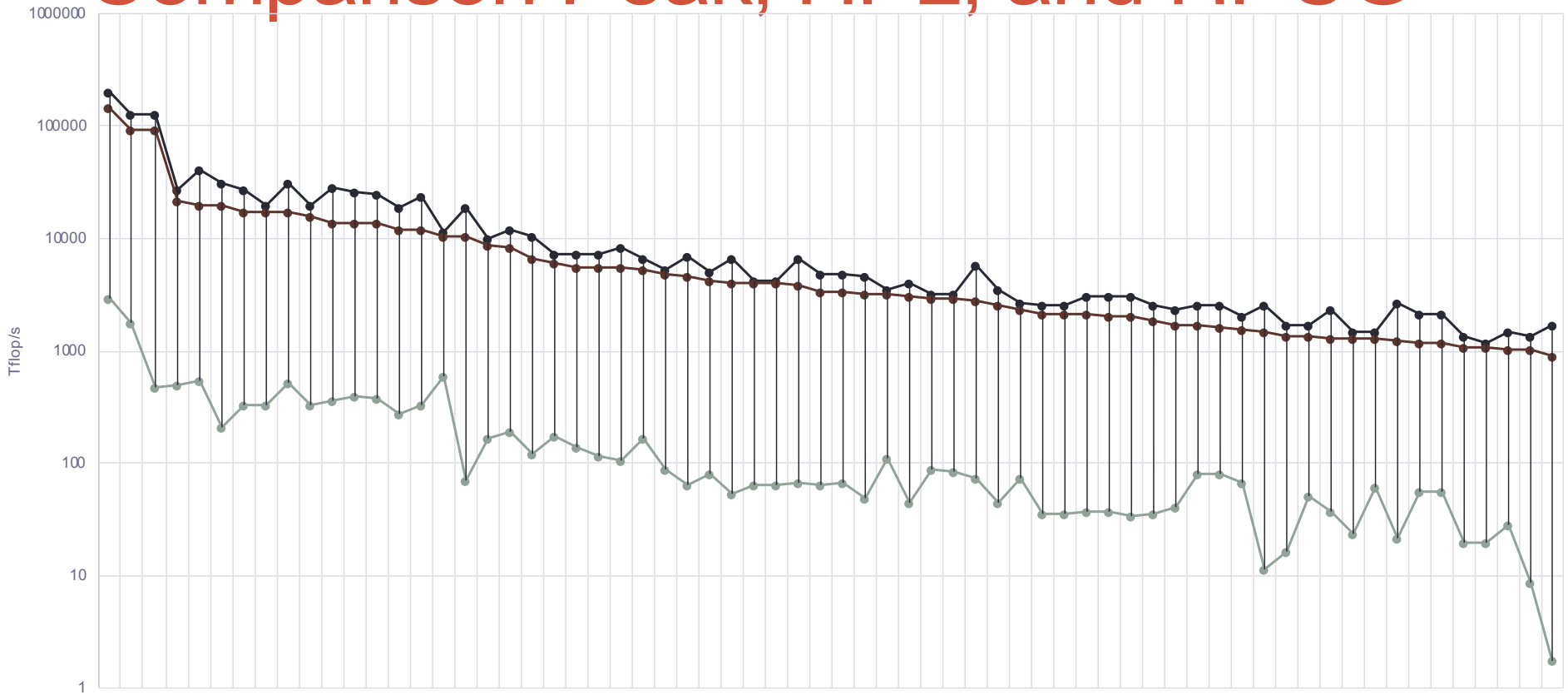
HPCG Benchmark June 2019

Rank	Site	Computer	Cores	HPL Rmax (Pflop/s)	TOP500 Rank	HPCG (Pflop/s)	Fraction of Peak
1	DOE/SC/ORNL USA	**Summit**, AC922, IBM POWER9 22C 3.7GHz, Dual-rail Mellanox FDR, NVIDIA Volta V100, IBM	2,397,824	148.60	1	2.926	1.5%
2	DOE/NNSA/LLNL USA	**Sierra**, S922LC, IBM POWER9 20C 3.1 GHz, Mellanox EDR, NVIDIA Volta V100, IBM	1,572,480	94.64	2	1.796	1.4%
3	RIKEN Advanced Institute for Computational Science Japan	**K computer**, SPARC64 VIIIfx 2.0GHz, Tofu interconnect, Fujitsu	705,024	10.51	18	0.603	5.3%
4	DOE/NNSA/LANL/SNL USA	**Trinity,** Cray XC40, Intel Xeon E5-2698 v3 16C 2.3GHz, Aries, Cray	979,072	20.16	6	0.546	1.3%
5	Natl. Inst. Adv. Industrial Sci. and Tech. (AIST) Japan	**ABCI**, PRIMERGY CX2570M4, Intel Xeon Gold 6148 20C 2.4GHz, Infiniband EDR, NVIDIA Tesla V100, Fujitsu	368,640	16.86	10	0.509	1.7%
6	Swiss National Supercomputing Centre (CSCS) Switzerland	**Piz Daint**, Cray XC50, Intel Xeon E5-2690v3 12C 2.6GHz, Cray Aries, NVIDIA Tesla P100 16GB, Cray	387,872	21.23	5	0.497	1.8%
7	National Supercomputing Center in Wuxi China	**Sunway** TaihuLight, Sunway MPP, SW26010 260C 1.45GHz, Sunway, NRCPC	10,649,600	93.02	3	0.481	0.4%
8	Korea Institute of Science and Technology Information Republic of Korea	**Nurion**, CS500, Intel Xeon Phi 7250 68C 563584C 1.4GHz, Intel Omni-Path, Intel Xeon Phi 7250, Cray	570,020	13.93	13	0.391	1.5%
9	Joint Center for Advanced High Performance Computing Japan	**Oakforest-PACS**, PRIMERGY CX600 M1, Intel Xeon Phi Processor 7250 68C 1.4GHz, Intel Omni-Path Architecture, Fujitsu	556,104	13.55	14	0.385	1.5%
10	DOE/SC/LBNL/NERSC USA	**Cori**, XC40, Intel Xeon Phi 7250 68C 1.4GHz, Cray Aries, Cray	622,336	14.02	12	0.355	1.3%

Comparison Peak, HPL, and HPCG



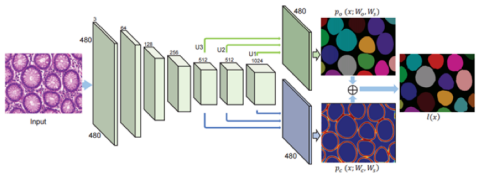
Comparison Peak, HPL, and HPCG

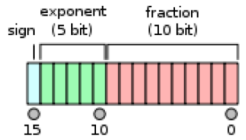


IEEE 754 Half Precision (16-bit) Floating Pt Standard

A lot of interest driven by "machine learning"

- Many fields are beginning to adopt machine learning to augment modeling and simulation methods
  - Climate
  - Biology
  - Drug Design
  - Epidemology
  - Materials
  - Cosmology
  - High-Energy Physics
  - …





AMD

AMD Radeon Instinct	Instinct MI6	Instinct MI8	Instinct MI25
Memory Type	16GB GDDR5	4GB HBM	"High Bandwidth Cache and Controller"
Memory Bandwidth	224GB/sec	512GB/sec	?
Single Precision (FP32)	5.7 TFLOPS	8.2 TFLOPS	12.5 TFLOPS
Half Precision (FP16)	5.7 TFLOPS	8.2 TFLOPS	25 TFLOPS
TDP	<150W	<175W	<300W
Cooling	Passive	Passive (SFF)	Passive
GPU	Polaris 10	Fiji	Vega
Manufacturing Process	GloFo 14nm	TSMC 28nm	?

Tesla Product	Tesla K40	Tesla M40	Tesla P100	Tesla V100
GPU	GK110 (Kepler)	GM200 (Maxwell)	GP100 (Pascal)	GV100 (Volta)
SMs	15	24	56	80
TPCs	15	24	28	40
FP32 Cores / SM	192	128	64	64
FP32 Cores / GPU	2880	3072	3584	5120
FP64 Cores / SM	64	4	32	32
FP64 Cores / GPU	960	96	1792	2560
Tensor Cores / SM	NA	NA	NA	8
Tensor Cores / GPU	NA	NA	NA	640
GPU Boost Clock	810/875 MHz	1114 MHz	1480 MHz	1455 MHz
Peak FP32 TFLOP/s*	5.04	6.8	10.6	15
Peak FP64 TFLOP/s*	1.68	2.1	5.3	7.5
Peak Tensor Core TFLOP/s*	NA	NA	NA	120

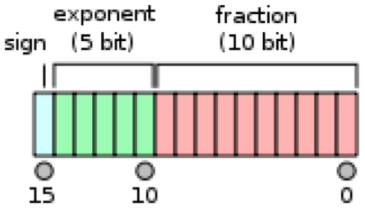
NVIDIA

Mixed Precision

- Today many precisions to deal with (IEEE Standard)

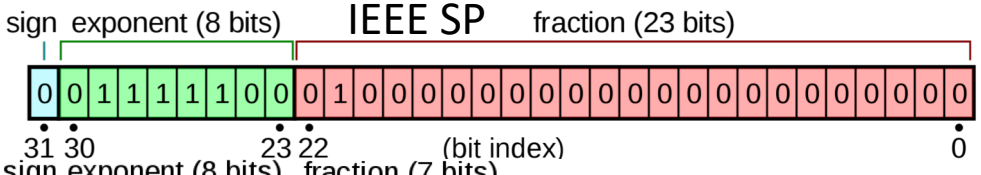
Type	Size	Range	$u = 2^{-t}$
half	16 bits	$10^{\pm 5}$	$2^{-11} \approx 4.9 \times 10^{-4}$
single	32 bits	$10^{\pm 38}$	$2^{-24} \approx 6.0 \times 10^{-8}$
double	64 bits	$10^{\pm 308}$	$2^{-53} \approx 1.1 \times 10^{-16}$
quadruple	128 bits	$10^{\pm 4932}$	$2^{-113} \approx 9.6 \times 10^{-35}$

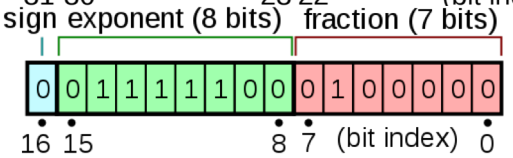
- Note the number range with half precision (16 bit fl.pt.)



float16

largest fl pt number 65,504





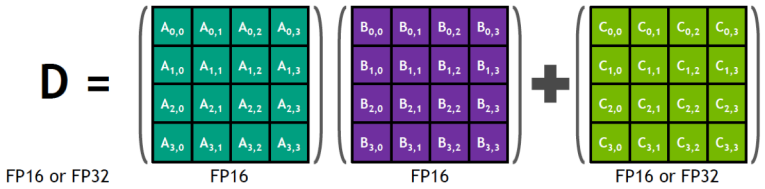
Google TPU: bfloat16

largest fl pt number $O(10^{38})$

Nvidia Volta Peak Rates

- Four Performance levels for the different precision
  - 64 bit floating point (FMA): 7.5 Tflop/s
  - 32 bit floating point (FMA): 15 Tflop/s
  - 16 bit floating point (FMA): 30 Tflop/s
  - 16 bit floating point with Tensor core: 120 Tflop/s
    - Numerical characteristics of arithmetic on Tensor core different

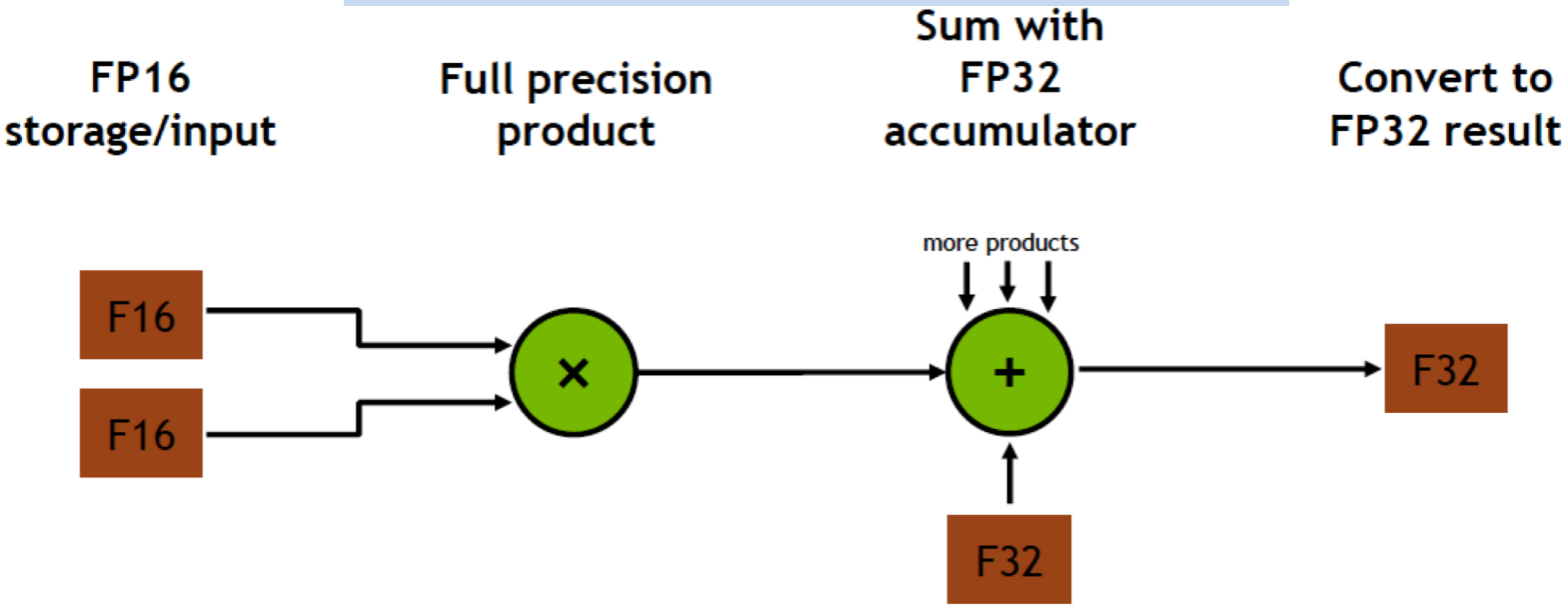
Tensor Core Performance from:
Mixed Precision Matrix Multiply
4x4 Matrices



$$D = AB + C$$

VOLTA TENSOR OPERATION

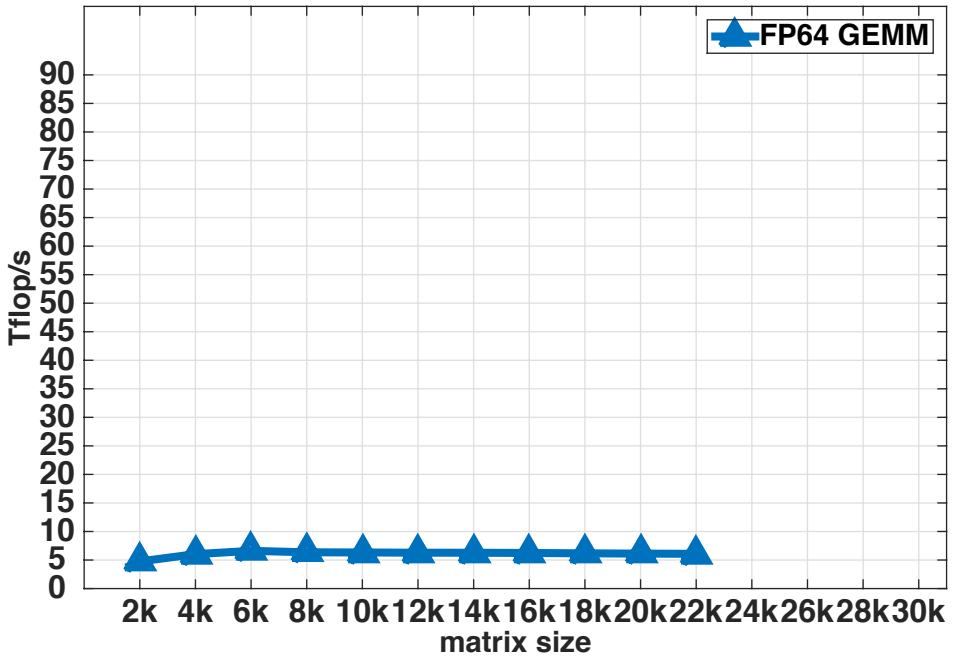
4x4 matrix multiply:
32 bit floating point accuracy with 16 bit inputs



Also supports FP16 accumulator mode for inferencing

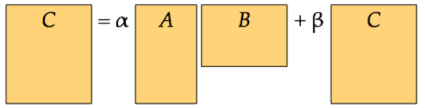
Leveraging Half Precision in HPC on V100

Study of the Matrix Matrix multiplication kernel on Nvidia V100



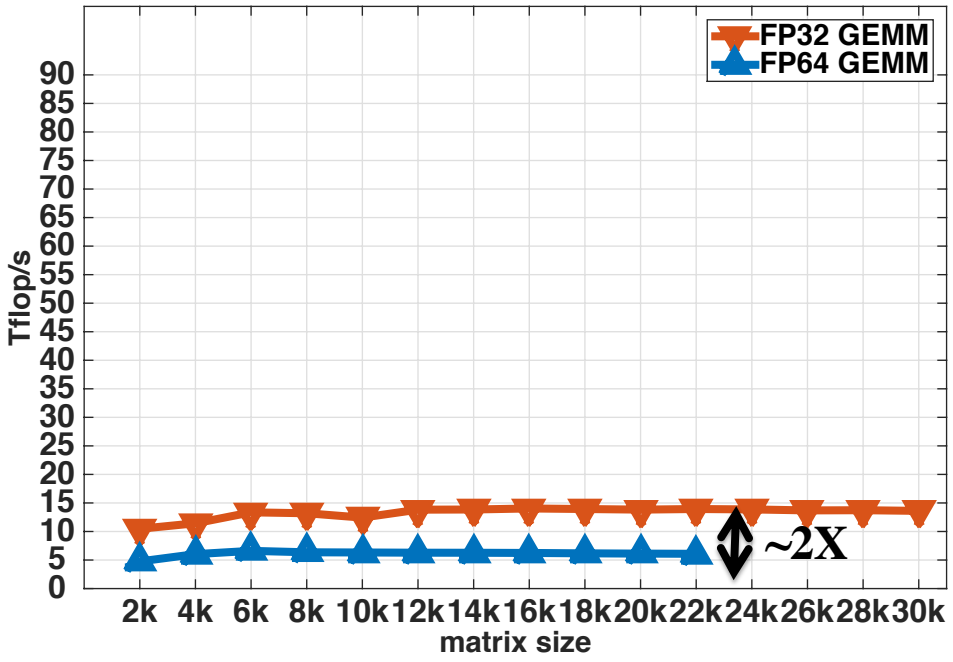
- dgemm achieve about 6.4 Tflop/s

Matrix matrix multiplication GEMM



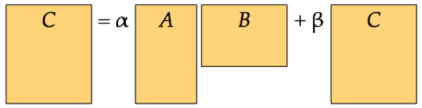
Leveraging Half Precision in HPC on V100

Study of the Matrix Matrix multiplication kernel on Nvidia V100



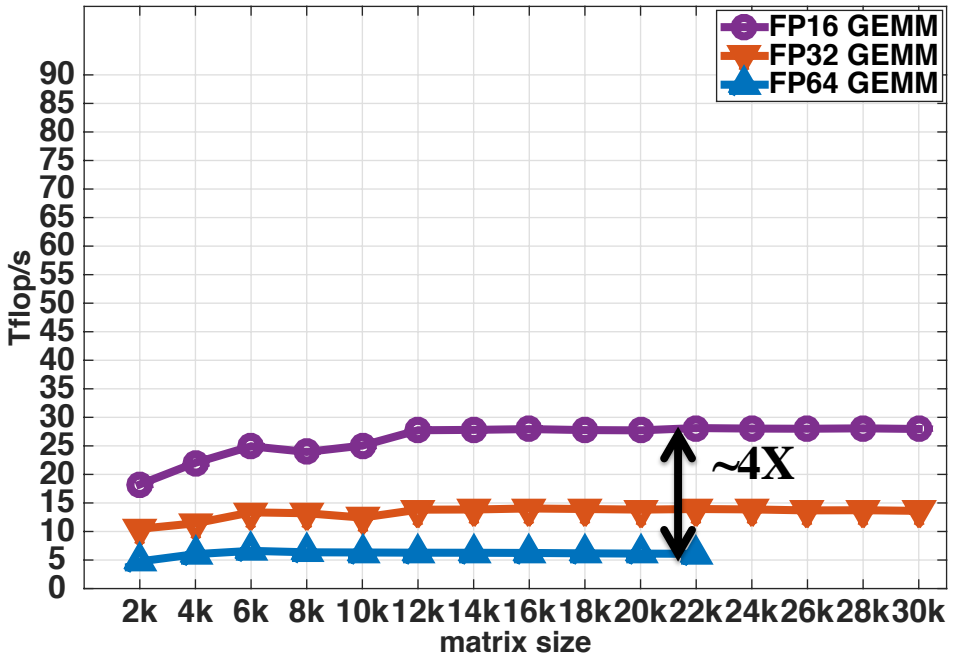
- dgemm achieve about 6.4 Tflop/s
- sgemm achieve about 14 Tflop/s

Matrix matrix multiplication GEMM



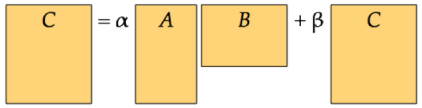
Leveraging Half Precision in HPC on V100

Study of the Matrix Matrix multiplication kernel on Nvidia V100



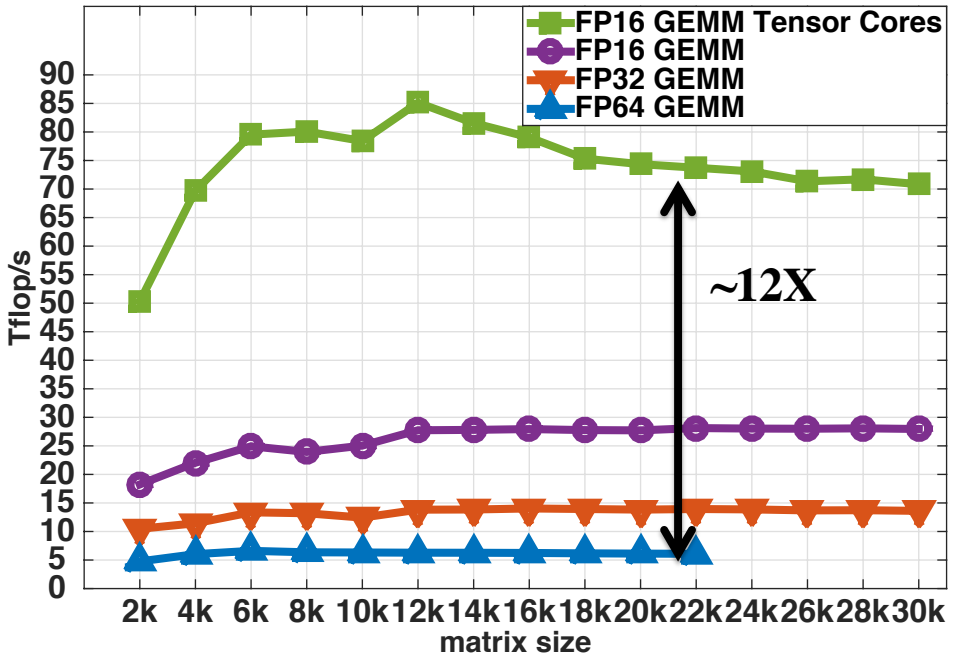
dgemm achieve about 6.4 Tflop/s
sgemm achieve about 14 Tflop/s
hgemm achieve about 27 Tflop/s

Matrix matrix multiplication GEMM



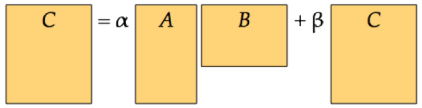
Leveraging Half Precision in HPC on V100

Study of the Matrix Matrix multiplication kernel on Nvidia V100



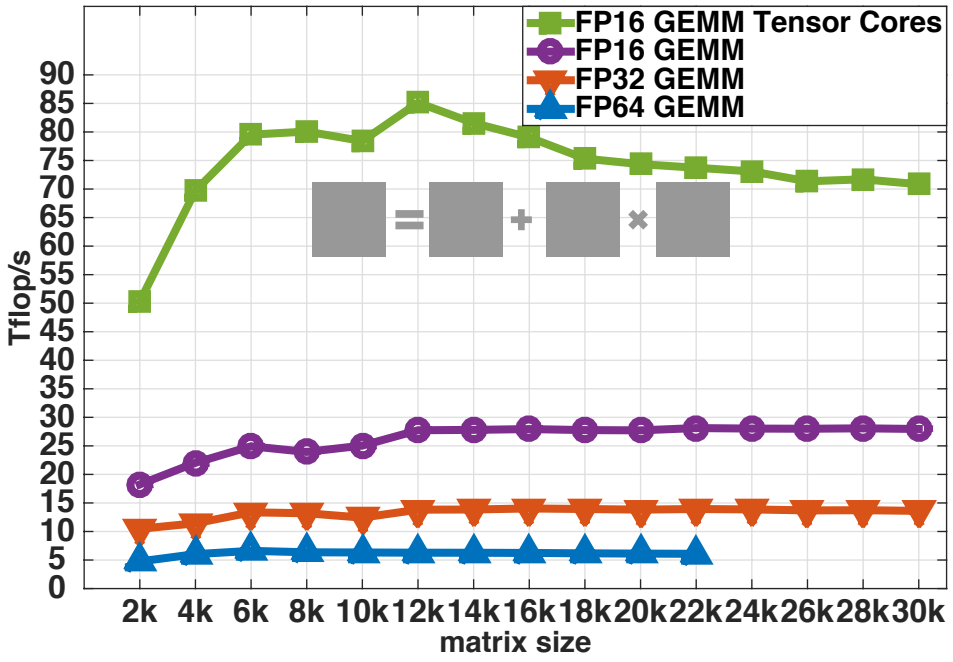
dgemm achieve about 6.4 Tflop/s
sgemm achieve about 14 Tflop/s
hgemm achieve about 27 Tflop/s
Tensor cores gemm reach about 85 Tflop/s

Matrix matrix multiplication GEMM



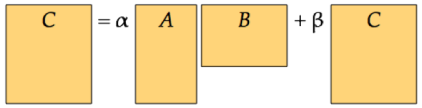
Leveraging Half Precision in HPC on V100

Study of the Matrix Matrix multiplication kernel on Nvidia V100



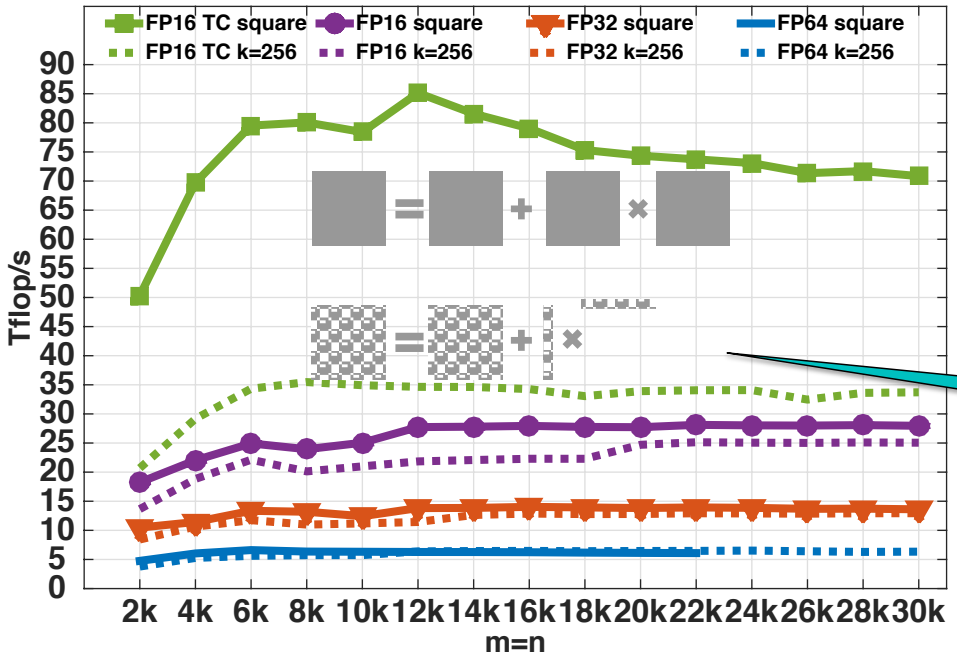
dgemm achieve about 6.4 Tflop/s
sgemm achieve about 14 Tflop/s
hgemm achieve about 27 Tflop/s
Tensor cores gemm reach about 85 Tflop/s

Matrix matrix multiplication GEMM



Leveraging Half Precision in HPC on V100

Study of the rank k update used by the LU factorization algorithm on Nvidia V100



- In LU factorization need matrix multiple but operations is a rank-k update computing the Schur complement

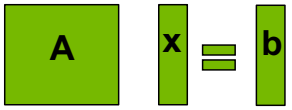
Rank-k GEMM needed by LU does not perform as well as square but still OK

Leveraging Half Precision in HPC on V100
solving linear system Ax = b

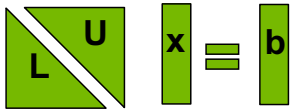
solving linear system Ax = b
LU factorization

- LU factorization is used to solve a linear system Ax=b

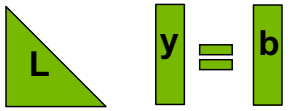
A x = b



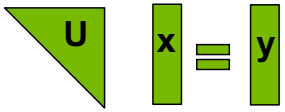
LUx = b



Ly = b



then
Ux = y

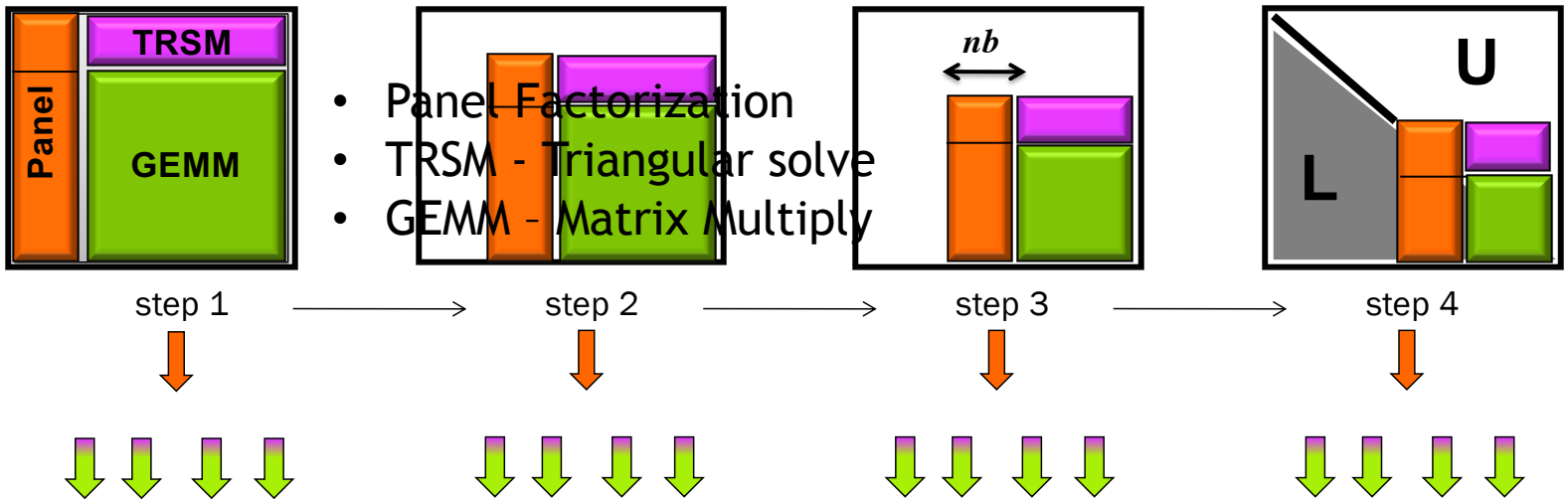


Leveraging Half Precision in HPC on V100
solving linear system Ax = b

For s = 0, nb, .. N

1. panel factorize
2. update trailing matrix

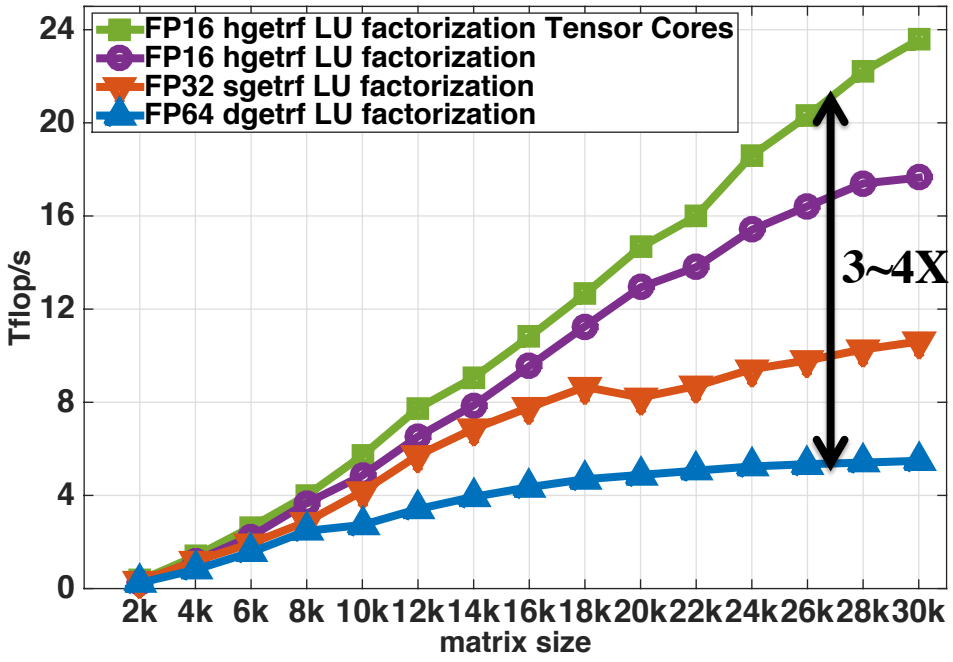
LU factorization requires $O(n^3)$
most of the operations are spent in GEMM



- Panel Factorization
- TRSM - Triangular solve
- GEMM - Matrix Multiply

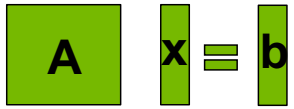
Leveraging Half Precision in HPC on V100

Study of the LU factorization algorithm on Nvidia V100



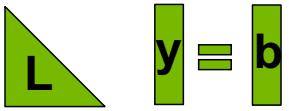
- LU factorization is used to solve a linear system Ax=b

A x = b



LUx = b

Ly = b

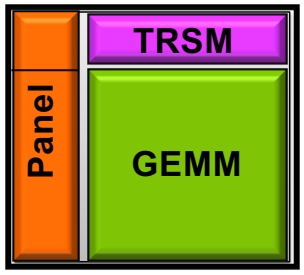


then
Ux = y

Leveraging Half Precision in HPC on V100 solving linear system Ax = b

For s = 0, nb, .. N

1. panel factorize
2. update trailing matrix



- Panel Factorization performed with 32 bit fl pt
  - Done using MAGMA on the front-end system
- TRSM - Triangular solve performed with 32 bit fl pt
  - Done using V100 (no Tensor core)
- GEMM – Matrix Multiply performed with 16 bit fl pt
  - Done on V100 with Tensor cores

Most of the performance comes from GEMM using 16 bit fl pt

Leveraging Half Precision in HPC on V100

Use Mixed Precision algorithms

- Achieve higher performance
  - → faster time to solution
- Reduce power consumption by decreasing the execution time
  - → Energy Savings !!!

Reference:

A. Haidar, P. Wu, S. Tomov, J. Dongarra,
Investigating Half Precision Arithmetic to Accelerate Dense Linear System Solvers,
SC-17, ScalA17: 8th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems, ACM, Denver, Colorado, November 12-17, 2017.

A. Haidar, S. Tomov, J. Dongarra, and N. J. Higham,
Harnessing GPU Tensor Cores for Fast FP16 Arithmetic to Speed up Mixed-Precision Iterative Refinement Solvers, SC-18, Dallas, TX, IEEE, November 2018.

Leveraging Half Precision in HPC on V100

Idea: use low precision to compute the expensive flops (LU $O(n^3)$) and then iteratively refine the solution in order to achieve the FP64 arithmetic

Iterative refinement for dense systems, $Ax = b$, can work this way.

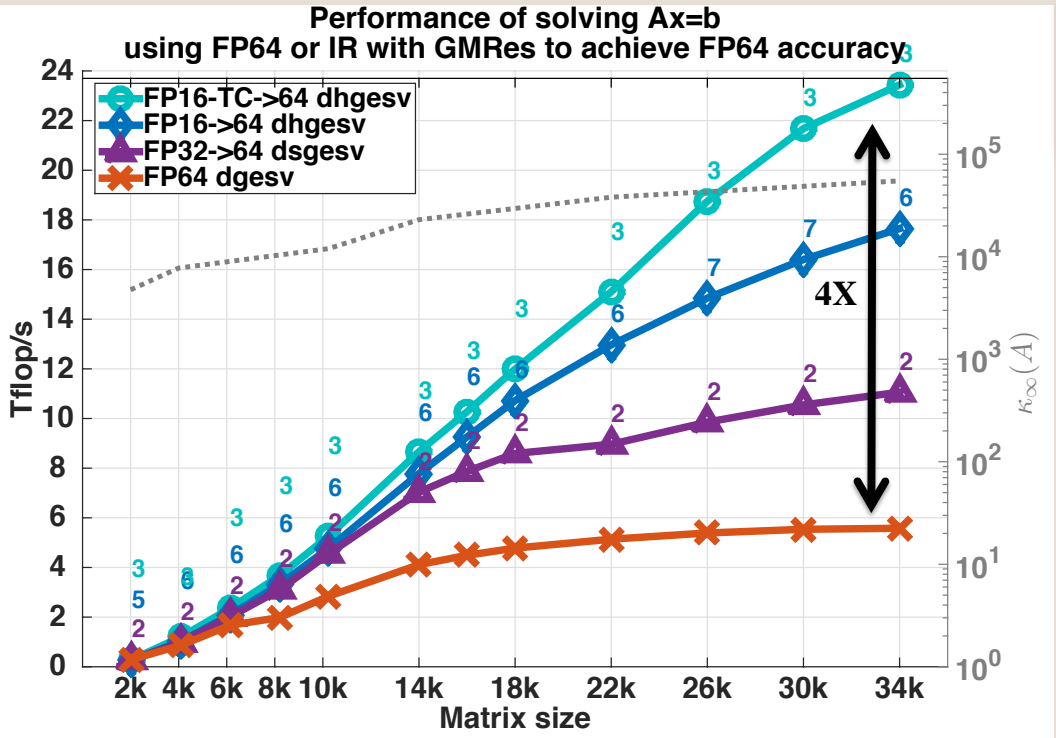
L U = lu(A)		lower precision	$O(n^3)$
x = U\(L\b)		lower precision	$O(n^2)$
r = b – Ax		FP64 precision	$O(n^2)$
WHILE || r || not small enough			
1. find a correction "z" to adjust x that satisfy Az=r solving Az=r could be done by either:			
➢ z = U\(L\r)	Classical Iterative Refinement	lower precision	$O(n^2)$
➢ GMRes preconditioned by the LU to solve Az=r	Iterative Refinement using GMRes	lower precision	$O(n^2)$
2. x = x + z		FP64 precision	$O(n^1)$
3. r = b – Ax		FP64 precision	$O(n^2)$
END			

Higham and Carson showed can solve the inner problem with iterative method and not infect the solution.

- Wilkinson, Moler, Stewart, & Higham provide error bound for SP fl pt results when using DP fl pt.
- It can be shown that using this approach we can compute the solution to 64-bit floating point precision.
- Need the original matrix to compute residual (r) and matrix cannot be too badly conditioned

E. Carson & N. Higham, "Accelerating the Solution of Linear Systems by Iterative Refinement in Three Precisions *SIAM J. Sci. Comput.*, 40(2), A817–A847.

Leveraging Half Precision in HPC on V100
Performance Behavior



Flops = $2n^3/(3 \text{ time})$
meaning twice higher is twice faster

- solving Ax = b using **FP64 LU**
- solving Ax = b using **FP32 LU** and iterative refinement to achieve FP64 accuracy
- solving Ax = b using **FP16 LU** and iterative refinement to achieve FP64 accuracy
- solving Ax = b using **FP16 Tensor Cores LU** and iterative refinement to achieve FP64 accuracy

Problem generated with an arithmetic distribution of the singular values $\sigma_i = 1 - (\frac{i-1}{n-1})(1 - \frac{1}{cond})$ and positive eigenvalues.

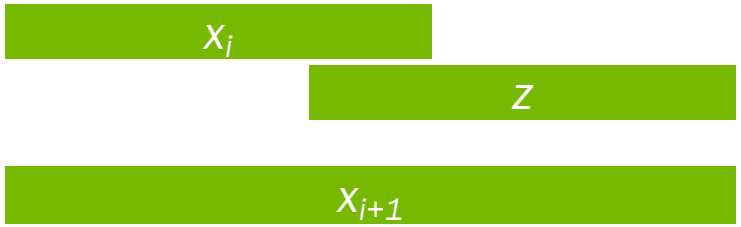
Improving Solution

- z is the correction or $(x_{i+1} - x_i)$
- Computed in lower precision and then added to the approximate solution in higher precision $x_i + z$

```
Iterative refinement for dense systems, Ax = b, can work this way.
L U = lu(A)
x = U\(L\b)
r = b - Ax

WHILE || r || not small enough
     1.  find a correction "z" to adjust x that satisfy Az=r
         solving Az=r could be done by either:


     2.  x = x + z
     3.  r = b - Ax
END
```



- Can be used in situations like this …

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

$$x_{i+1} - x_i = -\frac{f(x_i)}{f'(x_i)}$$

Recent Results Run at Scale…

- Mixed precision iterative refinement approach solved a matrix of order 10,091,520 on ORNL's Summit system.
  - Composed of nodes made up of 2 IBM Power-9 processors (22 cores each) plus 6 Nvidia V100 GPUs (84 SMs each)
  - The run used 4500 nodes of Summit, 2,466,000 cores = 4500*(22*2 + 84*6)
  - Used a random matrix with large diagonal elements to insure convergence of the method.
- Mixed precision HPL achieved 445 PFLOPS or 2.95X over DP precision HPL result on the Top500 (148 PFLOPS).
  - 43 Gflops/Watt
- Same accuracy compared to full 64 bit precision

Conclusion:

- We accelerated the solution of linear system **Ax = b** solver using hardware-accelerated FP16 arithmetic on GPUs;
- We introduced a framework for exploiting mixed-precision FP16-FP32/FP64 iterative refinement solvers and describe the path to draw high-performance and energy-aware GPU implementations;
  - Ideas can be applied to other 1 sided reductions (LU, LL^T, LDL^T, QR) and also for 2 sided in the case of eigenvalue/vectors. Building this into the SLATE LA library (will replace LAPACK and ScaLAPACK, part of DOE-ECP).
- Our technique shows that a number of problems can be **accelerated** up to **4X** by the usage of the **FP16-TC** or **2X** using the **FP32** arithmetic.
- We have rigorous error analysis to support everything.
- Potentially provide an additional benchmarks for ML Supercomputers, looking at mixed precision performance.