An Overview of High Performance Computing and Experiments with Energy Savings and Short Precision

Jack Dongarra

University of Tennessee
Oak Ridge National Laboratory
University of Manchester

Outline

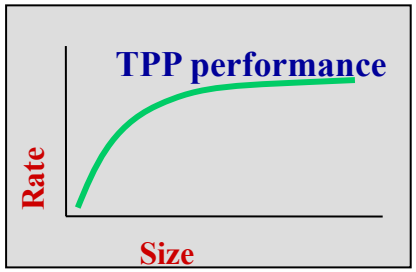
- Overview of High Performance Computing
- With Extreme Computing the “rules” for computing have changed

TOP500 The List. Since 1993

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

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

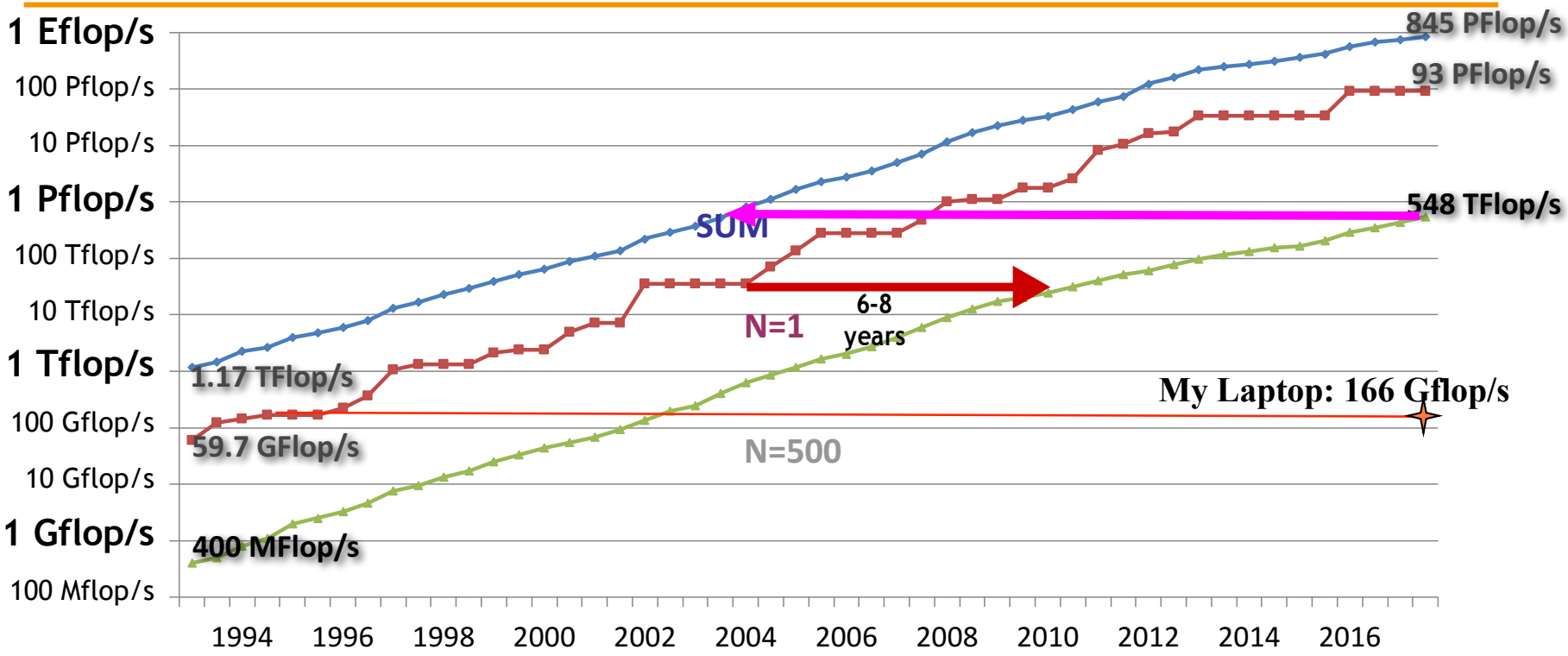
  $Ax=b$, *dense problem*



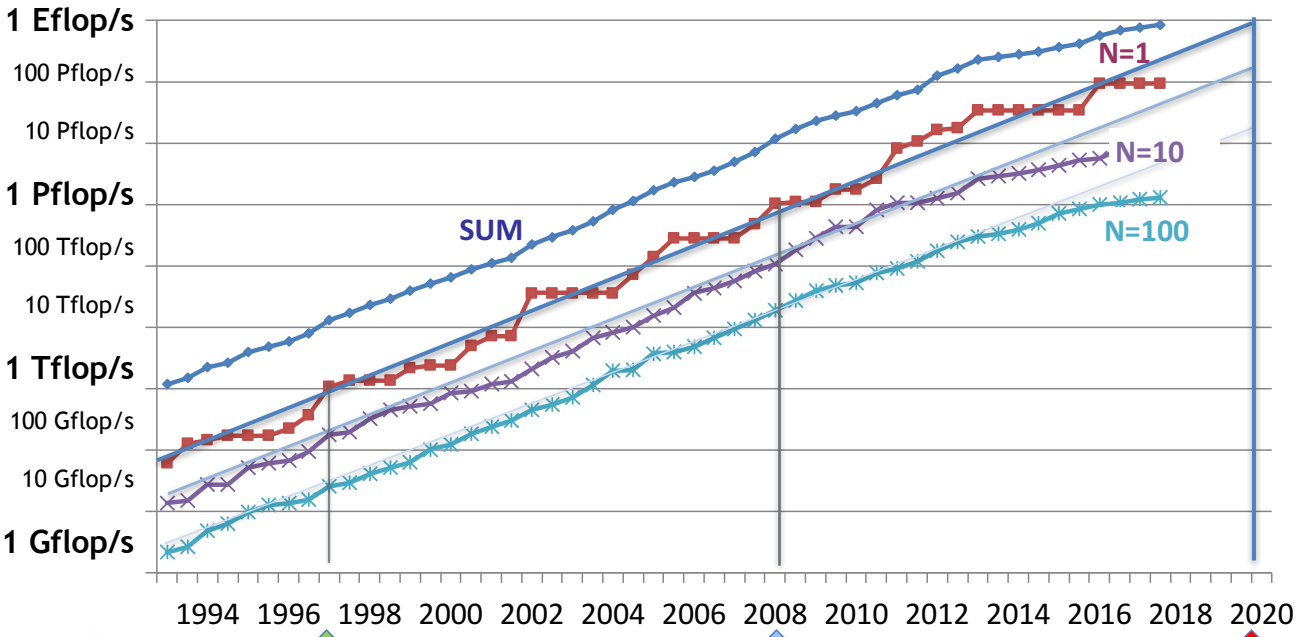
- Updated twice a year
  SC‘xy in the States in November
  Meeting in Germany in June

- All data available from **www.top500.org**

PERFORMANCE DEVELOPMENT OF HPC OVER THE LAST 25 YEARS FROM THE TOP500



PERFORMANCE DEVELOPMENT



≈ Tflops (10^{12}) Achieved ASCI Red Sandia NL

Today: SW 26010 & Intel KNL ~ 3 Tflop/s

≈ Pflops (10^{15}) Achieved RoadRunner Los Alamos NL

Today: 1 cabinet of TaihuLight ~ 3 Pflop/s

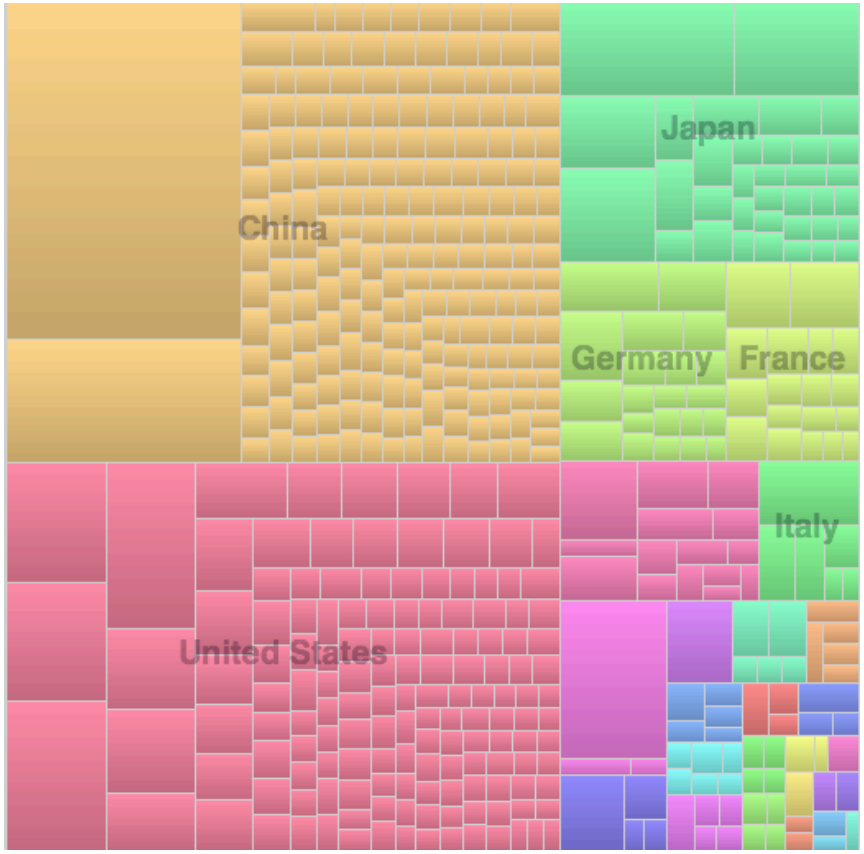
Eflops (10^{18}) Achieved?

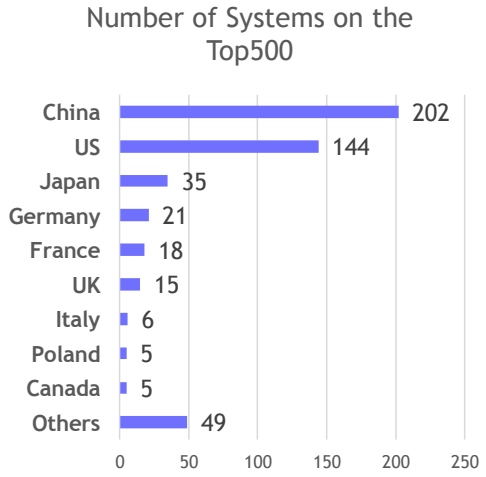
China says 2020
U.S. says 2021

State of Supercomputing in 2018

- Pflops (> 10^{15} Flop/s) computing fully established with 181 systems.
- Three technology architecture possibilities or "swim lanes" are thriving.
  - Commodity (e.g. Intel)
  - Commodity + accelerator (e.g. GPUs) (101 systems)
  - Special purpose lightweight cores (e.g. IBM BG, ARM, Knights Landing, TaihuLight, PEZY-SC2)
- Interest in supercomputing is now worldwide, and growing in many new markets (~50% of Top500 computers are in industry).
- Exascale (10^{18} Flop/s) projects exist in many countries and regions.
- Intel processors largest share, 94% followed by AMD, 1%.

Countries Share





US has fallen to the lowest point since the TOP500 list was created.
China has 35% of the performance
US has 30% of the performance

November 2017: The TOP 10 Systems

Rank	Site	Computer	Country	Cores	Rmax [Pflops]	% of Peak	Power [MW]	GFlops/ Watt
1	National Super Computer Center in Wuxi	Sunway TaihuLight, SW26010 (260C) + Custom	China	10,649,000	93.0	74	15.4	6.04

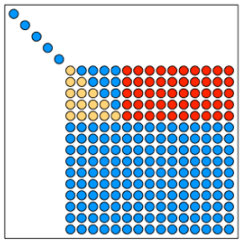
TaihuLight is Equal to Sum of All Top500 Systems in Japan

TaihuLight is about 1/3 the Sum of the other Asian Machines in Top500

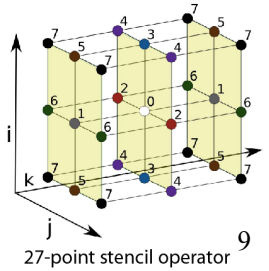
4	Japan Agency Marine-Earth S&T	ExaScalar ZettaScaler-2.2 Xeon (16c) + PEZY-SC2 (2048C)	Japan	19,840,000	19.1	67	1.35	14.1
5	DOE / OS Oak Ridge Nat Lab	Titan, Cray XK7, AMD (16C) + Nvidia Kepler GPU (14C) + Custom	USA	560,640	17.6	65	8.21	2.14
6	DOE / NNSA L Livermore Nat Lab	Sequoia, BlueGene/Q (16C) + custom	USA	1,572,864	17.2	85	7.89	2.18
7	DOE / NNSA / Los Alamos & Sandia	Trinity, Cray XC40,Xeon Phi (68C) + Custom	USA	979,968	14.1	80	3.84	3.67
8	DOE / OS L Berkeley Nat Lab	Cori, Cray XC40, Xeon Phi (68C) + Custom	USA	622,336	14.0	50	3.94	3.65
9	Joint Center for Advanced HPC	Oakforest-PACS, Fujitsu Primergy CX1640, Xeon Phi (68C) + Omni-Path	Japan	558,144	13.6	54	2.72	4.98
10	RIKEN Advanced Inst for Comp Sci	K computer Fujitsu SPARC64 VIIIfx (8C) + Custom	Japan	705,024	10.5	93	12.7	.827
500	Flemish Supercomputer	Intel (14C)		16128	.548	88		

HPL and HPCG

- HPL (Top500) based on solving a dense matrix problem.
  - A x = b
  - A is dense (full of nonzeros)
  - Use a direct solver
  - Gaussian Eli. w/partial pivoting
  - Main driver is matrix multiply
  - Matrix random



- HPCG based on solving a sparse matrix problem.
  - A x = b
  - A is sparse (mainly zeros)
  - Use an iterative solver
  - Conjugate Gradient Method
  - Main driver is matrix vector ops
  - Matrix from a synthetic discretized 3D PDE



27-point stencil operator

HPCG Benchmark November 2017

Rank	Site	Computer	Cores	HPL Rmax (Pflop/s)	TOP500 Rank	HPCG (Pflop/s)	Fraction of Peak
1	RIKEN Advanced Institute for Computational Science Japan	**K computer** – , SPARC64 VIIIfx 2.0GHz, Tofu interconnect, Fujitsu	705,024	10.51	10	0.603	5.3%
2	NSCC / Guangzhou China	**Tianhe-2 (MilkyWay-2)** – TH-IVB-FEP Cluster, Intel Xeon 12C 2.2GHz, TH Express 2, Intel Xeon Phi 31S1P 57-core, NUDT	3,120,000	33.86	2	0.580	1.1%
3	DOE/NNSA/LANL/SNL USA	**Trinity** – Cray XC40, Intel Xeon E5-2698 v3 300160C 2.3GHz, Aries, Cray	979,072	14.13	7	0.546	1.8%
4	Swiss National Supercomputing Centre (CSCS) Switzerland	**Piz Daint** – Cray XC50, Intel Xeon E5-2690v3 12C 2.6GHz, Cray Aries, NVIDIA Tesla P100 16GB, Cray	361,760	19.59	3	0.486	1.9%
5	National Supercomputing Center in Wuxi China	**Sunway TaihuLight** – Sunway MPP, SW26010 260C 1.45GHz, Sunway, NRCPC	10,649,600	93.01	1	0.481	0.4%
6	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	557,056	13.55	9	0.385	1.5%
7	DOE/SC/LBNL/NERSC USA	**Cori** – XC40, Intel Xeon Phi 7250 68C 1.4GHz, Cray Aries, Cray	632,400	13.83	8	0.355	1.3%
8	DOE/NNSA/LLNL USA	**Sequoia** – IBM BlueGene/Q, PowerPC A2 1.6 GHz 16-core, 5D Torus, IBM	1,572,864	17.17	6	0.330	1.6%
9	DOE/SC/Oak Ridge Nat Lab USA	**Titan** – Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x, Cray	560,640	17.59	5	0.322	1.2%
10	GSIC Center, Tokyo Institute of Technology Japan	**TSUBAME3.0** – SGI ICE XA (HPE SGI 8600), IP139-SXM2, Intel Xeon E5-2680 v4 15120C 2.9GHz, Intel Omni-Path Architecture, NVIDIA TESLA P100 SXM2 with NVLink, HPE	136,080	12.12	13	0.189	1.6%

Exascale Race/Technologies

Projected Exascale Dates and Suppliers

U.S.

- Sustained ES: 2023
- Peak ES: 2021
- Vendors: U.S.
- Processors: U.S. (some ARM?)
- Initiatives: NSCI/ECP
- Cost: $300-600M per system, plus heavy R&D investments

EU

- Sustained ES: 2022-2023
- Peak ES: 2021
- Vendors: U.S., Europe
- Processors: Likely ARM
- Initiatives: EuroHPC
- Cost: $300-$350M per system, plus heavy R&D investments

China

- Sustained ES: 2021-2022
- Peak ES: 2020
- Vendors: Chinese
- Processors: Chinese (plus U.S.?)
- 13th 5-Year Plan
- Cost: $350-500M per system, plus heavy R&D

Japan

- Sustained ES: 2022
- Peak ES: Likely as a AI/ML/DL system
- Vendors: Japanese
- Processors: Japanese
- Cost: $800M-$1B, this includes both 1 system and the R&D costs, will also do many smaller size systems

- *Sustained ES on a single 64-bit real application*

Exascale Race/Technologies

Projected Exascale Investment Levels (In Addition to System Purchases)

U.S.

- $1 to $2 billion a year in R&D
- Investments by both governments & vendors
- Plans are to purchases multiple exascale systems each year

EU

- About 5 billion euros in total
- Investments in multiple exascale and pre-exascale systems
- Investments mostly by country governments with a little from the EU

China

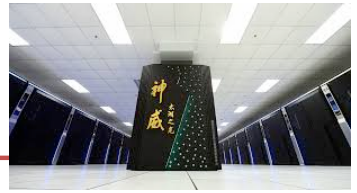
- Over $1 billion a year in R&D
- Investments by both governments & vendors
- Plans are to purchases multiple exascale systems each year
- Already investing in 3 pre-exascale systems starting in late 2018

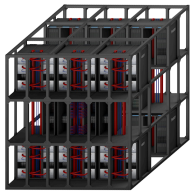
Japan

- Planned investment of just over $1billion* (over 5 years) for both the R&D and purchase of 1 exascale system
- To be followed by a number of smaller systems ~$100M to $150M each
- Creating a new processor and a new software environment

** Note that this includes both the system and R&D*

Toward Exascale





- China plans for Exascale 2020
  - Three separate developments in HPC; "Anything but from the US"
  - Wuxi
    - Upgrade the ShenWei O(100) Pflops all Chinese
  - National University for Defense Technology
    - Tianhe-2A O(100) Pflops will be Chinese ARM processor + accelerator
  - Sugon - CAS ICT
    - X86 based; collaboration with AMD
- US DOE - Exascale Computing Program – 7 Year Program
  - Initial exascale system based on *advanced architecture* and delivered in 2021
  - Enable *capable exascale* systems, based on ECP R&D, delivered in 2022 and deployed in 2023

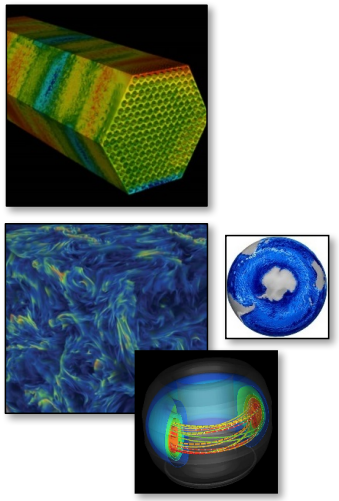
Exascale
- 50X the performance of today's 20 PF systems
- 20-30 MW power
- ≦ 1 perceived fault /week
- SW to support broad range of apps

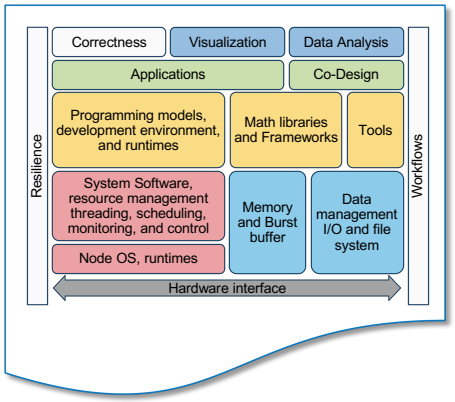
Next Big Systems in the US DOE

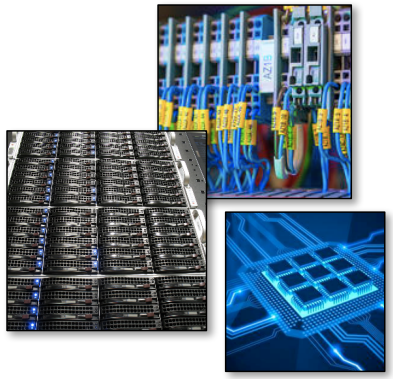
- Oak Ridge Lab and Lawrence Livermore Lab to receive IBM and Nvidia based systems
  - IBM Power 9 + Nvidia Volta V100
  - Installation started but not completed until 2018
  - 5-10 times Titan on apps
  - 4,600 nodes, each containing six 7.5-teraflop NVIDIA V100 GPUs, and two IBM Power9 CPUs, its aggregate peak performance should be > 200 petaflops.
  - ORNL June 2018 fully ready (Top500 number)
- In 2021 Argonne Lab to receive Intel based system
  - Exascale systems, based on a Future Intel proc, not Knights Hill
  - Aurora 21
    - Balanced architecture to support three pillars
    - Large-scale Simulation (PDEs, traditional HPC)
    - Data Intensive Applications (science pipelines)
    - Deep Learning and Emerging Science AI
    - Integrated computing, acceleration, storage
    - Towards a common software stack

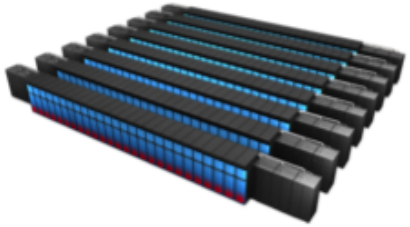
US Department of Energy Exascale Computing Program has formulated a holistic approach that uses co-design and integration to achieve capable exascale

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







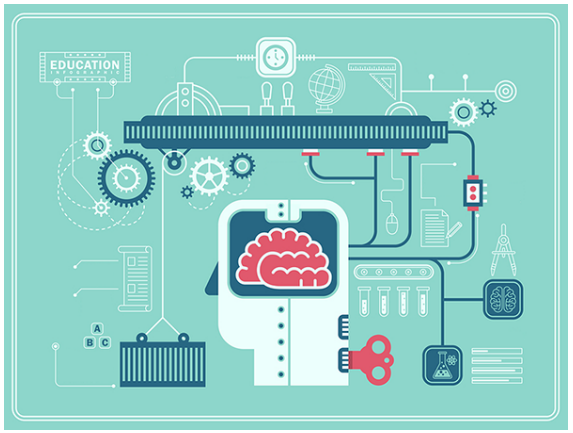


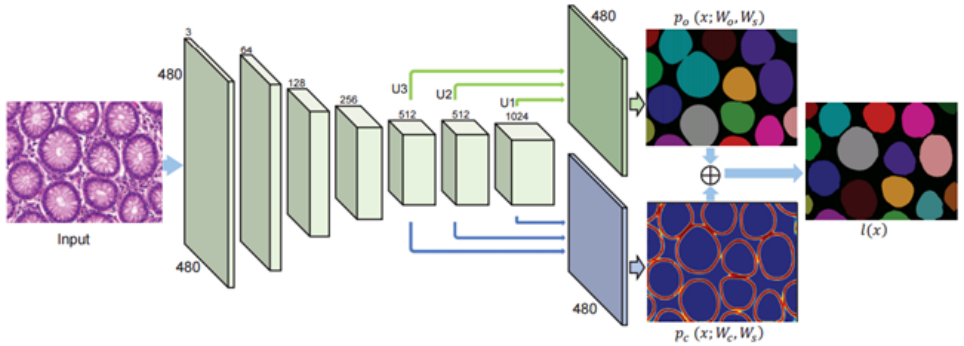
ECP's work encompasses applications, system software, hardware technologies and architectures, and workforce development

Machine Learning in Computational Science

Many fields are beginning to adopt machine learning to augment modeling and simulation methods

- Climate
- Biology
- Drug Design
- Epidemology
- Materials
- Cosmology
- High-Energy Physics



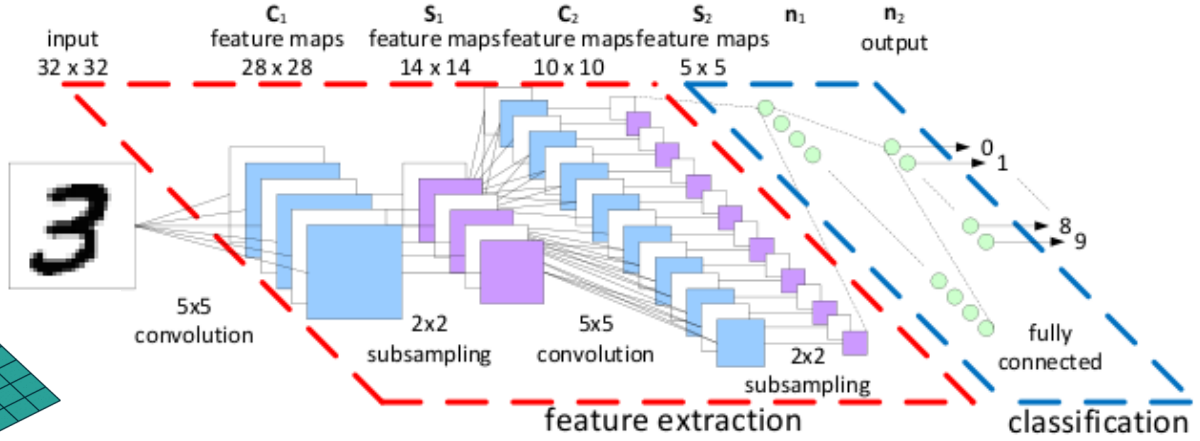


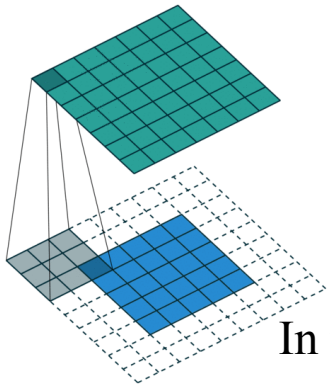
Deep Learning Needs Small Matrix Operations

Matrix Multiply is the time consuming part.

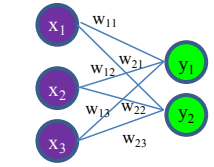
Convolution Layers and Fully Connected Layers require matrix multiply

There are many GEMM's of small matrices, perfectly parallel, can get by with 16-bit floating point





Convolution Step
In this case 3x3 GEMM

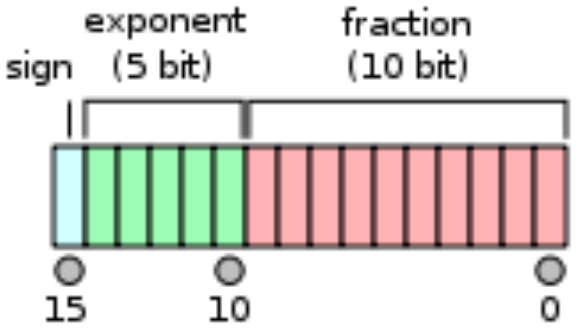


$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Fully Connected
Classification

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

A lot of interest driven by "machine learning"



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	?

intel

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

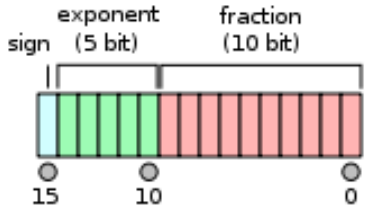
NVIDIA

Mixed Precision

- Today many precisions to deal with

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	$10^{\pm 4932}$	$2^{-113} \approx 9.6 \times 10^{-35}$

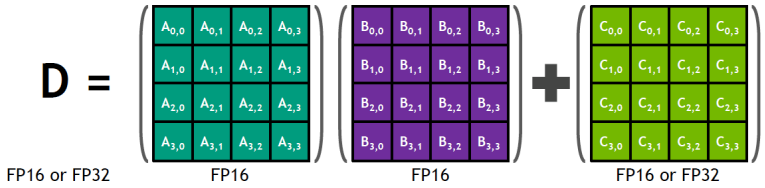
- Note the number range with half precision (16 bit fl.pt.)



Nvidia Volta peak rates

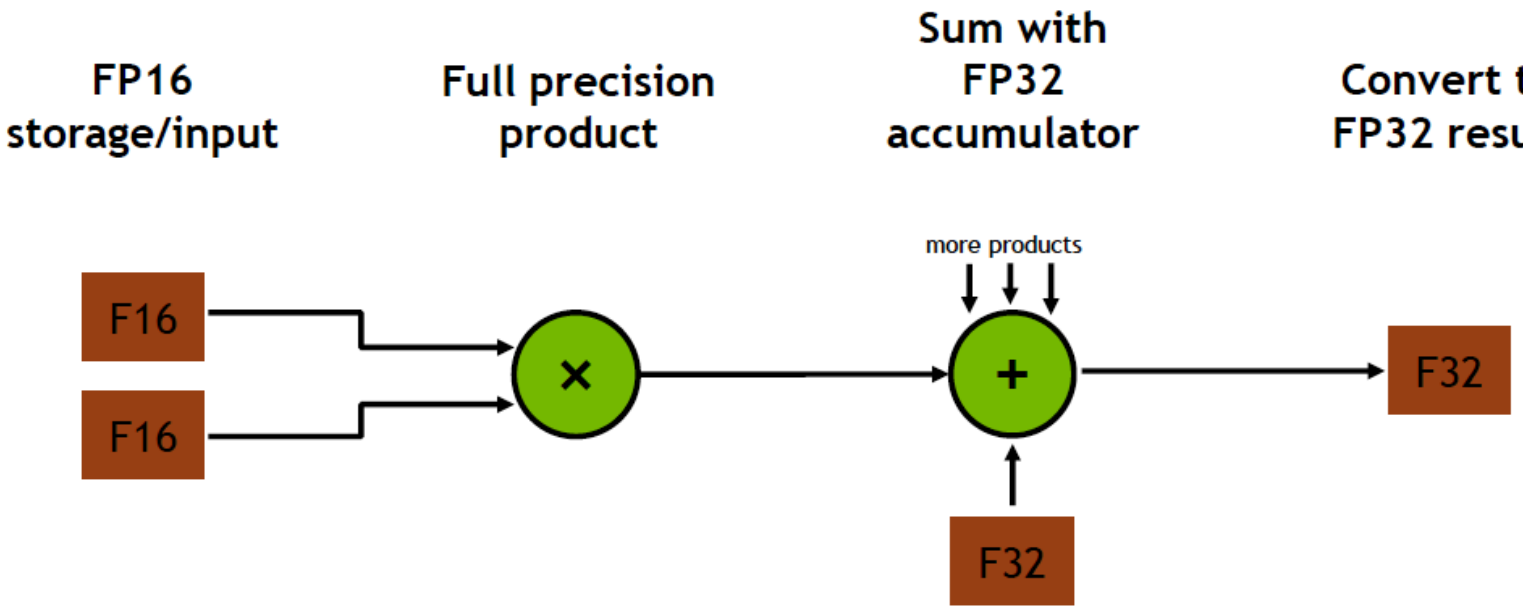
- 64 bit floating point (FMA): 7.5 Tflop/s
- 32 bit floating point (FMA): 15 Tflop/s
- 16 bit floating point (FMA): 30 Tflop/s
- 16 bit floating point with Tensor core: 120 Tflop/s

Mixed Precision Matrix Math
4x4 matrices



D = AB + C

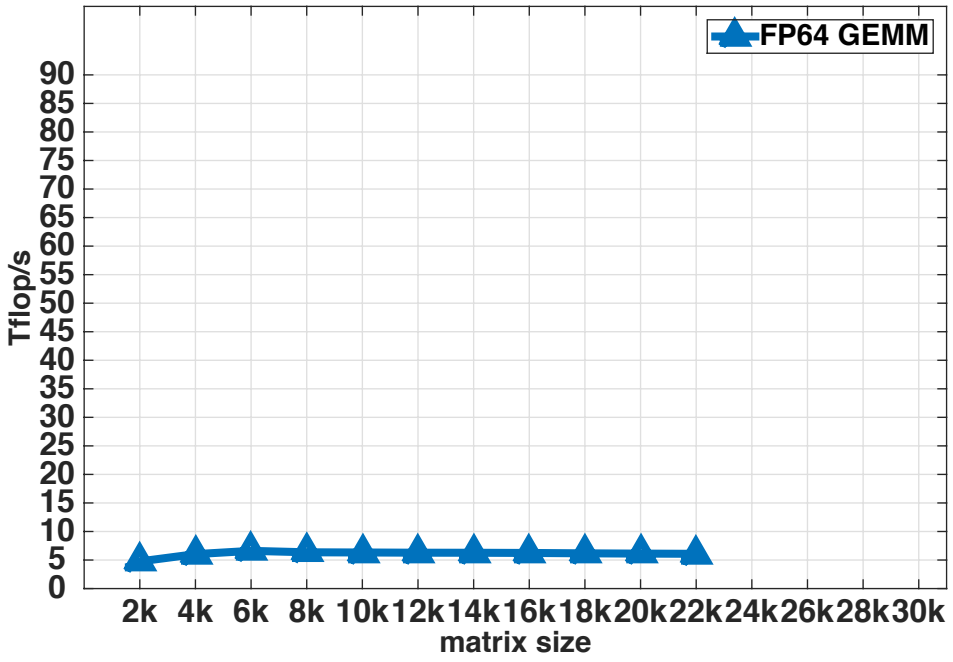
VOLTA TENSOR OPERATION



Also supports FP16 accumulator mode for inferencing

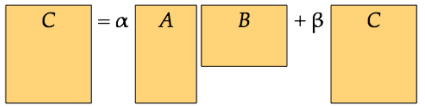
Leveraging Half Precision in HPC on V100

Study of the Matrix Matrix multiplication kernel on Nvidia V100



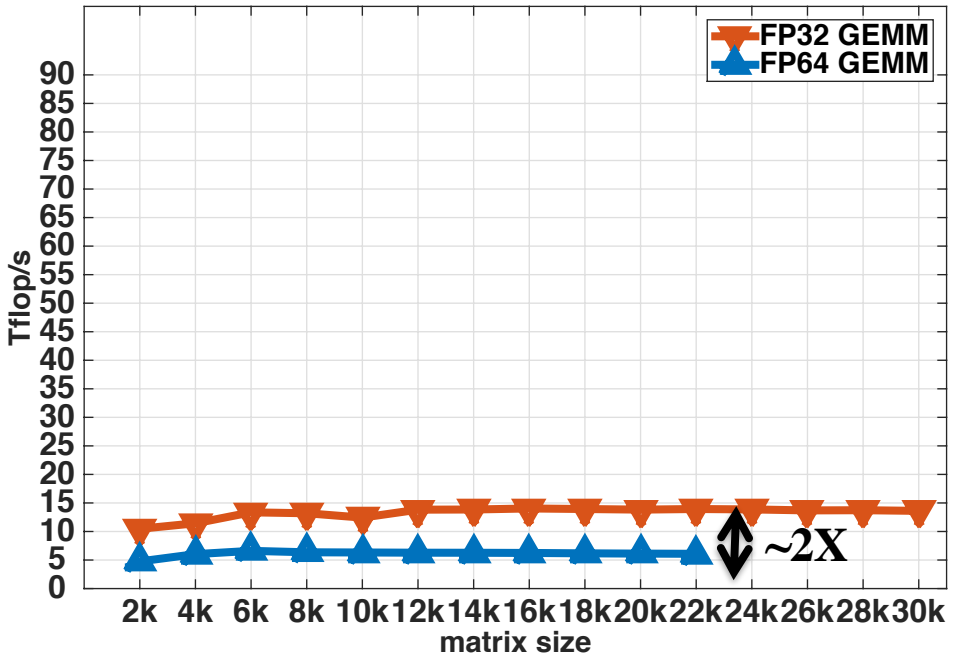
- dgemm achieve about 6.4 Tflop/s

Matrix matrix multiplication GEMM



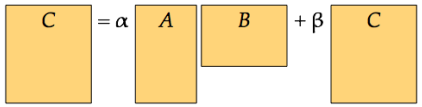
Leveraging Half Precision in HPC on V100

Study of the Matrix Matrix multiplication kernel on Nvidia V100



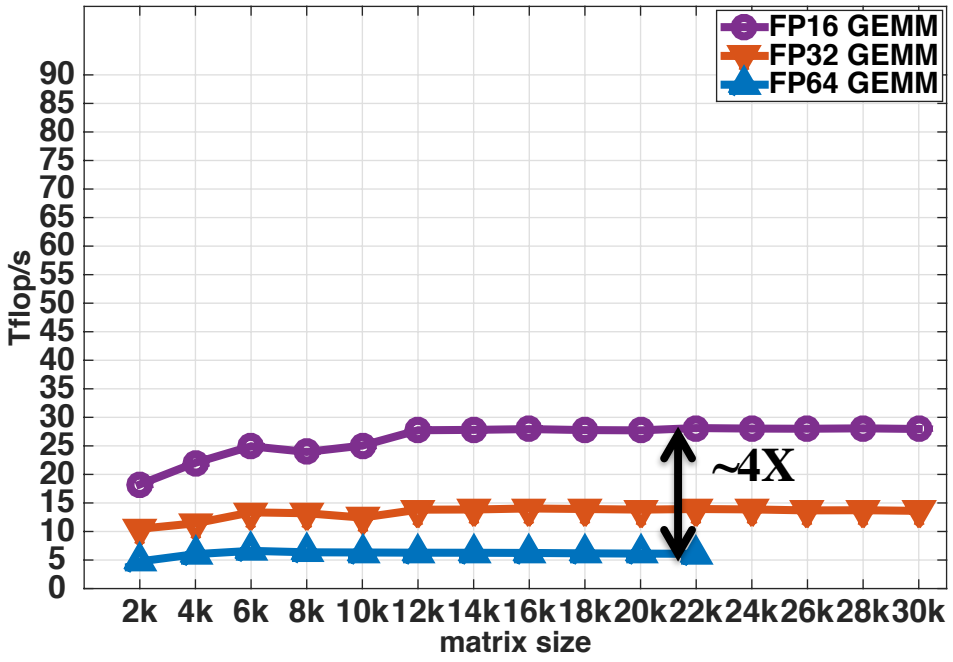
- dgemm achieve about 6.4 Tflop/s
- sgemm achieve about 14 Tflop/s

Matrix matrix multiplication GEMM



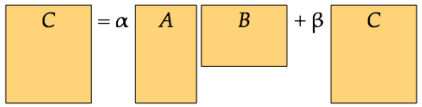
Leveraging Half Precision in HPC on V100

Study of the Matrix Matrix multiplication kernel on Nvidia V100



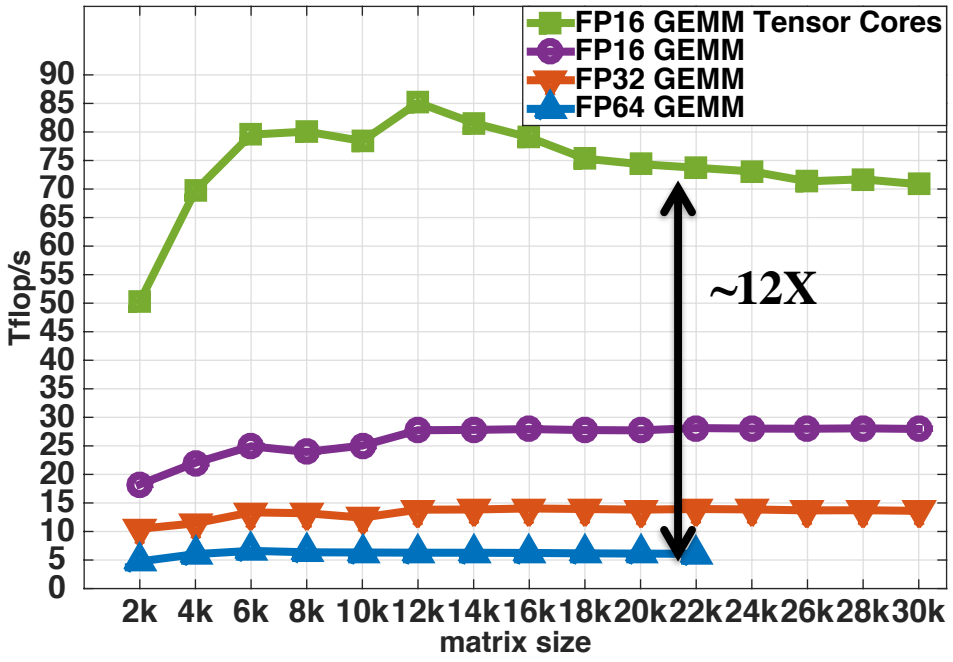
dgemm achieve about 6.4 Tflop/s
sgemm achieve about 14 Tflop/s
hgemm achieve about 27 Tflop/s

Matrix matrix multiplication GEMM



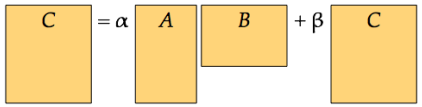
Leveraging Half Precision in HPC on V100

Study of the Matrix Matrix multiplication kernel on Nvidia V100



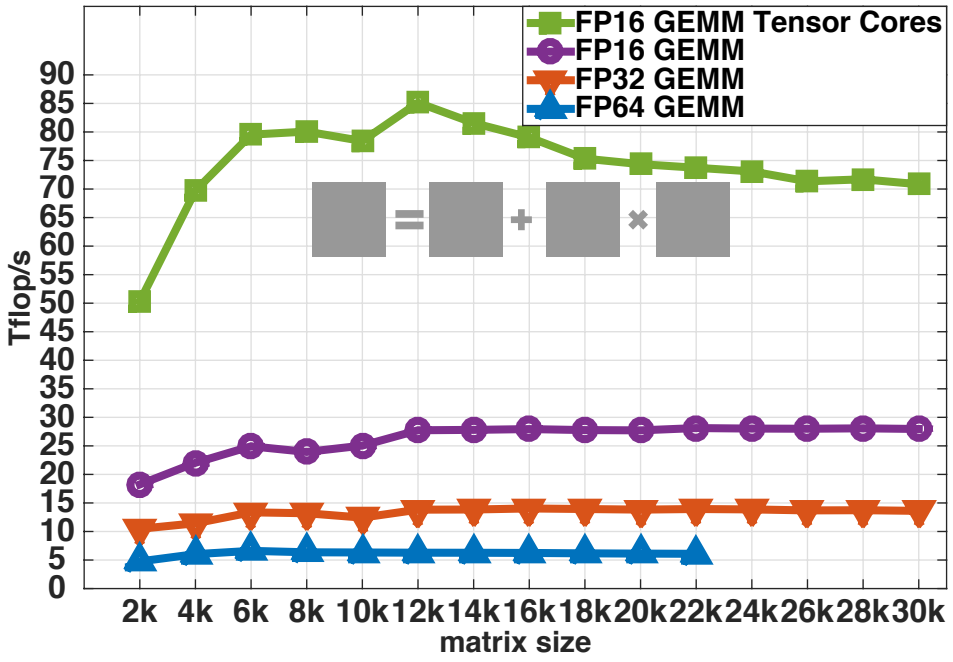
dgemm achieve about 6.4 Tflop/s
sgemm achieve about 14 Tflop/s
hgemm achieve about 27 Tflop/s
Tensor cores gemm reach about 85 Tflop/s

Matrix matrix multiplication GEMM



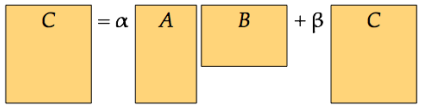
Leveraging Half Precision in HPC on V100

Study of the Matrix Matrix multiplication kernel on Nvidia V100



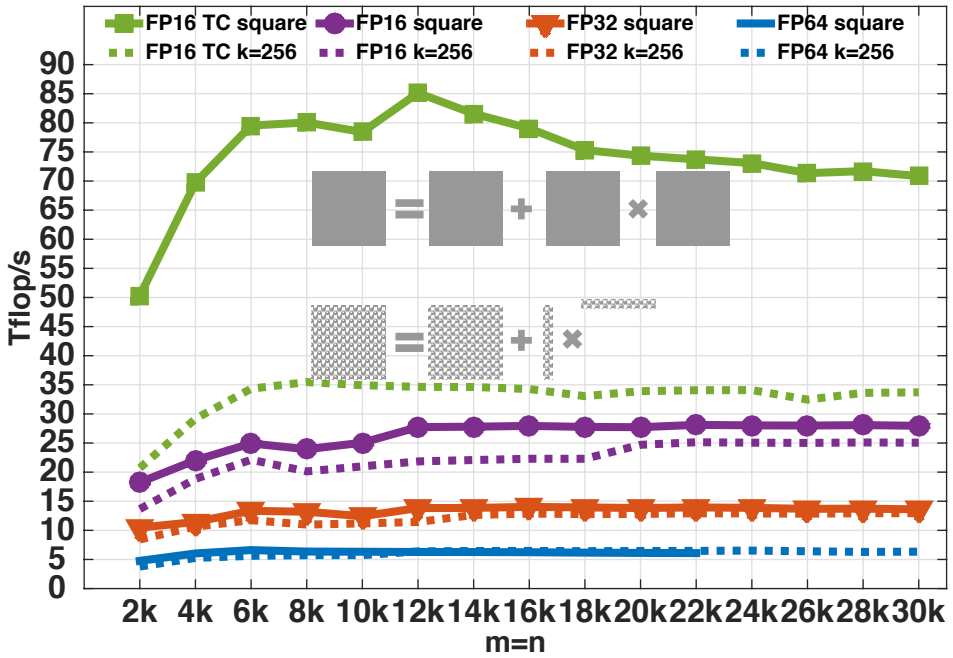
dgemm achieve about 6.4 Tflop/s
sgemm achieve about 14 Tflop/s
hgemm achieve about 27 Tflop/s
Tensor cores gemm reach about 85 Tflop/s

Matrix matrix multiplication GEMM



Leveraging Half Precision in HPC on V100

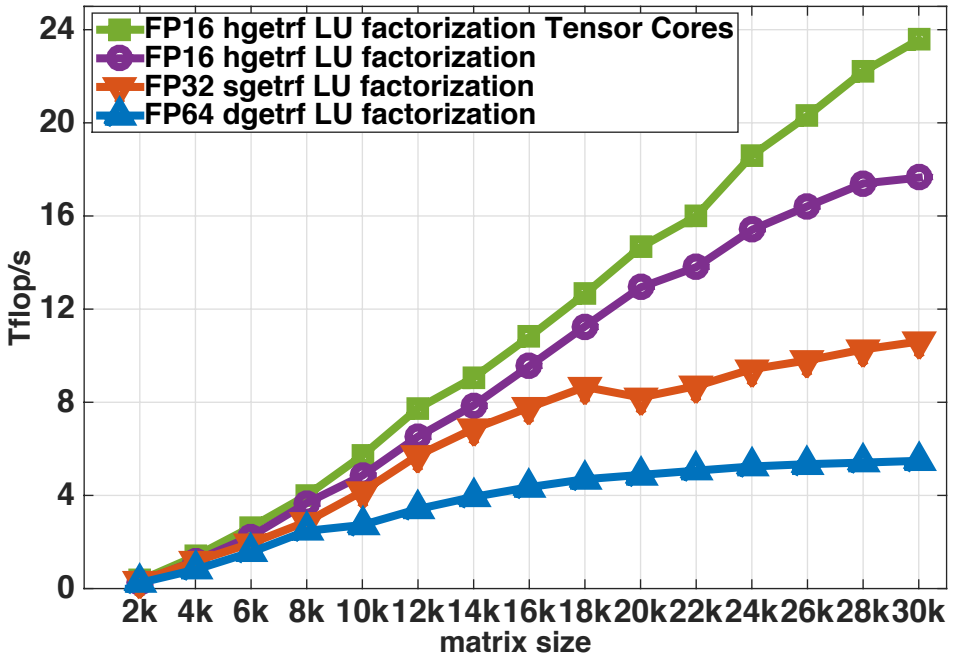
Study of the rank k update used by the LU factorization algorithm on Nvidia V100



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

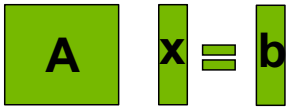
Leveraging Half Precision in HPC on V100

Study of the LU factorization algorithm on Nvidia V100



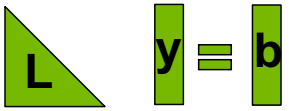
- LU factorization is used to solve a linear system Ax=b

A x = b



LUx = b

Ly = b



then

Ux = y

Leveraging half precision for HPC

Mixed Precision Methods

- Mixed precision, use the lowest precision required to achieve a given accuracy outcome
  - Improves runtime, reduce power consumption, lower data movement
  - Reformulate to find correction to solution, rather than solution; Δx rather than x.

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

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

Leveraging Half Precision in HPC on V100

Use Mixed Precision algorithms

- Achieve higher performance → faster time to solution
- Reduce power consumption 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.

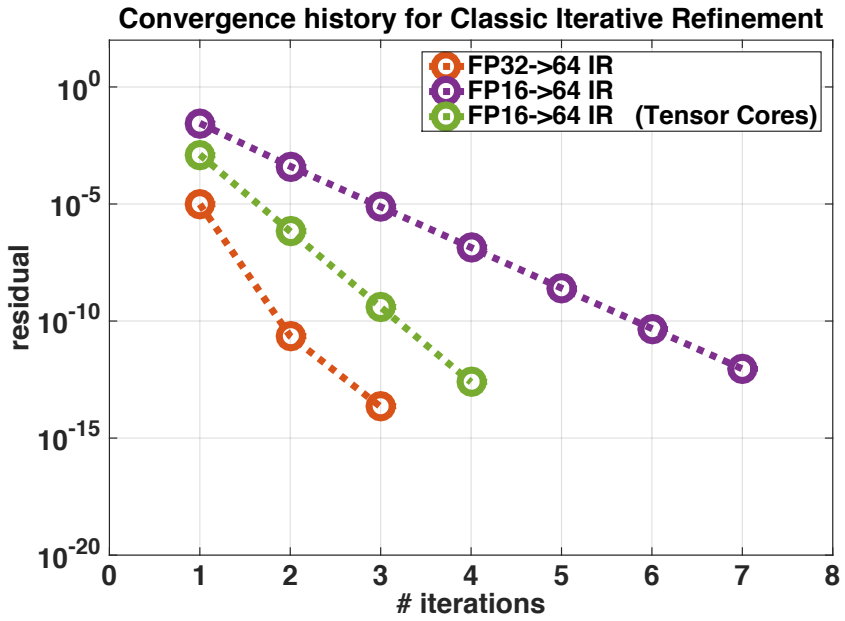
Leveraging Half Precision in HPC on V100

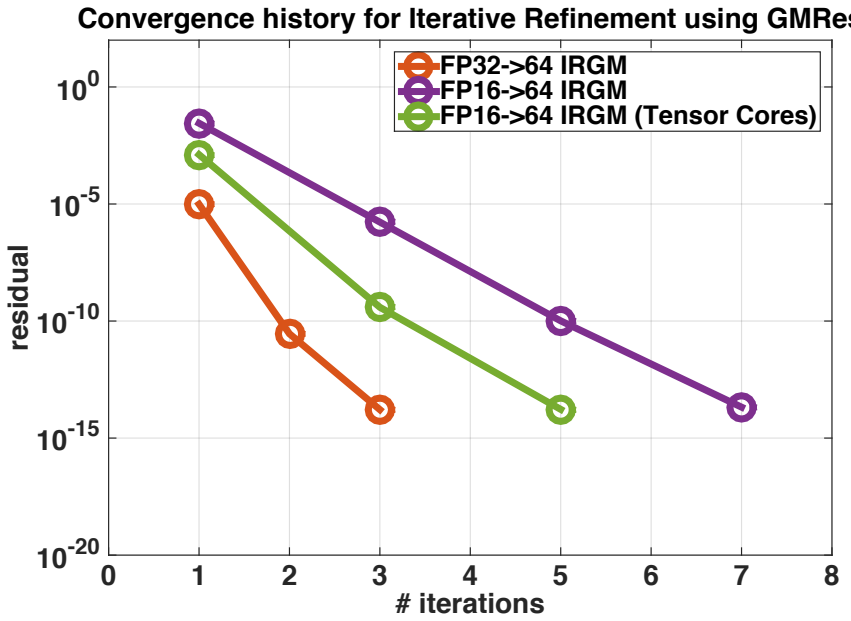
Idea: use low precision to compute the expensive flops (LU $O(n^3)$) and then iteratively refine the solution in order to achieve the FP64 arithmetic

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			

- 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.

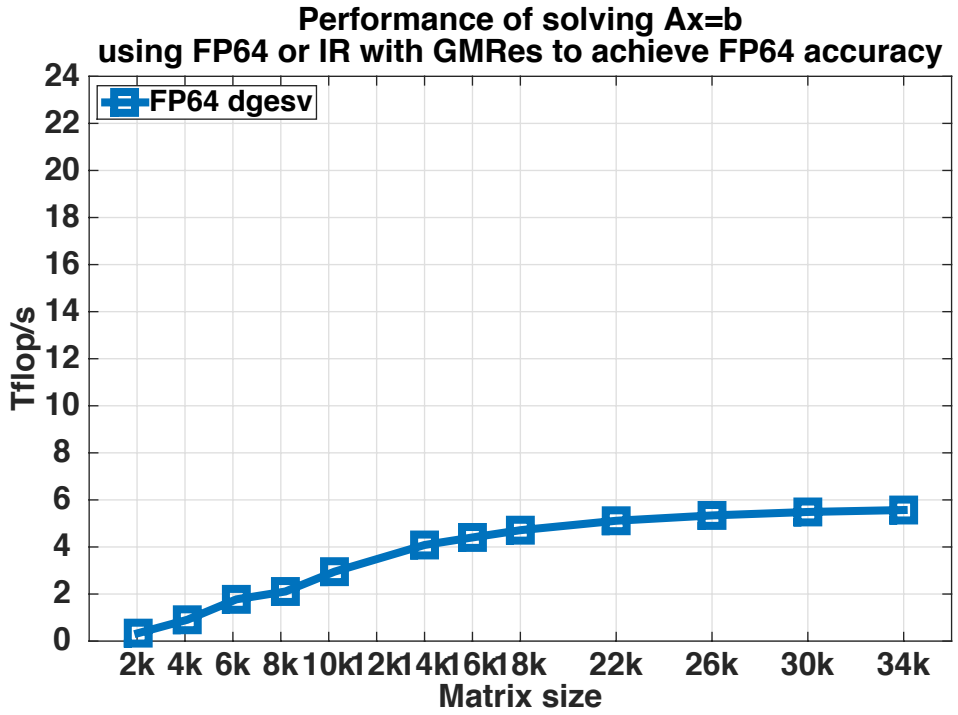
Leveraging Half Precision in HPC on V100





Matrix of size 10240 generated with positive λ and arithmetic distribution of its singular values $\sigma_i = 1 - (\frac{i-1}{n-1})(1 - \frac{1}{cond})$ and where its condition number is equal to 10^2.

Leveraging Half Precision in HPC on V100

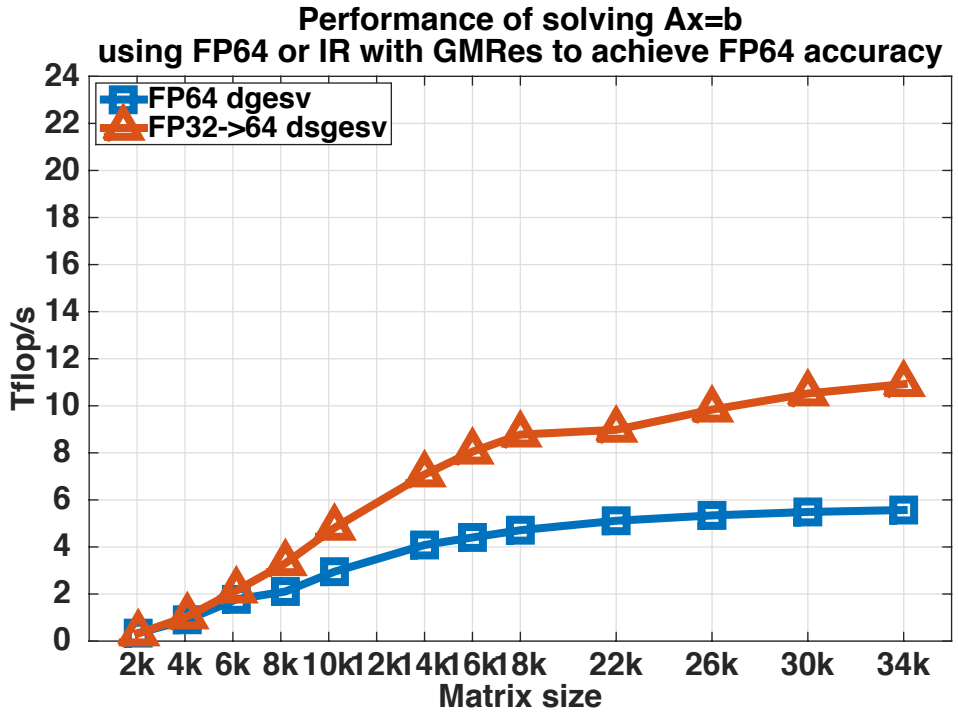


Flops = $2n^3/(3 \text{ time})$
meaning twice higher is twice faster

- solving Ax = b using **FP64 LU**

Matrices generated with positive λ and arithmetic distribution of its singular values $\sigma_i = 1 - (\frac{i-1}{n-1})(1 - \frac{1}{cond})$ and where its condition number is equal to 10^2.

Leveraging Half Precision in HPC on V100

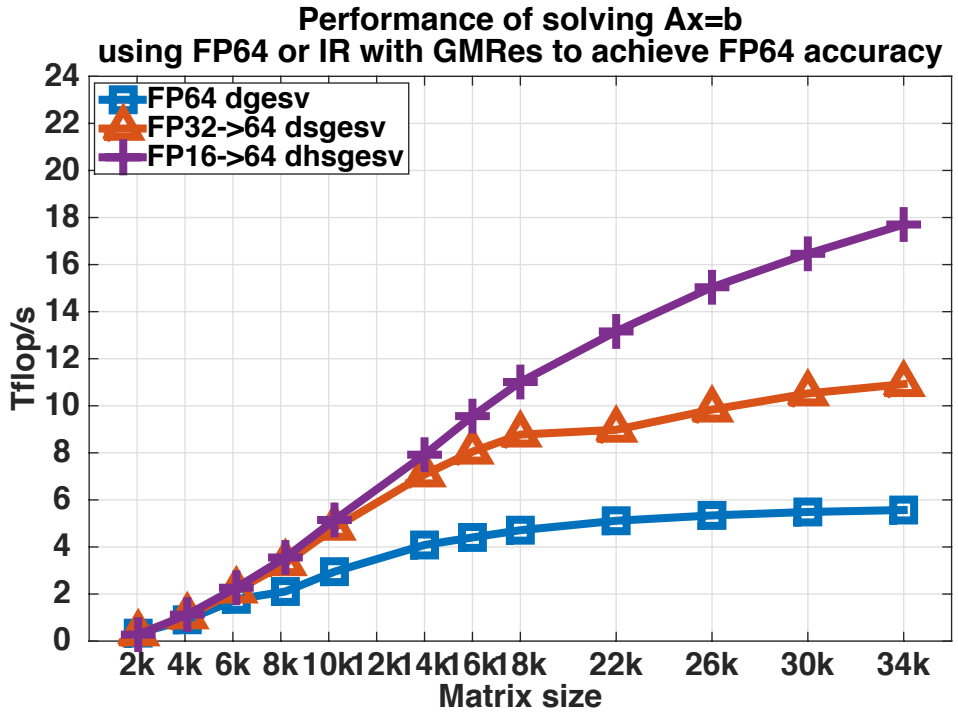


Flops = $2n^3/(3 \text{ time})$
meaning twice higher is twice faster

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

Matrices generated with positive λ and arithmetic distribution of its singular values $\sigma_i = 1 - (\frac{i-1}{n-1})(1 - \frac{1}{cond})$ and where its condition number is equal to 10^2.

Leveraging Half Precision in HPC on V100

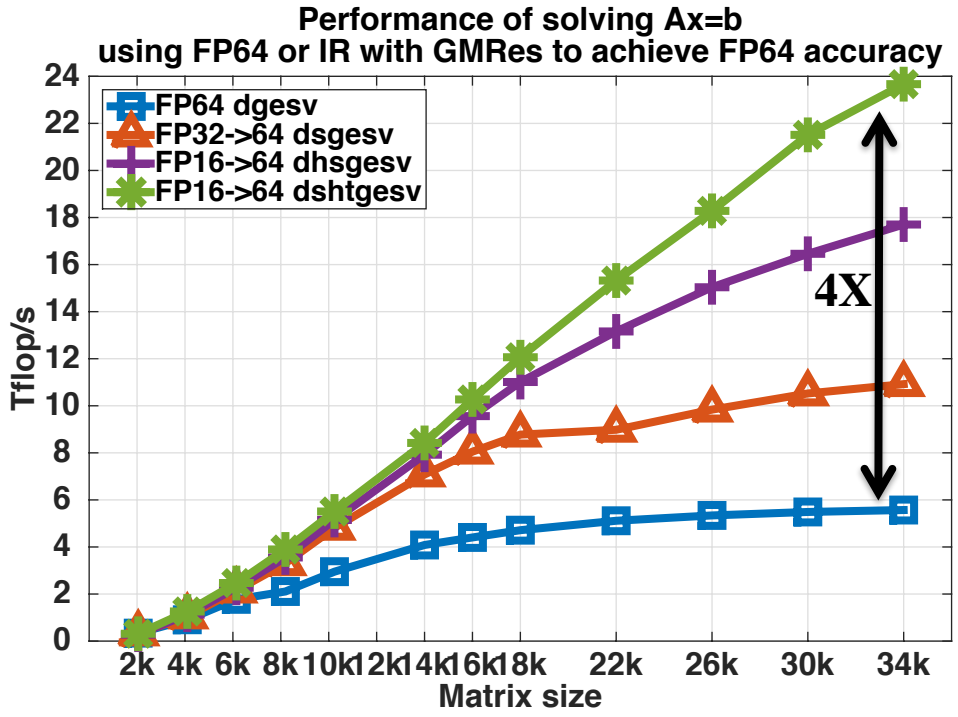


Flops = $2n^3/(3 \text{ time})$
meaning twice higher is twice faster

- 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

Matrices generated with positive λ and arithmetic distribution of its singular values $\sigma_i = 1 - (\frac{i-1}{n-1})(1 - \frac{1}{cond})$ and where its condition number is equal to 10^2.

Leveraging Half Precision in HPC on V100

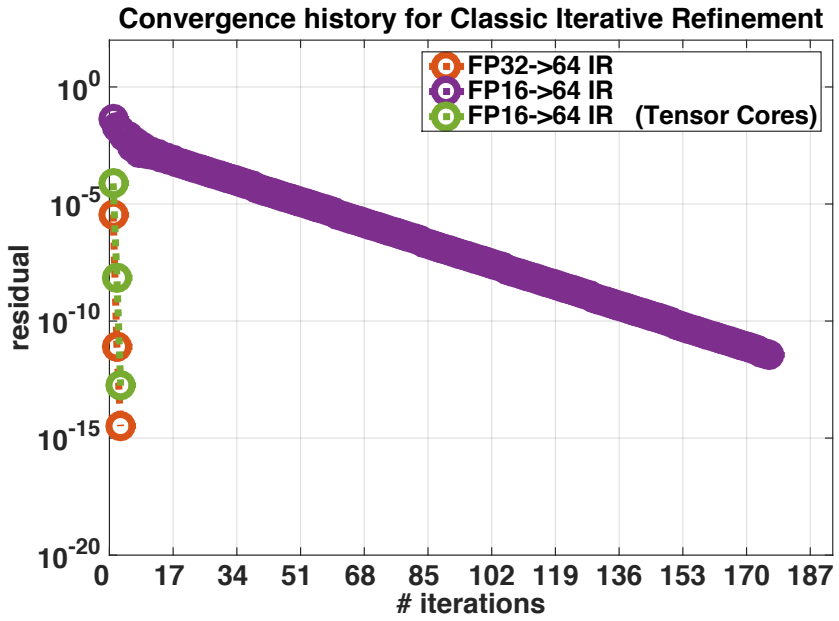


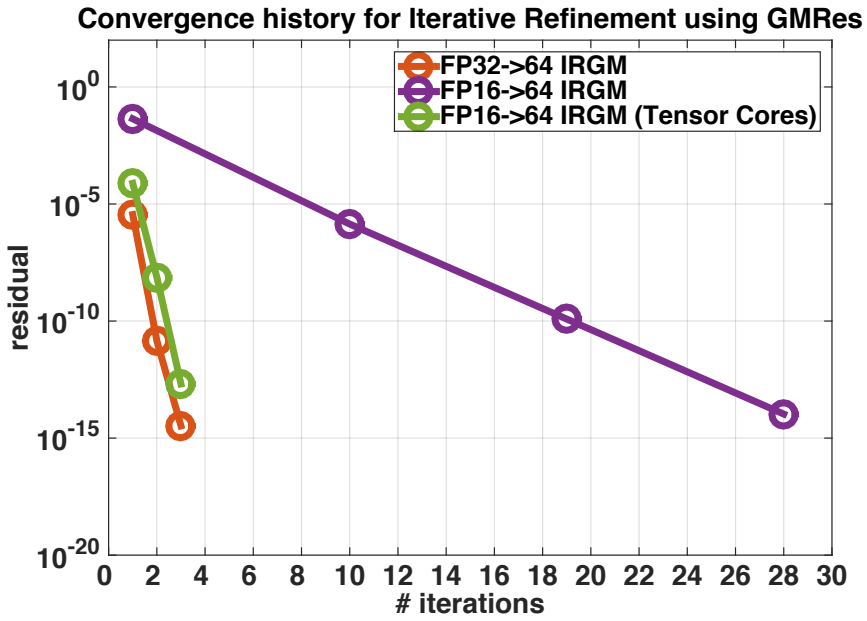
Flops = $2n^3/(3 \text{ time})$
meaning twice higher is twice faster

- 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

Matrices generated with positive λ and arithmetic distribution of its singular values $\sigma_i = 1 - (\frac{i-1}{n-1})(1 - \frac{1}{cond})$ and where its condition number is equal to 10^2.

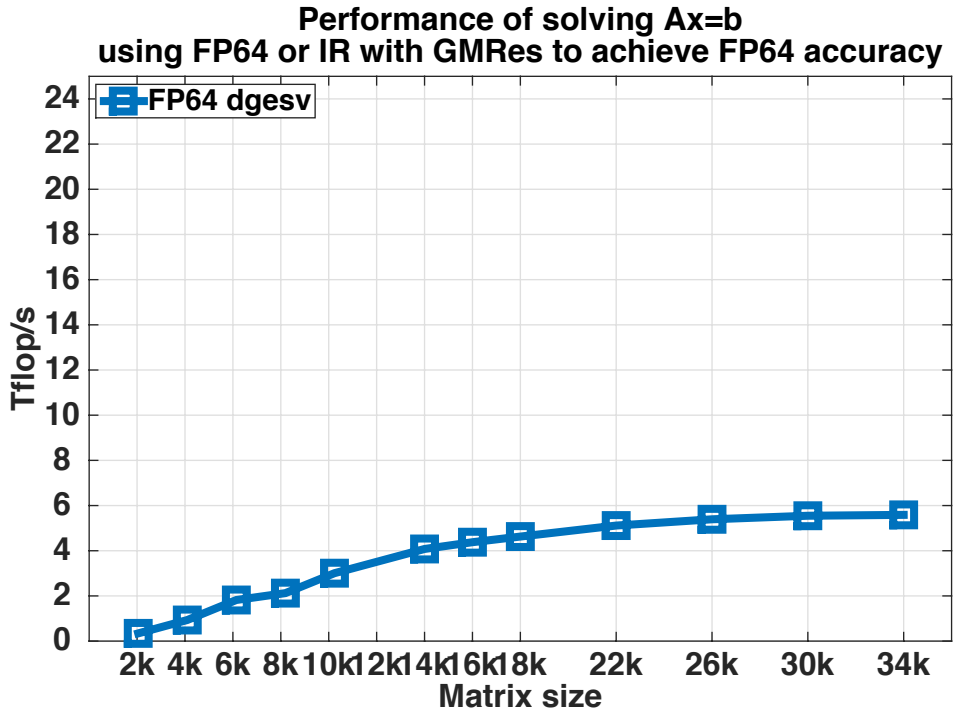
Leveraging Half Precision in HPC on V100





Matrix of size 10240 generated with positive λ and clustered singular values, σ_i=(1, $\cdots$, 1, $\frac{1}{cond}$) and where its condition number is equal to 10^2.

Leveraging Half Precision in HPC on V100

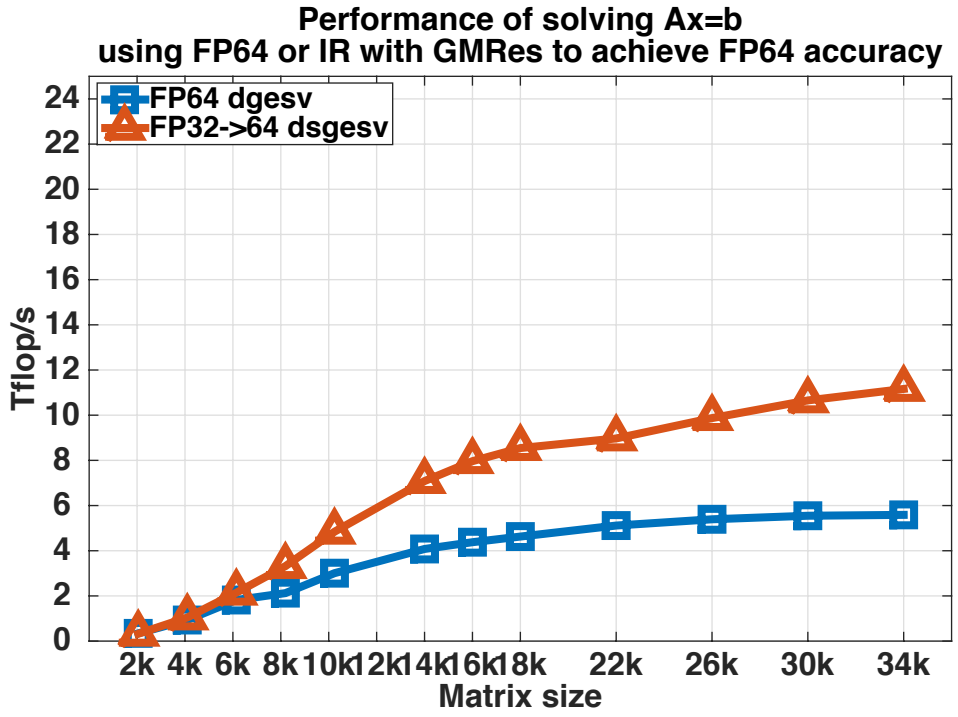


Flops = $2n^3/(3 \text{ time})$
meaning twice higher is twice faster

- solving Ax = b using **FP64 LU**

Matrices generated with positive λ and clustered distribution of its singular values $\sigma_i=(1, \cdots, 1, \frac{1}{cond})$ and where its condition number is equal to 10^2.

Leveraging Half Precision in HPC on V100

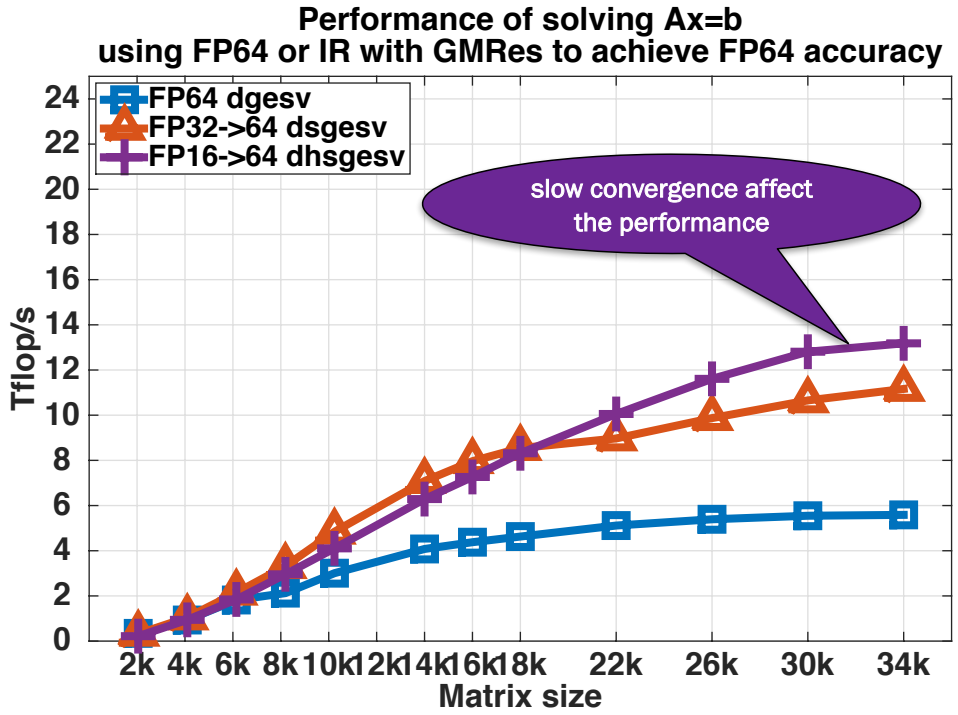


Flops = $2n^3/(3 \text{ time})$
meaning twice higher is twice faster

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

Matrices generated with positive λ and clustered distribution of its singular values $\sigma_i=(1, \cdots, 1, \frac{1}{cond})$ and where its condition number is equal to 10^2.

Leveraging Half Precision in HPC on V100

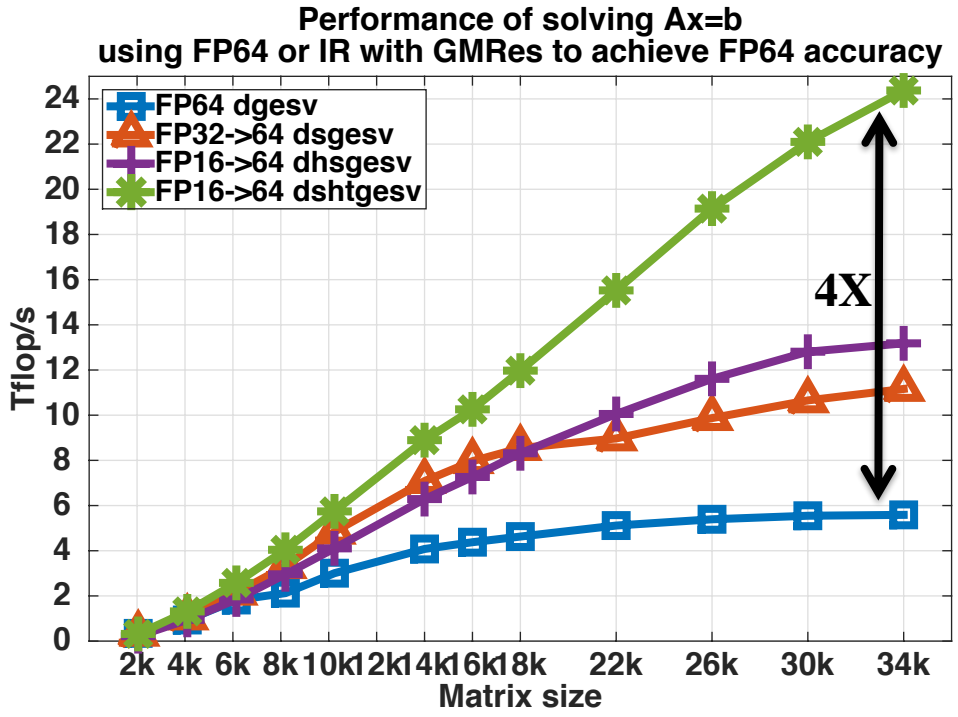


Flops = $2n^3/(3 \text{ time})$
meaning twice higher is twice faster

- 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

Matrices generated with positive λ and clustered distribution of its singular values $\sigma_i = (1, \cdots, 1, \frac{1}{cond})$ and where its condition number is equal to 10^2.

Leveraging Half Precision in HPC on V100



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

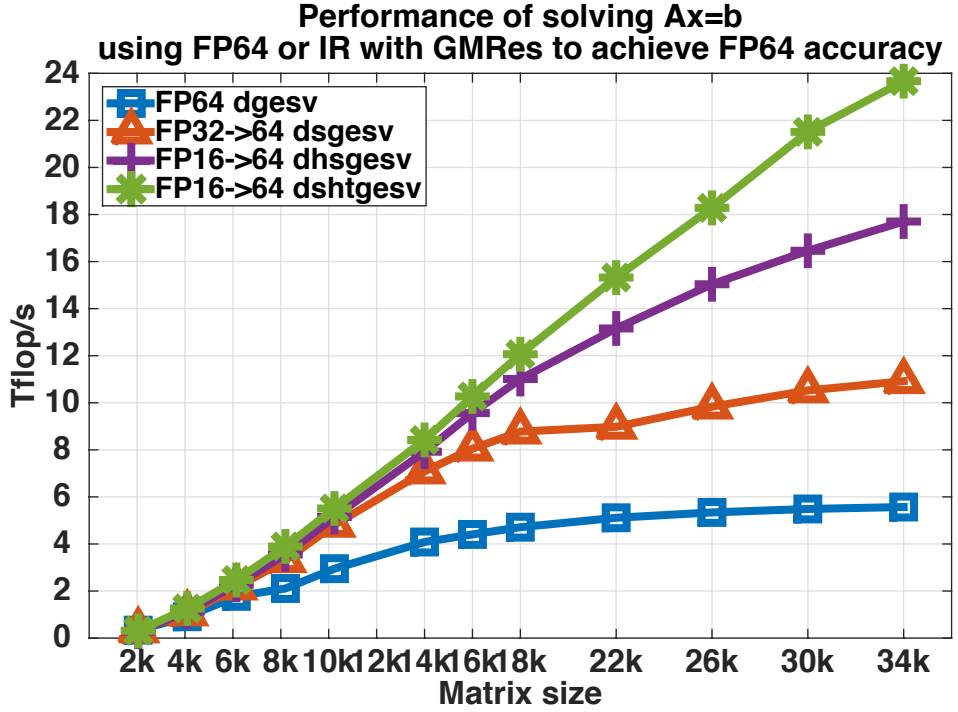
Matrices generated with positive λ and clustered distribution of its singular values $\sigma_i = (1, \cdots, 1, \frac{1}{cond})$ and where its condition number is equal to 10^2.

Leveraging Half Precision in HPC on V100

Use Mixed Precision algorithms

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

Leveraging Half Precision in HPC
Power awareness



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

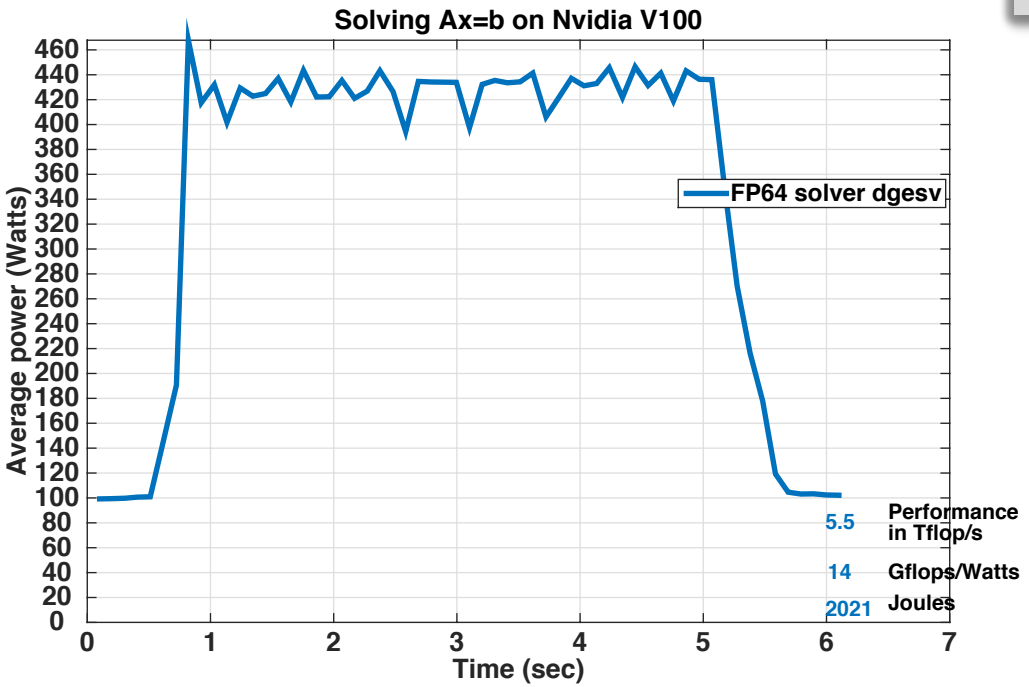
- Let's go back to this performance graph of solving Ax = b using the four different algorithms that achieve FP64 solution

Matrices generated with positive λ and arithmetic distribution of its singular values $\sigma_i = 1 - (\frac{i-1}{n-1})(1 - \frac{1}{cond})$ and where its condition number is equal to 10^2.

Leveraging Half Precision in HPC
Power awareness

CPU Intel Xeon E5-2650 v3 (Haswell) 2x10 cores @ 2.30 GHz

V100 NVIDIA Volta GPU 80 MP x 64 @ 1.38 GHz

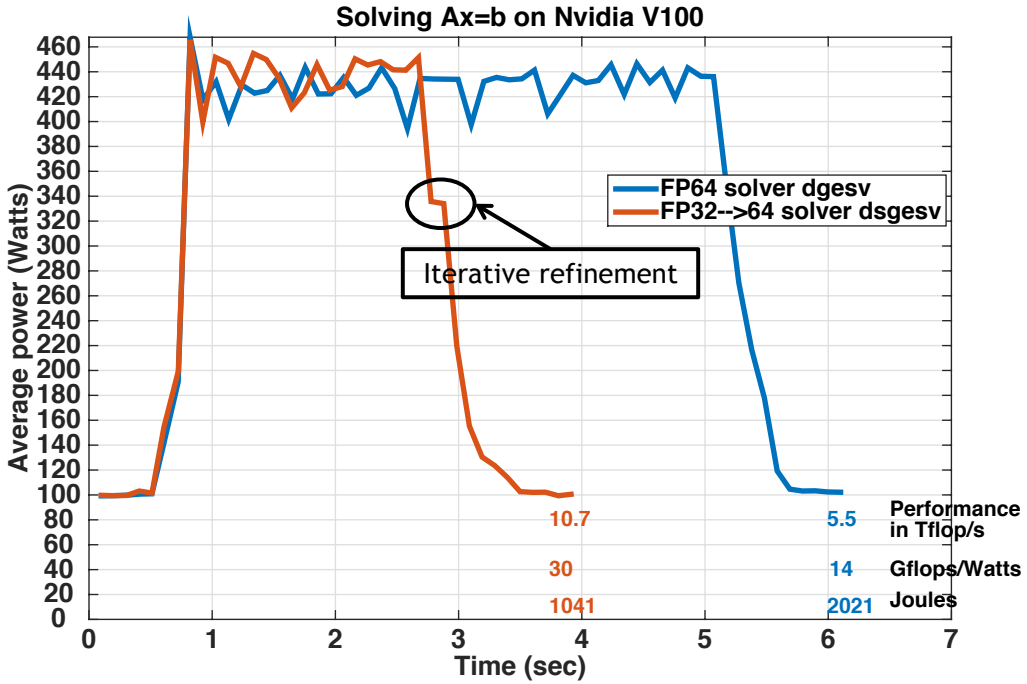


- Power consumption of the FP64 algorithm to solve Ax=b for a matrix of size 34K, it achieve 5.5 Tflop/s and requires about 2021 joules providing about 14 Gflops/Watts.

Power is for GPU + CPU + DRAM

Matrices generated with positive λ and arithmetic distribution of its singular values $\sigma_i = 1 - (\frac{i-1}{n-1})(1 - \frac{1}{cond})$ and where its condition number is equal to 10^2.

Leveraging Half Precision in HPC
Power awareness

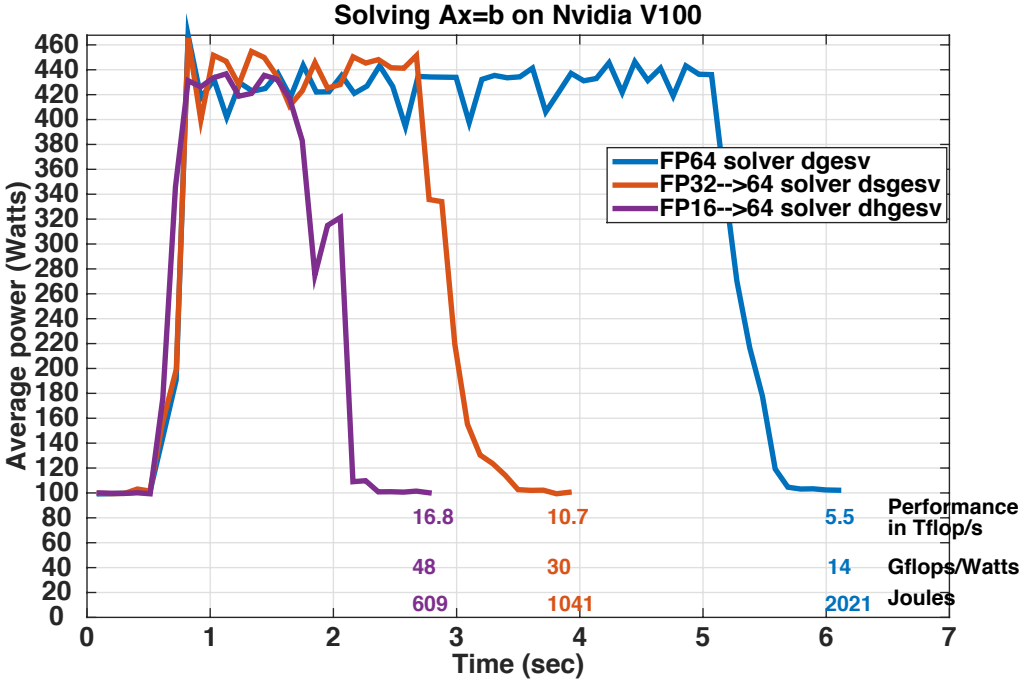


Mixed precision techniques can provide a large gain in energy efficiency

- Power consumption of the FP64 algorithm to solve Ax=b for a matrix of size 34K, it achieve 5.5 Tflop/s and requires about 2021 joules providing about 14 Gflops/Watts.
- Power consumption of the mixed precision FP32→64 algorithm to solve Ax=b for a matrix of size 34K, it achieve 10.7 Tflop/s and requires about 1041 joules providing about 30 Gflops/Watts.

Matrices generated with positive λ and arithmetic distribution of its singular values $\sigma_i = 1 - (\frac{i-1}{n-1})(1 - \frac{1}{cond})$ and where its condition number is equal to 10^2.

Leveraging Half Precision in HPC
Power awareness

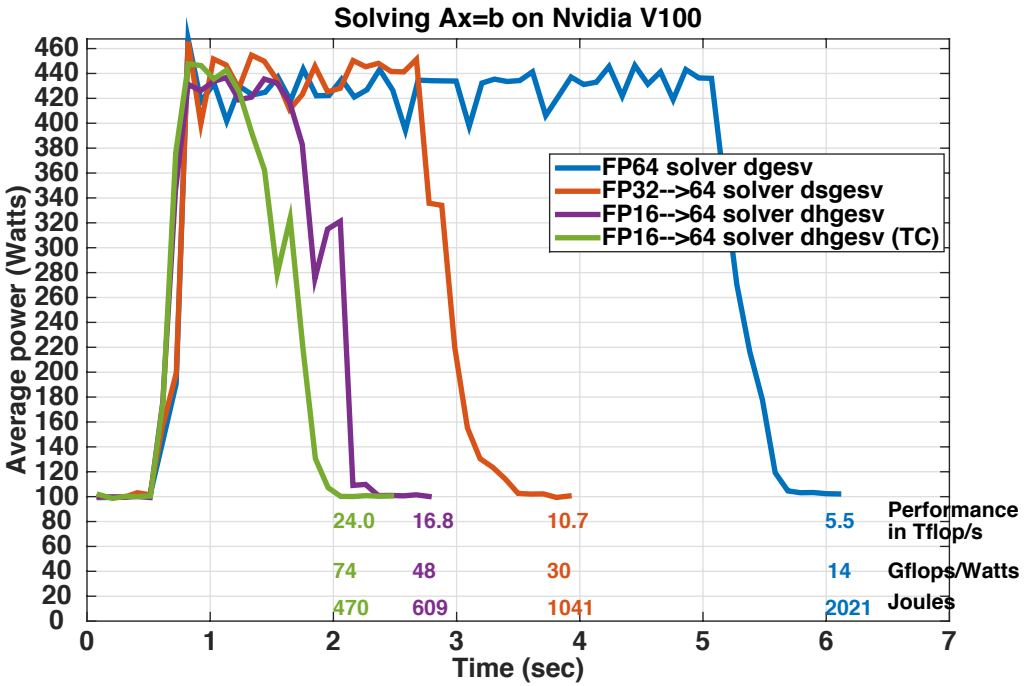


Mixed precision techniques can provide a large gain in energy efficiency

- Power consumption of the FP64 algorithm to solve Ax=b for a matrix of size 34K, it achieve 5.5 Tflop/s and requires about 2021 joules providing about 14 Gflops/Watts.
- Power consumption of the mixed precision FP32→64 algorithm to solve Ax=b for a matrix of size 34K, it achieve 10.7 Tflop/s and requires about 1041 joules providing about 30 Gflops/Watts.
- Power consumption of the mixed precision FP16→64 algorithm to solve Ax=b for a matrix of size 34K, it achieve 16.8 Tflop/s and requires about 609 joules providing about 48 Gflops/Watts.

Matrices generated with positive λ and arithmetic distribution of its singular values $\sigma_i = 1 - (\frac{i-1}{n-1})(1 - \frac{1}{cond})$ and where its condition number is equal to 10^2.

Leveraging Half Precision in HPC
Power awareness



Matrices generated with positive λ and arithmetic distribution of $\sigma_i = 1 - (\frac{i-1}{n-1})(1 - \frac{1}{cond})$ and where its condition number is equal

Mixed precision techniques can provide a large gain in energy efficiency

- Power consumption of the FP64 algorithm to solve Ax=b for a matrix of size 34K, it achieve 5.5 Tflop/s and requires about 2021 joules providing about 14 Gflops/Watts.
- Power consumption of the mixed precision FP32→64 algorithm to solve Ax=b for a matrix of size 34K, it achieve 10.7 Tflop/s and requires about 1041 joules providing about 30 Gflops/Watts.
- Power consumption of the mixed precision FP16→64 algorithm to solve Ax=b for a matrix of size 34K, it achieve 16.8 Tflop/s and requires about 609 joules providing about 48 Gflops/Watts.
- Power consumption of the mixed precision FP16→64 TC algorithm using Tensor Cores to solve Ax=b for a matrix of size 34K, it achieve 24 Tflop/s and requires about 470 joules providing about 74 Gflops/Watts.

Critical Issues and Challenges at Peta & Exascale for Algorithm and Software Design

- Synchronization-reducing algorithms
  - Break Fork-Join model
- Communication-reducing algorithms
  - Use methods which have lower bound on communication
- Mixed precision methods – from FP16 to FP64
  - Leverage speed of lower precision but retain the accuracy of higher precision
- Autotuning
  - Today's machines are too complicated, build "smarts" into software to adapt to the hardware
- Fault resilient algorithms
  - Implement algorithms that can recover from failures/bit flips
- Reproducibility of results
  - Today we can't guarantee this, without a penalty. We understand the issues, but some of our "colleagues" have a hard time with this.

Collaborators and Support

MAGMA team
http://icl.cs.utk.edu/magma

PLASMA team
http://icl.cs.utk.edu/plasma

Collaborating partners

University of Tennessee, Knoxville
University of California, Berkeley
University of Colorado, Denver
University of Manchester

U.S. DEPARTMENT OF
ENERGY

Umeå University

INRIA

Rutherford Appleton Laboratory

University of Manchester