Accélération de calculs de simulations des matériaux sur GPU

Francois Courteille

Senior Solution Architect, Accelerated Computing

AGENDA

Tesla, the NVIDIA compute platform

Quantum Chemistry Applications Overview

Molecular Dynamics Applications Overview

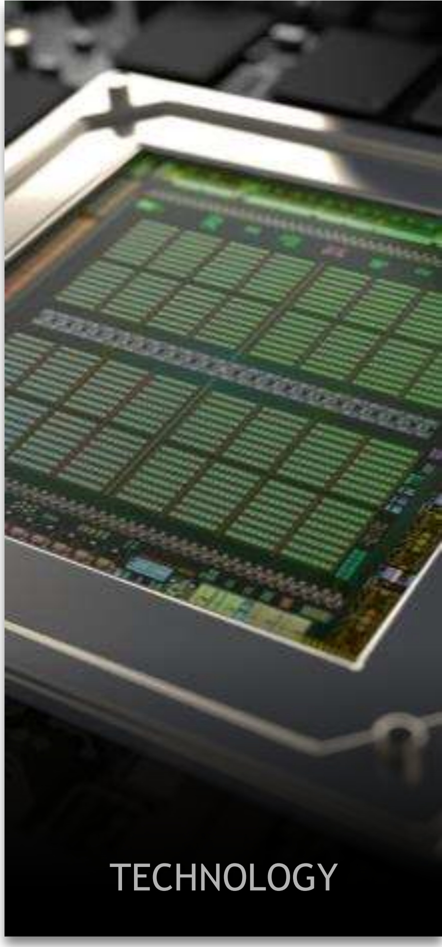
Material Science Applications at the start of the Pascal Era

GAMING

AUTO

ENTERPRISE

HPC & CLOUD



TECHNOLOGY

THE WORLD LEADER IN VISUAL COMPUTING

OUR TEN YEARS IN HPC

CUDA Launched

NVIDIA CUDA

World's First GPU Top500 System

Fermi: World's First HPC GPU

Oak Ridge Deploys World's Fastest Supercomputer w/ GPUs

Discovered How H1N1 Mutates to Resist Drugs

AlexNet beats expert code by huge margin using GPUs

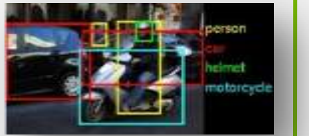
IMAGENET

World's First Atomic Model of HIV Capsid

Stanford Builds AI Machine using GPUs

World's First 3-D Mapping of Human Genome

Google Outperform Humans in ImageNet

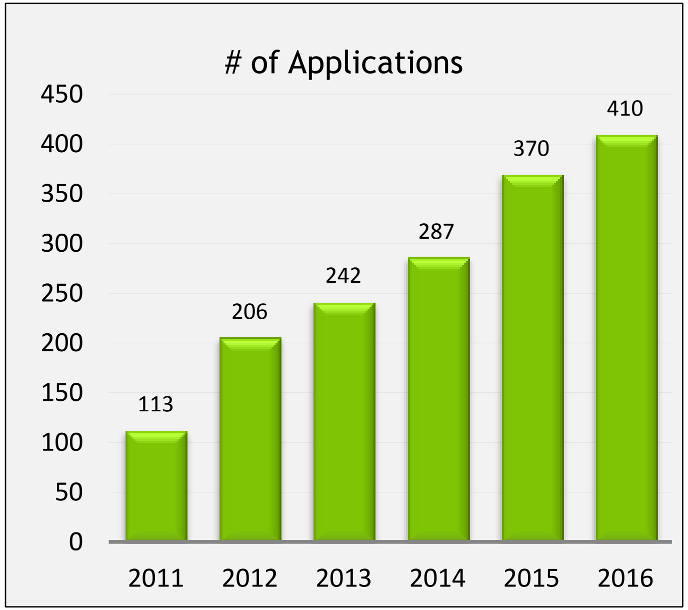


GPU-Trained AI Machine Beats World Champion in Go

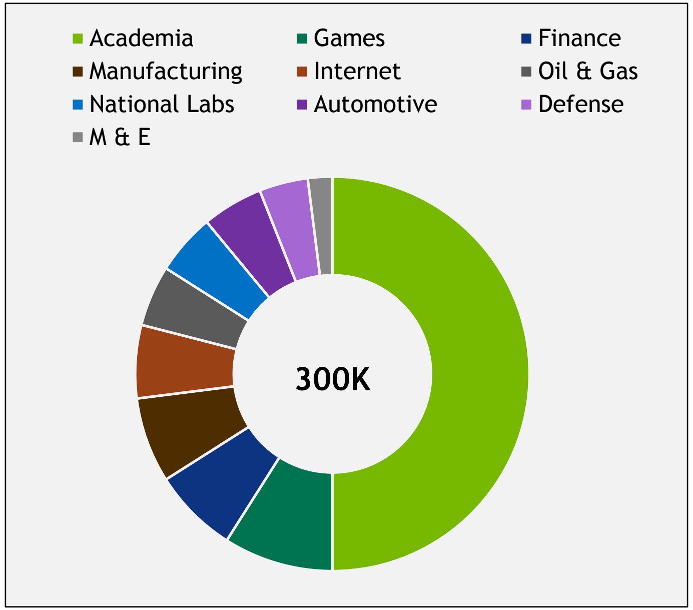
nature
At last — a computer program that can beat a champion Go player
ALL SYSTEMS GO

2006 2008 2010 2012 2014 2016

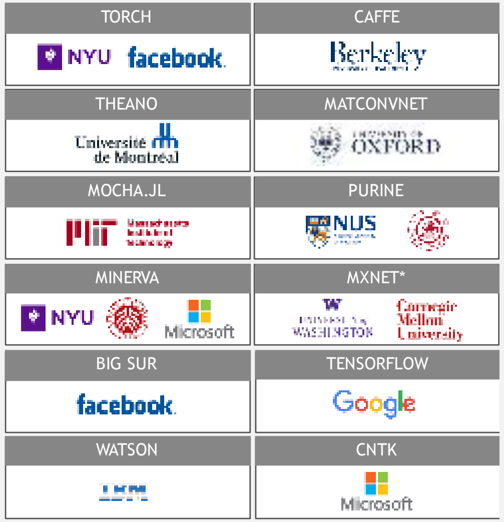
CREDENTIALS BUILT OVER TIME



Majority of HPC Applications are GPU-Accelerated, 410 and Growing



300K CUDA Developers, 4x Growth in 4 years



TORCH	CAFFE
NYU facebook	Berkeley
THEANO	MATCONVNET
Université de Montréal	OXFORD
MOCHA.JL	PURINE
MIT	NUS
MINERVA	MXNET*
NYU Microsoft	W UNIVERSITY of WASHINGTON Carnegie Mellon University
BIG SUR	TENSORFLOW
facebook	Google
WATSON	CNTK
IBM	Microsoft

100% of Deep Learning Frameworks are Accelerated

www.nvidia.com/appscatalo

END-TO-END TESLA PRODUCT FAMILY

HYPERSCALE HPC

Tesla M4, M40

Hyperscale deployment for DL training, inference, video & image processing

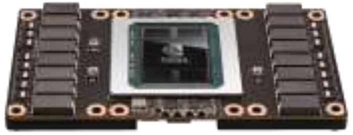
MIXED-APPS HPC

Tesla K80

HPC data centers running mix of CPU and GPU workloads

STRONG-SCALING HPC

Tesla P100



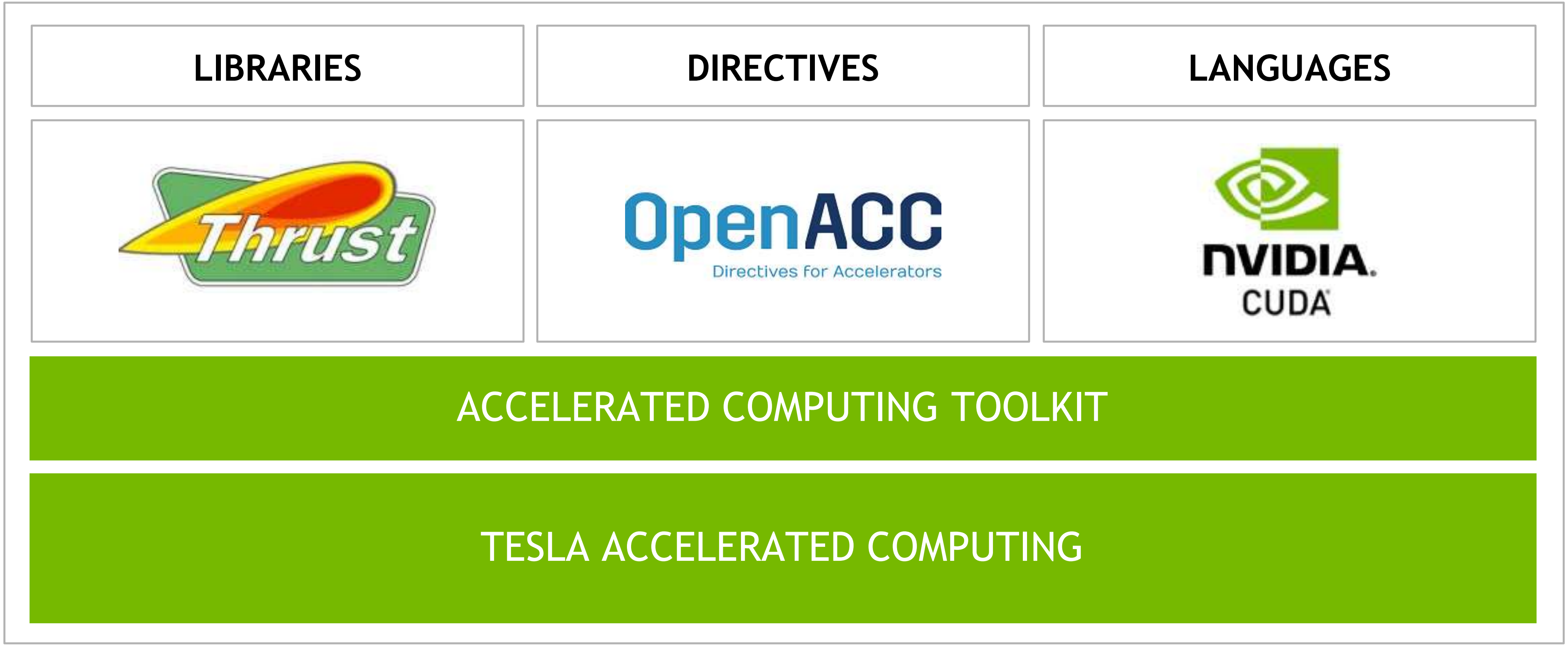
Hyperscale & HPC data centers running apps that scale to multiple GPUs

FULLY INTEGRATED DL SUPERCOMPUTER

DGX-1

For customers who need to get going now with fully integrated solution

TESLA FOR SIMULATION



LIBRARIES

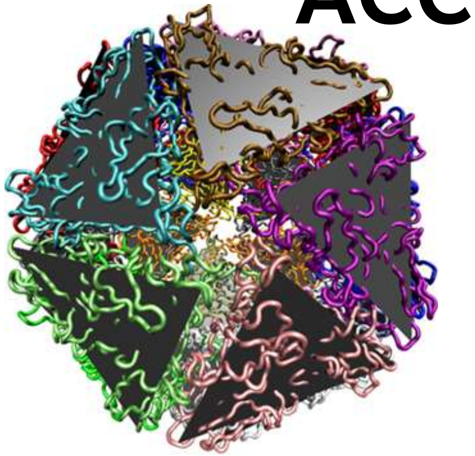
DIRECTIVES

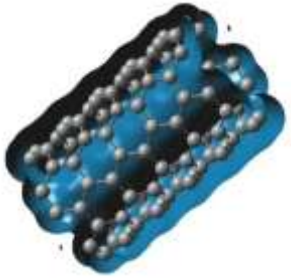
LANGUAGES

ACCELERATED COMPUTING TOOLKIT

TESLA ACCELERATED COMPUTING

OVERVIEW OF LIFE & MATERIAL ACCELERATED APPS





MD: All key codes are GPU-accelerated

- Great multi-GPU performance
- Focus on dense (up to 16) GPU nodes &/or large # of GPU nodes
- ACEMD*, AMBER (PMEMD)*, BAND, CHARMM, DESMOND, ESPResso, Folding@Home, GPUgrid.net, GROMACS, HALMD, HOOMD-Blue*, LAMMPS, Lattice Microbes*, mdcore, MELD, miniMD, NAMD, OpenMM, PolyFTS, SOP-GPU* & more

QC: All key codes are ported or optimizing

- Focus on using GPU-accelerated math libraries, OpenACC directives
- GPU-accelerated and available today:
  - ABINIT, ACES III, ADF, BigDFT, CP2K, GAMESS, GAMESS-UK, GPAW, LATTE, LSDalton, LSMS, MOLCAS, MOPAC2012, NWChem, OCTOPUS*, PEtot, QUICK, Q-Chem, QMCPack, Quantum Espresso/PWscf, QUICK, TeraChem*
- Active GPU acceleration projects:
  - CASTEP, GAMESS, Gaussian, ONETEP, Quantum Supercharger Library*, VASP & more

green* = application where >90% of the workload is on GPU

MD VS. QC ON GPUS

“Classical” Molecular Dynamics	Quantum Chemistry (MO, PW, DFT, Semi-Emp)
Simulates positions of atoms over time; chemical-biological or chemical-material behaviors	Calculates electronic properties; ground state, excited states, spectral properties, making/breaking bonds, physical properties
Forces calculated from simple empirical formulas (bond rearrangement generally forbidden)	Forces derived from electron wave function (bond rearrangement OK, e.g., bond energies)
Up to millions of atoms	Up to a few thousand atoms
Solvent included without difficulty	Generally in a vacuum but if needed, solvent treated classically (QM/MM) or using implicit methods
Single precision dominated	Double precision is important
Uses cuBLAS, cuFFT, CUDA	Uses cuBLAS, cuFFT, OpenACC
Geforce (Accademics), Tesla (Servers)	Tesla recommended
ECC off	ECC on

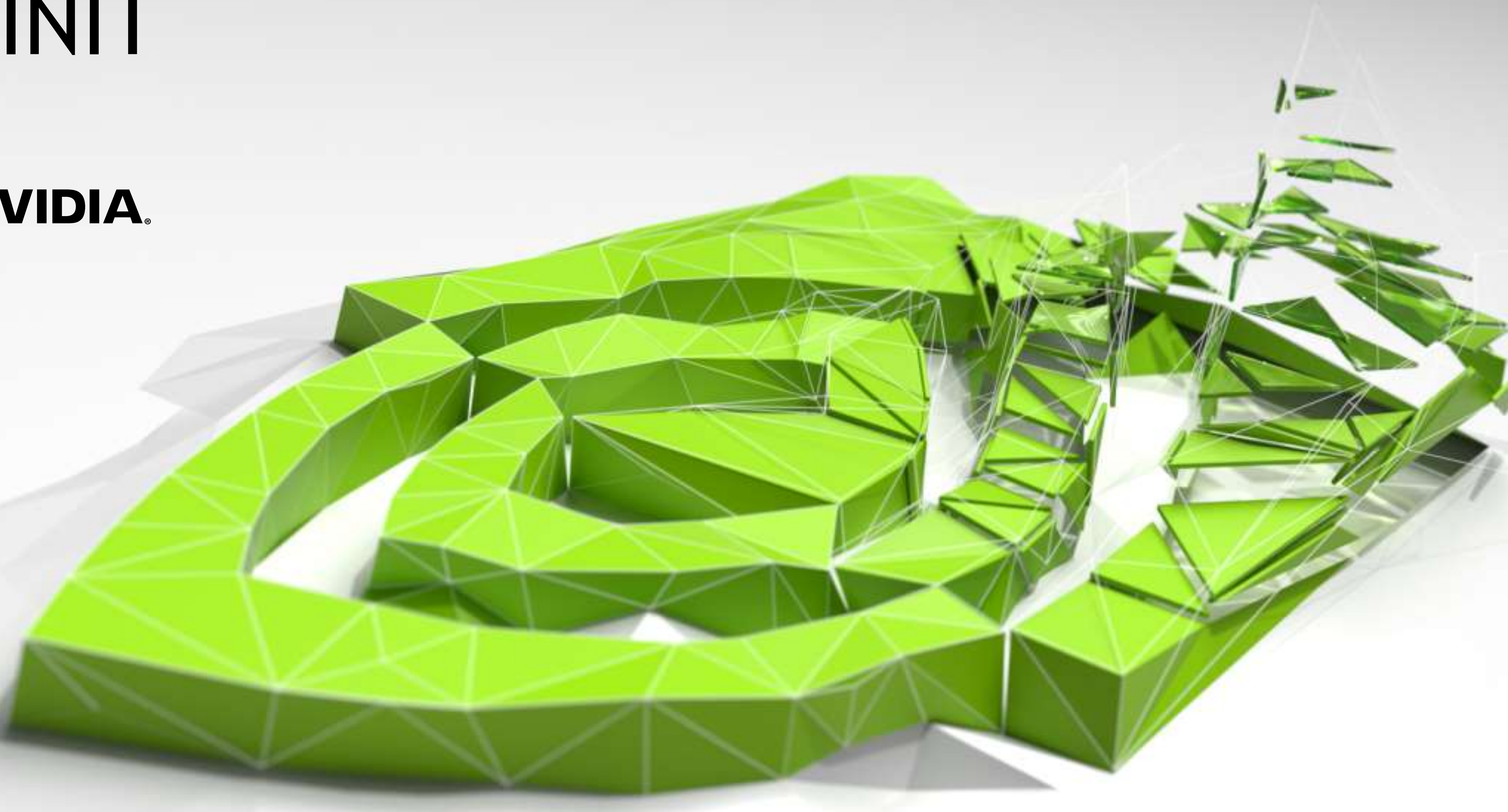
GPU-Accelerated Quantum Chemistry Apps

Green Lettering Indicates Performance Slides Included

- Abinit
- ACES III
- ADF
- BigDFT
- CP2K
- GAMESS-US
- Gaussian
- GPAW
- LATTE
- LSDalton
- MOLCAS
- Mopac2012
- NWChem
- Octopus
- ONETEP
- Petot
- Q-Chem
- QMCPACK
- Quantum Espresso
- Quantum SuperCharger Library
- TeraChem
- VASP
- WL-LSMS

GPU Perf compared against dual multi-core x86 CPU socket.

ABINIT



6th International ABINIT Developer Workshop

April 15-18, 2013 - Dinard, France

USING ABINIT ON GRAPHICS PROCESSING UNITS (GPU)

Marc Torrent
CEA, DAM, DIF, F-91297 Arpajon, France

Florent Dahm
C-S – 22 av. Galilée. 92350 Le Plessis-Robinson

With a contribution from Yann Pouillon for the build system

www.cea.fr

ABINIT ON GPU

OUTLINE

Introduction

- Introduction to GPU computing
  Architecture, programming model
- Density-Functional Theory with plane waves
  How to port it on GPU

ABINIT on GPU

- Implementation
- Performances of ABINIT v7.0

How To…

- Compile
- Use

LOCAL OPERATOR: FAST FOURIER TRANSFORMS

- Our choice (for a first version): Replace the MPI plane wave level by a FFT computation on GPU

$$\langle \vec g | \tilde v_{eff} | \tilde\psi \rangle = \underbrace{\int d\vec r \, e^{-i\vec g\cdot\vec r}}_{FTT} \tilde v_{eff}(\vec r) \underbrace{\left(\sum_{g'} c_{g'} e^{i\vec g'\cdot\vec r}\right)}_{FFT}$$

- Included in **Cuda** package : **cuFFT** library

  Interfaced with c and Fortran

- A system dependent efficiency is expected
  - *Small FFT underuse the GPU and are dominated by the time required to transfer the data to/from the GPU*
  - *Within PAW, small plane wave basis are used*

LOCAL OPERATOR: FAST FOURIER TRANSFORMS

Implementation

- *Interface Cuda / Fortran*
- *New routines that encapsulate cufft calls*
- *Development of 3 small cuda kernels*
- *Optimization of memory copy to overlap computation and data transfer*

Performance highly dependent on FFT size

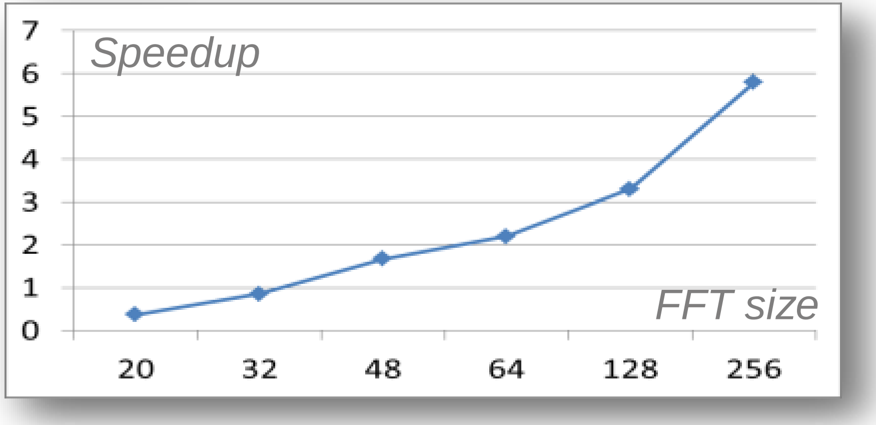
TEST CASE

107 gold atoms cell

TGCC-Curie

(Intel Westmere + Nvidia M2090)

fourwf – option 2



LOCAL OPERATOR: FAST FOURIER TRANSFORMS

Improvement : multiple band optimization

- The Cuda kernels are more intensively used

  Requires less transfers

- See *bandpp* variable

  When Locally Optimal Block Preconditioned Conjugate Gradient algorithm is used

107 gold atoms

bandpp	fourwf on GPU
1	112,94
2	84,55
4	70,55
8	61,91
18	54,79
72	50,55

31 copper atoms

bandpp	fourwf on GPU
1	36,02
2	20,79
4	14,14
10	9,16
20	9,50
100	7,94

TEST CASE

TGCC-Curie

(Intel Westmere + Nvidia M2090)

NON LOCAL OPERATOR

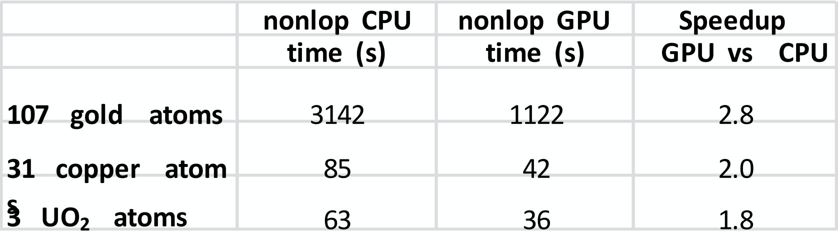
Implementation

- Development of 2 Cuda Kernels for $\langle \tilde{p}_i | \psi \rangle = \sum_{\vec{g}} \langle \tilde{p}_i | \vec{g}' \rangle \langle \vec{g}' | \psi \rangle$
- Development of 2 Cuda Kernels for $\langle \vec{g} | V_{NL} | \psi \rangle = \sum_{i,j} \langle \vec{g} | \tilde{p}_i \rangle D_{ij} \langle \tilde{p}_j | \psi \rangle$
- *Using collaborative memory zones as much as possible (threads linked to the same plane wave lie in the same collaborative zone)*

Used for :

- Non-local operator
- Overlap operator
- Forces and stress tensor
- PAW occupation matrix

Performances (NL operator only)



	nonlop CPU time (s)	nonlop GPU time (s)	Speedup GPU vs CPU
107 gold atoms	3142	1122	2.8
31 copper atoms	85	42	2.0
3 UO$_2$ atoms	63	36	1.8

TGCC-Curie
(Intel Westmere + Nvidia M2090)

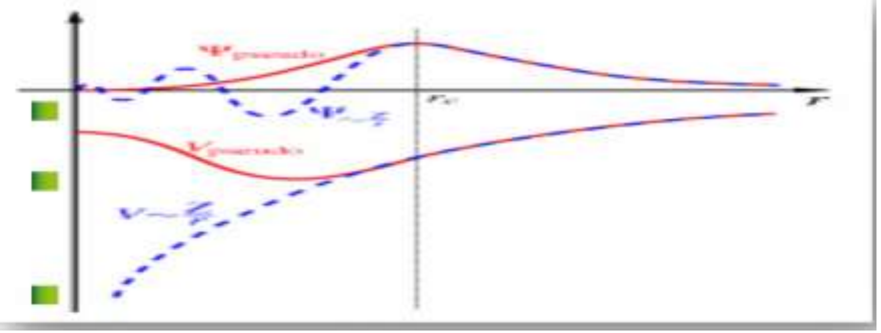
In addition to all MPI parallelization levels !

- Need a large number of PW
- Need a large number of NL projectors

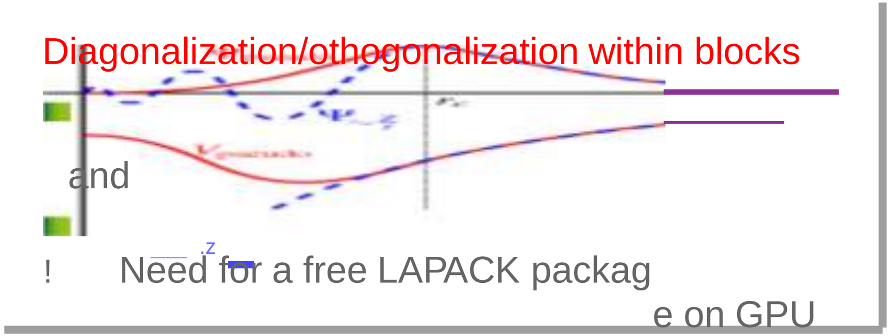
```
Algorithm 1 lobpcg
Require: $^0 = \{ ^0_1, \ldots, ^0_m \}$ close to the minimum and K a pre-
         conditioner; $P = \{P^{(0)}_1, \ldots, P^0_m\}$ is initialized to 0.
for i=0,1, , do
    $⌤^{(i)} = ⌤( ^{(i)})$
    $R^{(i)} = H ^{(i)} ⌤^{(i)} O ^{(i)}$
    $W^{(i)} = KR^{(i)}$
    The Rayleigh-Ritz method is applied within the subspace $⊼ =$
        $P^{(i)}_1, \ldots, P^{(i)}_m, ^{(i)}_1, \ldots, ^{(i)}_m, W^{(i)}_1, \ldots, W^{(i)}_m$
    $^{(i+1)} = ^{(i)} ^{(i)} + ⇜^{(i)} W^{(i)} + ^{(i)} P^{(i)}$
    $P^{(i+1)} = ⇜^{(i)} W^{(i)} + ^{(i)} P^{(i)}$
end for
```

Locally Optimal Block Preconditioned Conjugate Gradient

bandpp	Total time	Time Linear Algebra
1	674	119
2	612	116
10	1077	428
50	5816	5167



! Our choice : use of **cuBlas**



Diagonalization/othogonalization within blocks

and

! Need for a free LAPACK package on GPU

Our choice : use of **MAGMA**

MAGMA

Home

Overview

News

Downloads

People

Publications

Dcx:umentatb

Rartners

User Forum

Matrix Algebra on GPU and Multicore Architectures

The MAGMA project aims 1o oeveop a dense linear agebra library similar 1o LAPACK bu1 for heterogeneous!hybrd architectures, starting with current 'Multcore-tGPU' systems.

The MAGMA research is based on 1he dlea 1hat, 1o address the complex challenges of 1heemerging hybrd environments, optimal software solutbns will themselves have to hybrdize, combining the strengths of differen1 agornhms wnhin a single framework. Buitling on this idea, we aim to design linear algebra algorithms and frameworks for hybrd manycore and GPU systems that can enable applications to fully exploit the power that each of the hybrd components offers.

Latest MAGMA News

2013-03-12

MAGMA MIC 1.0 Beta for Intel Xeon Phi Coprocessors Re~ased

2012-11-14

MAGMA 1.3 Released

2012-11-13

MAGMA MIC 0.3 for Intel Xeon Phi Coprocessors Released

2012-10-24

clMAGMA 1.0 Released

2012-06-29

MAGMA 1.2.1 Released

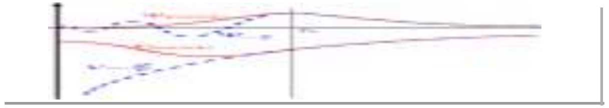
ICL{>l3r

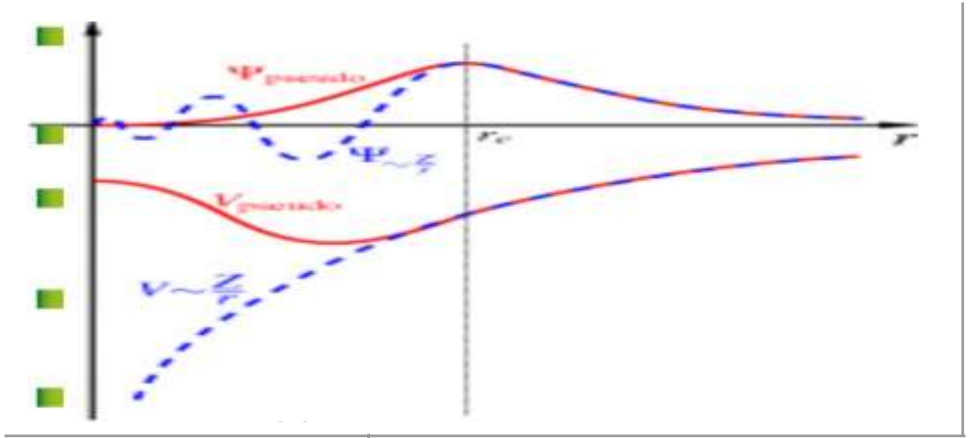
lver27 2013 Sponsored By: Industry Support From: AMD:I @ Mlaoeolt

License

Copyrqht 2013 Th.. Uni ,:.r.;it of T..-,nn.. - e , All rt}h re ..

Redistribution and u urce and binary forms, with or without modification, are permitted provided that the fol ing d. in conditbns are met:

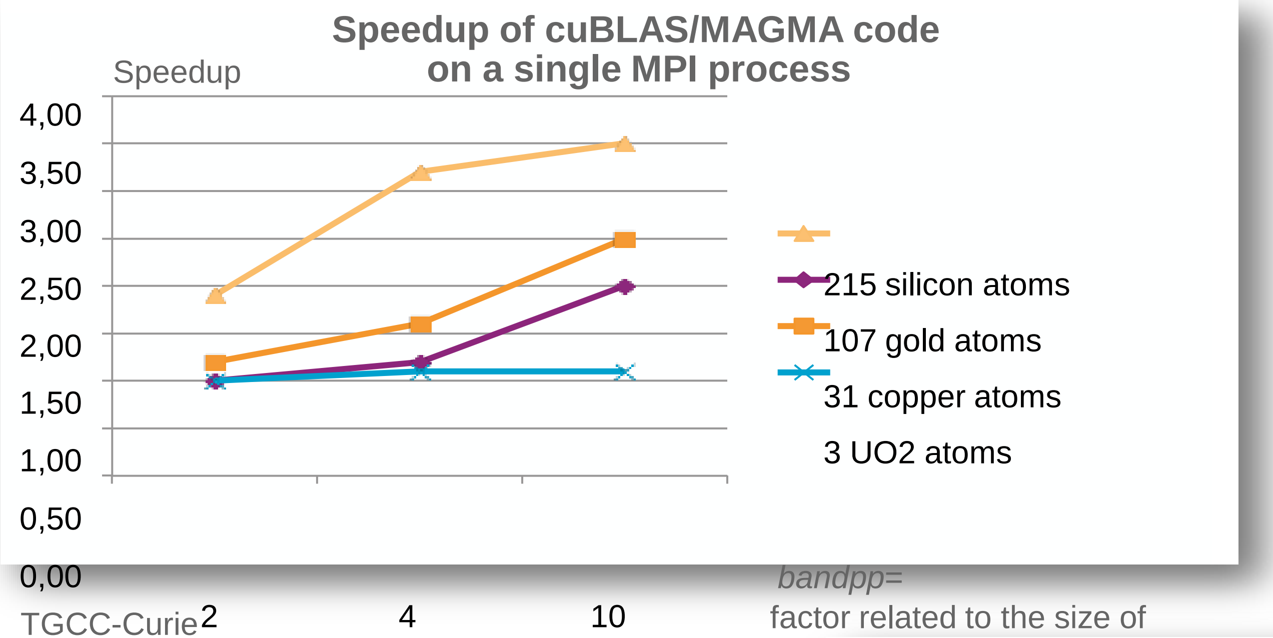




- Goal : address hybrid environments

ITERATIVE EIGENSOLVER

Performances



TGCC-Curie
(Intel Westmere + Nvidia M2090)

- Highly dependent on the physical system
- Depends only on the size of blocks used in the LOBPCG algorithm

OUTLINE

Introduction

Introduction to GPU computing
Architecture, programming model

Density-Functional Theory with plane waves
How to port it on GPU

ABINIT on GPU

Implementation

Performances of ABINIT v7.0

How To…

Compile

Use

PERFORMANCES ON A SINGLE GPU

Comparing architectures…

TGCC-Titane
Intel Nehalem+ NVidia S1070 (Tesla)

TGCC-Curie
Intel Westmere + NVidia M2090 (Fermi)

Time proc 0 (s)	Titane			Curie		
	CPU	**CPU+GPU**	**Speedup**	**CPU**	**CPU+GPU**	**Speedup**
107 gold atoms	4230	1857	2,3	5154	621	8.3
31 copper atoms	153	102	1,5	235	64	3,7
5 BaTiO3 atoms	85	98	0,9	86	73	1,2

- Strongly dependent on architecture
  Titane : fast CPU+ old GPU
- $BaTiO_3$ case is too small to take benefit from GPU

PERFORMANCES ON MULTIPLE GPU

TGCC-Curie

Intel Westmere + NVidia M2090 (Fermi)

8 CPU (MPI only) + 8 GPU

Time proc 0 (s)	**Curie**		
	CPU	**CPU+GPU**	**Speedup**
215 silicon atoms (1 iter.)	16160	3680	**4,4**
107 gold atoms	866	621	**4,7**
31 copper atoms	49	64	**1,5**

200 CPU (MPI only) + 200 GPU

Time proc 0 (s)	**Curie**		
	CPU	**CPU+GPU**	**Speedup**
215 silicon atoms (45 iter.)	25856	6309	**4,1**

- Less efficient when the number of MPI processes increases (the load decreases on the GPU)

PERFORMANCES VS OTHER CODES

- Other DFT codes have been ported on GPU.
  Speedup of plane wave codes are similar
- **Quantum espresso** (plane waves) : **x3/x4** (sequential), **x2/x3** (parallel) **[1,2]**
- **VASP** (plane waves) : **x3/x4** **[3,4]**
- **BigDFT** (wavelets) : **x5/x7** **[5]**
- **GPAW** (real space) : **x10/x15** **[1,6]**

[1] Harju et al, *App. Parallel and Sc. Comp. Lectures Notes in Computer Science* **7782**, p3 (2013)

[2] Spiga et al, pdp, 20th Euromicro International Conference on Parallel, Distributed and Network-based Processing, p368 (2012)

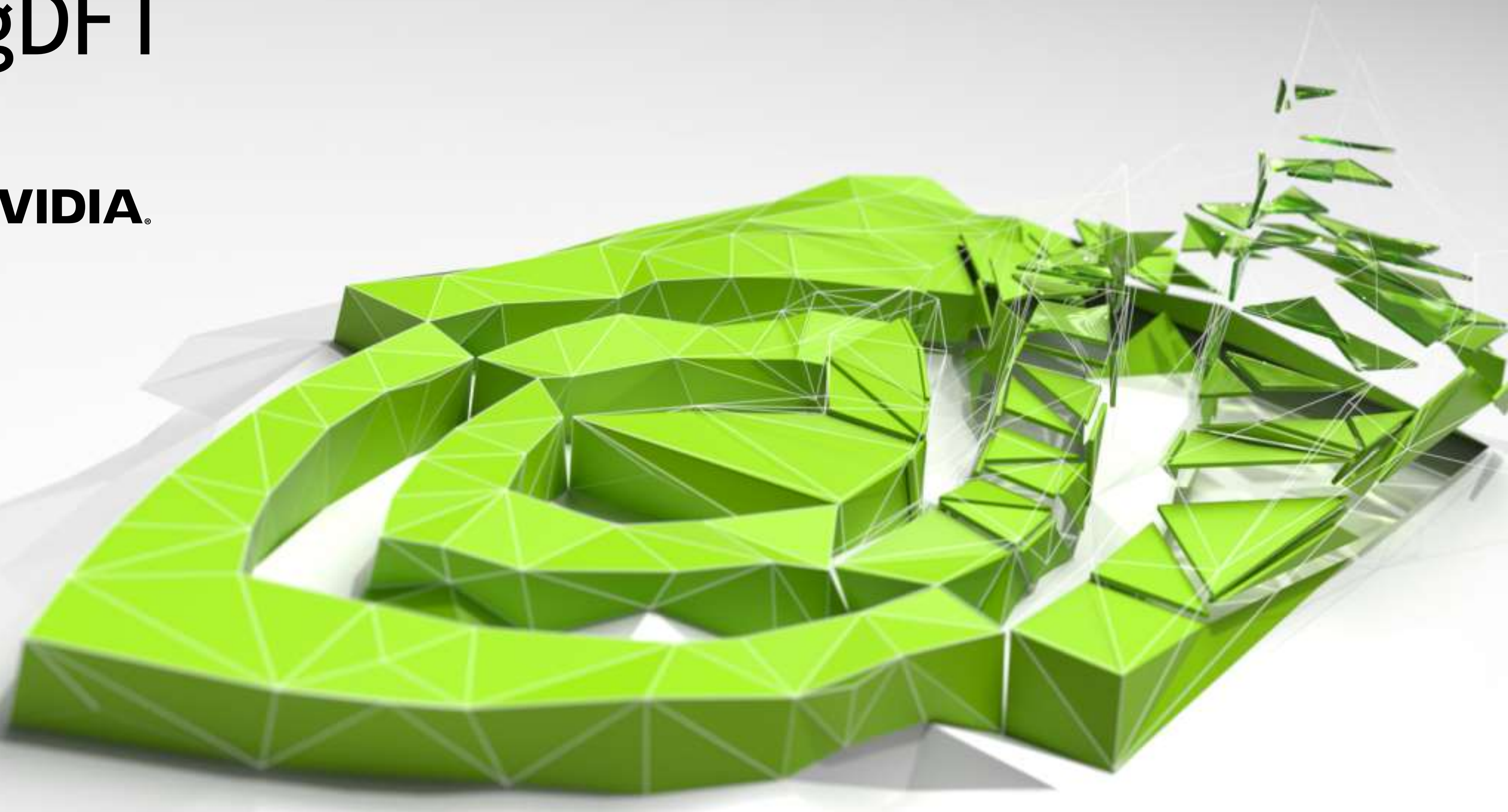
[3] Mainz et al, *Computer Physics Communication* **182**, p1421 (2011)

[4] Hacene et al,, *Journal of Computational Chemistry* **33**, p2581(2012)

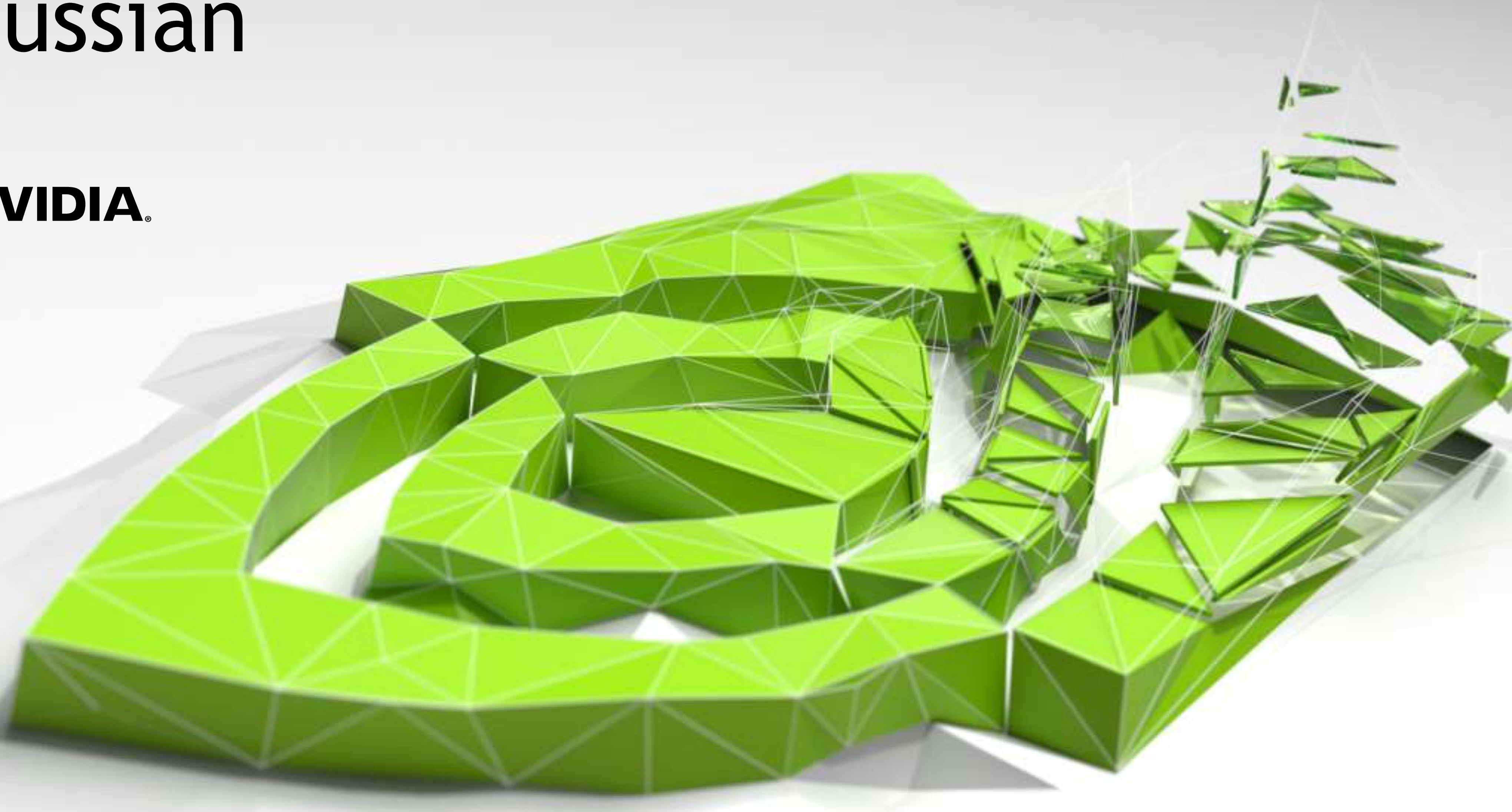
[5] Genovese et al, *Journal of Chemical Physics*, **131**, 034103 (2009)

[6] Hakala et al, *PARA 2012. LNCS*, vol **7782**, p63, Springer, Heidelberg (2013)

BigDFT



Gaussian



GPU TECHNOLOGY CONFERENCE

April 4-7, 2016 | Silicon Valley

Excerpts from . . .

ENABLING THE ELECTRONIC STRUCTURE PROGRAM GAUSSIAN ON GPGPUS USING OPENACC

Roberto Gomperts (NVIDIA), Michael Frisch (Gaussian, Inc.), Giovanni Scalmani (Gaussian, Inc.), Brent Leback (NVIDIA/PGI)

PRESENTED BY

NVIDIA.

PREVIOUSLY

Earlier Presentations

GRC Poster 2012

ACS Spring 2014

GTC Spring 2014 (recording at http://on-demand.gputechconf.com/gtc/2014/video/S4613-enabling-gaussian-09-gpgpus.mp4)

WATOC Fall 2014

Full presentation available

GTC Spring 2016 (this full recording at http://mygtc.gputechconf.com/quicklink/4r13O5r; requires registration)

CURRENT STATUS

Single Node

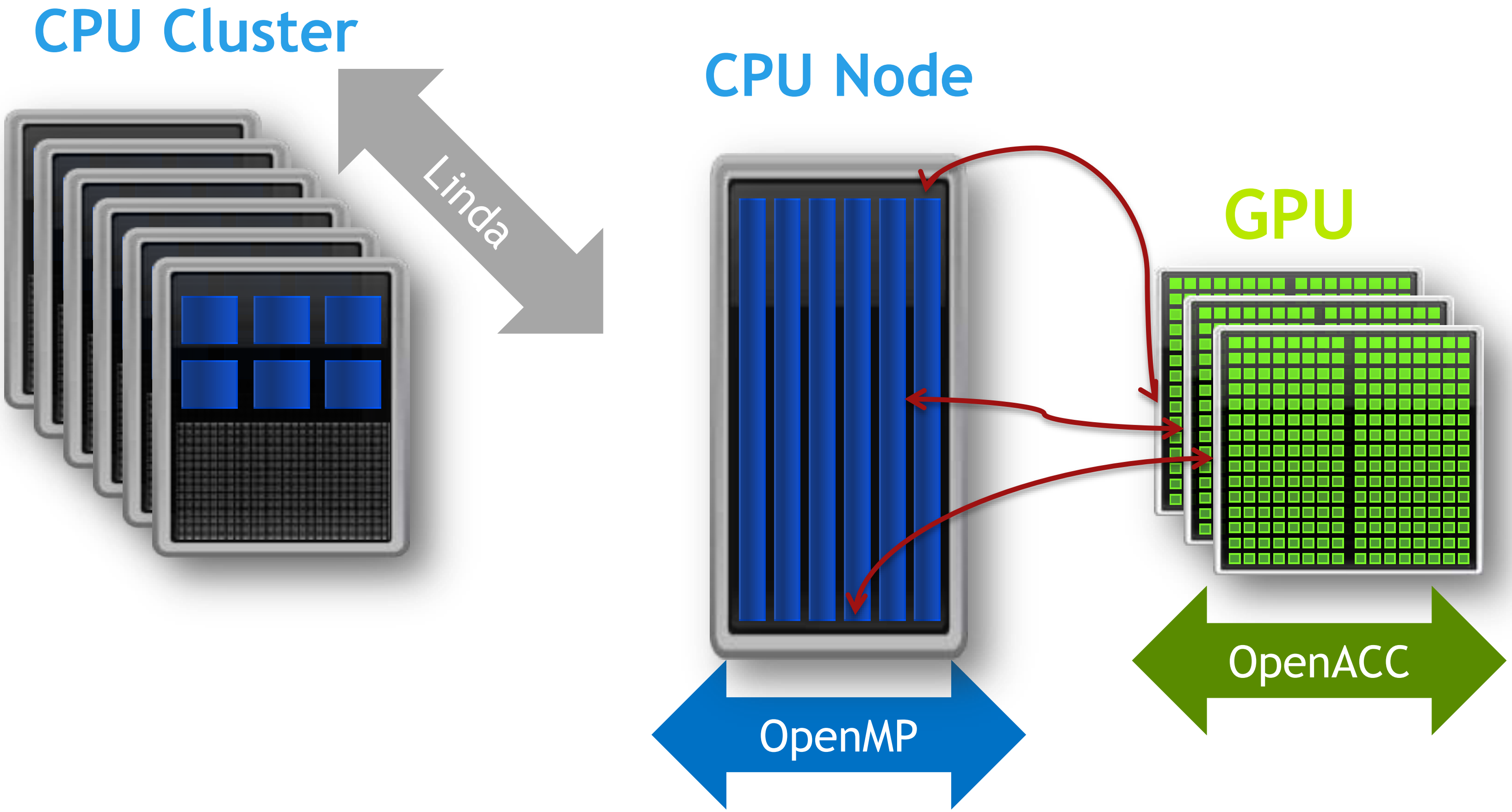
Implemented

- Energies for Closed and Open Shell HF and DFT (less than a handful of XC-functionals missing)
- First derivatives for the same as above
- Second derivatives for the same as above

Using only

- OpenACC
- CUDA library calls (BLAS)

GAUSSIAN PARALLELISM MODEL



CLOSING REMARKS

Significant Progress has been made in enabling Gaussian on GPUs with OpenACC

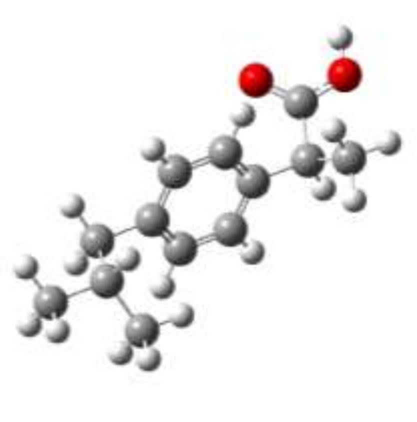
OpenACC is increasingly becoming more versatile

Significant work lies ahead to improve performance

Expand feature set:

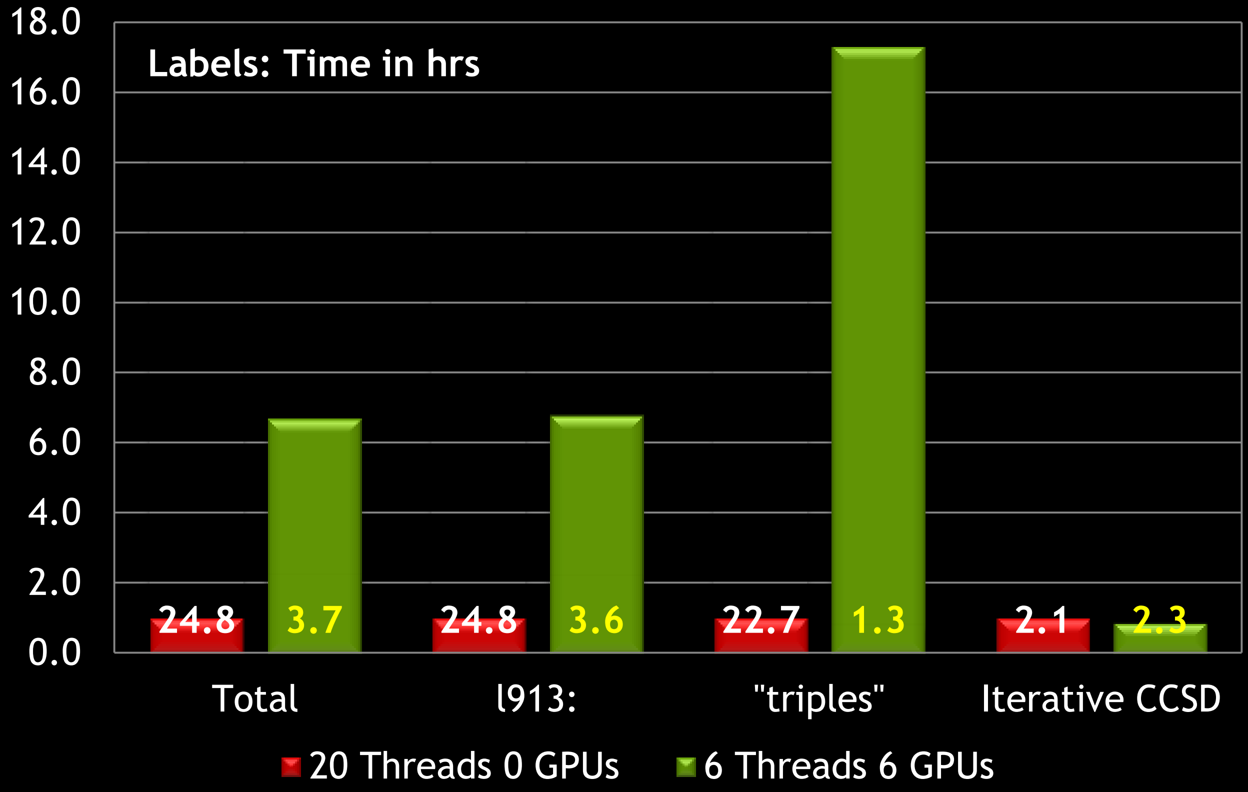
- PBC, Solvation, MP2, ONIOM, triples-Corrections

EARLY PERFORMANCE RESULTS (CCSD(T))



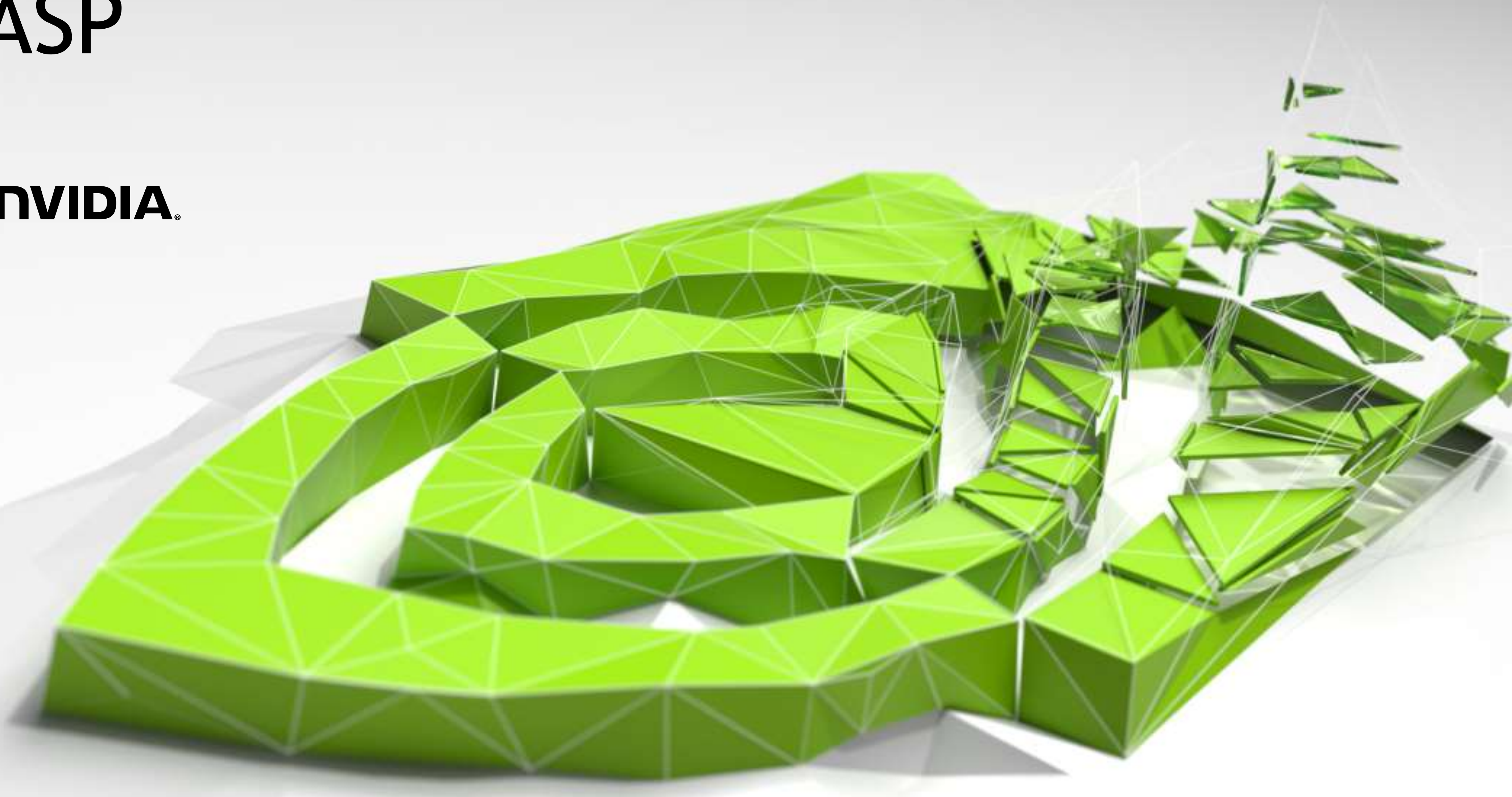
Method	CCSD(t)
No. of Atoms	33
Basis Set	6-31G(d,p)
No. of Basis Funcs	315
No. Occ Orbitals	41
No. Virt Orbitals	259
No. of Cycles	15
No. CCSD iters	16

Ibuprofen CCSD(t) Calculation
Speed Ups Relative to CPU-Only Full Node



@ 3.0 GHz); 128 GB RAM (DD3-1600); Used 108 GB
2 GB Global Memory

VASP



GPU VASP Collaboration

- ## Collaborators

U of Chicago

- ## 2013-2014 Project Scope

Minimization algorithms to calculate electronic ground state

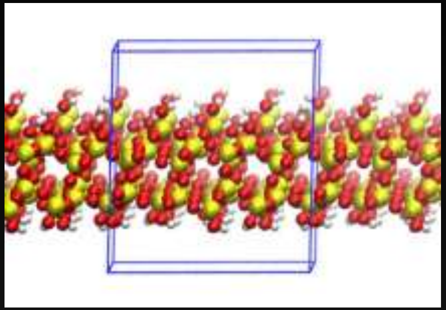
  - Blocked Davidson (ALGO = NORMAL & FAST)
  - RMM-DIIS (ALGO = VERYFAST & FAST)
  - K-Points
  - Optimization for critical step in exact exchange calculations

- ## Earlier work

  - *Speeding up plane-wave electronic-structure calculations using graphics-processing units*, Maintz, Eck, Dronskowski
  - *VASP on a GPU: application to exact-exchange calculations of the stability of elemental boron*, Hutchinson, Widom
  - *Accelerating VASP Electronic Structure Calculations Using Graphic Processing Units*, Hacene, Anciaux-Sedrakian, Rozanska, Klahr, Guignon, Fleurat-Lessard

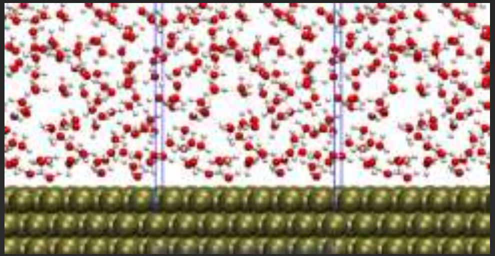
Target Workloads

- **Silica** *("medium")*
  - 7 Å thick slab of amorphous silica, 240 atoms ($Si_{68}O_{148}H_{24}$)
  - RMM-DIIS (ALGO = VERYFAST)



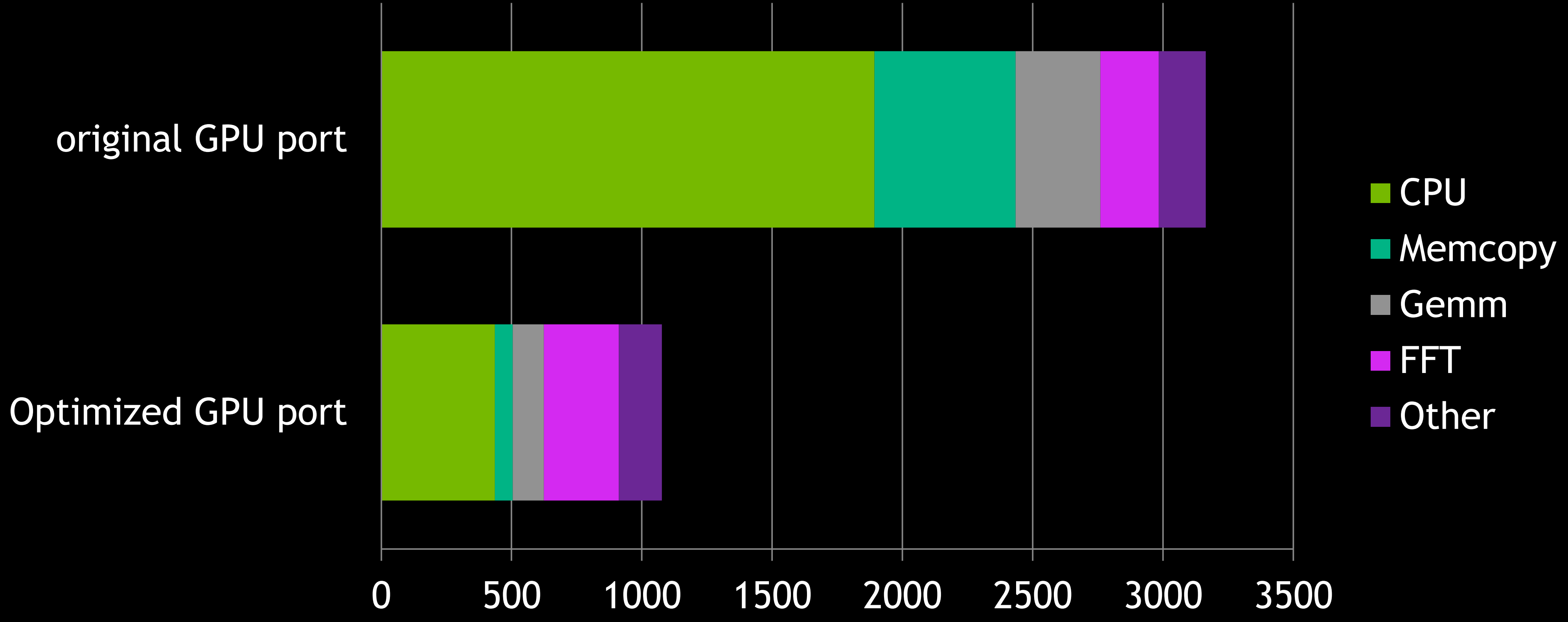
- **Nial-MD** *("large")*
  - Liquid metal molecular dynamics sample of Nickel-based superalloy
  - 500 atoms, 9 chemical species total
  - Blocked Davidson (ALGO = NORMAL)

- **INTERFACE** *("large")*
  - Interface of platinum metal with water
  - 108 Pt atoms, and 120 water molecules (468 atoms)
  - Blocked Davidson & RMM-DIIS (ALGO = FAST)



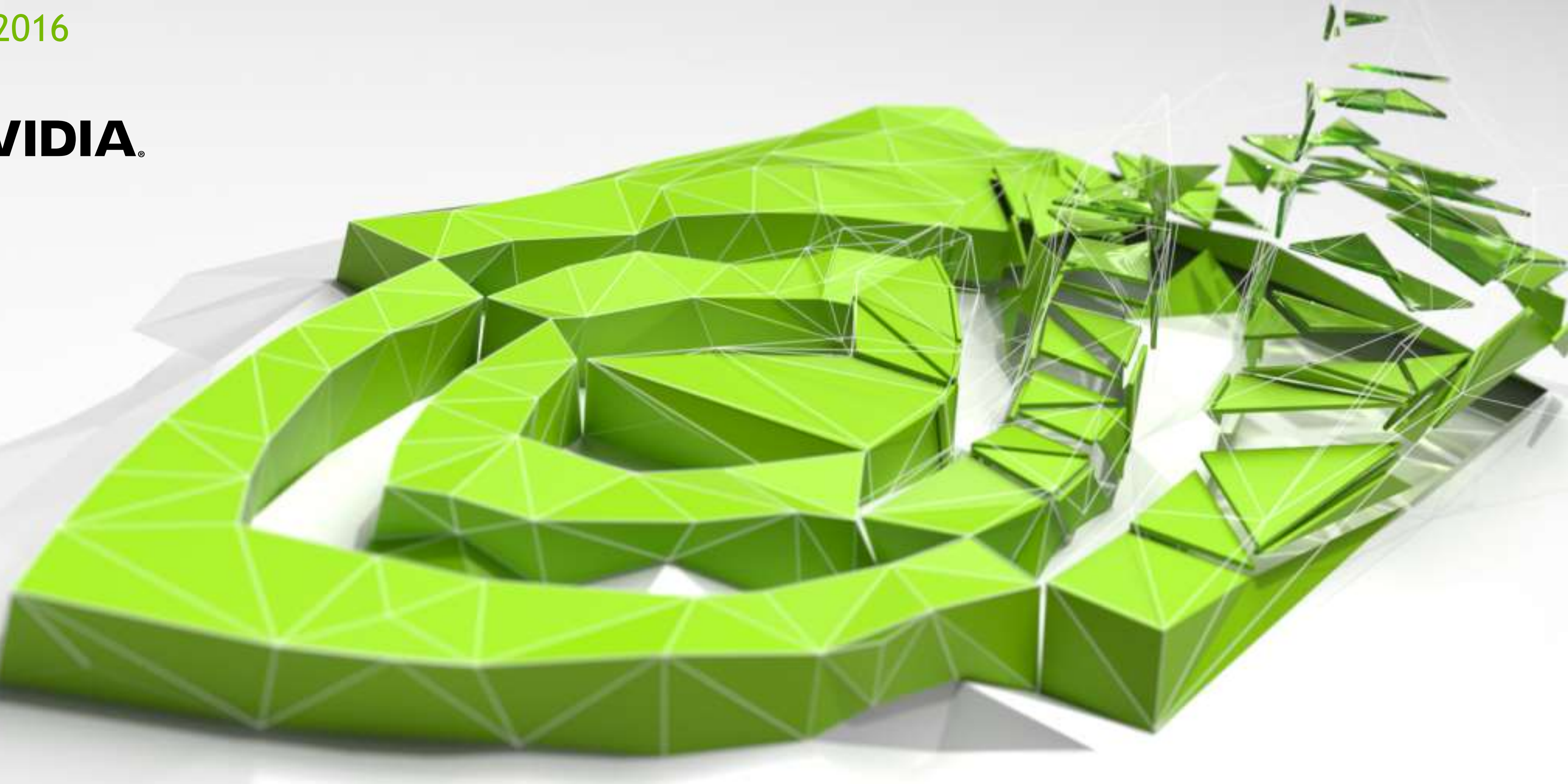
RUNTIME DISTRIBUTION FOR SILICA

Time in sec for 1 K40 GPU + 1 IvyBridge core

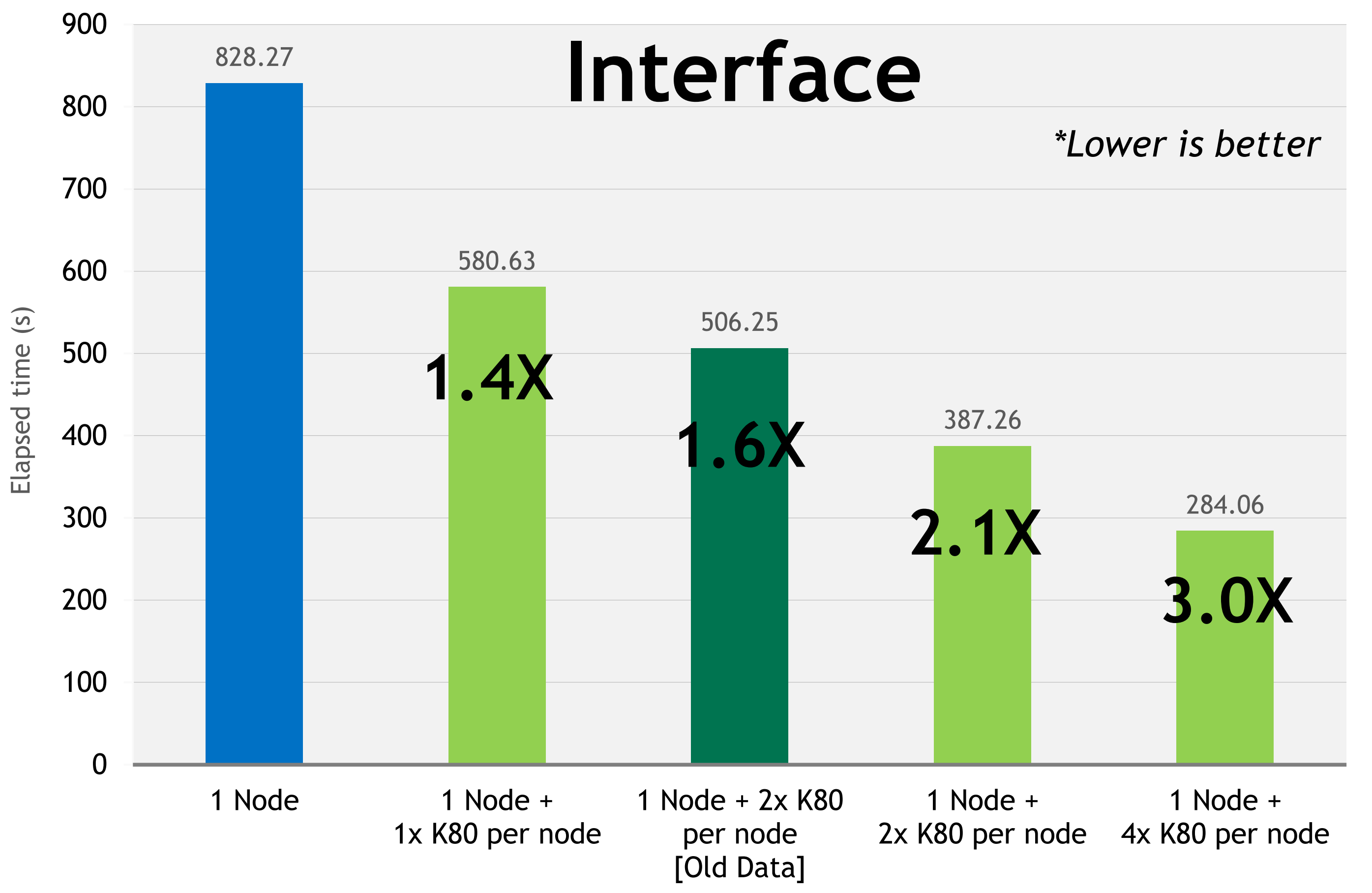


VASP 5.4.1 w/ Patch#1(Haswell & K80)

March 2016



VASP Interface Benchmark



Running VASP version 5.4.1

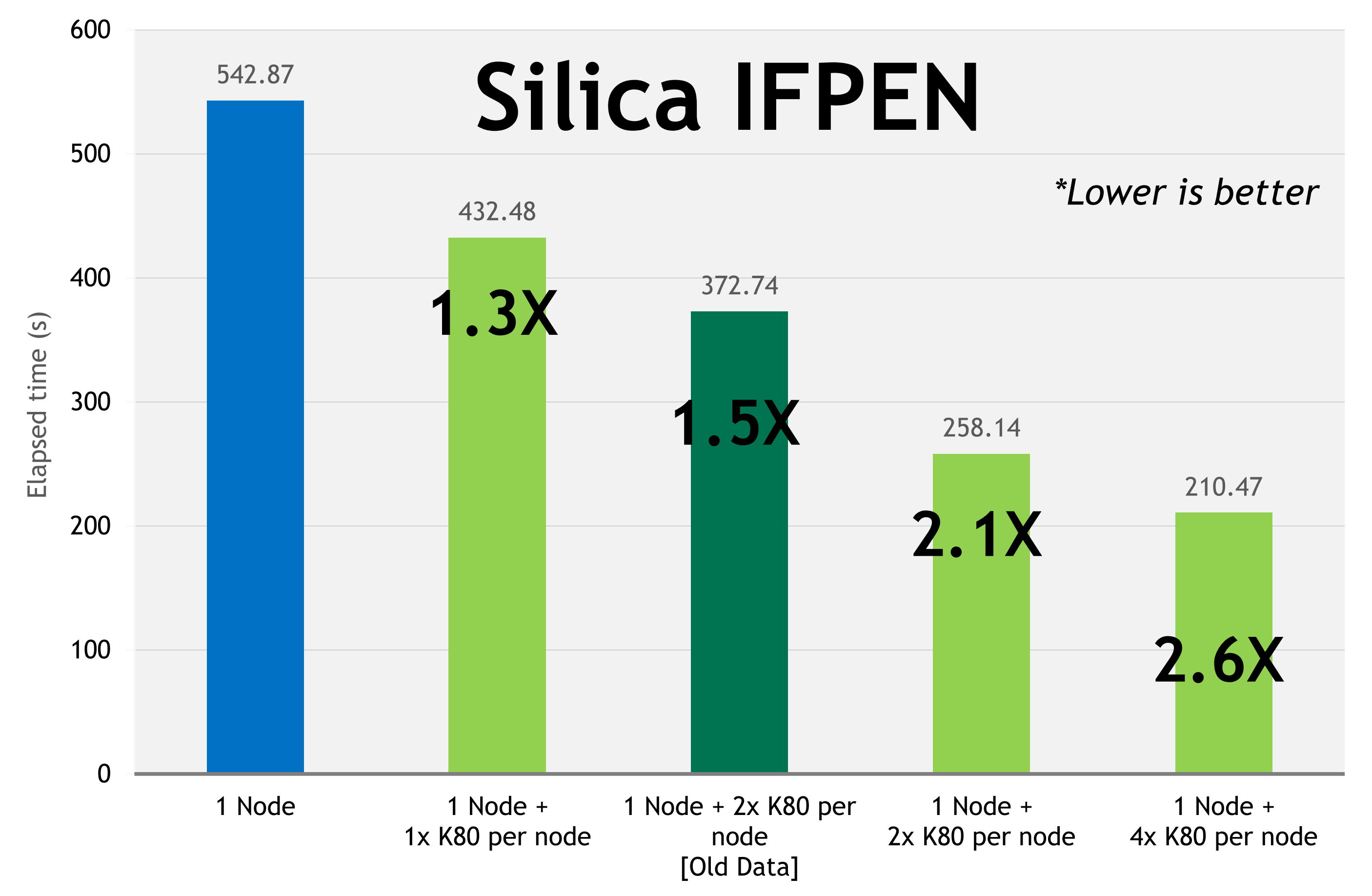
The blue node contains Dual Intel Xeon E5-2698 v3@2.3GHz (Haswell) CPUs

The green nodes contain Dual Intel Xeon E5-2698 v3@2.3GHz (Haswell) CPUs + Tesla K80 (autoboost) GPUs

“[Old Data]” = pre-Bugfix: patch #1 for vasp.5.4.1.05Feb15 which yield up to 2.7X faster calculations. (patch #1 available at VASP.at)

Blocked Davidson + RMM-DIIS (ALGO=Fast)

VASP Silica IFPEN Benchmark



Running VASP version 5.4.1

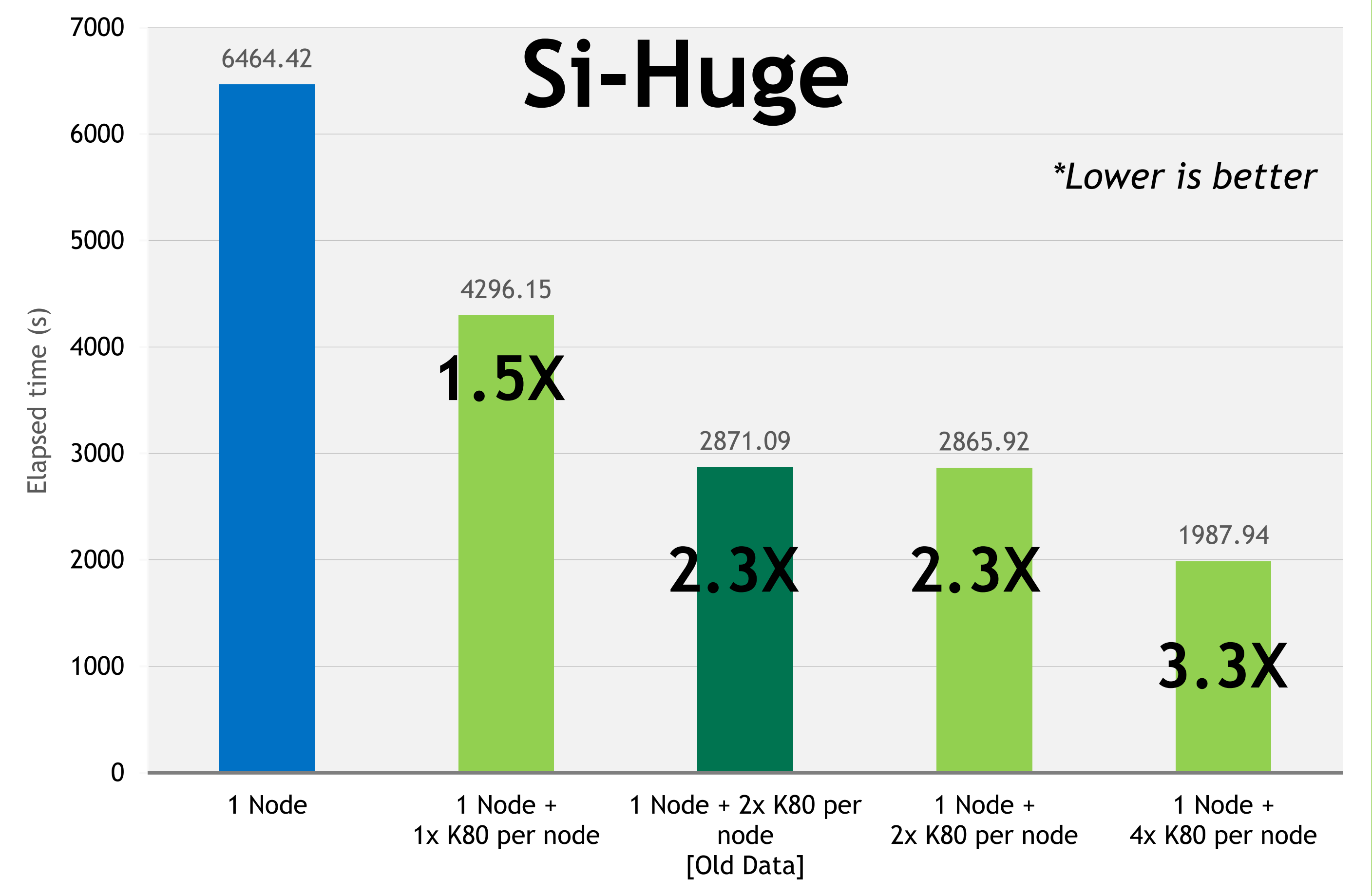
The blue node contains Dual Intel Xeon E5-2698 v3@2.3GHz (Haswell) CPUs

The green nodes contain Dual Intel Xeon E5-2698 v3@2.3GHz (Haswell) CPUs + Tesla K80 (autoboost) GPUs

“[Old Data]” = pre-Bugfix: patch #1 for vasp.5.4.1.05Feb15 which yield up to 2.7X faster calculations. (patch #1 available at VASP.at)

RMM-DIIS (ALGO=Veryfast)

VASP Si-Huge Benchmark



Running VASP version 5.4.1

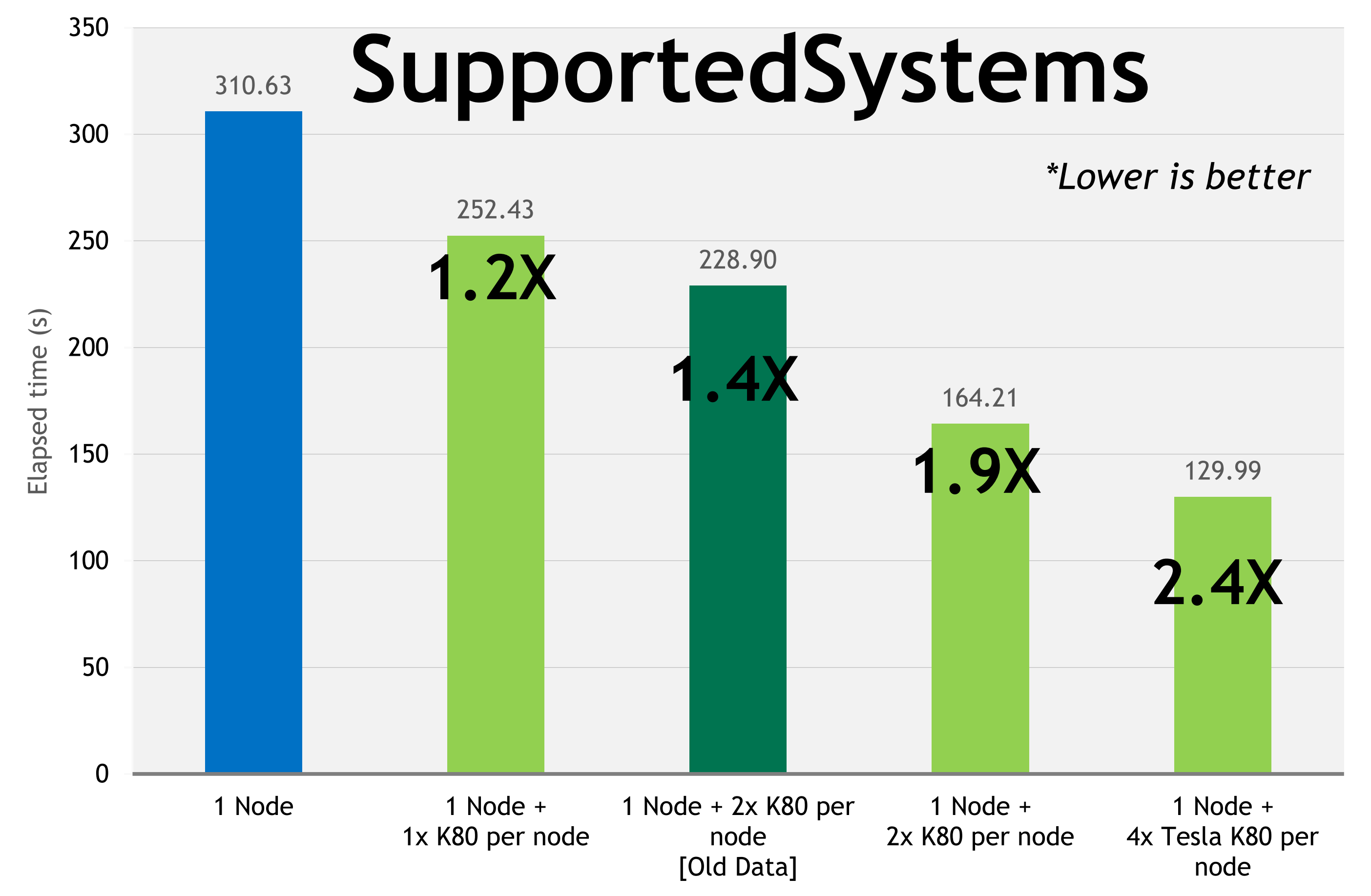
The blue node contains Dual Intel Xeon E5-2698 v3@2.3GHz (Haswell) CPUs

The green nodes contain Dual Intel Xeon E5-2698 v3@2.3GHz (Haswell) CPUs + Tesla K80 (autoboost) GPUs

“[Old Data]” = pre-Bugfix: patch #1 for vasp.5.4.1.05Feb15 which yield up to 2.7X faster calculations. (patch #1 available at VASP.at)

Blocked Davidson + RMM-DIIS (ALGO=Fast)

VASP SupportedSystems Benchmark



Running VASP version 5.4.1

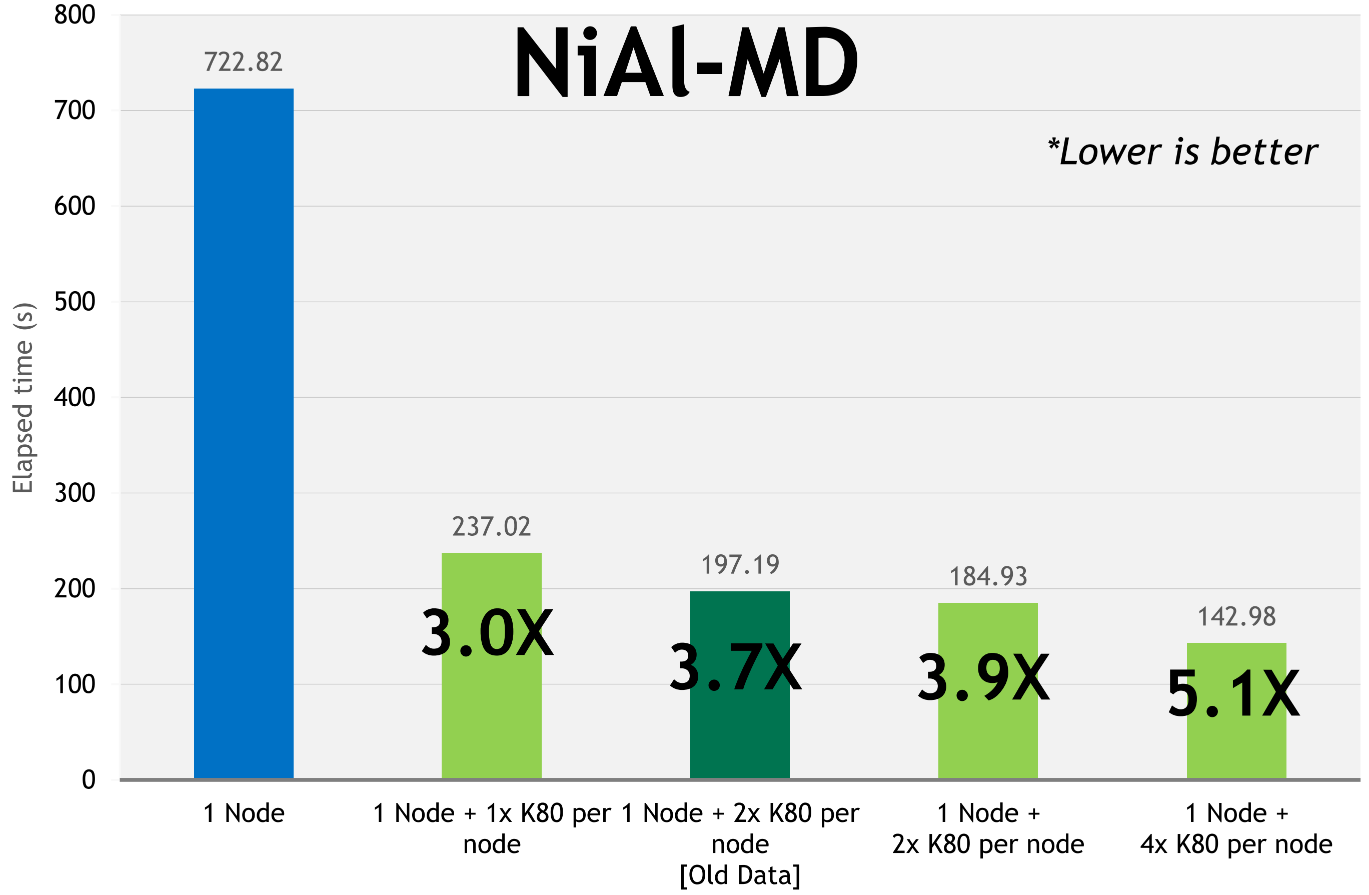
The blue node contains Dual Intel Xeon E5-2698 v3@2.3GHz (Haswell) CPUs

The green nodes contain Dual Intel Xeon E5-2698 v3@2.3GHz (Haswell) CPUs + Tesla K80 (autoboost) GPUs

“[Old Data]” = pre-Bugfix: patch #1 for vasp.5.4.1.05Feb15 which yield up to 2.7X faster calculations. (patch #1 available at VASP.at)

Blocked Davidson + RMM-DIIS (ALGO=Fast)

VASP NiAl-MD Benchmark



Running VASP version 5.4.1

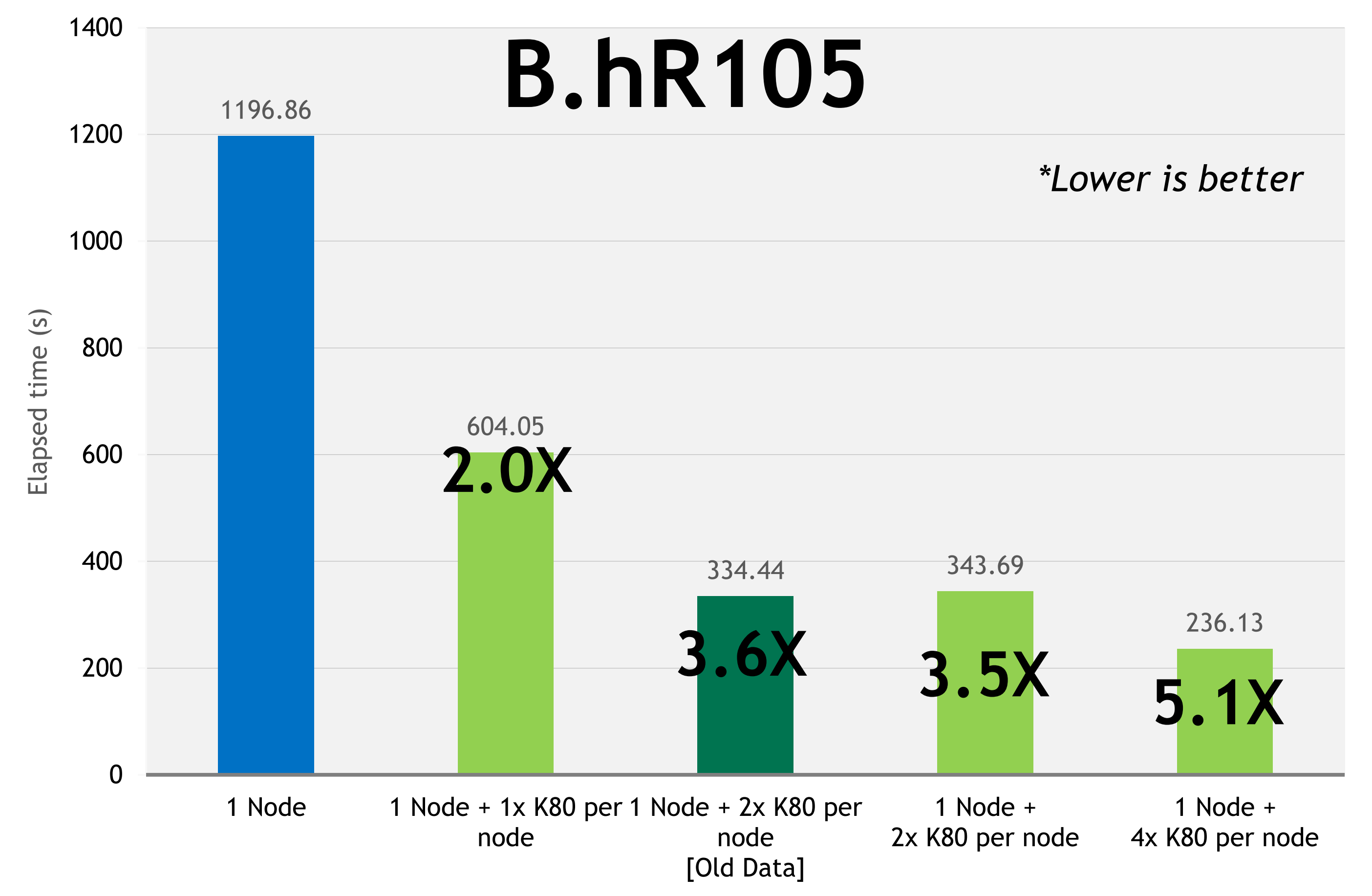
The blue node contains Dual Intel Xeon E5-2698 v3@2.3GHz (Haswell) CPUs

The green nodes contain Dual Intel Xeon E5-2698 v3@2.3GHz (Haswell) CPUs + Tesla K80 (autoboost) GPUs

“[Old Data]” = pre-Bugfix: patch #1 for vasp.5.4.1.05Feb15 which yield up to 2.7X faster calculations. (patch #1 available at VASP.at)

Blocked Davidson + RMM-DIIS (ALGO=Fast)

VASP B.hR105 Benchmark



Running VASP version 5.4.1

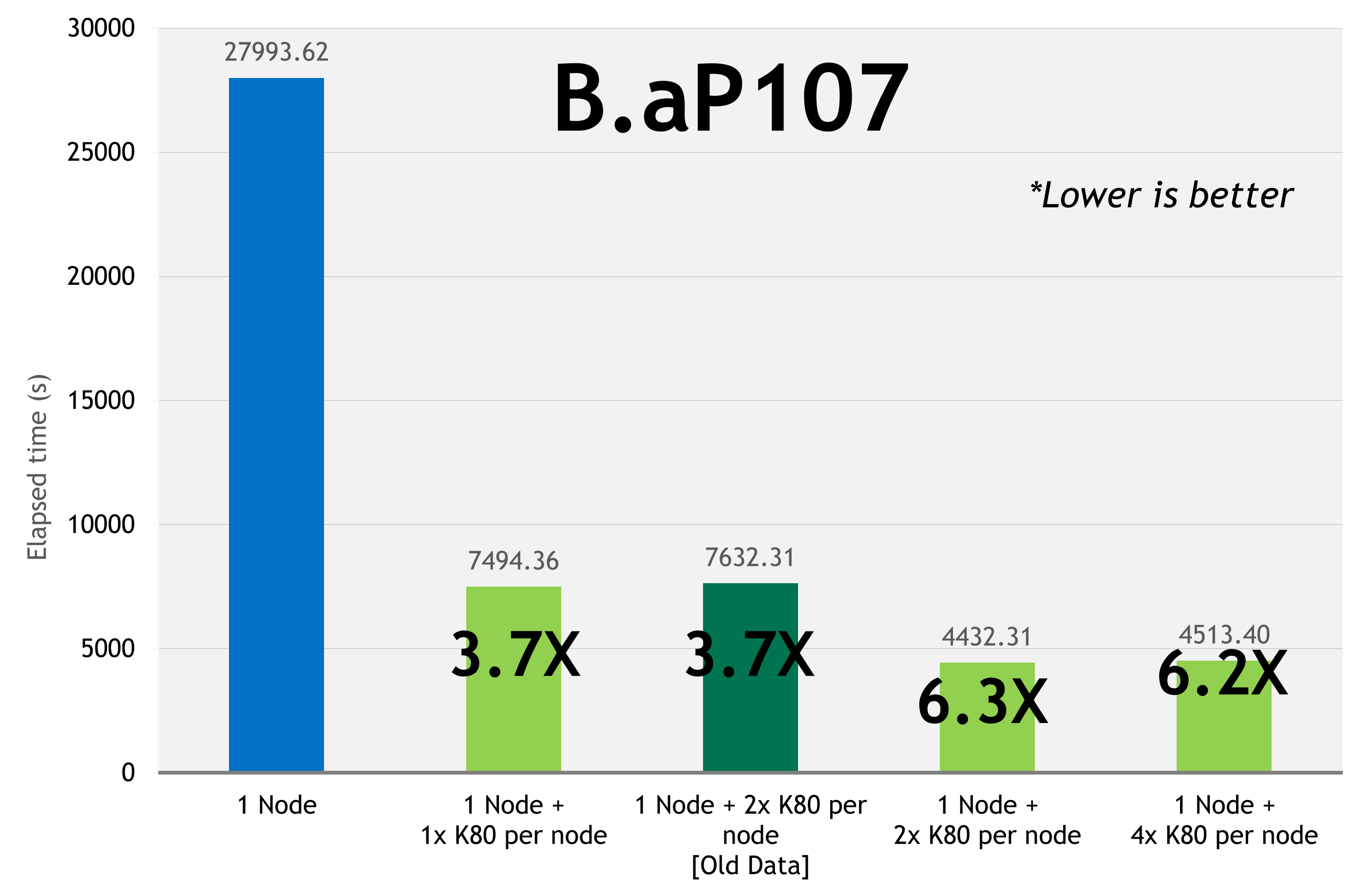
The blue node contains Dual Intel Xeon E5-2698 v3@2.3GHz (Haswell) CPUs

The green nodes contain Dual Intel Xeon E5-2698 v3@2.3GHz (Haswell) CPUs + Tesla K80 (autoboost) GPUs

“[Old Data]” = pre-Bugfix: patch #1 for vasp.5.4.1.05Feb15 which yield up to 2.7X faster calculations. (patch #1 available at VASP.at)

Hybrid Functional with blocked Davicson (ALGO=Normal)

VASP B.aP107 Benchmark



Running VASP version 5.4.1

The blue node contains Dual Intel Xeon E5-2698 v3@2.3GHz (Haswell) CPUs

The green nodes contain Dual Intel Xeon E5-2698 v3@2.3GHz (Haswell) CPUs + Tesla K80 (autoboost) GPUs

“[Old Data]” = pre-Bugfix: patch #1 for vasp.5.4.1.05Feb15 which yield up to 2.7X faster calculations. (patch #1 available at VASP.at)

Hybrid functional calculation (exact exchange) with blocked Davidson. No KPoint parallelization.

Hybrid Functional with blocked Davicson (ALGO=Normal)

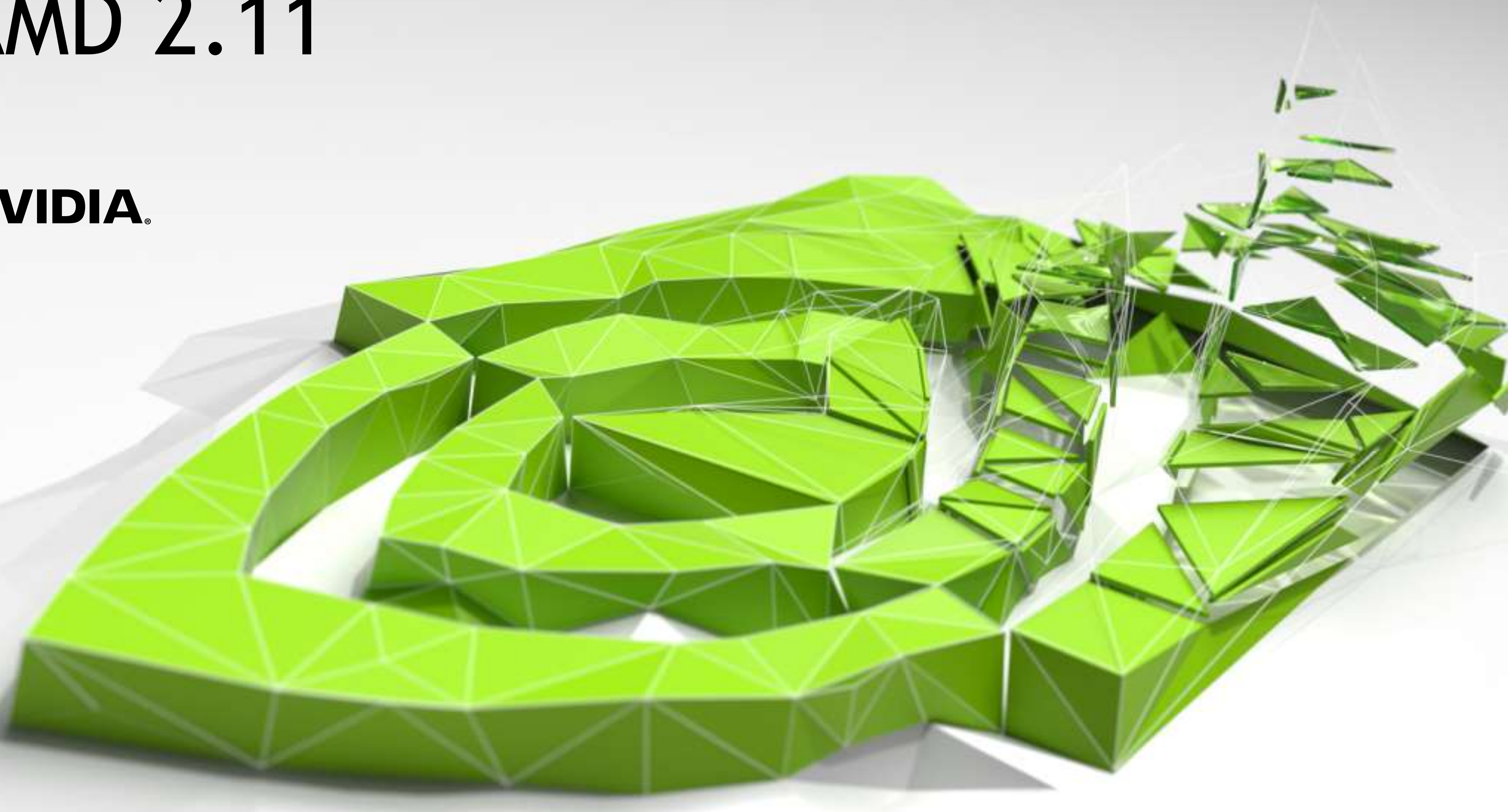
GPU-ACCELERATED MOLECULAR DYNAMICS APPS

Green Lettering Indicates Performance Slides Included

- ACEMD
- AMBER
- CHARMM
- DESMOND
- ESPResSO
- Folding@Home
- GPUGrid.net
- GROMACS
- HALMD
- HOOMD-Blue
- LAMMPS
- mdcore
- MELD
- NAMD
- OpenMM
- PolyFTS

GPU Perf compared against dual multi-core x86 CPU socket.

NAMD 2.11

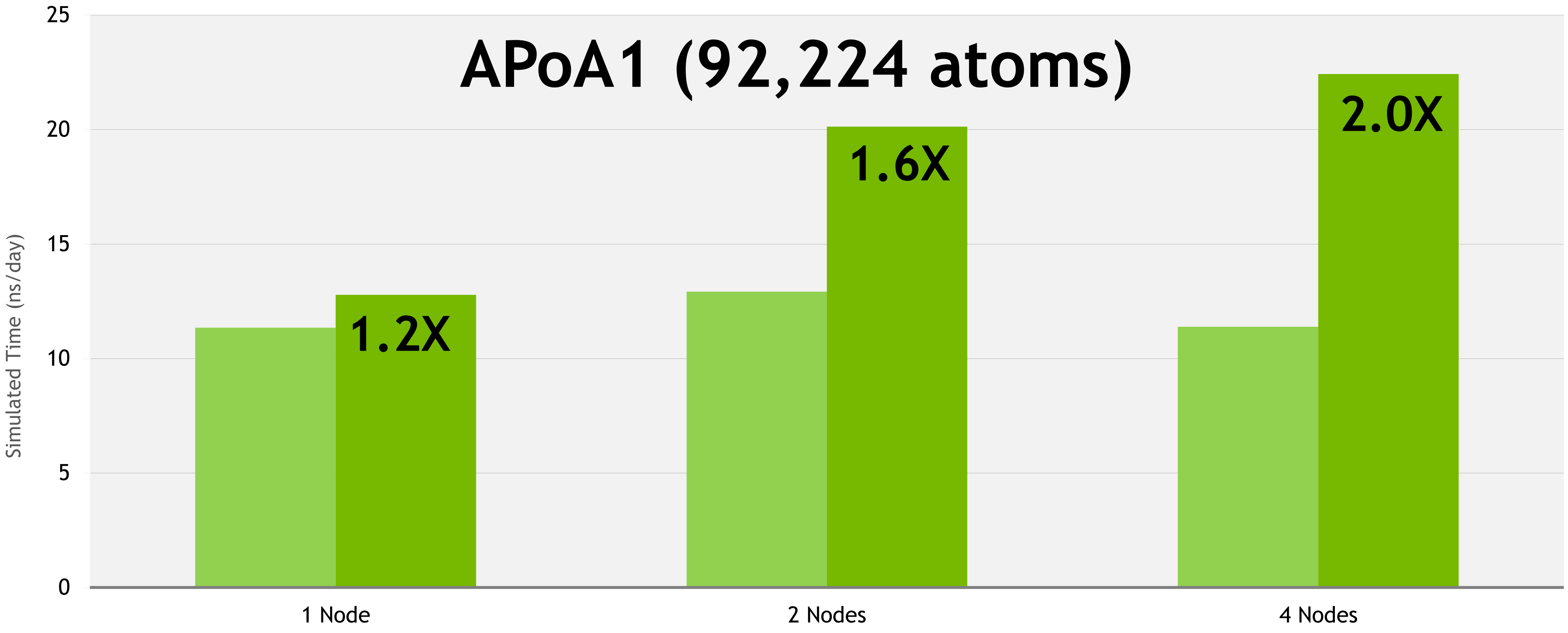


NEW GPU FEATURES IN NAMD 2.11

Selected Text from the NAMD website

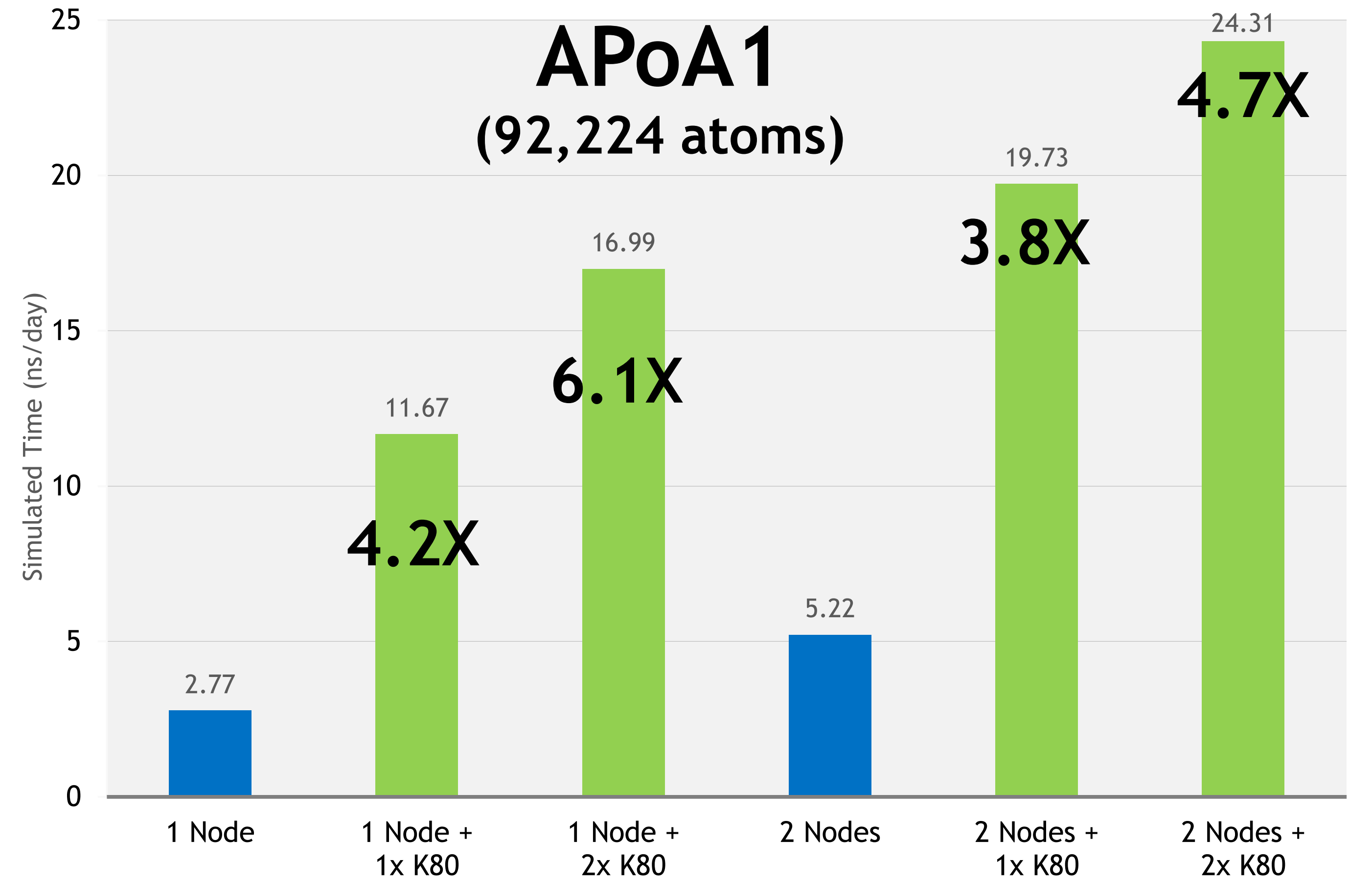
- **GPU-accelerated simulations up to twice as fast as NAMD 2.10**
- **Pressure calculation with fixed atoms on GPU works as on CPU**
- **Improved scaling for GPU-accelerated particle-mesh Ewald calculation**
    - CPU-side operations overlap better and are parallelized across cores.
- **Improved scaling for GPU-accelerated simulations**
    - Nonbonded force calculation results are streamed from the GPU for better overlap.
- **NVIDIA CUDA GPU-acceleration binaries for Mac OS X**

NAMD 2.11 IS UP TO 2X FASTER



NAMD 2.10 & NAMD 2.11 contain Dual Intel E5-2697 v2@2.7GHz (IvyBridge) CPUs + 2x Tesla K80 (autoboost) GPUs

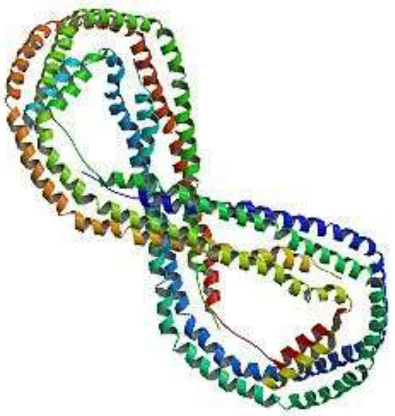
NAMD 2.11 APOA1 ON 1 AND 2 NODES



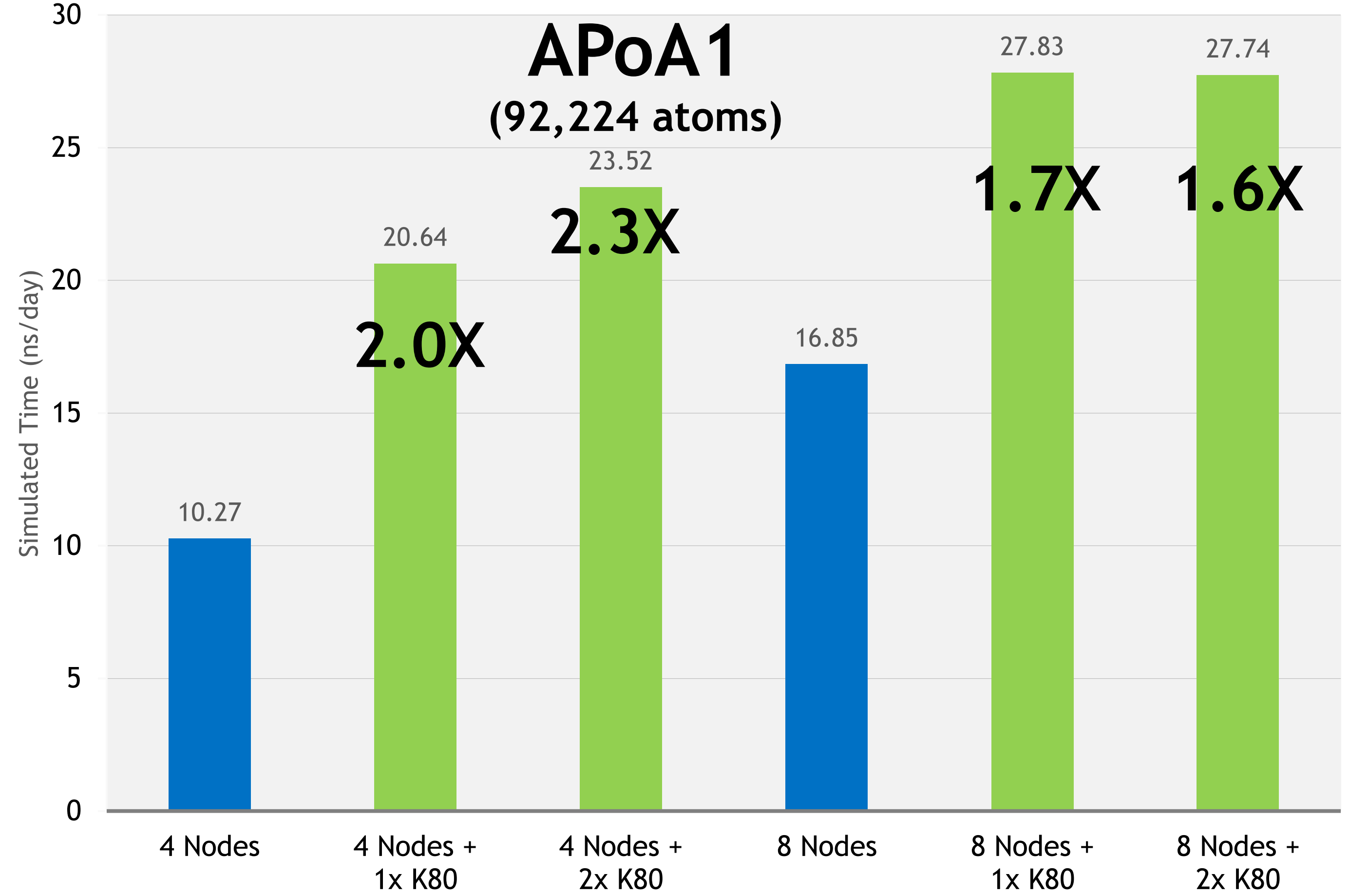
Running NAMD version 2.11

The blue nodes contain Dual Intel E5-2698 v3@2.3GHz (Haswell) CPUs

The green nodes contain Dual Intel E5-2698 v3@2.3GHz (Haswell) CPUs + Tesla K80 (autoboost) GPUs



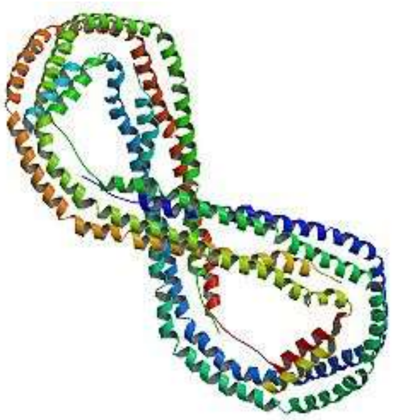
NAMD 2.11 APOA1 ON 4 AND 8 NODES



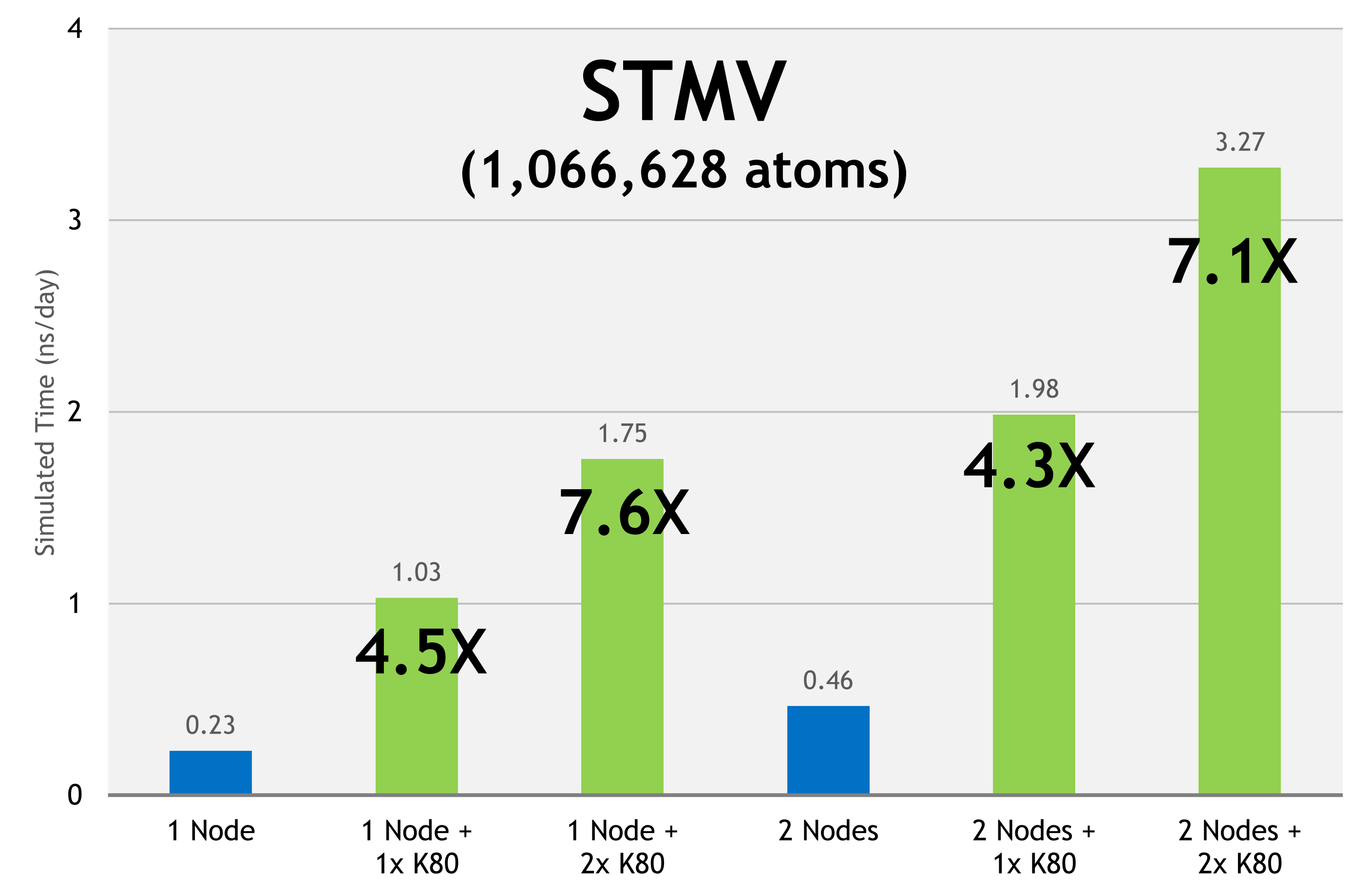
Running NAMD version 2.11

The blue nodes contain Dual Intel E5-2698 v3@2.3GHz (Haswell) CPUs

The green nodes contain Dual Intel E5-2698 v3@2.3GHz (Haswell) CPUs + Tesla K80 (autoboost) GPUs



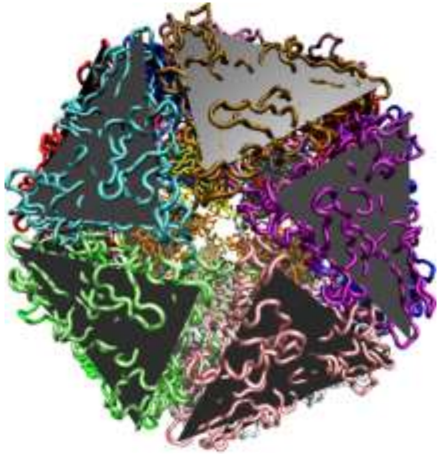
NAMD 2.11 STMV ON 1 AND 2 NODES



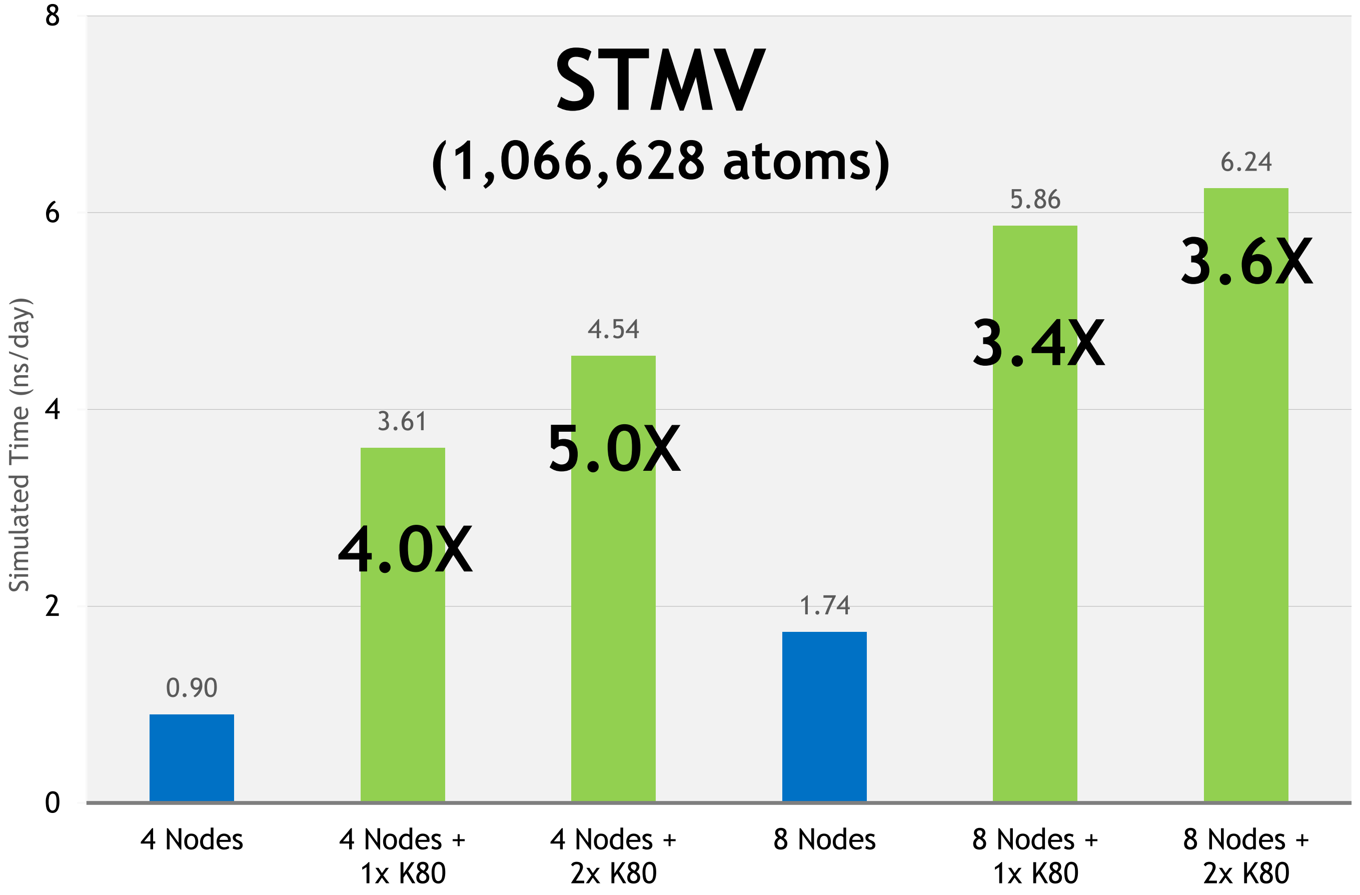
Running NAMD version 2.11

The blue nodes contain Dual Intel E5-2698 v3@2.3GHz (Haswell) CPUs

The green nodes contain Dual Intel E5-2698 v3@2.3GHz CPUs (Haswell) + Tesla K80 (autoboost) GPUs



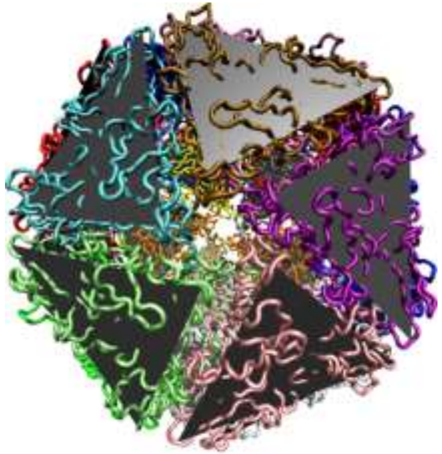
NAMD 2.11 STMV ON 4 AND 8 NODES



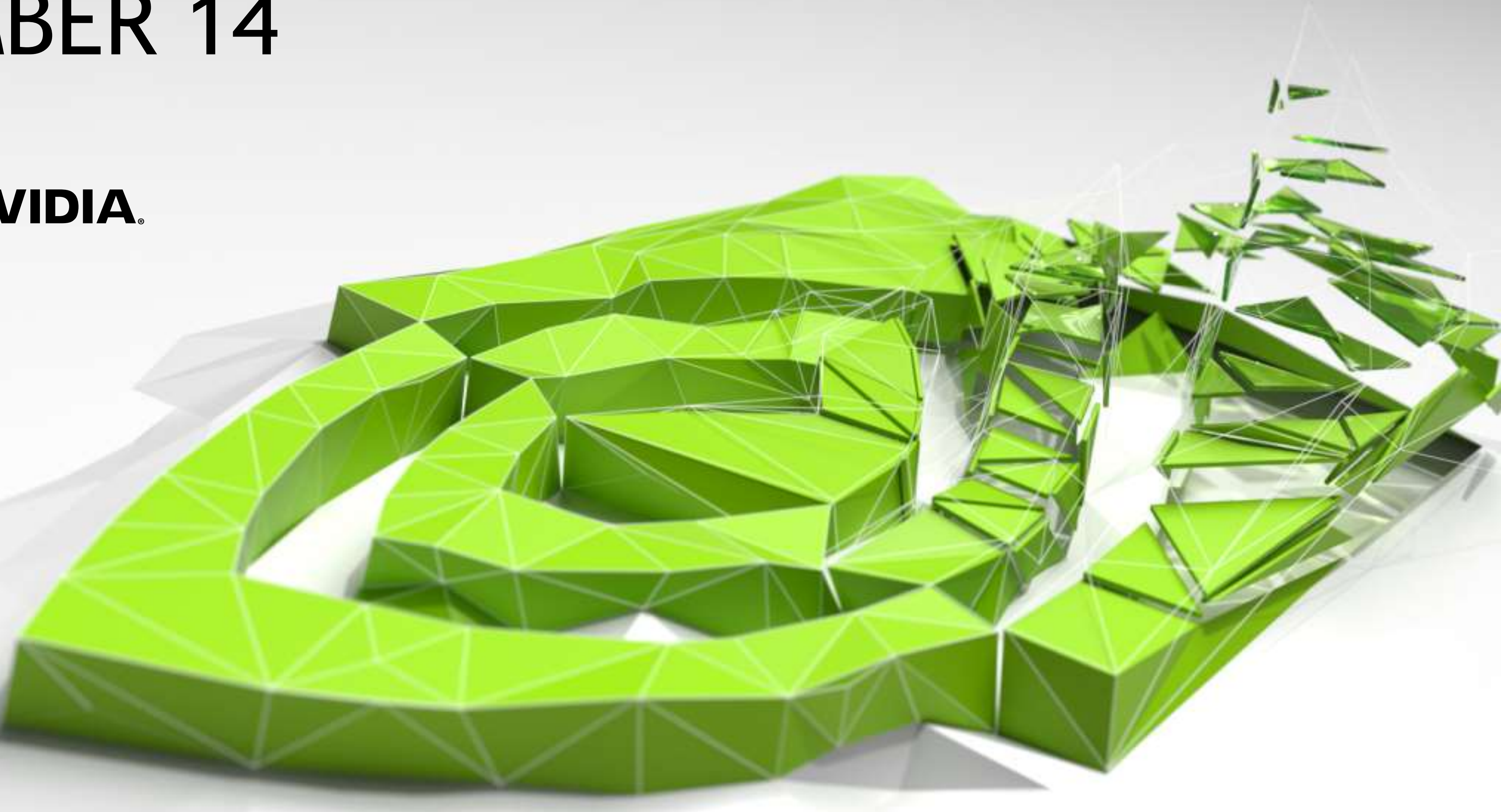
Running NAMD version 2.11

The blue nodes contain Dual Intel E5-2698 v3@2.3GHz (Haswell) CPUs

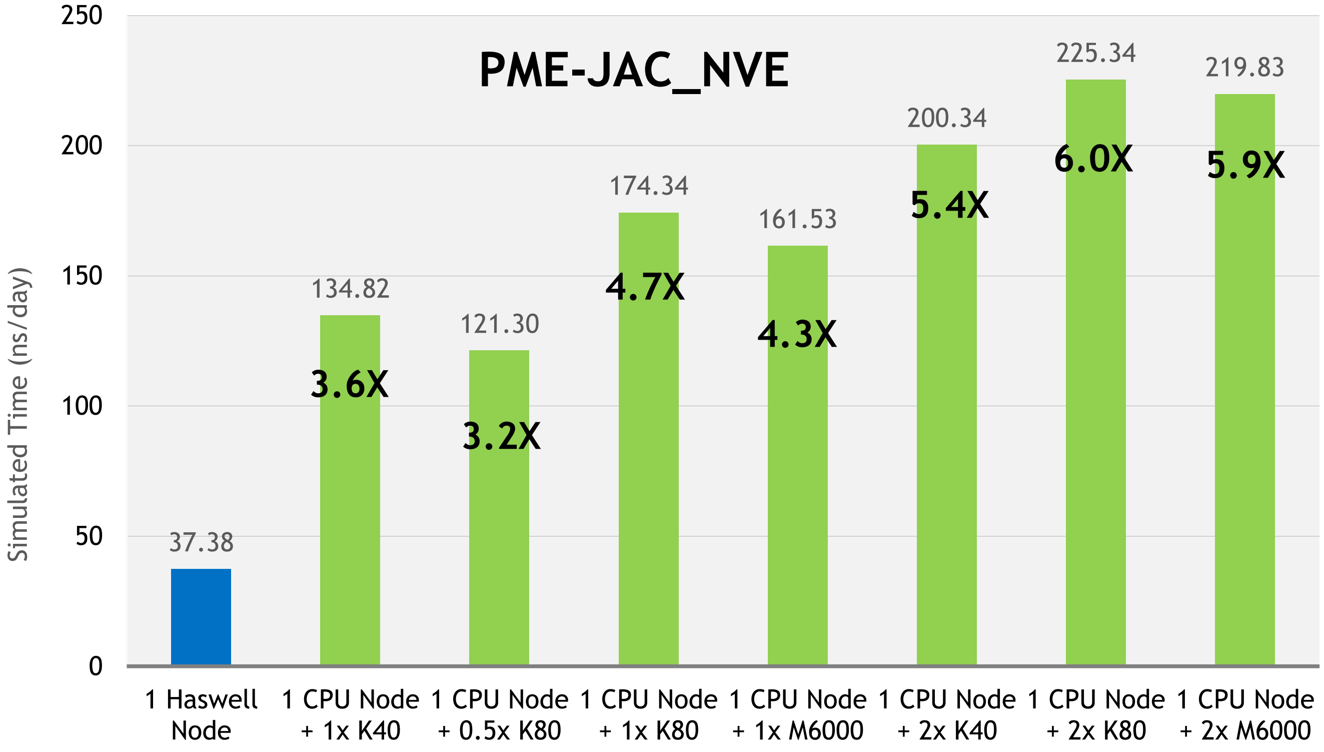
The green nodes contain Dual Intel E5-2698 v3@2.3GHz CPUs (Haswell) + Tesla K80 (autoboost) GPUs



AMBER 14



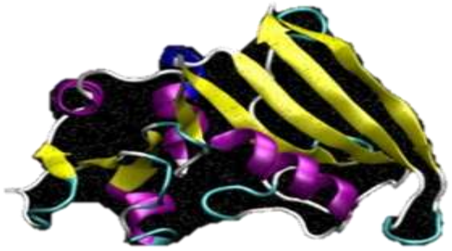
JAC on K40s, K80s and M6000s



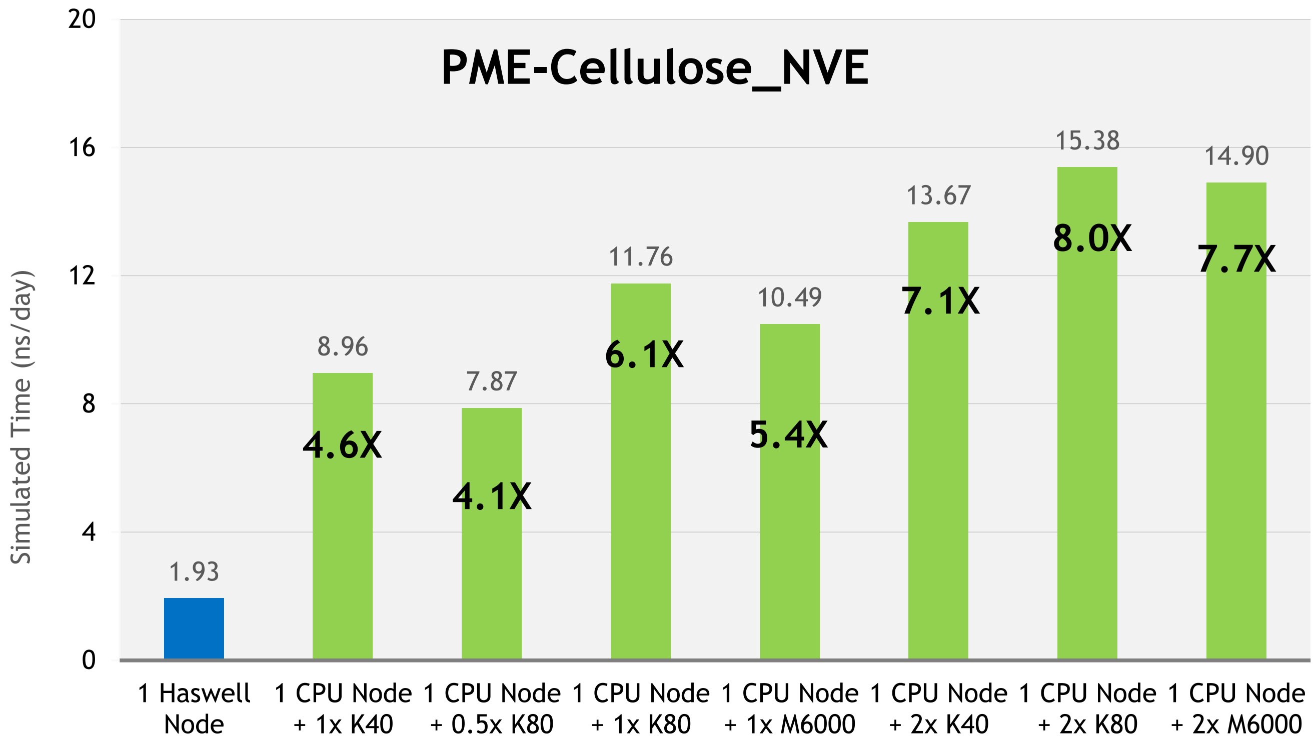
Running AMBER version 14

The blue node contains Dual Intel E5-2698 v3@2.3GHz, 3.6GHz Turbo CPUs

The green nodes contain Dual Intel E5-2698 v3@2.3GHz, 3.6GHz Turbo CPUs + either NVIDIA Tesla K40@875Mhz, Tesla K80@562Mhz (autoboost), or Quadro M6000@987Mhz GPUs



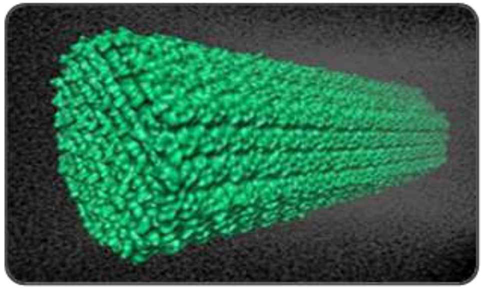
Cellulose on K40s, K80s and M6000s



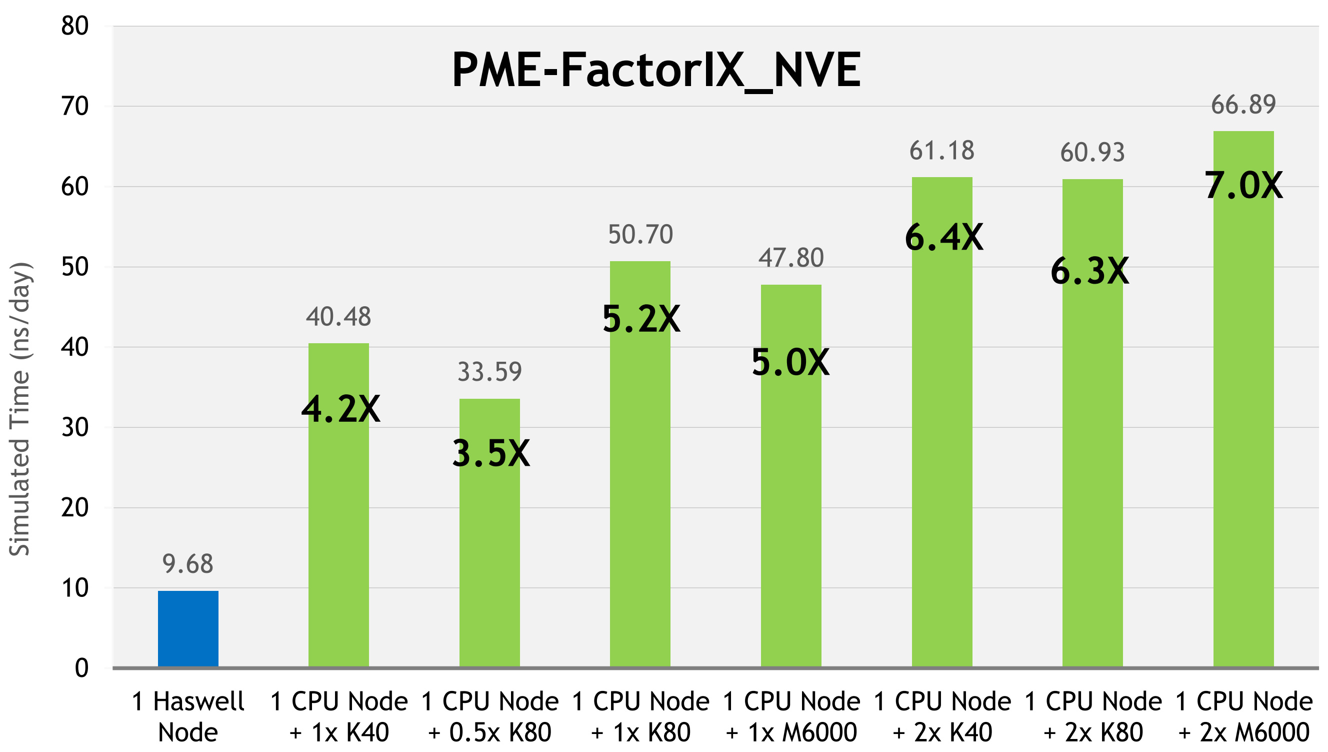
Running AMBER version 14

The blue node contains Dual Intel E5-2698 v3@2.3GHz, 3.6GHz Turbo CPUs

The green nodes contain Dual Intel E5-2698 v3@2.3GHz, 3.6GHz Turbo CPUs + either NVIDIA Tesla K40@875Mhz, Tesla K80@562Mhz (autoboost), or Quadro M6000@987Mhz GPUs



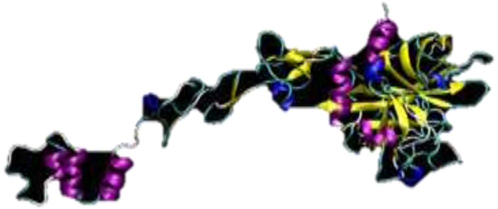
Factor IX on K40s, K80s and M6000s



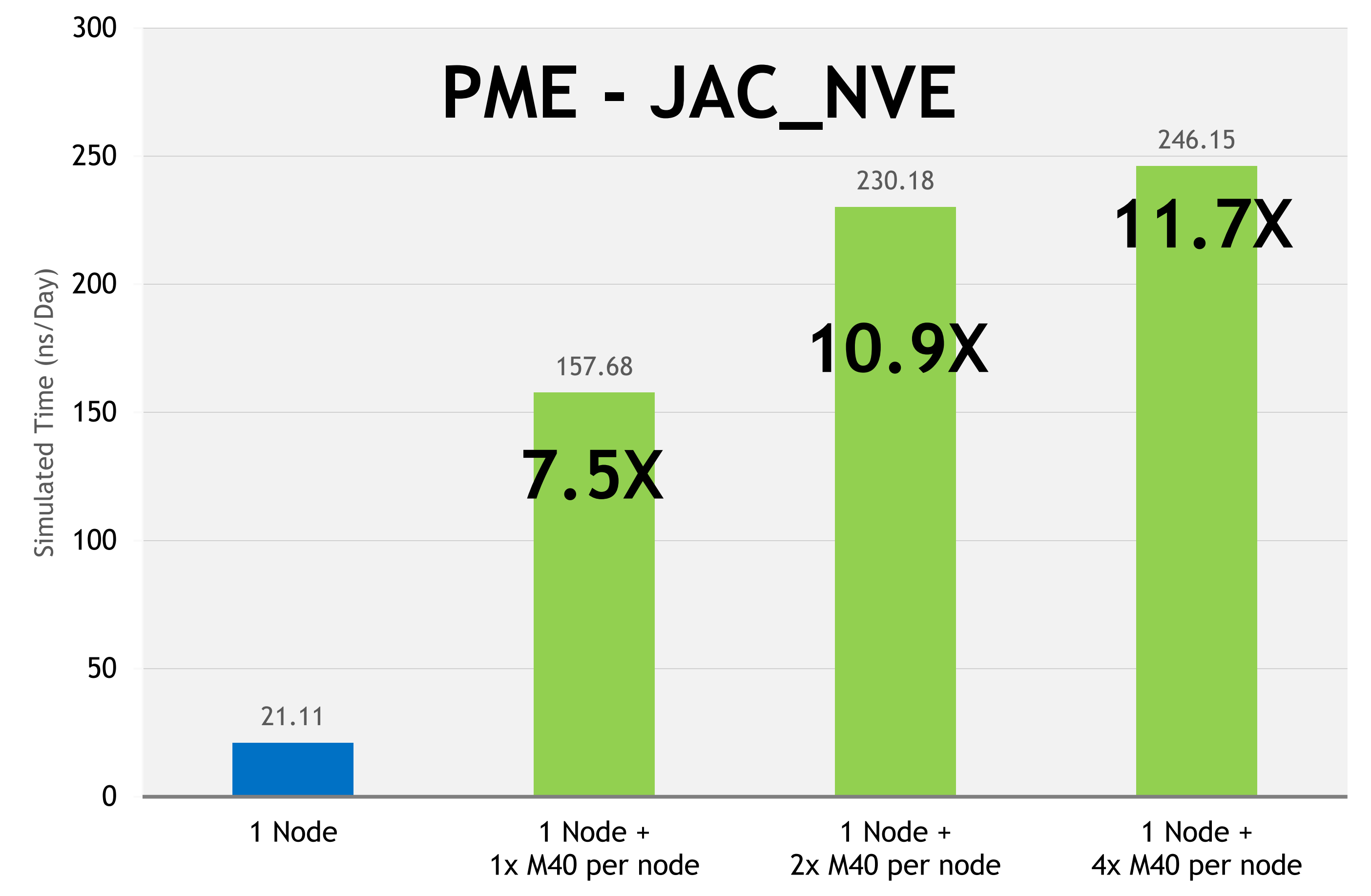
Running AMBER version 14

The blue node contains Dual Intel E5-2698 v3@2.3GHz, 3.6GHz Turbo CPUs

The green nodes contain Dual Intel E5-2698 v3@2.3GHz, 3.6GHz Turbo CPUs + either NVIDIA Tesla K40@875Mhz, Tesla K80@562Mhz (autoboost), or Quadro M6000@987Mhz GPUs



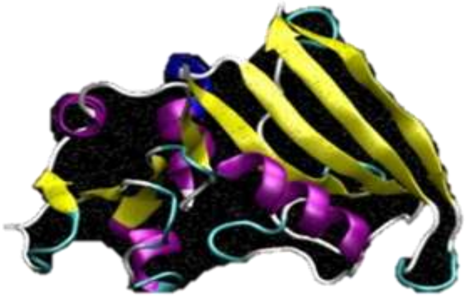
JAC on M40s



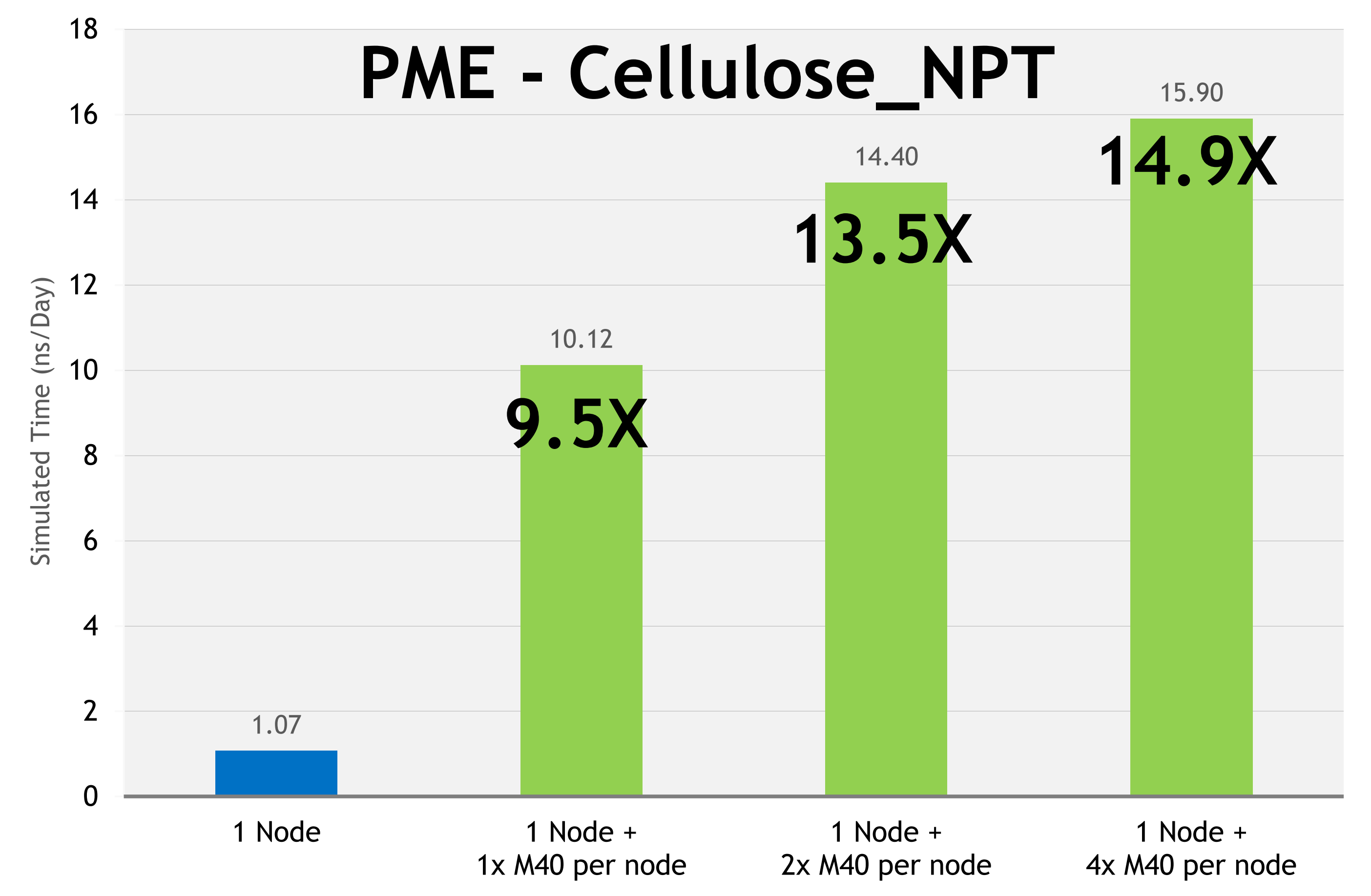
Running AMBER version 14

The blue node contain Single Intel Xeon E5-2698 v3@2.30GHz (Haswell) CPUs

The green nodes contain Single Intel Xeon E5-2697 v2@2.70GHz (IvyBridge) CPUs + Tesla M40 (autoboost) GPUs



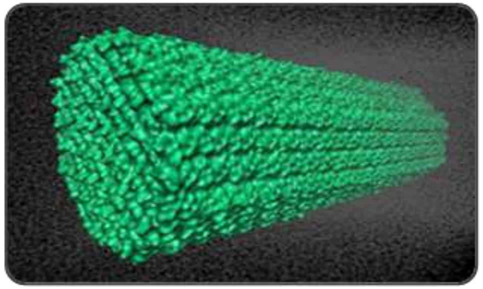
Cellulose on M40s



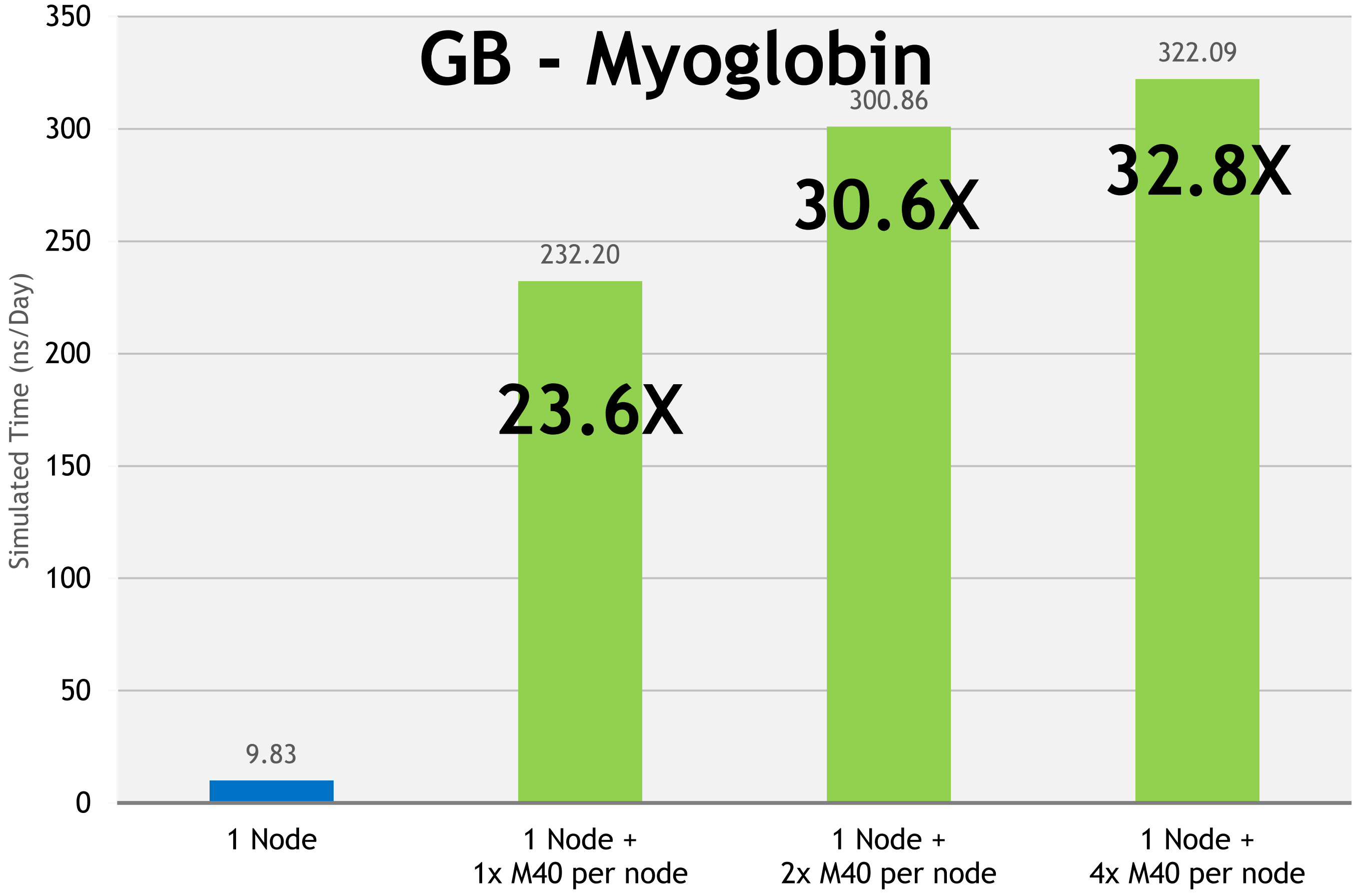
Running AMBER version 14

The blue node contain Single Intel Xeon E5-2698 v3@2.30GHz (Haswell) CPUs

The green nodes contain Single Intel Xeon E5-2697 v2@2.70GHz (IvyBridge) CPUs + Tesla M40 (autoboost) GPUs



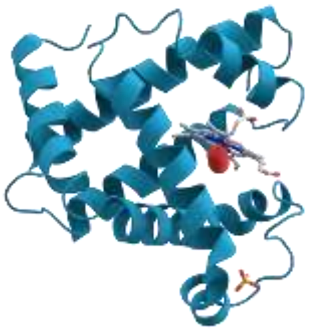
Myoglobin on M40s



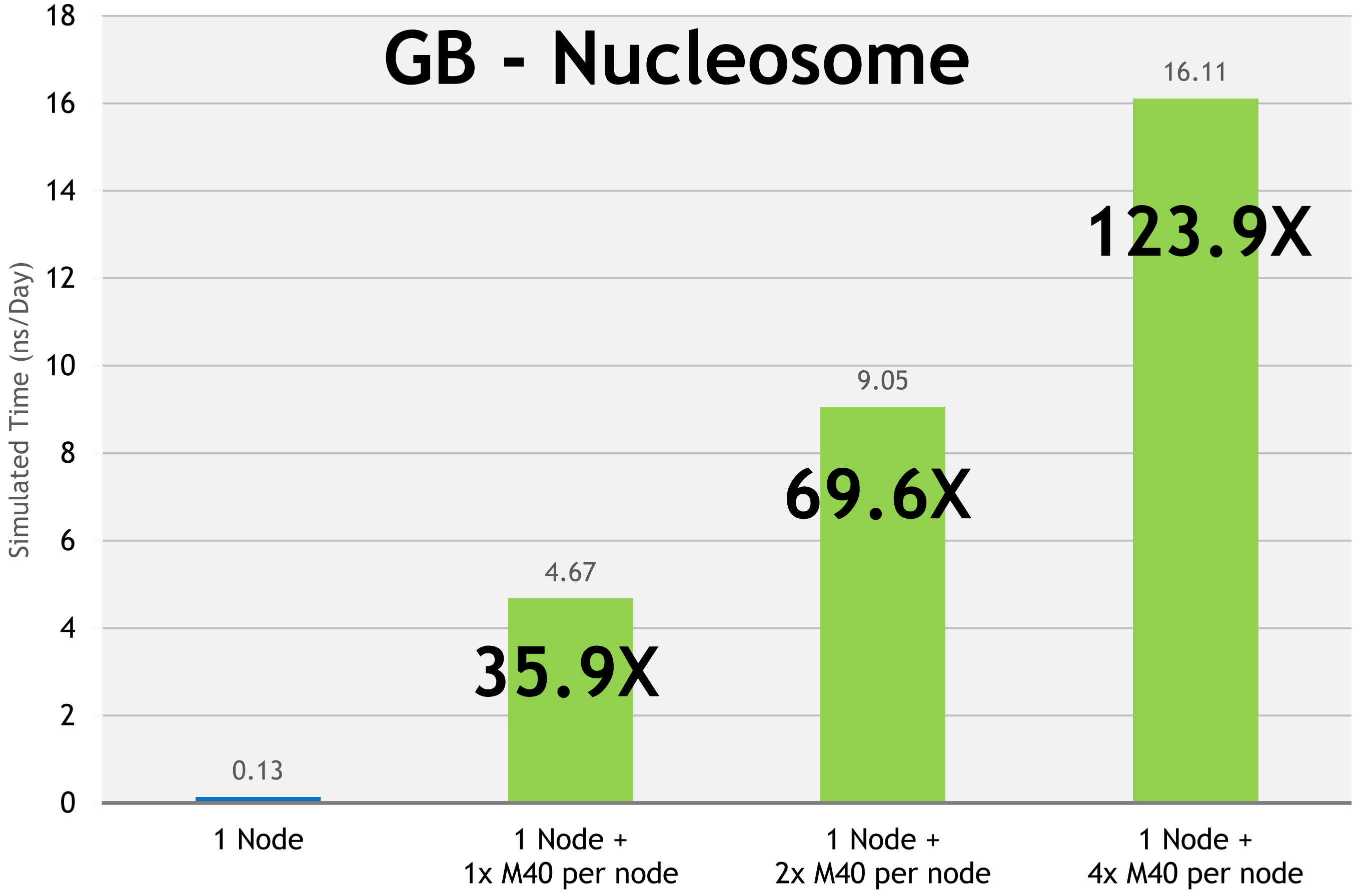
Running AMBER version 14

The blue node contain Single Intel Xeon E5-2698 v3@2.30GHz (Haswell) CPUs

The green nodes contain Single Intel Xeon E5-2697 v2@2.70GHz (IvyBridge) CPUs + Tesla M40 (autoboost) GPUs



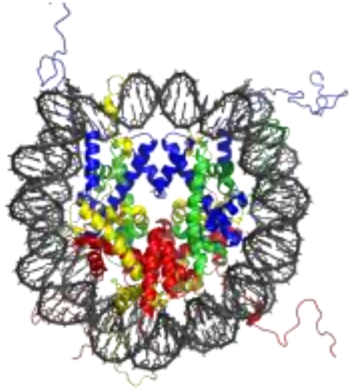
Nucleosome on M40s



Running AMBER version 14

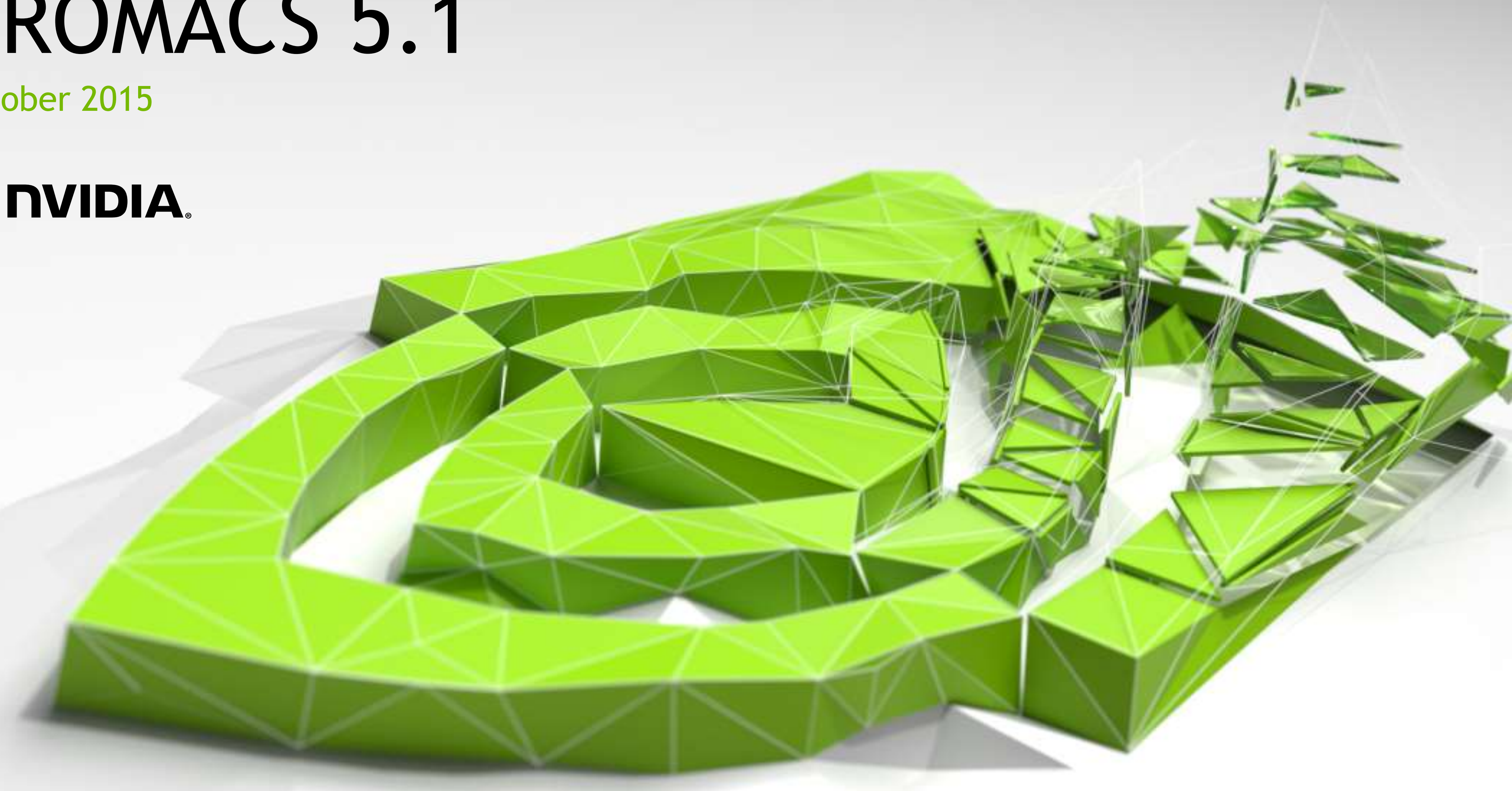
The blue node contain Single Intel Xeon E5-2698 v3@2.30GHz (Haswell) CPUs

The green nodes contain Single Intel Xeon E5-2697 v2@2.70GHz (IvyBridge) CPUs + Tesla M40 (autoboost) GPUs

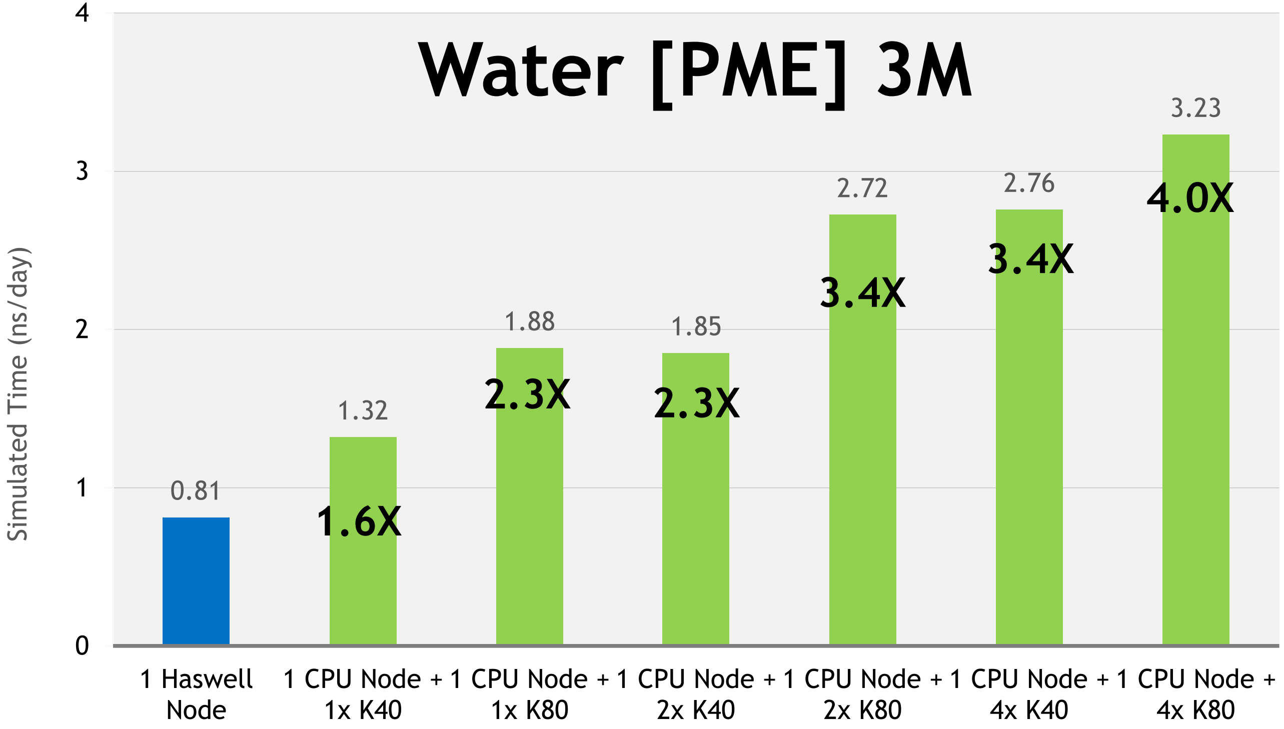


GROMACS 5.1

October 2015



3M Waters on K40s and K80s

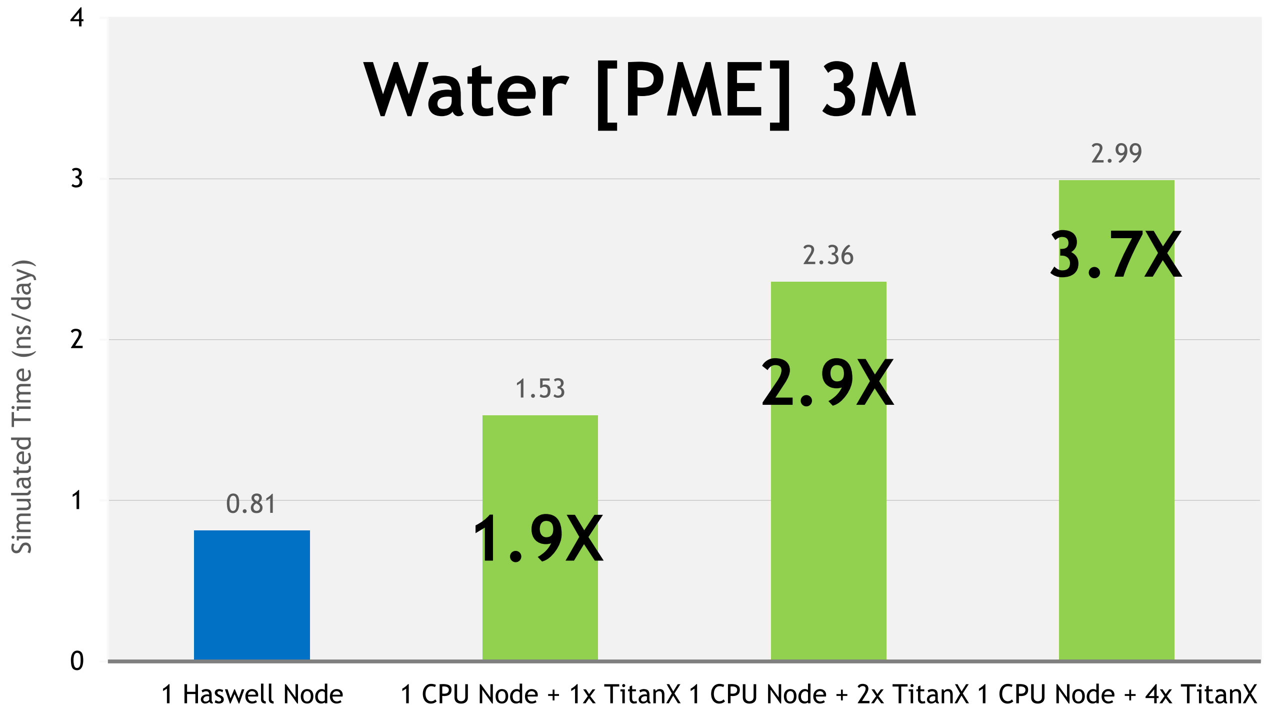


Running GROMACS version 5.1

The blue node contains Dual Intel E5-2698 v3@2.3GHz CPUs

The green nodes contain Dual Intel E5-2698 v3@2.3GHz CPUs + either NVIDIA Tesla K40@875Mhz or Tesla K80@562Mhz (autoboost) GPUs

3M Waters on Titan X



Running GROMACS version 5.1

The blue node contains Dual Intel E5-2698 v3@2.3GHz CPUs

The green nodes contain Dual Intel E5-2698 v3@2.3GHz CPUs + GeForce GTX TitanX@1000Mhz GPUs

TESLA P100 ACCELERATORS

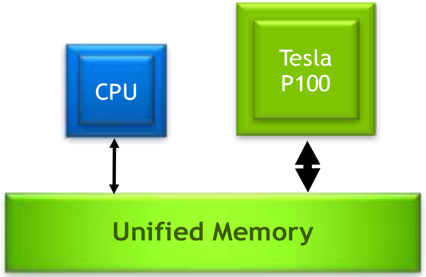
Tesla P100
for NVLink-enabled Servers

5.3 TF DP · 10.6 TF SP · 21 TF HP
720 GB/sec Memory Bandwidth, 16 GB

Tesla P100
for PCIe-Based Servers

4.7 TF DP · 9.3 TF SP · 18.7 TF HP
Config 1: 16 GB, 720 GB/sec
Config 2: 12 GB, 540 GB/sec

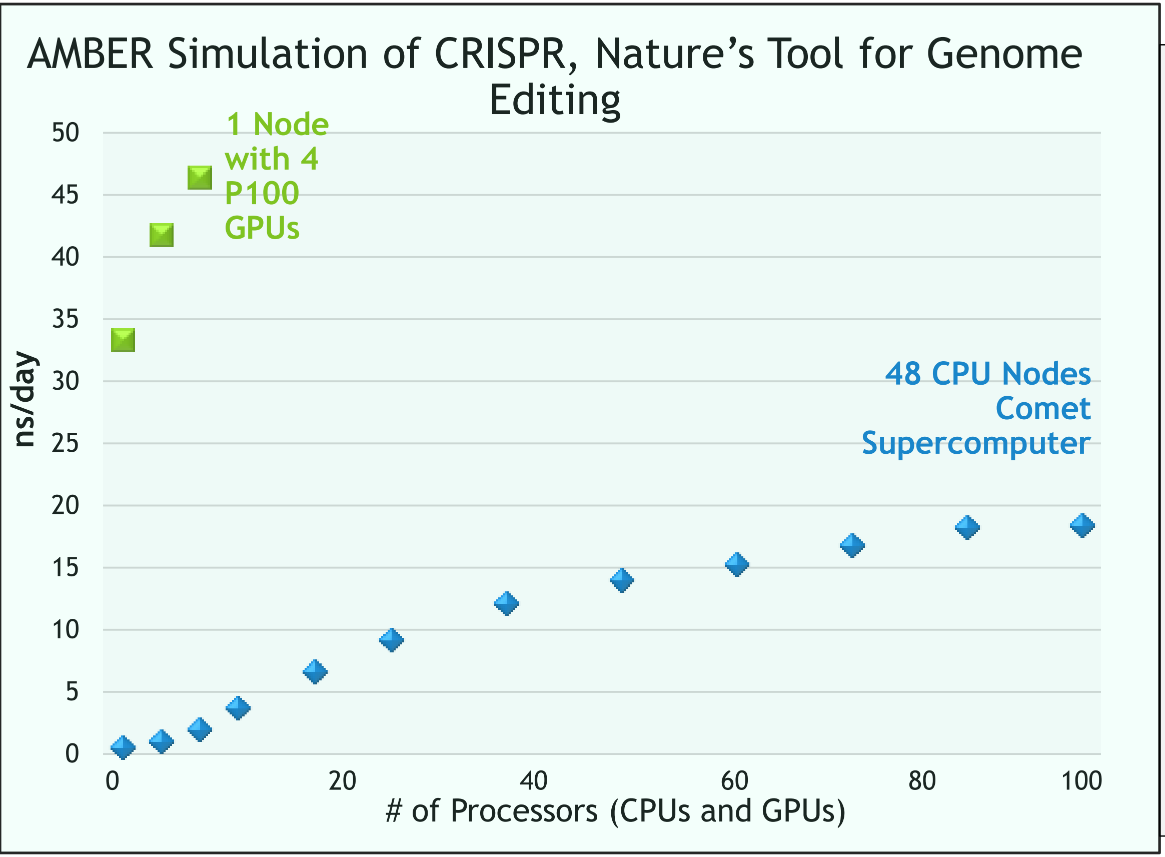
PAGE MIGRATION ENGINE

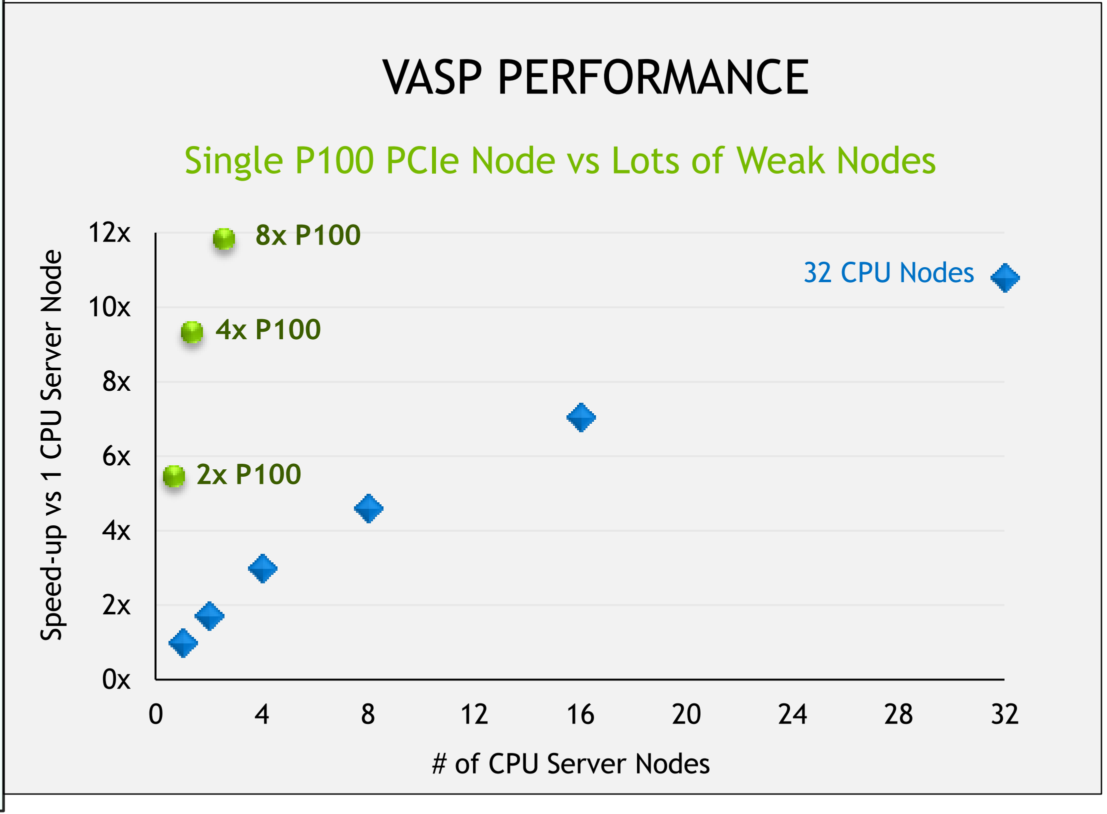


Simpler Parallel Programming
Virtually Unlimited Data Size
Performance w/ data locality

EXTRAORDINARY STRONG SCALING

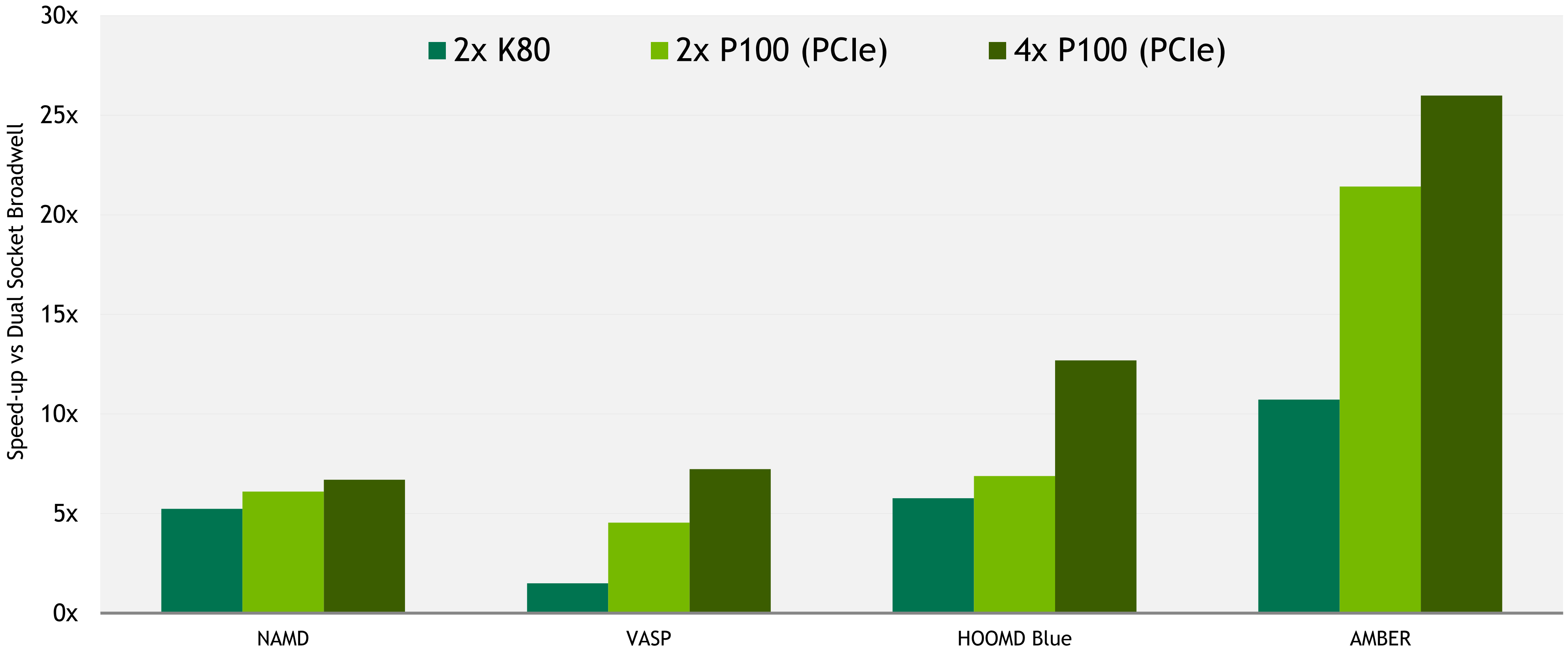
One Strong Node Faster Than Lots of Weak Nodes





CPU: Dual Socket Intel E5-2680v3 12 cores, 128 GB DDR4 per node, FDR IB
VASP 5.4.1_05Feb16, Si-Huge Dataset. 16, 32 Nodes are estimated based on same scaling from 4 to 8 nodes
AMBER 16 Pre-release, CRSPR based on PDB ID 5f9r, 336,898 atoms

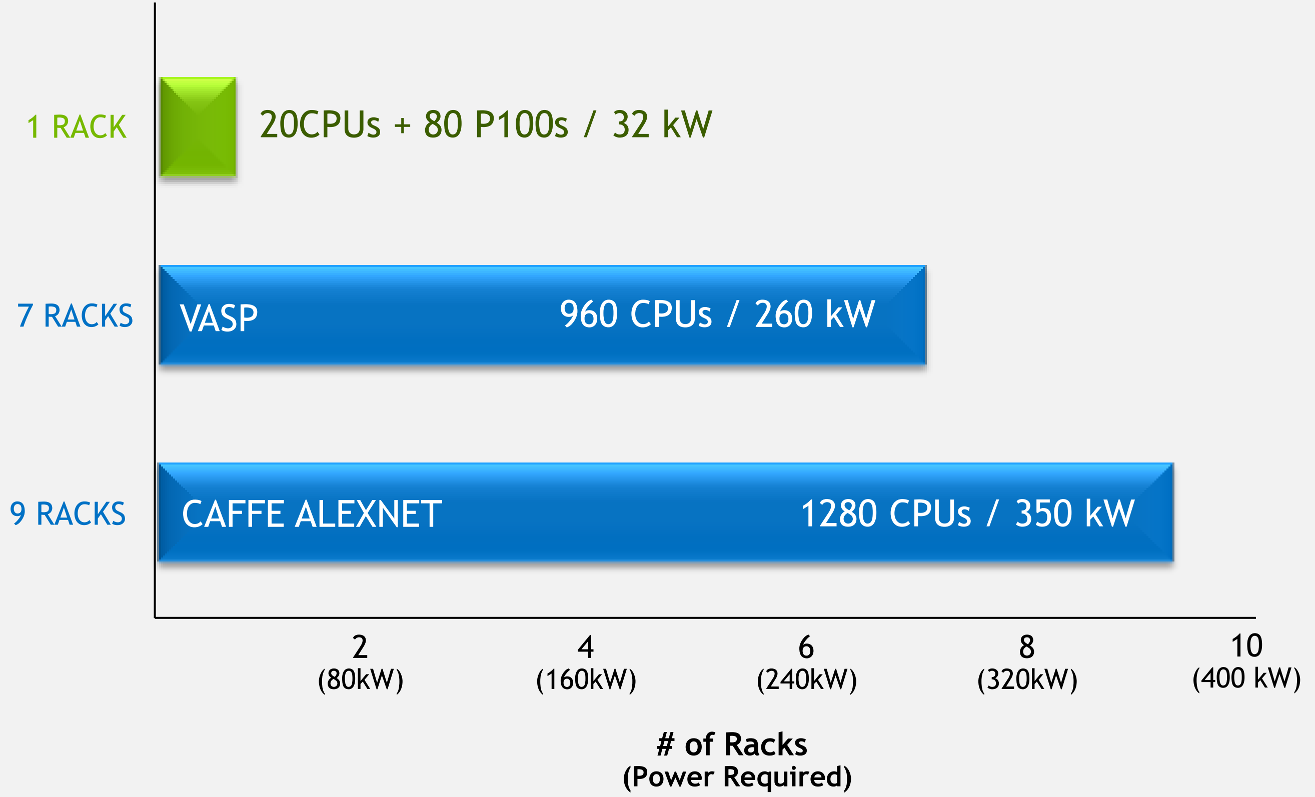
MASSIVE LEAP IN PERFORMANCE



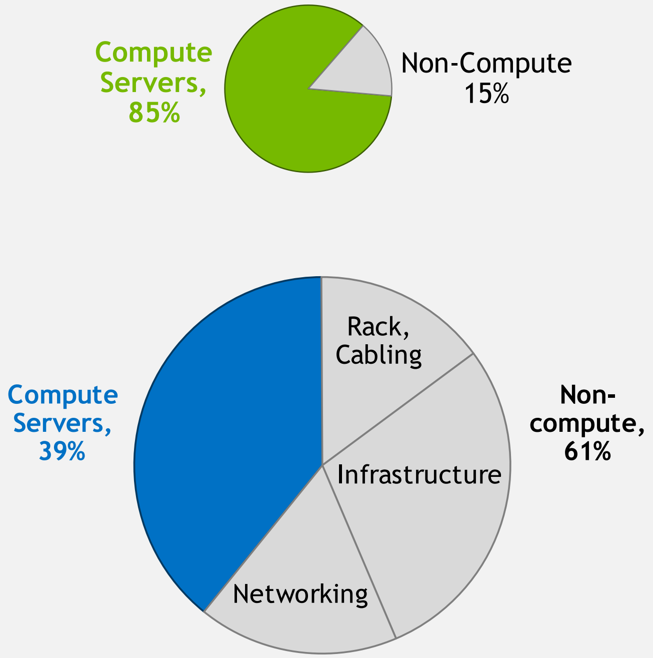
CPU: Xeon E5-2697v4, 2.3 GHz, 3.6 GHz Turbo
Caffe Alexnet: batch size of 256 Images; VASP, NAMD, HOOMD-Blue, and AMBER average speedup across a basket of tests

TCO OVERVIEW

EQUAL PERFORMANCE IN LESS RACKS



BUDGET: SMALLER, EFFICIENT



Sources: Microsoft Research on Datacenter Costs

NVIDIA DGX-1

WORLD'S FIRST DEEP LEARNING SUPERCOMPUTER

42.5 TFLOPS FP64/85 TFLOPS FP32/170 TFLOPS FP16

8x Tesla P100 16GB

NVLink Hybrid Cube Mesh

Accelerates Major AI Frameworks

Dual Xeon

7 TB SSD Deep Learning Cache

Dual 10GbE, Quad IB 100Gb

3RU – 3200W

THANK YOU!

Q&A: WHAT CAN I HELP YOU BUILD?