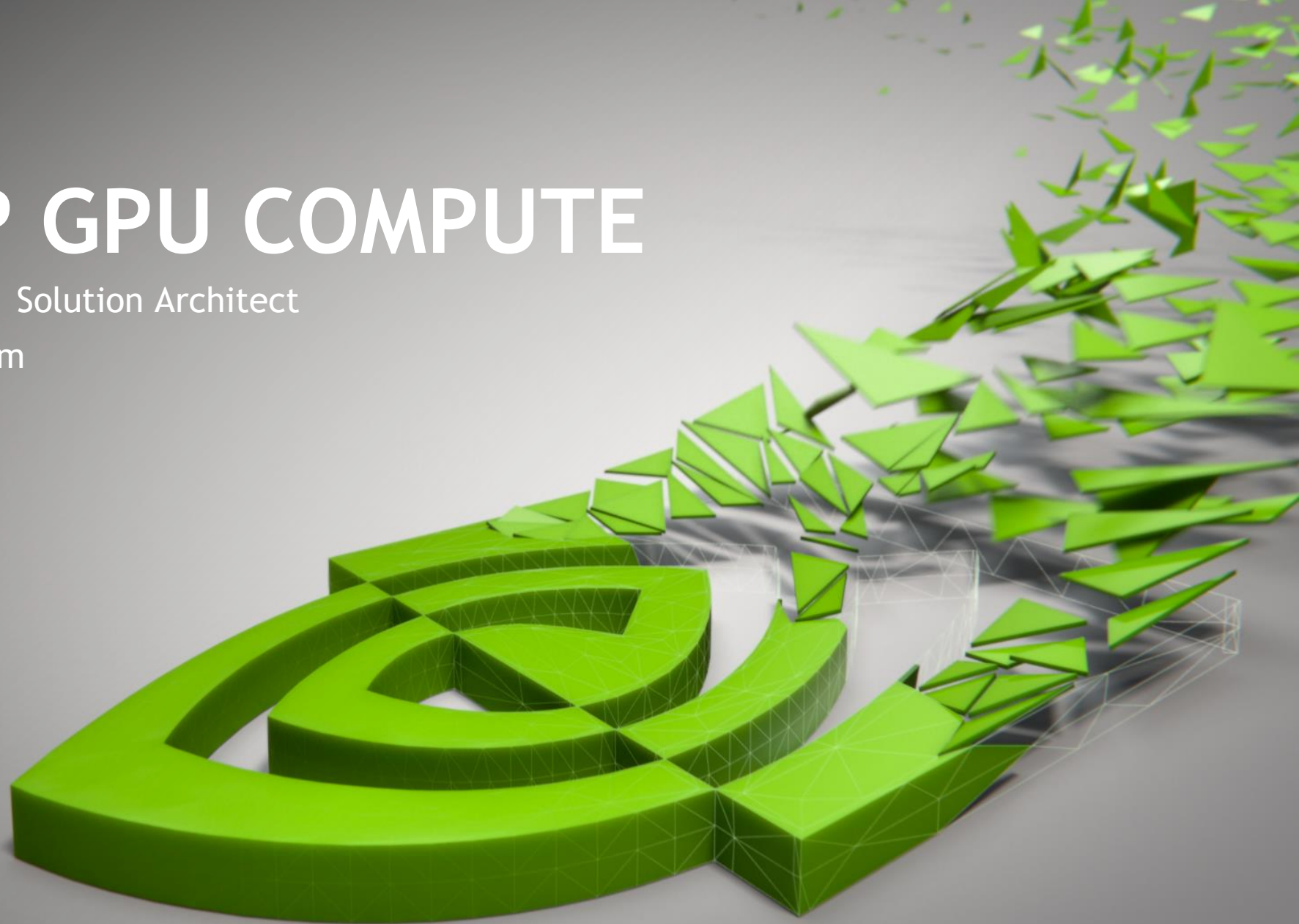


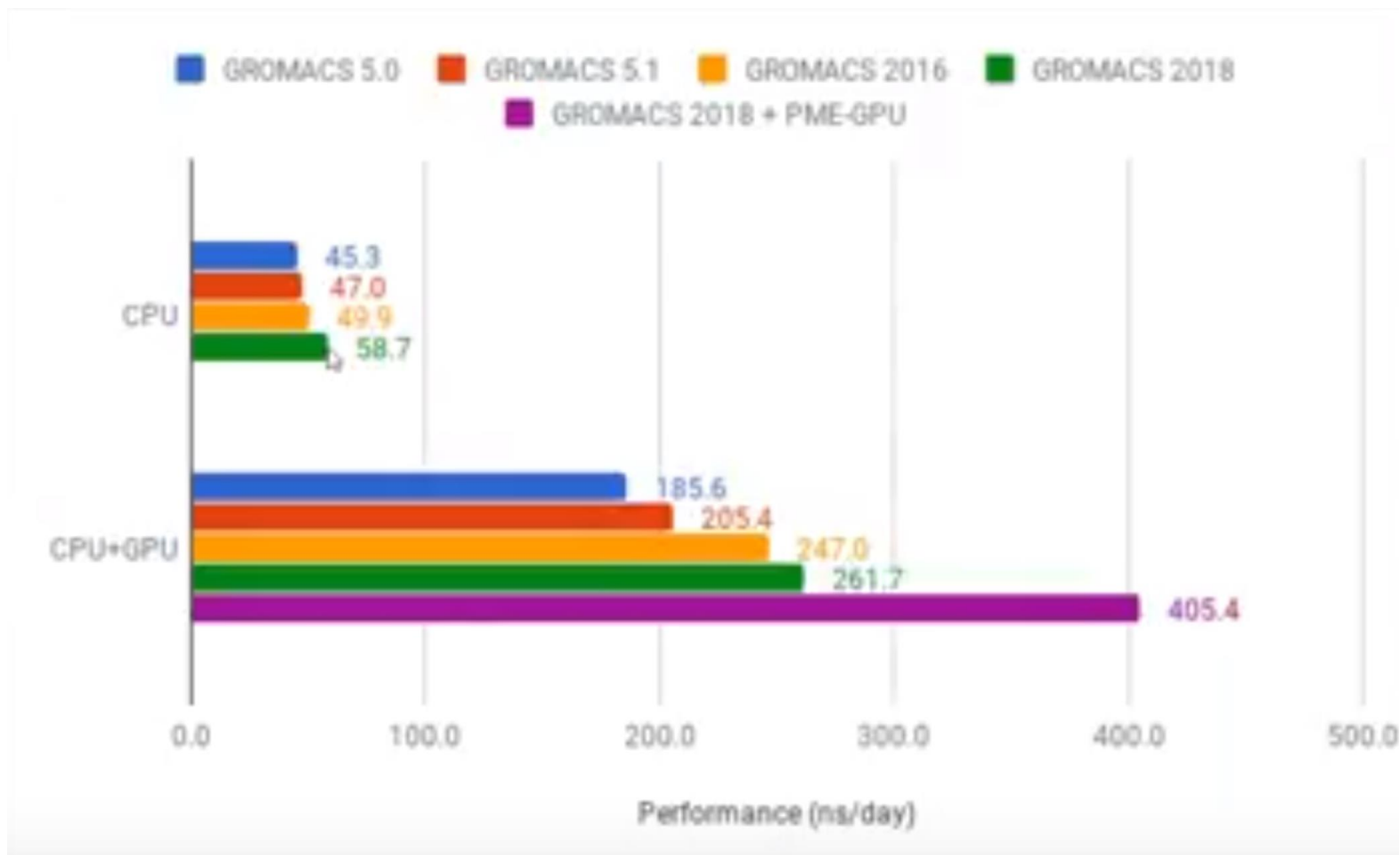
MD APP GPU COMPUTE

Cheng Yi (易成), Solution Architect

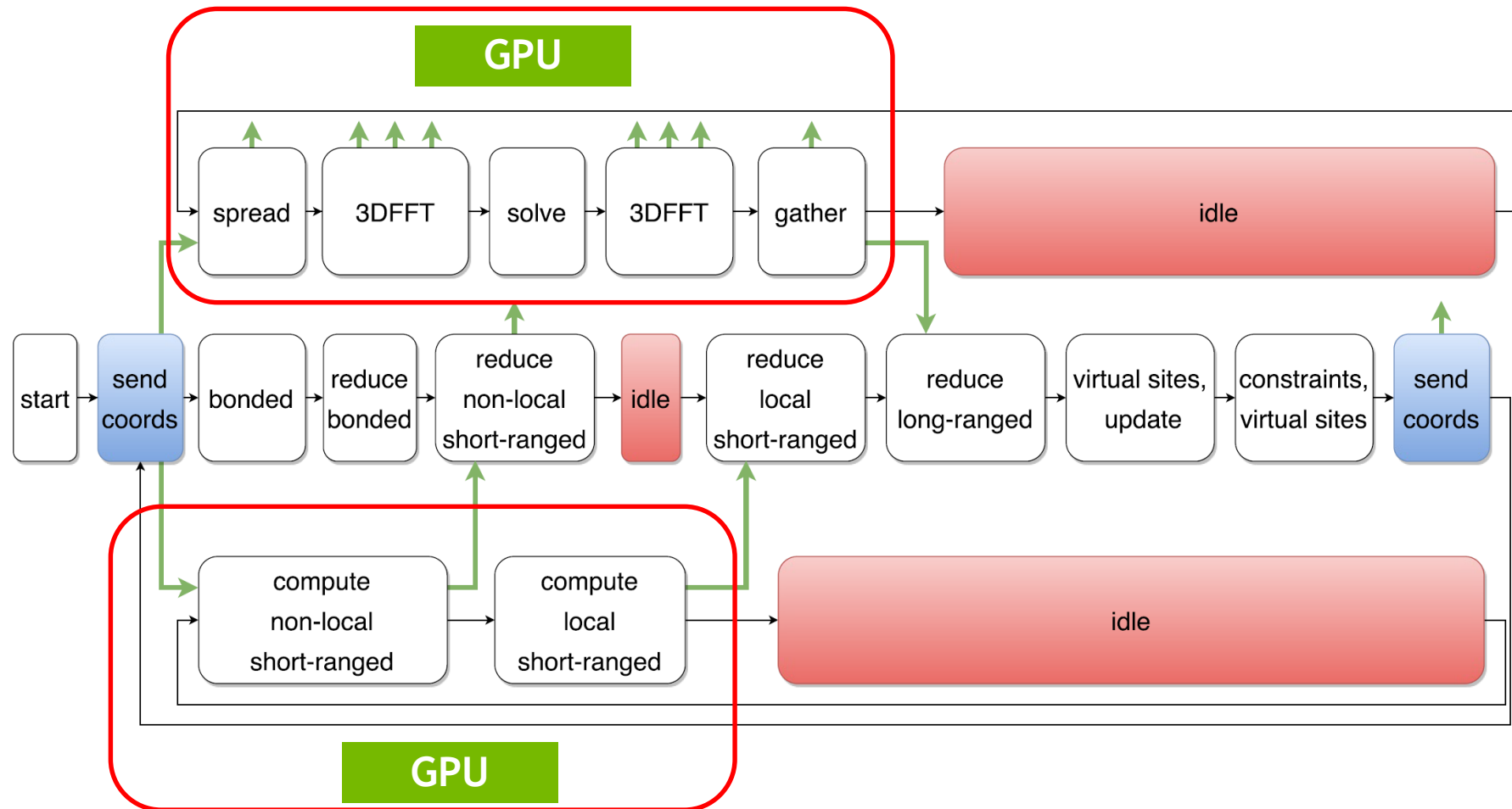
chengyi@nvidia.com



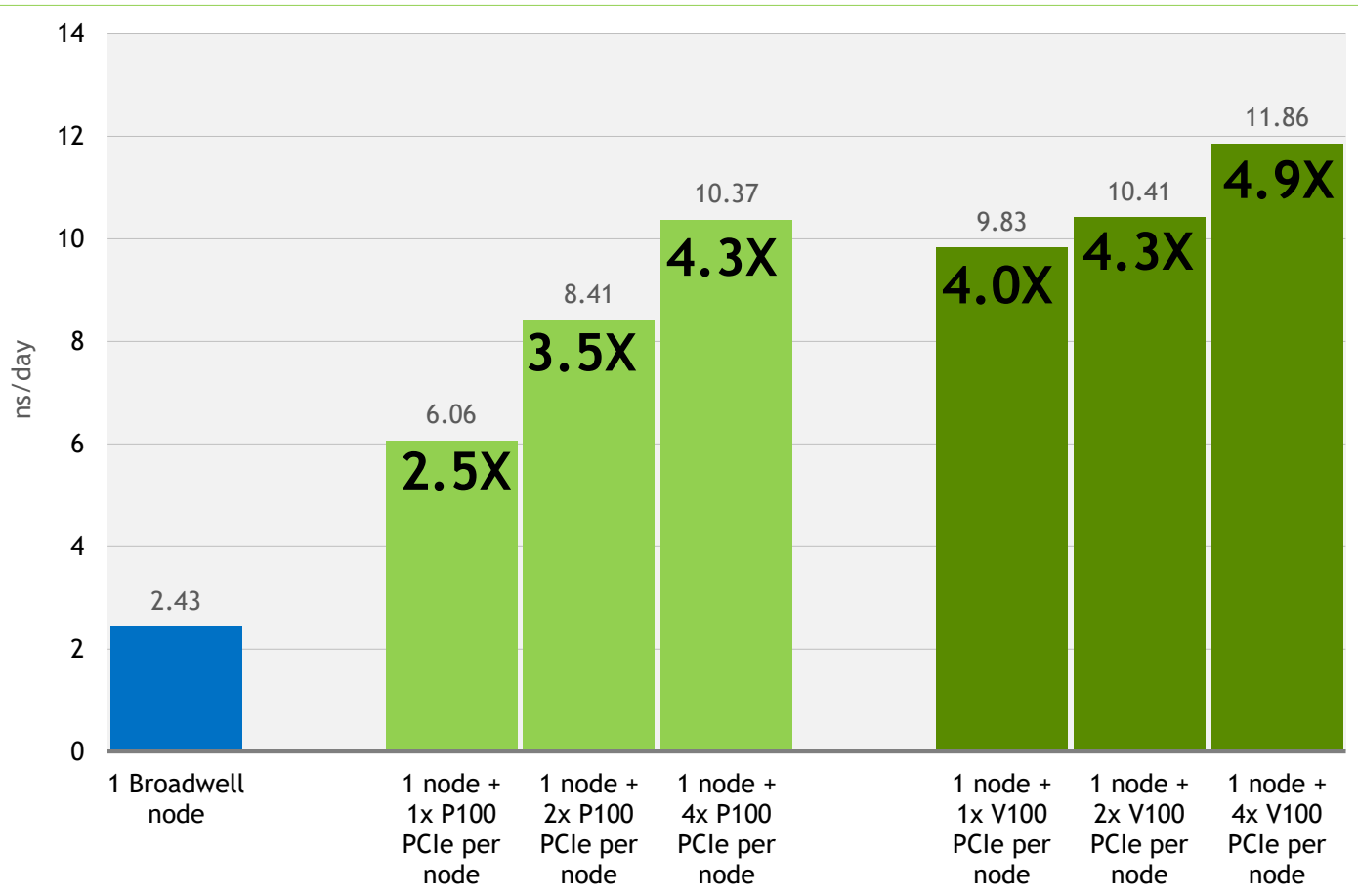
GROMACS PERFORMANCE TREND



GROMACS MULTIPLE RANKS, SEPARATE PME RANKS, AND WITH GPU



GROMACS WATER 1.5M ON V100 VS P100 (PCIE 16GB)



(Untuned on Volta)
Running **GROMACS** version 2018

The **blue node** contains Dual Intel Xeon E5-2690 v4@2.6GHz [3.5GHz Turbo] (Broadwell) CPUs

The **green nodes** contain Dual Intel Xeon E5-2690 v4@2.6GHz [3.5GHz Turbo] (Broadwell) CPUs + Tesla P100 PCIe (16GB) or V100 PCIe (16GB) GPUs

GROMACS的安装和运行

安装方法：

- 源码编译安装
- 使用NGC容器镜像，免安装，直接使用

最新版本：

- GROMACS version: 2018
- PME on GPU
- 基于CUDA编程，GPU加速好

Repositories

nvcr.io

- caffe
- caffe2
- cntk
- cuda
- digits
- mxnet
- pytorch
- tensorflow
- tensorrt
- theano
- torch

hpc

- candle
- gamess
- gromacs**
- lammps
- lattice-microbes
- namd
- relicon
- vmd

nvcr.io/hpc/gromacs

```
docker pull nvcr.io/hpc/gromacs:2016.4
```

```
nvidia-docker run -ti --rm -v $(pwd):/results nvcr.io/hpc/gromacs:2016.4 "/workspace/run_cont_adh.sh"
```

Note you could also point the CLI command to your local directory instead and run your own scripts (xxx.sh for example). The script below starts the gromacs container and runs the xxx.sh script from your results directory.

```
nvidia-docker run -ti --rm -v $(pwd):/results -it --rm nvcr.io/hpc/gromacs:2016.4 "/results/xxx.sh"
```

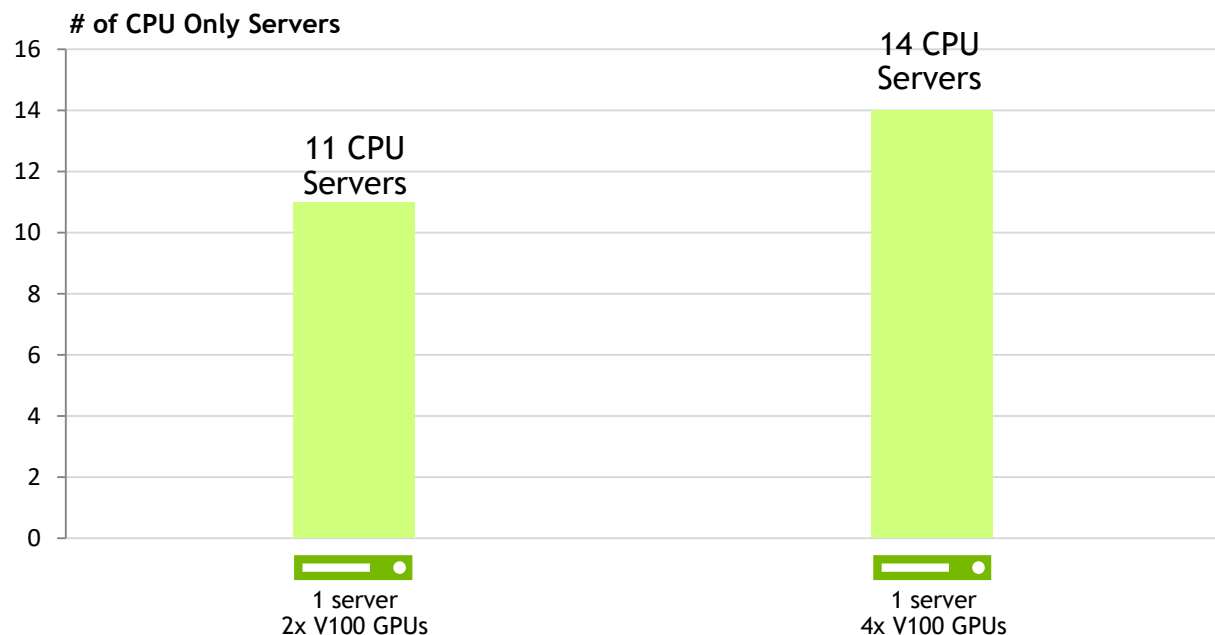
1.2 Running GROMACS Interactively

In this example, we are running the adh_cubic model benchmark again while inside the /workspace directory in the container. Running interactively is useful for making multiple GROMACS containers run within the same OS image. To run the GROMACS container interactively, issue the following command which starts the container and also mounts your current directory to /results so it is available inside the container. (see the -v options on the

TAG	SIZE	USER	LAST MODIFIED	PULL
2016.4	1.15 GB		November 14, 2017	↓

NAMD Performance Equivalency

Single GPU Server vs Multiple Skylake CPU-Only Servers



Speed up vs
CPU server

9x

12x

CPU Server: Dual Xeon Gold 6140@2.30GHz, GPU Servers: same CPU server w/ V100 PCIe

CUDA Version: CUDA 9.0.176, Dataset: STMV

To arrive at CPU node equivalency, we use measured benchmark with up to 8 CPU nodes. Then we use linear scaling to scale beyond 8 nodes.

NAMD

Geoscience (Oil & Gas)

Designed for high-performance simulation of large molecular systems

VERSION

2.13

ACCELERATED FEATURES

Full electrostatics with PME and most simulation features

SCALABILITY

Up to 100M atom capable, multi-GPU, Scale Scales to 2xP100

More Information

<http://www.ks.uiuc.edu/Research/namd/>

NAMD的新特性大都与GPU相关

NIH CENTER FOR MACROMOLECULAR MODELING & BIOINFORMATICS | UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Type Keywords

SEARCH

THEORETICAL *and* COMPUTATIONAL
BIOPHYSICS GROUP



Home Research Publications Software Instruction News Galleries Facilities About Us

Home

Overview

Publications

Research

Software

- ▶ NAMD
- ▶ VMD
- ▶ GPU Computing
- ▶ Lattice Microbes
- ▶ Atomic Resolution Brownian Dynamics
- ▶ MDFF
- ▶ QwikMD
- ▶ Other

Outreach

NAMD 2.13 New Features

Ready for Testing in NAMD 2.13b1

The following features have been released in NAMD 2.13. Any bug fixes will appear in the nightly build version on the [download site](#). Any documentation updates will appear in the [Nightly Build User's Guide \(online or 2.4M PDF\)](#) and [release notes](#).

GPU-accelerated simulations less limited by CPU performance

Contributed by Antti-Pekka Hynninen. Bonded forces and exclusions are now offloaded to the GPU by default. (May be disabled with "bondedCUDA 0". Note that this is a bit flag so default is "bondedCUDA 255"; setting "bondedCUDA 1" will offload only bonds, not angles, dihedrals, etc.) Removes all load-balanced work from the CPU, leaving only integration and optional features. The only benefit is for machines that were limited by CPU performance, so a high-end dual-socket workstation with an older GPU may see no benefit, while a single-socket desktop with the latest 1080 Ti will see the most.

Improved GPU-accelerated performance for non-orthogonal cells

The "useCUDA2" and "usePMECUDA" kernels introduced in NAMD 2.12 now also support non-orthogonal periodic cells.

Updated CUDA compatibility

Updated CUDA kernels for version 9.x compatibility, as required by Volta.

Replica exchange solute tempering, 2nd generation (REST2)

Enhanced sampling method that scales intramolecular potential energy of a protein to lower barriers separating different conformations. Flexible implementation that allows independent scaling controls over electrostatics, van der Waals, and bonded energy terms. Compatible with CUDA acceleration. Contributed by Wei Jiang and Jonathan Thirman.

Constant-pH MD

Makes use of nonequilibrium MD / Monte Carlo moves to enable constant-pH MD simulation of explicit solvent systems, with computational cost that increases linearly with the number of titratable groups. Contributed by Brian Radak.

Improvements to hybrid QM/MM simulation

Combine QM calculations for regions such as active sites of enzymes with fast classical calculation for the rest of the biomolecular system. Interfaces provided to QM packages MOPAC and ORCA. Improvements since 2.12 include simulation setup with QwikMD and visualization of molecular orbitals using VMD, plus various bug fixes to NAMD. Contributed by Marcelo Melo, Rafael Barardi, and Till Budack.

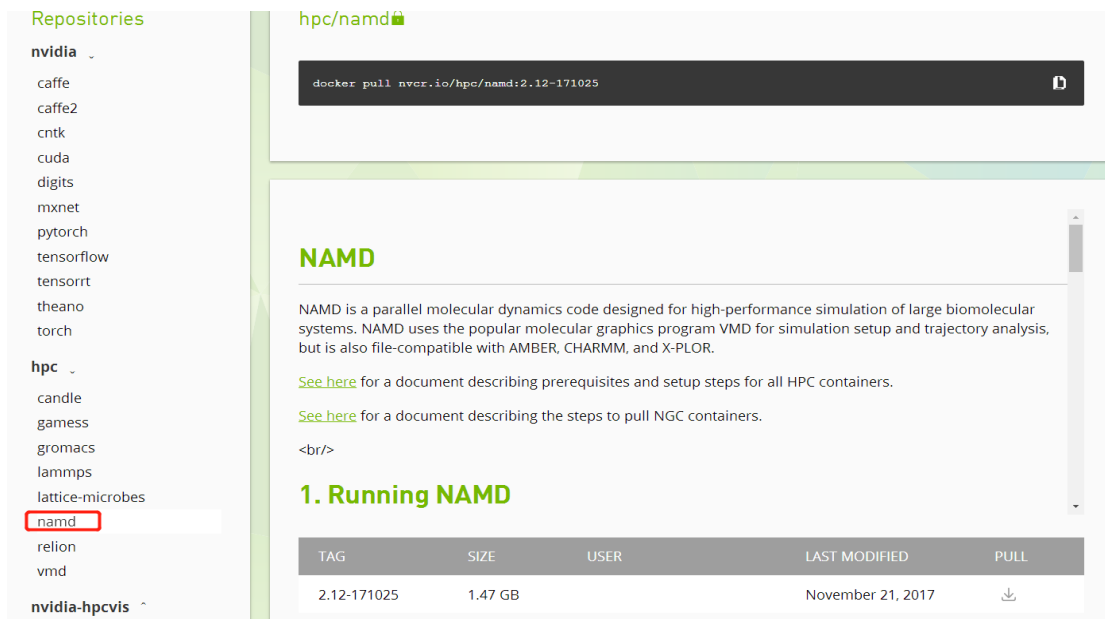
NAMD的安装和运行(非常适合GPU)

安装方法：

- 源码编译安装
- 使用NGC容器镜像，免安装，直接使用

最新版本：

- NAMD version: 2.13
- 基于CUDA编程，GPU加速好



The screenshot shows the NGC container registry interface. On the left, a sidebar lists repositories under 'nvidia' and 'hpc'. The 'namd' repository is highlighted with a red box. The main content area shows the 'hpc/namd' repository page, which includes a terminal snippet for pulling the image, a description of NAMD, and a table of available tags.

```
docker pull nvr.io/hpc/namd:2.12-171025
```

NAMD

NAMD is a parallel molecular dynamics code designed for high-performance simulation of large biomolecular systems. NAMD uses the popular molecular graphics program VMD for simulation setup and trajectory analysis, but is also file-compatible with AMBER, CHARMM, and X-PLOR.

[See here](#) for a document describing prerequisites and setup steps for all HPC containers.

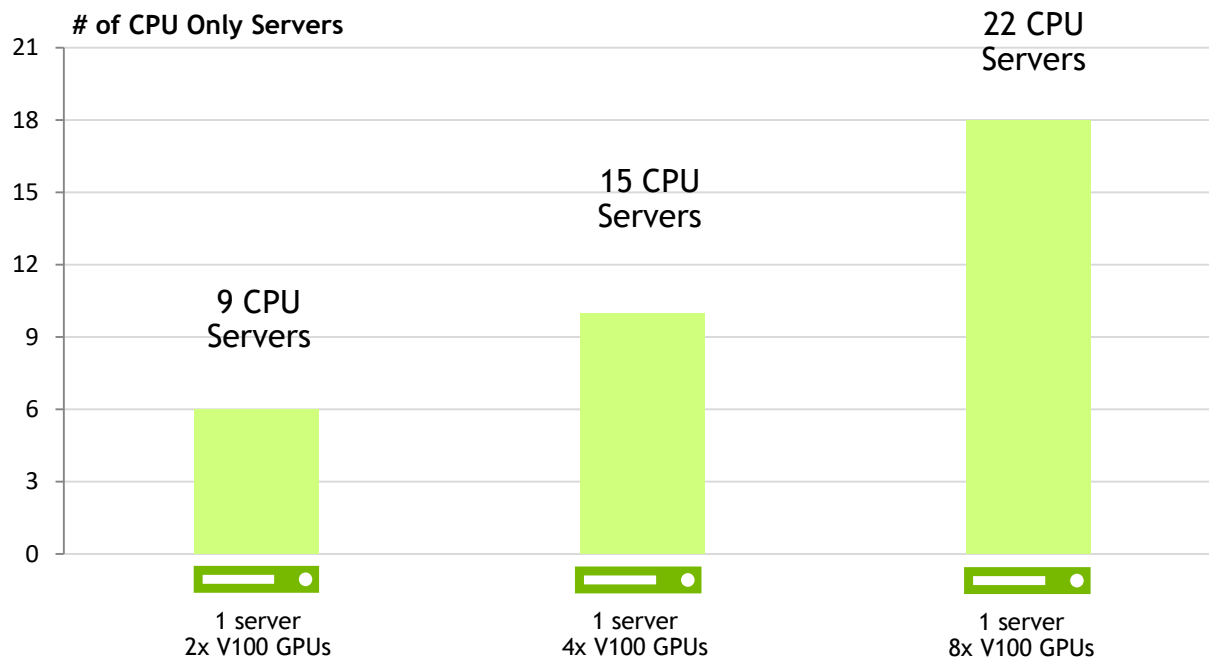
[See here](#) for a document describing the steps to pull NGC containers.

1. Running NAMD

TAG	SIZE	USER	LAST MODIFIED	PULL
2.12-171025	1.47 GB		November 21, 2017	↓

LAMMPS Performance Equivalency

Single GPU Server vs Multiple Skylake CPU-Only Servers



Speed up vs
CPU server

9x

15x

21x

LAMMPS

Molecular Dynamics

Classical molecular dynamics package

VERSION

2018

ACCELERATED FEATURES

Lennard-Jones, Gay-Berne, Tersoff, many more potentials

SCALABILITY

Multi-GPU and Multi-Node

More Information

<http://lammps.sandia.gov/index.html>

CPU Server: Dual Gold 6140@2.30GHz, GPU Servers: same CPU server w/ V100 PCIe or V100

CUDA Version: CUDA 9.0.176, Dataset: Atomic-Fluid Lennard-Jones 2.5 Cutoff

To arrive at CPU node equivalency, we use measured benchmark with up to 8 CPU nodes. Then we use linear scaling to scale beyond 8 nodes.

LAMMPS的安装和运行

安装方法：

- 源码编译安装
- 使用NGC容器镜像，免安装，直接使用

最新版本：

- 22 Aug 2018
- 基于CUDA编程，GPU加速好

The screenshot shows the NVIDIA NGC Registry interface. On the left, a sidebar lists repositories under the 'hpc' category, with 'lammps' highlighted in a red box. The main content area shows the 'hpc/lammps' repository page. At the top right, there is a green button labeled 'Get API Key'. Below the repository name, a terminal window displays the command `docker pull nver.io/hpc/lammps:patch23oct2017`, which is highlighted with a red box. Below the terminal, there are two links: [See here](#) for prerequisites and setup steps, and [See here](#) for steps to pull NGC containers. The main heading is '1. Running LAMMPS', followed by the text 'There are two options to run the LAMMPS container.' and a bulleted list of instructions.

Registry Get API Key

[Documentation](#)
How to use NGC containers on supported platforms >

Repositories

- ▼ nvidia
- ▼ nvidia/k8s
- ^ hpc
 - bigdft
 - candle
 - chroma
 - gamess
 - gromacs
 - lammps**
 - lattice-microbes
 - milc
 - namd
 - pgi-compilers
 - picongpu
 - relion

hpc/lammps

```
docker pull nver.io/hpc/lammps:patch23oct2017
```

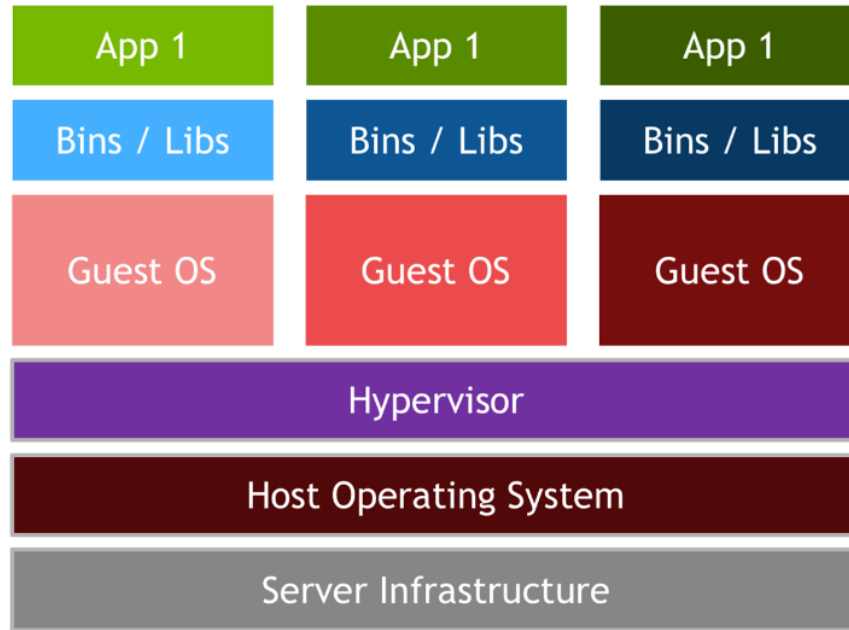
[See here](#) for a document describing prerequisites and setup steps for all HPC containers.
[See here](#) for a document describing the steps to pull NGC containers.

1. Running LAMMPS

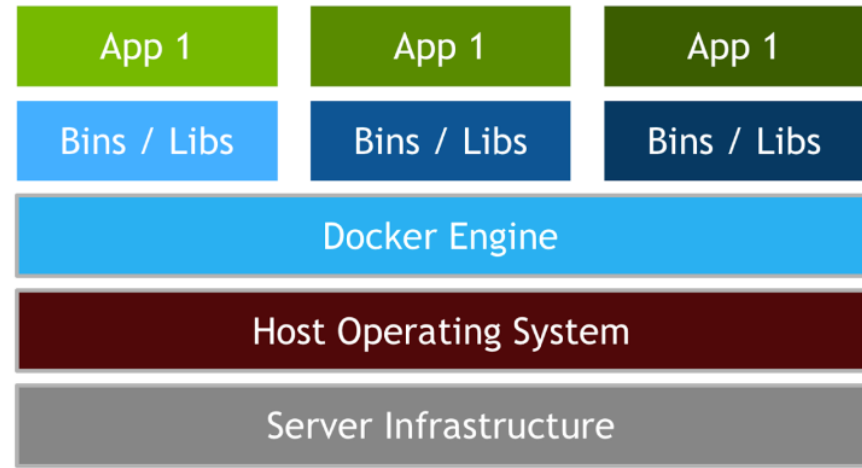
There are two options to run the LAMMPS container.

- You can run LAMMPS in detached mode from the `nvidia-docker run` command
- You can start the container in interactive mode and run LAMMPS interactively within the container

Docker VS VM



VIRTUAL MACHINES



CONTAINERS

Not so similar

GROMACS运行方法

1. NGC下载image镜像或编译源码安装。
2. 启动docker，在docker中运行。
3. Docker的目录映射，和主机的数据传输。
4. Docker的多节点并行

```
nvidia-docker run -it --name=gromacs --privileged --shm-size=1g --ulimit memlock=-1 --  
ulimit stack=67108864 --network=host --device=/dev/infiniband/uverbs0 -v  
/raid/cheng/./mnt -v /mnt/share/ssh:/root/.ssh nvcr.io/nvidia/gromacs:2016.4
```

BENCHMARKING

How to Start

- ▶ We choose one container as Primary worker, in which we start the distributed job with mpirun. And the rest containers are assumed as Secondary workers.
- ▶ Firstly, in Secondary workers, we start one sshd daemon with port 12345, and set the computing environment into infinity sleep state.

```
$ /usr/sbin/sshd -p 12345; sleep infinity
```

- ▶ Then, in Primary worker, we run like following:

```
$ mpirun -np 2 -x LD_LIBRARY_PATH -x PATH -host dgx1,dgx2 -mca plm_rsh_args "-p 12345" --allow-run-as-root hostname
```

```
dgx1  
dgx2
```

-mca plm_rsh_args “-p 12345”: Specifies arguments that will be added to ssh or rsh, here add “-p 12345” to ssh.



1.在command line里面加上-sf gpu，这个指令是把原来的input file转变成适合gpu的input file。然后-pk gpu #（package，gpu #Ngpu数量）

2.在input file里面的force filed后面加上 / gpu 具体可以查看lammps手册。

具体来说可以看一下lammps screen输出的各个部分计算的百分比，保证conmmunication低一点。一般我们组一个线程对应一个simulation（小体系，10000atom），也有24线程对应两个gpu跑的（大体系，gpu加速bond calculation）

开gpu是为了加速simulation其中一部分的计算，比如bond interaction，因为gpu上面核很多（2000cores），虽然做不了复杂的逻辑调度与高级计算，但是简单的数学运算还ok，simulation计算不复杂，所以远比cpu快很多，但不是说你gpu越多，一个task就越快的，这里面有很多因素影响，首先第一个问题是cpu和gpu，gpu和gpu的信息交换速度问题，lammps screen file里面跑完simulation有一个计算百分比的统计，信息交换也就是communication那一项，一般如果设置比列正确的话，在10左右，我之前做benchmark，有用过4cpu thread -2gpu，那一项占比50，然后我就用1cpu thread -1gpu发现竟然更快，这时communication占用10，所以你可以检查一下这一项；还有一个因素是你体系的大小，笼统来说，体系越大，gpu加速越明显，也就是说你的体系有好几万atom的时候，gpu越多越好，这个在lammps document gpu benchmark里面可以轻松得出结论，所以你体系不大，一个task用一个gpu足已，不过具体问题具体分析哈，

Supported features

- **Integrators:** *md/md-vv/md-vv-avek, sd/sd1* and *bd*. OpenMM implements only the velocity-verlet algorithm for MD simulations. Option *md* is accepted but keep in mind that the actual algorithm is not leap-frog. Thus all three options *md*, *md-vv* and *md-vv-avek* are equivalent. Similarly, options *sd* and *sd1* are also equivalent.
- **Long-range interactions:** *Reaction-Field, Ewald, PME, No-cutoff, Cut-off*.
 - for No-cutoff use *rcoulomb=0* and *rvdw=0*
 - for Ewald summation only 3D geometry is supported, while dipole correction is not.
 - the cut-off method is supported only for implicit solvent simulations.
- **Temperature control:** Supported only with the *sd/sd1, bd, md/md-vv/md-vv-avek* integrators. OpenMM implements only the Andersen thermostat. All values for *tcoupl* are thus accepted and equivalent to *andersen*. Multiple temperature coupling groups are not supported, only *tc-grps=System* will work.
- **Force Fields:** Supported FF are Amber, CHARMM. GROMOS and OPLS-AA are not supported.
 - CMAP dihedrals in CHARMM are not support, so use the *-nocmap* option with *pdb2gmx*.
- **Implicit solvent:** Supported only with reaction-field electrostatics. The only supported algorithm for GB is OBC, and the default Gromacs values for the scale factors are hardcoded in OpenMM, i.e. *obc alpha=1, obc beta=0.8* and *obc gamma=4.85*.
- **Constraints:** Constraints in OpenMM are done by a combination of SHAKE, SETTLE and CCMA. Accuracy is based on the SHAKE tolerance as set by the *shake_tol* option.
- **Periodic Boundary Conditions:** Only *pbcs=xyz* and *pbcs=no* in rectangular cells (boxes) are supported.
- **Pressure control:** OpenMM implements the Monte Carlo barostat. All values for *Pcoupl* are thus accepted.
- **Simulated annealing:** **Not supported.**
- **Pulling:** **Not supported.**
- **Restraints:** Distant, orientational, angle and dihedral restraints are not supported in the current implementation.
- **Free energy calculations:** **Not supported** in the current implementation.
- **Walls:** **Not supported.**
- **Non-equilibrium MD:** Option *acc_grps* is **not supported**.
- **Electric Fields:** **Not supported.**
- **QMMM:** **Not supported.**