High-Performance Computing in Atomistic Simulations

V.I. Yamakov[1], E.H. Glaessgen[2], Y. Mishin[3]

[1]National Institute of Aerospace, Hampton, VA 23666
[2]NASA Langley Research Center, Hampton, VA 23681
[3]George Mason University, Fairfax, VA 22030

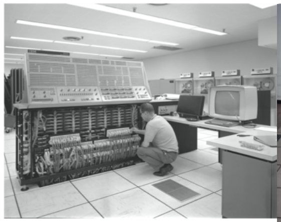
Outline

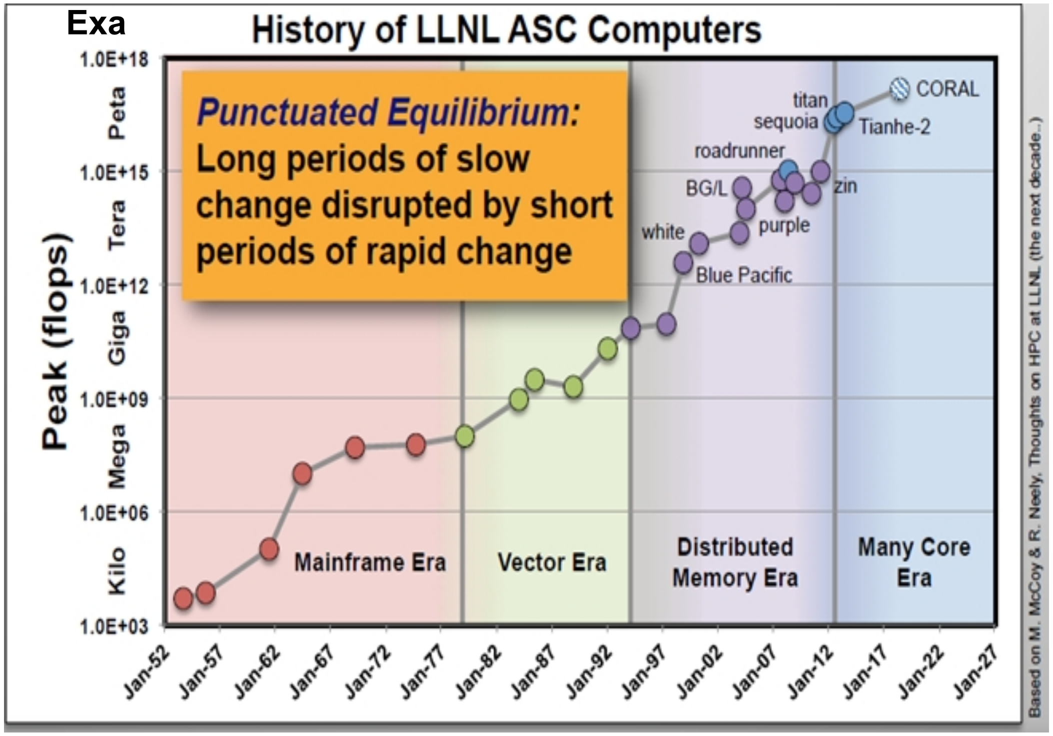
- Current advanced HPC hardware and levels of parallelism
- Current approaches and challenges in HPC for atomistic simulations
- Parallelization strategies in molecular dynamics
- Artificial neural networks in atomistic simulations
- HPCI lessons learned: Current achievements in the developed in-house software

Development Trends in Computing Technology

HPC NASA/LaRC

1960s	1970s	1980s	1990s	2000s	… 2017
PDP, IBM, Vax Series	ILLIAC IV, VPS-32, Convex, CDC Cyber	Cray Series, SGI, National Aerodynamic Simulator	Intel Paragons, IBM SPs	HyperCube, MasPar, CM-2, CM-5	SGI ICE Altix, HP Apollo 6000 + NVidia Kepler K40 GPU





Punctuated Equilibrium: Long periods of slow change disrupted by short periods of rapid change

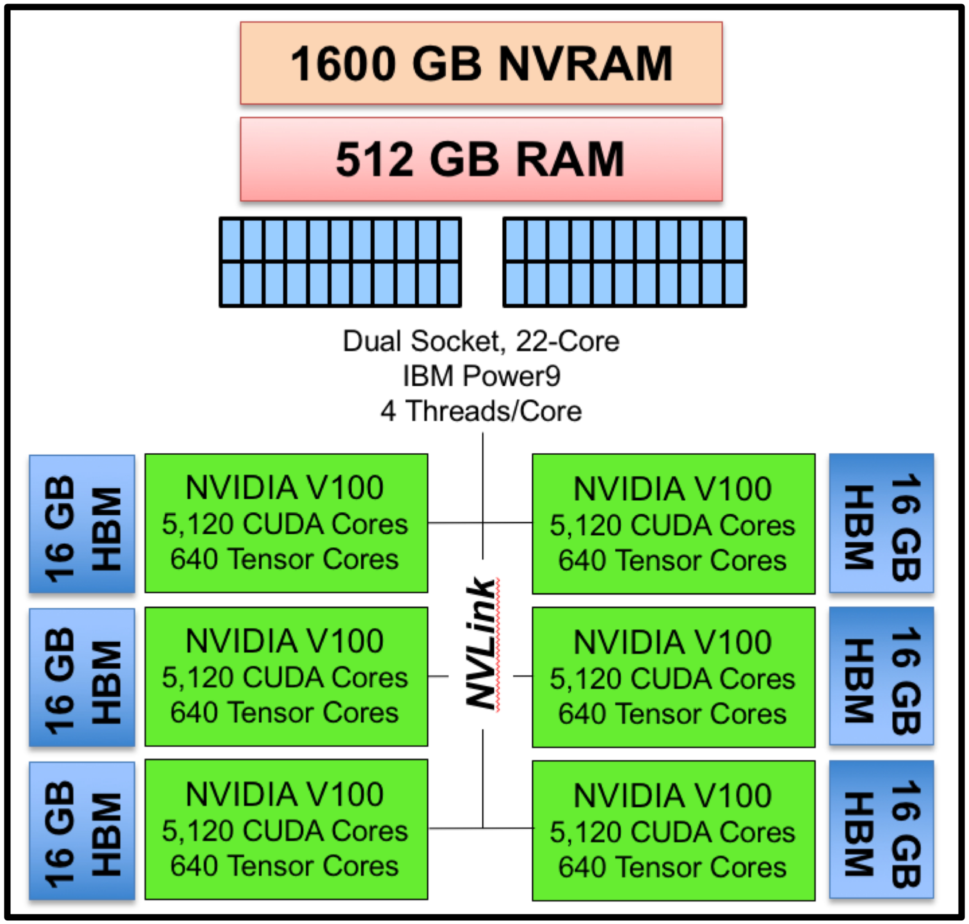
Based on M. McCoy & R. Neely, Thoughts on HPC at LLNL (the next decade..)

Every architectural paradigm shift requires a **substantial investment in dollars and workforce**. Updating skillsets and rewriting applications.

Current Advanced HPC Hardware

ORNL Summit Node *~40 TF*

Oak Ridge NL: Summit documentation for users



1600 GB NVRAM

512 GB RAM

Dual Socket, 22-Core
IBM Power9
4 Threads/Core

16 GB HBM

NVIDIA V100
5,120 CUDA Cores
640 Tensor Cores

NVLink

NVIDIA V100
5,120 CUDA Cores
640 Tensor Cores

16 GB HBM

16 GB HBM

NVIDIA V100
5,120 CUDA Cores
640 Tensor Cores

NVIDIA V100
5,120 CUDA Cores
640 Tensor Cores

16 GB HBM

16 GB HBM

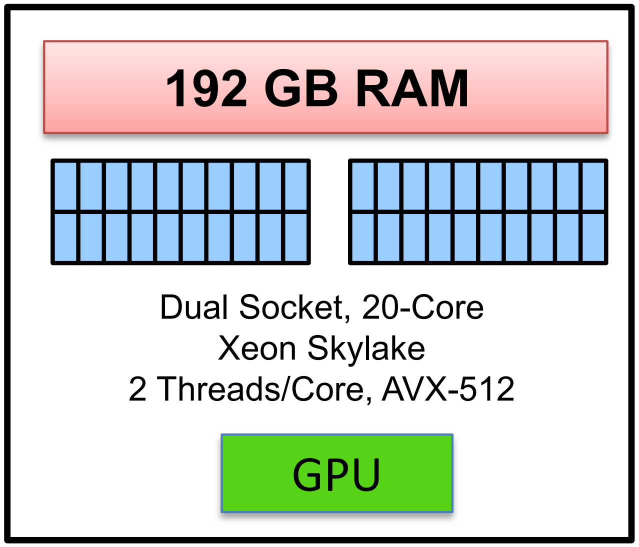
NVIDIA V100
5,120 CUDA Cores
640 Tensor Cores

NVIDIA V100
5,120 CUDA Cores
640 Tensor Cores

16 GB HBM

4,356 nodes *~122 Pflops*

NASA Advanced Supercomputing Division Electra Node
~1 TF



192 GB RAM

Dual Socket, 20-Core
Xeon Skylake
2 Threads/Core, AVX-512

GPU

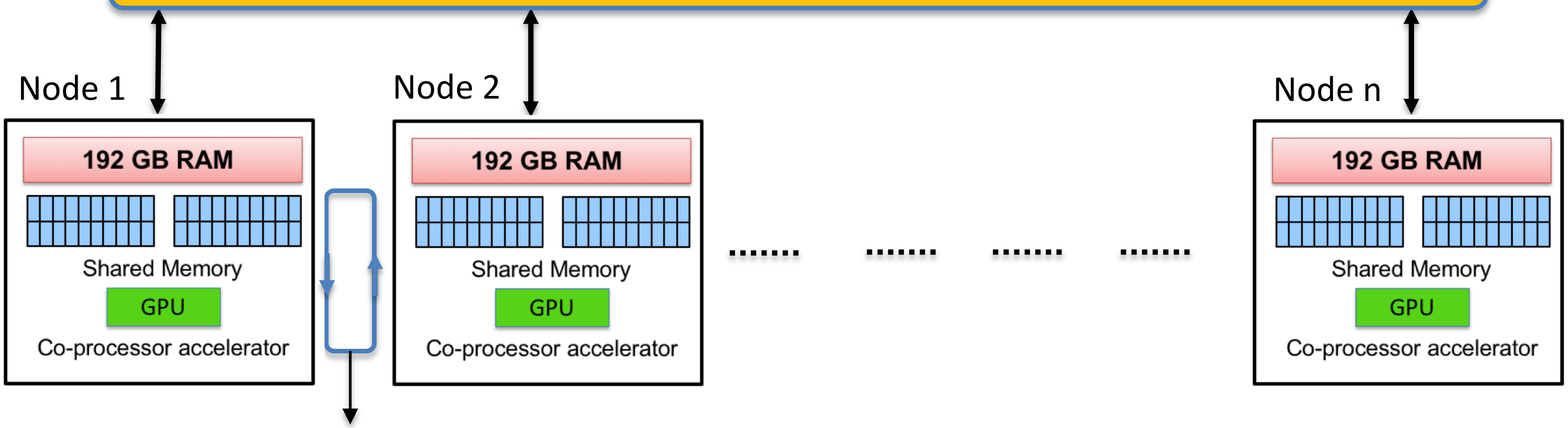
11,440 nodes
64 nodes with 184,320 NVIDIA GPU cores
~7.24 Pflops

Computing hardware of increasing complexity requires constant redevelopment of the software to utilize it efficiently.

Levels of Parallelism

Distributed Memory - Message Passing Interface (MPI)

Inter-node communication: slow, but scalable to unlimited number of nodes



Node 1

192 GB RAM

Shared Memory

GPU

Co-processor accelerator

Node 2

192 GB RAM

Shared Memory

GPU

Co-processor accelerator

....

Node n

192 GB RAM

Shared Memory

GPU

Co-processor accelerator

Intra-node communication: fast, but scalability is limited to the node's resources

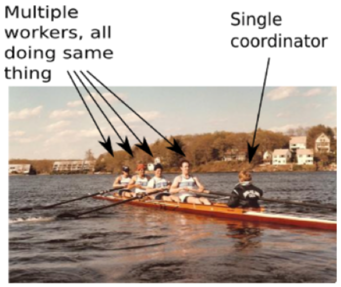
Hierarchy in hardware defines levels of parallelism

- **Inter-node parallelism (Multiple Instructions – Multiple Data):** independent tasks with little communication
- **Intra-node parallelism (MIMD + Single Instructions – Multiple Data):** multiple workers on the same tasks with intensive communication
- **GPU: multiple repetition of identical operations (SIMD: e.g., matrix algebra)**

MIMD (soccer)

Workers with same objective, doing different things

SIMD (rowing)



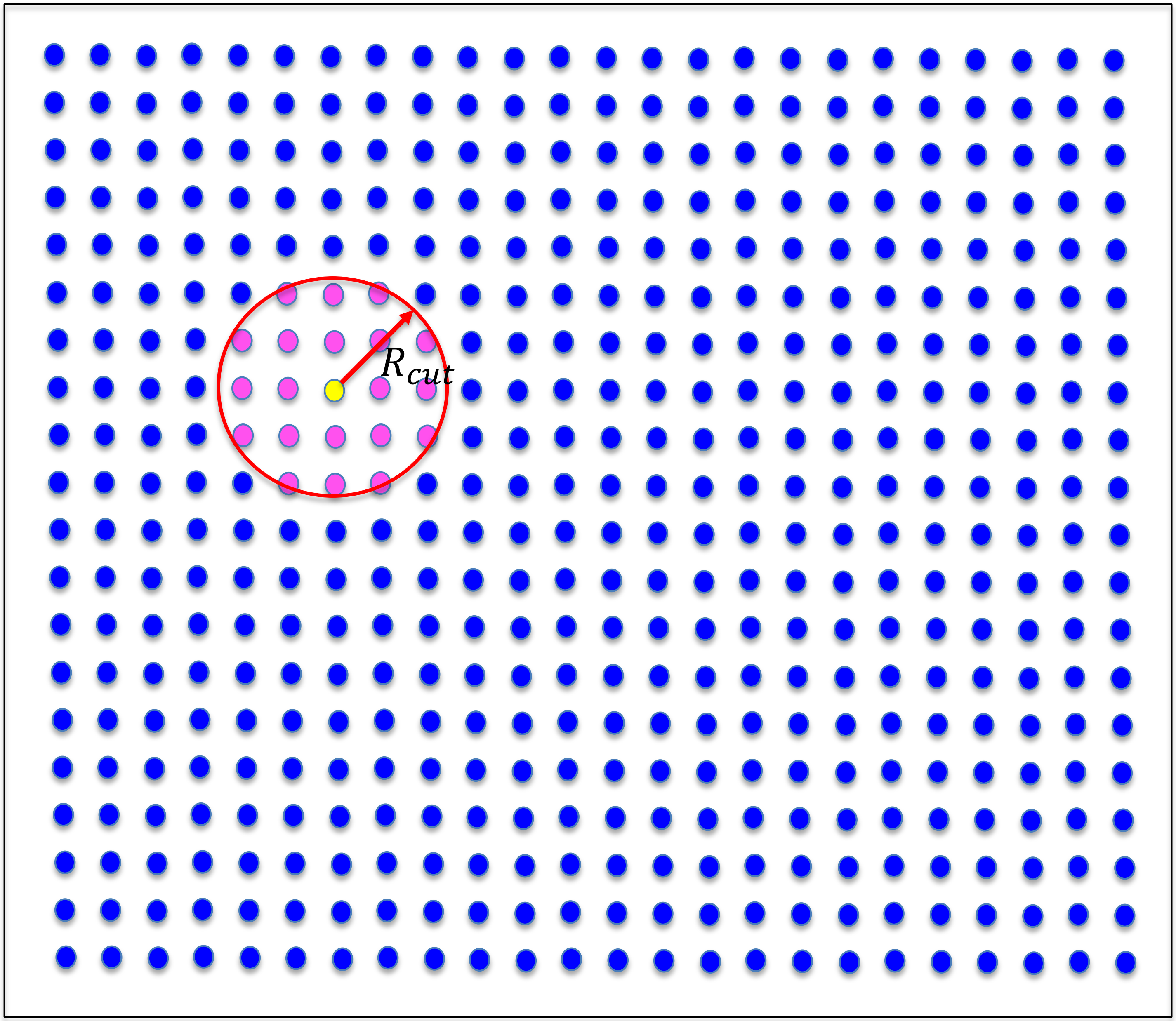
The software code must be structured to efficiently explore the different levels of parallelism in the current HPC hardware

Current Approaches and Challenges in HPC for Atomistic Simulations

HPCI
High Performance Computing Incubator

How can atomistic simulations efficiently explore the most current HPC architecture

Molecular Dynamics Algorithm



Evolve a system of a large number of atoms according to the classical Newtonian dynamics: a = F/m

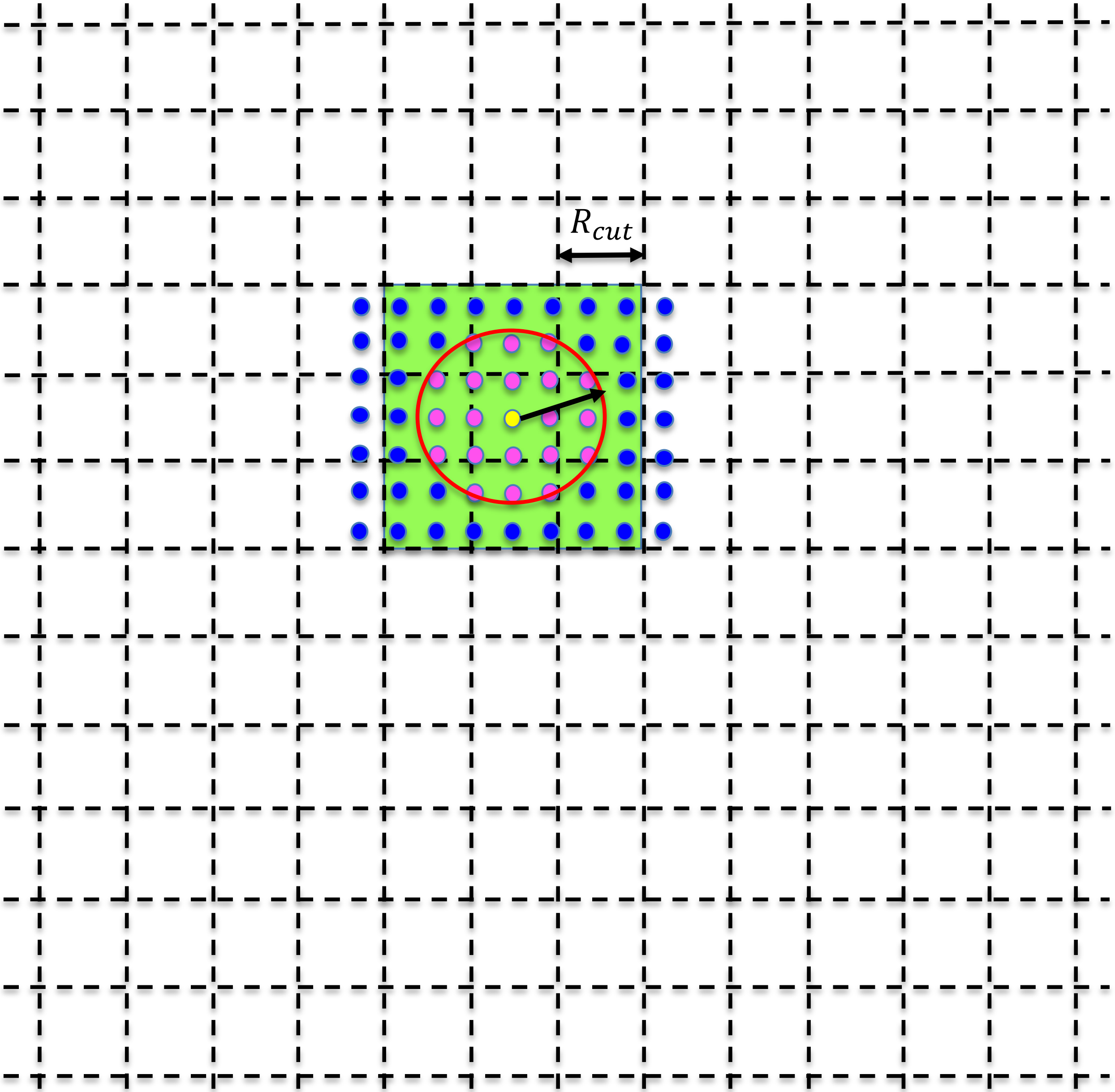
One MD step:

For each atom:

1. Identify interacting neighbors at $R < R_{cut}$
2. Calculate the force on each atom exerted by its neighbors
3. Integrate Newtonian equation of motion for each atom

Move all atoms and repeat

Link-Cell Technique
(Search for nearest neighbors)



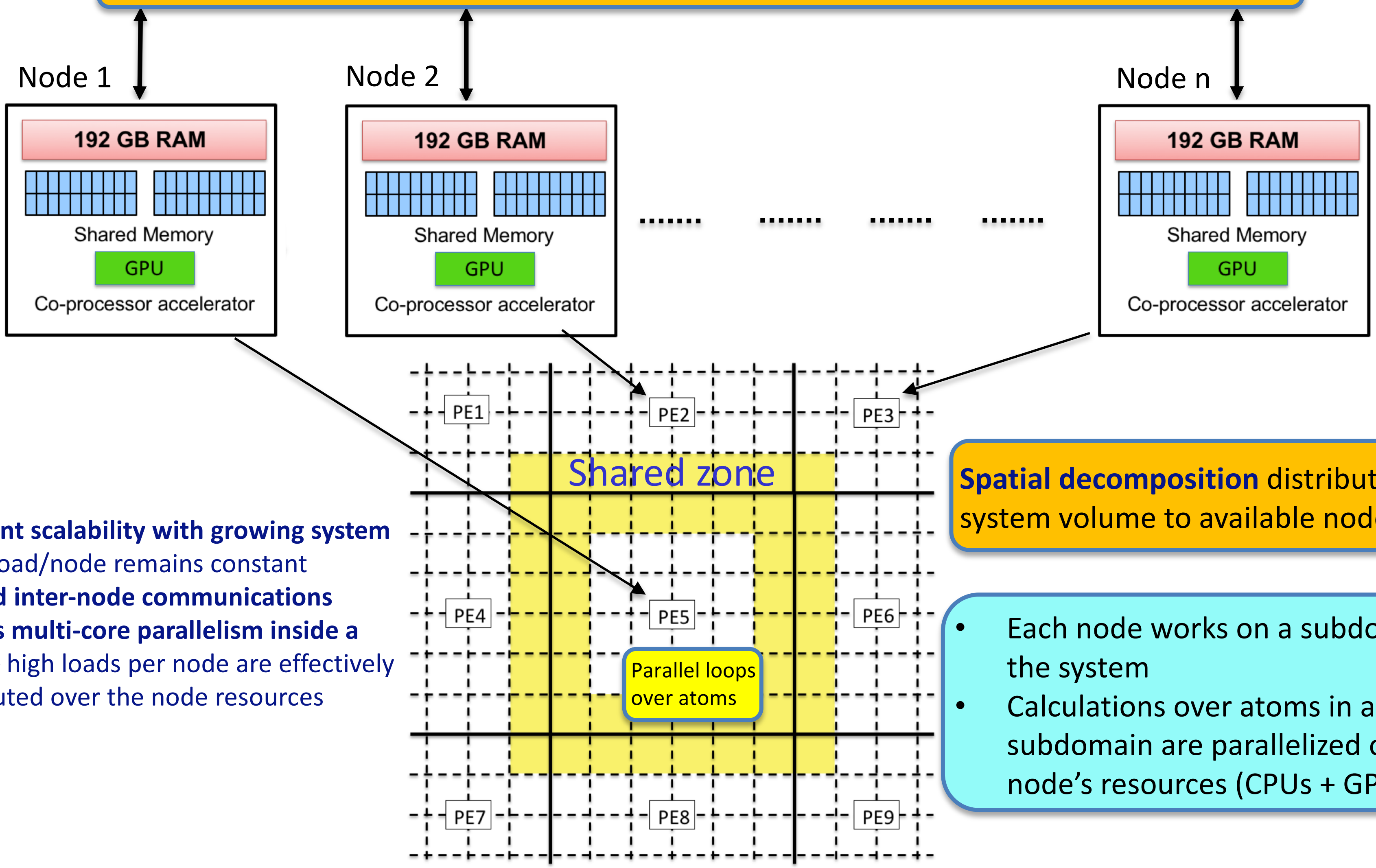
Divide the system box into cells of size $\geq R_{cut}$.
Limit the neighbor search up to the **nearest neighbor cells**.

Efficient for short range interactions, and allows for parallelization.

Parallelization Strategy in MD

Distributed Memory - Message Passing Interface (MPI)

Inter-node communication: slow, but scalable to unlimited number of nodes



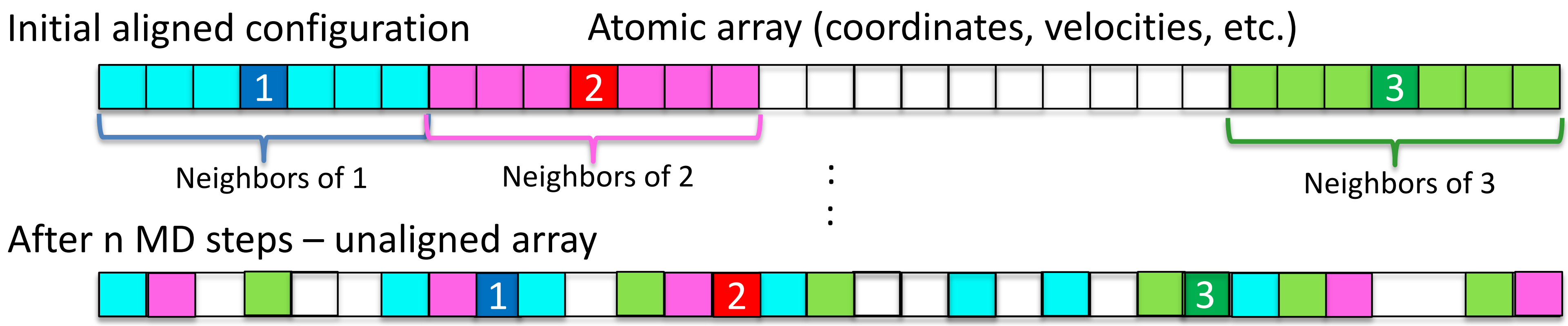
Spatial decomposition distributes system volume to available nodes

- **Excellent scalability with growing system size** – load/node remains constant
- **Limited inter-node communications**
- **Utilizes multi-core parallelism inside a node** – high loads per node are effectively distributed over the node resources

- Each node works on a subdomain of the system
- Calculations over atoms in a subdomain are parallelized over the node's resources (CPUs + GPUs)

High Performance Computing Incubator

Factors Impeding Scalability



Initial aligned configuration

Atomic array (coordinates, velocities, etc.)

Neighbors of 1

Neighbors of 2

Neighbors of 3

After n MD steps – unaligned array

- Need to update interacting neighbors after each MD step
- Unaligned access (realignment takes time)
- Low arithmetic intensity (mostly book keeping and data transfer)
- Complex functional forms (not easily scalable)

$$S_{ijk} = 1 - f_c\left(r_{ik} + r_{jk} - r_{ij}\right)e^{-\lambda_{ijk}\left(r_{ik}+r_{jk}-r_{ij}\right)}$$

$$S_{ij} = \prod_{k\neq i,j} S_{ijk}$$

$$W_i^{(p)} = -\sigma_i\left(\sum_{j\neq i} f_c\left(r_{ij}\right)S_{ij}b_{ij}\right)^{1/2}$$

Artificial Neural Networks in Atomistic Simulations

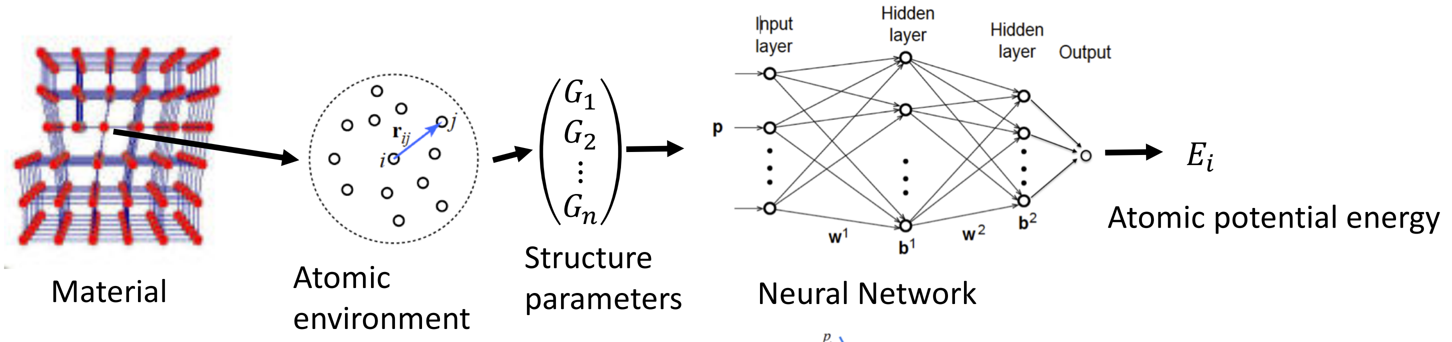
Problems with MD/MC:

- Classical MD/MC simulations use empirical potentials, $E_i = \frac{1}{2}\sum_{j\neq i}\left[A_i e^{-\alpha r_{ij}} - B_i S_{ij} b_{ij} e^{-\beta r_{ij}}\right] f_c(r_{ij})$ – **fast**, but **inaccurate**.

  A_i, B_i, α, β – fitted parameters

- Quantum mechanics based (*ab-initio*) MD/MC simulations – **accurate**, but **slow**.
- Both, classical and QM simulations have poor scalability.

Straight ANN potentials (gives atomic energy as an output)



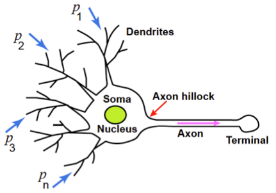
Material

Atomic environment

Structure parameters

Neural Network

Atomic potential energy



- **As accurate** as, **but faster** than QM
- Shows **poor transferability** outside the training data set

ANNs offer highly efficient utilization of the manycore - GPU computing architecture

Artificial Neural Networks in Atomistic Simulations

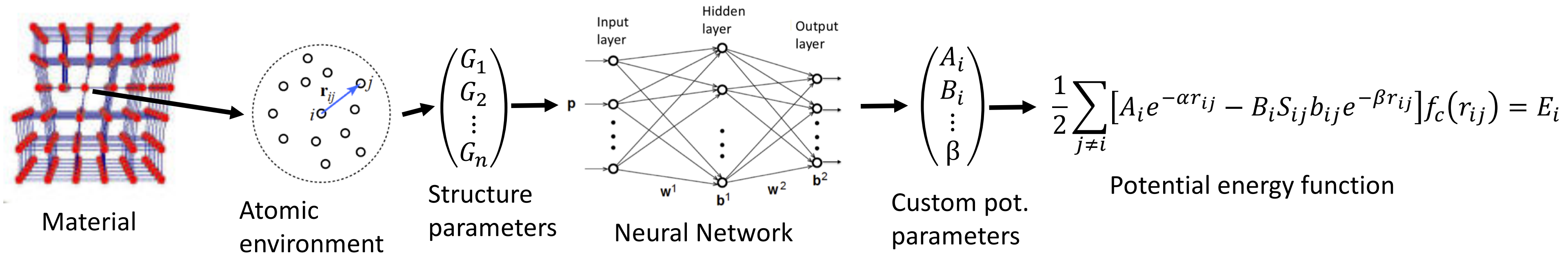
Problems with MD/MC:

- ➢ Classical MD/MC simulations use empirical potentials, $E_i = \frac{1}{2}\sum_{j \neq i}\left[A_i e^{-\alpha r_{ij}} - B_i S_{ij} b_{ij} e^{-\beta r_{ij}}\right] f_c(r_{ij})$ – **fast**, but **inaccurate**.

  A_i, B_i, α, β – fitted parameters

- ➢ Quantum mechanics based (*ab-initio*) MD/MC simulations – **accurate**, but **slow**.
- ➢ Both, classical and QM simulations have poor scalability.

Guided ANN potentials (customized pot. parameters for each atom)

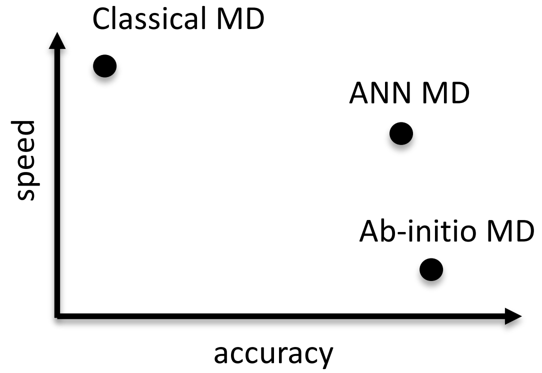


Material → Atomic environment → $\begin{pmatrix} G_1 \\ G_2 \\ \vdots \\ G_n \end{pmatrix}$ Structure parameters → Neural Network → $\begin{pmatrix} A_i \\ B_i \\ \vdots \\ \beta \end{pmatrix}$ Custom pot. parameters →

$$\frac{1}{2}\sum_{j \neq i}\left[A_i e^{-\alpha r_{ij}} - B_i S_{ij} b_{ij} e^{-\beta r_{ij}}\right] f_c(r_{ij}) = E_i$$

Potential energy function

As accurate (~meV/atom) as **but faster** than QM, and shows **improved transferability** due to the involvement of the potential function.
High computational efficiency on modern HPC hardware



ANNs offer highly efficient utilization of the manycore - GPU computing architecture

HPCI Lessons Learned

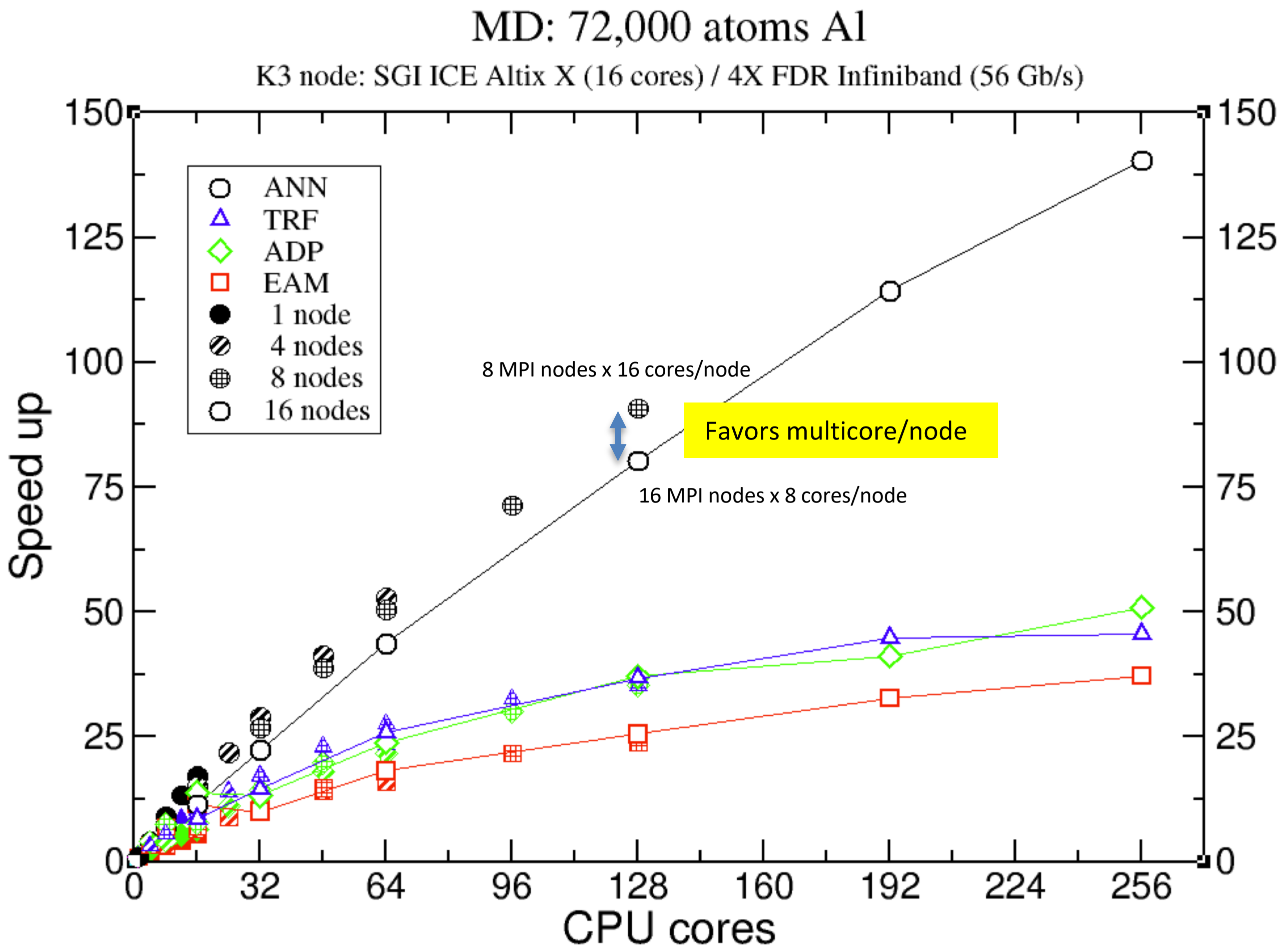
HPCI
High Performance Computing Incubator

Current Achievements in the Developed In-House Software

Speed Up of MD simulations

Scalability comparison for different potential types

Distributed (MPI) + shared memory (OpenMP) parallelization on 16 MPI nodes with 1-16 cores/node



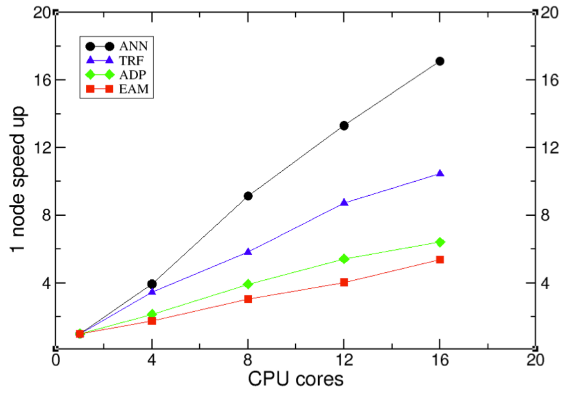
8 MPI nodes x 16 cores/node

Favors multicore/node

16 MPI nodes x 8 cores/node

ANN potential shows superior scalability

Scalability on a single node



ANN – Artificial Neural Network
Machine Learning potential

TRF – Tersoff Potential
3-body potential (molecular crystals)

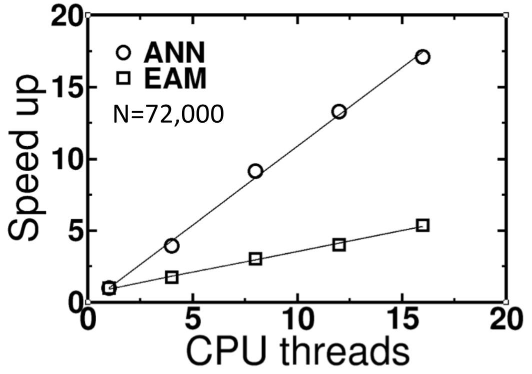
ADP – Angular Dependent Potential
many-body anisotropic potential (rare earth metals)

EAM – Embedded Atom Method
many-body isotropic potential (metal alloys)

Time Comparison of Various Sim. Methods

N=500 100 MDS	EAM 16 threads	ANN 16 threads	DFT 32 nodes
Time, t (s)	0.39	14	46,688
t/t_{EAM}	1	35.7	119,107

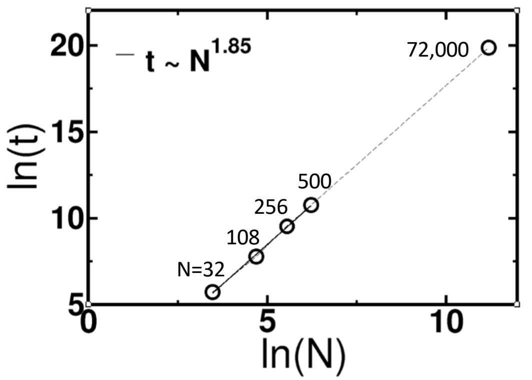
MD Spead Up for Al

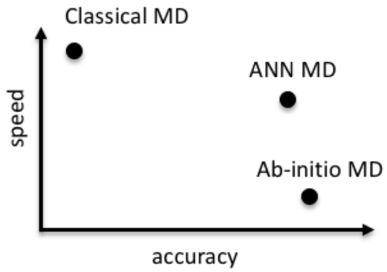


N=72,000 100 MDS	EAM 4 nodes x 8 threads	ANN* 4 nodes x 8 threads	DFT 32 nodes extrapolated
Time, t	2.66 s	778 s	13.5 years
t/t_{EAM}	1	294.1	161,000,000

*No GPU yet

VASP DFT Scaling for Al





ANN: 10^2 – 10^3X slower than EAM, but is much faster than DFT with comparable accuracy
ANN is very efficient for massive parallelization

Conclusions

- The ongoing paradigm shift in HPC hardware requires redevelopment of the simulation algorithms and codes
- Current MD codes do not efficiently explore the new HPC architecture, but progress is being made
- Machine learning strategies, such as ANN, introduce new approaches (and new challenges) in atomistic simulations
- Lessons learned from the HPCI effort at NASA LaRC helped significantly to improve in-house codes
- Future work: exploring GPU performance