US DOE SC ASCR
FY11 Software Effectiveness
SC GG 3.1/2.5.2 Improve Computational Science Capabilities

K. J. Roche
High Performance Computing Group, Pacific Northwest National Laboratory
Nuclear Theory Group, Department of Physics, University of Washington

Presentation to ASCAC, Washington, D.C. , November 2, 2011

LAMMPS

Paul Crozier, Steve Plimpton, Mark Brown, Christian Trott

OMEN, NEMO 5

Gerhard Klimeck, Mathieu Luisier, Sebastian Steiger, Michael Povolotskyi, Hong-Hyun Park, Tillman Kubis

OSIRIS

Warren Mori, Ricardo Fonseca, Asher Davidson, Frank Tsung, Jorge Vieira, Frederico Fiuza, Panagiotis Spentzouris

eSTOMP

Steve Yabusaki, Yilin Fang, Bruce Palmer, Manojkumar Krishnan

Rebecca Hartman-Baker
Ricky Kendall
Don Maxwell
Bronson Messer
Vira j Paropkari
David Skinner
Jack Wells
Cary Whitney
Rice HPC Toolkit group
DOE ASCR, OLCF and NERSC staff

- Metric Statement, Processing

- FY11 Applications
  - Intro
  - Problems
  - Enhancements
  - Results

- FY12 Nominations

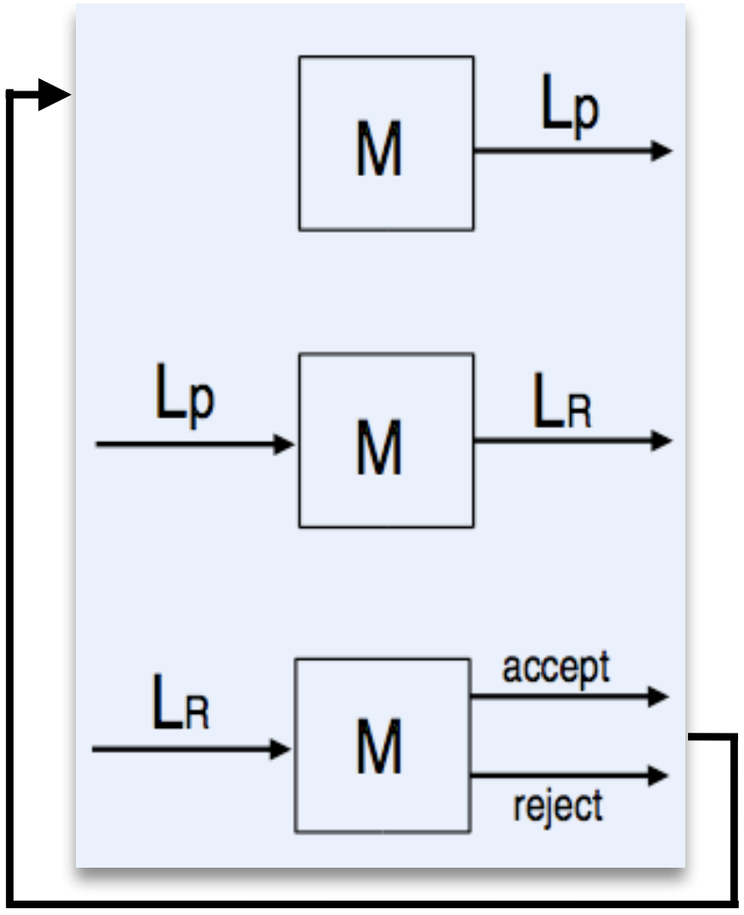
US OMB PART DOE SC ASCR

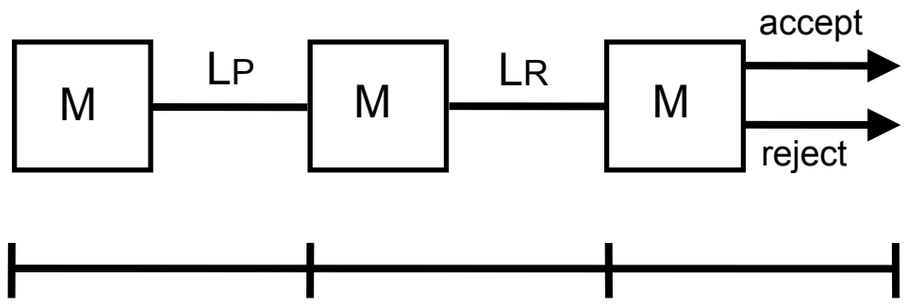
Annual Goal with Quarterly Updates

(SC GG 3.1/2.5.2) Improve computational science capabilities, defined as the average annual percentage increase in the computational effectiveness (either by simulating the same problem in less time or simulating a larger problem in the same time) of a subset of application codes.

To those 'on the clock' with this work, it means more than just satisfying the language of this metric.

- COMPLEXITY
  - PROBLEMS
  - ALGORITHMS
  - MACHINES





Measured time for machine M to generate the language of the problem plus time to generate the language of the result plus the time to accept or reject the language of the result.

Asking questions, solving problems is recursive process

Accepting a result means a related set of conditions is satisfied

$$S = S1 \wedge S2 \wedge \ldots \wedge Sn$$

Computational Effectiveness

- Total elapsed time to execute a problem instance with a specific software instance (algorithm) on a machine instance

"simulating the same problem in less time"

Algorithm, machine strong scaling :

Q4 problem := Q2 problem
Q4 algorithm := Q2 algorithm
Q4 machine ~ k * Q2 machine
Q4 time ~ 1/k * Q2 time

Algorithm enhancements, performance optimizations:

Q4 problem := Q2 problem
Q4 algorithm ~ enhanced Q2 algorithm
Q4 machine := Q2 machine
Q4 time ~ 1/k * Q2 time

*Could consider other variations: algorithm and machine are varied to achieve reduction of compute time

"simulating a larger problem in same time"

Algorithm, machine weak scaling (100%):

Q4 problem ~ k * Q2 problem
Q4 algorithm := Q2 algorithm
Q4 machine ~ k * Q2 machine
Q4 time := Q2 time

Algorithm enhancements, performance optimizations:

Q4 problem ~ k * Q2 problem
Q4 algorithm ~ enhanced Q2 algorithm
Q4 machine := Q2 machine
Q4 time := Q2 time

*Could consider other variations: problem, algorithm and the machine are varied to achieve fixed time assertion

Machine Perspective of Software Effectiveness

Strong Scaling

Machine Events	Q2	Q4
INS	2.147E+15	2.1130E+15
FP_OP	5.896E+14	5.8947E+14
PEs	5632	11264
Time[s]	121.252233	57.222988

INS:
2113046508030116 /
2146627269408190 = **.9843**

FP_OP:
589469277576687 /
589624961638025 = **.9997**

PEs: 11264 / 5632 = **2**

Time[s]:
57.222988 / 121.252233 = **.472**

Optimization

Machine Events	Q2	Q4
INS	3.16E+12	4.37E+11
FP_OP	5.50E+11	5.53E+11
PEs	1	1
L2DCM	823458808	34722900
Time[s]	826.494142	79.414198

INS: 0.1381 (7.239x)

FP_OP: 1.0053 (0.99475x)

PEs: 1

L2DCM: 0.0422 (23.715x)

Time[s]: 0.0961 (10.407x)

Weak Scaling

Machine Events	Q2	Q4
INS	5.18E+17	1.93E+18
FP_OP	4.63E+17	1.81E+18
PEs	7808	31232
Time[s]	25339	23791

INS: 3.72

FP_OP: 3.92

PEs: 4

Time[s]: .938

NB: .938 * 4 = 3.752

How Are Applications Performing on Today's Systems

FY 2011 US OMB PMM
DOE SC ASCR Software Metric SC GG
3.1/2.5.2: Improve Computational Science Capabilities

October 31, 2011

- Description of Problem Domain, Target Problems

- Description of Application Software, Algorithm Implementation

- Benchmark Parameters Q2, Q4
  - -problem instance
  - -build environment, build
  - -runtime environment, run script

- Benchmark Results Q2, Q4
  - -performance data
    - --wall time
    - --machine events
  - -simulation results

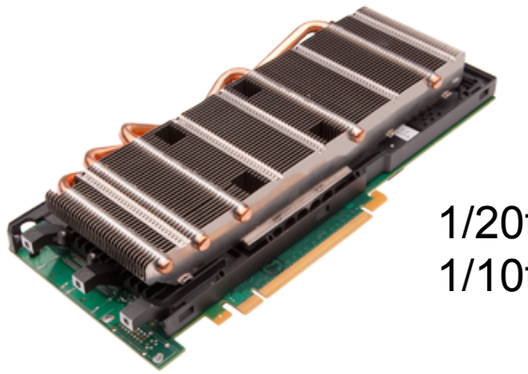
- Comparative Analysis of Q2 and Q4 results
  - -description of problem related findings
  - -description of software enhancements

Target Computing Platform: Cray XT5 -JaguarPF

Hex-Core AMD Opteron (TM)	2.6e9 Hz clock	4 FP_OPs / cycle / core 128 bit registers
PEs	18,688 nodes	224,256 cpu-cores (processors)
Memory	16 GB / node 6 MB shared L3 / chip 512 KB L2 / core 64 KB D,I L1 / core	dual socket nodes 800 MHz DDR2 DIMM 25.6 GBps / node memory bw
Network	AMD HT SeaStar2+	3D torus topology 6 switch ports / SeaStar2+ chip 9.6 GBps interconnect bw / port 3.2GBps injection bw
Operating Systems	Cray Linux Environment (CLE) (xt-os2.2.41A)	SuSE Linux on service / io nodes

FY	Aggregrated Cycles	Aggregated Memory	Aggregated FLOPs	Memory/FLOPs
2008	65.7888 THz	61.1875 TB	263.155 TF	0.2556
2009	343.8592 THz	321.057 TB	1.375 PF	0.2567
2010 / 11	583.0656 THz	321.057 TB	2.332 PF	0.1513

Target Computing Platform: Dirac GPU Testbed



1/20th the power
1/10th the cost

Quad-Core (5530) Intel Nehalem (TM)	2.4e9 Hz clock
PEs	44 nodes 352 PEs
Memory/node	24 GB DDR3 1066 Reg ECC 8 MB Cache
Network	QDR IB
Operating Systems	Linux
PCIe2 x16	4000 [8000] MBps
Memory Spd	1.5 GHz
Memory BW	144 GBps
SP FP Perf	1.03 TFLOPs

NVIDIA Corporation NVIDIA CUDA

Device 0:	"Tesla C2050"
CUDA Driver Version:	3.20
CUDA Runtime Version:	3.20
Type of device:	GPU
Compute capability:	2
Double precision support:	Yes
Total amount of global memory:	2.62445 GB
Number of compute units/multiprocessors:	14
Number of cores:	448
Total amount of constant memory:	65536 bytes
Total amount of local/shared memory per block:	49152 bytes
Total number of registers available per block:	32768
Warp size:	32
Maximum number of threads per block:	1024
Maximum group size (# of threads per block)	1024 x 1024 x 64
Maximum item sizes (# threads for each dim)	65535 x 65535 x 1
Maximum memory pitch:	2147483647 bytes
Texture alignment:	512 bytes
Clock rate:	1.147 GHz
Concurrent copy and execution:	Yes
Run time limit on kernels:	No
Integrated:	No
Support host page-locked memory mapping:	Yes
Compute mode:	Default
Concurrent kernel execution:	Yes
Device has ECC support enabled:	Yes

http://ark.intel.com/products/37103/Intel-Xeon-Processor-E5530-(8M-Cache-2_40-GHz-5_86-GTs-Intel-QPI)

Programmers have to ...
Re-Invent Hit-or-Miss Strategies of Today

- **non-temporal writes,** ie don't cache the data writes since it won't be used again soon (i.e. n-tuple initialization)
  - avoids reading cache line before write, avoids wasteful occupation of cache line and time for write (*memset()*); does not evict useful data
  - *sfence()* compiler set barriers
- **loop unrolling** , transposing matrices
- **vectorization**
  - 2,4,8 elements computed at the same time (SIMD) w/ multi-media extensions to ISA
- **reordering** elements so that elements that are used together are stored together -pack CL gaps w/ usable data (i.e. try to access structure elements in the order they are defined in the structure)
- **stack alignment,** as the compiler generates code it actively aligns the stack inserting gaps where needed ... is not necessarily optimal -if statically defined arrays, there are tools that can improve the alignment; separating n-tuples may increase code complexity but improve performance
- **function inlining**, may enable compiler or hand -tuned instruction pipeline optimization (ie dead code elimination or value range propagation) ; especially true if a function is called only once
- **prefetching**, hardware, tries to predict cache misses -with 4K page sizes this is a hard problem and costly penalty if not well predicted; software (*void _mm_prefetch(void *p, enum _mm_hint h) --_MM_HINT_NTA* -when data is evicted from L1d -don't write it to higher levels)

ref. U. Drepper

(some) Challenges for the Semiconductor Industry

Fabricate and demonstrate devices that:

1) will work "because" and not "in spite" of quantum mechanical effects
2) will keep Moore's law going as long as possible
3) will outperform the conventional silicon metal-oxide-semiconductor field-effect transistors (MOSFETs)
4) will reduce the power consumption of integrated circuits

OMEN and NEMO 5:

Software that enables the study of next generation nanoelectronic devices where quantum mechanical effects such as energy quantization and electron tunneling play a very important role

- multi-dimensional, parallel quantum transport solvers
- atomistic resolution of the simulation domain
- full-band description of the material properties
- go beyond coherent transport and include electron-phonon scattering
- address the accuracy problems of the semi-classical drift-diffusion and Monte Carlo

- 3-D nanowire transistors in the ballistic limit
- 2-D structures such as ultra-thin-body field-effect transistors
- new materials such as graphene and wurtzite
- devices such as band-to-band tunneling transistors

-transport of tens of thousands of atoms **(OMEN)**

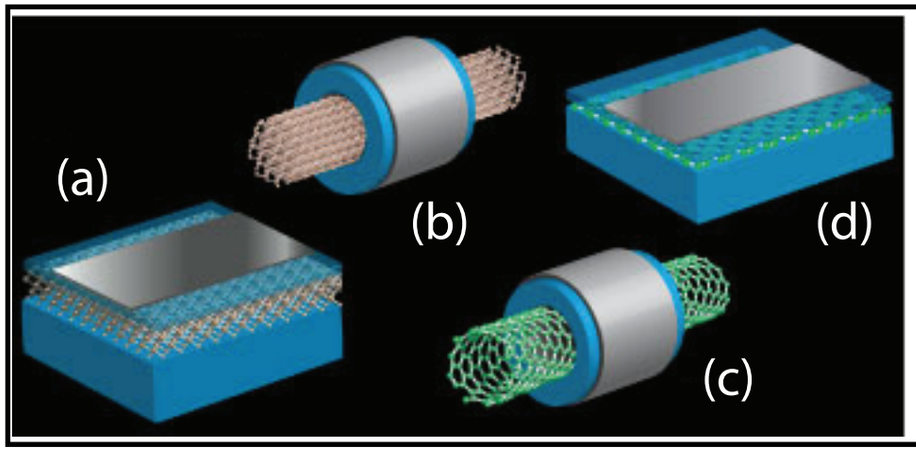
-structures, strain, quantum eigenstates of tens of millions of atoms **(NEMO5)**

-quantum computing

-optoelectronics

OMEN:

Si and InGaAs
UTB transistor



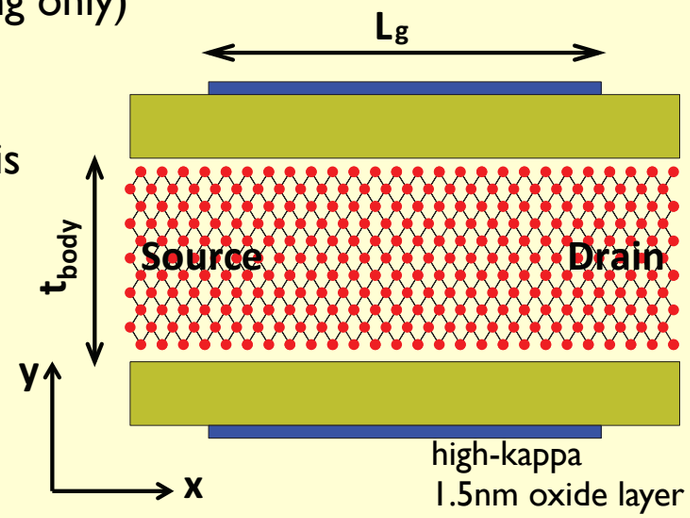
a) Si, Ge, III-V semiconductors for single and double-gate ultra-thin-body field effects
b) gate-all-around nanowire FET
c) graphene nanoribbon FET
d) coaxially-gated carbon nanotube

n - type device (donor doping only)

electron transport
  along the <110> crystal axis
  confinement along <100>

tbody = 5nm, Lg = 22nm
total length L = 42nm

6068 atoms, rel. diel ~ 20



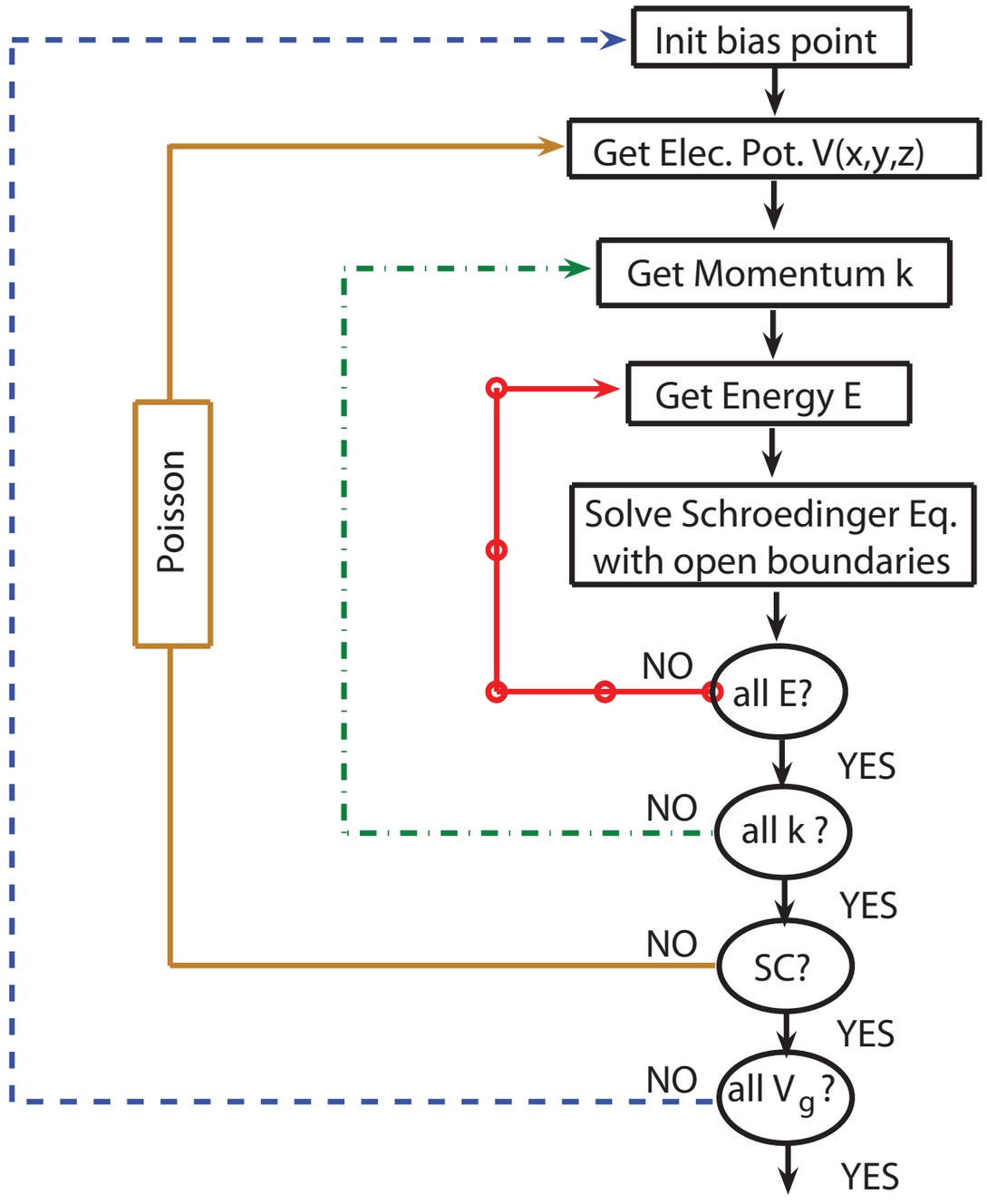
10 orbitals w/ sp^3d^5s* tight binding and no spin orbit
  (1 s + 3 p + 1s* (excited) + 5 d orbitals)

dim(Hamiltonian) = 60,680

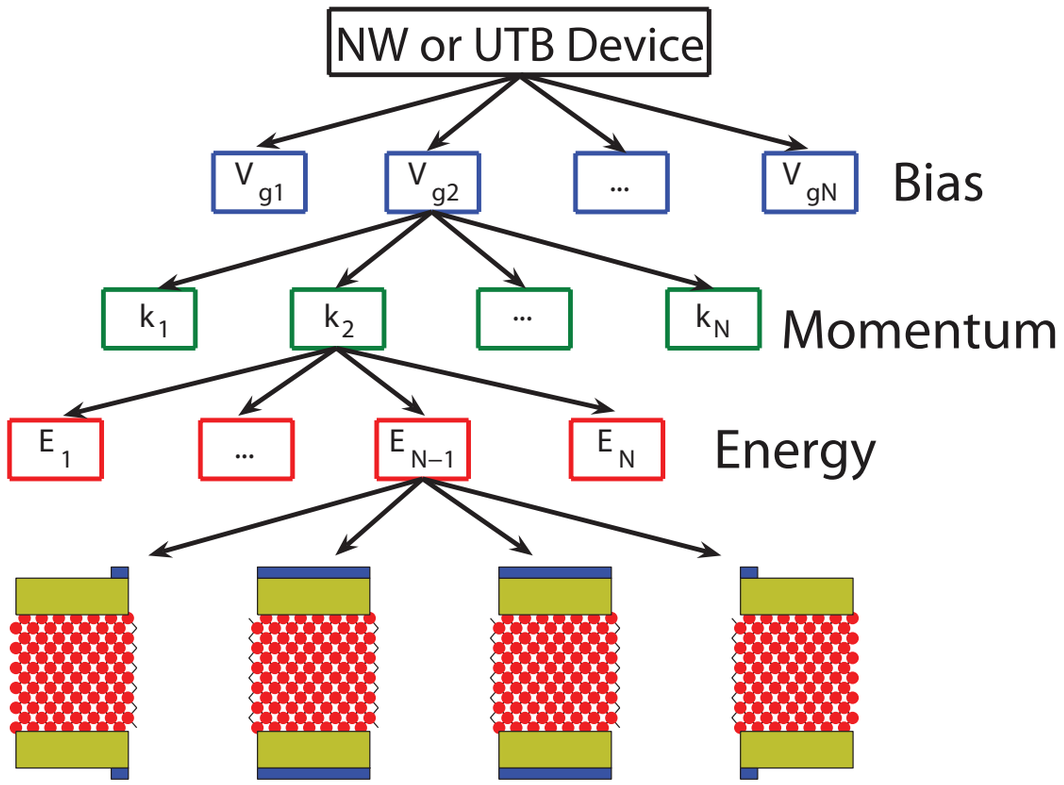
25 momentum points in $-\pi/a_0$ to π/a_0
  a_0 = .58686nm
  energy points per k-value varies [1241,1914]

10 phonon energies

external voltage configurations:
  source-to-drain voltage V=0.7 V
  gate-to-source voltage V=0.5 V
  1% uni-axial stress applied along the transport direction



OMEN: Enhancement in Problem Formulation



1-D Spatial Domain Decomposition

	Q2	Q4	Q4'	Q2/Q4
PEs	85,200	85,200	42600	1, 2
Time[s]	6400	2950	5900	2.17, 1.08
FP OPs	1.44E+18	4.13E+17	4.13E+17	3.49
INS	2.67E+18	1.17E+18	1.16E+18	2.28, 2.3
L2 DCM	9.52E+14	3.56E+14	3.47E+14	2.67, 2.74

A Si UTB transistor including electron-phonon scattering until convergence between the Green's Functions and self-energies was achieved. [i.e. 151,467 CPU Hrs (Q2) - 69,817 CPU Hrs (Q4) = 81,650 CPU Hrs]

- 3D nanowires are not periodic -rather confined in the lateral component
- work with the primitive unit cell of UTB transistors instead of a more convenient, but larger unit cell
- exploit physical symmetry that imposes k-dependence on observables density of states, transmission

$$\mathbf{H} = \mathbf{H_{00}} + \mathbf{H_{01}} \cdot \exp(ik\Delta_z) + \mathbf{H_{10}} \cdot \exp(-ik\Delta_z)$$

H(0,0) describes central unit cell

H(0,1) , H(1,0) the connection to the unit cell from above and below by Δ_z

- each sub-matrix can be generated independently
- the sparse pattern of the matrix sum is stored in H during initialization

NEMO 5: Quantum Dot

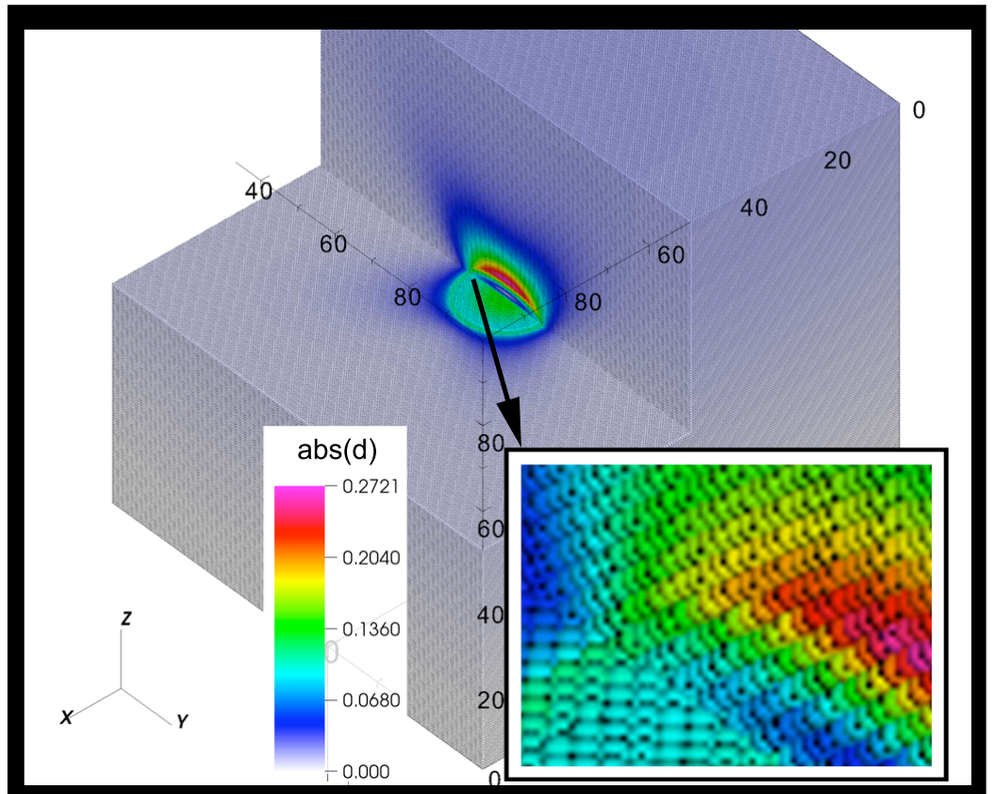
Optimization of ballistic quantum transport

quantum dots are nanostructures where a 3D confinement of carriers is achieved; energy spectrum of carriers is discrete, like in atoms

example of a quantum dot is indium arsenide (InAs) embedded in gallium arsenide (GaAs)

(strain) a large lattice mismatch of 7% between InAs and GaAs makes it energetically preferable for InAs to form little chunks rather than a flat plane when epitaxially grown on GaAs beyond a critical thickness of about 1.6 monolayers leading to a nontrivial 3D strain influencing the carrier energies

(eigenstates / structure) an accurate description of the electronic states in the strained quantum dot is needed to model these devices; quantum dots themselves typically contain a few hundred thousand to a few million atoms giving rise to a countable number of continued electron and hole states; the neighborhood surrounding the dots must be included and valence bands need to be modeled with spin-orbit leading to tens of millions of atoms to be modeled



- size of quantum dot at cube center is held constant to so that results converge to well-defined values and is (5 nm)^3 in our tests

- cube of varying size containing N atoms is partitioned into Nx Ny Nz smaller cubes,

- nPEs := Nx Ny Nz

- (strain) nDOFs = 3N ; harmonic Keating variant of the valence force field model

- (eigenstates / electronic structure) nDOFs = 20N ; employs tight-binding model

NEMO 5: enhancements

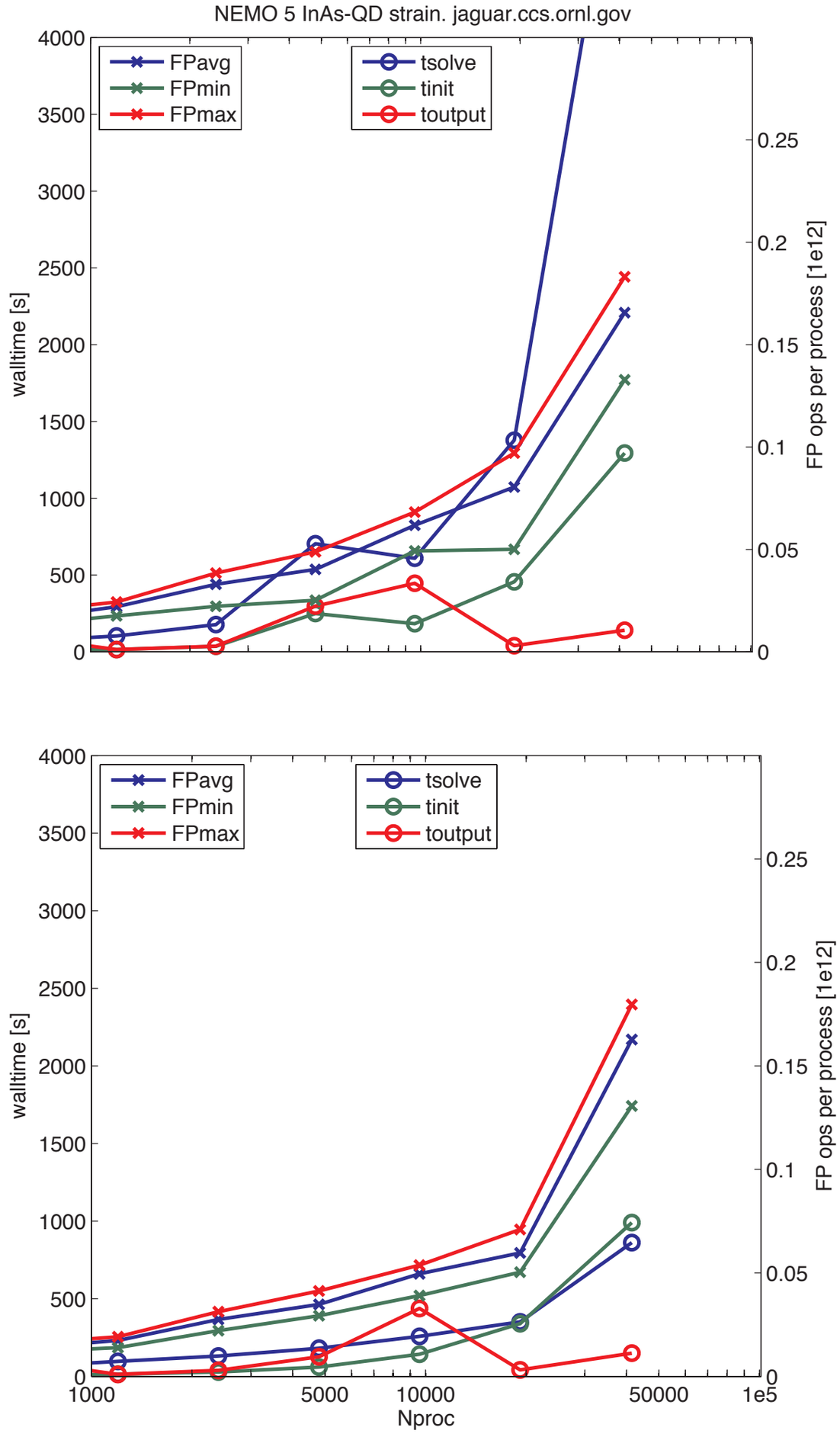
- compiling PETSc and SLEPc specifying the flag `--with-debugging=0`
- improved matrix preconditioning:
    - in Q2 was block Jacobi (bjacobi in PETSc) since the adaptive Schwarz (asm in PETSc), though superior numerically, lacked memory scalability (some arrays in the implementation scaled with the global problem number of variables)
- physical domain decomposition requires couplings to neighboring partitions as well as neighbor-neighbor couplings
    - unnecessary barriers in communication were corrected allowing an asynchronous execution of this step
- material parameter input from FILE
    - global access to the input was restricted to a master process that read and distributed the initialization parameters

NEMO 5: Quantum Dot

Optimized strain computation

Metric	Q2	Q4
DOFs	994,121,664	994,121,664
FPsolve	6.87e15	6.75e15
FPmax	183.17e9	179.67e9
FPmin	132.81e9	130.66e9
tinit	1294.57s	990.83s
tsolve	5610.34s	862.17s
toutput	141.03s	149.08s
ttotal	7045.94s	2002.08s

3.52X more efficient in Q4 at 41,472 PEs in both Q2, Q4 -same problem and results in both runs!

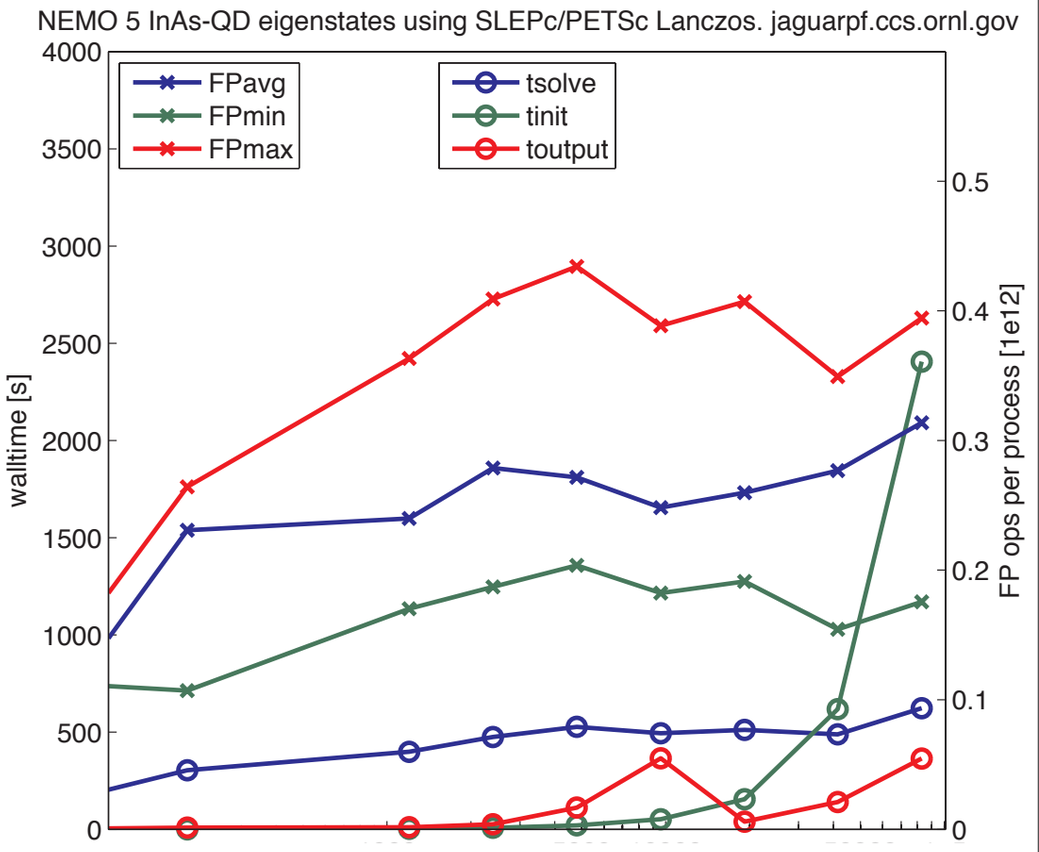


NEMO 5: Quantum Dot

Optimized eigenstate computation

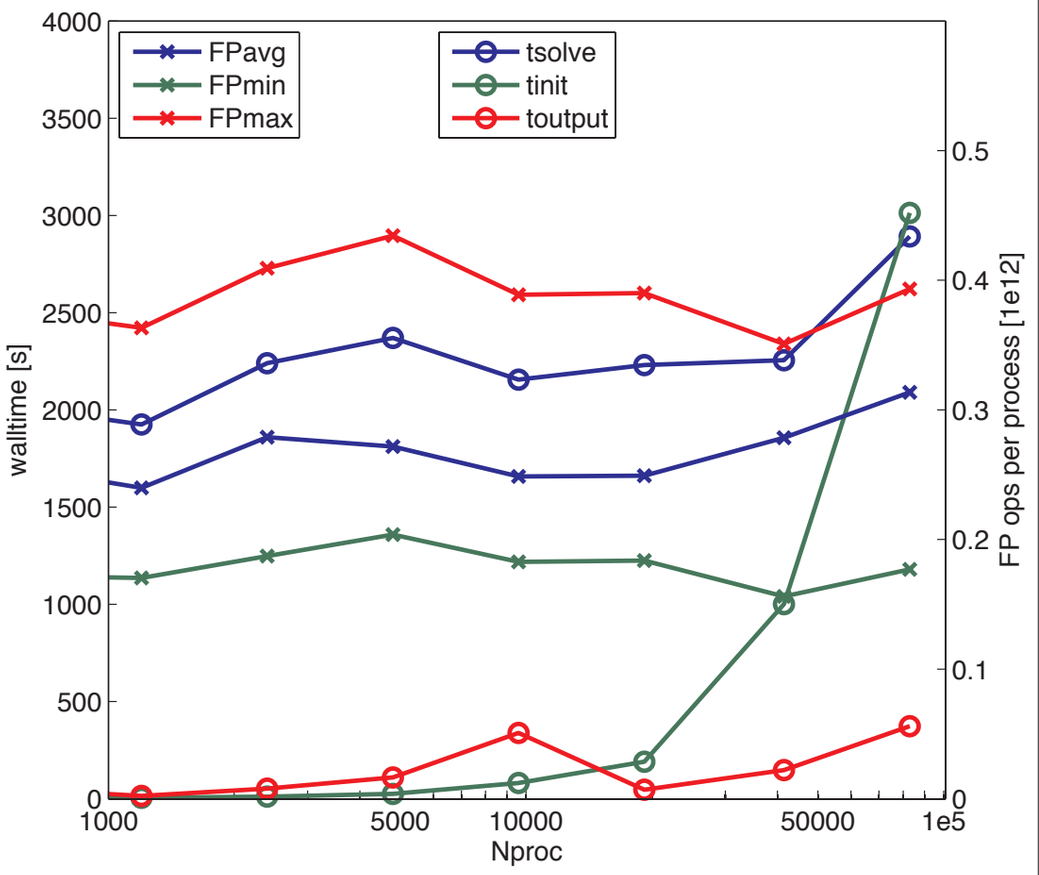
Number of Lanczos iterations to achieve six converged eigenstates: 41,472 PEs in both Q2, Q4 but Q4 algorithm is 2.73X faster!

Metric	Q2	Q4
Nlancz	349	349
FPsolve	11.55e15	11.48e15
FPmax	351.0e9	349.3e9
FPmin	156.1e9	154.4e9
tinit	1001.3s	618.4s
tsolve	2256.1s	488.4s
toutput	147.5s	141.0s
ttotal	3404.9s	1247.8s

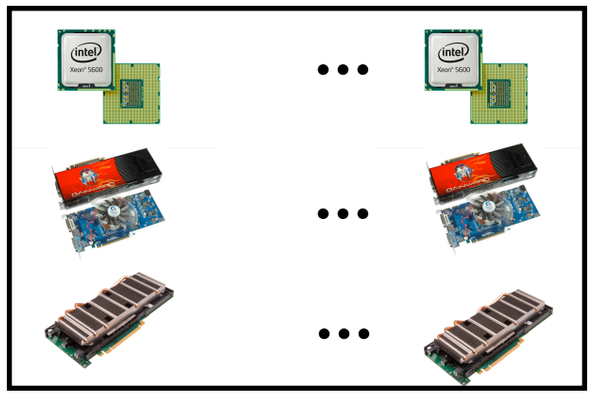


Q4(weak)

# MPI procs	DOFs	Atoms	# Lanczos its.	tinit [s]	tsolve [s]	toutput [s]
12	116.640	5.832	140	0.24	138.95	0.67
24	276.480	13.824	161	0.18	162.46	0.75
48	540.000	27.000	161	0.26	178.58	2.03
96	933.120	46.656	186	0.34	197.85	4.81
192	1.946.720	97.336	296	0.58	303.85	9.24
1200	12.721.120	636.056	294	3.57	399.68	11.00
2400	25.194.240	1.259.712	345	7.91	475.58	26.86
4800	48.122.080	2.406.104	352	20.63	526.81	111.94
9600	98.260.000	4.913.000	314	52.37	494.67	365.34
19200	196.006.880	9.800.344	327	155.44	511.45	40.34
41472	420.491.520	21.024.576	349	618.36	488.44	140.98
82944	842.883.840	42.144.192	395	2405.81	623.49	364.42



LAMMPS: Large-scale Atomic/Molecular Massively Parallel Simulator (lammps.sandia.gov)



Classical molecular dynamics and particle simulation software

- biomolecules, polymers
- metals, semiconductors
- coarse grained / mesoscopic systems

- not aiming to optimize for running on a single node + GPU
  - `user-cuda` package aimed at balanced ratio of processor cores to GPUs
  - `gpu` package aimed at hybrid nodes coupling multi-core hosts to GPUs
    - spatial decomposition with MPI between CPU cores
    - force (atom) decomposition with CUDA on individual GPUs

- main operations are neighbor lists formation, link cell binning, and function evaluation
  - stored (Verlet) lists of neighboring atoms reused multiple times prior to rebuild
  - link cell binning into 3D cells, O(N) cost of finding neighbors in 27 bins
  - combination of the above
    - search half the bins of each atom to form its neighbor list -Newton's Law 3

LAMMPS: Benchmark Problems and GPU - enabled Algorithms

- **Cu cluster**
  - more complex than 1/r interaction w/ cutoff
    - embedded atomic interaction
      - atoms are embedded in field composed of all other atoms in host; impurities are locally uniform and determined by electron density of host prior to adding impurity
      - total interaction energy sum of unperturbed host electron density profile and position and charge of impurity

- **Atomic "Melt" Problem**
  - basically a vanilla 1/r pair force computation
    - employs Lennard-Jones (12,6) atomic interaction

- Compare speed-ups in different scaling modes w/ increased numbers of time steps and atoms between the MPI and MPI + { `user-cuda` , `gpu` } versions of the algorithms
  - `user-cuda` was designed to coordinate computation between 1 MPI process and GPU
  - `gpu` was designed to coordinate an arbitrary number of MPI processes to share GPU

LAMMPS: Cu cluster

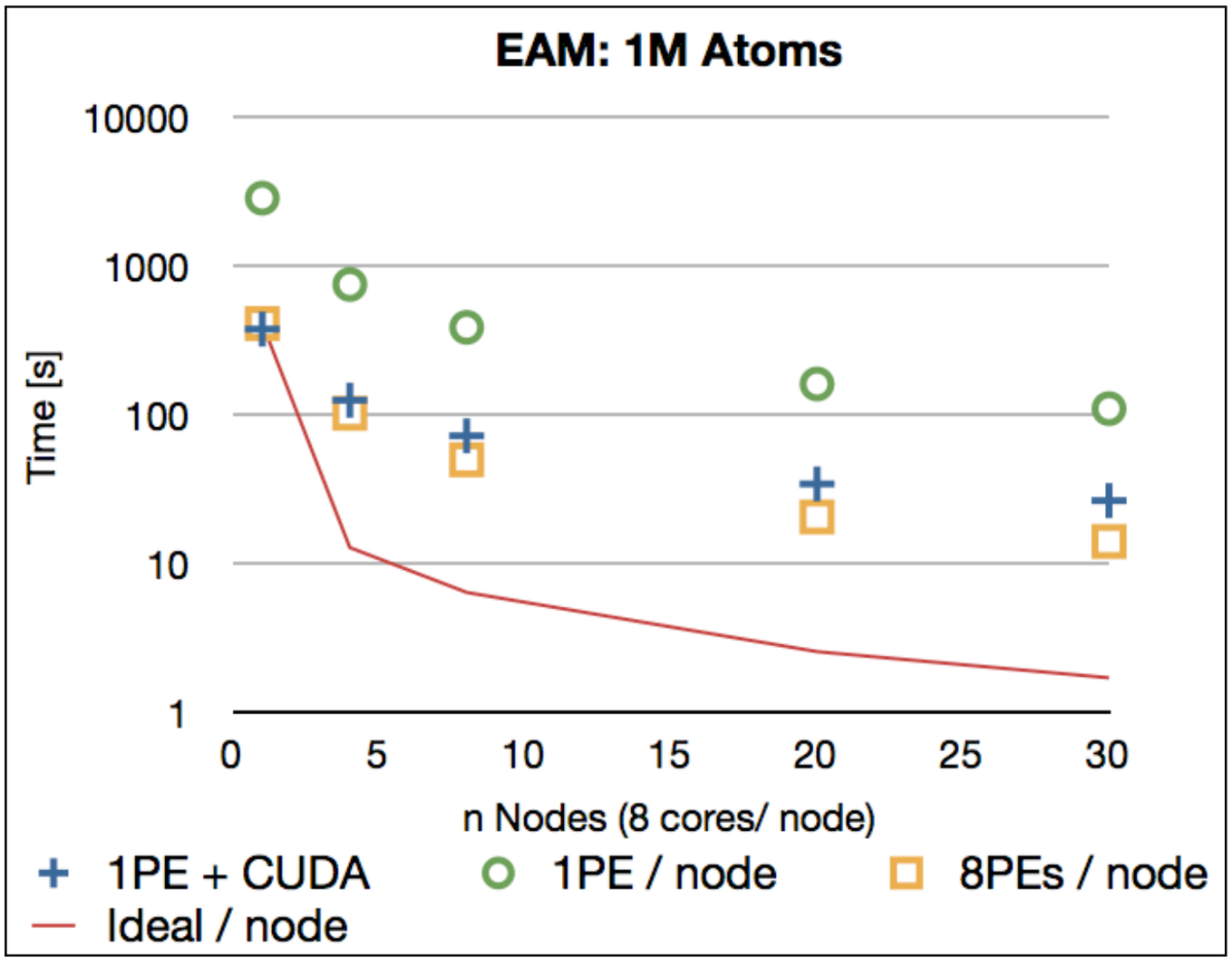
Embedded Atom Potential (`user-cuda`)

	Nodes	Processes	Time [s]
MPI	1	1	2872.59
	1	4	764.202
	4	4	755.475
	1	8	412.402
	8	8	389.187
	20	20	161.931
	30	30	110.603
	4	32	102.942
	8	64	50.1122
	20	160	20.6911
	30	240	14.1628
+CUDA	1	1	378.595
	4	4	125.999
	8	8	72.6575
	20	20	34.5002
	30	30	26.6687

$$E_i = F(\rho_{h,i}) + \frac{1}{2}\sum_{j \neq i} \phi(R_{ij}) \,,$$

$$E_{tot} = \sum_i F_i(\rho_{h,i}(R_{ij})) + \frac{1}{2}\sum_{i,j,i \neq j} \phi_{i,j}(R_{ij})$$

$$\phi_{ij}(r) \sim Z_i(r) Z_j(r)/r$$



LAMMPS: Cu cluster

- Simulating 1000188 atoms for 1000 time steps.
- Lattice spacing in x,y,z = 3.615 3.615 3.615
- Created orthogonal box =
(0 0 0) to (227.745 227.745 227.745)
- Loop time of 375.115 on 1 procs

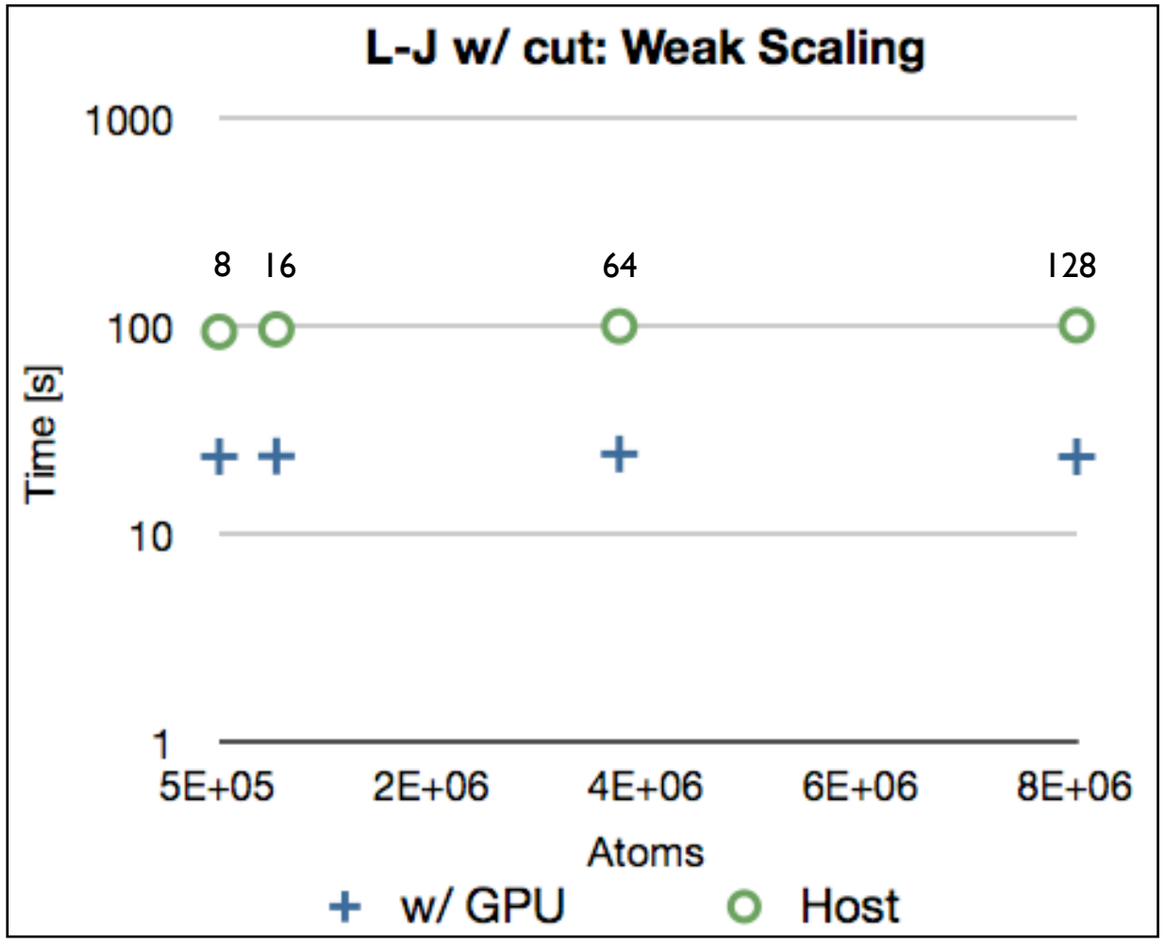
LAMMPS:: USER_CUDA in.eam
Pair time (%) = 54.3035 (14.47)
Neigh time (%) = 32.5784 (8.68)
Comm time (%) = 25.6912 (6.84)
Other (%) = 262.436 (69.96)
Nghost: 176122
Total # of neighbors: 75511371
Ave neighs/atom: 75.4972
Neighbor list builds: 163

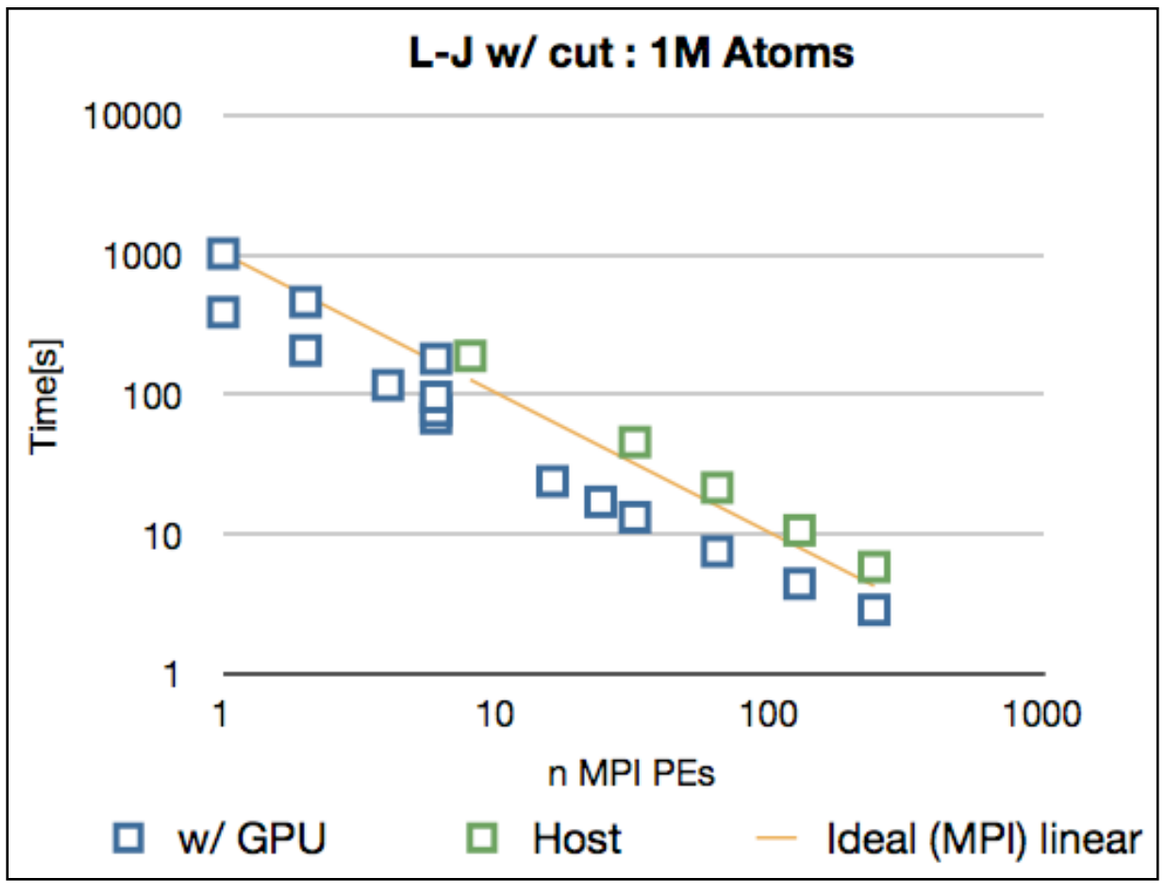
MPI (607) + CUDA (321007)			
cudaThreadSynchronize	87.79s	59672	22.87%
cudaMemcpy	63.99s	44616	16.67%
cudaGetDeviceCount	1.44s	1	0.38%
cudaMemcpyToSymbol	0.87s	41563	0.23%
cudaLaunch	0.17s	20569	0.04%
cudaSetupArgument	0.06	111206	0.02%
cudaHostAlloc	0.06	8	0.02%
cudaFreeHost	0.05	8	0.01%
cudaConfigureCall	0.03	20569	0.01%
cudaFree	0.02	81	0.01%
cudaMalloc	0.01	272	
cudaMemset	0.01	1183	
cudaGetLastError	0.01	21251	
MPI_Allreduce	488		
cudaStreamCreate		3	
cudaSetDevice		2	
cudaGetDeviceProper		1	
MPI_Bcast	101		
MPI_Barrier	2		
MPI_Scan	1		
cudaGetDevice		1	
cudaSetDeviceFlags		1	
MPI_Comm_size	4		
MPI_Comm_rank	9		
MPI_Init	1		
MPI_Finalize	1		

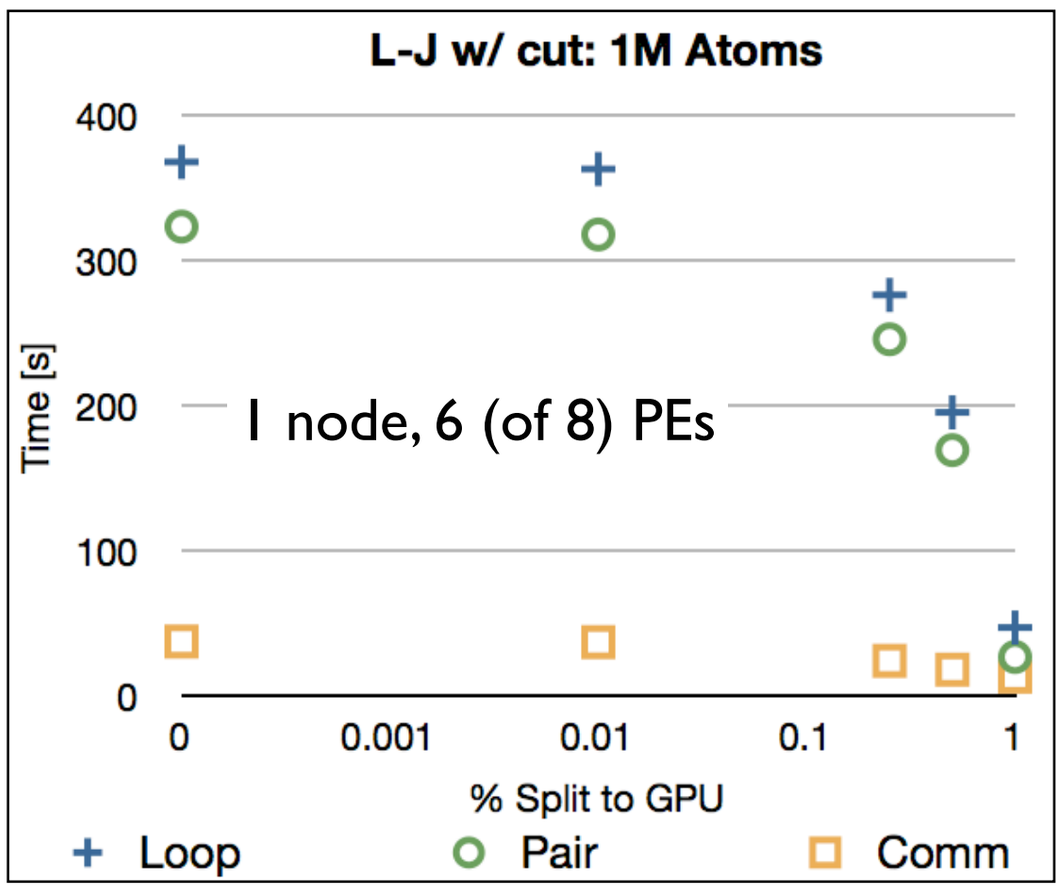
LAMMPS: system melt

Lennard-Jones (12,6) Potential (gpu)

$$4\epsilon[(\frac{\sigma}{r})^{12} - (\frac{\sigma}{r})^{6}]$$







OSIRIS: Laser Wakefields

How does a short and intense driver evolve over large distances?
How is the wake excited and how does it evolve?
How do the properties of the witness beams evolve as they are accelerated?

• short and intense laser or relativistic particle beams propagate through a plasma near the speed of light

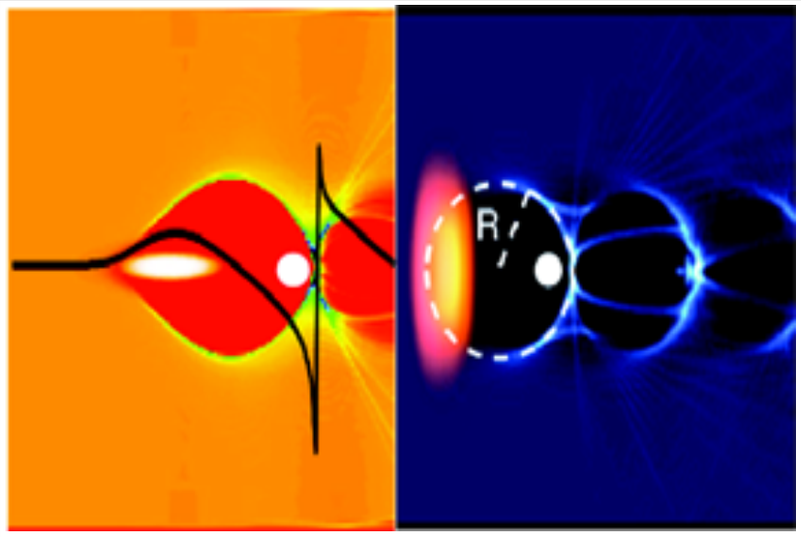
• light pressure of the laser or the space charge forces from the particle beam displaces plasma electrons

• the ions pull the electrons back towards where they started creating a plasma wave wake with a phase velocity near the speed of light

• accelerating (electric) fields in these wakes are more than 1000 times higher than those in existing accelerators.

• properly shaped and phased electrons or positron beams (witness beams) are loaded onto the wake and they surf to ultra-high energies in very short distances.

• Experiments using a laser driver have demonstrated the feasibility of generating GeV class quasi-monoenergetic beams

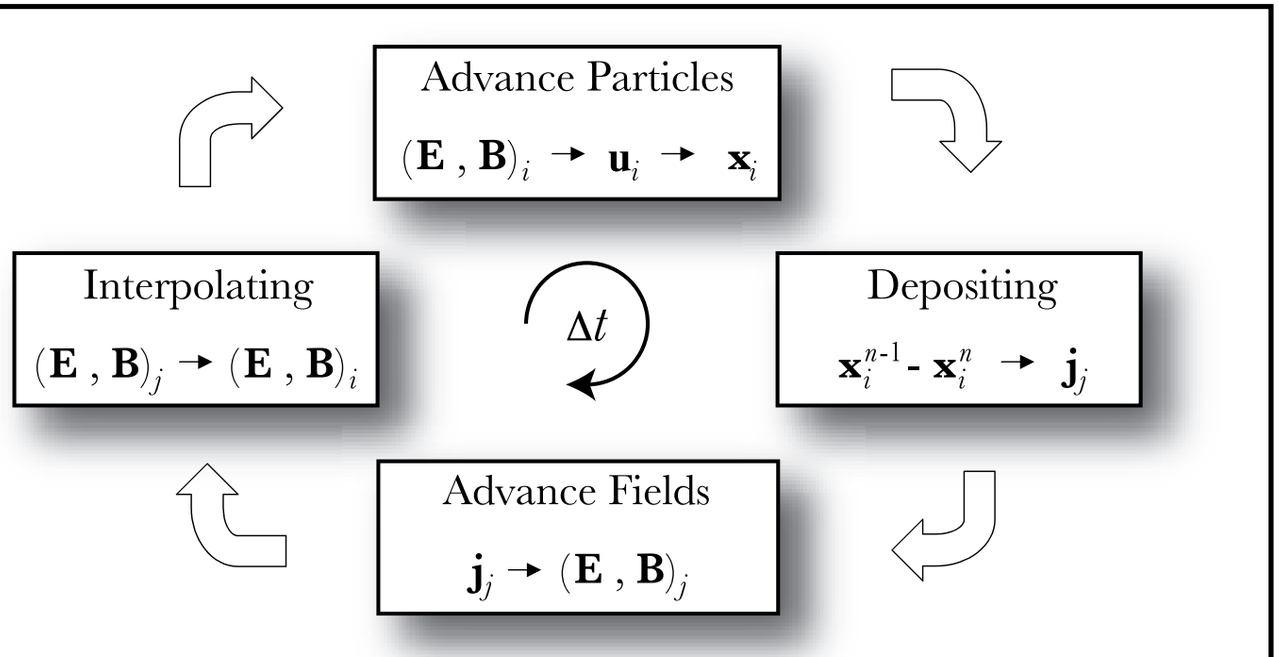


On the left is an electron beam (white) moving from right to left.
It forms a wakefield (density of plasma is shown. A lineout of the accelerating field is shown in black. A trailing bunch is shown in white in the back of the wakefield. On the right a laser (orange) is moving from right to left. It also creates a wakefield. The wakefield in both cases is a moving bubble of a radius R. A trailing beam is shown in white as well.

OSIRIS:

The fields within the wake structure demand a full electromagnetic treatment is needed.

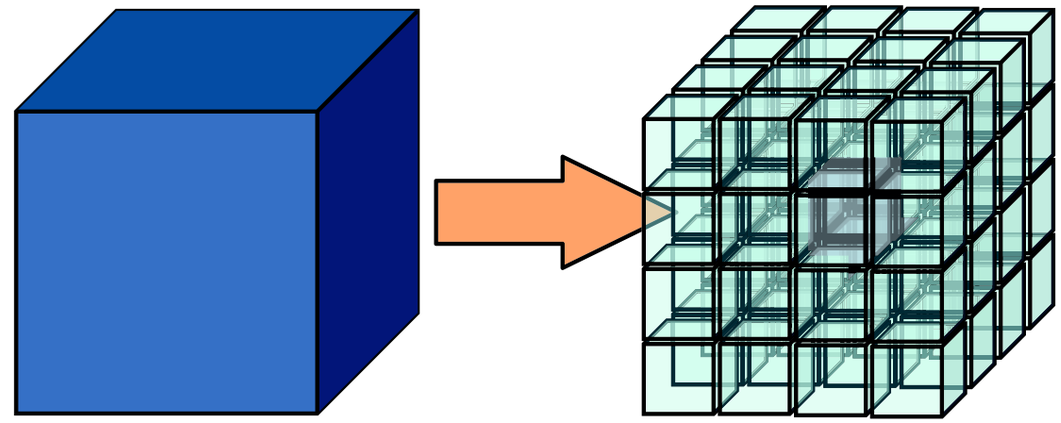
The leading kinetic description is the particle-in-cell (PIC) method.



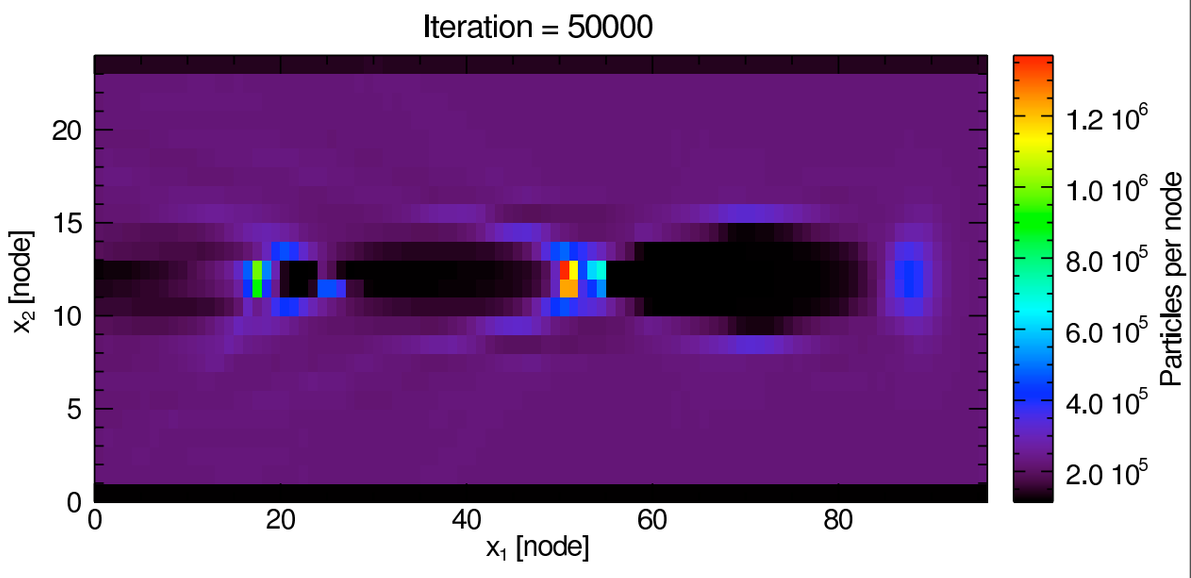
- deposit some particle quantity, such as a charge, is accumulated on a grid via interpolation to produce a source density. Various other quantities can also be deposited, such as current densities

- field solver, which solves Maxwells equations or a subset to obtain the electric and/or magnetic fields from the source densities

- particle forces are found by interpolation from the grid, and the particle coordinates are updated, using Newtons second law and the Lorentz force. The particle processing parts dominate over the field solving parts

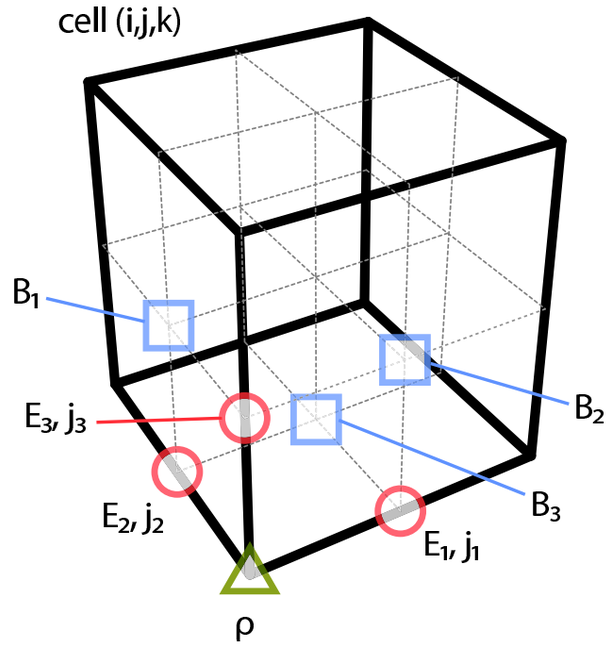


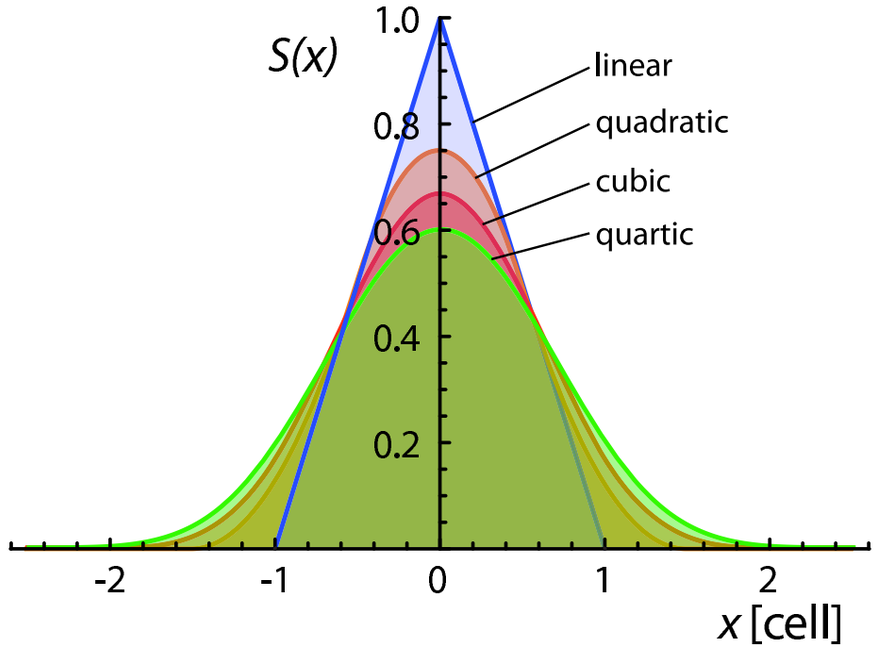
Sim.Volume Parallel Domain



Balancing the particle load is hard problem!

OSIRIS:





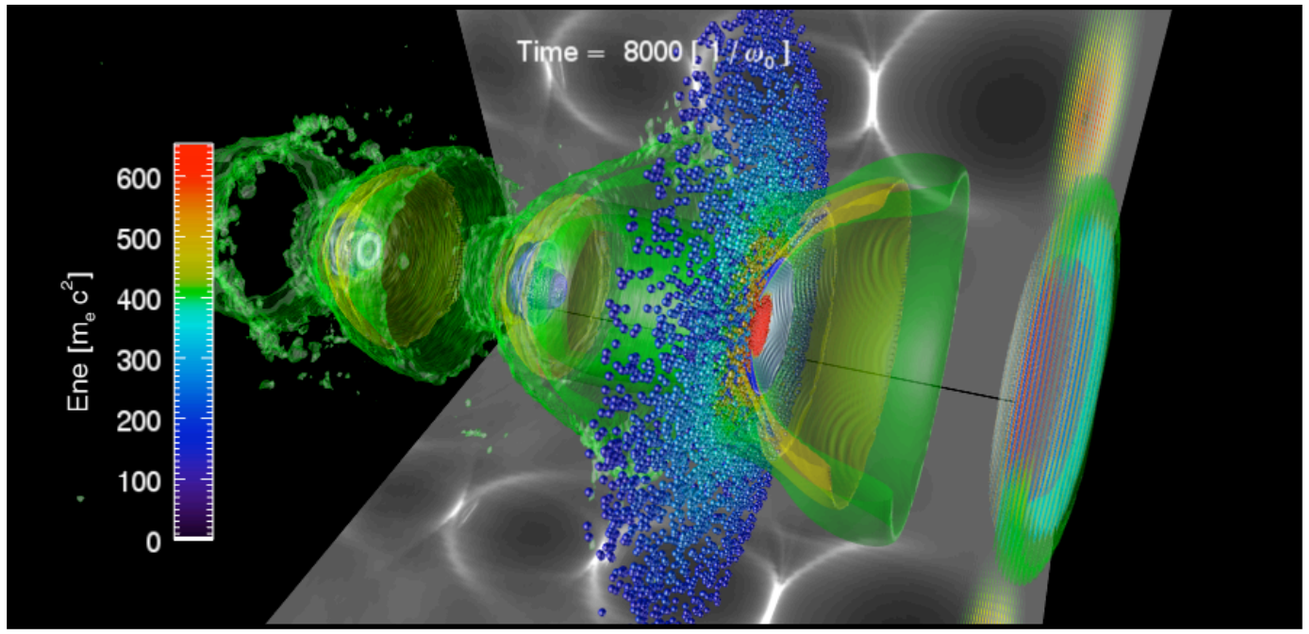
	linear	quadratic	cubic	quartic
S_{-2}				$\frac{1}{384}(1-2x)^4$
S_{-1}		$\frac{1}{8}(1-2x)^2$	$\frac{1}{6}(1-x)^3$	$\frac{1}{96}\left(-16x^4+16x^3+24x^2-44x+19\right)$
S_0	$1-x$	$\frac{3}{4}-x^2$	$\frac{1}{6}\left(3x^3-6x^2+4\right)$	$\frac{1}{4}x^4-\frac{5}{8}x^2+\frac{115}{192}$
S_1	x	$\frac{1}{8}(1+2x)^2$	$\frac{1}{6}\left(-3x^3+3x^2+3x+1\right)$	$\frac{1}{96}\left(-16x^4-16x^3+24x^2+44x+19\right)$
S_2			$\frac{1}{6}x^3$	$\frac{1}{384}(1+2x)^4$

•need a method to effectively connect grid and particles quantities to determine the force acting on the particle.

•field interpolation calculations require knowledge of the grid point index closest to the particle position, and the distance between the particle and the grid point, normalized to the cell size.

•OSIRIS implements 1st to 4th order interpolation schemes (linear, quadratic, cubic and quartic splines)

OSIRIS: Problems



Uniform Plasma

•**(1)** warm plasma with a temperature distribution parameter of u_thermal = 0.01c

• a perfectly load balanced simulation

• particle diffusion across parallel nodes happens uniformly so the total number of particles per node remains approximately constant.

• good performance test as these plasma conditions

• resemble those on most of the simulation box for the laser wakefield runs.

*quadratic shaped particles for the current deposition and field interpolation for all the simulations

Laser Wakefield scenarios

•**(2,3)** interaction of a 200 TW (6 Joule) laser interacting with uniform plasma with a density of 1.5e18 cm^-3

•plasma with an intensity sufficient to trigger self-injection, under different numerical and physical conditions.

•different grid resolutions, different number of particles per cell, and mobile/immobile ions.

•**(4)** a PW (30J) laser propagating in a .5e 18 cm^-3 plasma where ion motion is expected to play an important role

Run	Grid	Simulation Box $[c/\omega_0]$	Particles	Iterations	Laser a_0	Ions
Warm test	$6144 \times 6144 \times 1536$	$614.4 \times 614.4 \times 153.6$	4.46×10^{11}	5600	n/a	n/a
Run 1	$8064 \times 480 \times 480$	$806.4 \times 1171.88 \times 1171.88$	3.72×10^{9}	41000	4.0	fixed
Run 2	$8832 \times 432 \times 432$	$1766.4 \times 2041.31 \times 2041.31$	6.59×10^{9}	47000	4.58	fixed
Run 3	$4032 \times 312 \times 312$	$806.4 \times 1171.88 \times 1171.88$	1.26×10^{10}	52000	4.0	moving

OSIRIS: Enhancements

SIMD Optimizations and SSE Implementation

- 90 / 10 rule - advancing particles and deposting the current

- optimized the use of memory and L2 cache for vector version

- store individual components in separate sequential arrays
  -one for x, one for y and one for z

particles:
i) load 4 particles into the vector unit
ii) interpolate the EM fields for these 4 particles
iii) push the 4 particles
iv) create up to 4 × 4 virtual particles for current deposition
v) store the 4 particles back to main memory.

virtual particles:
i) load 4 virtual particles into the vector unit
ii) calculate the current contribution for the 4 virtual particles
iii) accumulate this current in the global electric current grid

- make use of vector shuffle operation to efficiently exchange parts of the vector registers:
  i) we read 3 vectors (12 positions) sequentially
  ii) shuffle them to get a vector of 4 x positions,
  one vector of 4 y positions, one vector of 4 z positions

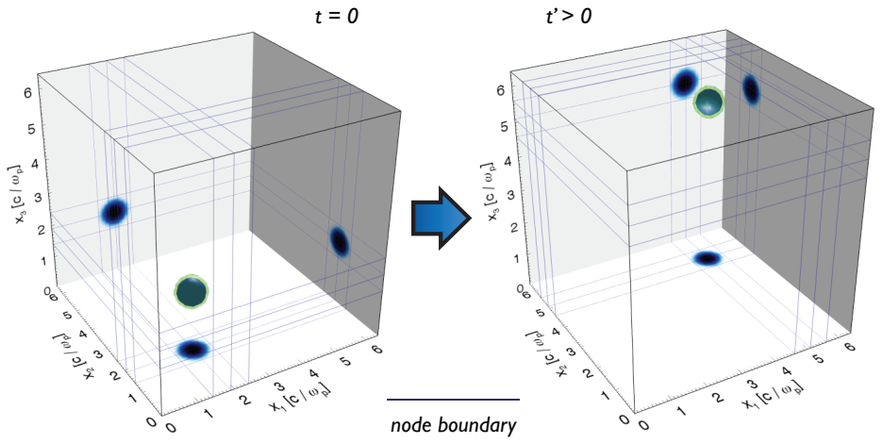
- 4 × 3 transpose is done in the registers and is very efficient (10 cycles overhead)
  -enables efficient use of vector memory read operations

- storing the particles back to memory, the opposite operation is performed

OSIRIS: Other Enhancements

Dynamic Load balancing

- 30% improvement in imbalance, but a 5% drop in overall performance
  - i) determine best partition from current load
  - ii) redistribute boundaries



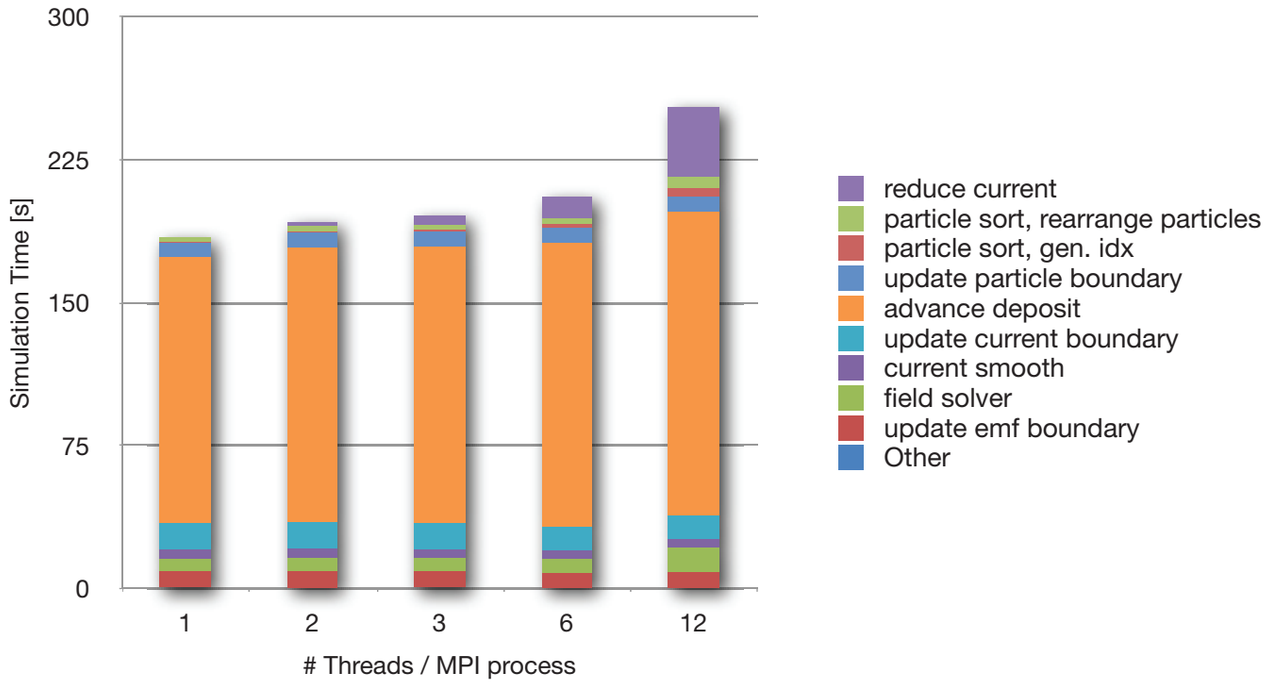
SMP version of major distributed kernels

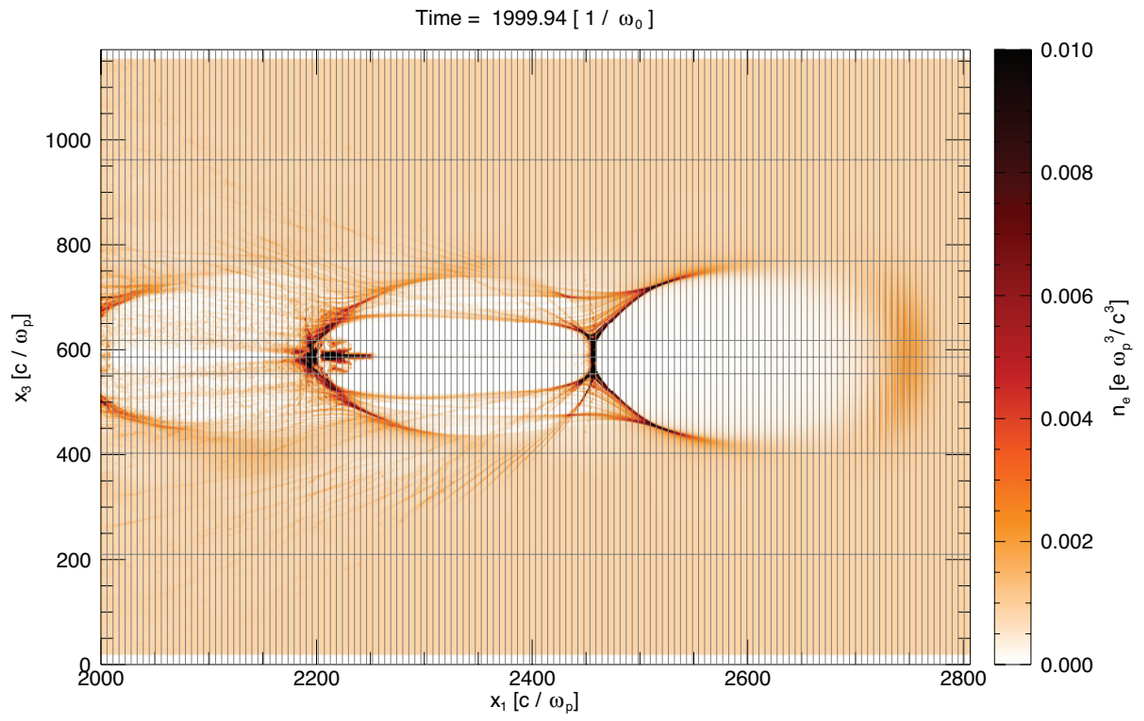
- the volume handled by each group of cores is much larger,
- the probability for significant load imbalance will be lower

- particle pusher, the field solver, current smoother, boundary processing of particles / fields and particle sorting.

- fairly simple since routines generally consist of an external loop that can be easily split among threads

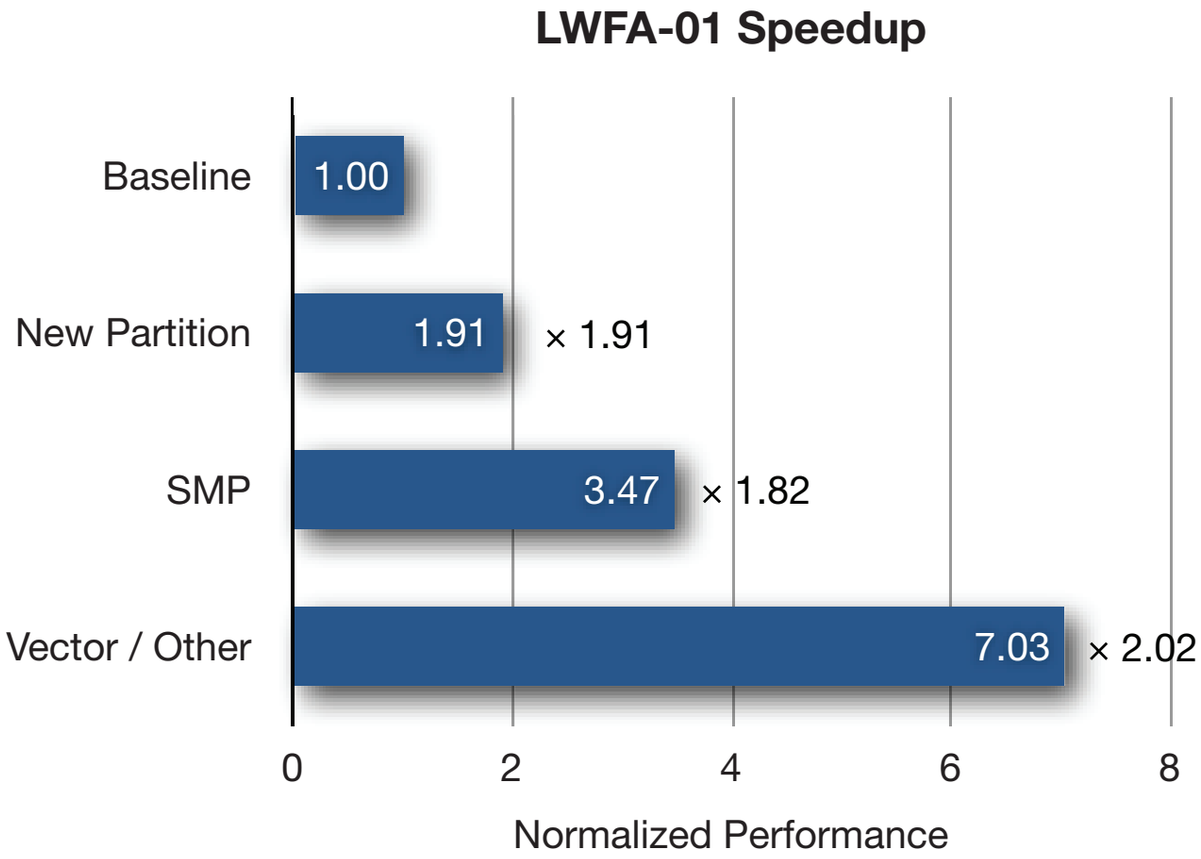
- reduced the total node communication volume
- threads per MPI process must match the number of cores per cpu -or less





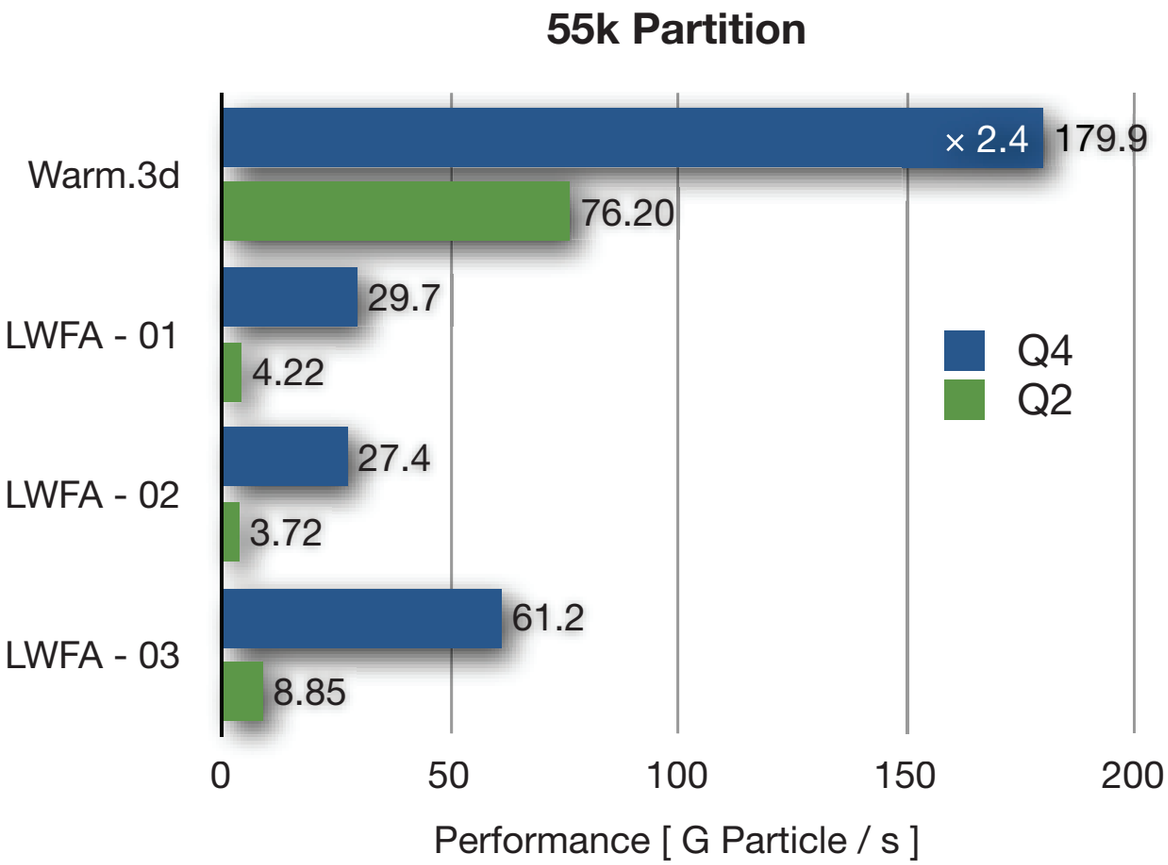
OSIRIS : Particle Injection in Laser Wakefield

Run	Partition [cores]	Performance [G part/s]	Push Time [μs]	Average Imbalance	TFLOPS	INS/FP	Speedup
Warm.3d	55296	179.95	0.307	1.00	169.92	1.28	2.36
LWFA - 01		29.66	1.864	3.64	31.18	6.39	7.03
LWFA - 02		27.43	2.016	4.75	28.02	7.69	7.37
LWFA - 03		61.20	0.903	2.31	58.25	3.84	6.92
Frozen.3d linear	221184	1463.52	0.151	1.00	516.92	1.34	n / a
Frozen.3d quadratic		784.04	0.282	1.00	736.12	1.20	n / a
Warm.3d weak scale		741.20	0.298	1.00	700.09	1.21	9.73
Warm.3d strong scale		719.80	0.307	1.00	679.68	1.28	9.45
LWFA - 01 - strong scale		70.91	3.119	4.66	76.55	9.48	16.80



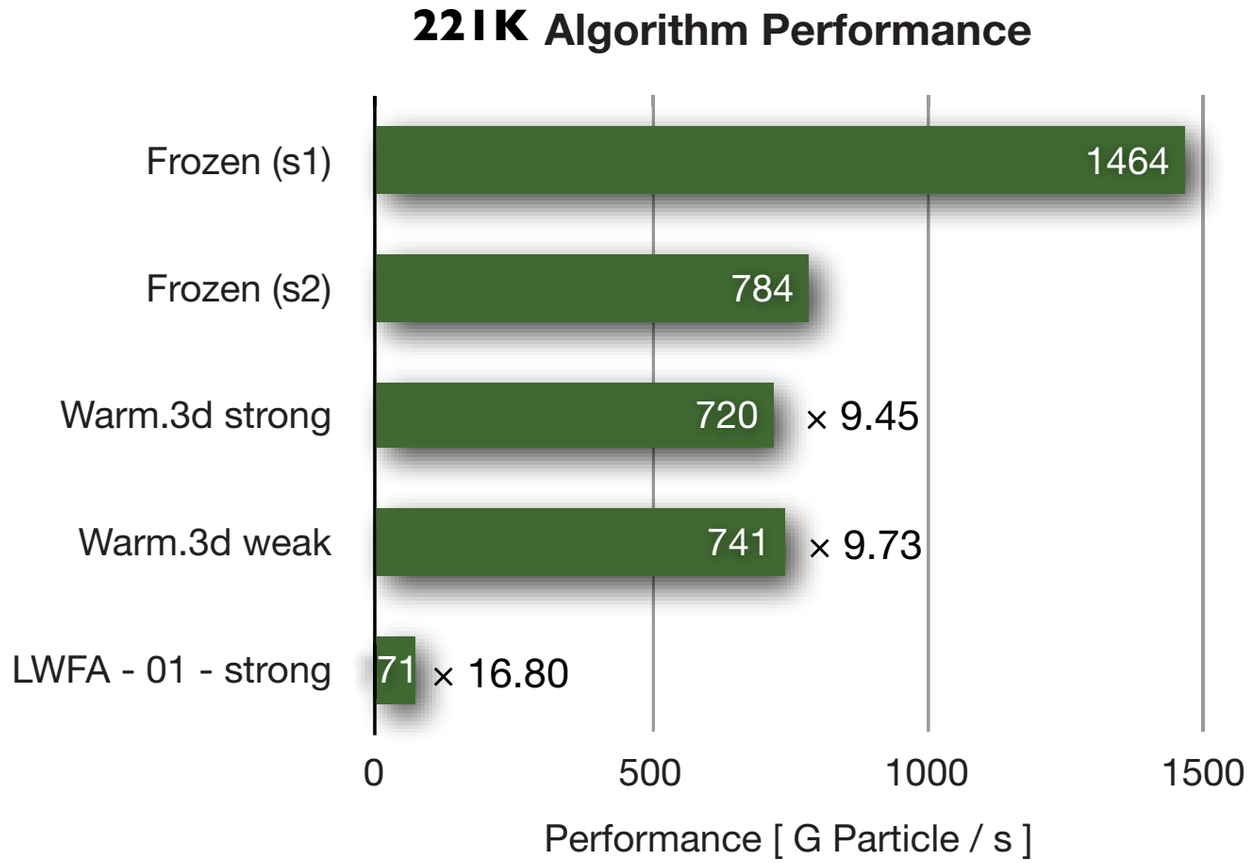
OSIRIS : Particle Injection in Laser Wakefield

Run	Partition [cores]	Performance [G part/s]	Push Time [μs]	Average Imbalance	TFLOPS	INS/FP	Speedup
Warm.3d	55296	179.95	0.307	1.00	169.92	1.28	2.36
LWFA - 01		29.66	1.864	3.64	31.18	6.39	7.03
LWFA - 02		27.43	2.016	4.75	28.02	7.69	7.37
LWFA - 03		61.20	0.903	2.31	58.25	3.84	6.92
Frozen.3d linear	221184	1463.52	0.151	1.00	516.92	1.34	n / a
Frozen.3d quadratic		784.04	0.282	1.00	736.12	1.20	n / a
Warm.3d weak scale		741.20	0.298	1.00	700.09	1.21	9.73
Warm.3d strong scale		719.80	0.307	1.00	679.68	1.28	9.45
LWFA - 01 - strong scale		70.91	3.119	4.66	76.55	9.48	16.80



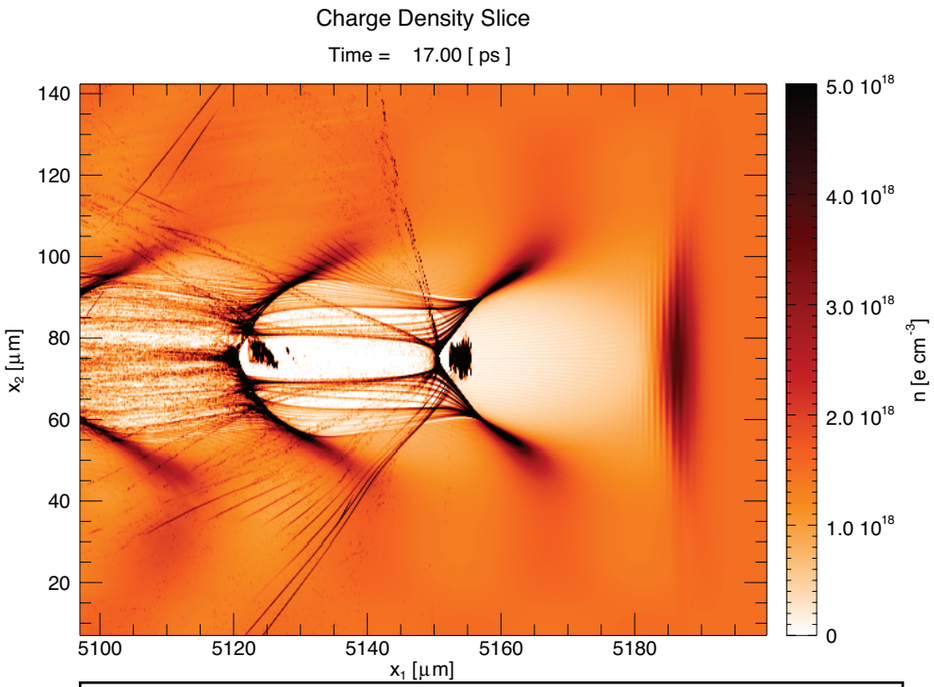
OSIRIS : Particle Injection in Laser Wakefield

Run	Partition [cores]	Performance [G part/s]	Push Time [μs]	Average Imbalance	TFLOPS	INS/FP	Speedup
Warm.3d	55296	179.95	0.307	1.00	169.92	1.28	2.36
LWFA - 01		29.66	1.864	3.64	31.18	6.39	7.03
LWFA - 02		27.43	2.016	4.75	28.02	7.69	7.37
LWFA - 03		61.20	0.903	2.31	58.25	3.84	6.92
Frozen.3d linear	221184	1463.52	0.151	1.00	516.92	1.34	n / a
Frozen.3d quadratic		784.04	0.282	1.00	736.12	1.20	n / a
Warm.3d weak scale		741.20	0.298	1.00	700.09	1.21	9.73
Warm.3d strong scale		719.80	0.307	1.00	679.68	1.28	9.45
LWFA - 01 - strong scale		70.91	3.119	4.66	76.55	9.48	16.80

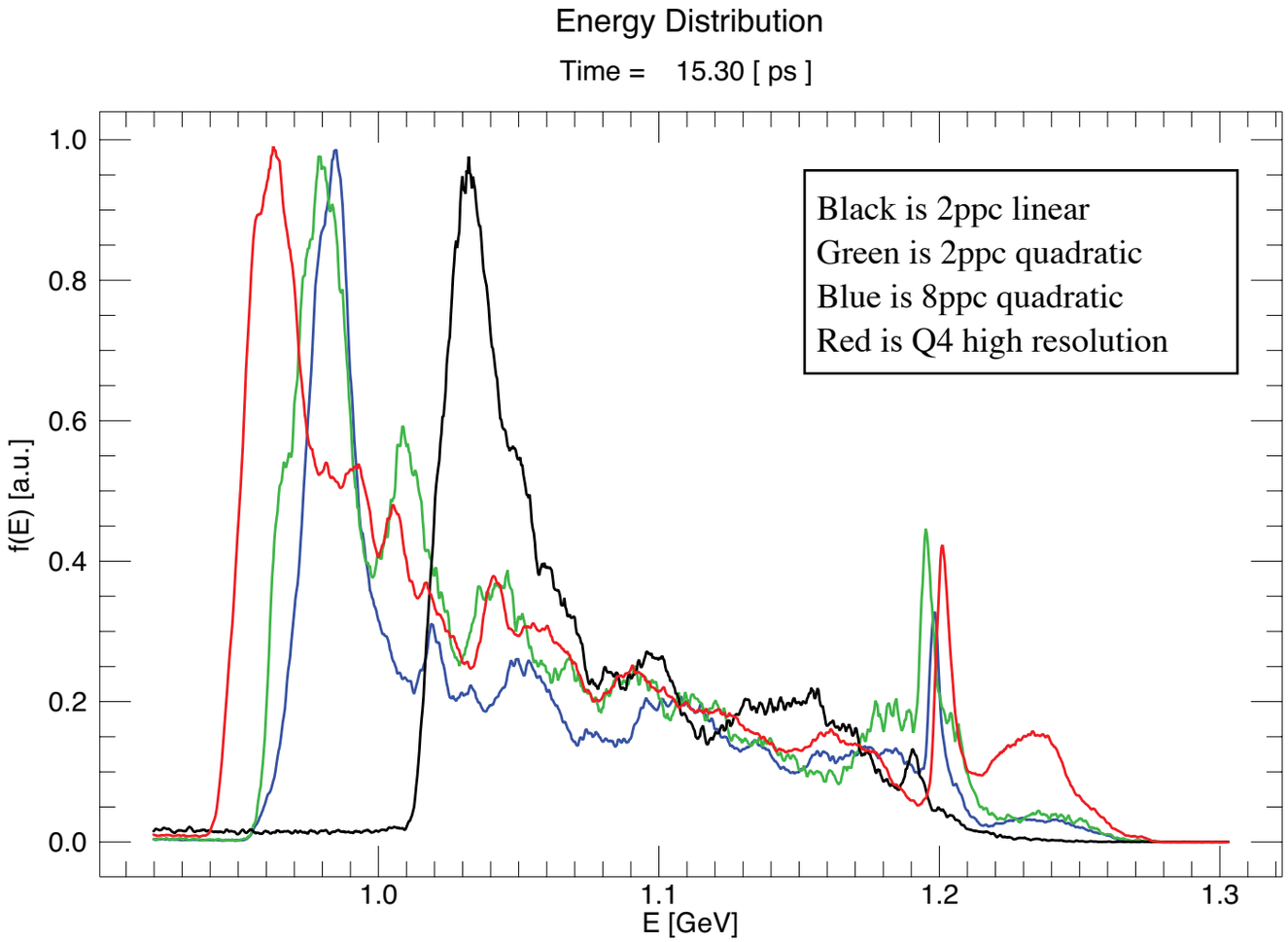


OSIRIS : Particle Injection in Laser Wakefield

	2ppc Linear	2ppc Quad	8ppc Quad	Q4 HR
Charge [pC]	284	339	347	366
Avg. Ene [MeV]	1074.7	1052.6	1054.7	1048.1
StdDev Ene [MeV]	53.4	76.3	75.6	85.5
Peak Ene [MeV	1031.8	979.0	984.5	962.6
Ene FWHM [MeV]	34.5	50.1	19.8	44.5
ϵ_{Ny} mm mr	29.6	26.7	28.8	19.2
ϵ_{Nz} mm mr	33.9	30.7	25.6	18.6



A 2D slice of the electron density showing the electrons injected into the first two buckets.



Comparison of the energy spectra of the beam in the first bucket for the runs.

- Charge (the linear particle shape run has 25% less charge) and the emittance are significantly reduced in the higher resolution (Q4) run.

- The high resolution run has 50% lower RMS value for the two transverse planes.

- This improvement in emittance is very important for both collider and light source applications.

eSTOMP : (extreme scale) Subsurface Transport Over Multiple Phases

Detailed numerical simulations of complex subsurface systems:

- addressing the spatially varying material properties required for predictions
- address long simulation periods (e.g., 10,000 year period for spent nuclear fuel regulatory compliance)
- comprehensive treatment of coupled processes (e.g., geology, hydrology, biology, chemistry)
- resolution of multiscale variability in material properties
- uncertainties in conceptual process models and parameters

Diverse models:

- molecular-scale models of mineral surface reactions
- in silico models of metal reducing bacteria
- pore-scale models of multiphase flow and multicomponent reactive transport
- simulations of multiphase flow and multicomponent reactive transport

Large range of subsurface environments:

- nonisothermal conditions
- fractured media
- multiple-phase systems
- nonwetting fluid entrapment
- soil freezing conditions
- nonaqueous phase liquids
- biogeochemical reactions
- radioactive decay
- solute transport
- dense brines
- nonequilibrium dissolution
- surfactant-enhanced dissolution and mobilization of organics

Benchmark Problem: uranium bioremediation

18m x 20m x 6.3m , 180 x 200 x 63 = 2,268,000 grid cells , lattice spacing of .1m

300 time steps, 1 simulated day, checkpoint each 6 sim hours (0,6,12,18,24)

5 lithofacies,102 biogeochemical species, 7 mineral reaction network

eSTOMP : enhancements

Memory

• per core memory demand limited the number of computational cores per node to 4

  • 52,800 process job had to allocate, but not use, an additional 105,600 processor cores

• software was modified to use only distributed arrays for the chemical species eliminated the temporary allocation of 102 field arrays in local memory

  •modification resulted in 1.72 GB less memory usage per core

• improved the on chip-processor utilization by a factor of three; permitted the problem to be solved with smaller processor counts.

I/O

1) `nga_put` move local data to a global array
2) `nga_sync` insist on completion of 'put's
3) `nga_get` head node gets the data in 4 Nx Ny buffered chunks

• reduction `ga_dgop` (essentially a wrapper for MPI_Reduce) was faster than the sequence of the GA `get`, `sync`, and `put`

• buffer size changed to entire field

• remove ASCII formatting

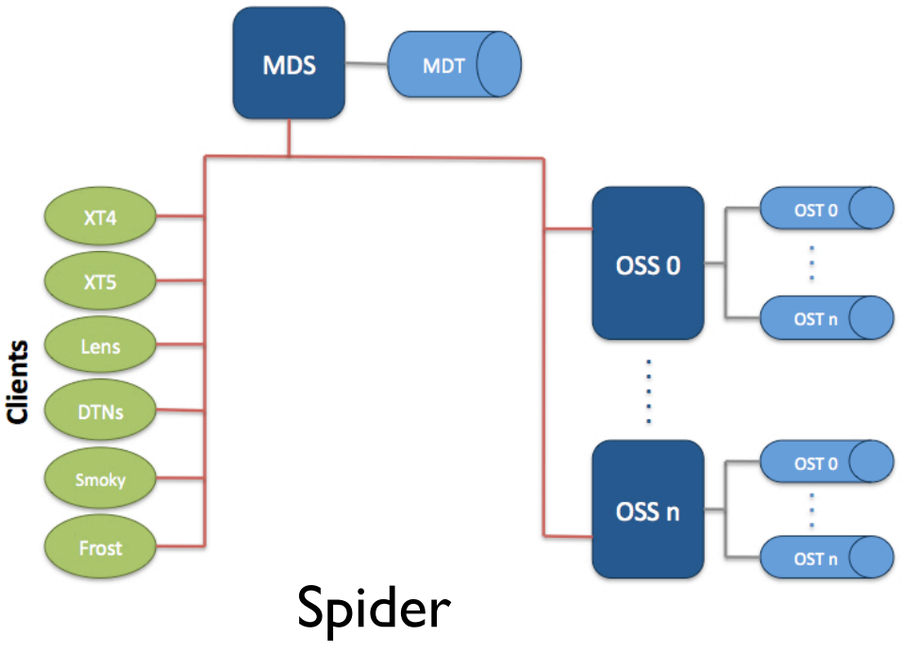
ASCII	Binary
3253456688	2739744000

	Q2	Q4	Q2 : Q4
PEs	158,400	26,400	6
INS	2.11e+18	2.619e+17	8.0665
FP OPs	378976827416438	416965910025780	0.9089
L2 DCM	19491864496712	13739811264029	1.4186
Time[s]	12278.840 (11620.592)	4798.922 (4554.582)	2.5587
Cost [CPU Hrs]	540,269	35,192	15.352

eSTOMP : Aside on FILEs and I/O

Dominant I/O Demands for Checkpoint / Restart and Data Dumps :

- the magnitude of data to be moved to disk is small 2,739,744,000 BYTEs, or 2.55 GB per big event
- 5 events at 0, 6, 12, 18, 24 Hrs in simulation time
- 180 × 200 × 63 spatial lattice distributed over some virtual process volume (ie, 52800 ~ 60 × 44 × 20 in Q2)
- 151 double precision fields defined at each lattice point in space
  - 136 double precision fields and 2 integer fields are written per check point
  - 15 double precision fields for plotting



Spider

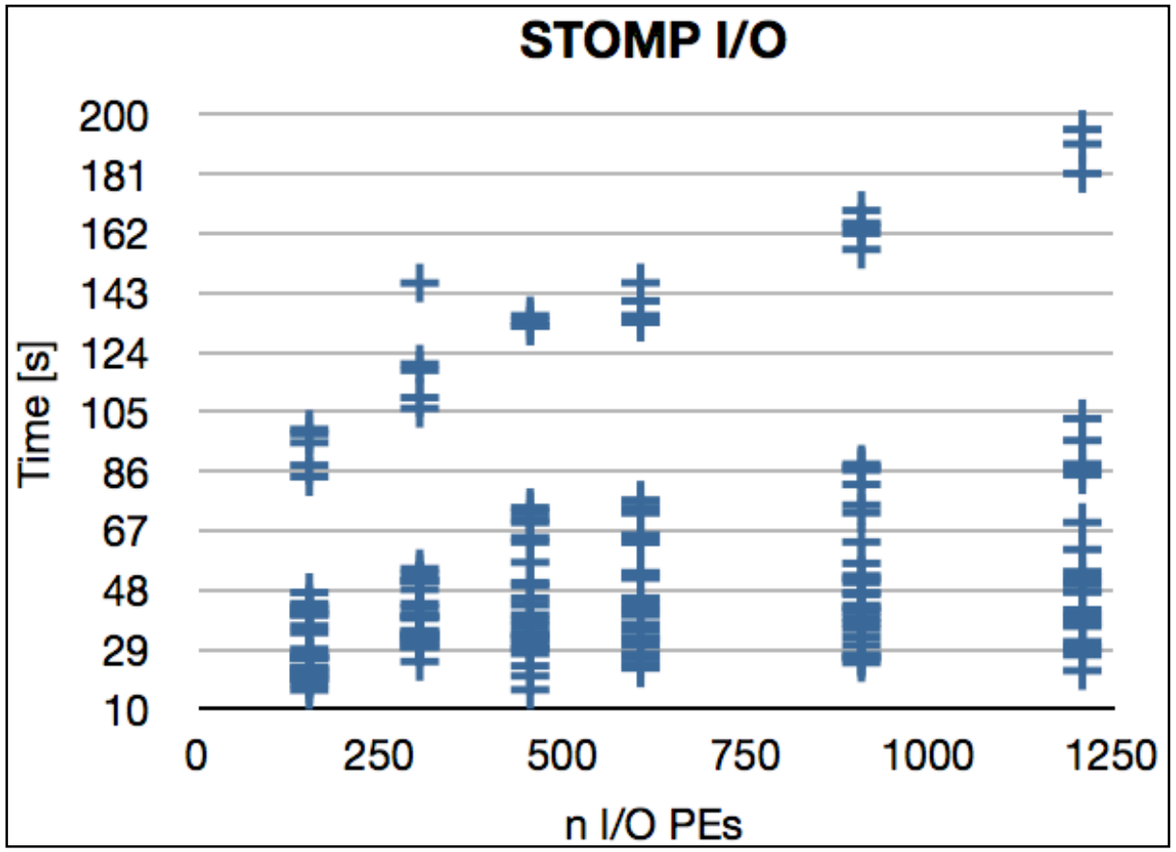
- form modulo classes from MPI communicator over the number of I/O groups
  - new algorithm delivers 2 levels of parallelism:
    - over the fields
    - spatial decomposition w/ correct indexing
- fit the stripe size to the largest single data item if possible
- set the stripe pattern (I use round-robin) and number of target OSTs for target PATH / FILE
  - lfs setstripe /tmp/work/roche/kio -s 2m -i -1 -c 88

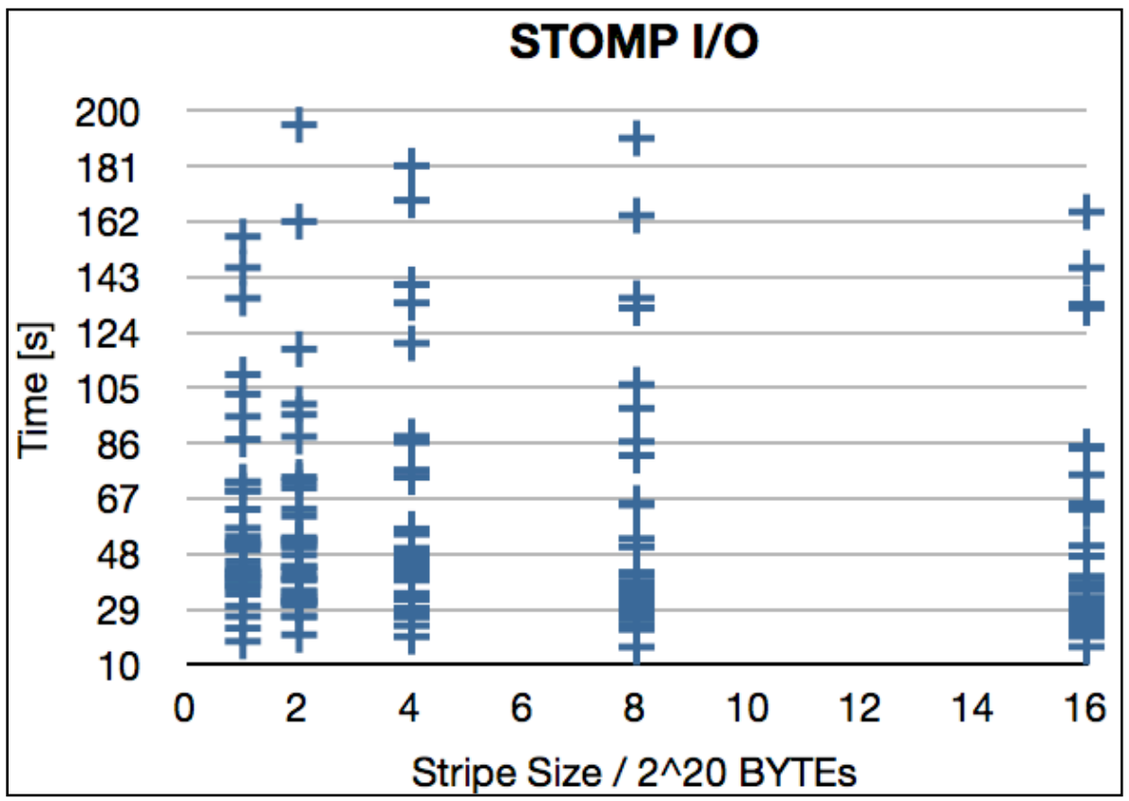
eSTOMP : Lustre - Oracle Study

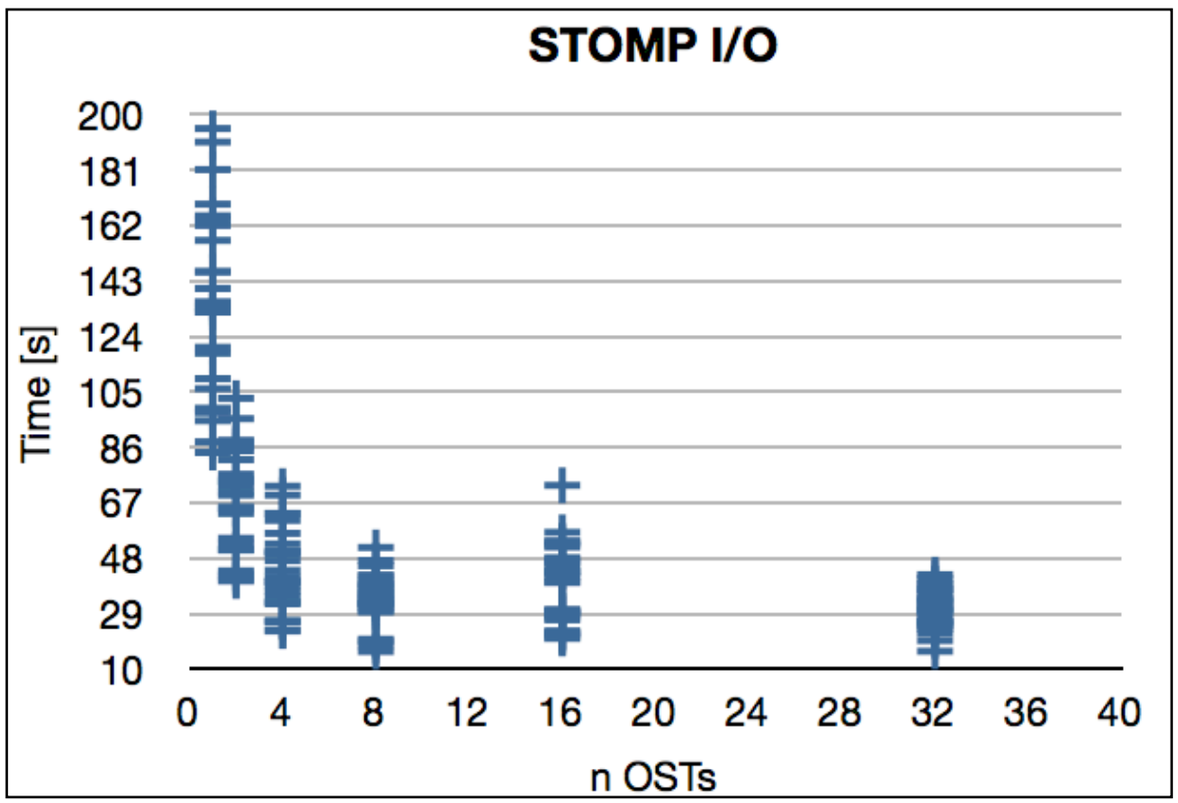
Parameters set in the file system related to but independent from the problem parameters:

- Number of OSTs
  1, 2, 4, 8, 16, 32
- Stripe size in BYTEs
  1 MB, 2 MB, 4MB, 8 MB, 16 MB
- access pattern (round robin)
- Number of I/O PEs for spatial decomposition
  kio ~ 1, 2, 3, 4, 6, 8
- Total number of I/O PEs is kio * nfld
  since nfld =151, 151, 302, 453, 604, 906, 1208

```
module load liblut ; -LUT
lut__open() ;
lut__close() ;
lut_putl() ;
pwrite() ;
pread() ;
```







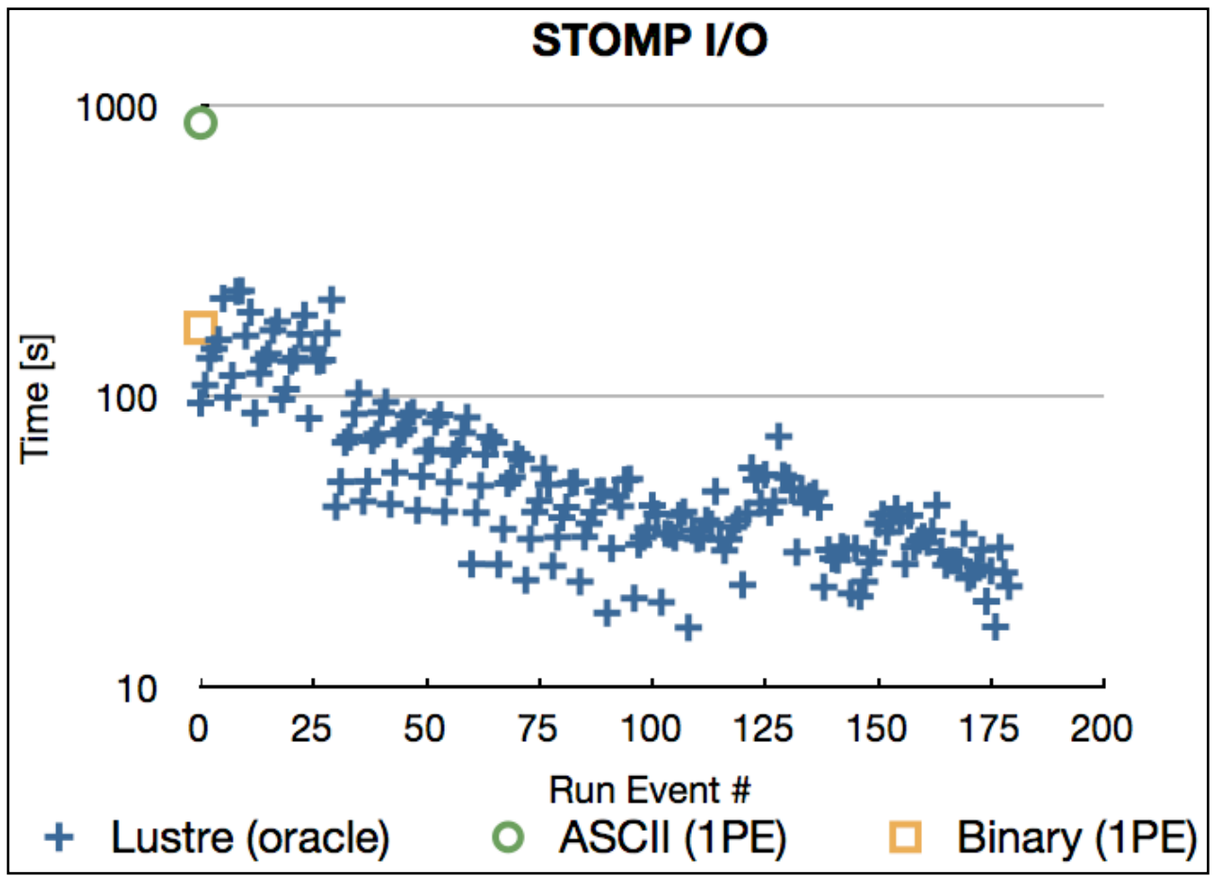
eSTOMP :

Lustre - based I/O prototype

Stand-alone MPI test codes that exactly mimick STOMP's GA-based I/O are compared

Parameter sets that completed all the Q2 problem I/O in less than 20s are printed

Best version of the new algorithm is 54.5X faster than the ASCII version and 10.7X faster than the best binary version (POSIX ~ 15 MBps , use of LUSTRE ~ 814 MBps)



Lustre-based STOMP I/O			
Time[s]	Stripe Size [B]	n OSTs	n I/O PEs
16.064482	8388608	8	151
16.156173	16777216	32	453
18.029697	1048576	8	151
19.627054	4194304	8	151
19.776406	16777216	32	151
STOMP ASCII Unformatted I/O			
875.422139			1
STOMP Binary Formatted I/O			
172.573337			1

FY12 Application Nominations

1) **QMCPACK**, Jeongnim Kim (kimj1@ornl.gov), Oak Ridge National Laboratory

QMCPACK implements continuum quantum Monte Carlo (QMC) methods for predicting the properties of matter from first principles.

2) **Drekar::CFD,** John Shadid (jnshadi@sandia.gov), Roger Pawlowski, Sandia National Laboratory

Drekar::CFD is a next-generation massively parallel multi-physics simulation code that is being developed for computational simulations of turbulent fluid flow and heat transfer in nuclear fission reactor-cores.

3) **NIMROD**, Carl Sovinec (csovinec@cae.wisc.edu), U. Wisconsin

The NIMROD code (http://nimrodteam.org) is a flexible computational tool for numerical studies of extended magnetohydrodynamics, which includes MHD, two-fluid plasma modeling, minority ion kinetics, and nonlocal parallel kinetics.

4) **Materials Project**, Gerbrand Ceder (gceder@MIT.EDU), MIT

Invent machine learning techniques to mine chemical bonding knowledge from the tens of thousands of relaxations that have already been executed in the Materials Project (www.materialsproject.org) and use that knowledge to a) pre-optimize any input cell to VASP, and b) transform the Cartesian coordinate system used in VASP into a more relevant coordinate that reflects the variance these coordinates have on the total energy.

ASCR's Benchmark Trends (FY04 - FY11)

climate research	4
condensed matter	4
fusion	5
high energy physics	3
nuclear	2
subsurface modeling	2
astrophysics	2
combustion chemistry	4
bioinformatics	1
math, data analytics	2
molecular dynamics, electronic structure	3
nuclear energy	1
Total	33

Cray	X1
	X1E
	XT3
	XT4
4-core	XT5
6-core	XT5
IBM	SP Power3
	P690
	Power5
	BG/L
SGI	Altix
HP Itanium-2	
QCDOC	
Intel / NVIDIA	w/ IB

**DOE's Advanced Scientific Computing Advisory Committee approves annual application / machine studies

Benchmark Aggregated CPU Hours

Fiscal Year*	Benchmark CPU-Hours
2005	24,814
2006	211,888
2007	314,459
2008	2,718,788
2009	39,300,189
2010	78,289,735
2011	56,208,435

*FY04 numbers are available but unreliable

Fiscal Year	CPU-Hours Awarded
2010	150M
2011	100M + Dirac at NERSC

Remaining Time Goes to Applications for Production

Floating Point Intensity of DOE Mission Applications: Are We Really Dominated by FLOPs?

Application	1	2	3	4	5	6	7	8
Instructions Retired	1.99E+15	8.69E+17	1.86E+19	2.45E+18	1.24E+16	7.26E+16	8.29E+18	2.67E+18
Floating Point Ops	3.52E+11	1.27E+15	1.95E+18	2.28E+18	6.16E+15	4.15E+15	3.27E+17	1.44E+18
INS / FP_OP	5.64E+03	6.84E+02	9.56	1.08	2.02	17.5	25.3	1.85

REFERENCE FLOATING POINT INTENSE PROBLEM :: Dense Matrix Matrix Multiplication

C <--- a A B + b C :: OPERATIONAL COMPLEXITY : A[m,n] , B[n,p] , C[m,p] :: [8mpn + 13mp] FLOP

E.g. m=n=p=1024 ---> 8603566080 FLOP , measure 8639217664