SCEC Application Performance and Software Development

Yifeng Cui [2],

T. H. Jordan [1], K. Olsen [3], R. Taborda [4], J. Bielak [5], P. Small [6], E. Poyraz [2], J. Zhou [2], P. Chen [14], E.-J. Lee [1], S. Callaghan [1], R. Graves [7], P. J. Maechling [1], D. Gill [1], K. Milner [1], F. Silva [1], S. Day [3], K. Withers [3], W. Savran [3], Z. Shi [3], M. Norman [8], H. Finkel [9], G. Juve [10], K. Vahi [10], E. Deelman [10], H. Karaoglu [5], Y. Isbiliroglu [11], D. Restrepo [12], L. Ramirez-Guzman [13]

[1] Southern California Earthquake Center, [2] San Diego Supercomputer Center, [3] San Diego State Univ., [4] Univ. Memphis, [5] Carnegie Mellon Univ., [6] Univ. Southern California, [7] U.S. Geological Survey, [8] Oak Ridge Leadership Computing Facility, [9] Argonne Leadership Computing Facility, [10] Information Science Institute, [11] Paul C. Rizzo Associates, Inc., [12] Universidad Eafit, [13] National Univ. Mexico, [14] Univ. Wyoming

OLCF Symposium, 22 July 2014

SCEC Computational Pathways

Hybrid MPI/CUDA

1. Standard seismic hazard analysis
2. Ground motion simulation (AWP-ODC, Hercules, RWG)
3. Dynamic rupture modeling (SORD)
4. Ground-motion inverse problem (AWP-ODC, SPECFEM3D)

AWP-ODC – Yifeng Cui and Kim Olsen

Hercules – Jacobo Bielak and Ricardo Tarbora

SORD – Steven Day

CyberShake - Scott Callaghan

OpenSHA/UCERF3 - Kevin Milner

UCVM, CVM-H - David Gill

Broadband - Fabio Silva

AWP-ODC

- Started as personal research code (Olsen 1994)
- 3D velocity-stress wave equations

$$\partial_t v = \frac{1}{\rho}\nabla\cdot\sigma \quad \partial_t\sigma = \lambda(\nabla\cdot v)\mathrm{I} + \mu(\nabla v + \nabla v^{\mathrm{T}})$$

solved by explicit staggered-grid 4th-order FD

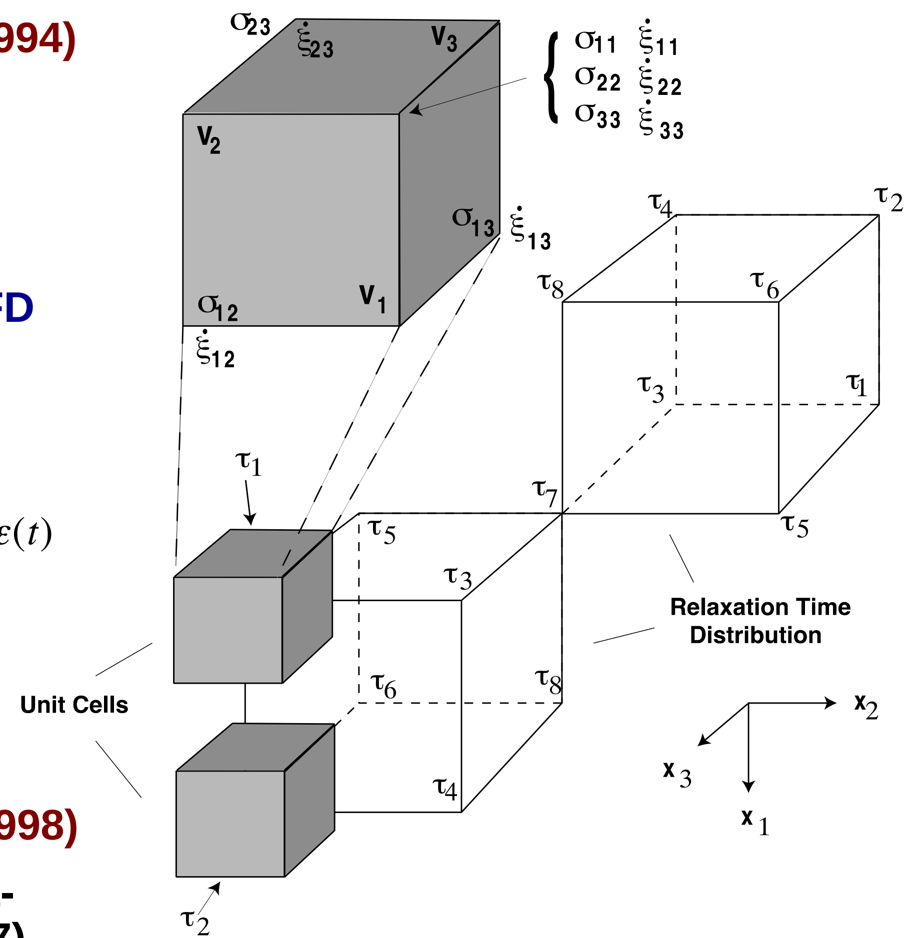
- Memory variable formulation of inelastic relaxation

$$\sigma(t) = M_u\left[\varepsilon(t) - \sum_{i=1}^{N}\varsigma_i(t)\right] \qquad \tau_i\frac{d\varsigma_i(t)}{dt} + \varsigma_i(t) = \lambda_i\frac{\delta M}{M_u}\varepsilon(t)$$

$$Q^{-1}(\omega) \approx \frac{\delta M}{M_u}\sum_{i=1}^{N}\frac{\lambda_i\omega\tau_i}{\omega^2\tau_i^2+1}$$

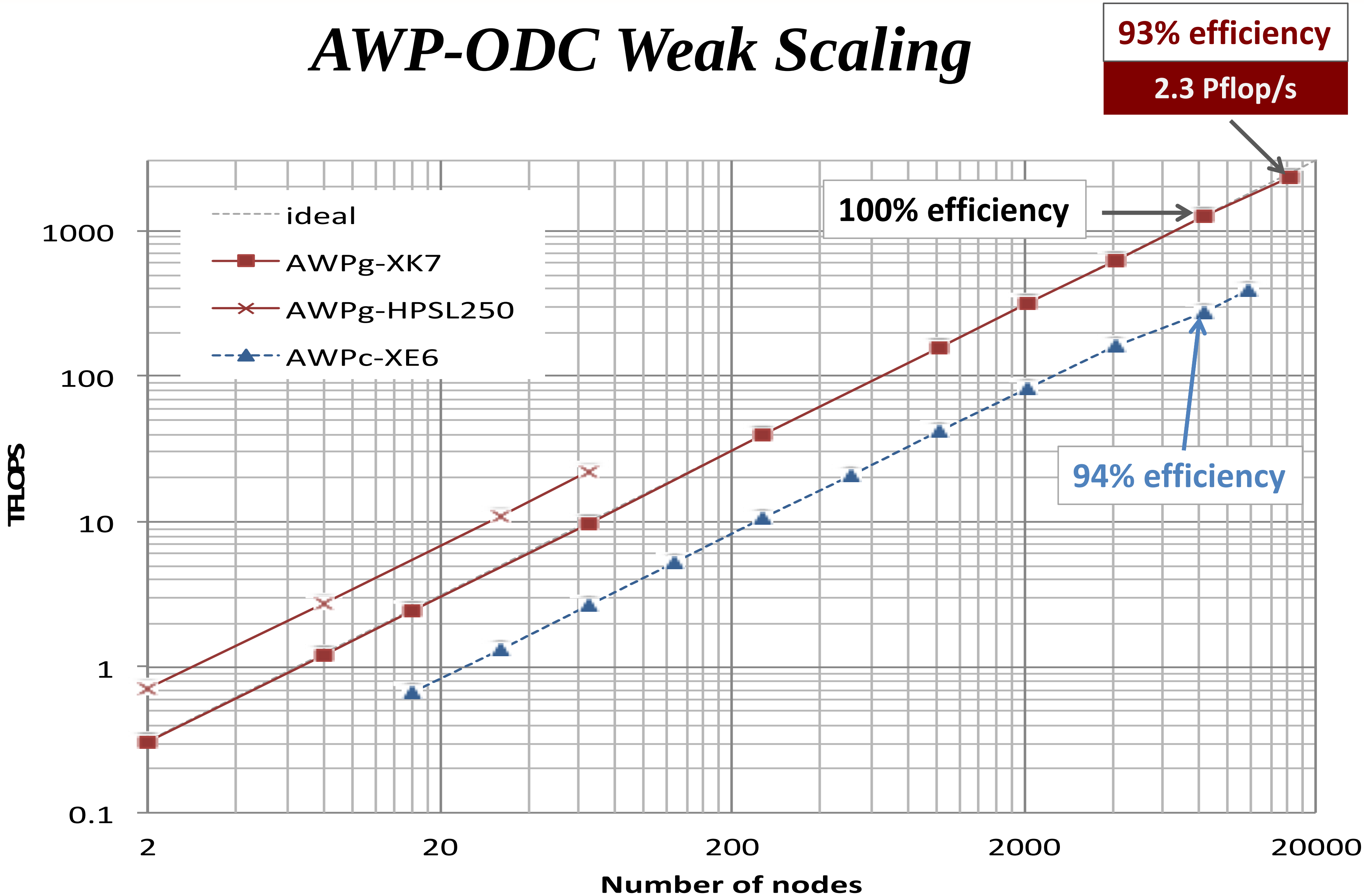
using coarse-grained representation (Day 1998)

- Dynamic rupture by the staggered-grid split-node (SGSN) method (Dalguer and Day 2007)
- Absorbing boundary conditions by perfectly matched layers (PML) (Marcinkovich and Olsen 2003) and Cerjan et al. (1985)



Inelastic relaxation variables for memory-variable ODEs in AWP-ODC

AWP-ODC Weak Scaling



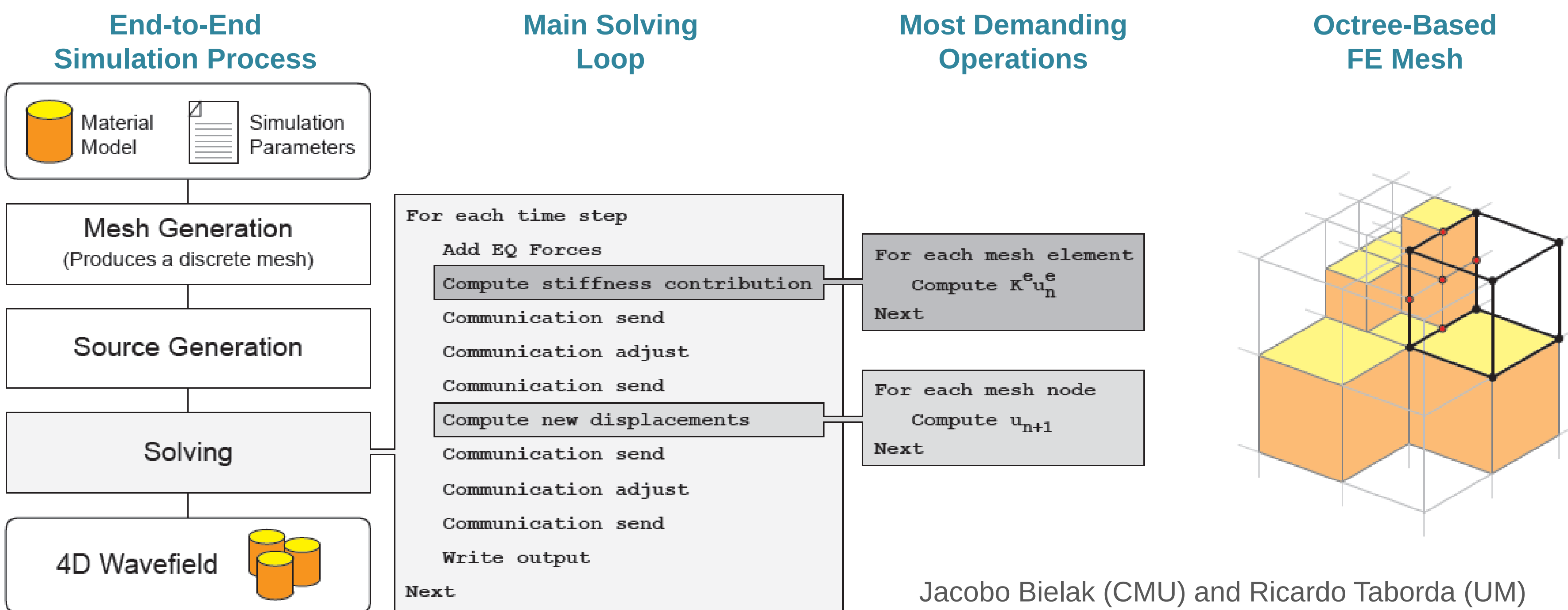
(Cui et al., 2013)

Hercules – General Architecture

Southern California Earthquake Center

- » Finite-Element Method
- » Integrated Meshing (unstructured hexahedral)
- » Uses and octree-based library for meshing and to order elements and nodes in memory
- » Explicit FE solver
- » Plane wave approximation to absorbing boundary conditions
- » Natural free surface condition
- » Frequency Independent Q

Hercules was developed by the Quake Group at Carnegie Mellon University with support from SCEC/CME projects. Its current developers team include collaborators at the National University of Mexico, the University of Memphis, and the SCEC/IT team among others.



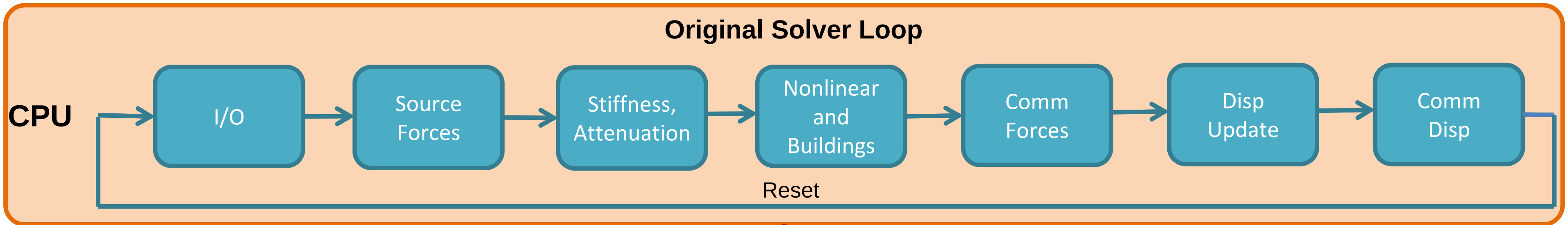
Jacobo Bielak (CMU) and Ricardo Taborda (UM)
See Refs. Taborda et al. (2010) and Tu et al. (2006)

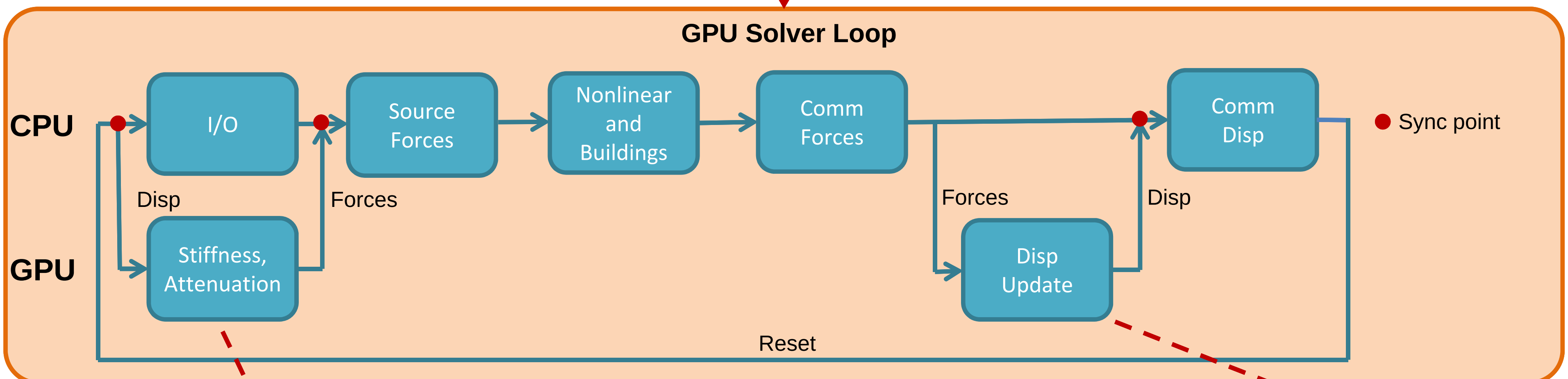
SCEC Southern California Earthquake Center

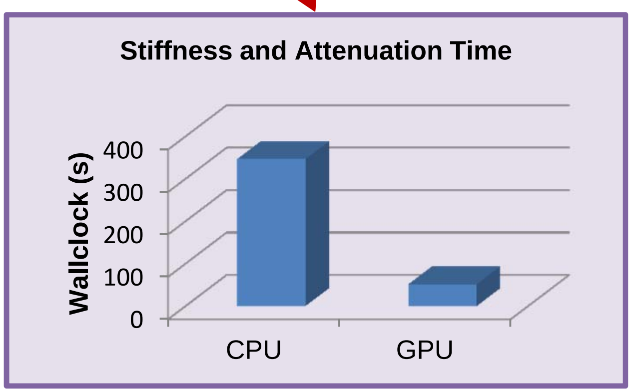
Hercules on Titan – GPU Module Implementation

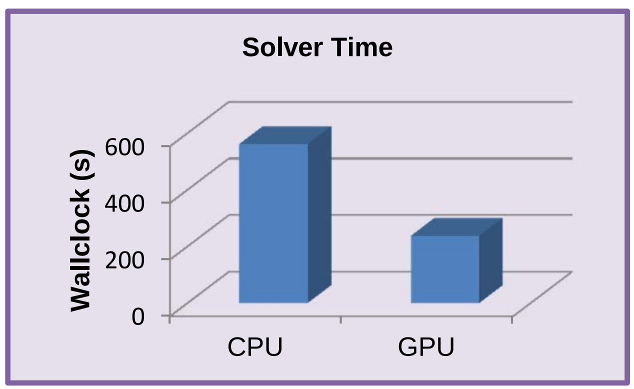
Modifications to Solver Loop

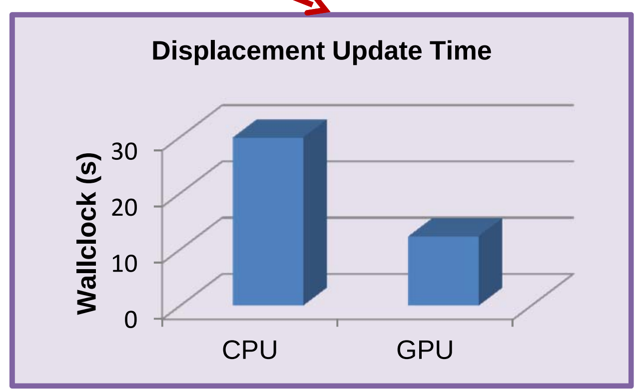
Chino Hills 2.8 Hz, BKT damping, 1.5 B elements, 2000 time steps (512 compute nodes)











(Patrick Small of USC and Ricardo Taborda of UM, 2014)

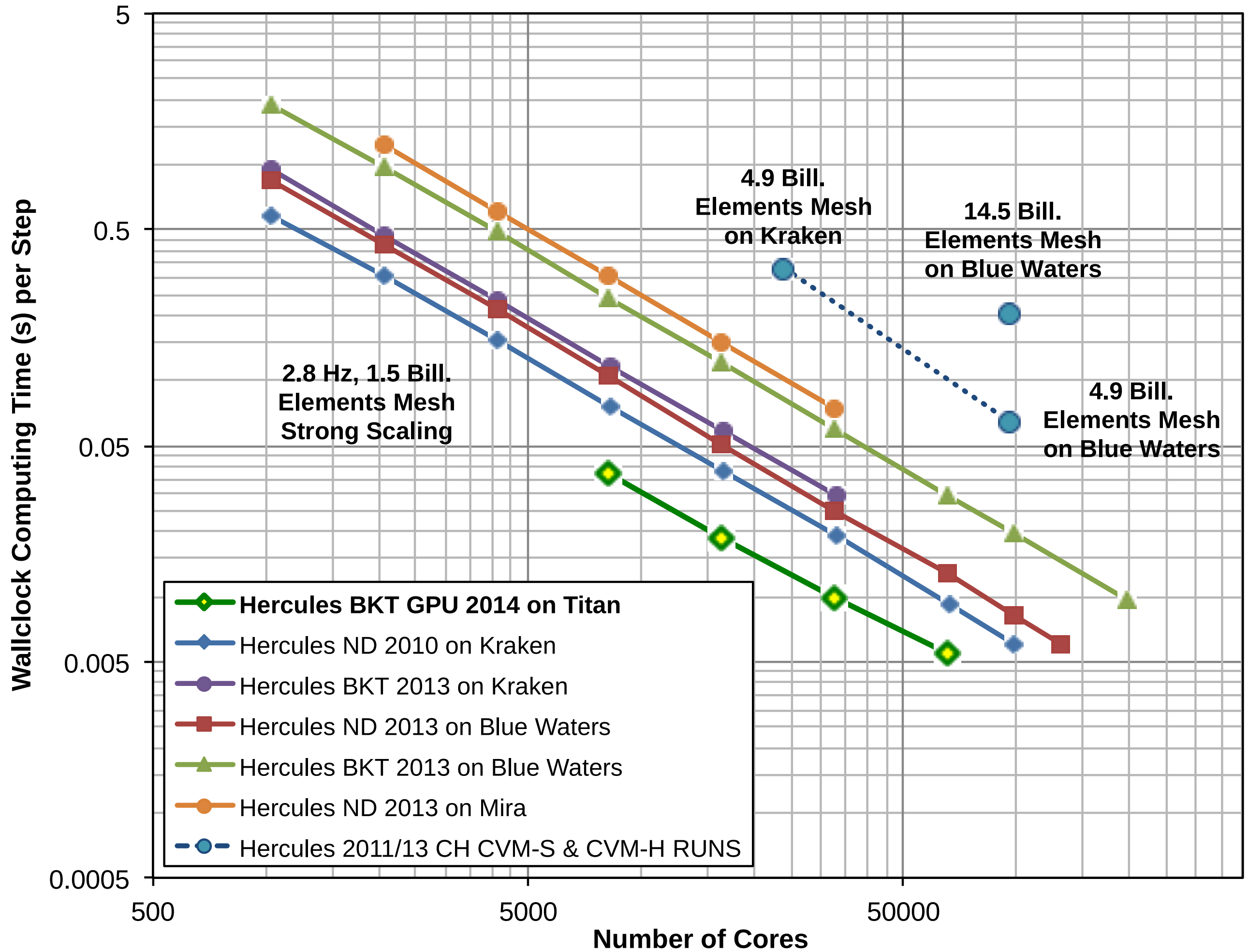
SCEC

Southern California Earthquake Center

Hercules on Titan – GPU Performance

Initial Strong Scaling Tests on Titan (in green) compared to other systems

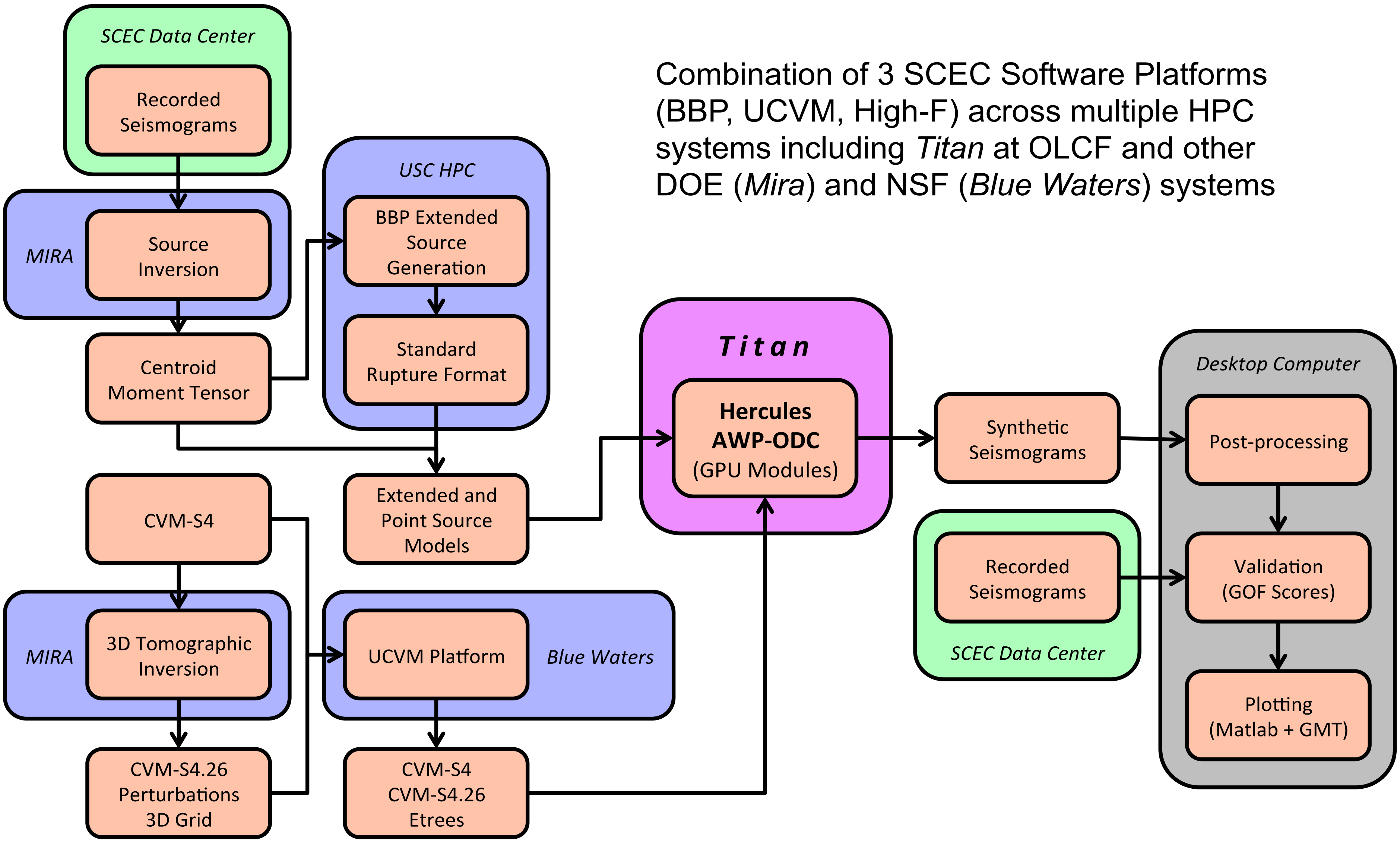
- Recent Hercules developments include GPU capabilities using CUDA
- Performance tests for a benchmark 2.8 Hz Chino Hills simulation show near perfect strong and weak scalability on multiple HPC systems including TITAN using GPU
- The acceleration ratio of the GPU code with respect to the CPU is of a factor of 2.5x overall



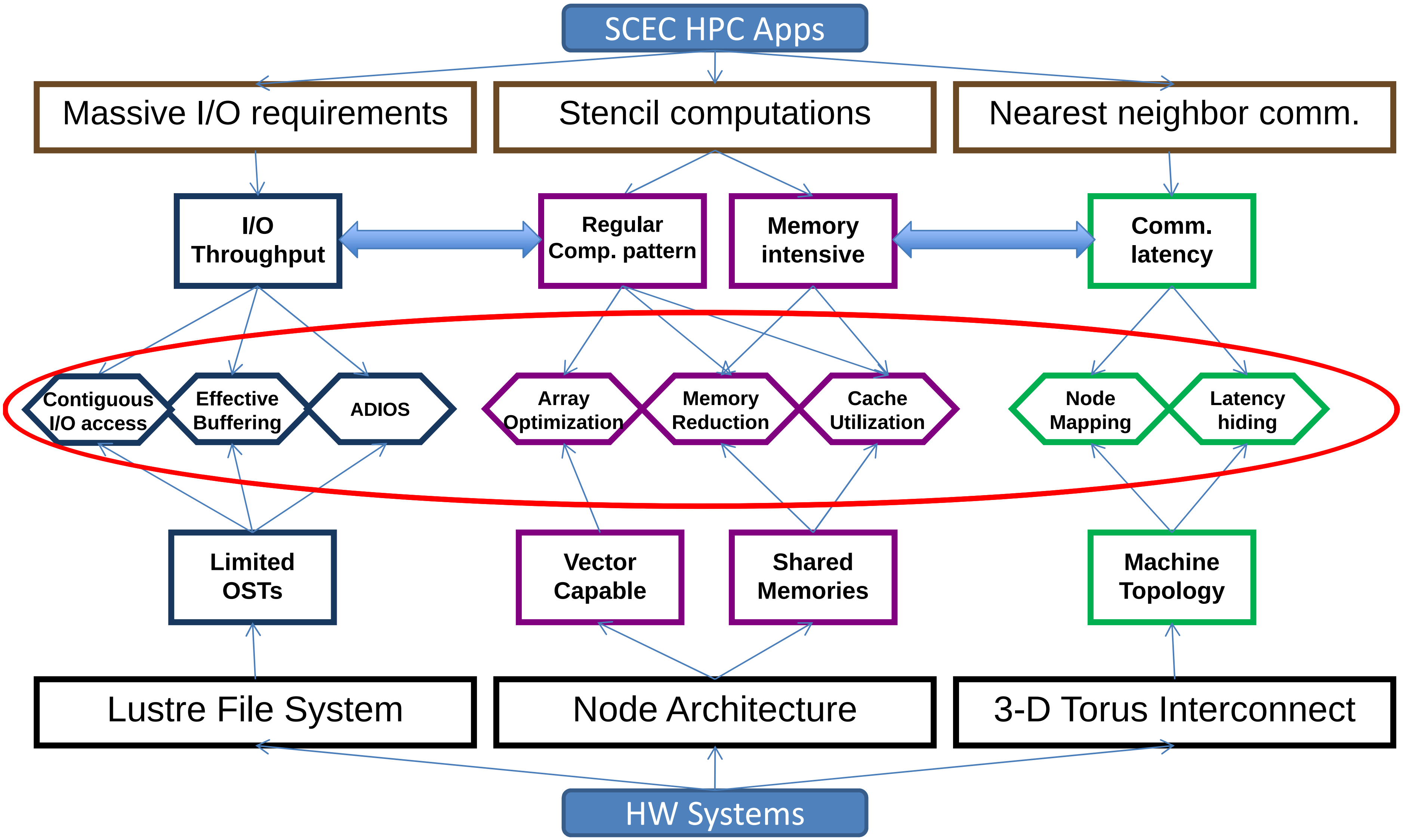
(Jacobo Bielak of CMU, Ricardo Taborda of UM and Patrick Small of USC, 2014)

La Habra Validation Experiment

Combination of 3 SCEC Software Platforms (BBP, UCVM, High-F) across multiple HPC systems including *Titan* at OLCF and other DOE (*Mira*) and NSF (*Blue Waters*) systems

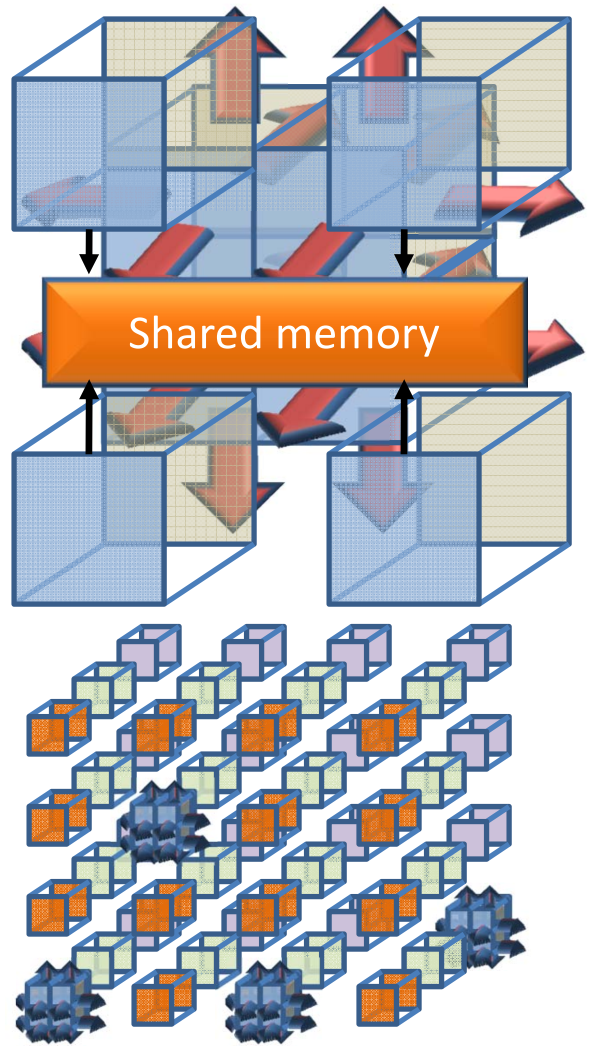


Algorithms and Hardware Attributes



AWP-ODC Communication Approach on Jaguar

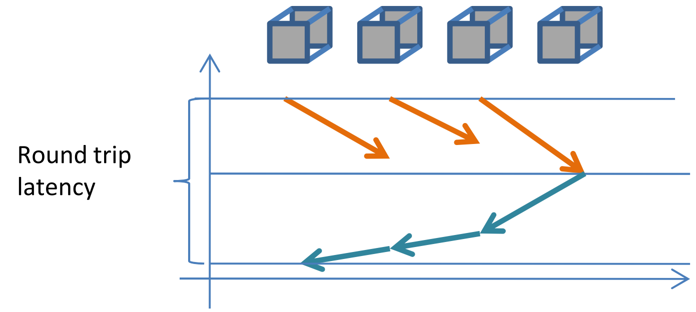
- **Rank placement technique**
  - Node filling with X-Y-Z orders
  - Maximizing intra-node and minimizing inter-node communication

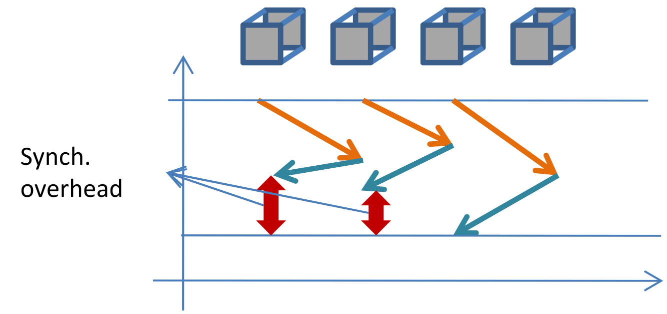


(Joint work with Zizhong Chen of CSM)

AWP-ODC Communication Approach on Jaguar

- Rank placement technique
  - Node filling with X-Y-Z orders
  - Maximizing intra-node and minimizing inter-node communication
- **Asynchronous communication**
  - **Significantly reduced latency through local communication**
  - **Reduced system buffer requirement through pre-post receives**

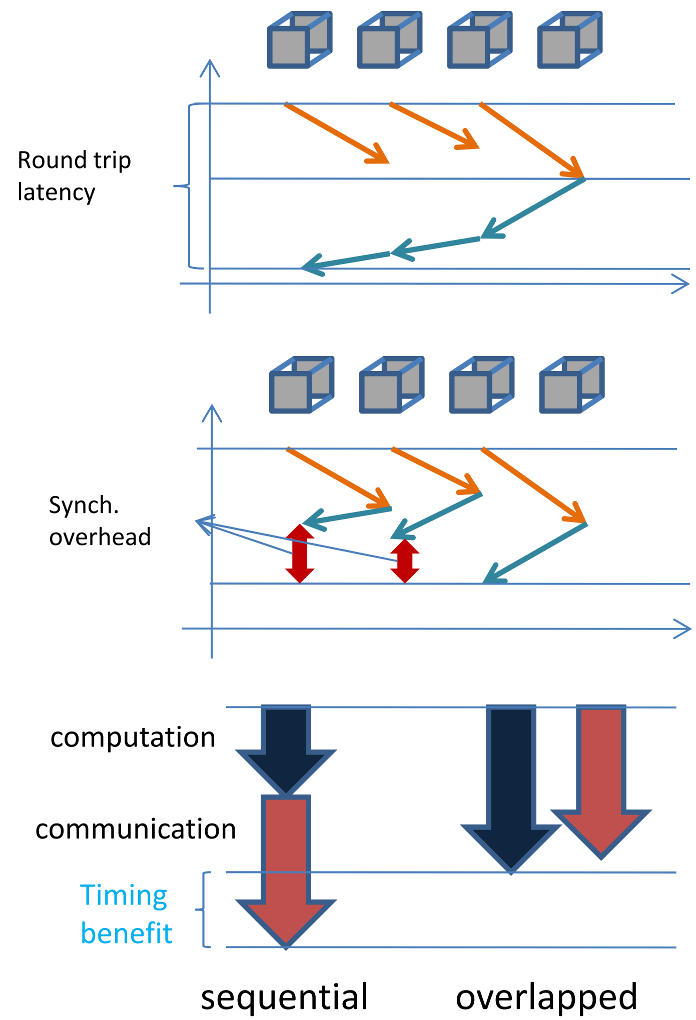




(Joint work with DK Panda Team of OSU)

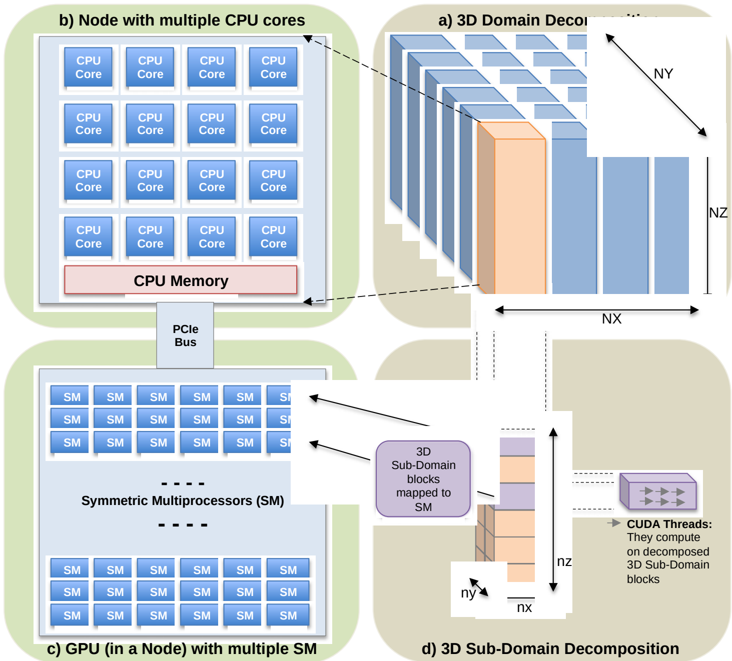
AWP-ODC Communication Approach on Jaguar

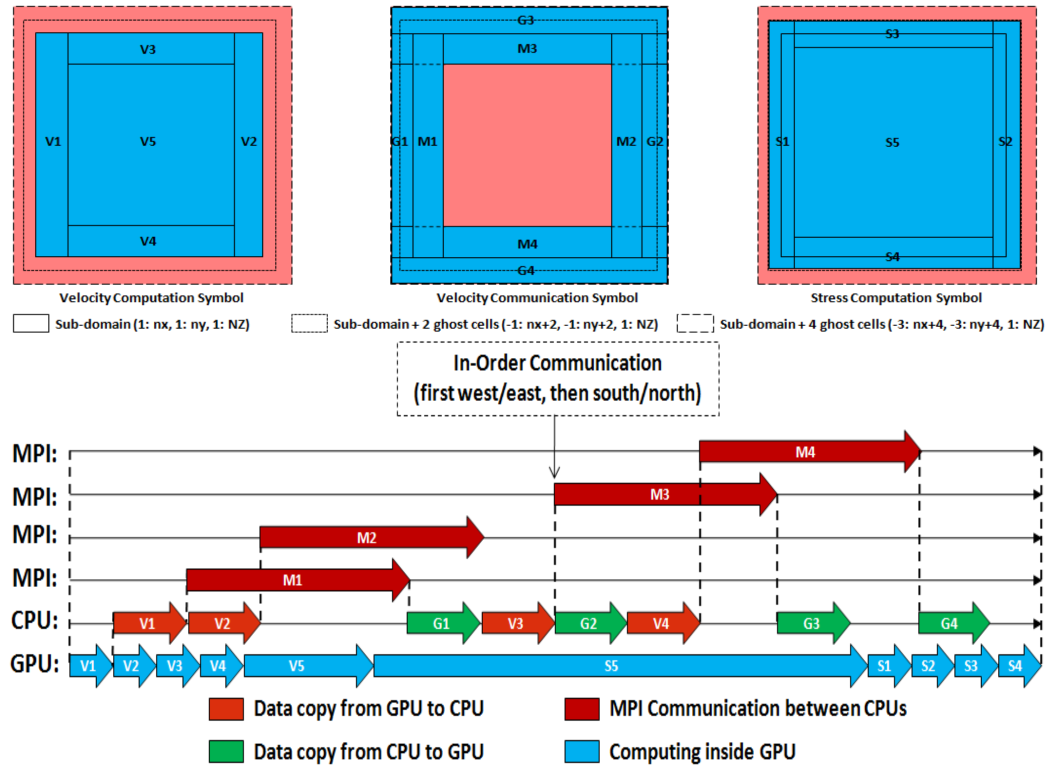
- Rank placement technique
  - Node filling with X-Y-Z orders
  - Maximizing intra-node and minimizing inter-node communication
- Asynchronous communication
  - Significantly reduced latency through local communication
  - Reduced system buffer requirement through pre-post receives
- **Computation/communication overlap**
  - **Effectively hide computation times**
  - **Effective when *Tcompute_hide>Tcompute_overhead***
  - **One-sided Communications (on Ranger)**



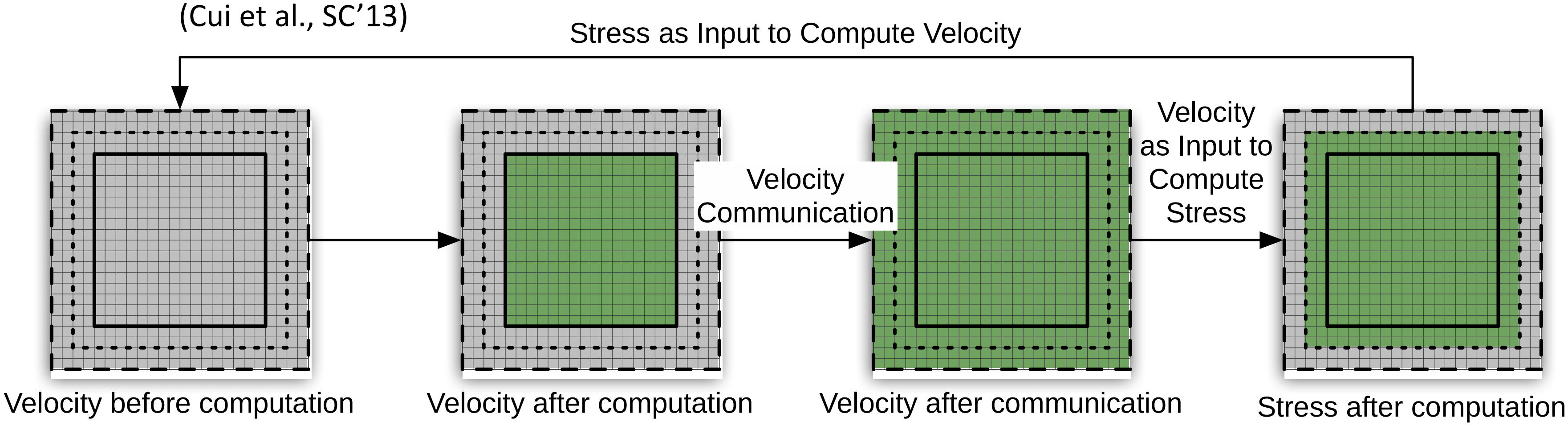
(Joint work with DK Panda Team of OSU)

AWP-ODC Communication Approach on Titan

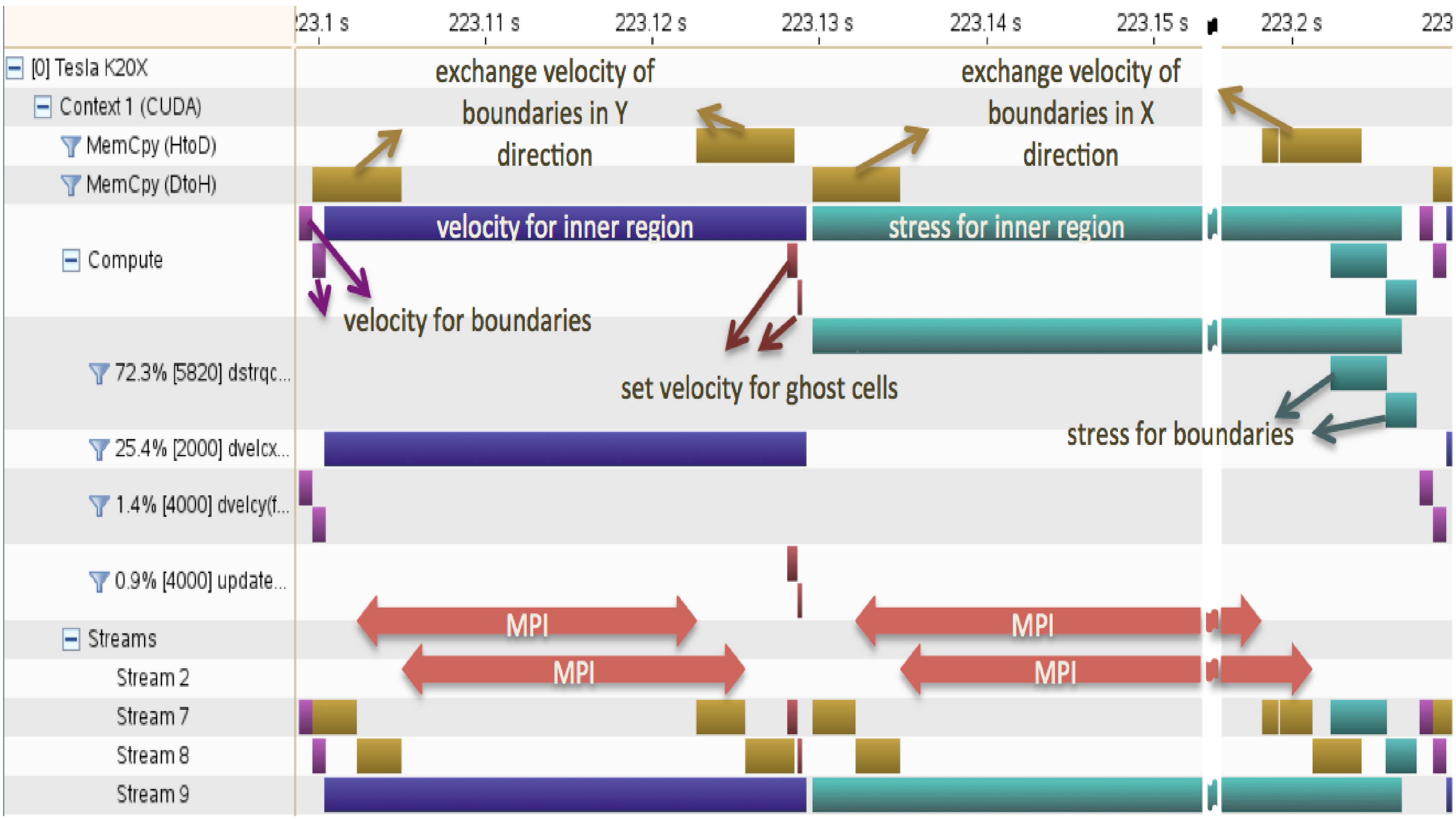




(Cui et al., SC'13)

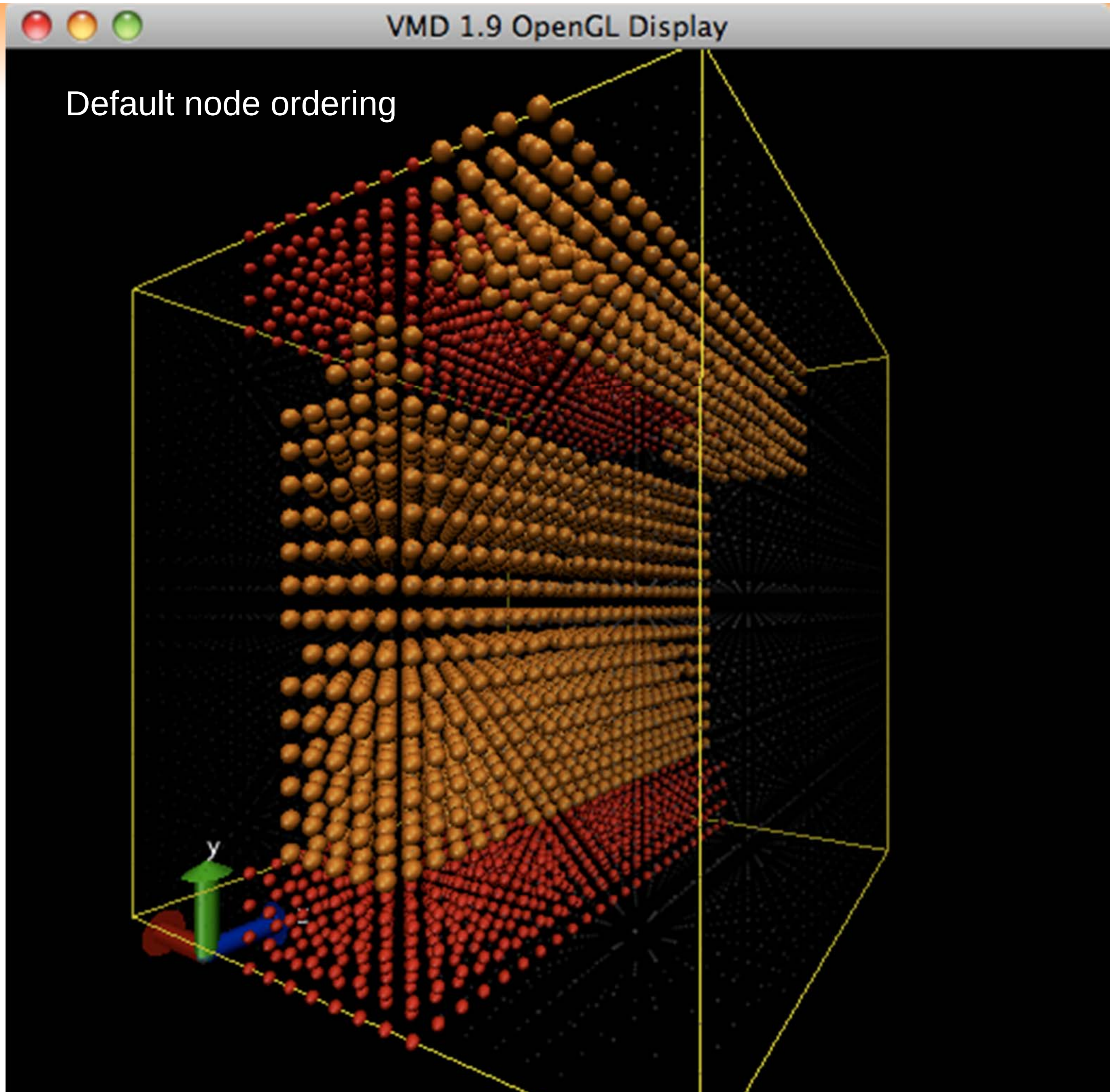


nvvp Profiling



Topology Tuning on XE6/XK7

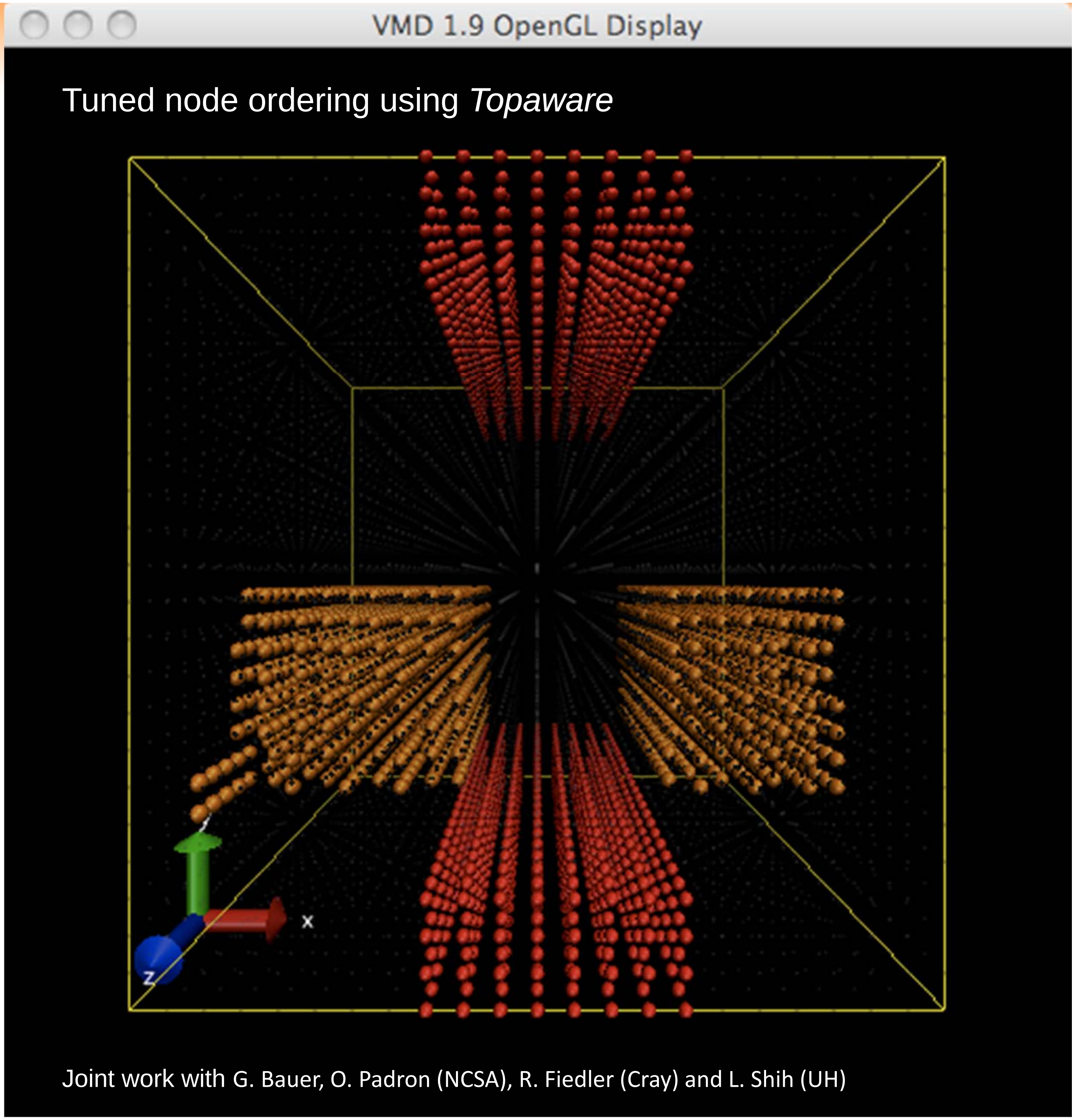
- Matching the virtual 3D Cartesian to an elongated physical subnet prism shape
- Maximizing faster connected BW *XZ* plane allocation
- Obtaining a tighter, more compact and cuboidal shaped BW subnet allocation
- Reducing inter-node hops along the slowest BW torus *Y* direction



Joint work with G. Bauer, O. Padron (NCSA), R. Fiedler (Cray) and L. Shih (UH)

Topology Tuning on XE6/XK7

- **Matching the virtual 3D Cartesian to an elongated physical subnet prism shape**
- **Maximizing faster connected BW *XZ* plane allocation**
- **Obtaining a tighter, more compact and cuboidal shaped BW subnet allocation**
- **Reducing inter-node hops along the slowest BW torus *Y* direction**



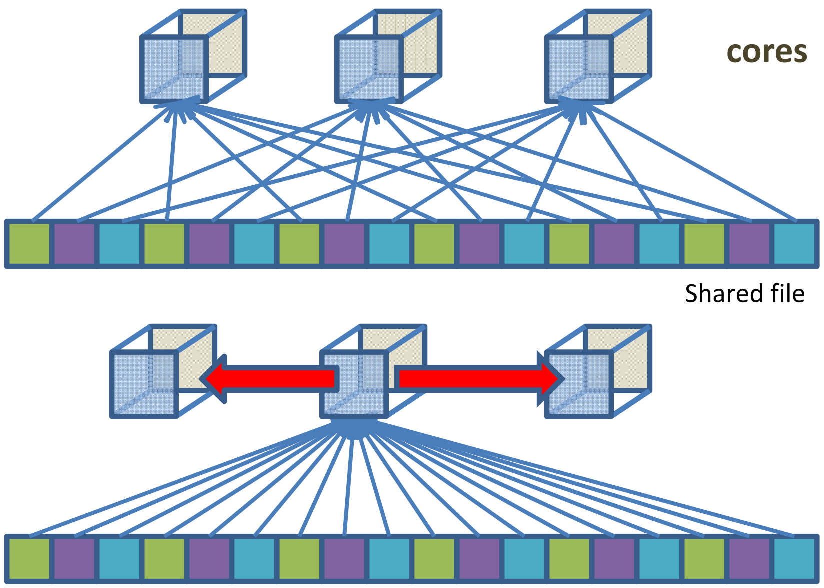
Topology Tuning on XE6/XK7

- **Matching the virtual 3D Cartesian to an elongated physical subnet prism shape**
- **Maximizing faster connected BW *XZ* plane allocation**
- **Obtaining a tighter, more compact and cuboidal shaped BW subnet allocation**
- **Reducing inter-node hops along the slowest BW torus *Y* direction**

# nodes	Default	Topaware	Speedup	Efficiency
64	4.006	3.991	0.37%	100%
512	0.572	0.554	3.15%	87.5%->90%
4096	0.119	0.077	35.29%	52.6%->81%

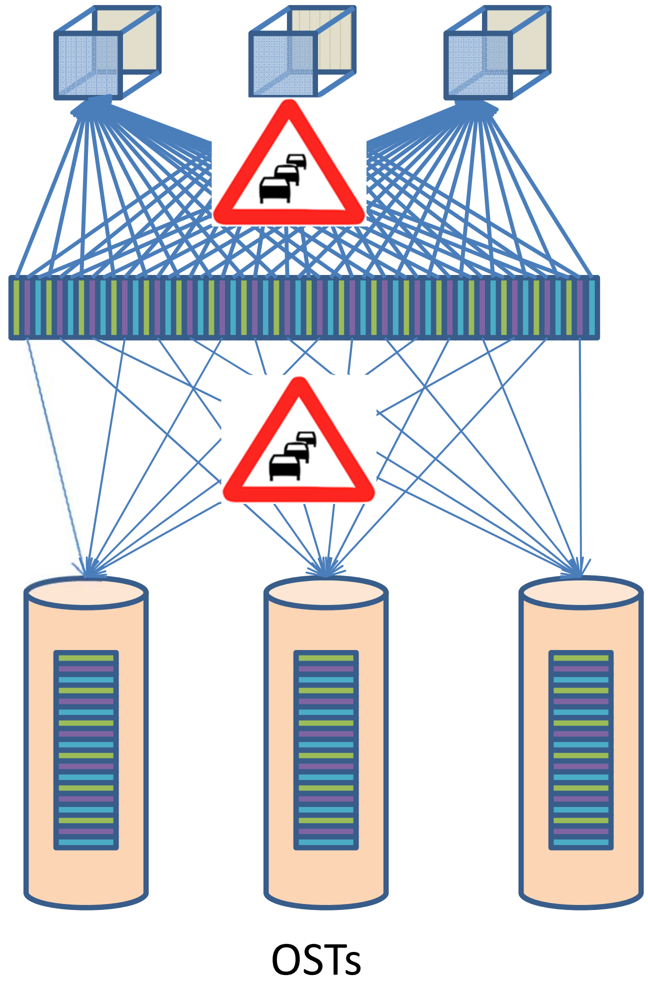
Two-layer I/O Model

- **Parallel I/O**
  - **Read and redistribute multiple terabytes inputs (19 GB/s)**
    - **Contiguous block read by reduced number of readers**
    - **High bandwidth asynchronous point-to-point communication redistribution**



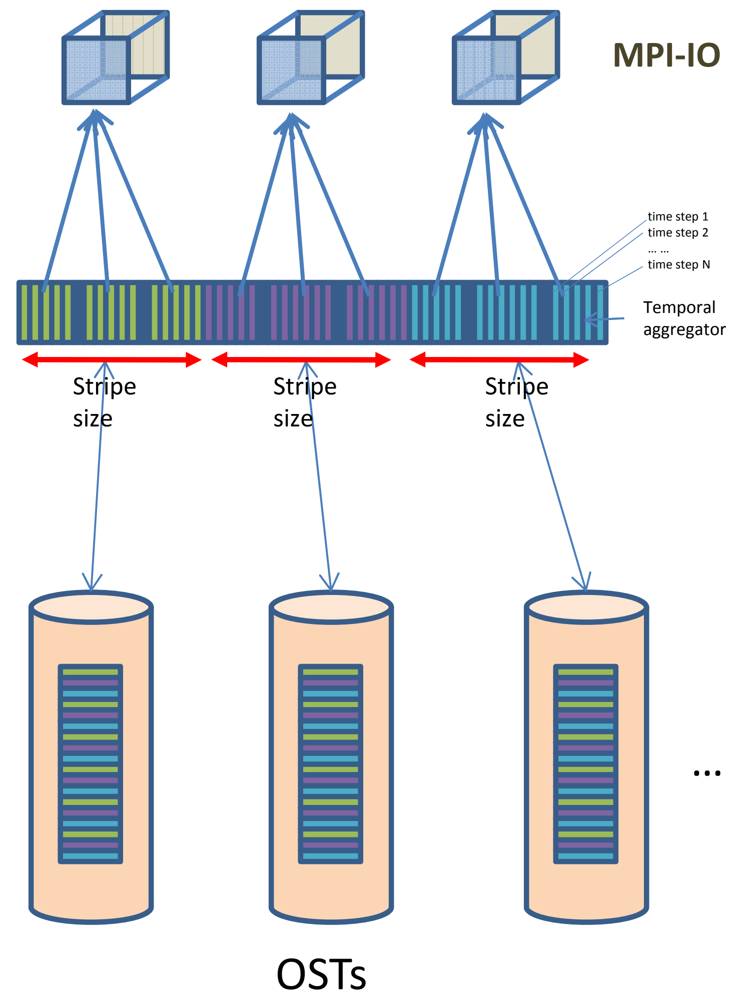
Two-layer I/O Model

- Parallel I/O
  - Read and redistribute multiple terabytes inputs (19 GB/s)
    - Contiguous block read by reduced number of readers
    - High bandwidth asynchronous point-to-point communication redistribution
- **Aggregate and write (10GB/s)**



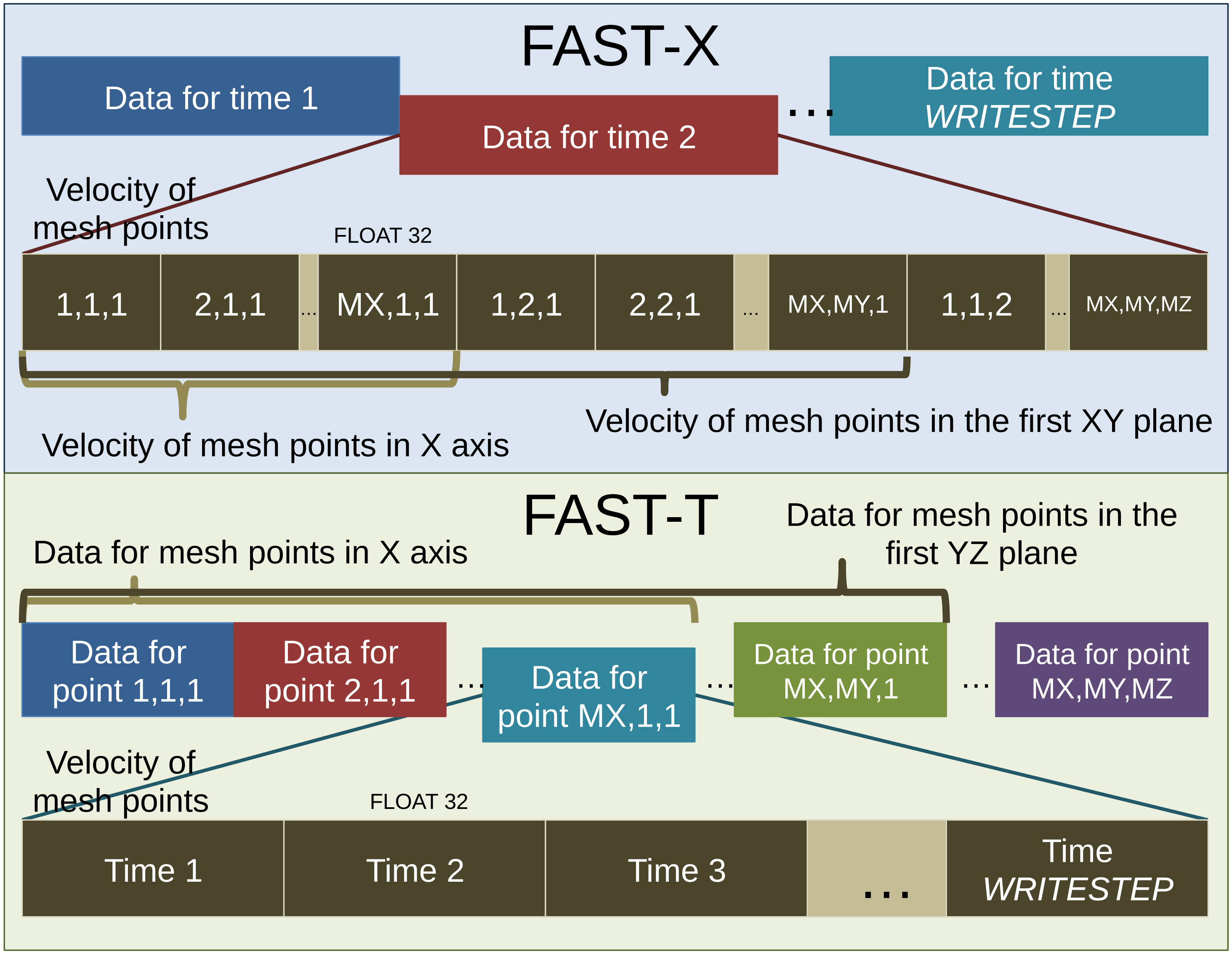
Two-layer I/O Model

- Parallel I/O
  - Read and redistribute multiple terabytes inputs (19 GB/s)
    - Contiguous block read by reduced number of readers
    - High bandwidth asynchronous point-to-point communication redistribution
- **Aggregate and write (10GB/s)**
  - **Temporal aggregation buffers**
  - **Contiguous writes**
  - **Throughput**
  - **System adaptive at run-time**



Fast X or Fast T

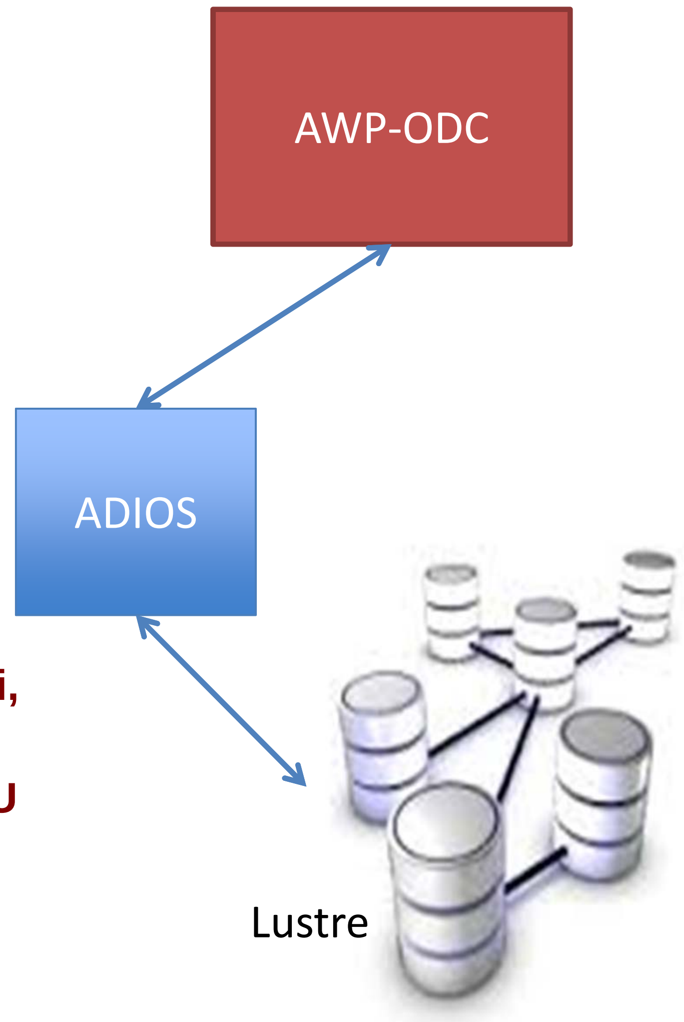
- **Fast-X: small-chunked and more interleaved. Fast-T: large-chunked and less interleaved**



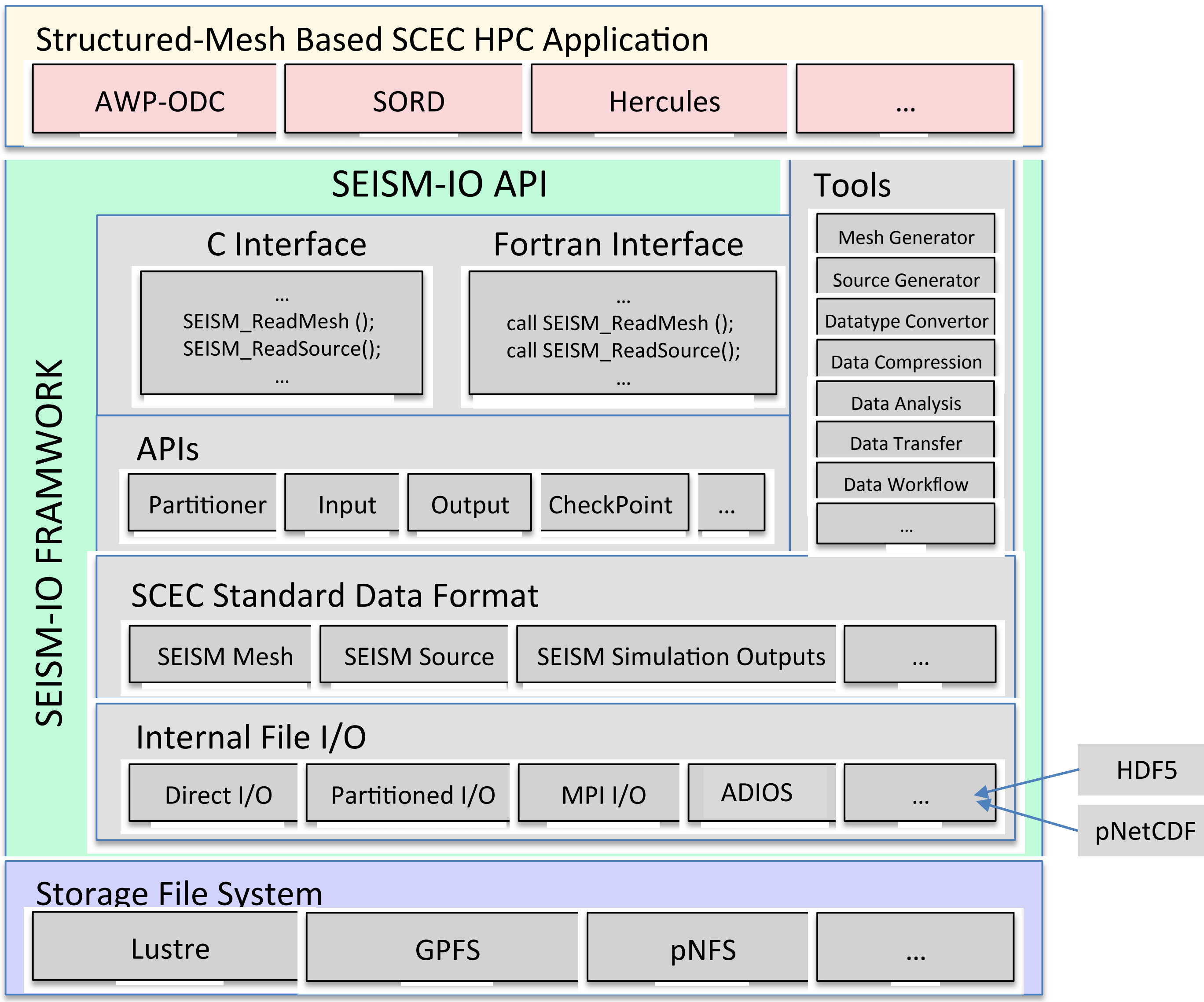
(Poyraz et al., ICCS''14)

ADIOS Checkpointing

- **Problems at M8 on Jaguar: system instabilities, 32 TB checkpointing per time step**
- **Chino Hills 5Hz simulation validated ADIOS implementation:**
  - ➢ **Mesh size: 7000 x 5000 x 2500**
  - ➢ **Nr. of cores: 87,500 on Jaguar**
  - ➢ **WCT: 3 hours**
  - ➢ **Total timesteps: 40K**
  - ➢ **ADIOS saved checkpoints at 20Kth timestep and validated the outputs at 40Kth timestep**
  - ➢ **Avg. I/O performance: 22.5 GB/s (compared to 10 GB/s writing achieved with manually-tuned code using MPI-IO)**
- **Implementation Supported by Norbert Podhorszki, Scott Klasky, and Qing Liu at ORNL**
- **Future plan: add ADIOS Checkpointing to the GPU code**



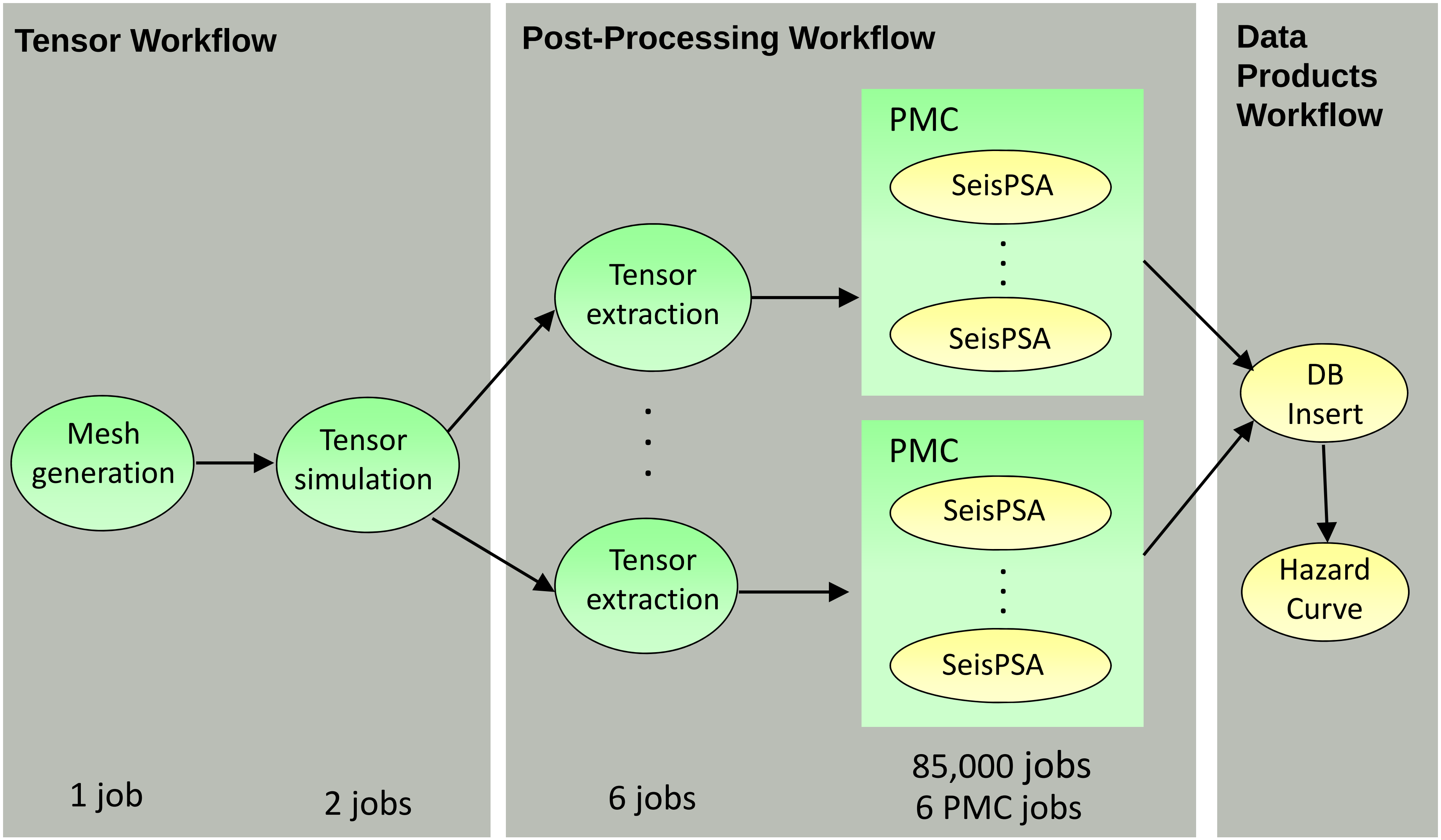
SEISM-IO: An IO Library for Integrated Seismic Modeling



CyberShake Calculations

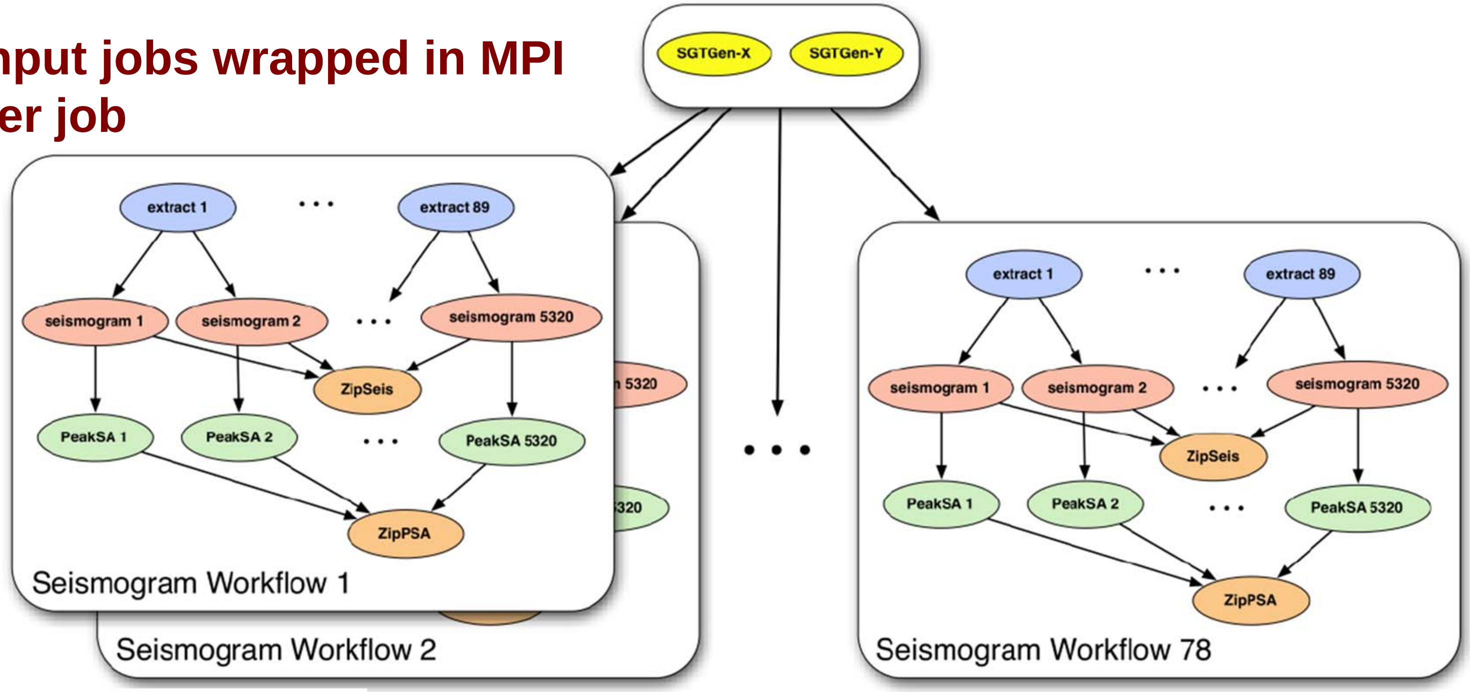
- CyberShake contains two phases
- Strain Green Tensor (SGT) calculation
  - Large MPI jobs
  - AWP-ODC-SGT GPU
  - 85% of CyberShake compute time
- Post-processing (reciprocal calculation)
  - Many (~400k) serial, high throughput, loosely coupled jobs
  - Workflow tools used to manage jobs
- Both phases are required to determine seismic hazard at one site
- For a hazard map, must calculate ~200 sites

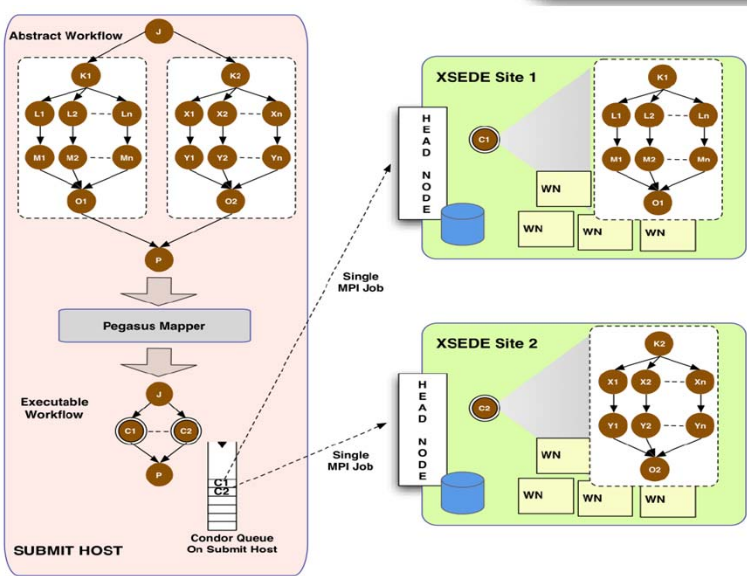
CyberShake Workflows

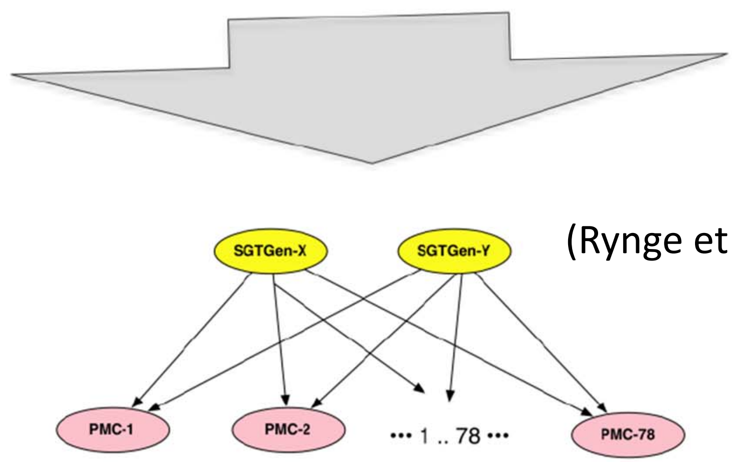


CyberShake Workflows Using Pegasus-MPI-Cluster

- **High throughput jobs wrapped in MPI master-worker job**

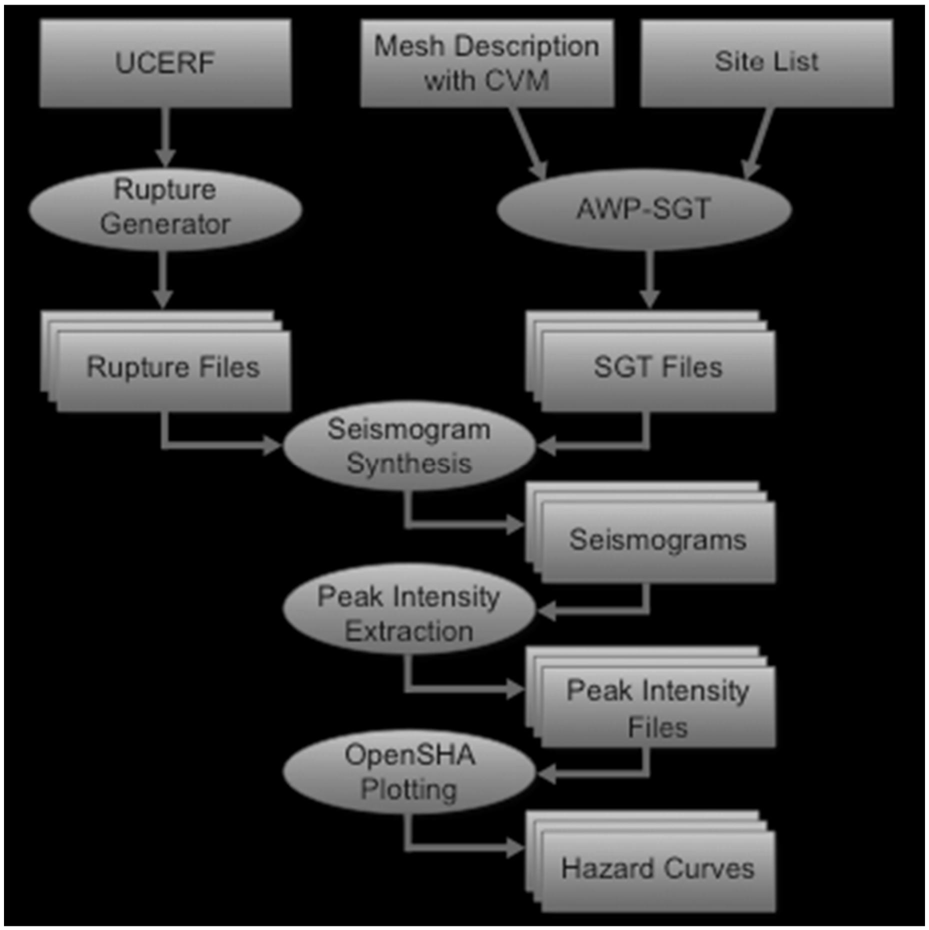






(Rynge et al., XSEDE'2012)

CPUs/GPUs Co-scheduling

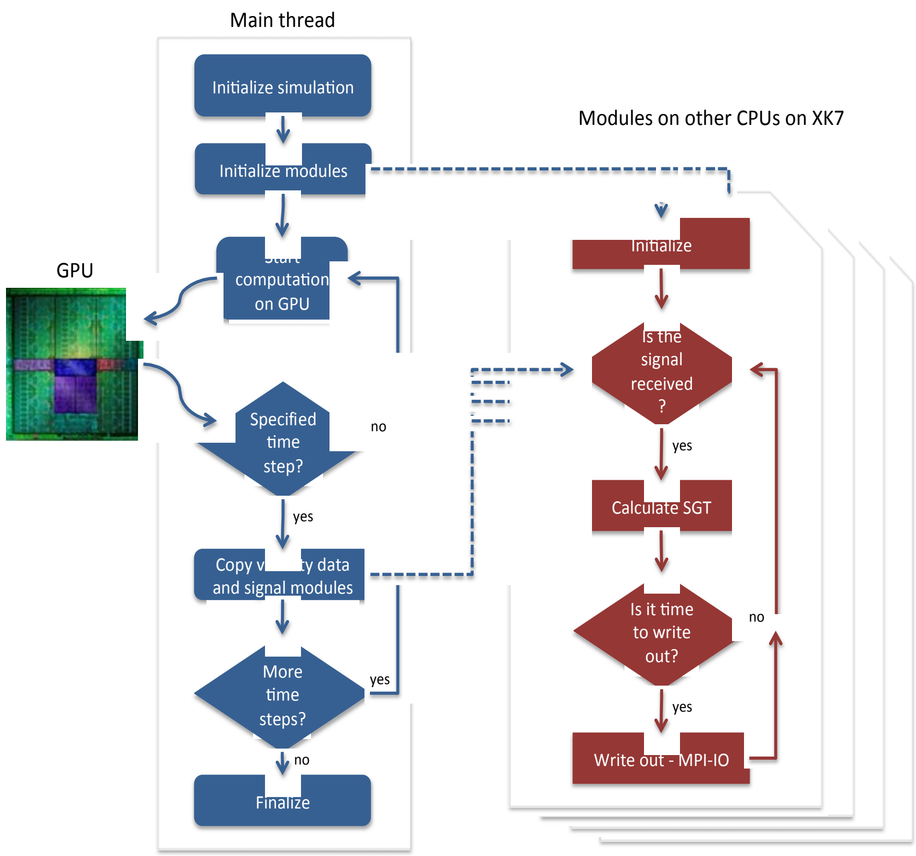


- **CPUs run reciprocity-based seismogram and intensity computations while GPUs are used for strain Green tensor calculations**
- **Run multiple MPI jobs on compute nodes using Node Managers (MOM)**

```
aprun -n 50  <GPU executable> <arguments> &
get the PID of the GPU job
cybershake_coscheduling.py:
    build all the cybershake input files
    divide up the nodes and work among a customizable number of jobs
    for each job:
        fork extract_sgt.py cores --> performs pre-processing and launches
              "aprun -n <cores per job> -N 15 -r 1 <cpu executable A>&"
        get PID of the CPU job
    while executable A jobs are running:
        check PIDs to see if job has completed
        if completed: launch
           "aprun -n <cores per job> -N 15 -r 1 <cpu executable B>&"
    while executable B jobs are running:
        check for completion
check for GPU job completion
```

Post-processing on CPUs: API for Pthreads

- **AWP-API lets individual pthreads make use of CPUs: post-processing**
  - Vmag, SGT, seismograms
  - Statistics (real-time performance measuring)
  - Adaptive/interactive control tools
  - In-situ visualization
  - Output writing is introduced as a pthread that uses the API



CyberShake Study 14.2 Metrics

- **1,144 hazard curves (4 maps) on NCSA Blue Waters**
- **342 hours wallclock time (14.25 days)**
- **46,720 CPUs + 225 GPUs used on average**
  - **Peak of 295,040 CPUs, 1100 GPUs**
- **GPU SGT code 6.5x more efficient than CPU SGT code** (XK7 vs XE6 at node level)
- **99.8 million jobs executed (81 jobs/second)**
  - **31,463 jobs automatically run in the Blue Waters queue**
- **On average, 26.2 workflows (curves) concurrently**

CyberShake SGT Simulations on XK7 vs XE6

CyberShake 1.0 Hz	XE6	XK7	XK7 (CPU-GPU co-scheduling)
Nodes	400	400	400
SGT hrs per site	**10.36**	**2.80**	**2.80**
Post-processing hours per site**	0.94	1.88**	2.00
Total Hrs per site	**11.30**	**4.68**	**2.80**
Total SUs(Millions)*	723 M	299 M	179 M
SUs saving (Millions)		**424 M**	**543 M**

3.7x speedup

* Scale to 5000 sites based on two strain Green tensor runs per site

** based on CyberShake 13.4 map

CyberShake SGT Simulations on XK7 vs XE6

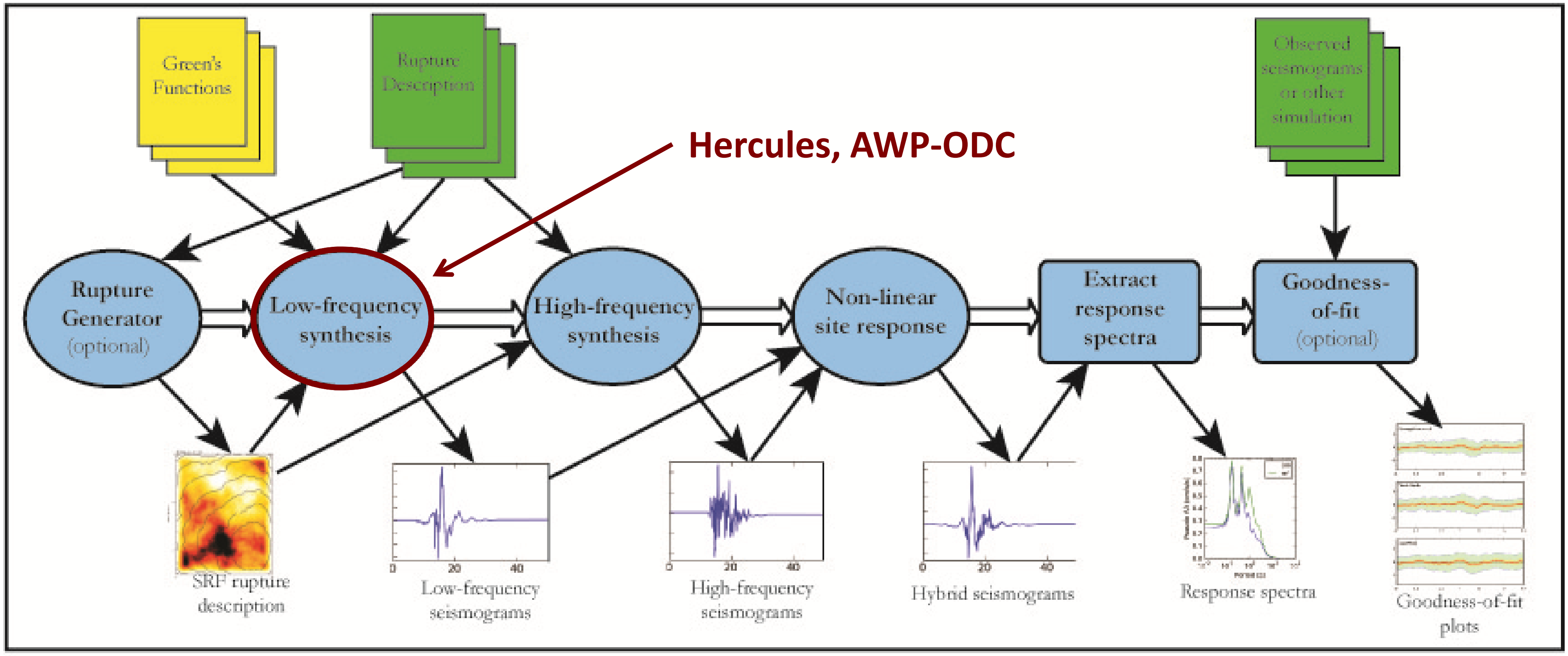
CyberShake 1.0 Hz	XE6	XK7	XK7 (CPU-GPU co-scheduling)
Nodes	400	400	400
SGT hrs per site	**10.36**	**2.80**	**2.80**
Post-processing hours per site**	0.94	1.88**	2.00
Total Hrs per site	**11.30**	**4.68**	**2.80**
Total SUs(Millions)*	723 M	299 M	179 M
SUs saving (Millions)		**424 M**	**543 M**

3.7x speedup

* Scale to 5000 sites based on two strain Green tensor runs per site

** based on CyberShake 13.4 map

Broadband Platform Workflow



Broadband Platform Software Distributions:

Source Codes and Input Config Files: 2G (increases as platform runs)

Data Files (Greens Functions): 11G (static input files)

Earthquake Problems at Extreme Scale

- **Dynamic rupture simulations**
  - **Current 1D outer/inner scale: $6x10^5$**
  - **Target: 1D 600000m/0.001m ($6x10^8$)**
- **Wave propagation simulations**
  - **Current 4D scale ratio: $1x10^{17}$**
  - **Target 4D scale ratio: $3x10^{23}$**
- **Data-intensive simulations**
  - **Current tomography simulations: ~ 0.5 PB**
    - **2015-2016 plan to carry out 5 iterations, 1.9 TB for each seismic source, total at least 441 TB for the duration of the inversion**
  - **Target tomography simulations:~ 32 XBs**

SCEC

Southern California Earthquake Center

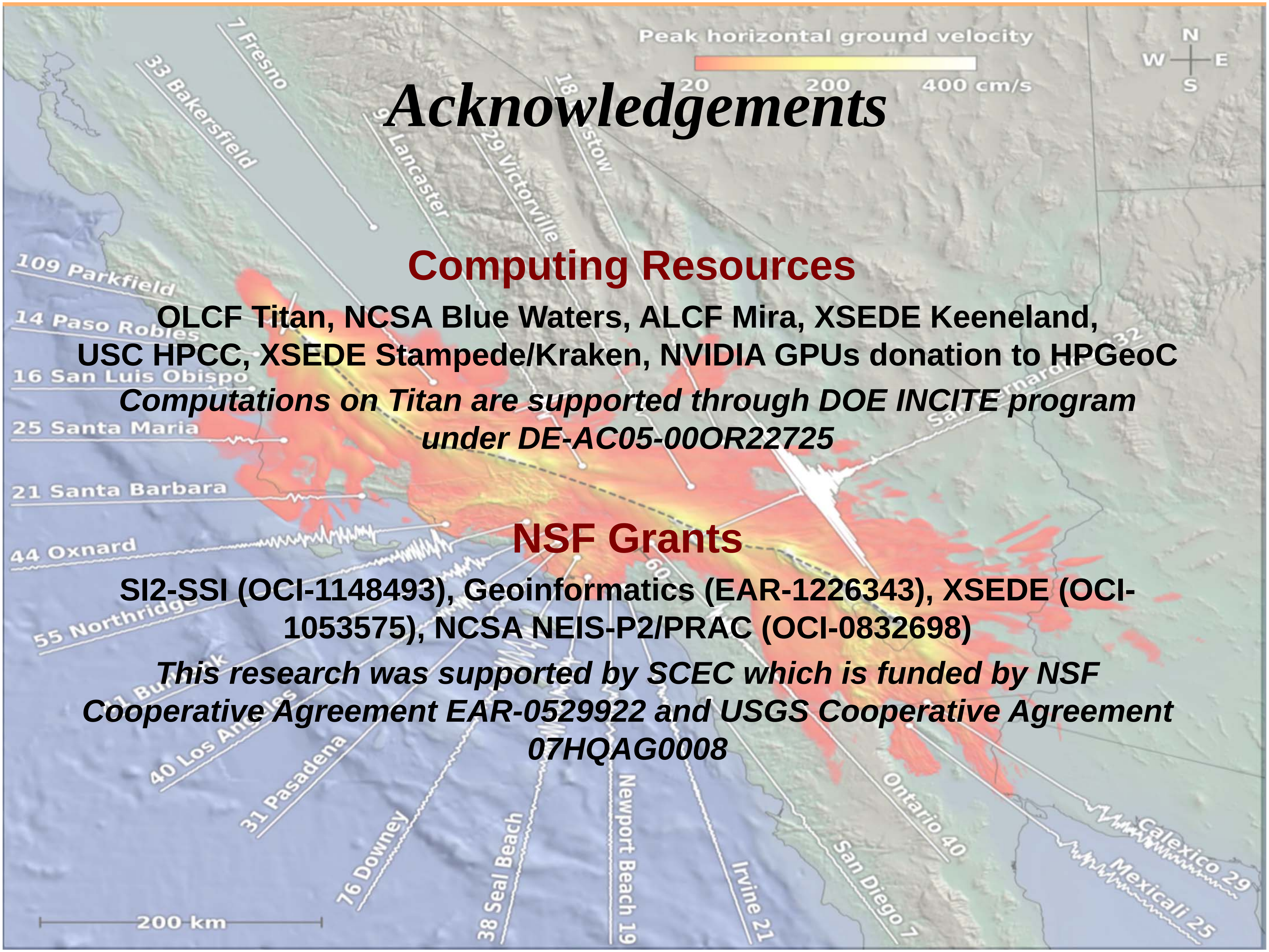
SCEC 2015-2016 Computational Plan on Titan

Research	Milestone	Code	Nr. Of Runs	M SUs
Material heterogeneities wave propagation	2-Hz regional simulations for CVM with small-scale stochastic material perturbations	AWP-ODC-GPU	8	13
Attenuation and source wave propagation	10-Hz simulations integrating rupture dynamic results and wave propagation simulator	AWP-ODC-GPU SORD	5	19
Structural representation and wave propagation	4 Hz scenario and validation simulation, integration of frequency dependent Q, topography, and nonlinear wave propagation	Hercules-GPU	5	20
CyberShake PSHA	1.0-Hz hazard map	AWP-SGT-GPU	300	100
CyberShake PSHA	1.5-Hz hazard map	AWP-SGT-GPU	200	130

-> 282 M SUs

SCEC Software Development

- **Advanced algorithms**
  - Development of Discontinuous Mesh AWP
  - New physics: near-surface heterogeneities, frequency-dependent attenuation, fault roughness, near-fault plasticity, soil non-linearities, topography
  - High-F simulation of ShakeOut scenario 0-4 Hz or higher
- **Prepare SCEC HPC codes for next-generation systems**
  - Programming model
    - Three levels of parallelism to address accelerating technology. Portability. Data locality and communication avoiding
  - Automation: Improvement of SCEC workflows
  - I/O and fault tolerance
    - Cope with millions of simultaneous I/O requests. Support multi-tiered I/O systems for scalable data handling. MPI/network and node level fault tolerance
  - Performance
    - Hybrid heterogeneous computing. Support for in-situ and post-hoc data processing. Load balancing
  - Benchmark SCEC mini-applications and tune on next-generation processors and interconnects



Acknowledgements

Computing Resources

OLCF Titan, NCSA Blue Waters, ALCF Mira, XSEDE Keeneland, USC HPCC, XSEDE Stampede/Kraken, NVIDIA GPUs donation to HPGeoC

Computations on Titan are supported through DOE INCITE program under DE-AC05-00OR22725

NSF Grants

SI2-SSI (OCI-1148493), Geoinformatics (EAR-1226343), XSEDE (OCI-1053575), NCSA NEIS-P2/PRAC (OCI-0832698)

This research was supported by SCEC which is funded by NSF Cooperative Agreement EAR-0529922 and USGS Cooperative Agreement 07HQAG0008