

Accelerating CyberShake Calculations on Heterogeneous Architectures

Yifeng Cui

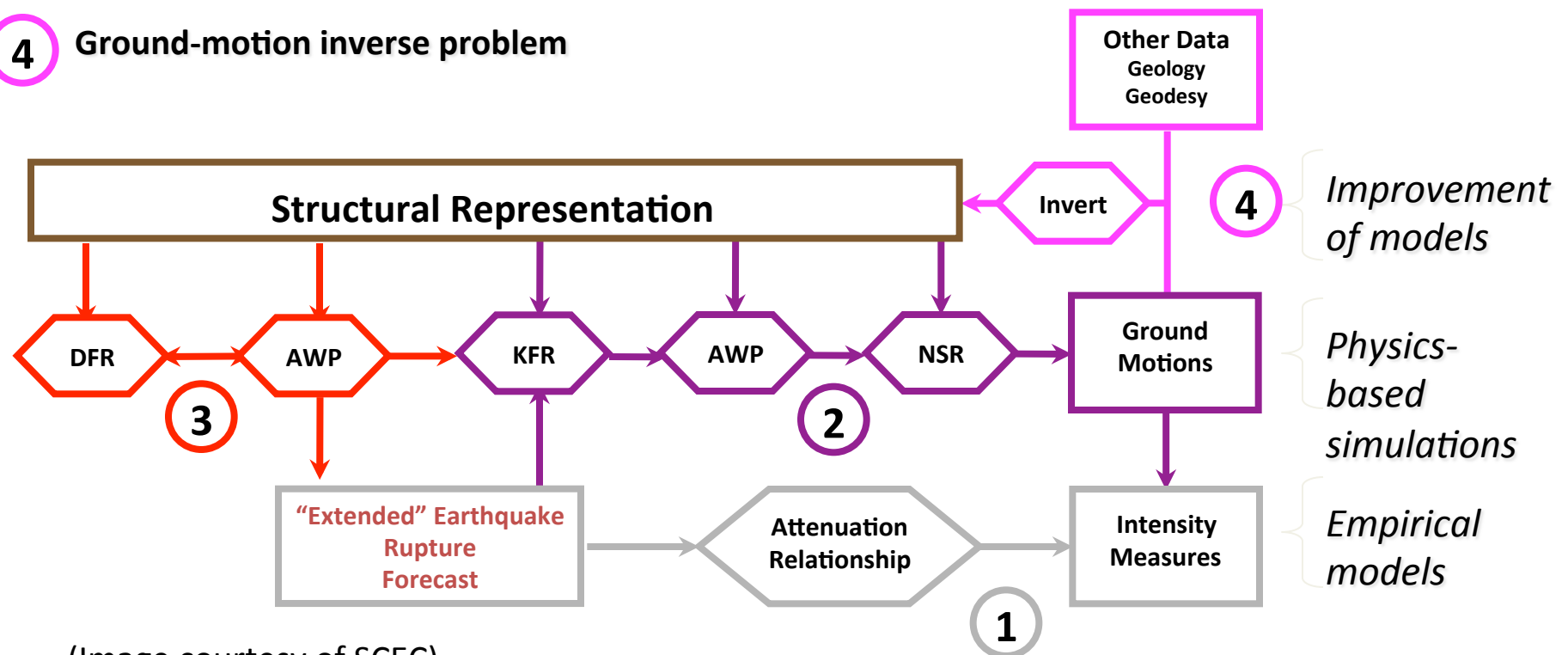
San Diego Supercomputer Center

Acknowledgements

- ECSS consultants: Jay Alameda, Amit Chourasia, Yifeng Cui, Yaakoub El-Khamra, Matt Mckenzie.
- This work was also supported by Advanced Support for TeraGrid Applications, including funding for Jun Zhou's initial GPU work, DongJu Choi is a collaborator
- Implementation and tests were carried out on six M2050 and M2090 GPUs NVIDIA donated to HPGeoC. Benchmarks are performed on XSEDE Keeneland and NCCS TitanDev system
- Thanks to:
 - Carl Ponder of NVIDIA for technical support
 - Sreeram Potluri, Karen Tomko and DK Panda of OSU for providing MVAPICH compiler support and Keeneland porting assistance
 - Patrick Small of SCEC for preparing Chino Hills velocity model, and
 - Amit Chourasia for visualization of validation exercises

Southern California Earthquake Center (SCEC) Computational Pathways

- 1 Standard seismic hazard analysis
- 2 Ground motion simulation
- 3 Dynamic rupture modeling
- 4 Ground-motion inverse problem



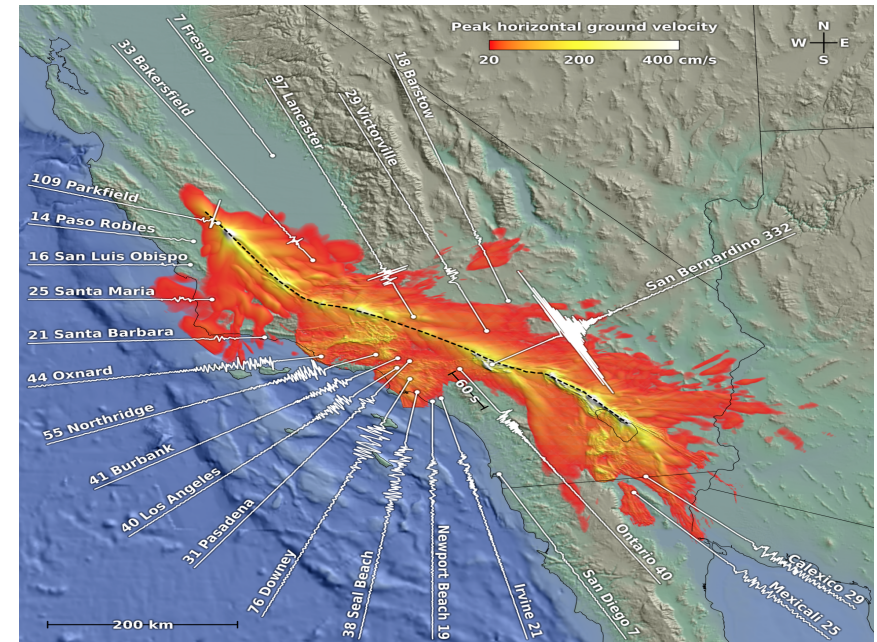
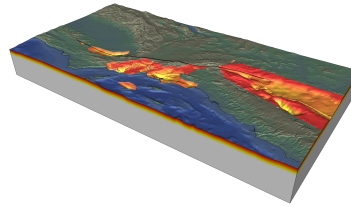
(Image courtesy of SCEC)

KFR = Kinematic Fault Rupture
DFR = Dynamic Fault Rupture

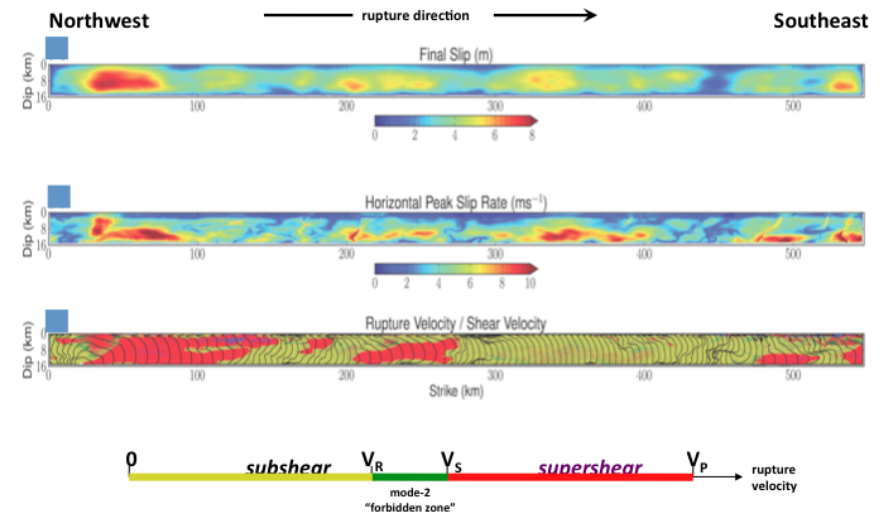
AWP = Anelastic Wave Propagation
NSR = Nonlinear Site Response

M8 Simulation using AWP-ODC

- **Magnitude 8.0 wall-to-wall scenario, worst-case for southern San Andreas Fault**
 - Fault length: 545 km
 - Minimum wavelength: 200 m
 - NW→SE rupture propagation
- **Dynamic rupture simulation performed on Kraken, 7.5 hours using 2160 cores**
 - 881,475 subfaults
 - 250 sec of rupture
- **Wave propagation simulation performed on Jaguar, 24 hours using 223,074 cores (220 Tflop/s sustained)**
 - 436 billion grid points representing SCEC CVM4 of 810 x 405 x 85 km (spatial resolution of 40 m)
 - Minimum shear-wave velocity of 400 m/s
 - 368 s of ground motions (160K time steps of 0.0023 s) representing frequencies up to 2 Hz



ACM Gordon Bell Finalist, 2010



<http://visservices.sdsc.edu/projects/scec/m8/1.0/>

The computing resources are provided by NSF LRAC awards and DOE INCITE under Contract No. DOE-AC05-00OR22725, ALCF DE-AC02-06CH11357

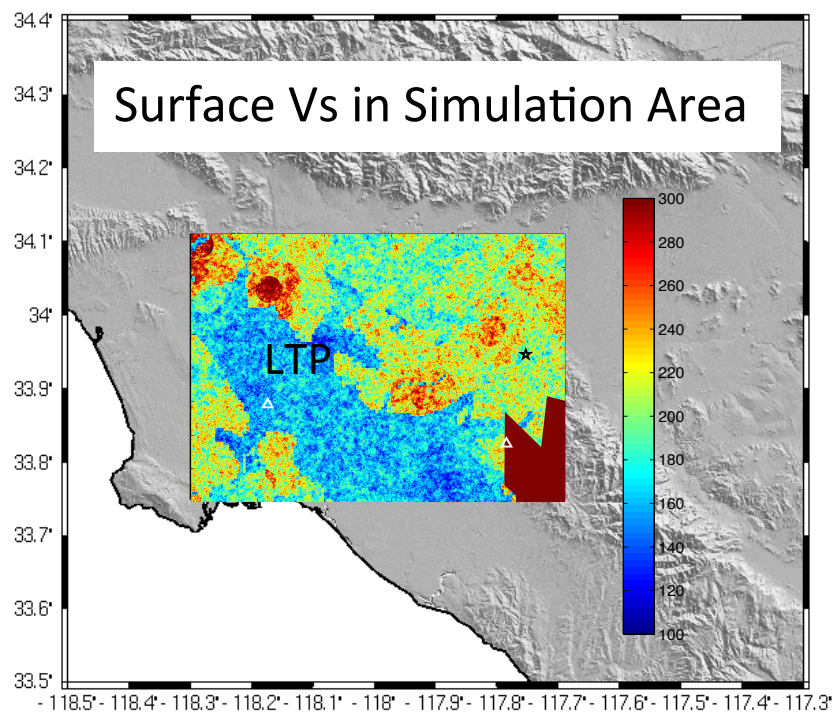
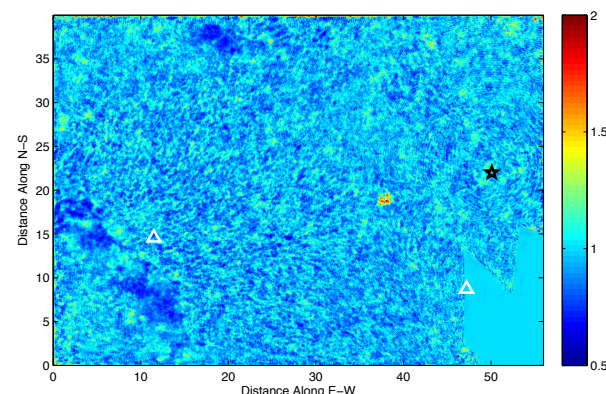
(Cui et al., SC10, 2010)

0-5Hz $M_w 5.4$ Chino Hills Simulation With Statistical Model of Small-Scale Inhomogeneities

Code	AWP-ODC (FD)
Max Frequency	5 Hz
Minimum Vs	200 m/s
Hurst Humber, σ	0.1, 10%
Δx	8 m
Wall-clock Time	11 Hours

Amplification/
deamplification
of PGVs by up to
a factor of 2
(with popular Q
model included)

Ratios of PGVs With and Without Statistical Model

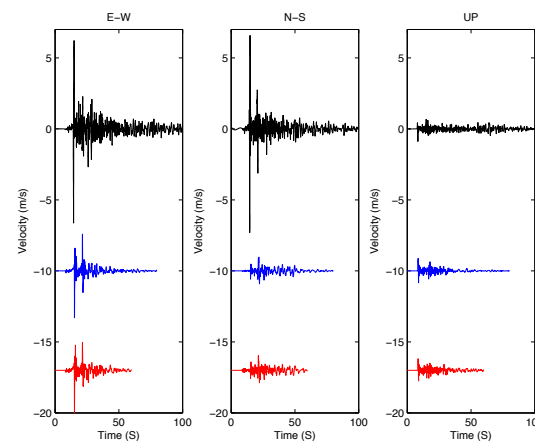


(Olsen, Cui, Poyraz, Savran, Jacobsen, 2012)

Data

Syn, $Q_s=50V_s$

Syn, $Q_s=50V_s+$
statistical model

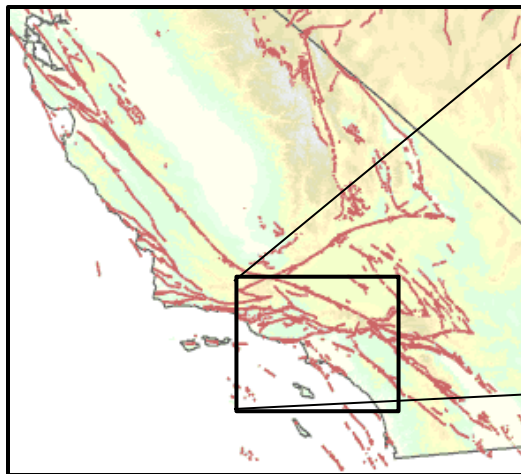
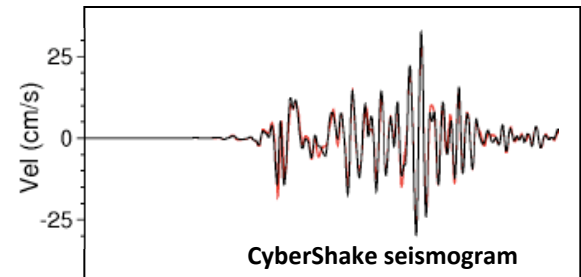


Preliminary Result:

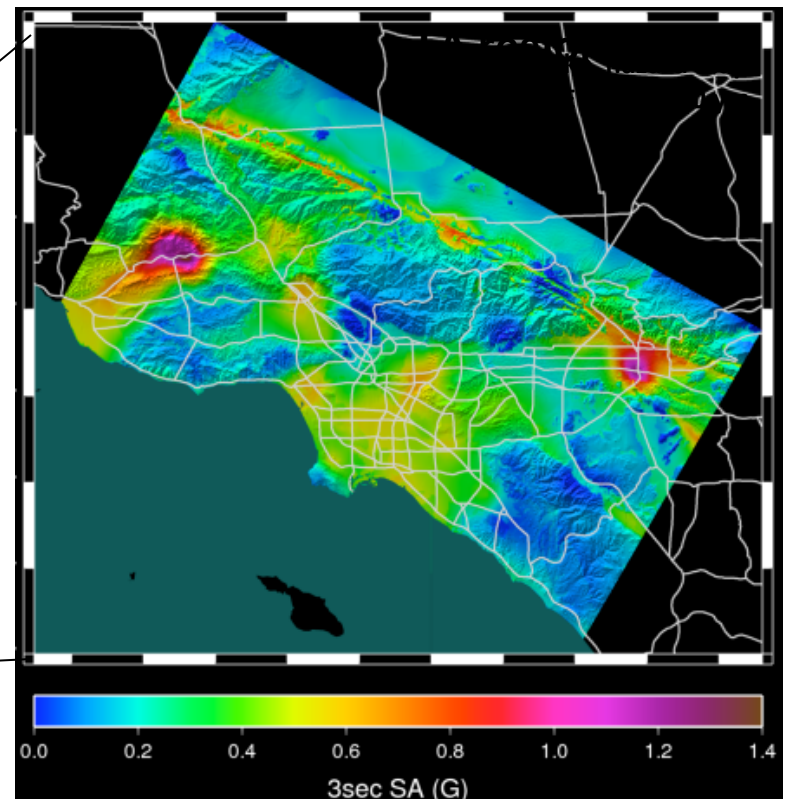
Popular So Cal Q-model underpredicts 0-5 Hz data duration in basin, with or without statistical model. Thus, the Q-model may need tuning (toward higher Q) when high frequencies (and statistical model) are included

CyberShake Hazard Model

- **CyberShake 1.0 computation: 225 sites in LA region, $f < 0.5$ Hz, completed on TACC Ranger in 2009, build on UCERF2**
 - 440,000 simulations per site
 - 5.5 million CPU hrs (50-day run on *Ranger* using 4,400 cores)
 - 189 million jobs
 - 165 TB of total output data
 - 10.6 TB of stored data
 - 2.1 TB of archived data

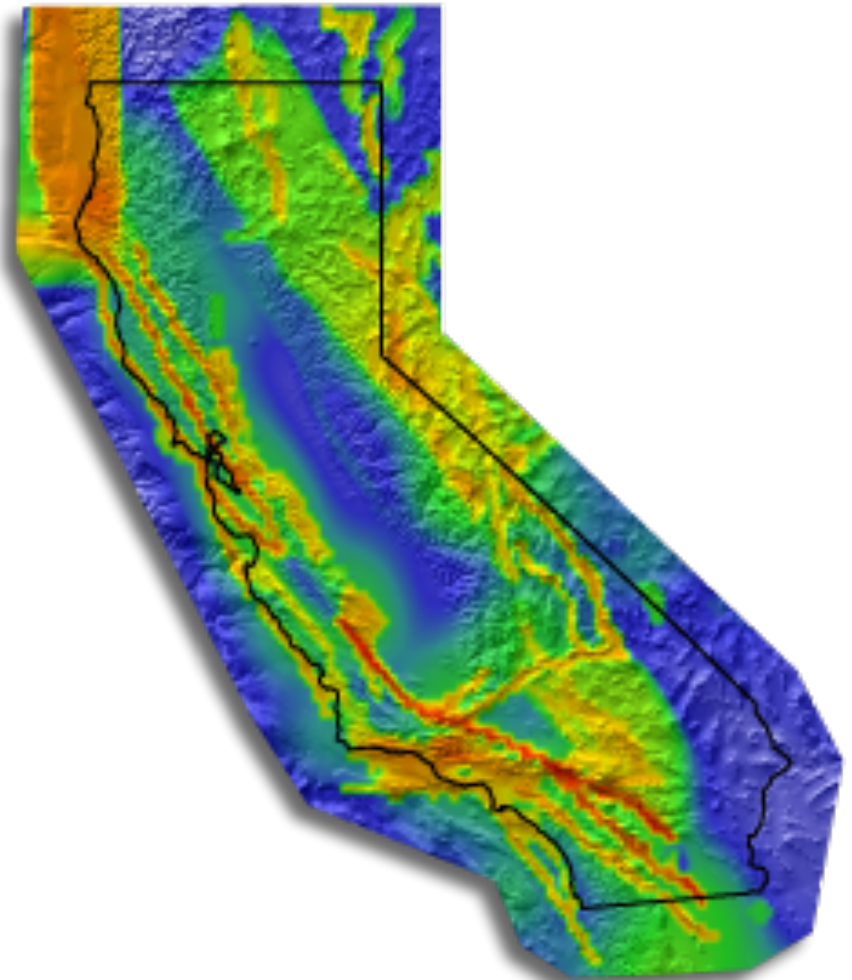


(Source: Philip
Maechling of SCEC)



CyberShake 3.0 Statewide Hazard Model

- California statewide wave propagation-based PSHA map up to 10Hz
- Build on UCERF3.0
- 4240 sites, $f < 1.0$ Hz computed using AWP-Grave and AWP-ODC codes
 - 750 million CPU hrs
 - 3.6 billion jobs
 - 48 PB of total output data
 - 2.6 PB of stored data
 - 77 TB of archived data
- Requires high-capability machines to be used in a high-capacity mode
 - 3000 hrs on 1/2 Jaguar
 - 770 hrs on 1/2 Blue Waters



Why GPU?

- GPU architectures are among the top systems, offering a fast and inexpensive solution, NCCS Titan, Blue Waters are among the large heterogeneous systems
- Reduced power consumption for price-to-performance ratio improvement, power efficiency plays an important role in today's architectures
- An energy efficient GPU code will reduce time-to-solution for SCEC CyberShake calculations and dramatically reduce the allocation requirement

CyberShake 3.0	CPU	GPU	CPU+GPU
Cores	14400	900	14400+900
Hours per SGT run	2.56	0.51	0.44
Total Power \$*	\$805,000	\$72,000	\$202,000
Memory (TB)	28.8 TB	5.4 TB	29.9 TB
Flop/s	15.4Tflops	100Tflops	~115Tflops

* Assuming 5000 site and two runs per site, based on Cray XK6 power consumption

AWP-ODC

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

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

solved by explicit **staggered-grid 4th-order FD**

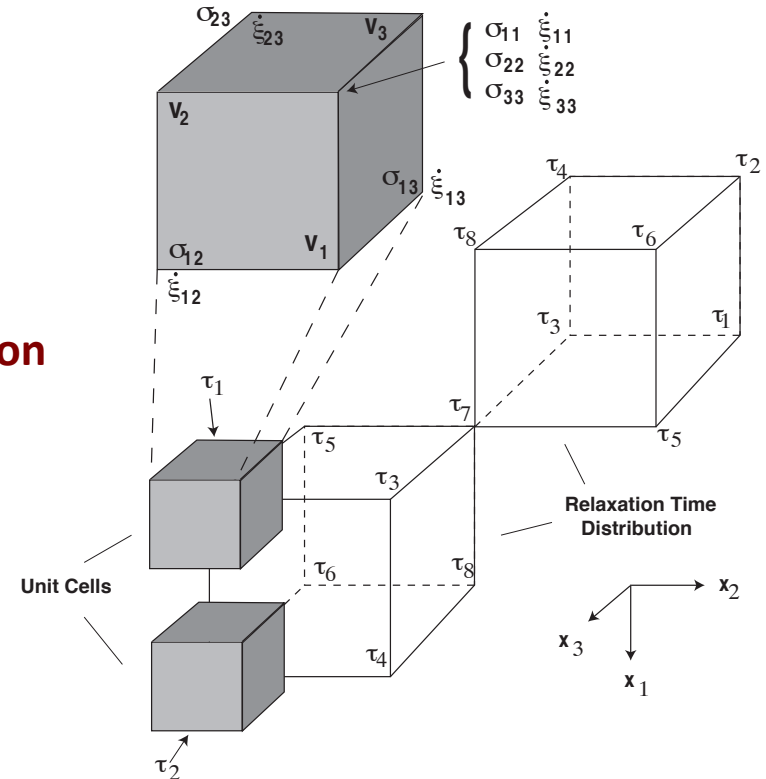
- Memory variable formulation of **inelastic relaxation**

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

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

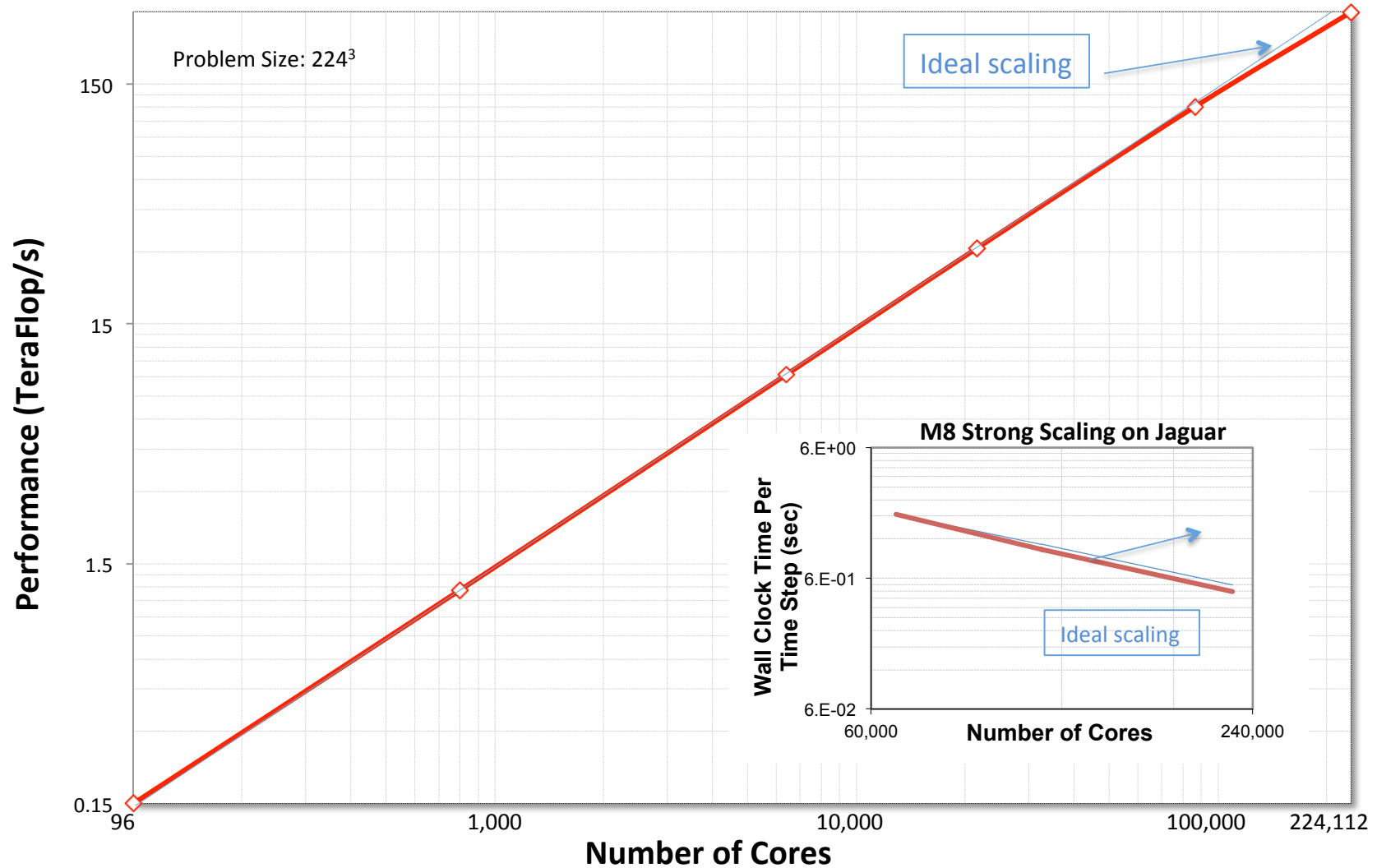
using coarse-grained representation (Day 1998)

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



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

AWP-ODC CPU Code Scales Well up to 223K cores on NCCS Jaguar



AWP-ODC Computational Kernel



Main Loop:

Do T= timestep 0 to timestep N:

Compute velocities (vx, vy, vz) using stress (xx, yy, zz, xy, yz, xz)

 Update values of velocities (vx, vy) along the surface

 Update values of velocities (vz) based on (vx, vy) along the surface

Compute stress (xx, yy, zz) based on velocities (vx, vy, vz)

Compute stress (xy, xz, yz) based on velocities (vx, vy, vz)

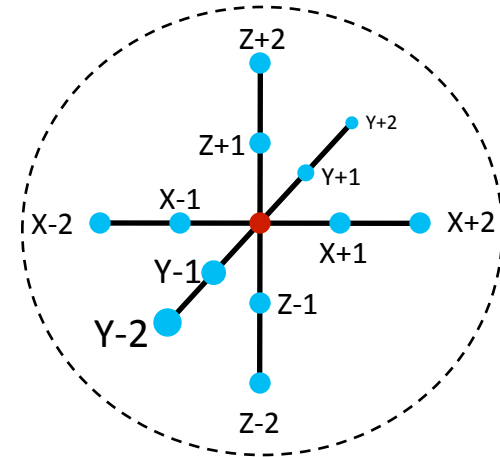
 Update values of stress (zz, xz, yz) along the surface

END DO

Velocity Computation Kernel (only vx computation is shown here)

$$\begin{aligned} vx(i, j, k) \text{ += } d1(i, j, k) * (&c1 * (xx(i, j, k) - xx(i-1, j, k) + c2 * (xx(i+1, j, k) - xx(i-2, j, k) \\ &c1 * (xy(i, j, k) - xy(i, j-1, k) + c2 * (xy(i, j+1, k) - xy(i, j-2, k) \\ &c1 * (xz(i, j, k) - xz(i, j, k-1) + c2 * (xz(i, j, k+1) - xz(i, j, k-2)) \end{aligned}$$

Stress Computation Kernel (only xy computation is shown here)

$$\begin{aligned} vxy &= c1 * (vx(i, j+1, k) - vx(i, j, k)) + c2 * (vx(i, j+2, k) - vx(i, j-1, k)) \\ vyx &= c1 * (vy(i, j, k) - vy(i-1, j, k)) + c2 * (vy(i+1, j, k) - vy(i-2, j, k)) \\ xy(i, j, k) &= xy(i, j, k) + xmu1(i, j, k) * (vxy + vyx) + x1 * r4(i, j, k) \\ r4(i, j, k) &= x2(i, j, k) * r4(i, j, k) + h1(i, j, k) * (vxy + vyx) \end{aligned}$$


3D 13-points Stencil

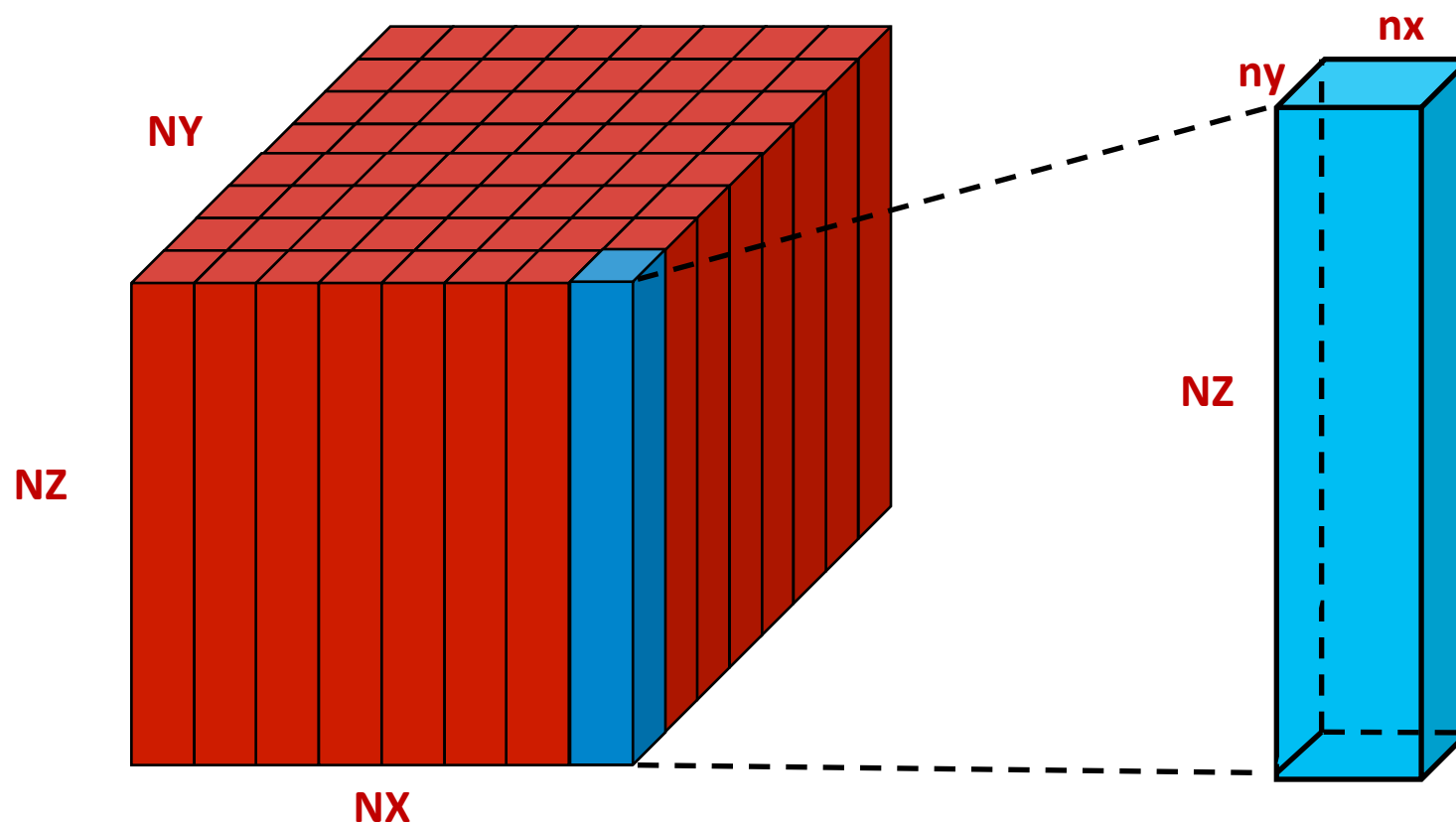
- **3D Arrays (21 in total):**
 - **velocities: 3 variables**
 - **stress: 6 variables**
 - **mesh info: 6 variables**
 - **memory: 6 variables**
- **13 points stencil computation**
 - **s1: update velocity by stress**
 - **s2: update stress by velocity**
 - **go back to s1**

Flops to Bytes Ratio of AWP-ODC Kernels

- AWP-ODC is memory bandwidth bound with low flops to bytes ratio

Three most time consuming Kernels	Reads	Writes	Flops	Flops/ Bytes
Velocity Comp.	51	3	86	0.398
Stress-1 Comp.	85	12	221	0.569
Total	136	15	307	0.508

Decomposition on CPU



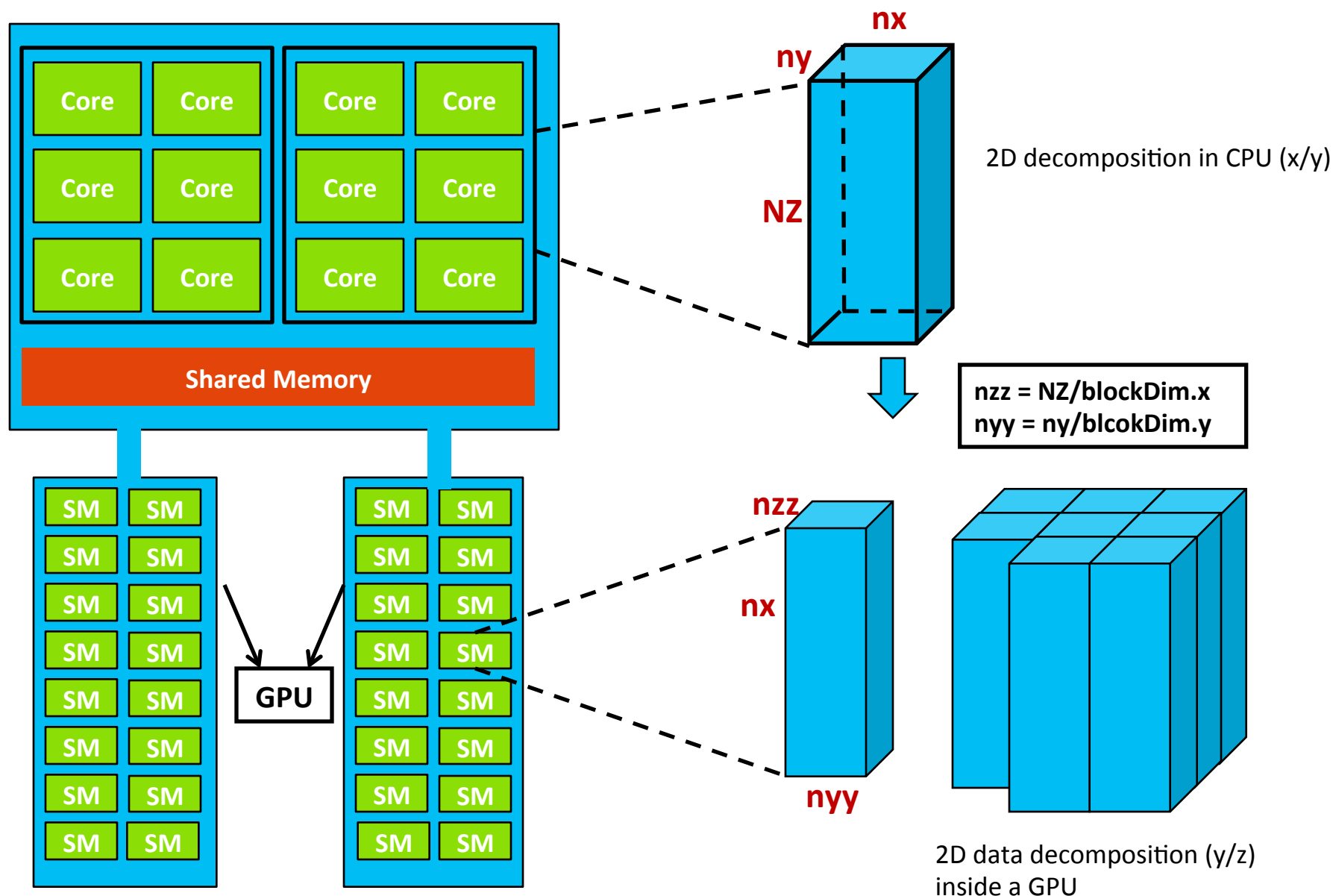
Domain size : $NX*NY*NZ$

Topology: $PX*PY*1$

Grid Size: $nx*ny*NZ$ ($nx = NX/PX$, $ny = NY/PY$)

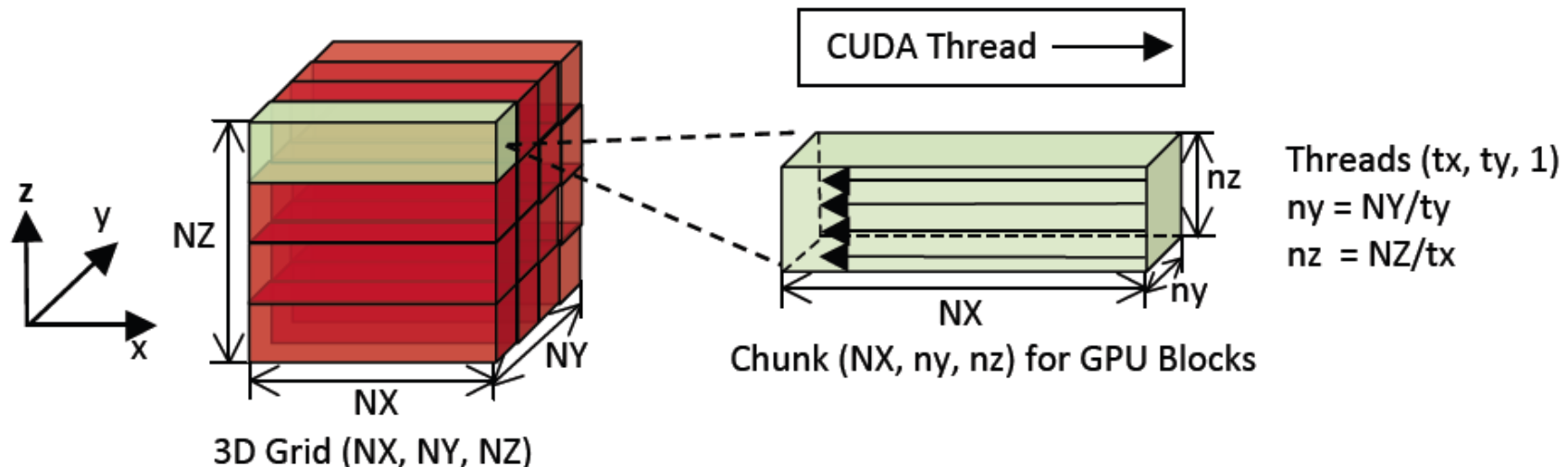
3D decomposition: 2D in CPU (x/y), and 1D in GPU (z)

Decomposition on GPU



Single GPU Implementation

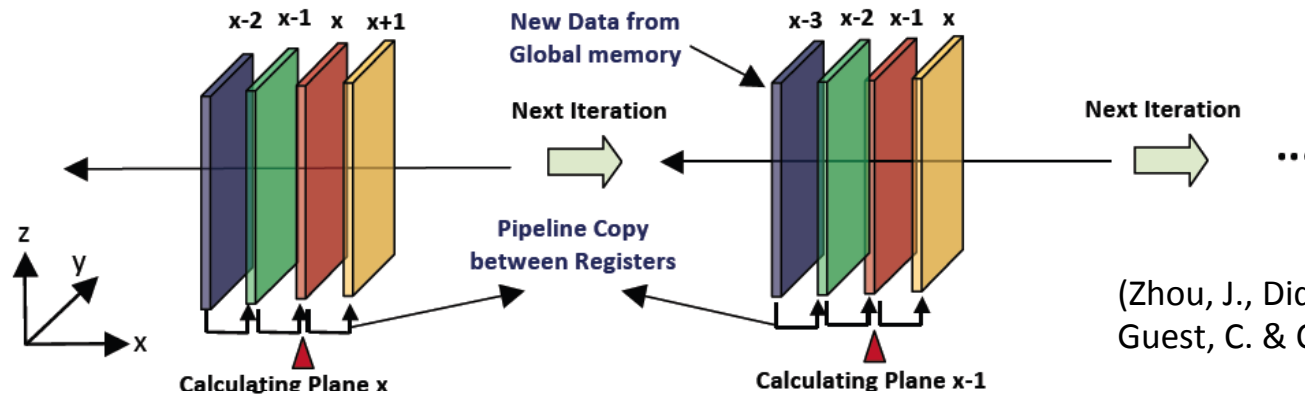
- Completely restructured the code for GPU Fermi architecture, converted from Fortran to C, Highlights of the current CUDA features
 - 2D decomposition in y/z directions only
 - Global memory coalesced, read only memory (texture/constant memory)
 - Pipelined register copy to reduce memory access
 - Rely on L1/L2 cache rather using on-chip shared memory



(Zhou, J., Didem, U., Choi, D., Guest, C. & Cui, Y., ICCS 2012)

Register Optimization

- Register optimization: pipeline register copy to reduce global memory access. Neighboring data in y and z directions in the aligned memory cached or prefetched, for neighboring data in travel direction (x axis), three out of four values can be reused for computation on the next iteration. Four private registers are defined to store these values and utilize pipeline to copy between registers, reducing 75% of the global memory access in x direction.



(Zhou, J., Didem, U., Choi, D., Guest, C. & Cui, Y., ICCS 2012)

VX computation kernel in thread (ty, tz) is to describe the Register Optimization:

Four registers: Y (Yellow), R (Red), G (Green), B (Blue)

Global memory: vx[i, ty, tz], xx[i, ty, tz] - 3D array (NX+4)*(NY+4)*(NZ+2*pad)

On-chip memory: xy[i, j, k], xz[i, j, k] - One plane Loaded to L1 cached or pre-fetched to shared memory

1. Preload R=xx[NX+2,ty,tz], G=xx[NX+1, ty, tz], B=xx[NX, ty, tz]

2. For (i=NX+1; i>=2; i--)

a. Y=R; R=G; G=B;

- pipeline copy between registers, very fast

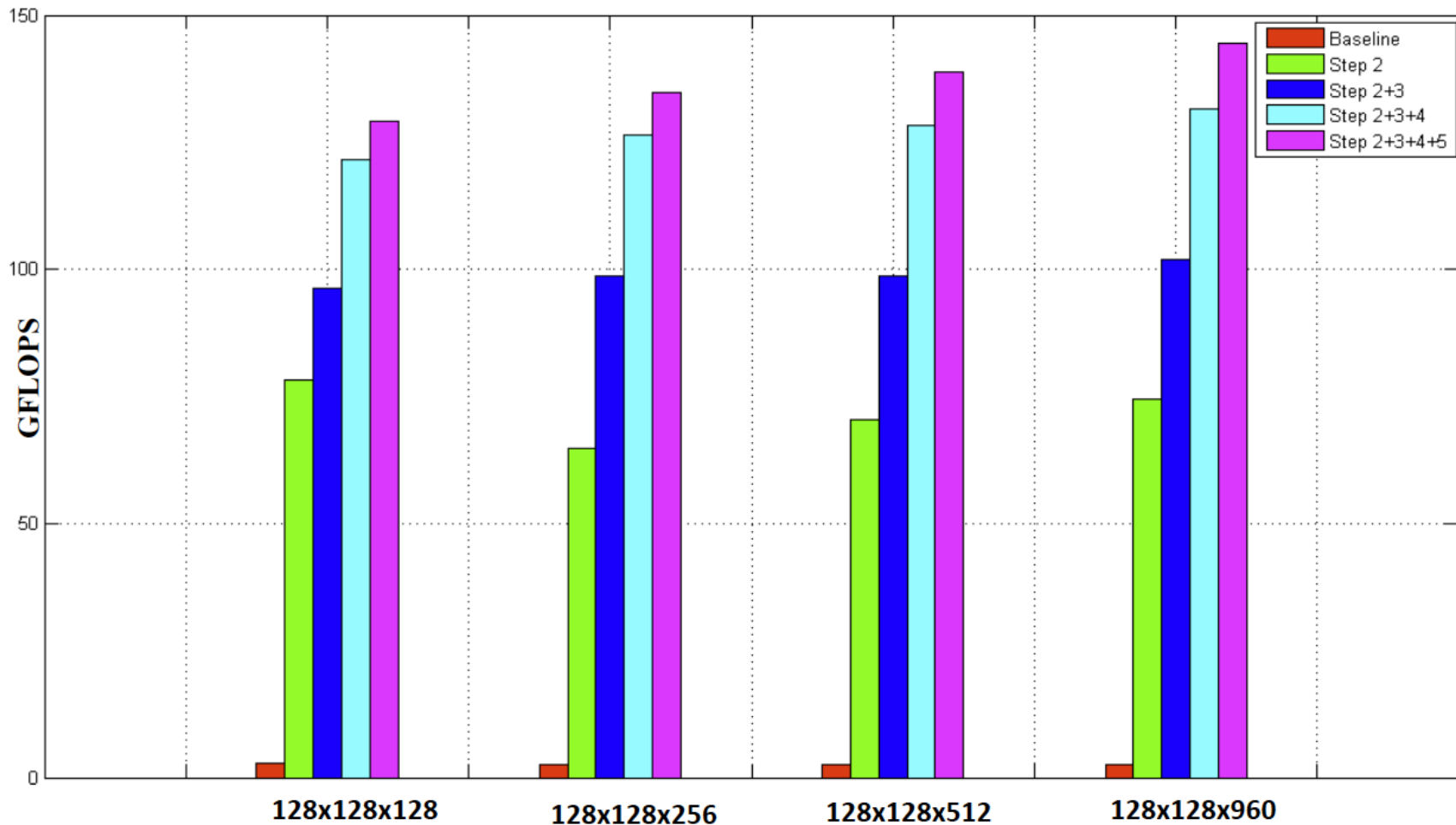
b. B = xx[i-2, ty tz];

- only access xx from global memory once in each iteration

c. VX Computation via Y, R, G, B, xy & xz

3. End For Loop and Return vx

FLOPS Optimization

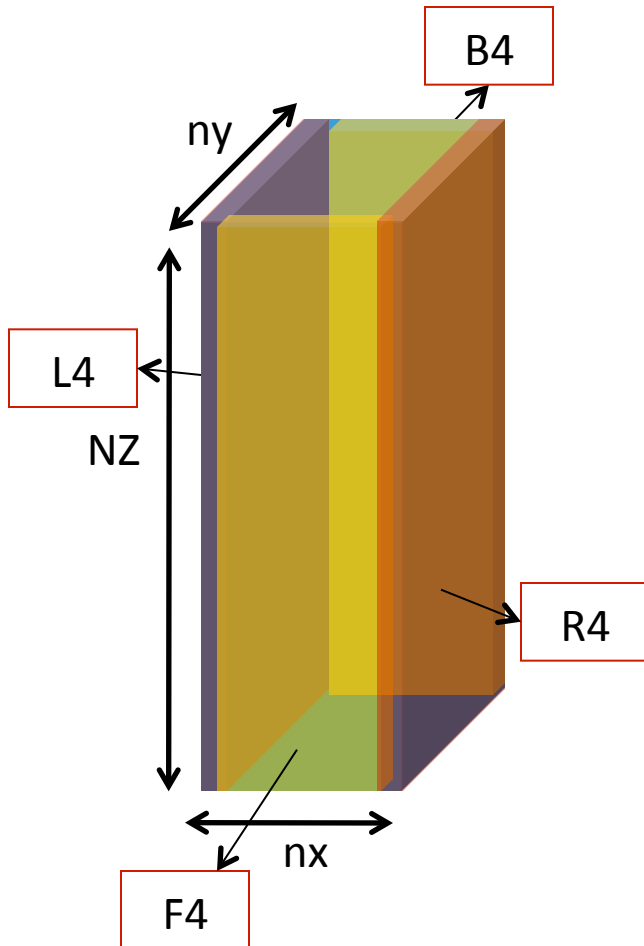


Step 2: GPU 2D Decomposition in y/z vs x/y; Step-3: Global memory Optimization

Step-4: Register Optimization; Step-5 L1 cache vs shared memory

(Zhou, J., Didem, U., Choi, D.,
Guest, C. & Cui, Y., ICCS 2012)

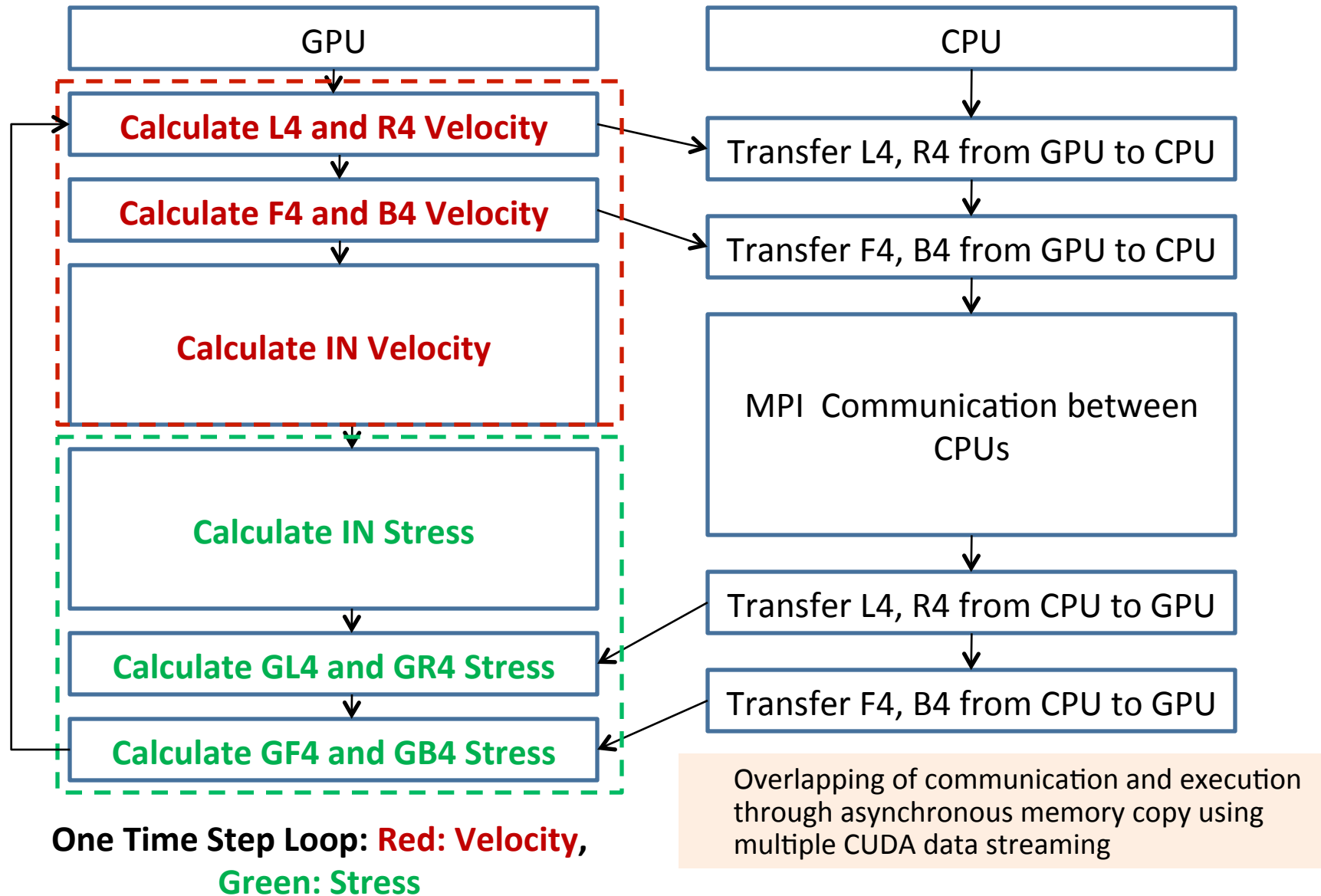
Multi-GPU Workload Definition



- Z direction is decomposed in GPU, so no MPI comm. in Z axis
- 3D Domain is (NX, NY, NZ) , 3D grid is (nx, ny, NZ)
- $nx = NX/npx$ (npx is the number of processors in x direction)
- $ny = NY/np_y$ (np_y is the number of processors in y direction)
- $L4 = (1:4, 1:ny, 1:NZ)$
- $R4 = (nx-3:nx, 1:ny, 1:NZ)$
- $F4 = (1:nx, 1:4, 1:NZ)$
- $B4 = (1:nx, ny-3:ny, 1:NZ)$
- $IN = (5:nx-4, 5:ny-4, 1:NZ)$ (All other part except L4, R4, F4, B4)
- $GL4 = (-1:4, 1:ny, 1:NZ)$ (L4 plus 2 more ghost layers)
- $GR4 = (nx-3:nx+2, 1:ny, 1:NZ)$ (R4 plus 2 more ghost layers)
- $GF4 = (1:nx, -1:4, 1:NZ)$ (F4 plus 2 more ghost layers)
- $GB4 = (1:nx, ny-3:ny+2, 1:NZ)$ (B4 plus 2 more ghost layers)

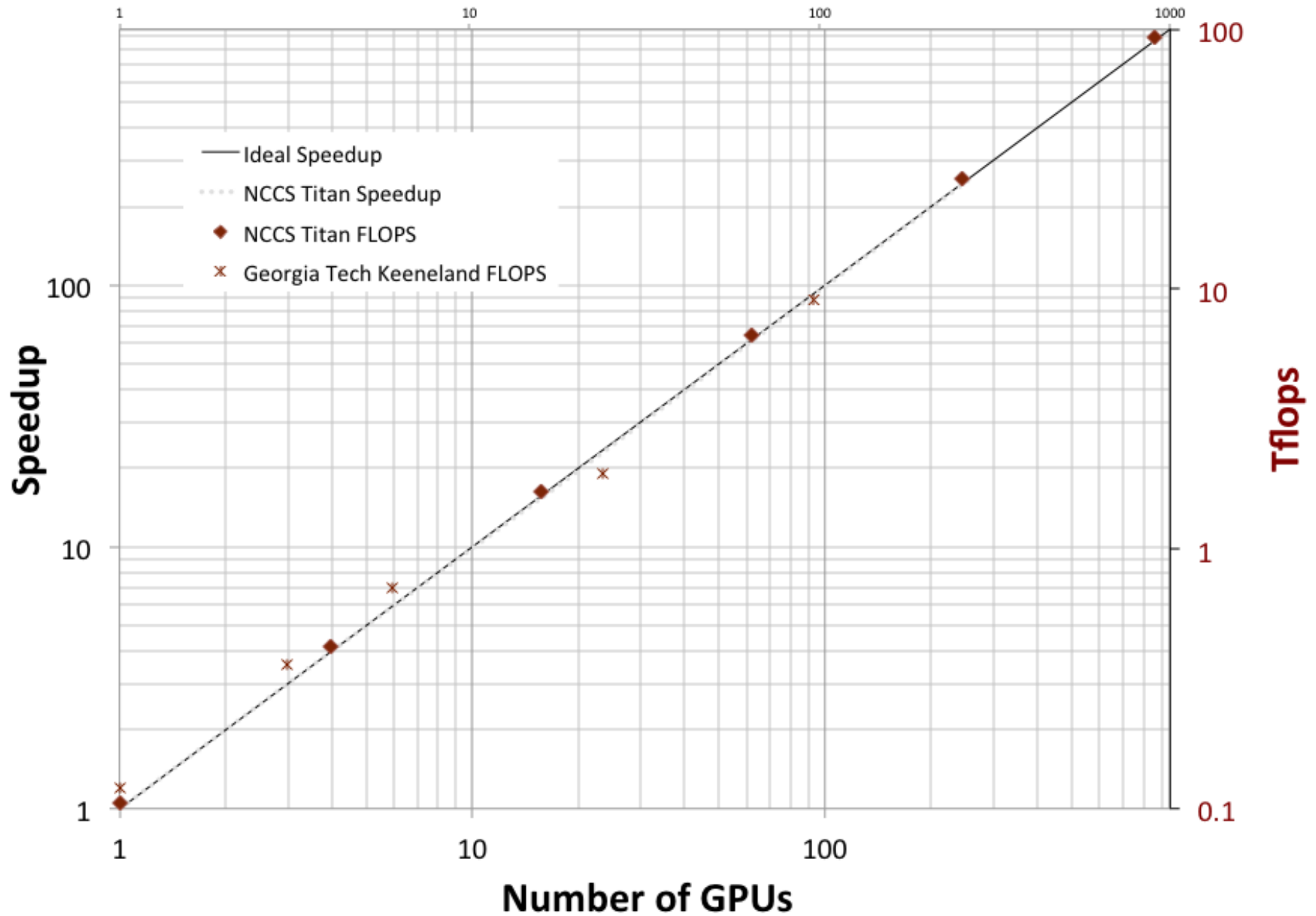
(Zhou et al., in preparation, 2012)

Multi-GPU Overlapping Procedure



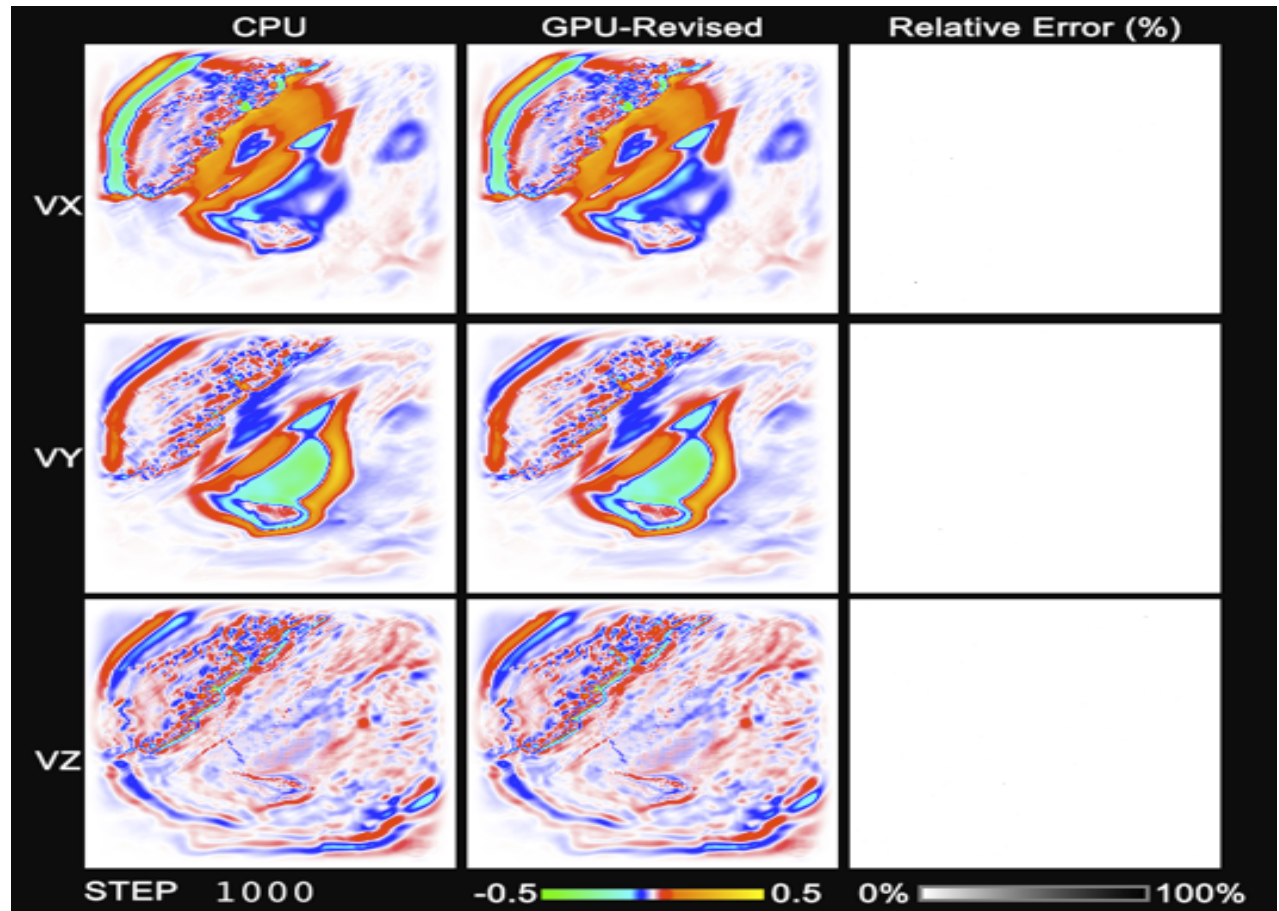
(Zhou et al., in preparation, 2012)

Parallel Multi-GPU Performance

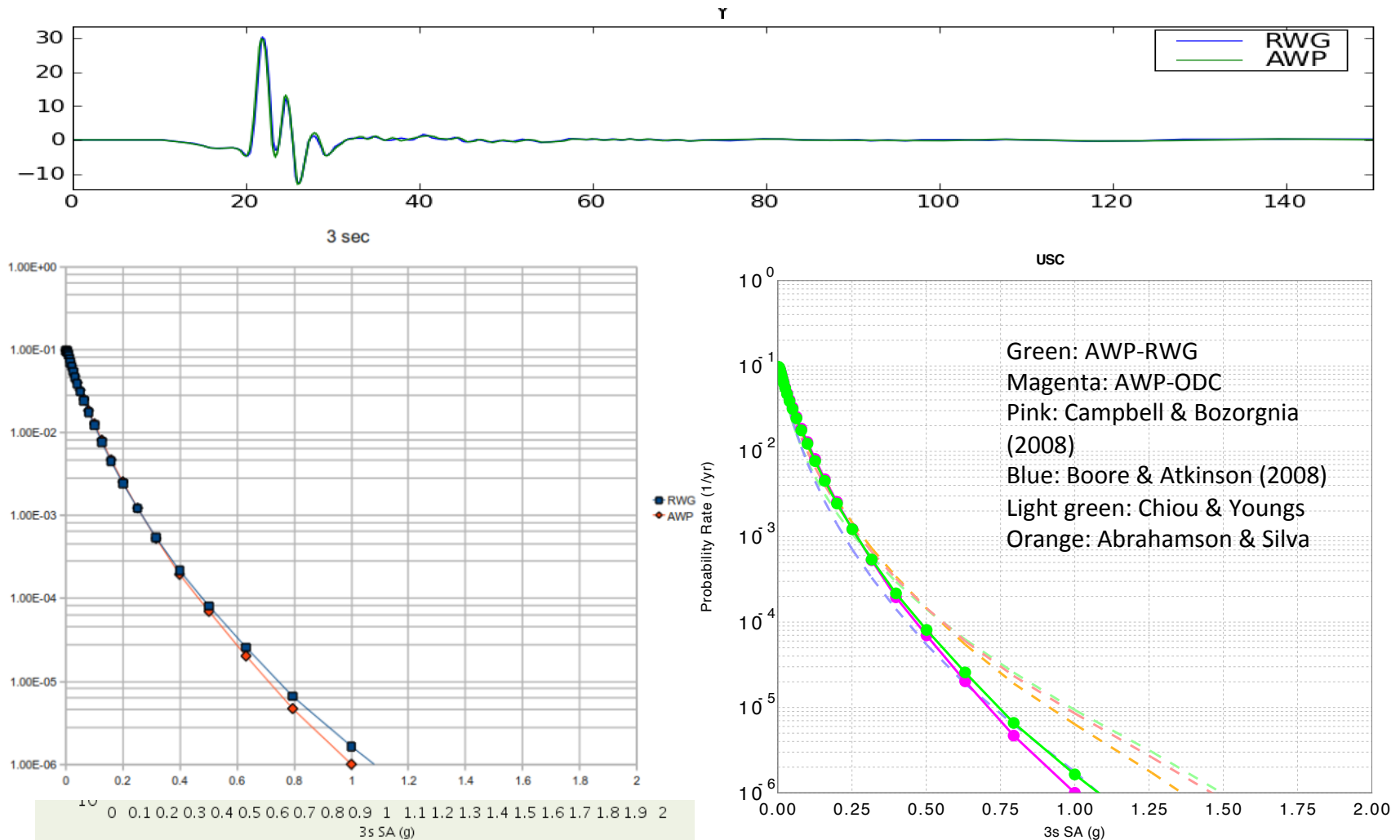


Verification with CPU Code using Chino Hills Case

- CVM-H mesh
- $dx=100m$, $dt=0.006$, $tmax=6.0$
- Problem size $25.6km^3$
- Decomposition in y and z directions only for GPUs
- Point source, 66 steps
- Boundary condition cerjan ($nd=20$, $arbc=0.92$)
- $fl=0.01$, $fh=25.0$, $fp=0.5$
- Credits: Amit Chourasia, Jun Zhou, DongJu Choi (SDSC), Patrick Small (USC), Kim Olsen (SDSU)



Comparison of Hazard Curves at USC using AWP-ODC and AWP-RWG



(image courtesy of Scott Callaghan of SCEC)

Future Work

- Add strain Green tensors calculations to the kernel
- Complete implementation of parallel I/O for GPU code
- Continue verification of AWP-ODC against AWP-RWG code
- Develop hybrid multicore implementation for CPU-GPGPU, targeting heterogeneous computing using XSEDE resources such as NCSA Blue Waters
- Prepare production capabilities that can be used to calculate the statewide deterministic and probabilistic seismic hazard in California
- Ultimate goal is to develop a CyberShake model that can assimilate information during earthquake cascades for operational forecasting and early warning

XSEDE names first Campus Champions Fellows

- **Liwen Shih**, professor and computer engineering chair at the University of Houston-Clear Lake, is paired with Yifeng Cui. The project focuses on Cui's XSEDE advanced support work on physics-based seismic research in collaboration with Southern California Earthquake Center (SCEC). Thomas Jordan at USC is the SCEC principal investigator. Phil Maechling is SCEC's information technology architect and also a member of the XSEDE Advisory Board.
- Campus Champions are volunteers who advise researchers on the use of high-end cyberinfrastructure (including XSEDE resources) at their respective campuses. The goal of the Fellows program is to increase expertise on campuses by including CCs as partners in XSEDE's ECSS projects.



End