Extreme-scale Earthquake Simulation with MVAPICH

Yifeng Cui (yfcui@sdsc.edu)
San Diego Supercomputer Center
MUG'23, Aug 21-23, 2023

AWP-ODC Evolution &
Large-scale Earthquake Simulation

HPGeoC
HIGH PERFORMANCE GEOCOMPUTING LABORATORY

Why High Frequency Earthquake Modeling

Seismic band

low-order free oscillations — *mantle waves* — *crustal waves* — *basin waves* — *strongly scattered waves*

period	1000 s	100 s	10 s	1 s	0.1 s
frequency	.001 Hz	.01 Hz	.1 Hz	1 Hz	10 Hz

tall buildings — *houses* — *stiff structures*

Earthquake engineering band

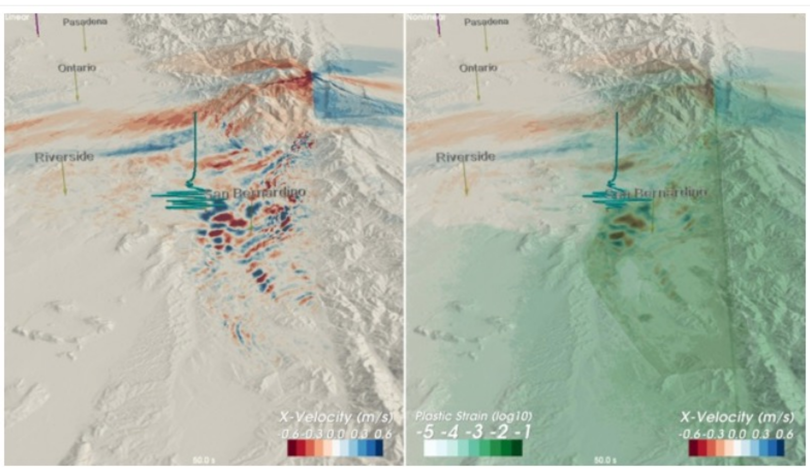
SAF simulation 2023

physics-based deterministic

4 Hz

CyberShake 1 Hz

empirical stochastic



(Snapshots from linear (left) and nonlinear (right) simulations using AWP-ODC showing wave propagation during a magnitude 7.7 SAF earthquake, Roten, SC'16)

Must validate new physics

- fault roughness
- near-fault plasticity
- frequency-dependent attenuation
- Topography
- small-scale near-surface heterogeneity
- multi-surface nonlinearity

8 Hz

SAF Simulation 2026

physics-based deterministic

physics-based stochastic

empirical stochastic

HPGeoC High Performance Geocomputing Laboratory

AWP-ODC

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

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

solved by explicit staggered-grid 4th-order FD

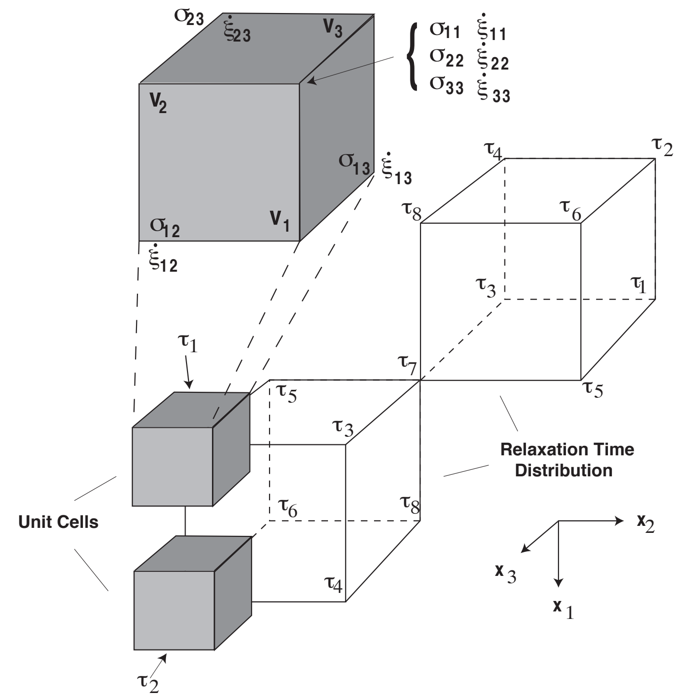
- **Memory variable formulation of inelastic relaxation**

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

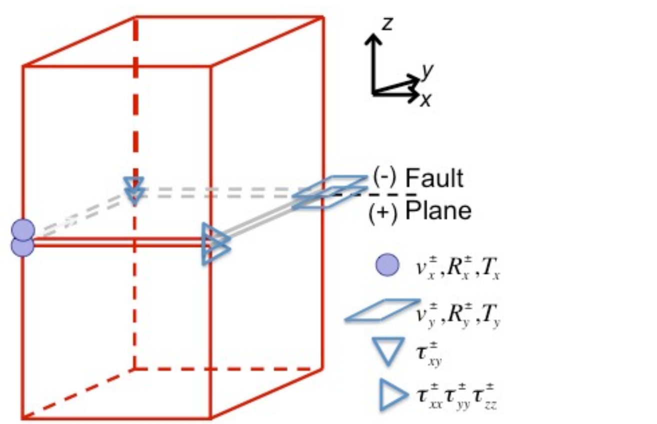
$$Q^{-1}(\omega) \approx \frac{\delta M}{M_u}\sum_{i=1}^{N}\frac{\lambda_i\omega\tau_i}{\omega^2\tau_i^{\,2}+1}$$

using coarse-grained representation (Day 1998)

- **Dynamic rupture by the staggered-grid split-node (SGSN) method (Dalguer and Day 2007)**
  - **Displacement nodes split at fault surface: explicitly discontinuous displacement & velocity**
  - **All interactions between sides occur through traction vector at displacement node**
- **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



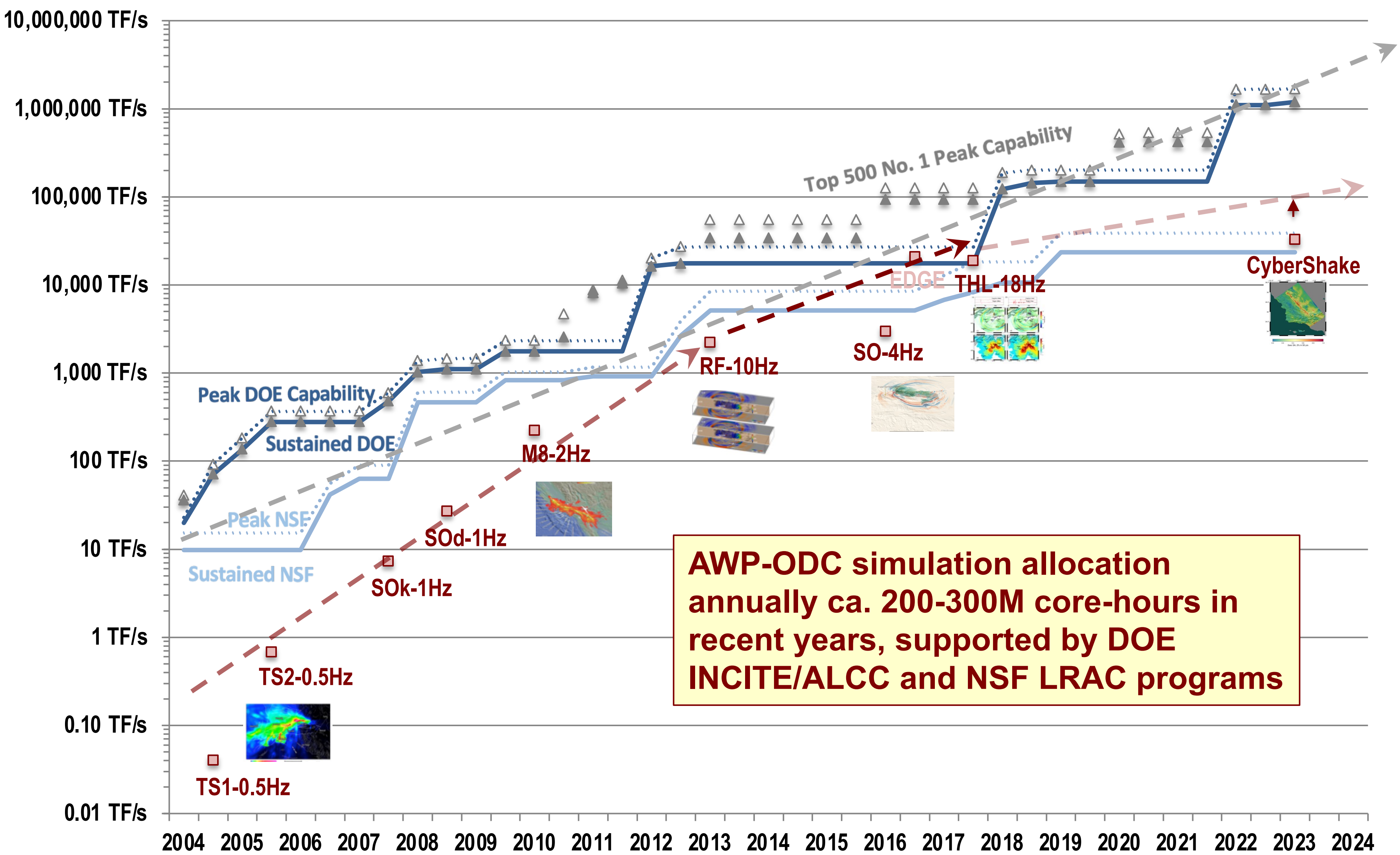
Variables:

$V_i^\pm$	split-node particle velocities
τ_{ij}	stresses
$T_i^\pm$	split-node traction (no jump)
$R_i^\pm$	stress divergence terms

HPGeoC
High Performance Geocomputing Laboratory

The Earthquake System Science Challenges at Extreme-Scale

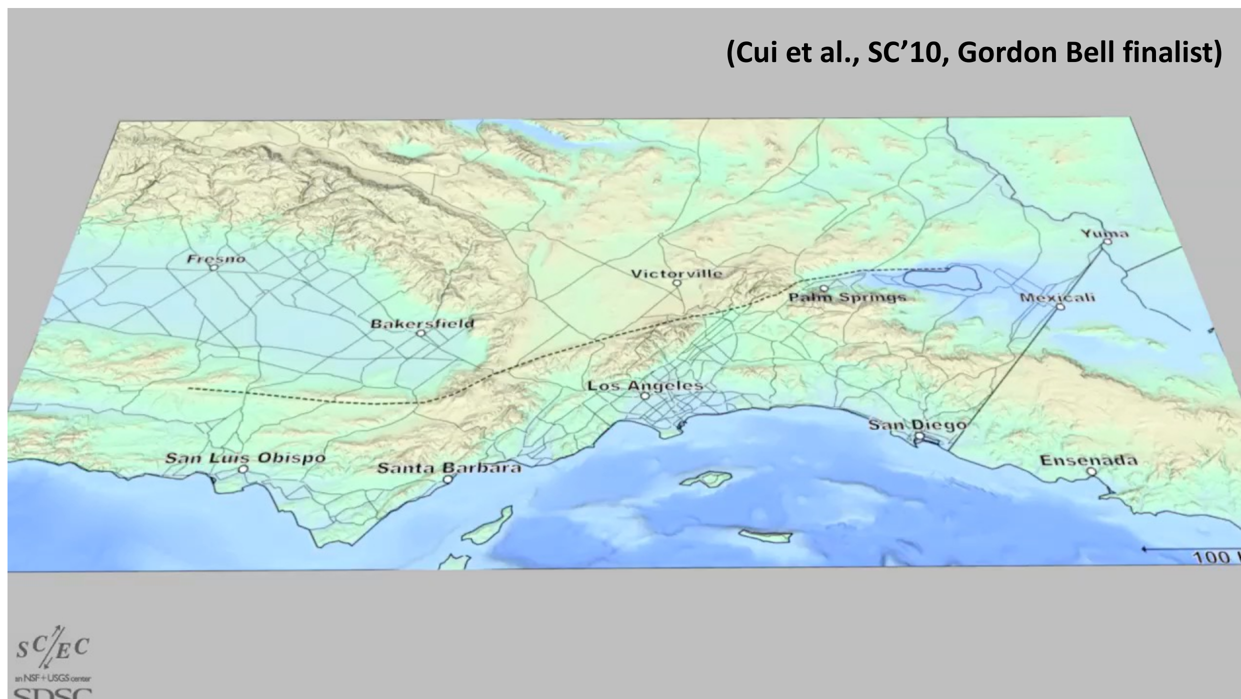
Evolution of AWP-ODC



inear Earthquake Simulation, 2010

- l scenario, worst-case for southern San Andreas Fault
  - wavelength: 200 m, NW→SE rupture propagation
- on performed on Kraken, 7.5 hours using 2160 cores
  - ure
-

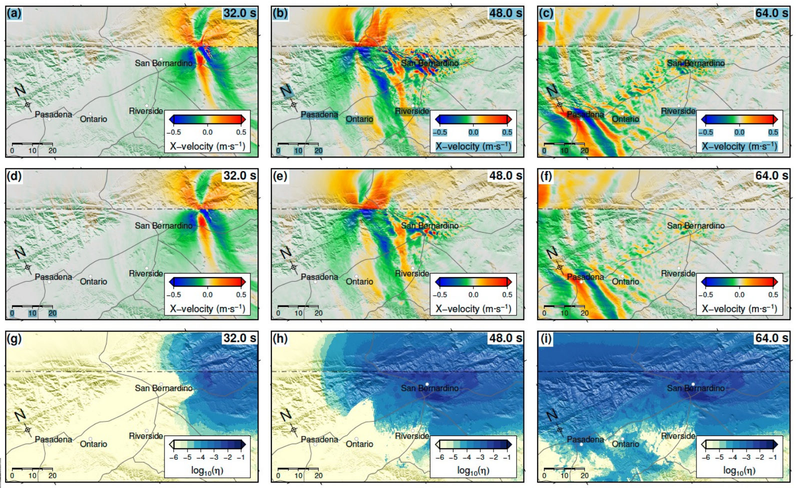
DK Panda team was part of the M8 effort



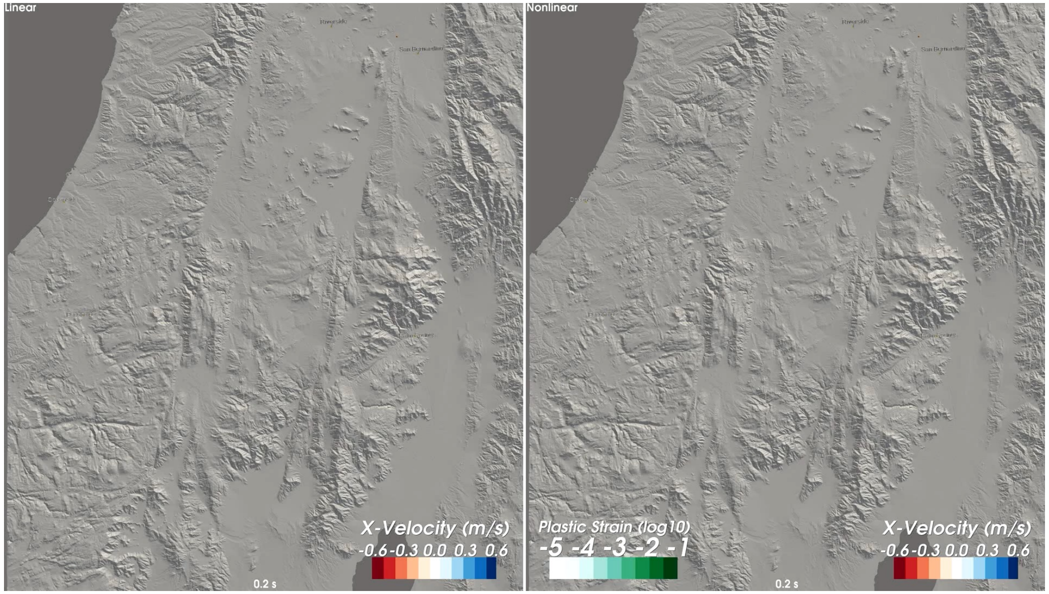
(Cui et al., SC'10, Gordon Bell finalist)

0-4 Hz Single-surface J2 Nonlinear ShakeOut Simulation, 2016

- **A First 4-Hz nonlinear M7.7 earthquake simulation on the southern San Andreas Fault**
- **Nonlinear dynamic rupture simulation was conducted using 24,000 CPU-cores on Blue Waters, running 37 hrs**
- **Nonlinear wave propagation simulation was conducted using 4,200 GPUs on Titan, running 12 hours**
- **Initially 400% computing time required compared to linear code. With optimized yield factor interpolation, this reduces the computing time from 400% to 165% only**



(Roten, et al., SC'16)



(Roten et al., SC'16)

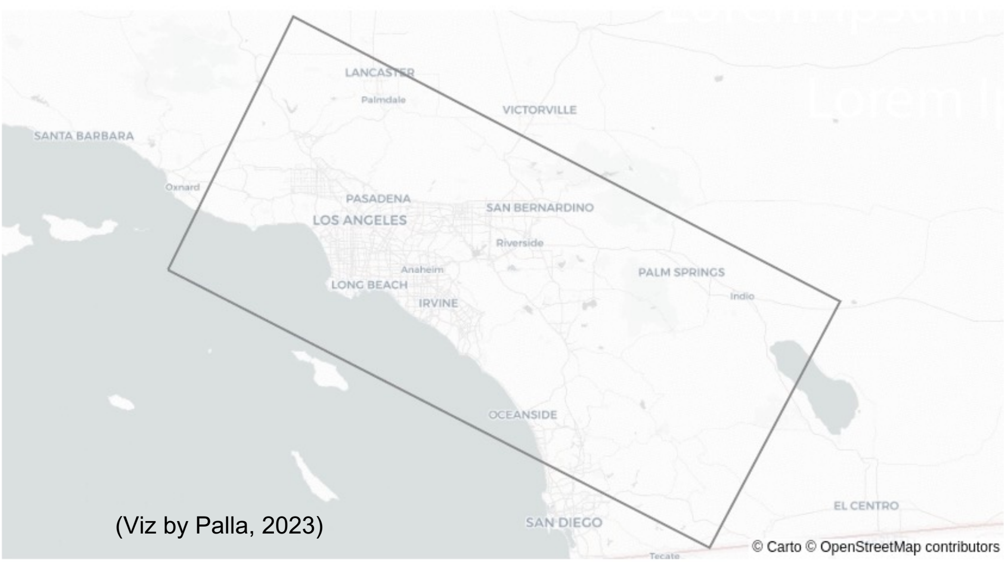
(Roten et al., 2016)

- **Inside the Whittier Narrows corridor, spectral accelerations at 3 seconds (3s-SAs) are reduced from 1g in the linear case to 0.3-0.6g in the nonlinear case, depending on the choice of reference strain.**
- **Plastic simulations obtained with a single von Mises yield surface predict 3s-SAs that are higher than those obtained with the multi-surface Iwan model, but lower than the linear values.**

https://www.youtube.com/watch?v=qOH0Oj3t6QM

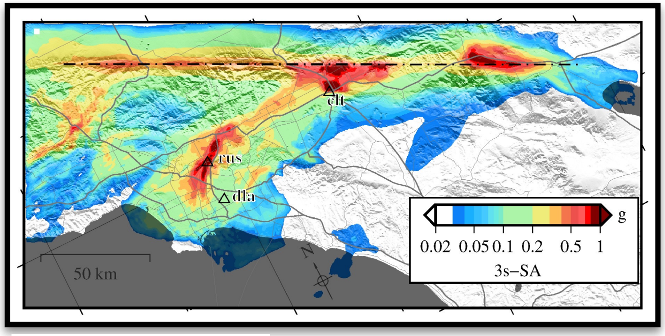
0-4 Hz Multi-surface Iwan Nonlinear ShakeOut Simulation, 2023

- **A multi-surface Iwan type plasticity model in AWP-CPU, verified against the established codes for 1D and 2D SH-wave benchmarks, has been applied to predict the impact of realistic soil nonlinearity on long-period surface waves during large earthquakes on the southern San Andres fault**
- **While ShakeOut simulations with a single yield surface reduces long period ground motion amplitudes by ~25% inside a wave guide in greater LA, Iwan nonlinearity further reduces the values by a factor of two**
- **Computational requirements with Iwan model is 20-30x more expensive, and memory use 5-13x more compared to linear solution**
- **Run 22.5 hrs using 7,680 TACC Frontera nodes**

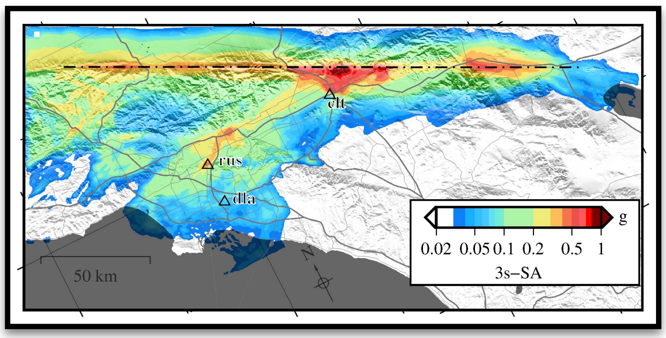


(Viz by Palla, 2023)

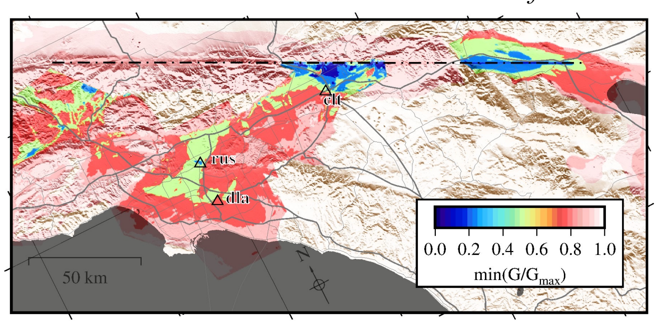
Linear



Iwan (*Darendeli*)



Max. shear modulus reduction at the surface



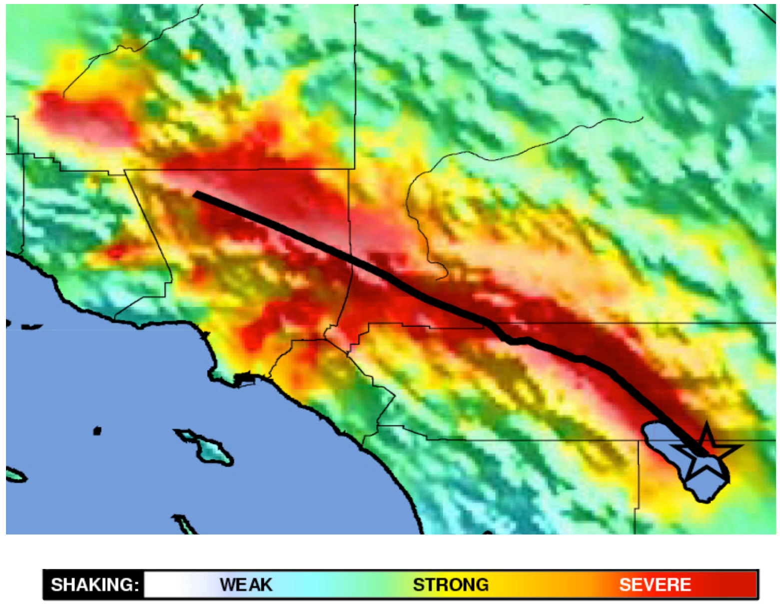
(Roten et al., BSSA, 2023, accepted)

The ShakeOut Scenario

M7.8 Earthquake on Southern San Andreas Fault

Scenario Results

- **M7.8 mainshock**
  - **Broadband ground motion simulation (0-10 Hz)**
- **Large aftershocks**
  M7.2, M7.0, M6.0, M5.7…
- **10,000-100,000 landslides**
- **1,600 fire ignitions**
- **$213 billion in direct economic losses**
  - **300,000 buildings significantly damaged**
  - **Widespread infrastructure damage**
  - **270,000 displaced persons**
  - **50,000 injuries**
  - **1,800 deaths**
- **Long recovery time**



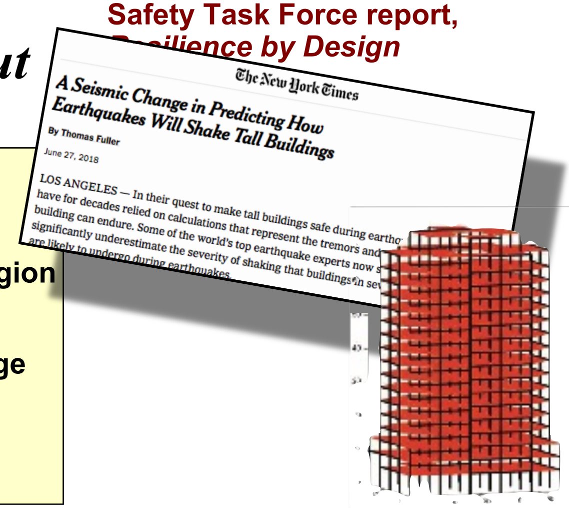
shaking intensity

Great Southern California ShakeOut

November 13, 2008

Exercise Results

- **Largest emergency response exercise in US history, 45M people worldwide participating in 2022**
  - **Golden Guardian exercise**
  - **Public events involving multi-million registered participants**
- **Demonstrated that existing disaster plans are inadequate for an event of this scale**
  - **Motivated reformulation of system preparedness and emergency response**
  - **Scientific basis for the LA Seismic Safety Task Force report, *Resilience by Design***



Waveguide amplification in LA Basin

- **Caused by string of contiguous sedimentary basins (Olsen et al, 2006, 2009)**
- **ShakeOut scenario predict strong long-period ground motions in Los Angeles region**
- **Hazard to pre-Northridge high-rise buildings**
- **All these approaches assume a linear stress-strain relationship in the fault damage zone and shallow sediments**
- **Simulations with DP-plasticity predict 30-70% lower ground motions than linear solutions (Roten et al., 2014, 2017)**

Porting to Various CPU Architectures - 2010

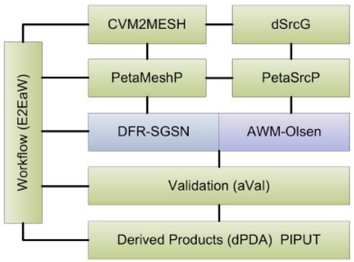
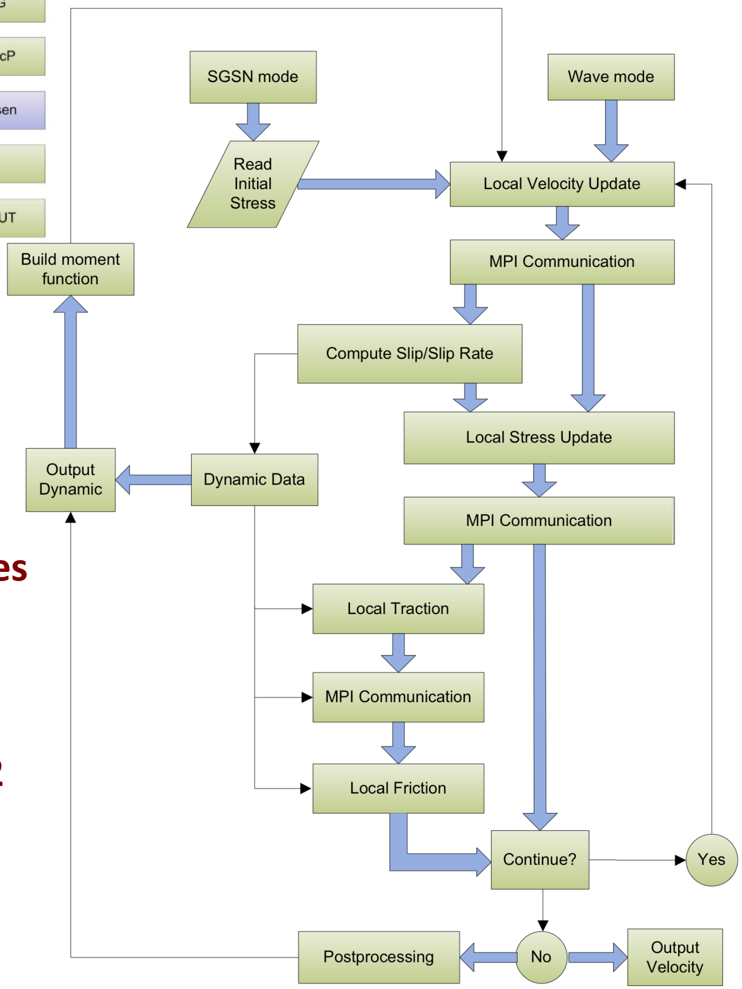


Fig. 4. Components of AWP-ODC



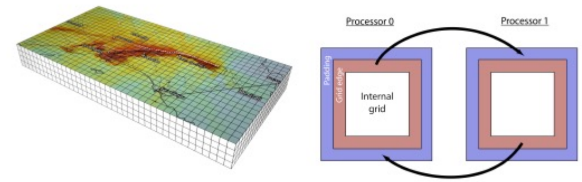


Fig. 5: (left) Decomposition of the M8 simulation region with 810 km long, 405 km wide and 85 km deep; (right) communication between neighboring subgrids

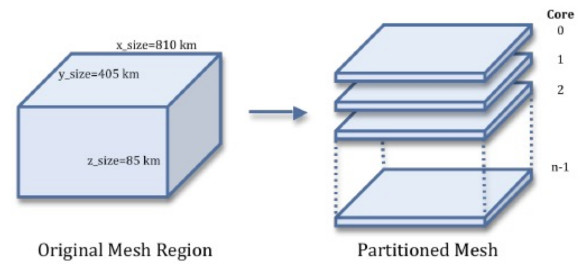


Fig. 7. The 3-D mesh region is partitioned into slices along the z-axis. Each slice is assigned to a core in the MPI job, and each core queries the underlying CVM for the points in its slice only.

AWP-ODC I/O features have been converted to a separate library called SEISM-IO, supported by NSF SI2 program

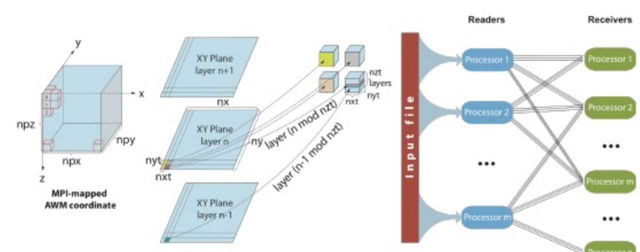


Fig. 9. (left) Cubes and (center) planes for contiguous burst reading and efficient data distributing; (right) high performance I/O with data redistribution

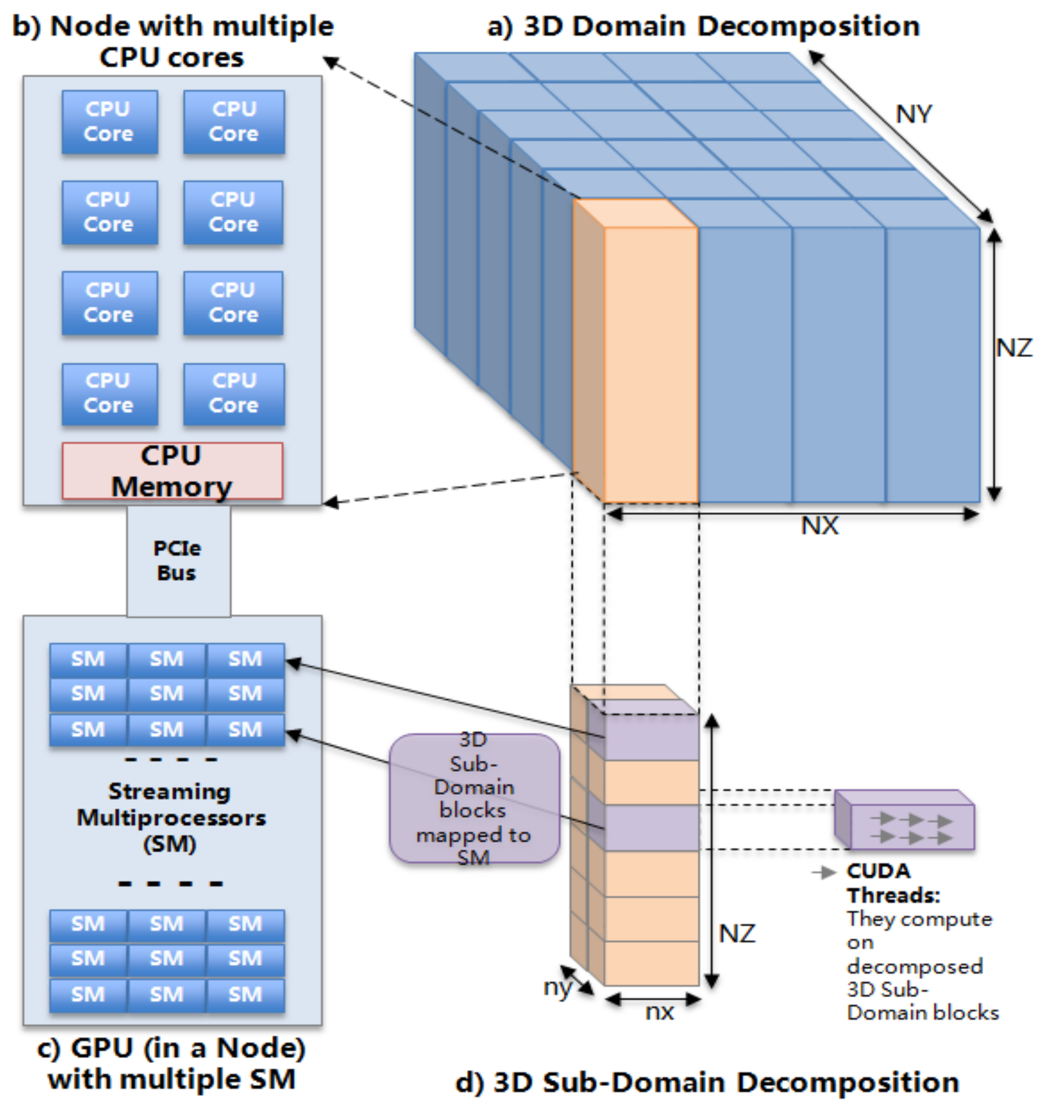
(Cui et al., SC'10)

HPGeoC
High Performance Geocomputing Laboratory

Porting to GPUs – 2012

- **Two-layer 3D domain decomposition on CPU-GPU based heterogeneous supercomputers**
  - **first step X&Y decomposition for CPUs**
  - **second step Y&Z decomposition for GPU SMs**

(Zhou et al., ICCS 2012, Cui et al., SC'13)



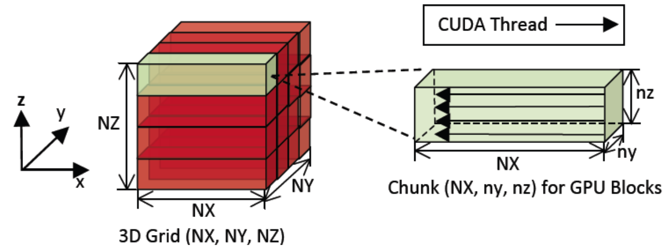
Single-GPU Optimizations - 2012

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

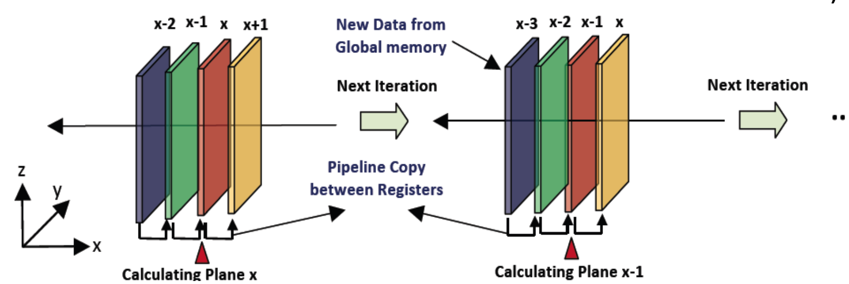
  Global memory coalesced, texture memory for six 3D constant variables, constant memory for scalar constants
- ✓ **Step-4: Register Optimization**

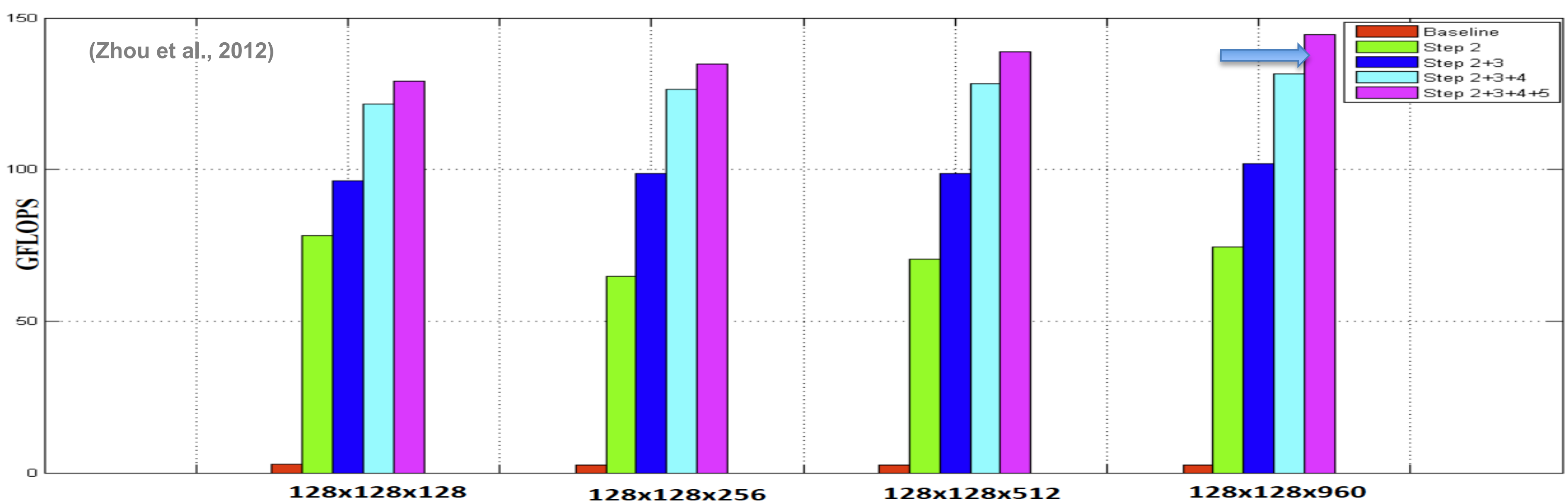
  Pipelined register copy to reduce memory access
- ✓ **Step-5: L1/L2 cache vs shared memory**

  Rely on L1/L2 cache rather on-chip shared memory

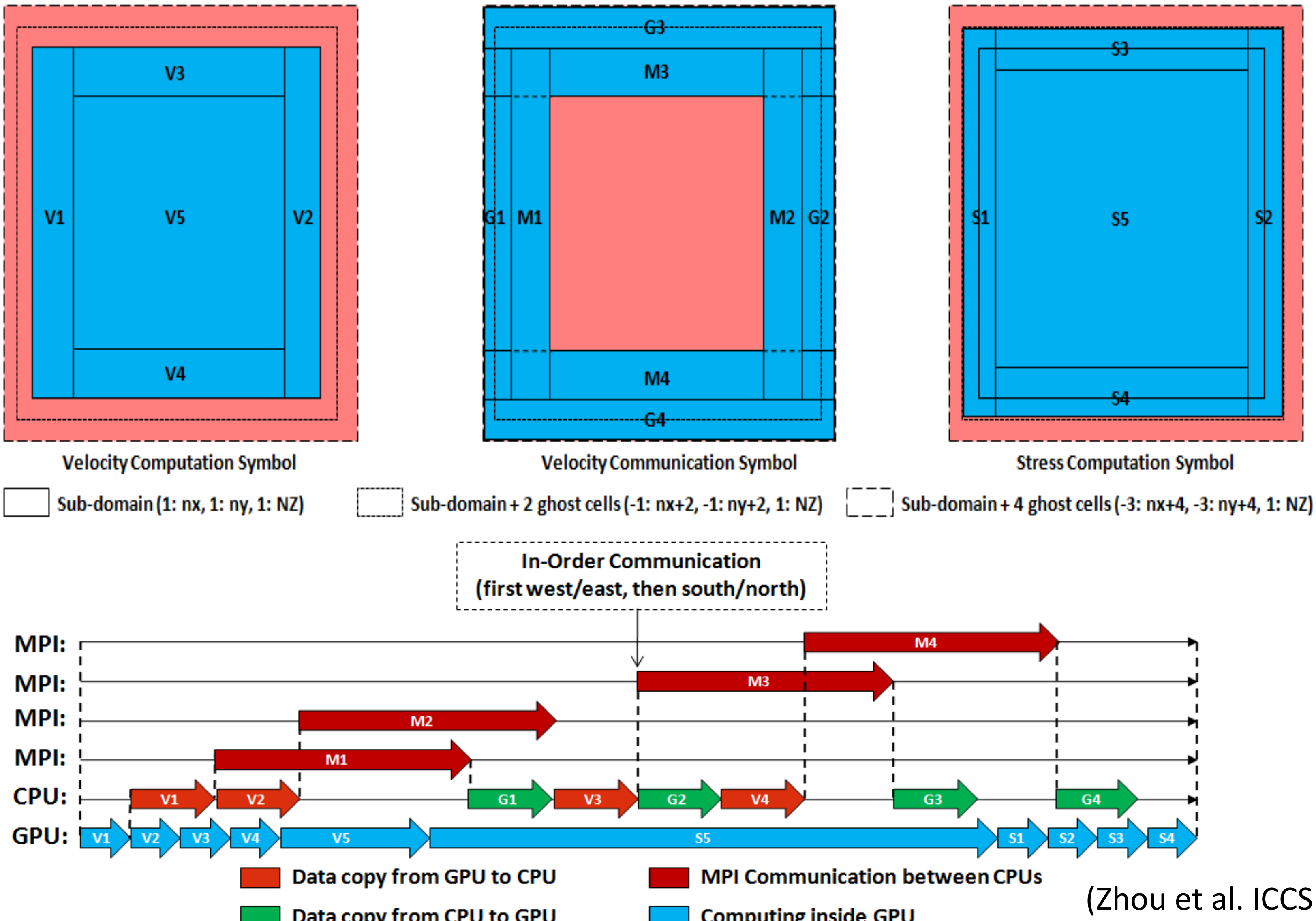


(Zhou, J et al., ICCS 2012)





Computing and Communication Overlapping on GPUs - 2013



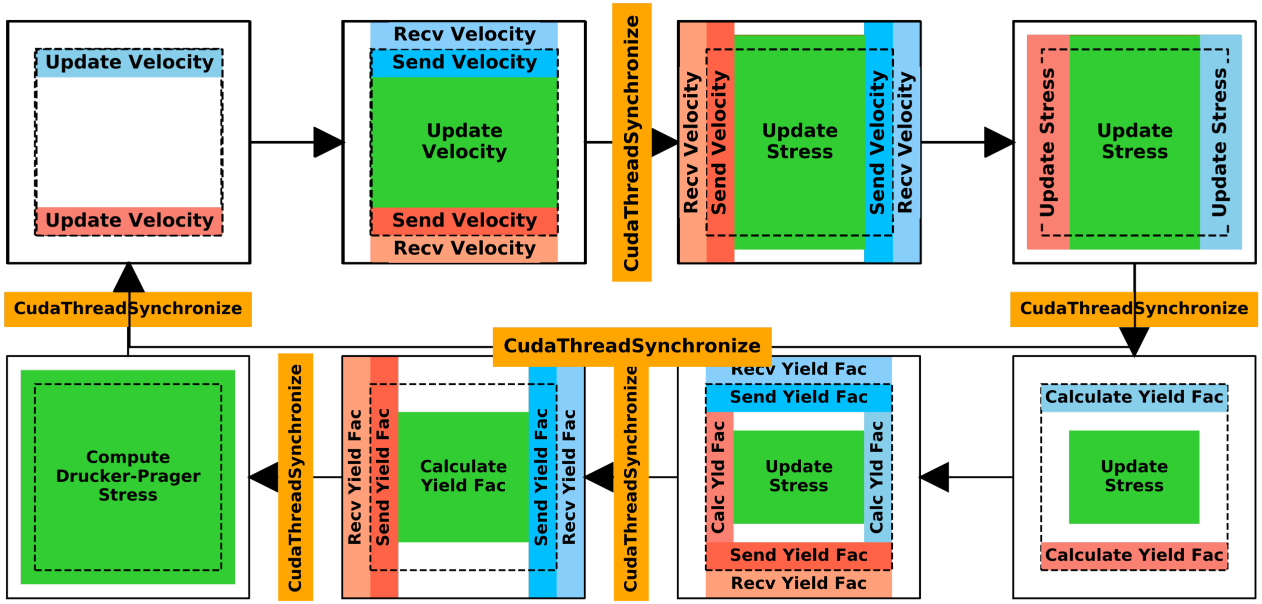
(Zhou et al. ICCS 2013,
Cui et al. SC'13)

Porting DP-Plasticity on GPUs – 2016

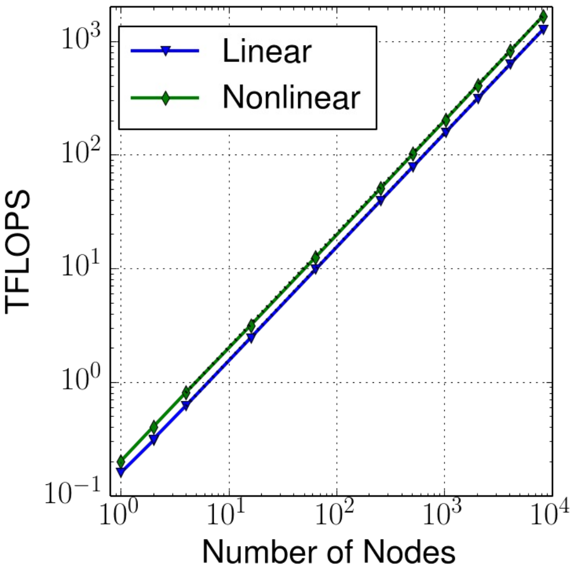
Communication / computation schedule

Nonlinear case

Linear case



OLCF Titan
Weak Scaling Test

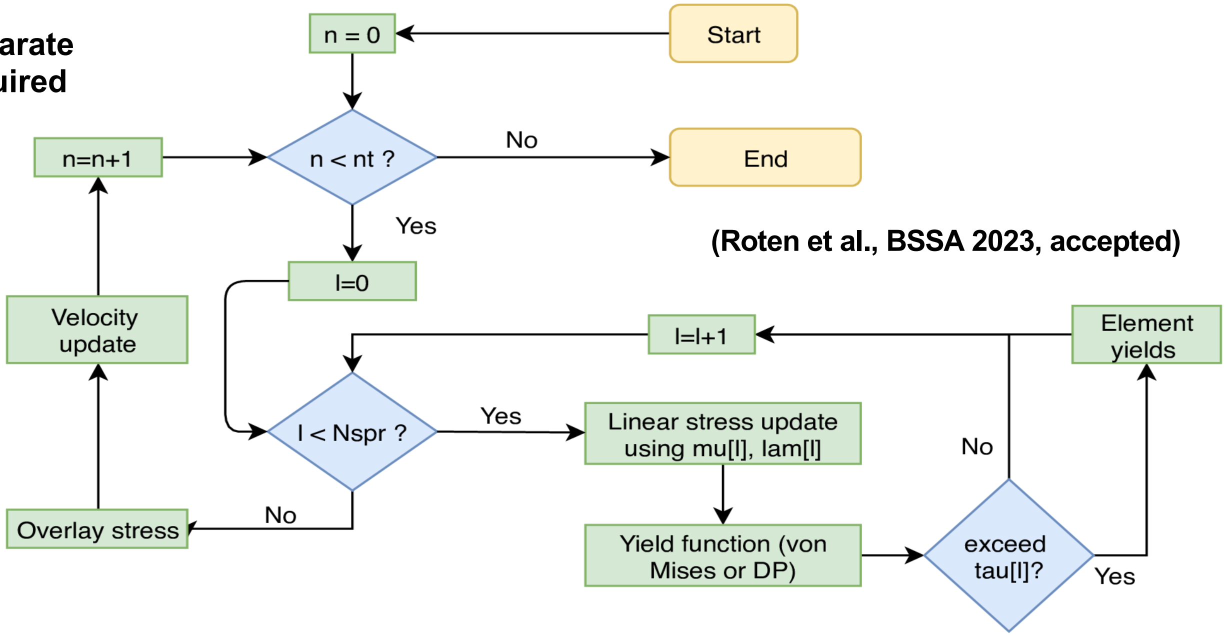


(Roten et al., SC'16)

Porting Iwan Model on CPUs and GPUs – 2021

❖ Computational challenges:

- Computationally expensive: separate stress and plasticity update required for each yield surface
- Memory requirements: each yield surface requires a separate copy of stress tensor τxx, τyy, , τzz, τxz, τyz, τxy, Lamé parameters μ, λ, and yield factor r.
- MPI communication overhead: stress tensor and yield factor of each yield surface needs to be swapped during each time step (reduced scalability)

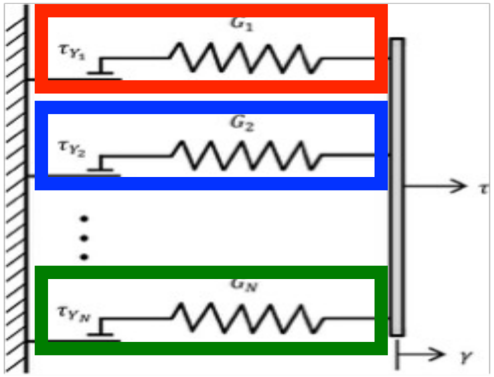


(Roten et al., BSSA 2023, accepted)

- Shear modulus reduction reduces max. resolvable frequency
- 10-20x more expensive compared to our 2016 nonlinear simulation which used a simple J2 nonlinear material model, or 20-30x compared to linear solution
- Memory increased by (1 + 0.4* Nspr) to linear simulation (*Nspr = nr of yield surfaces)*

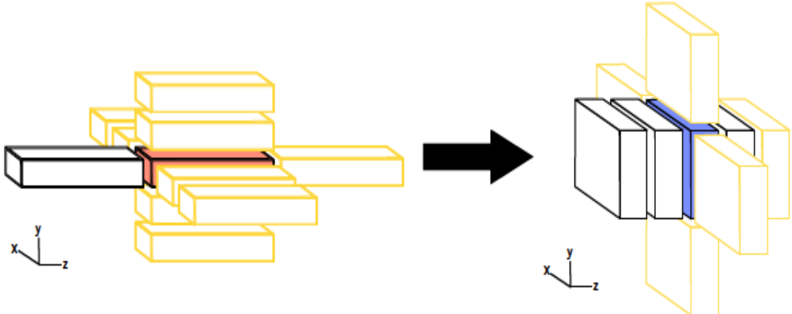
❖ Iwan Concept

- Hysteretic yielding behavior of material represented by a collection of perfectly elasto-plastic spring-slider elements, each element has different constants, shared strain and a fraction of stress, generalized to 3D using a collection of Drucker-Prager yield surfaces



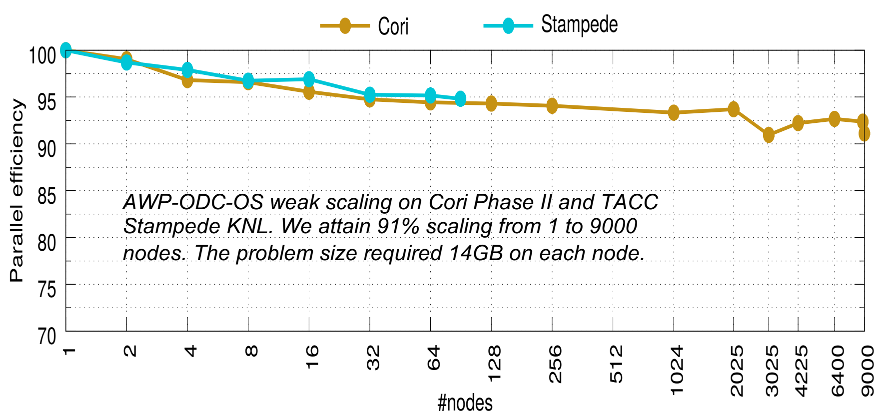
Porting to Intel Xeon Phi – 2017

- **Stencil generation and vector folding through YASK tool: https://github.com/01org/yask**
- **Hybrid placement of grids in DDR and MCDRAM**
- **Normalized cross architecture evaluation in Mega Lattice Updates per Second (MLUPS): Xeon Phi KNL 7290 achieves 2x speedup over NVIDIA K20X, 97% of NVIDIA Tesla P100 performance**
- **Performance on 9,000 nodes of Cori-II equivalent to performance of over 20,000 K20X GPUs at 100% scaling**
- **Memory bandwidth accurately predicts performance of architectures (as measured by STREAM and HPCG-SpMv)**



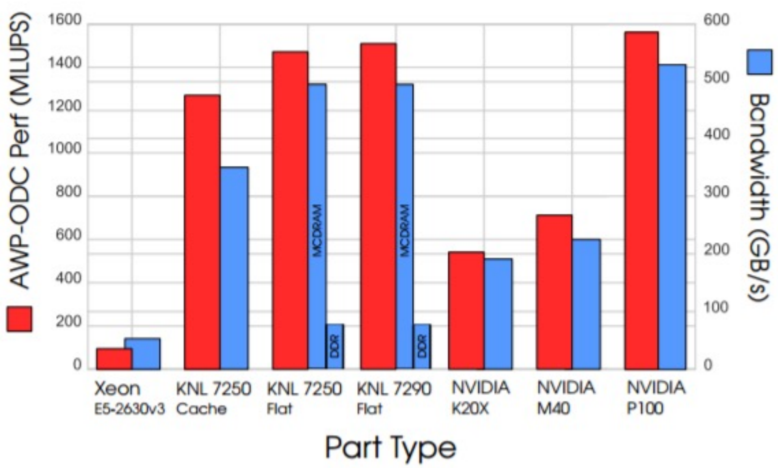
Memory dependence for a radius 2 stencil with a conventional memory layout. In orange is a single cache-line of the data array, in gold are the required new cache-lines that must be read for an update

Memory dependence for a radius 2 stencil with a 4x4x1 vector fold. In blue is a single cache-line of the data array, in gold are the required new cache-lines that must be read for an update.



AWP-ODC-OS weak scaling on Cori Phase II and TACC Stampede KNL. We attain 91% scaling from 1 to 9000 nodes. The problem size required 14GB on each node.

(Tobin et al., ISC'17)



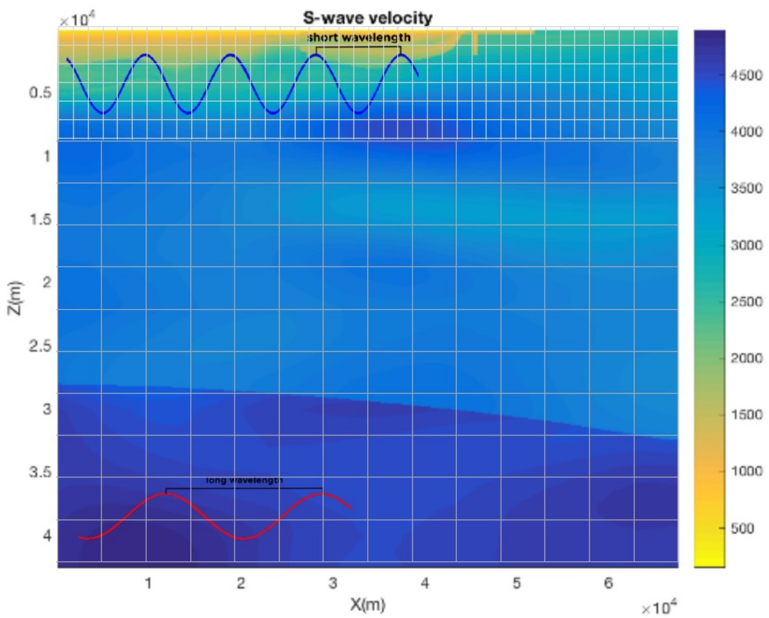
Single node performance comparison of AWP-ODC-OS on a variety of architectures. Also displayed is the bandwidth of each architecture, as measured by a STREAM and HPCG-SpMv.

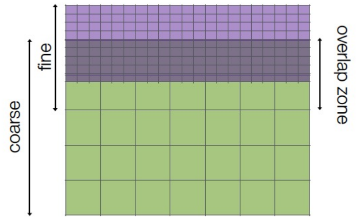
Porting Discontinuous Mesh on GPUs – 2018

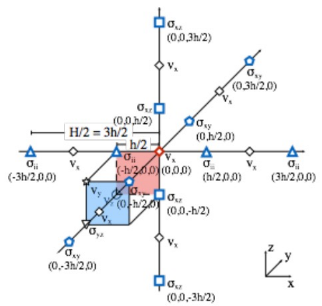
Kim Olsen, SDSU

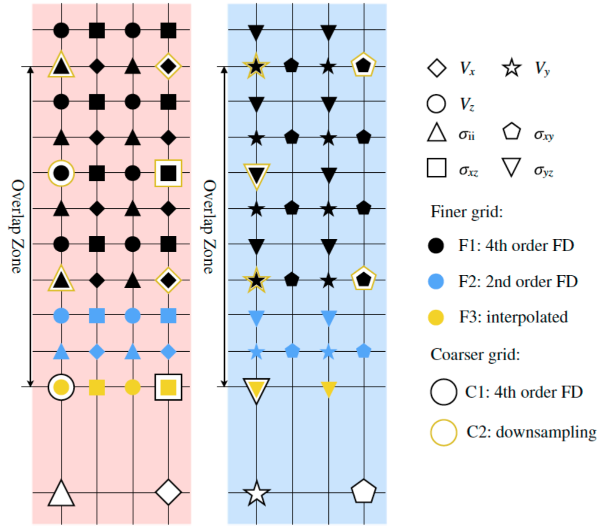
- **Let the interpolation be expressed as: $u = W * U$,**
  where U is the field value on the coarse grid, u is the missing point on the fine grid and W is the interpolation operator matrix
- **Corresponding downsampling method: $U' = W^T * u'$,**
  where u' is the field in the fine grid region, U' is located in the coarse grid, and we set W^T as downsampling matrix
- **Significant performance improvement with respect to a uniform grid solution**
  - A factor of 4 achieved for simulating the M9 megathrust earthquake in Cascadia, 650x1000x60 km³, 100/300m mesh sizes
  - A factor of 8 achieved for simulating the Mw 5.1 La Habra earthquake up to 4 Hz, using a grid spacing of 20 m in the fine grid and a minimum shear-wave velocity of 500 m/s

wavelength = velocity/frequency









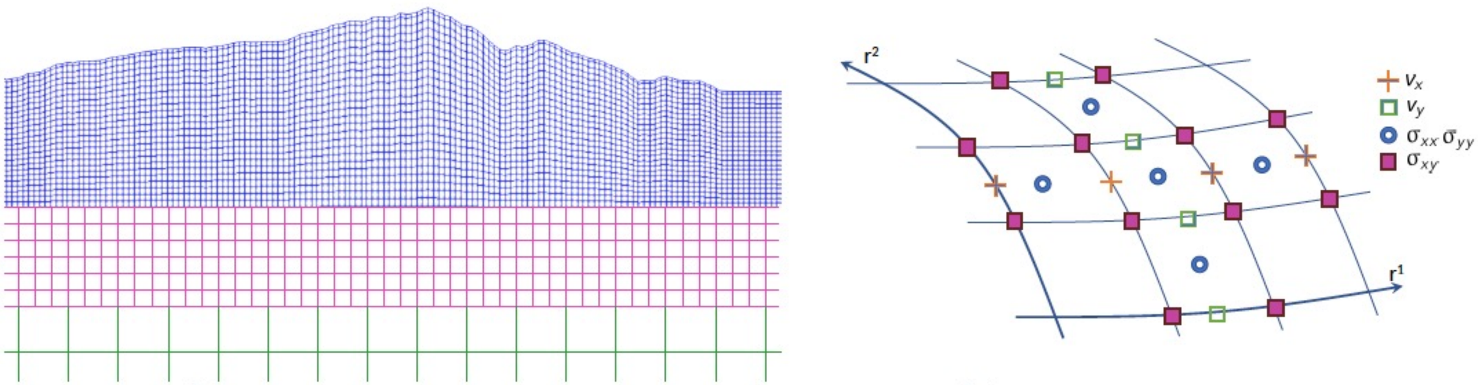
Using a DM with dx^{fine} = 100 m in upper 1 km, dx^{coarse} = 300 m in bottom 39 km, resulting in 0.28B grids or 72% reduction in grid points

(Nie et al., BSSA 2017, Roten et al., 2018)

Porting to Topography – 2019

Christine Goulet, USC

- ❖ **Topography has been added to AWP-ODC in GPU code, a separate version using curvilinear grids**
- ❖ **Comparable accuracy to the code on a Cartesian grid, with negligible extra memory requirements, longer simulation times due to small timesteps for complex topography**
- ❖ **Perfectly recover a forward simulation using reciprocity – a key result needed for CyberShake-related work**
- ❖ **94% weak scaling efficiency tested up to 1024 GPUs**
- ❖ **Future plan is to let this curvilinear grid rest on top of layers of Cartesian grids that extend downward with depth**



(a) Nonconforming multiblock grid (b) Curvlinear staggered grid

Figure 1: (a) Curvilinear grid, used for discretizing topography, overlaying cartesian grids with decreasing grid resolution with depth. (b) Arrangement of velocity and stresses in a curvilinear staggered grid

(O'Reilly et al., BSSA 2022)

HPGeoC
High Performance Geocomputing Laboratory

Porting to Microsoft Azure – 2022
Co-PI: Hari Subramoni

Challenges

- Digesting the wide breadth of options and configurations
- Higher threshold of initial setup needed
- Lack of comprehensive forums for debugging errors

- **Microsoft Internet2/Azure Accelerator for Research Fall 2022 program, $7k credits awarded through Cloudbank**
- **Future plan is to compare performance with MVAPICH2-AZURE**

Benefits

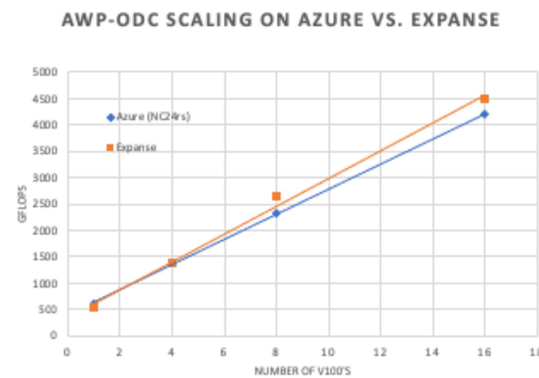
- Wide flexibility and options of hardware and software allows infrastructure to be tailored to specific workload
- Spin up large VM instances instantly without waiting in a queue/system quotas
- We demonstrated that the AWP performance with a benchmark of ground motion simulation on various GPU based cloud instances, and a comparison of the cloud solution to on-premises bare-metal systems.

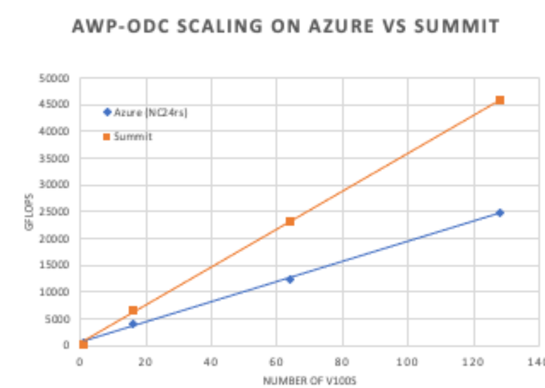
CloudBank

DASHBOARD LEARN ABOUT HELP LOG OUT

Accelerating Earthquake Simulation on Microsoft Azure

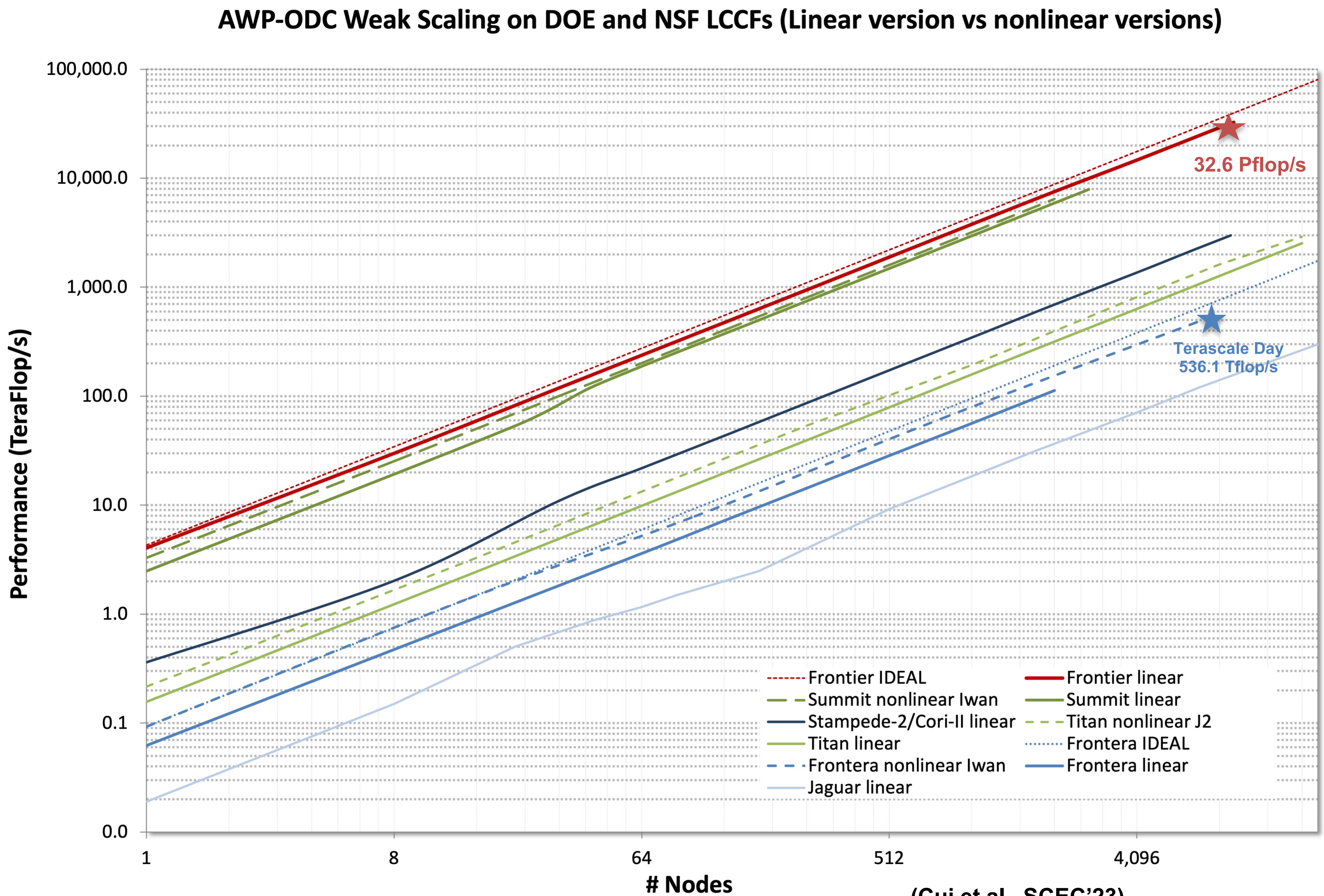
Specs	Azure (NC24rs)	Expanse	Summit
GPUs/Node	4 x V100	4 x V100	6 x V100
CPU	Xeon E5-2690 v4	Xeon Gold 6248	IBM Power 9
Memory/Node (GB)	480	384	512
Compiler:	OpenMPI	OpenMPI	IBM XL Compiler
File System:	NFS	Lustre	GPFS
Infinniband (Gbps):	FDR(56)	HDR(200)	EDR(100)





(Palla, SCEC'23)

Porting CUDA Linear Code to HIP – 2023

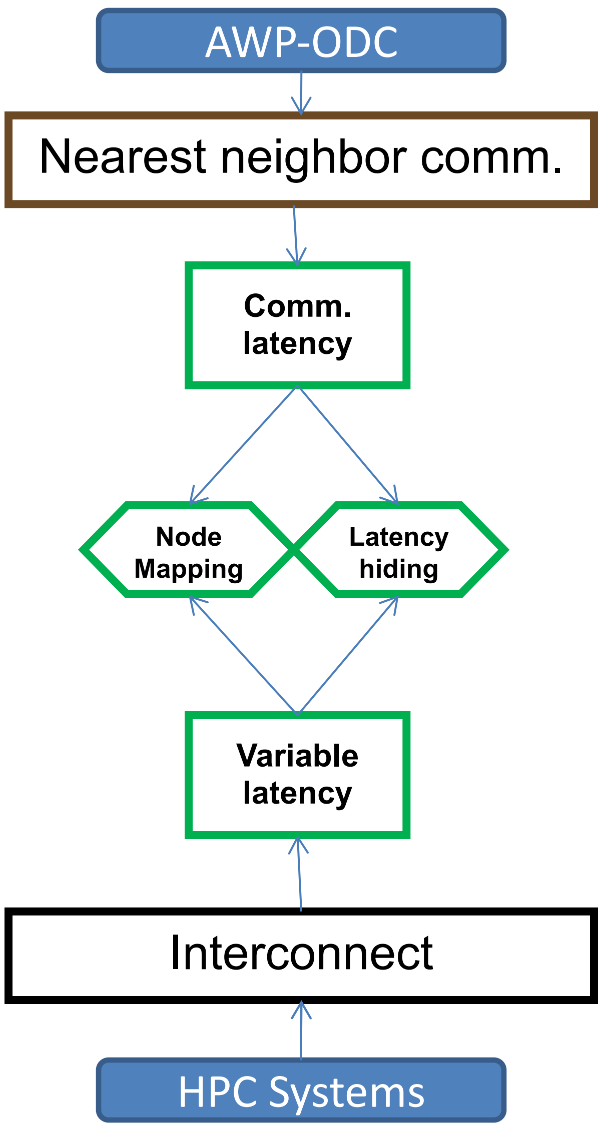


(Cui et al., SCEC'23)

Accelerate AWP-ODC Performance with MVAPICH

MVAPICH
MPI, PGAS and Hybrid MPI+PGAS Library

Communication Enhancement on CPUs - 2010



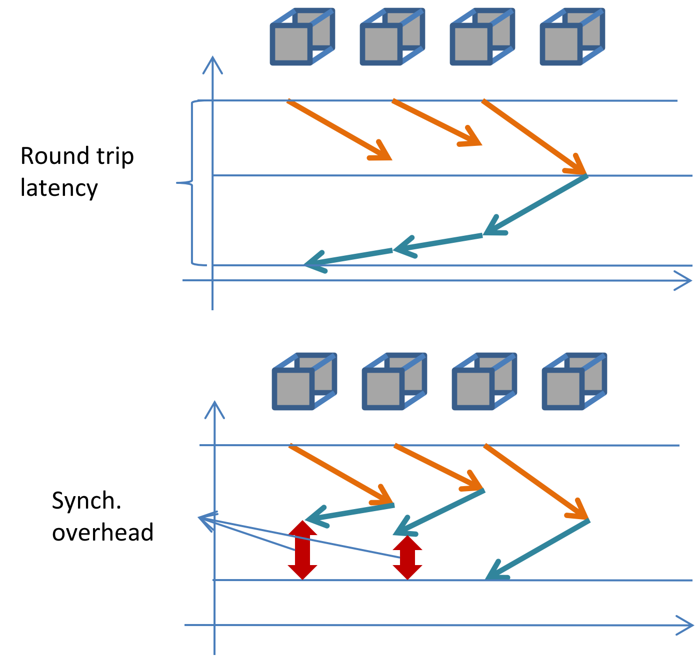
- **Performance challenge**
  - **Large variation in communication latencies among neighbors**
  - **System/user memory overhead**
- **Scalability challenge**
  - **Increased latency for larger simulation**

(Cui et al., SC'10)

- Asynchronous communication
- Rank re-placement
- Message pre-posting without data reorders
- Computation and communication overlapping, 2-sided and 1-sided

Communication Enhancement on CPUs - 2010

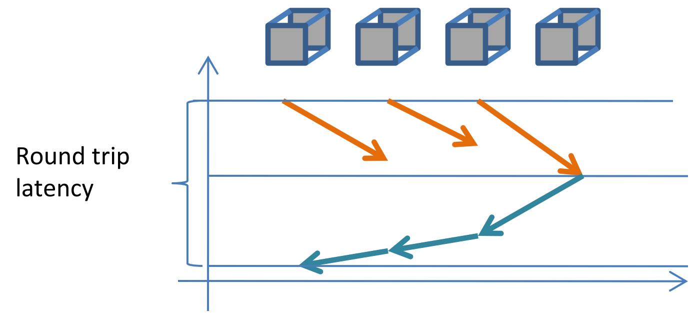
- **Asynchronous communication**
  - Significantly reduced latency through local communication
  - Reduced system buffer requirement through pre-post receives

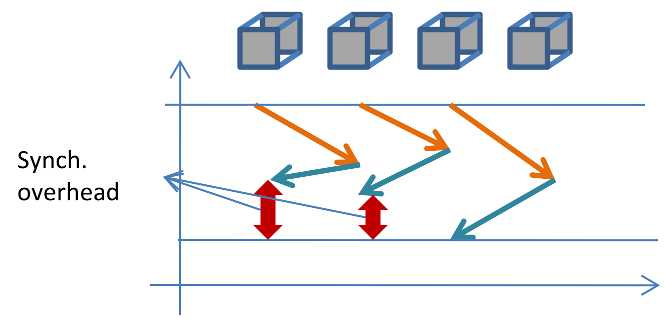


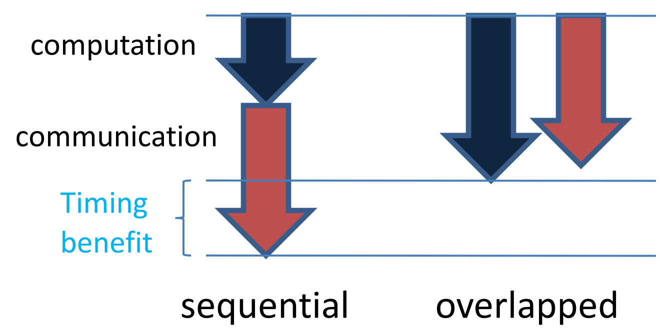
(Cui et al., SC'10)

Communication Enhancement on CPUs - 2010

- 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***







HPGeoC High Performance Geocomputing Laboratory

MVAPICH MPI, PGAS and Hybrid MPI+PGAS Library

Communication Enhancement on CPUs - 2010

Sreeram Potluri of DK Panda Team

- 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*
  - **MPI-1 non-blocking two-sided Communications**

```
Velocity Exchange
    s2n(u1,north-mpirank, south-mpirank)
    ! recv from south, send to north
    n2s(u1, south-mpirank, north-mpirank)
    ! send to south, recv from north
    ... repeat for east-west, up-down directions
    ... repeat for other velocity components v1,w1
    wait_onedirection()
    s2nfill(u1, recvbuffer, south-mpirank)
    n2sfill(u1, recvbuffer, north-mpirank)
    ... repeat for east-west, up-down directions
    ... repeat for other velocity components v1,w1

S2N
    Copy 2 planes of data from variable to sendbuffer
    !copy north boundary excluding ghost cells
    MPI_Isend(sendbuffer, north-mpirank)
    MPI_Irecv(recvbuffer, south-mpirank)

WAIT_ONEDIRECTION
    MPI_Waitall(list of receive requests)

S2NFILL
    Copy 2 planes of data from recvbuffer to variable
    ! copy to south ghost cells
```

(Potluri, S., et al., ICS'10)

Communication Enhancement on CPUs - 2010

Sreeram Potluri of DK Panda Team

- 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*
  - MPI-1 non-blocking two-sided Communications
  - **MPI-2 one-sided Communications (on Ranger)**

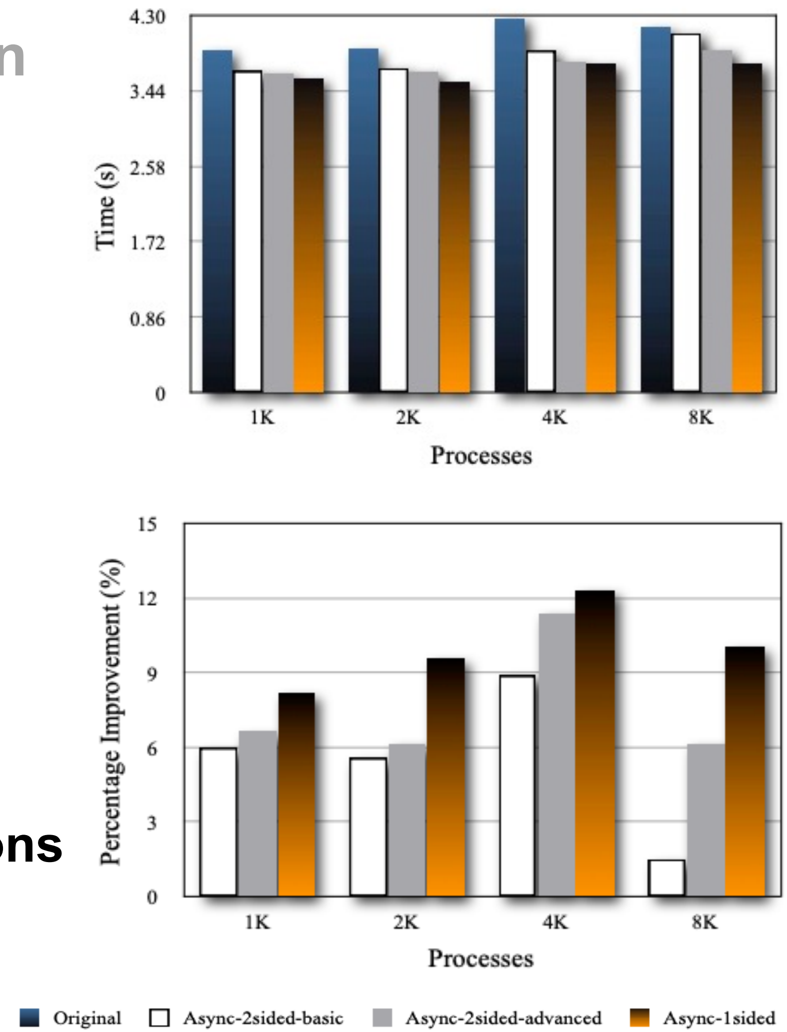
```
MPI_Win_post(group, 0, window) ! pre-posting the window
to all neighbors
Main Loop in AWM-Olsen
    Compute velocity component u
    Start exchanging velocity component u
    Compute velocity component v
    Start exchanging velocity component v
    Compute velocity component w
    Start exchanging velocity component w
    Complete Exchanges of u,v and w
    MPI_Win_post(group, 0, window)
    ! For the next iteration
Start exchange
    MPI_Win_start(group, 0, window)
    s2n(u1,north-mpirank, south-mpirank)
    ! recv from south, send to north
    n2s(u1, south-mpirank, north-mpirank)
    ! send to south, recv from north
    ... repeat for east-west and up-down
Complete exchange
    MPI_Win_wait(window)
    MPI_Win_complete(window)
    s2nfill(u1, window buffer, south-mpirank)
    n2sfill(u1, window buffer, north-mpirank)
    ... repeat for east-west and up-down

S2N
    Copy 2 planes of data from variable to sendbuffer
    !copy north boundary excluding ghost cells
    MPI_Put(sendbuffer, north-mpirank)

S2NFILL
    Copy 2 planes of data from window buffer to variable
    ! copy into south ghost cells
```

(Potluri, S., et al., ICS'10)

Communication Enhancement on CPUs - 2010

Sreeram Potluri of DK Panda Team

- 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*
  - MPI-1 non-blocking two-sided Communications
  - **MPI-2 one-sided Communications (on Ranger)**

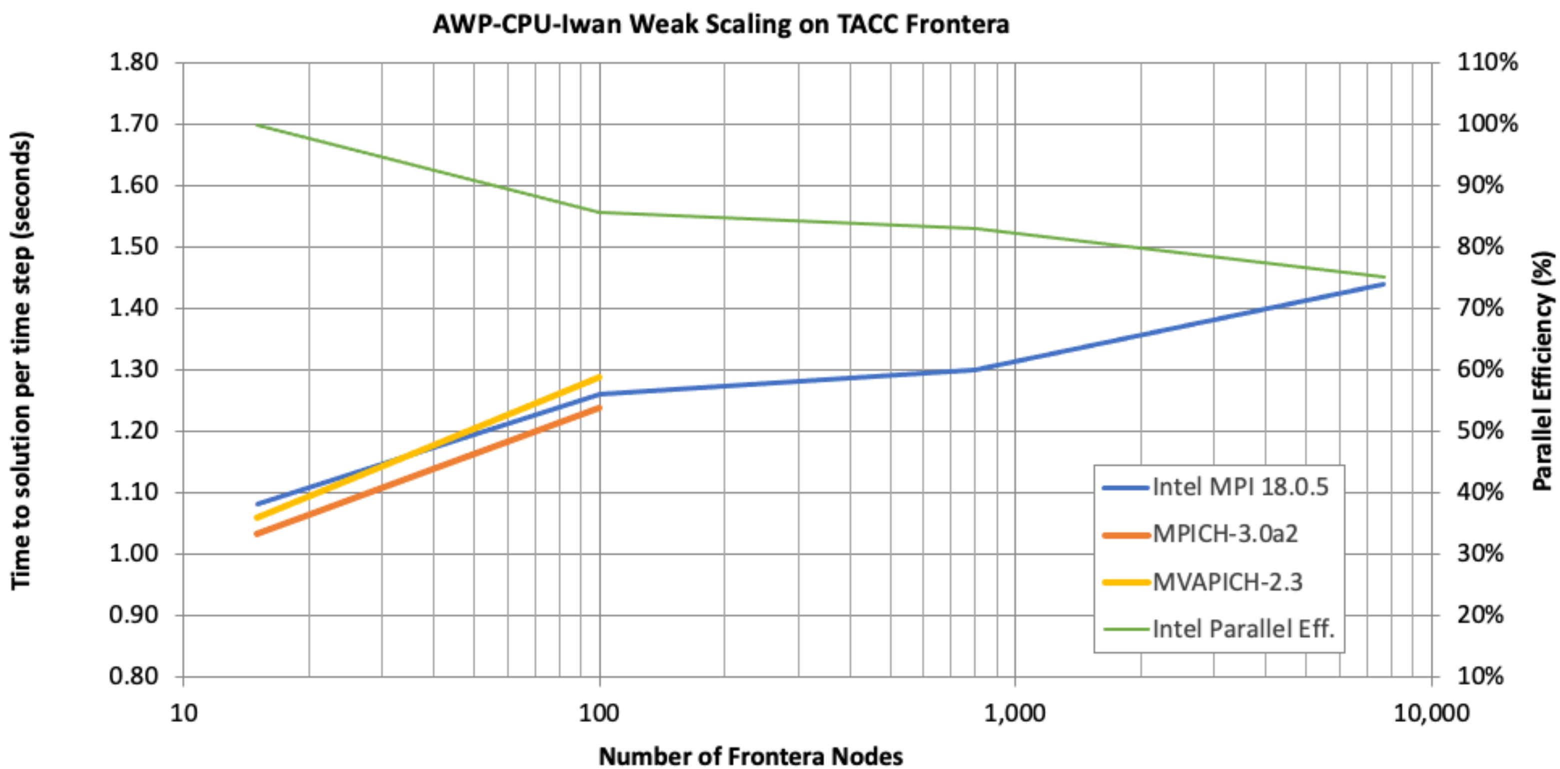


(Potluri, S., et al., ICS'10)

Iwan Code Performance on TACC Frontera

```
module load intel/18.0.5 mvapich2-x/2.3
export MV2_USE_MCAST=0
export MV2_USE_RDMA_CM_MCAST=0
export MV2_SMP_EAGERSIZE=28673
export MV2_SMP_NUM_SEND_BUFFER=8192
```

```
module load intel/18.0.5 mvapich2-x/3.oa2
export MV2_USE_MCAST=0
export MV2_USE_RDMA_CM_MCAST=0
export MV2_SMP_EAGERSIZE=28673
export MV2_SMP_NUM_SEND_BUFFER=8192
```



CUDA-aware Support Enhances AWP-ODC Performance

- **MVAPICH2 improves performance 20% over OpenMPI on Expanse, connected via NVLinks**
- **MVAPICH2 improves performance 20% over IMPI on Lonestar-6, connected via HDR**
- **CUDA-aware supported code gains additional 14% in MVAPCICH2/2.37-gdr over MVAPICH-2**

Expanse A100s	Teraflop/s	Time (sec/step)
gcc10.2.0+openmpi4.1.3 (2x2)	2.22	0.0294
nvhpc21.9 (openmpi4.1.1) (2x2)	2.21	0.0295
intel19.0.5+mvapich2/2.3.4 (2x2)	2.70	0.0243
intel19.0.5+mvapich2/2.3.7 (4x2)	3.55	0.0370
intel19.0.5+mvapich2/2.3.7-gdr (4x2)	4.03	0.0326

Lonestar 6 A100s	Teraflop/s	Time (sec/step)
gcc11.2.0+impi19.0.9 (2x3)	1.68	0.0585
gcc11.2.0+mvapich2/2.3.7 (2x3)	2.03	0.0488
gcc11.2.0+mvapich2/2.3.7 gdr (2x3)	2.30	0.0399
gcc11.2.0+mvapich2/latest gdr (2x3)	3.15	0.0311

On-the-fly Compression on GPUs – 2021

Qinghua Zhou of DK Panda team, IPDPS’21 Best Paper finalist

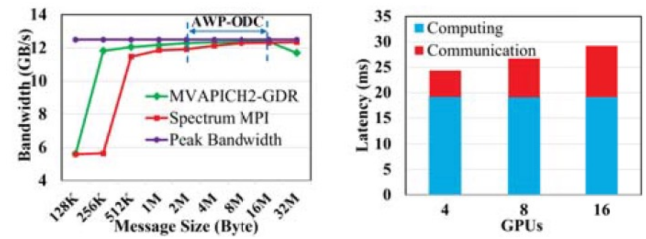
❖ **Motivation**

- **AWP-ODC has significant communication times on large-scale**
- **Disparity between intra-node and inter-node GPU communication bandwidths that precent efficient scaling**

❖ **Implementation**

- **Designed on-the-fly message compression schemes in MVAPICH2-GDR**
- **Accelerated point-to-point communication performance of transferring large GPU-to-GPU data**

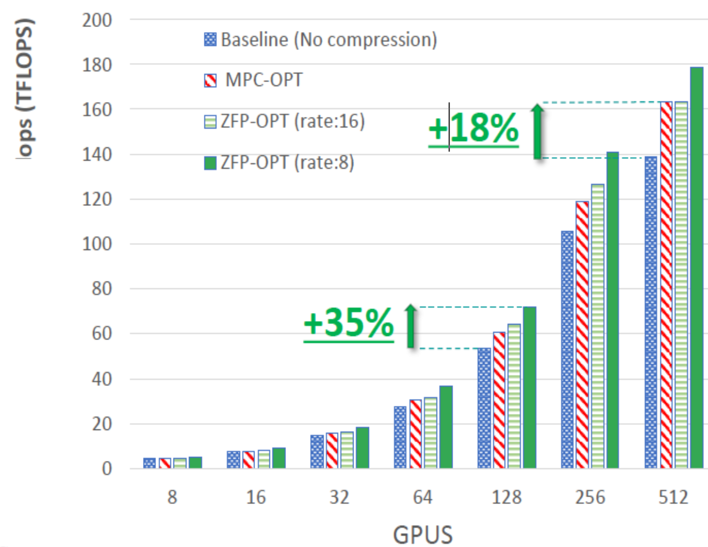
- **Compression algorithm for floating-point data, integrated to MVAPICH-GDR**
  - **MPC: Lossless, high throughput**
  - **ZFP: lossy, high throughput**
- **Weak scaling of AWP-ODC on V100 nodes with IB EDR**
  - **MPC-OPT achieved +18% flops, or -15% runtime**
  - **ZFP-OPT achieved +35% flops, or -26% runtime**



(a) Inter-node D-D Bandwidth (b) AWP-ODC time breakdown

Fig. 2. Motivating Example: production-quality and optimized CUDA-Aware MPI libraries can saturate IB EDR network while the communication time remains a significant bottleneck for HPC applications e.g. AWP-ODC. The message range for AWP-ODC is 2M to 16M as shown in Figure (a).

(Q. Zhou et al. IPDPS’21)



(Q. Zhou et al., IPDPS’21)

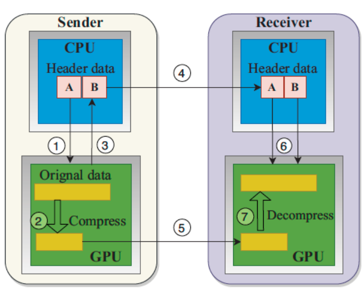


Fig. 4. Data flow of GPU communication with compression. There are seven steps: 1) Launch compression kernel with control parameters 2) Run compression kernel on GPU 3) Returned compressed size 4) Send header data with RTS packet 5) Send compressed GPU data 6) Launch decompression kernel with header data 7) Run decompression kernel to restore the data.

(Q. Zhou et al., IPDPS’21)

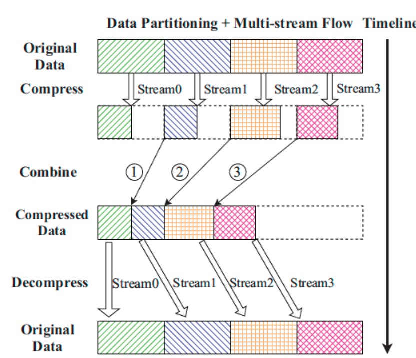
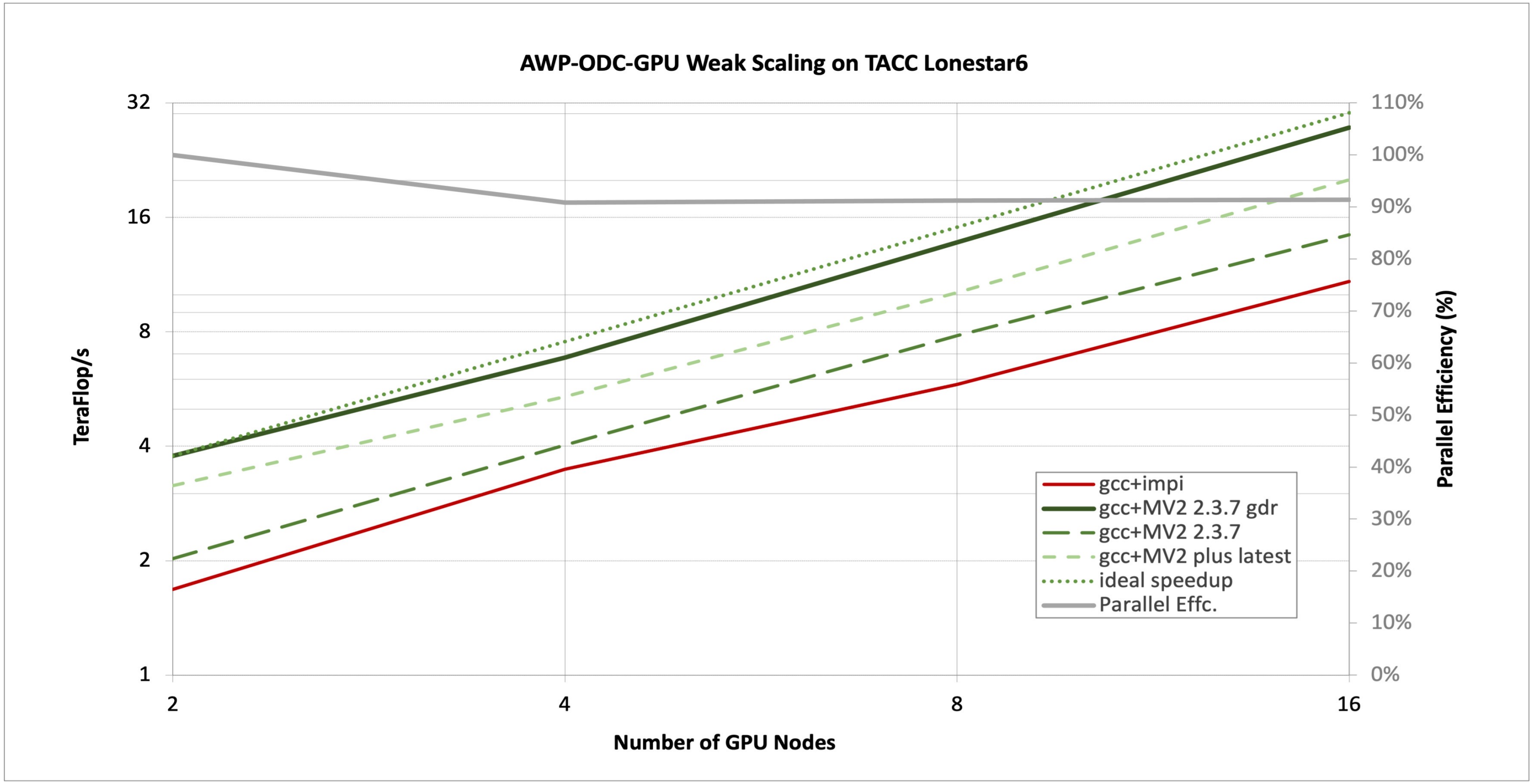


Fig. 7. Data partitioning and multi-stream flow for MPC.

(Q. Zhou et al., IPDPS’21)

Performance Evaluation on Lonestar-6



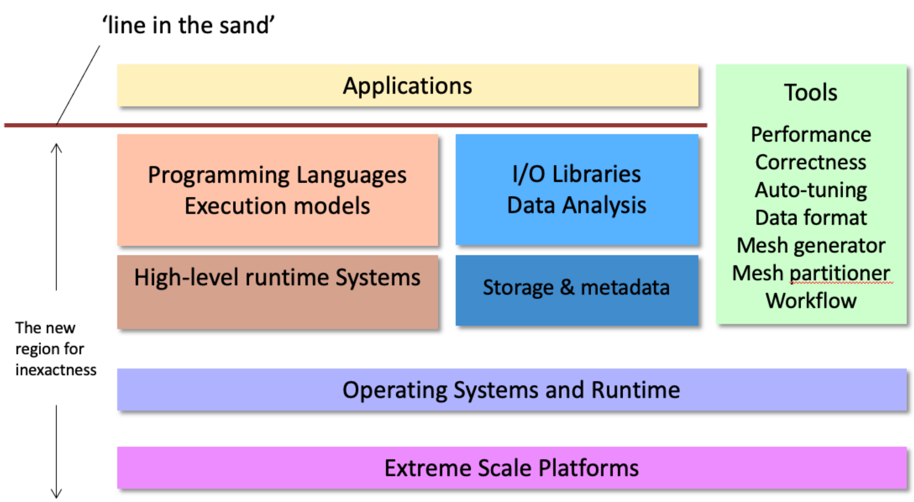
- **48%-64% benefits using on-the-fly MPC compression using MPC over GDR**
- **Combined MVAPICH2-GDR enhancement over IMPI, including both CUDA-aware support and on-the-fly compression, improves application performance by 125%, 97%, 137% and 154% on 2, 4, 8 and 16 nodes, respectively**

Lonestar6	mvapich2-2.3.7			mvapich2-2.3.7-gdr			mvapich2-2.3.7-gdr-compresson		
a100	gcc11.2.0			gcc11.2.0			gcc11.2.0		
nodes	Tflop/s	sec/step	parall eff.	Tflop/s	sec/step	parall eff.	Tflop/s	sec/step	parall eff.
2	2.0250	0.0488	100.0%	2.2960	0.0399	100.0%	3.7710	0.0261	100.0%
4	4.0270	0.0494	99.4%	4.5260	0.0436	98.6%	6.8510	0.0288	90.8%
8	7.8250	0.0510	96.6%	9.3250	0.0425	101.5%	13.7560	0.0288	91.2%
16	14.4130	0.1543	89.0%	17.1360	0.0460	93.3%	27.5580	0.0288	91.3%
	impi19.0.9			mvapich2-plus-3.0a2			mvapich2-plus-latest		
	gcc11.2.0			gcc11.2.0			gcc11.2.0		
	Tflop/s	sec/step	parall eff.	Tflop/s	sec/step	parall eff.	Tflop/s	sec/step	parall eff.
2	1.6800	0.0585	100.0%	2.391	0.0411	100.0%	3.151	0.0311	100.0%
4	3.4800	0.0572	103.6%	4.579	0.0431	95.8%	5.399	0.0366	85.7%
8	5.8170	0.0686	86.6%	7.796	0.0509	81.5%	10.136	0.0391	80.4%
16	10.8380	0.0737	80.6%	15.214	0.0523	79.5%	20.097	0.0395	79.7%

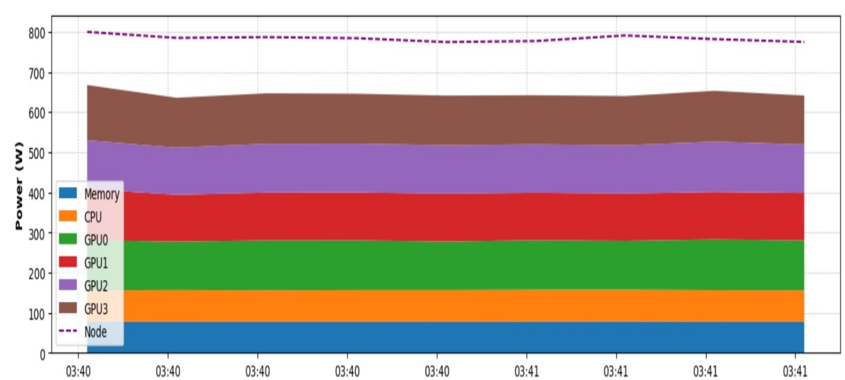
AWP-ODC software Engineering Challenges and Opportunities

Challenges for United and Continued Software Development

- **Inexact computing is required for reducing energy consumption**
- **Application level can tolerate a degree accuracy, e.g. discontinuous mesh, error tolerance and precision reduction**
- **AWP-ODC is highly efficient for regional earthquake simulation and physics-based seismic hazard analysis**



AWP-ODC power consumption study on Perlmutter (Govind, 2023)



Summary and Outlook

- AWP-ODC is accelerated with enhanced MVAPICH library on both CPU and GPU architectures
- We see 154% benefits over IMPI in MVAPICH2-GDR with CUDA-aware support and on-the-fly compression for AWP-ODC on 16 Lonestar6 A100 nodes, future plan is to apply these benefits to Iwan and CyberShake SGT codes
- The Iwan model introduces 20-30x more computation and 5-13x more memory consumption when compared to linear solution, a major challenge for software engineering
- A joint project with NOWLAB will address load-aware design for MPI asynchronous communication, application-aware neighborhood collective communication, and partitioned point-to-point primitives for efficient communication and cross runtime coordination for MPI+X
- Ongoing NSF CSA project is preparing AWP-ODC for NSF next generation LCCF *Horizon* to be deployed at TACC – with a hybrid approach using CPUs for dynamic rupture simulation, and GPUs for Iwan-DM wave propagation simulation
- 3D ground motion at 8 Hz or higher is required to realistically capture the full dynamics of a potential Big One on the San Andreas fault

Acknowledgments

HPGeoC Team


Daniel Roten


Akash Palla


Anish Govind

Collaborators


Philip Maechling


Scott Callaghan


Kim Olsen


Lars Koesterke

NOWLAB Team


DK Panda


Hari Subramoni


Sreeram Potluri


Qinghua Zhou

Computing Allocation OLCF DD, TACC LSCP and CSA, ACCESS Delta, SDSC Expanse, AMD AAC, DOE INCITE & ALCC

Funding NSF LCCF/CSA, NSF CSSI, NSF/USGS SCEC Core, SDSC Core

Lonestar-6 Network

```
[c309-001.ls6(1001)$ nvidia-smi topo -m
        GPU0    GPU1    GPU2    NIC0    CPU Affinity    NUMA Affinity
GPU0     X      SYS     SYS     NODE    0-63    0
GPU1    SYS      X      NODE    SYS     64-127  1
GPU2    SYS     NODE     X      SYS     64-127  1
NIC0    NODE    SYS     SYS      X

Legend:

  X    = Self
  SYS  = Connection traversing PCIe as well as the SMP interconnect between NUMA nodes (e.g., QPI/UPI)
  NODE = Connection traversing PCIe as well as the interconnect between PCIe Host Bridges within a NUMA node
  PHB  = Connection traversing PCIe as well as a PCIe Host Bridge (typically the CPU)
  PXB  = Connection traversing multiple PCIe bridges (without traversing the PCIe Host Bridge)
  PIX  = Connection traversing at most a single PCIe bridge
  NV#  = Connection traversing a bonded set of # NVLinks

NIC Legend:

  NIC0: mlx5_0
```

Expanse Network

```
[yfcui@exp-16-57 ~]$ nvidia-smi topo -m
        GPU0    GPU1    GPU2    GPU3    NIC0    NIC1    NIC2    CPU Affinity    NUMA Affinity
GPU0     X      NV12    SYS     SYS     NODE    NODE    SYS     0-15    0
GPU1    NV12     X      SYS     SYS     SYS     SYS     NODE    16-31   1
GPU2    SYS     SYS      X      NV12    SYS     SYS     SYS     48-63   3
GPU3    SYS     SYS     NV12     X      SYS     SYS     SYS     48-63   3
NIC0    NODE    SYS     SYS     SYS      X      PIX     SYS
NIC1    NODE    SYS     SYS     SYS     PIX      X      SYS
NIC2    SYS     NODE    SYS     SYS     SYS     SYS      X

Legend:

  X    = Self
  SYS  = Connection traversing PCIe as well as the SMP interconnect between NUMA nodes (e.g., QPI/UPI)
  NODE = Connection traversing PCIe as well as the interconnect between PCIe Host Bridges within a NUMA node
  PHB  = Connection traversing PCIe as well as a PCIe Host Bridge (typically the CPU)
  PXB  = Connection traversing multiple PCIe bridges (without traversing the PCIe Host Bridge)
  PIX  = Connection traversing at most a single PCIe bridge
  NV#  = Connection traversing a bonded set of # NVLinks

NIC Legend:

  NIC0: mlx5_0
  NIC1: mlx5_1
  NIC2: mlx5_2
```