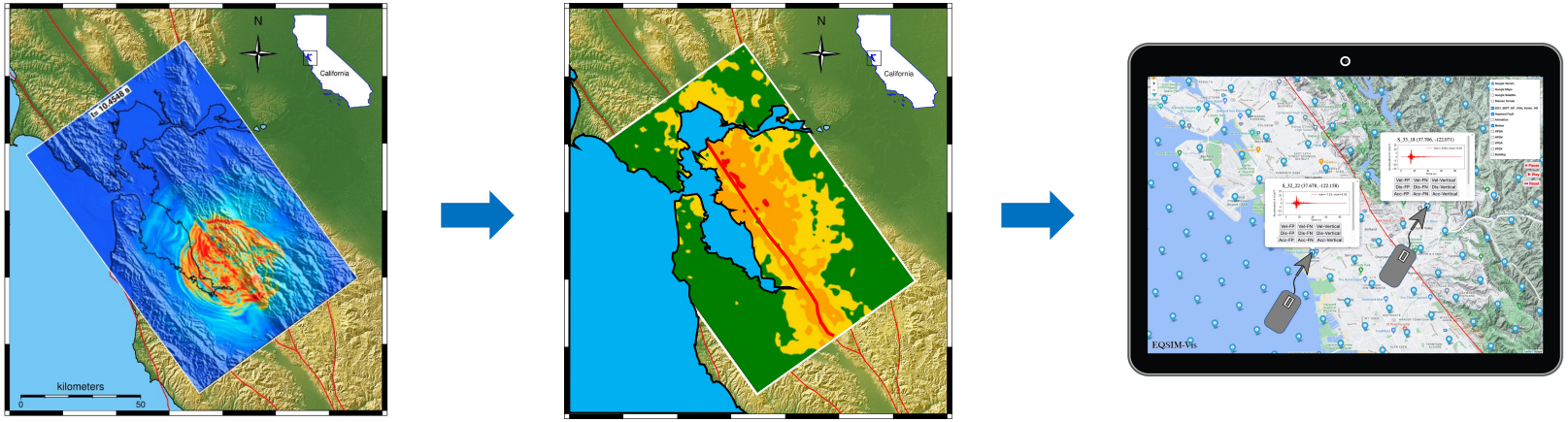


I/O, Storage, and Interrogation of Large Data

A PEER – LBNL workshop

January 18-19, 2024

Houjun Tang
Scientific Data Management Group
Lawrence Berkeley National Laboratory

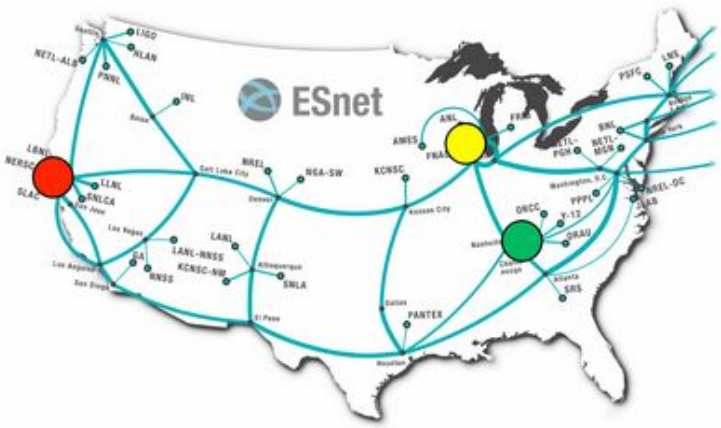
BERKELEY LAB

U.S. DEPARTMENT OF ENERGY | Office of Cybersecurity, Energy Security, and Emergency Response

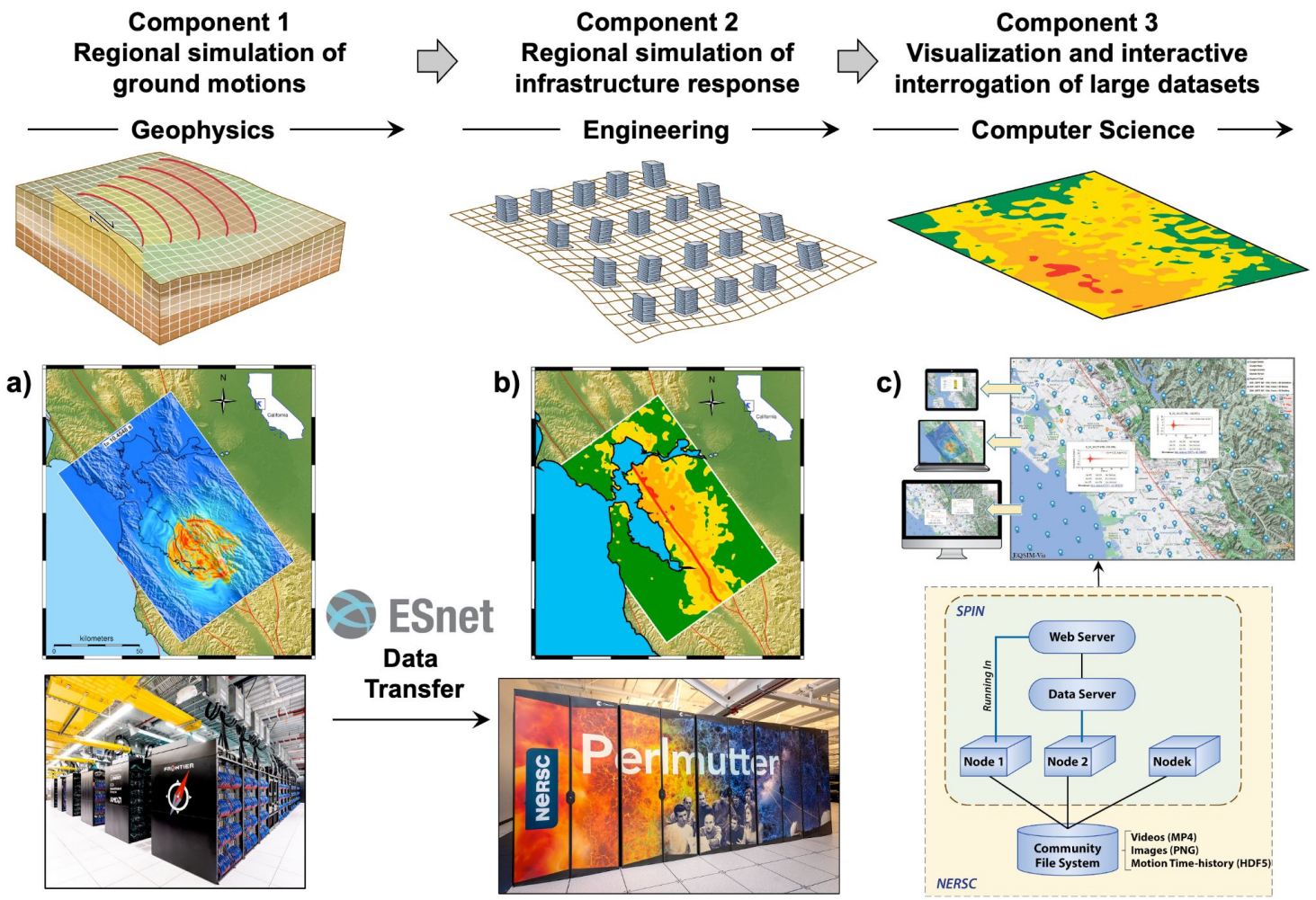
EXASCALE COMPUTING PROJECT

Challenges in Exascale Data Management

- New accelerator-based HPC architectures.
- Increased data volume.
- Effective data reduction.
- Sharing of both data and metadata across systems.
- Easy-to-use data search and access interfaces.



Data Management in the EQSIM Workflow



Managing EQSIM data with HDF5

- HDF5 (Hierarchical Data Format v5) is a data model, library, and file format for managing large and complex scientific data.
  - Supports heterogeneous data, easy sharing, cross platform, fast I/O, big data, and keep metadata with data.
  - Maintained for 25 years and widely adopted by the scientific community and the industries.

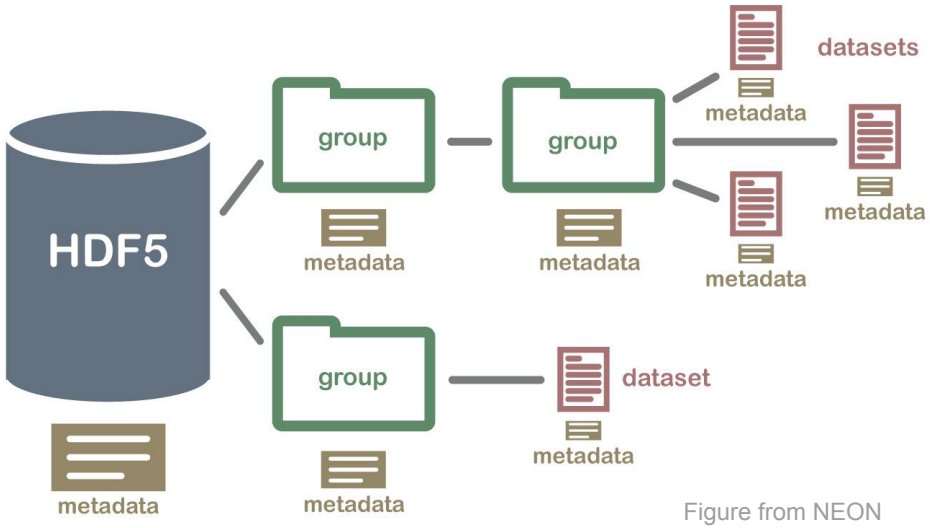
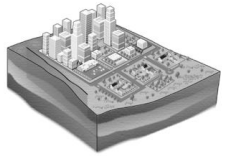
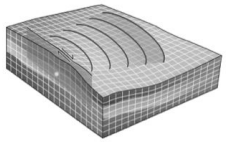


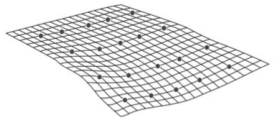
Figure from NEON

HDF5 Integration in the EQSIM Workflow

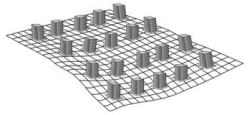


Ground motion simulation →



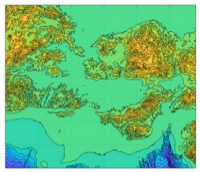


→ Infrastructure simulation →

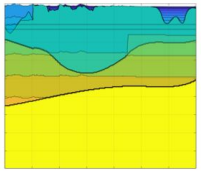


Pre-Run Time

Surface topography

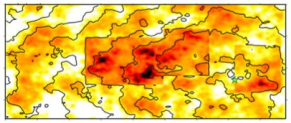


3D geology



HDF5
binary "S" file

Fault rupture scenario



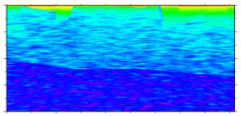
HDF5 file

Run time

Read HDF5 files

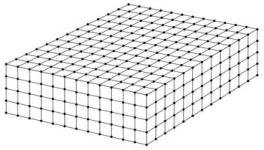
Read problem command data

Read stochastic geology parameters

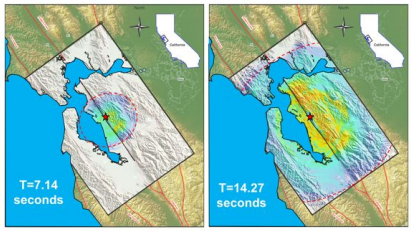


Overlay stochastic geology

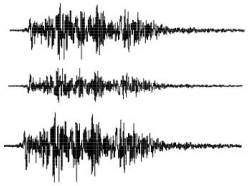
Auto - generate 200-300 billion grid point finite difference model



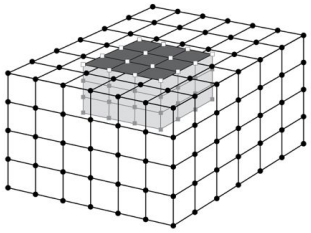
Time stepping (with checkpoint files)



Weakly coupled simulation: write HDF5 database of surface motions

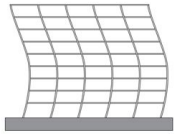


Strongly coupled simulation: write HDF5 database of specified volume motions

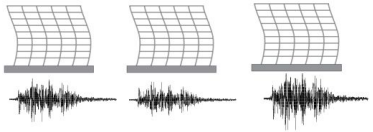


Infrastructure response

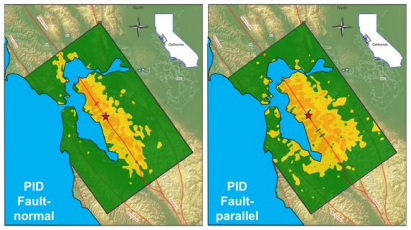
Select a specific building model from building library



Execute time stepping loop

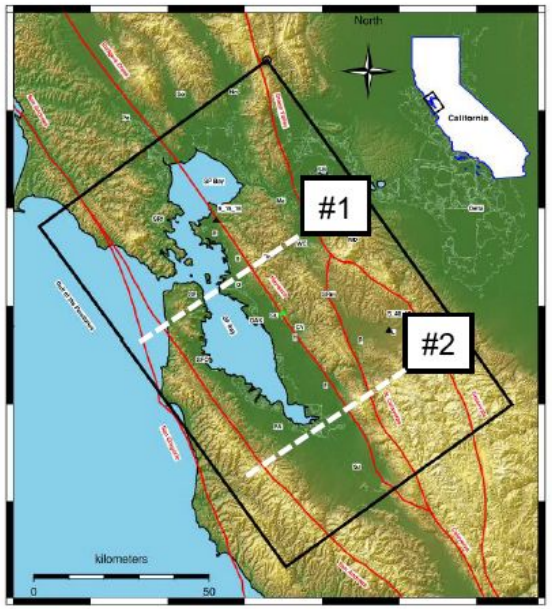


Distributed demand / risk data

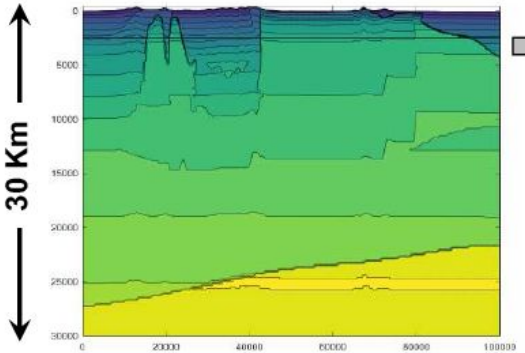


Sfile: a Multi-resolution Curvilinear Grid Format for Storing Velocity Models

USGS geologic data

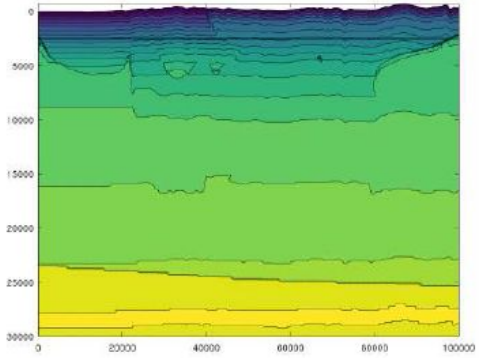


Earth Cross Section #1



100 Km

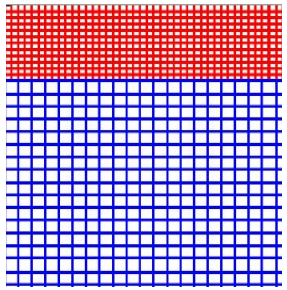
Earth Cross Section #2



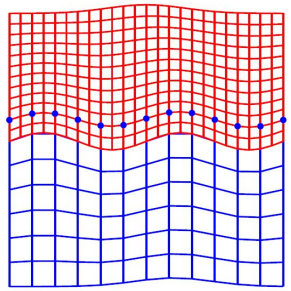
Vertical scale exaggerated

Newly developed "S" file for the 3D geologic model

- Enhanced material model inspection and visualization with the HDF5 format
- Enables material model output for both forward and inverse problems with SW4
- Allows converting existing material model data to an S file with SW4 grid and mesh refinement levels
- Allows horizontal and/or vertical down sampling to reduce the data size with acceptable interpolation error bounds



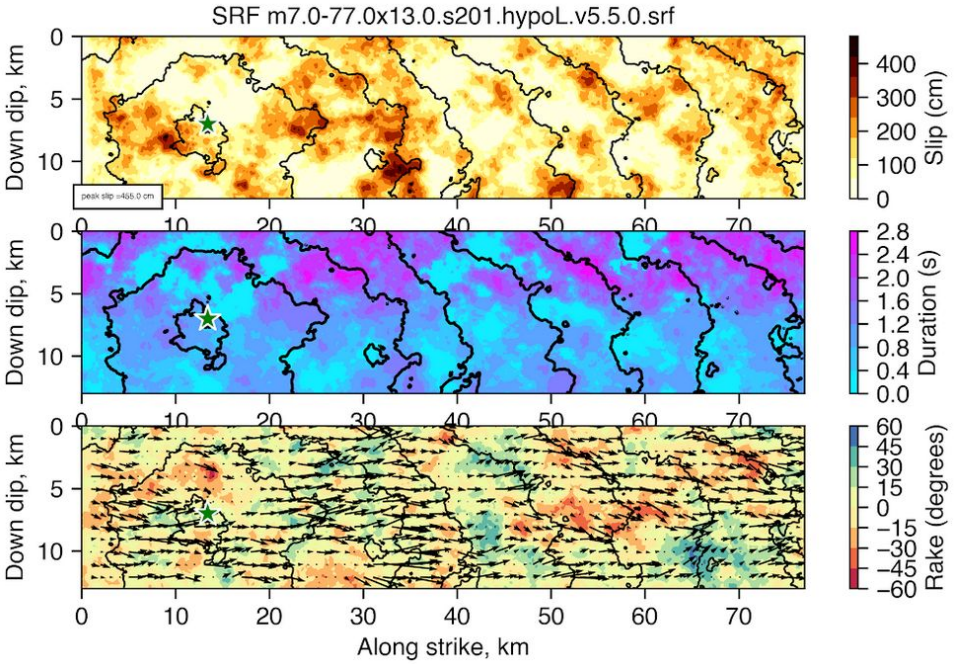
R file



S file
(HDF5)

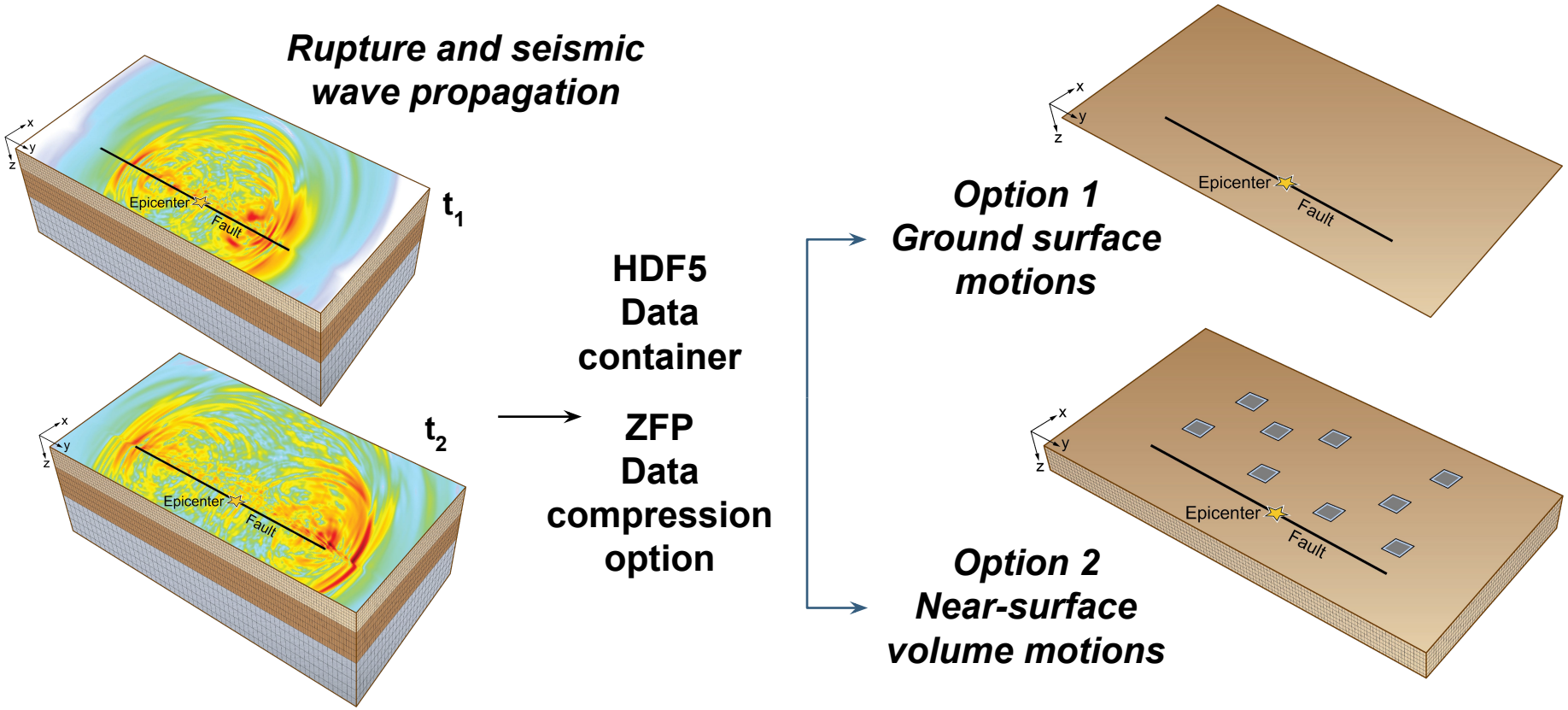
SRF-HDF5: storing text-based SRF data in HDF5

- Originally in SRF format (ASCII) that is not designed for parallel processing.
- Converted HDF5 file is ~1/3 the original size and can be read more efficiently in parallel (>5x speedup).



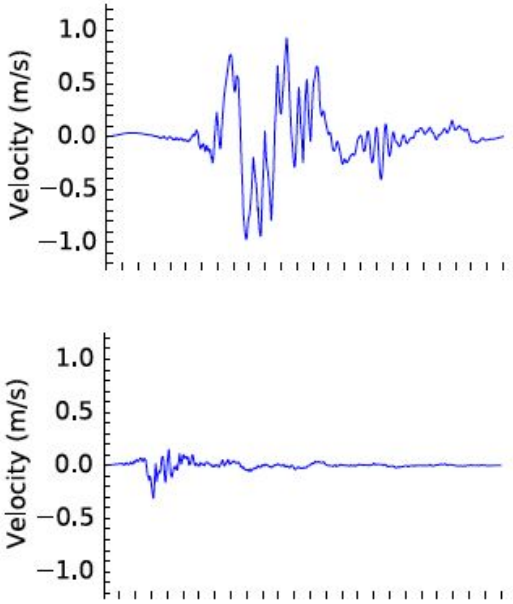
Handling Large Simulation Ground Motion Data

- Spatially dense grid of ground motions from high fidelity simulations
- Must accommodate multiple rupture realizations for each earthquake scenario
- Must include a down-sampling capability from a baseline dataset
- The database design must be created with ***future scalability*** in mind



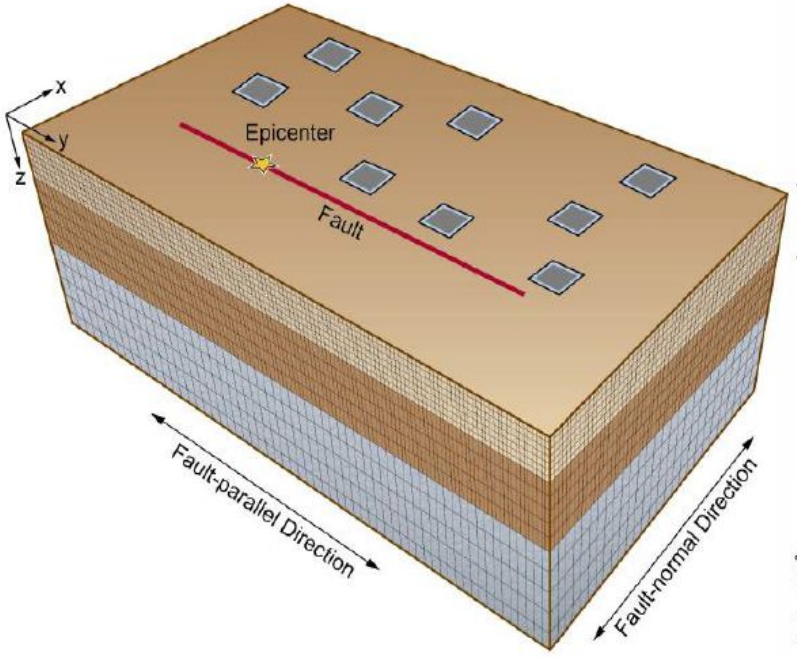
Output Ground Motions at User-Defined Locations

- **USGS format**
  - 1 text file per location, large number of files.
  - Easy to read.
- **SAC format**
  - 3 files per location, large number of files.
  - Required special reader to parse data.
- **SAC-HDF5 format**
  - Single HDF5 file for all locations.
  - Easy to read.
  - Write time is up to **5-9X** faster than USGS and SAC on SSDs.
  - SFBA simulations generate ~2300 locations, 2.2GB file.



SSIoutput: Motions of Near-surface Volume

- HDF5 format.
- Motions in the x, y, z directions.
- 4D datasets (Time + 3D Volume)
- Allows saving motions of the entire near-surface domain.
- Supports downsample in time.
- Easy to access and visualize.
- SFBA simulations generate surface motions of 260-300GB with downsampling every 16 steps (0.012s -> 0.19s), more with volume output.



HDF5 output with compression enables saving velocity time-history at *every grid point* in a near-surface volume (e.g. to 150m depth)

Error-bounded Lossy Compression

- ZFP is a library for compressing floating-point data with ***error-bounded lossy compression***.
- ZFP can be enabled with HDF5 to read and write compressed data transparently.

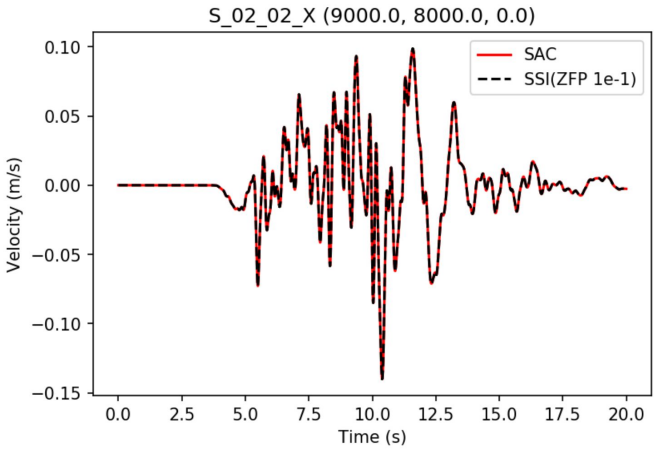
Config	CR	File Size
None	1	**76 TB**
accuracy= 1e-2	261	**293 GB**

```
# pip install hdf5plugin
import h5py
import hdf5plugin

h5file = h5py.File('data.h5')

data = h5file['vel_0'][:]

h5file.close()
```



https://computing.llnl.gov/projects/zfp

I/O Time Comparison on Cori

38 TB

# Node	# Rank	buffer	chk _x	chk _y	chk _z	I/O Time (s)	Exec. Time (s)	I/O %	Data Size (GB)
1024	8192	100	60	60	32	3,458	6,589	52.48%	155
		100	32	60	32	2,005	4,861	41.25%	164
		100	32	32	32	3,927	6,664	58.93%	176
		200	60	60	32	1,409	4,861	28.99%	155
		200	32	60	32	979	3,838	25.50%	164
		200	32	32	32	2,009	5,142	39.07%	176
		400	60	60	32	841	3,759	22.38%	155
		400	32	60	32	485	4,996	9.72%	164
		400	32	32	32	1,075	6,005	17.90%	176
		800	60	60	32	433	3,568	12.14%	155
		800	32	60	32	284	3,147	9.02%	164
		800	32	32	32	625	3,708	16.84%	176

CR=251

CR=237

CR=221

1.5 billion grid points, top grid size 2001x4001, 5m grids, 14179 simulation steps, with ZFP acc=1e-1

I/O Time on Frontier

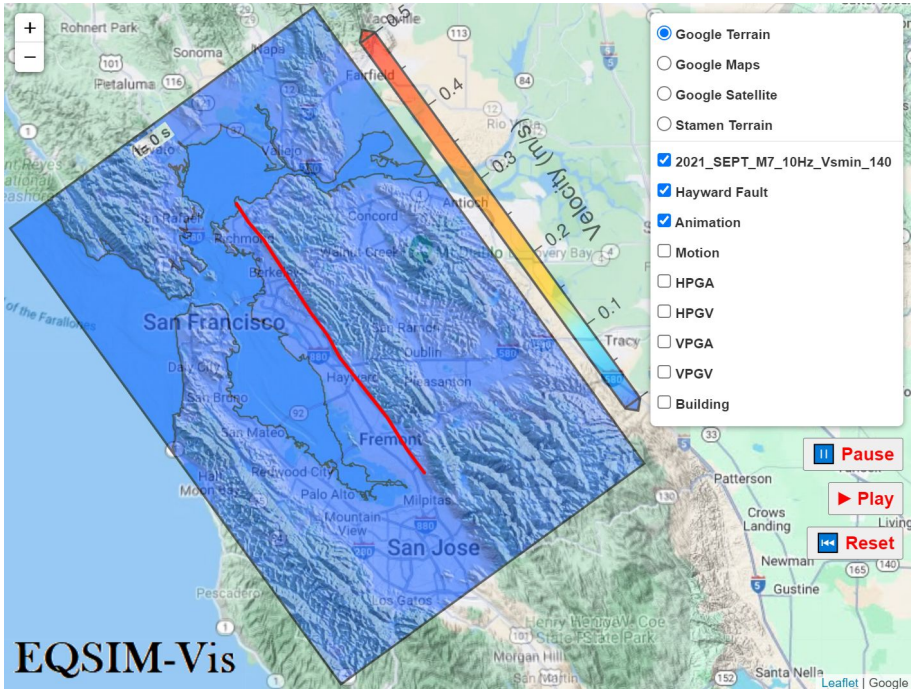
- 10Hz, Vsmin 140m/s, M7 Hayward Fault simulation.
- 435 billion total grid points, 202460 simulation steps (90 seconds).
- Surface motion output
  - 68577 x 45729, ~2.7 billion grid points with 1.75m grid size.
  - ZFP accuracy mode, 1e-2
  - Downsample factor of 10 (timesteps)
  - Written to the Orion Lustre parallel file system, utilizing 1024 OSTs.

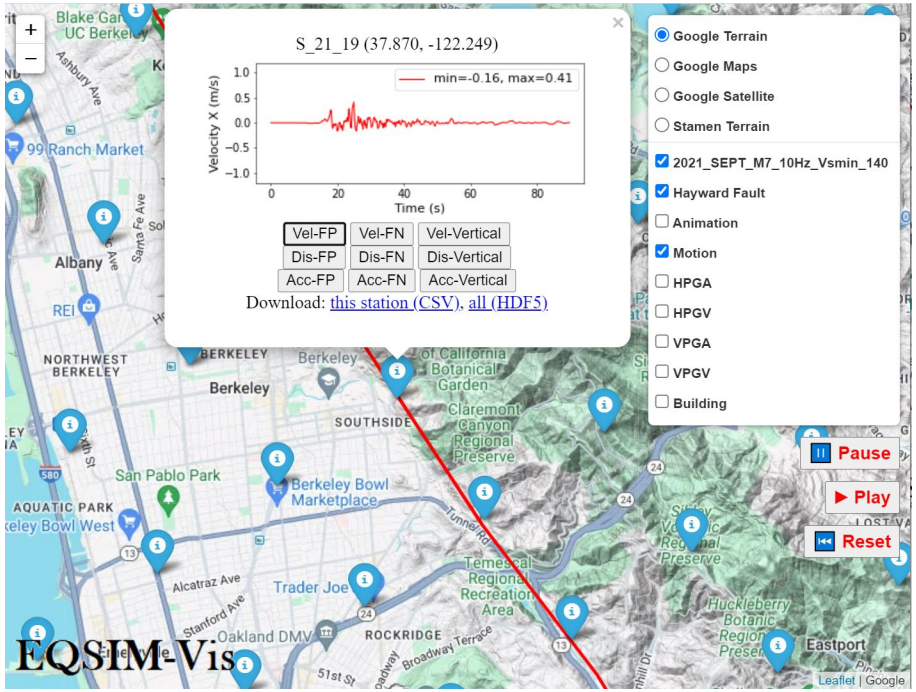
1385TB

# Node	# GPU	buffer	chk_x	chk_y	chk_z	I/O Time (h)	Exec. Time (h)	I/O %	Data Size (TB)
2048	16384	200	268	356	1	0.8	28	3%	11

CR=126

EQSIM-Vis: Inspect Ground Motions on an Interactive Map





Demo

Summary

- Custom formats are not ideal when data needs to be shared to many people with different backgrounds.
- HDF5 format and library is useful and effective for managing large data.
  - Cross-platform, multi-language support (C, Python, MATLAB).
  - Self-describing, stores metadata together with data.
  - Efficiently parallel I/O.
- Effective error-bounded compression can significantly reduce the total data size, allows saving high-resolution data with a large domain size.
- Visualization tools such as EQSIM-vis enables efficient data inspection.

Thanks!

email: htang4@lbl.gov

https://crd.lbl.gov/tang