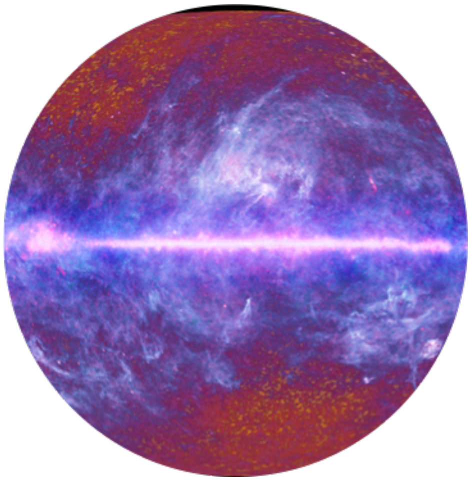
Scientific Workflows at DataWarp-Speed: Accelerated Data-Intensive Science using NERSC's Burst Buffer

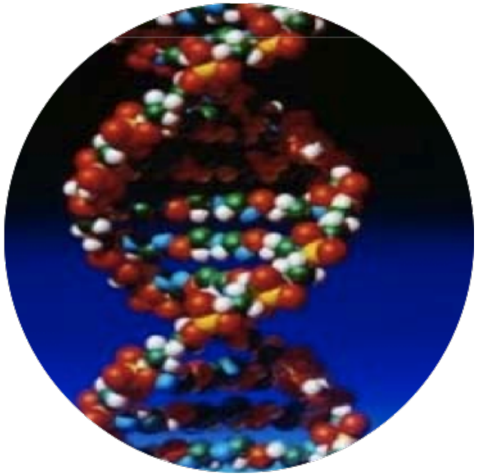
Andrey Ovsyannikov[1], Melissa Romanus[2], Brian Van Straalen[1], David Trebotich[1], Gunther Weber[1,3]

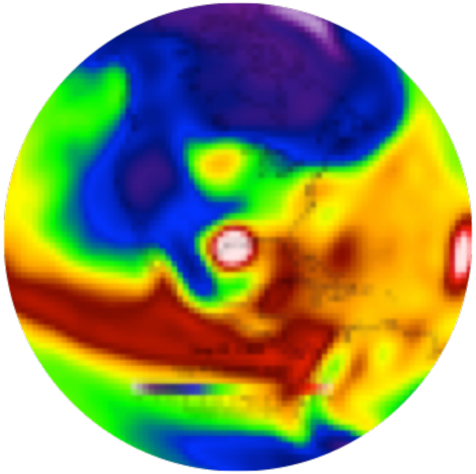
[1] Lawrence Berkeley National Laboratory
[2] Rutgers University
[3] University of California, Davis

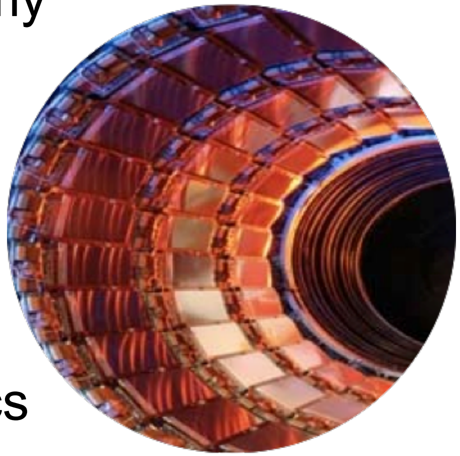
PDSW-DISCS 2016: 1st Joint International Workshop on Parallel Data Storage and Data Intensive Scalable Computing Systems
November 14, 2016, Salt Lake City, UT

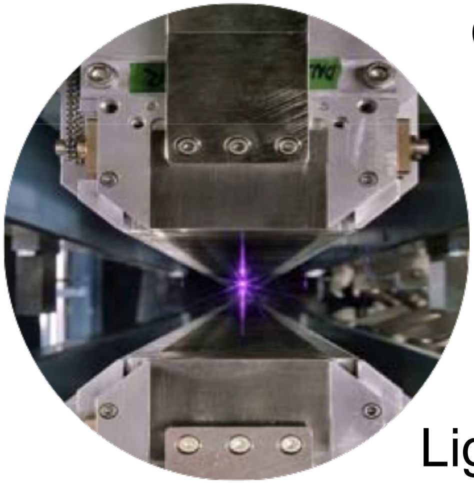
Data-intensive science


Astronomy


Genomics


Climate

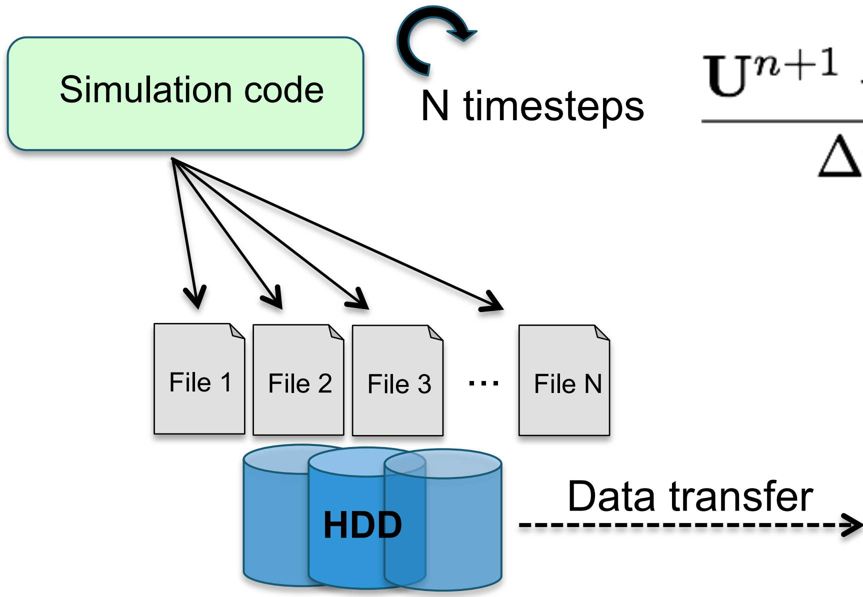

Physics


Light Sources

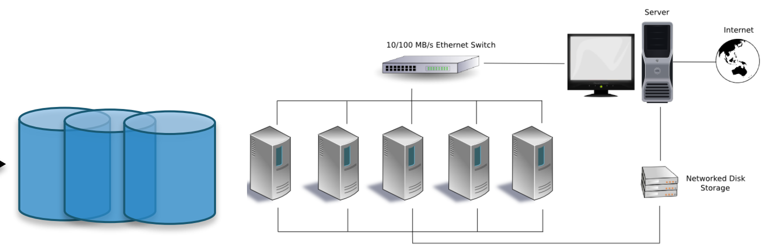
What do we mean by data-intensive applications?

- Applications analyzing data from experimental or observational facilities (telescopes, accelerators, etc.)
- Applications combining modeling/simulation with experimental/observational data
- Applications with complex workflows that require large amounts of data movement
- Applications using analytics in new ways to gain insights into scientific domains

Computational physics and traditional post-processing



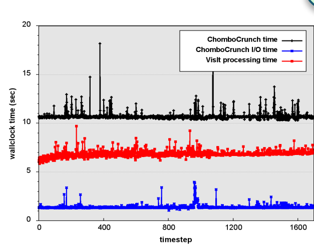
$$\frac{\mathbf{U}^{n+1} - \mathbf{U}^{n}}{\Delta t} = \mathbf{F}(\mathbf{U}^{n}, \mathbf{x}, t^{n})$$

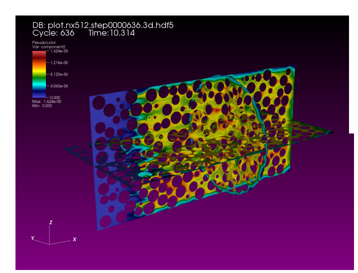


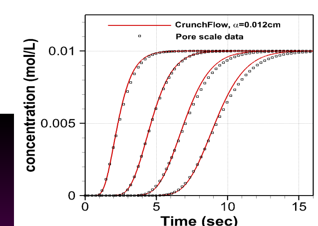
Remote storage: e.g. Globus Online, visualization cluster,...

Data analysis/ Visualization

Data transfer/storage and traditional post-processing is extremely expensive!



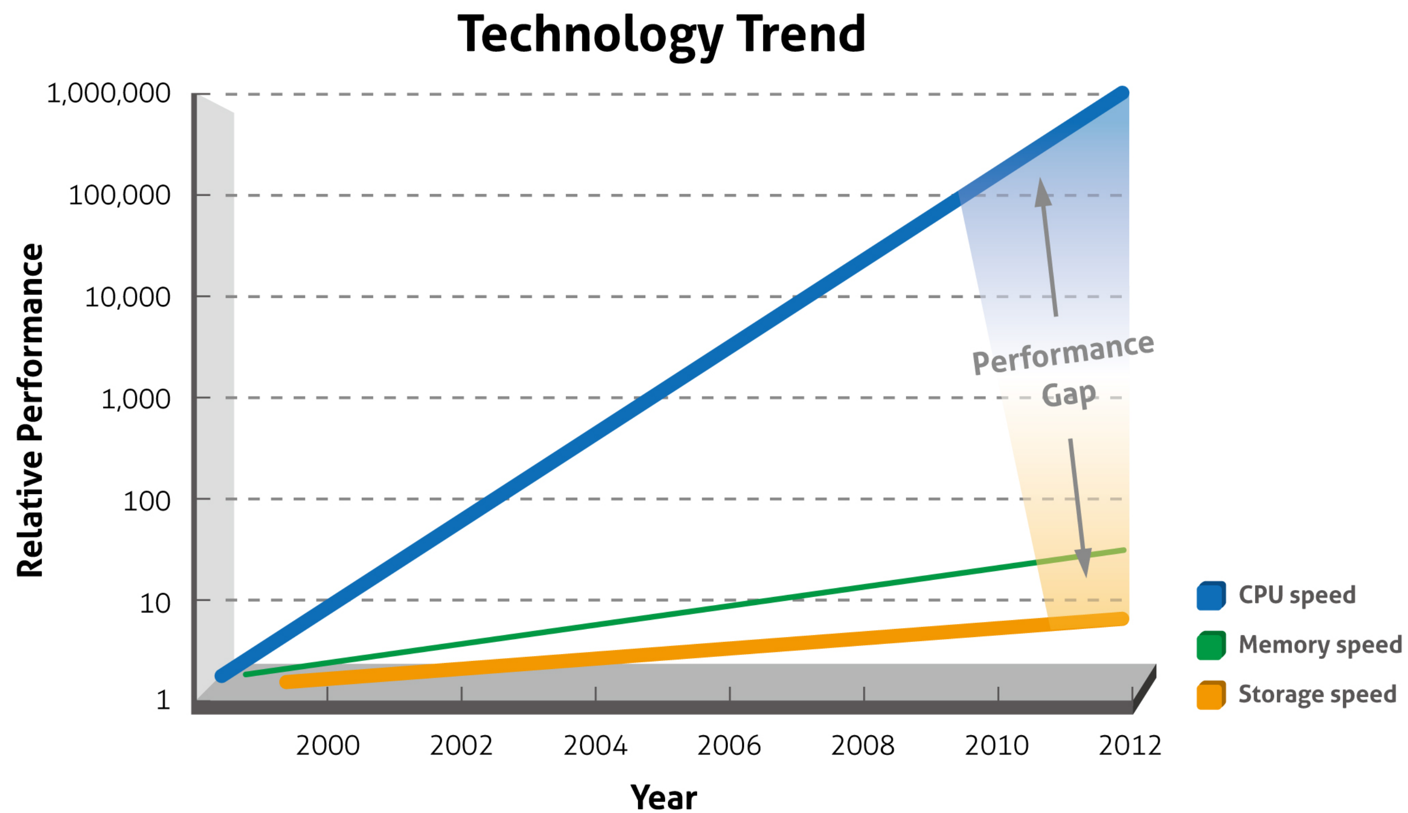




U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

Bandwidth gap

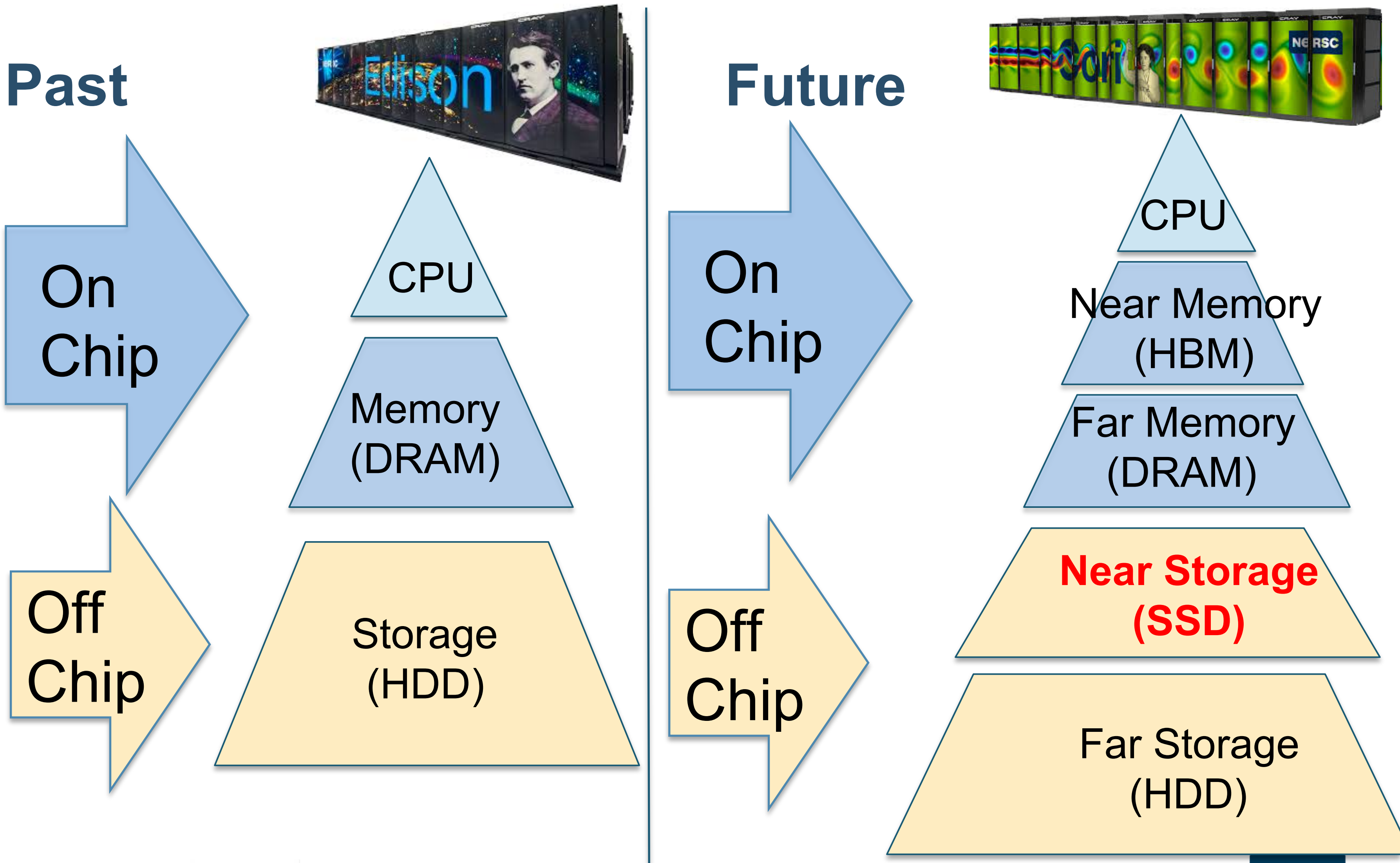


Growing gap between computation and I/O rates.
Insufficient bandwidth of persistent storage media.

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

HPC memory hierarchy

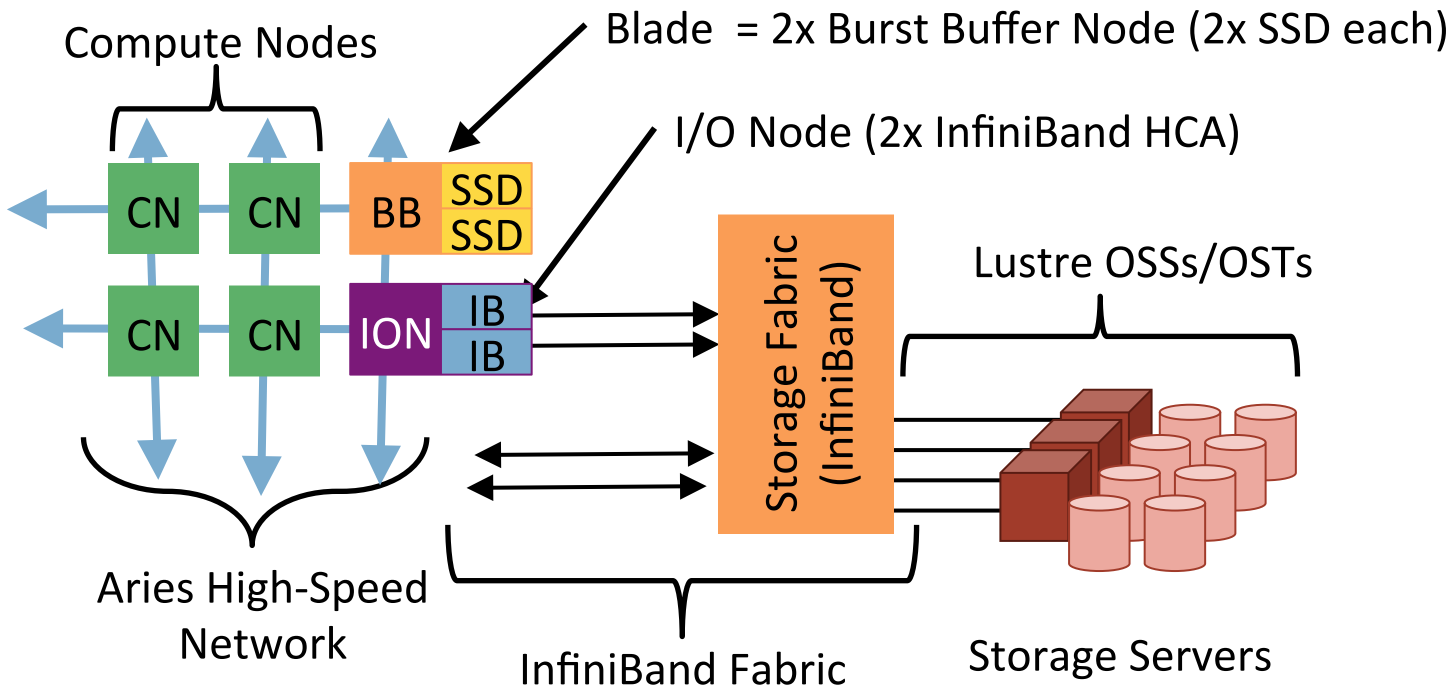


Data processing methods

Data processing execution methods (Prabhat & Koziol, 2015)

	Post-processing	In-situ	In-transit
Analysis Execution Location	Separate Application	Within Simulation	Burst Buffer
Data Location	On Parallel File System	Within Simulation Memory Space	Within Burst Buffer Flash Memory
Data Reduction Possible?	NO: All data saved to disc for future use	YES: Can limit output to only analysis products	YES: Can limit data saved to disk to only analysis products.
Interactivity	YES: User has full control on what to load and when to load data from disk	NO: Analysis actions must be pre-scribed to run within simulation	LIMITED: Data is not permanently resident in flash and can be removed to disk
Analysis Routines Expected	All possible analysis and visualization routines	Fast running analysis operations, statistical routines, image rendering	Longer running analysis operations bounded by the time until drain to file system. Statistics over simulation time

NERSC/Cray Burst Buffer Architecture



- Cori Phase 1 configuration: 920TB on 144 BB nodes (288 x 3.2 GB SSDs) 288 BB nodes on Cori Phase 2.
- DataWarp software (integrated with SLURM WLM) allocates portions of available storage to users per-job
- Users see a POSIX filesystem
- Filesystem can be striped across multiple BB nodes (depending on allocation size requested)

Burst Buffer User Cases @ NERSC

Burst Buffer User Cases	Example Early Users
IO Bandwidth: Reads/ Writes	• **Nyx/BoxLib** • VPIC IO
Data-intensive Experimental Science - "Challenging/ Complex" IO pattern, eg. high IOPs	• ATLAS experiment • TomoPy for ALS and APS
Workflow coupling and visualization: in transit / in-situ analysis	• **Chombo-Crunch / VisIt carbon sequestration simulation**
Staging experimental data	• ATLAS and ALS SPOT Suite

Many others projects not described here (~50 active users).

Benchmark performance

- **Burst Buffer is doing well against benchmark performance targets**
  - Out-performs Lustre (in tests using half the full Burst Buffer and only a fraction of the full Cori compute load)

Details on use cases and benchmark performance in Bhimji et al, CUG 2016

	IOR Posix FPP		IOR MPIO Shared File		IOPS	
	Read	**Write**	**Read**	**Write**	**Read**	**Write**
Best Measured (140 Burst Buffer Nodes : 1120 Compute Nodes; 4 ranks/node)*	905 GB/s	873 GB/s	803 GB/s	351GB/s (since improved)	12.6 M	12.5 M
Lustre (peak – 24 OSTs: 930 compute nodes, 4 ranks/node; 4 MB transfer)	708 GB/s	751 GB/s	573 GB/s	223 GB/s	-	-

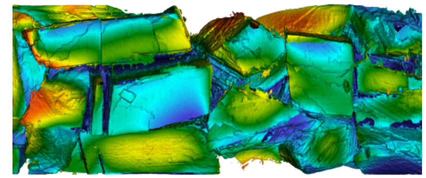
**Bandwidth tests: 8 GB block-size 1MB transfers IOPS tests: 1M blocks 4k transfer*

Chombo-Crunch (ECP application)

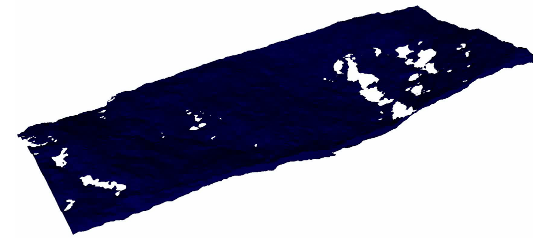
- **Simulates pore scale reactive transport processes associated with carbon sequestration**
- **Applied to other subsurface science areas:**
  - –Hydrofracturing (aka "fracking")
  - –Used fuel disposition (Hanford salt repository modeling)
- **Extended to engineering applications**
  - –Lithium ion battery electrodes
  - –Paper manufacturing (hpc4mfg)

The common feature is ability to perform direct numerical simulation from image data of arbitrary heterogeneous, porous materials

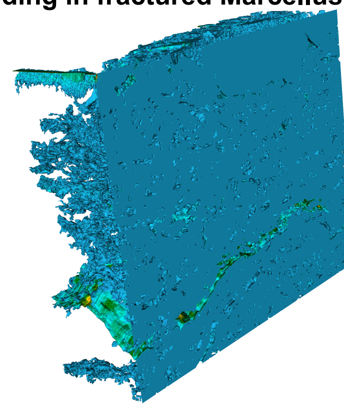
pH on crushed calcite in capillary tube



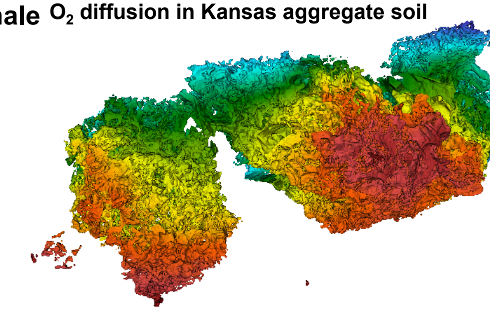
Transport in fractured dolomite



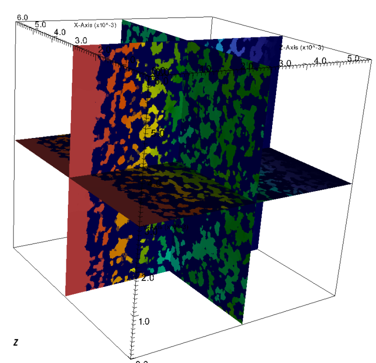
Flooding in fractured Marcellus shale



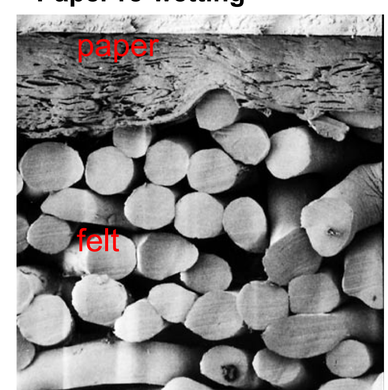
O_2 diffusion in Kansas aggregate soil



Electric potential in Li-ion electrode



Paper re-wetting

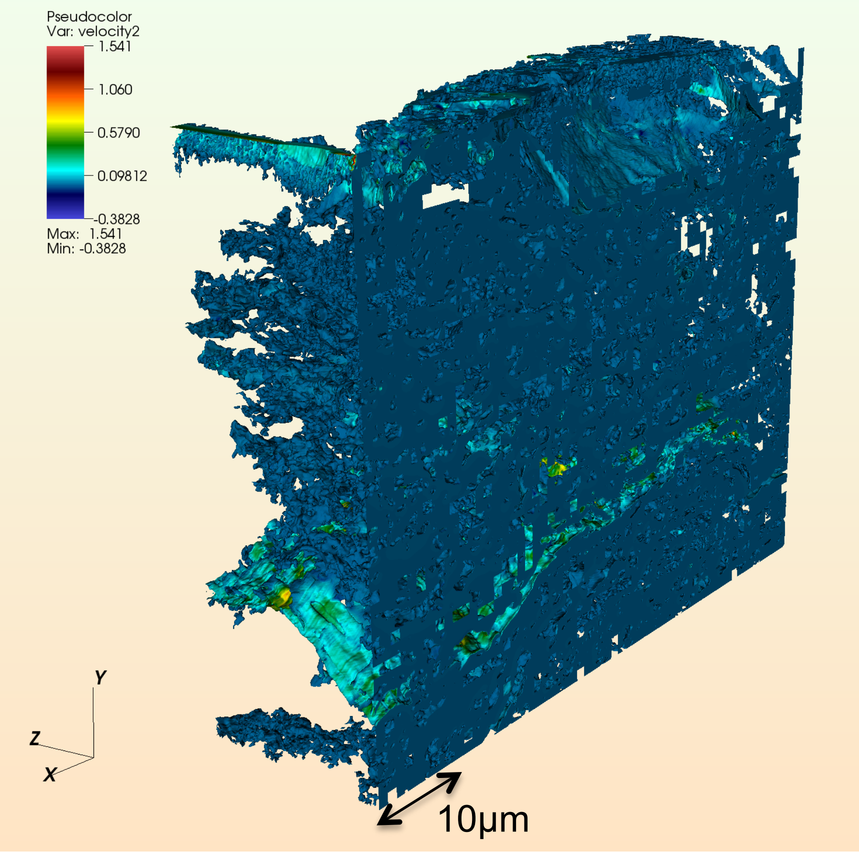


Data-intensive simulation at scale

Example: Reactive flow in a shale

- Required computational resources: **41K cores**
- Space discretization: **2 billion cells**
- Time discretization: **~1μs; in total $3*10^4$ timesteps**
- Size of 1 plotfile: **0.3TB**
- Total amount of data: **9PB***
- I/O: **61%** of total run time
- Time to transfer data:
  - to GlobusOnline storage: >**1000 days**
  - to NERSC HPSS: **120 days**

Sample of California's Monterey shale

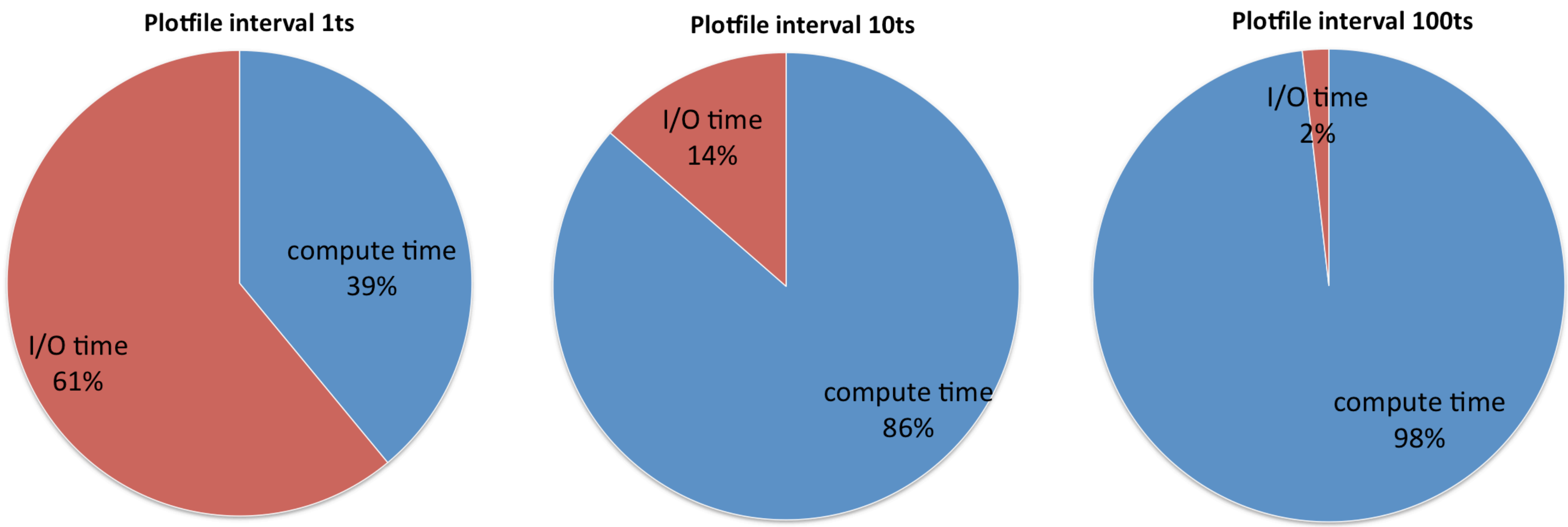


Complex workflow:
On-the-fly visualization/quantitative analysis
On-the-fly coupling of pore-scale simulation with reservoir scale model

I/O constraint: common practice

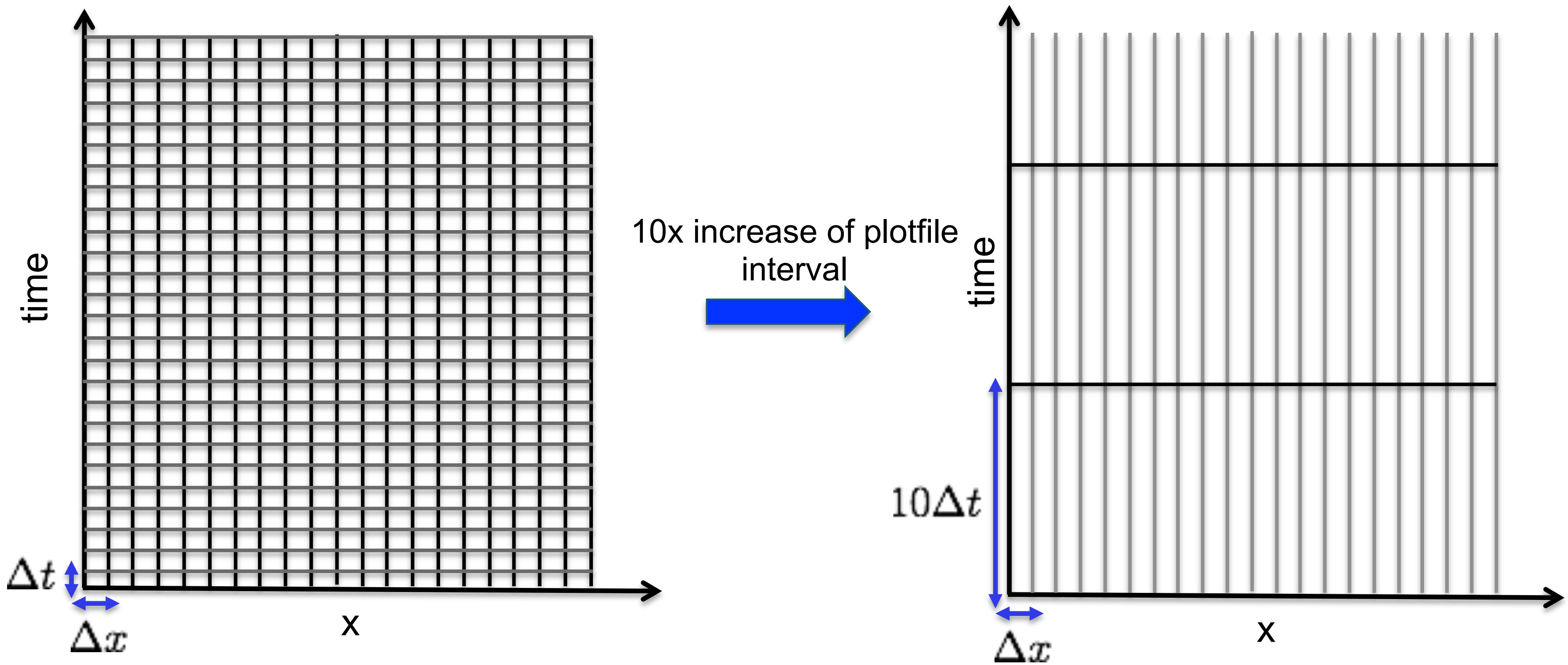
Common practice: increase I/O (plotfile) interval by 10x, 100x, 1000x,...

I/O contribution to Chombo-Crunch wall time at different plotfile intervals



Loss of temporal/statistics accuracy

Time evolution from 0 to T: $\frac{d\mathbf{U}}{dt} = \mathbf{F}(\mathbf{U}(x,t))$



Pros: less data to move and store
Cons: degraded accuracy of statistics (stochastic simul.)

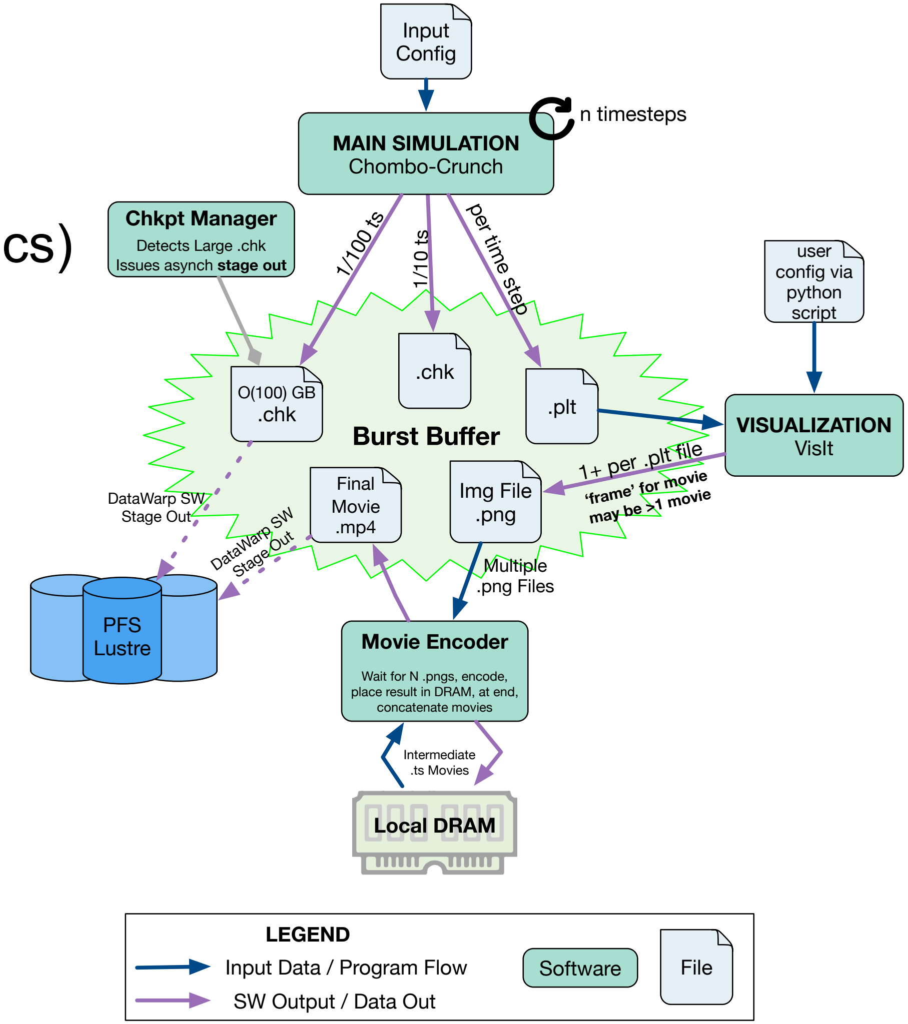
$$\varepsilon \sim \frac{1}{\sqrt{N}}, \ N \text{ is the sample size}$$

Proposed in-transit workflow

Workflow components:

- ❑ **Chombo-Crunch**
- ❑ **VisIt** (visualization and analytics)
- ❑ **Encoder**
- ❑ **Checkpoint manager**

I/O: HDF5 for checkpoints and plotfiles



Straightforward batch script

allocate BB capacity

copy restart file to BB

run each component

transfer output product to persistent storage

```
#!/bin/bash
#SBATCH --nodes=1291
#SBATCH --job-name=shale
#DW jobdw capacity=200TB access_mode=striped type=scratch
#DW stage_in type=file source=/pfs/restart.hdf5 destination
     =$DW_JOB_STRIPED/restart.hdf5
### Load required modules
module load visit
ScratchDir="$SLURM_SUBMIT_DIR/_output.$SLURM_JOBID"
BurstBufferDir="${DW_JOB_STRIPED}"
mkdir $ScratchDir
stripe_large $ScratchDir
NumTimeSteps=2000
EncoderInt=200
RestartFileName="restart.hdf5"
ProgName="chombocrunch3d.Linux.64.CC.ftn.OPTHIGH.MPI.PETSC.
ex"
ProgArgs=chombocrunch.inputs
ProgArgs="$ProgArgs check_file=${BurstBufferDir}check
     plot_file=${BurstBufferDir}plot pfs_path_to_checkpoint=
     ${ScratchDir}/check restart_file=${BurstBufferDir}${
     RestartFileName} max_step=$NumTimeSteps"
### Launch Chombo-Crunch
srun -N 1275 –n 40791 $ProgName $ProgArgs > log 2>&1 &
### Launch VisIt
visit -l srun -nn 16 -np 512 -cli -nowin -s VisIt.py &
### Launch Encoder
./encoder.sh -pngpath $BurstBufferDir -endts $NumTimeSteps
     -i $EncoderInt &
wait
### Stage-out movie file from Burst Buffer
#DW stage_out type=file source=$DW_JOB_STRIPED/movie.mp4
     destination=/pfs/movie.mp4
```

DataWarp API

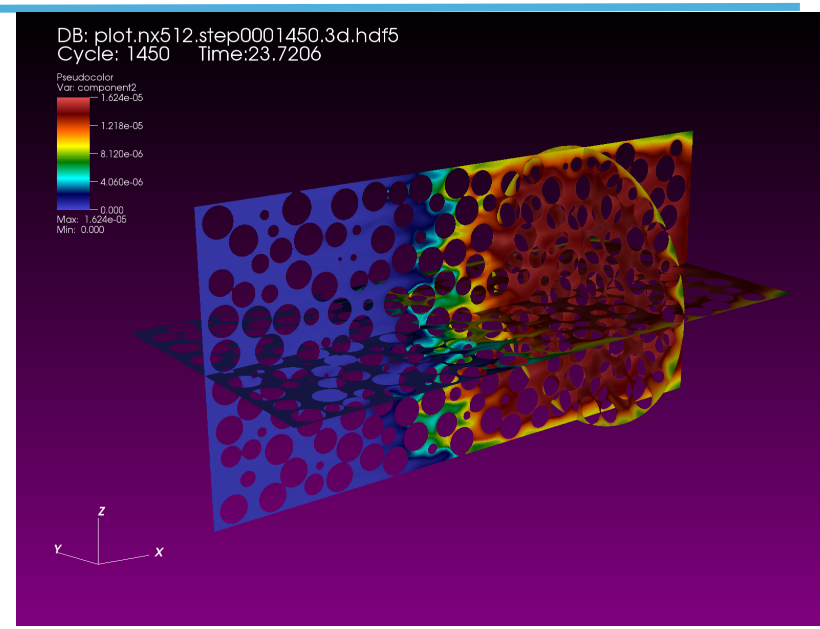
Asynchronous transfer of plot file/checkpoint from Burst Buffer to PFS

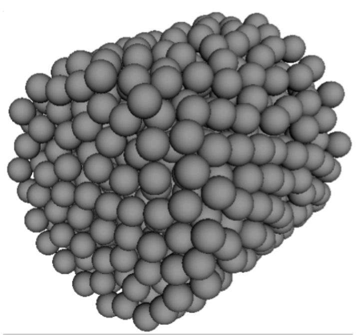
```
#ifdef CH_DATAWARP
// use DataWarp API stage_out call to move plotfile from BB to Lustre
   char lustre_file_path[200];
   char bb_file_path[200];
   if ((m_curStep%m_copyPlotFromBurstBufferInterval == 0) &&
(m_copyPlotFromBurstBufferInterval > 0))
   {
      sprintf(lustre_file_path, "%s.nx%d.step%07d.%dd.hdf5", m_LustrePlotFile.c_str(),
ncells, m_curStep, SpaceDim);
      sprintf(bb_file_path, "%s.nx%d.step%07d.%dd.hdf5", m_plotFile.c_str(), ncells,
m_curStep, SpaceDim);
      dw_stage_file_out(bb_file_path, lustre_file_path, DW_STAGE_IMMEDIATE);
   }
#endif
```

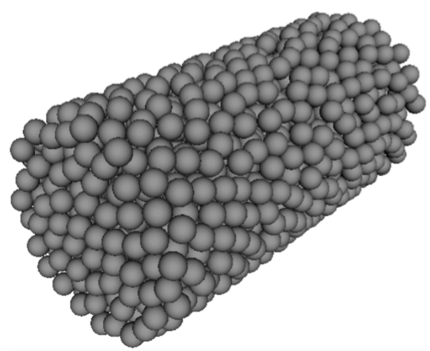
Scaling study: Packed cylinder

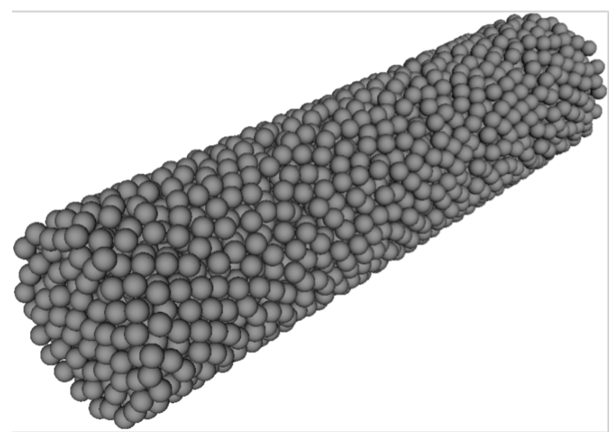
Weak scaling setup (*Trebotich&Graves,2015*)

- Geometry replication
- Number of compute nodes from 16 to 1024
- Ratio of number of compute nodes to BB nodes is fixed at 16:1
- Plotfile size: from 8GB to 500GB



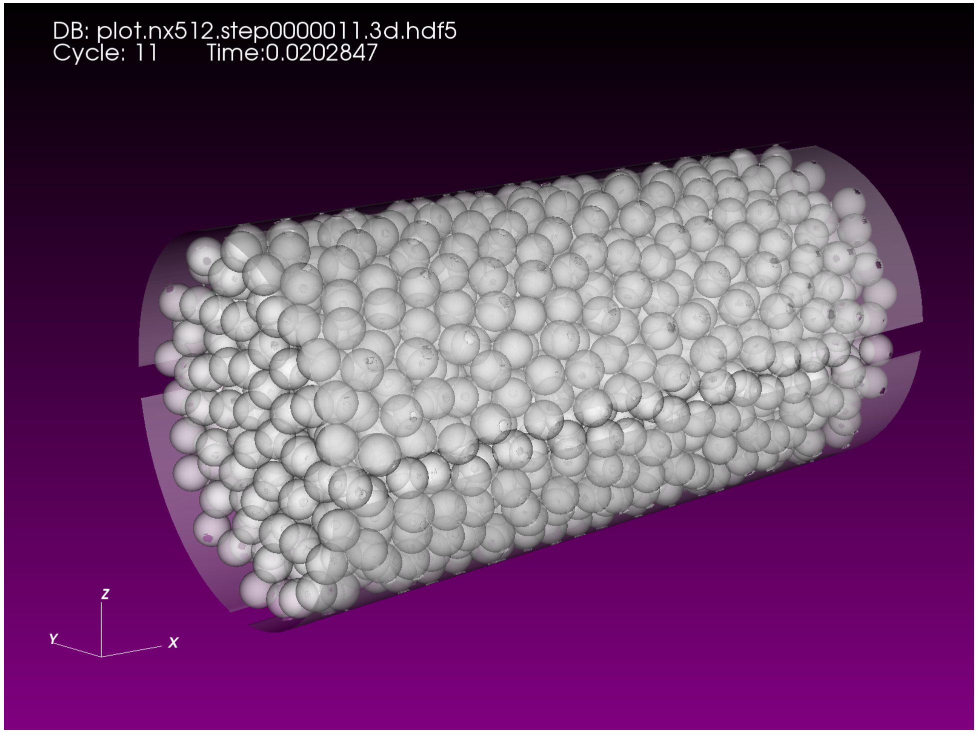


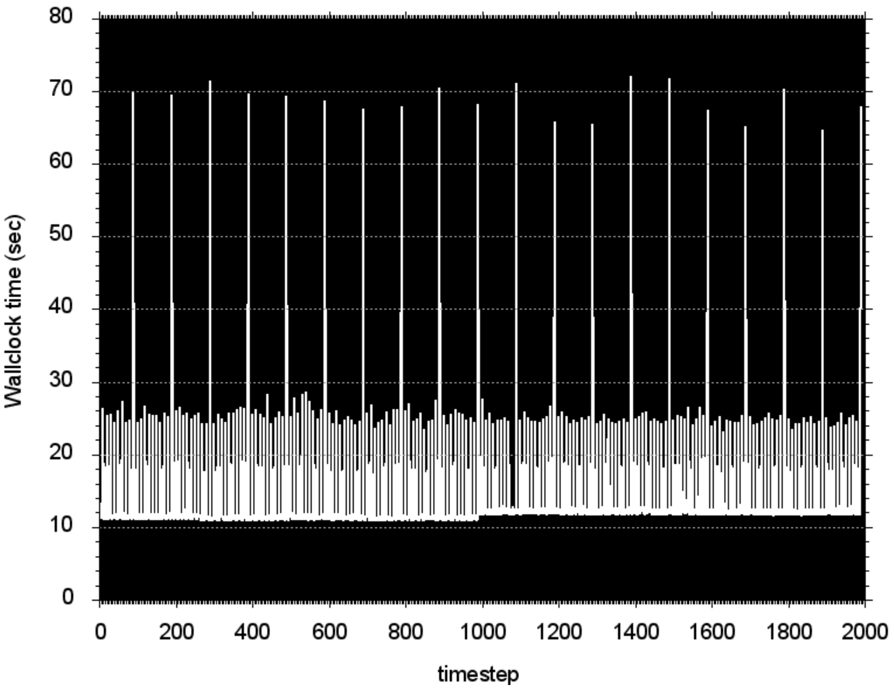




Wall clock history: I/O to Lustre

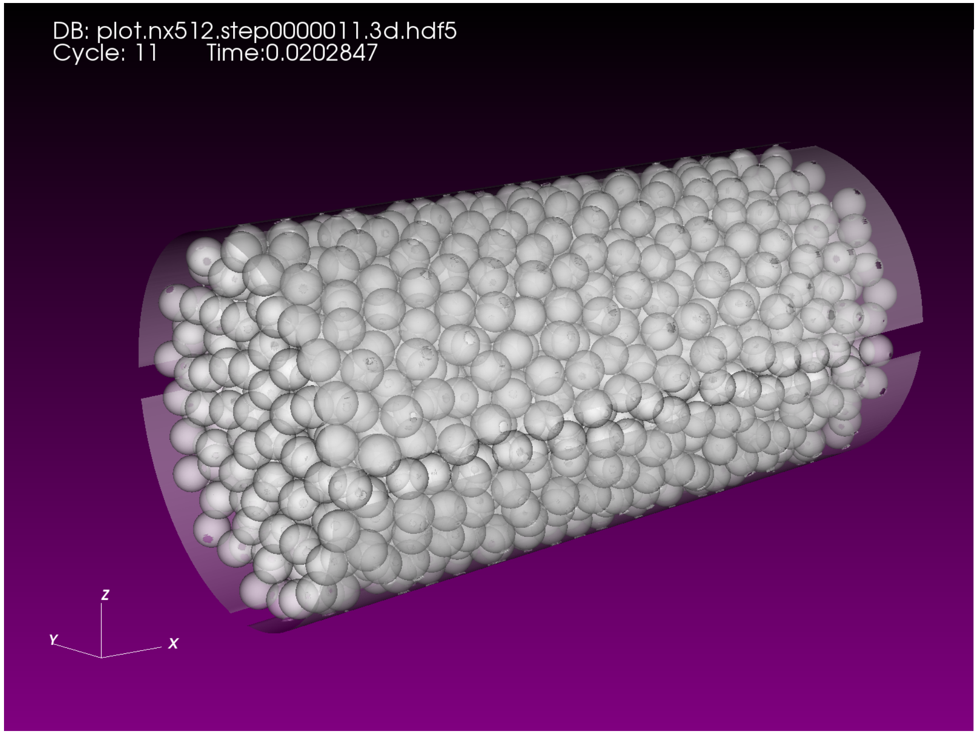
Reactive transport in packed cylinder: **256 compute nodes** (8192 cores) on Cori (HSW partition)
72 OSTs on Lustre (optimal for this file size). Peak I/O bandwidth: **5.6GB/sec**

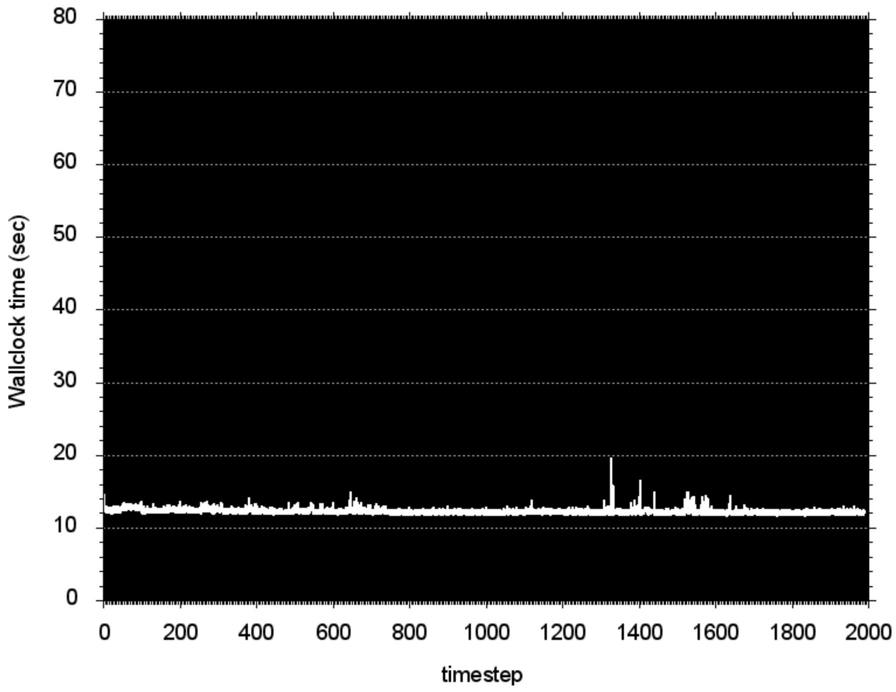




Wall clock history: I/O to BB

Reactive transport in packed cylinder: **256 compute nodes** (8192 cores) on Cori (HSW partition)
128 Burst Buffer nodes. Peak I/O bandwidth: **70.2GB/sec**

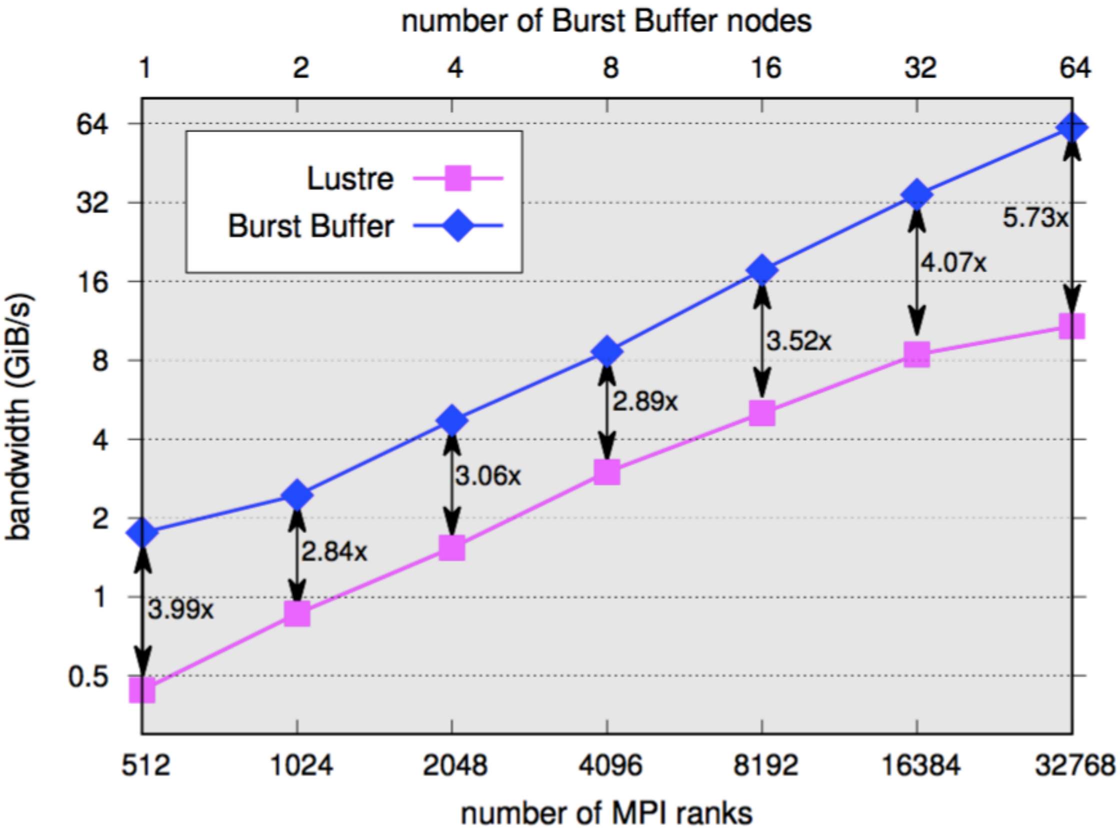




I/O bandwidth study (1)

Now: Number of compute nodes to BB nodes is fixed at 16:1

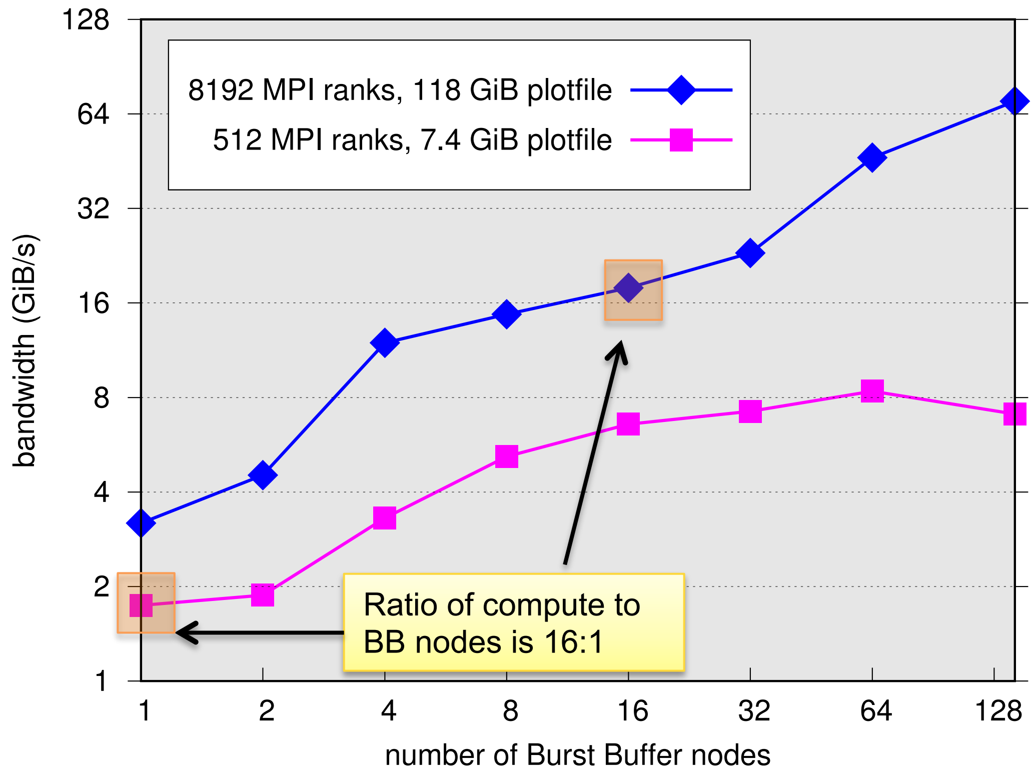
Collective write to shared file using HDF5 library



Scaling study for 16 to 1024 compute nodes on Cori Phase 1.

I/O bandwidth study (2)

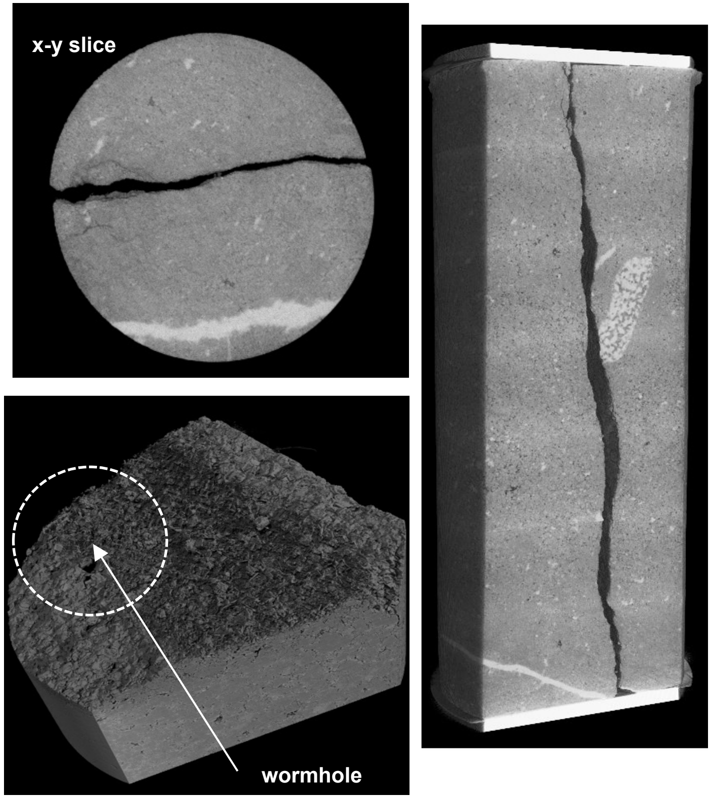
Collective write to shared file using HDF5 library



Write bandwidth study for 7.4GiB and 118GiB file sizes.

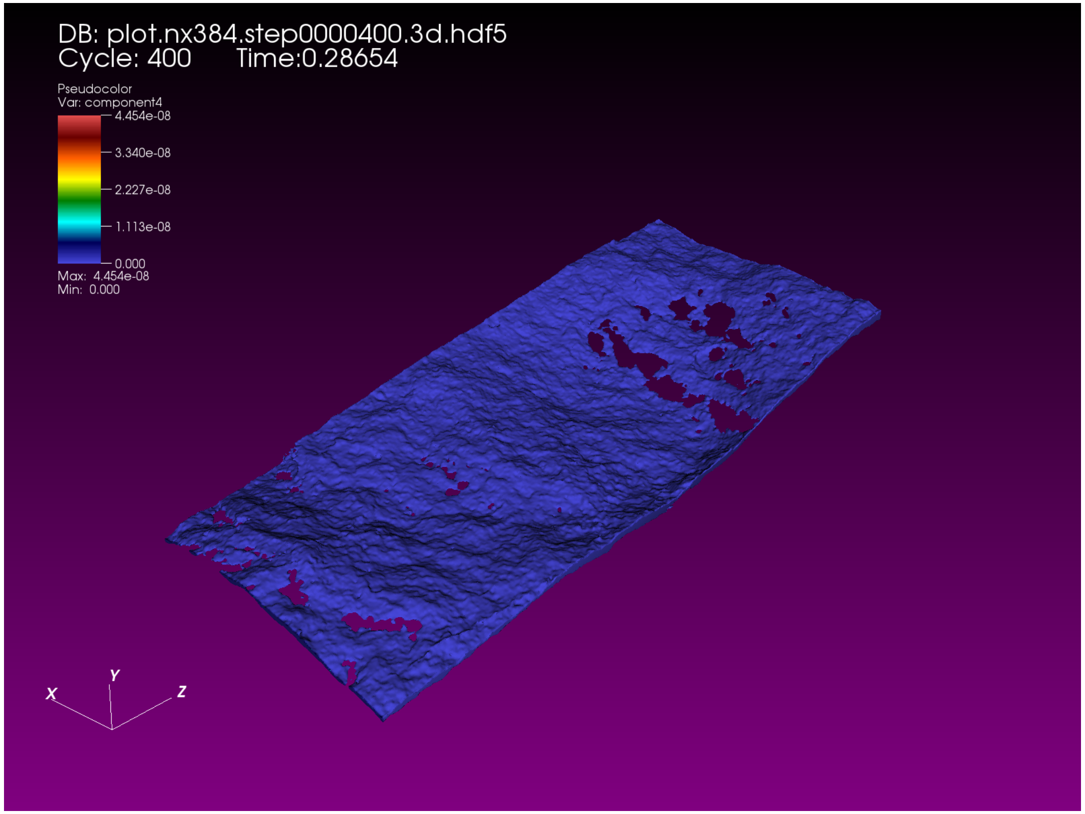
In-transit visualization (2)

Reactive transport in fractured mineral (dolomite): Simulation performed on Cori Phase 1: 512 cores (16 nodes) used by Chombo-Crunch, 64 cores (2 nodes) by VisIt, 128 Burst Buffer nodes for I/O.



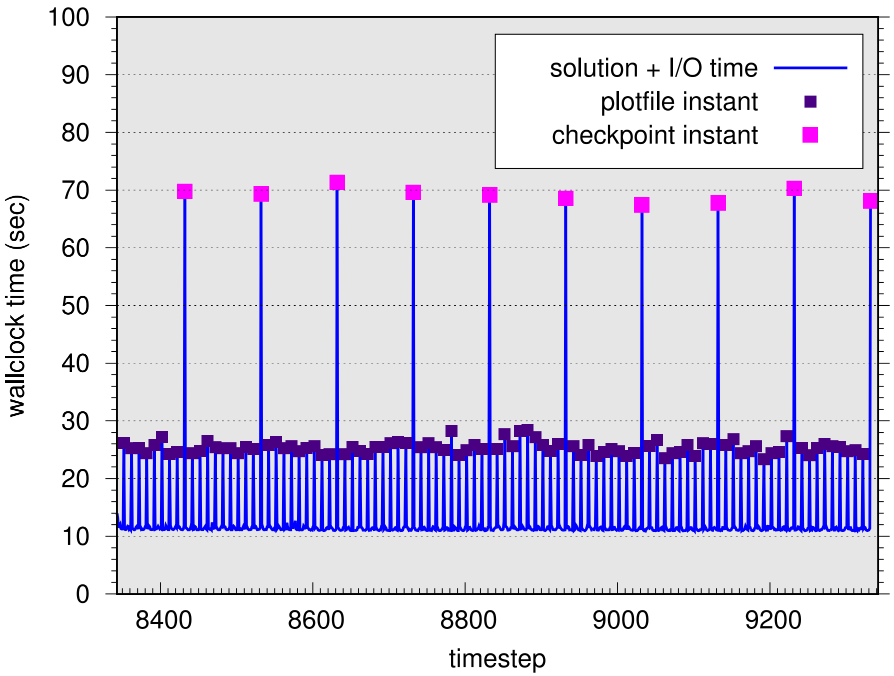
Experimental images courtesy of Jonathan Ajo-Franklin and Marco Voltolini, EFRC-NCGC and LBNL ALS.

Ca^{2+} concentration

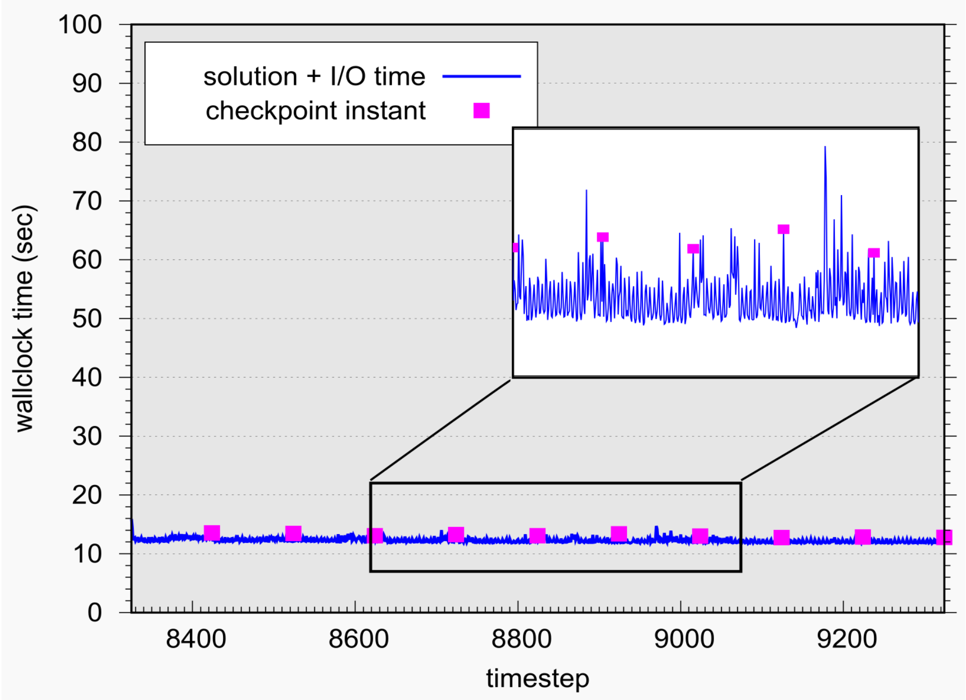


Wall clock time history

With I/O to Lustre PFS



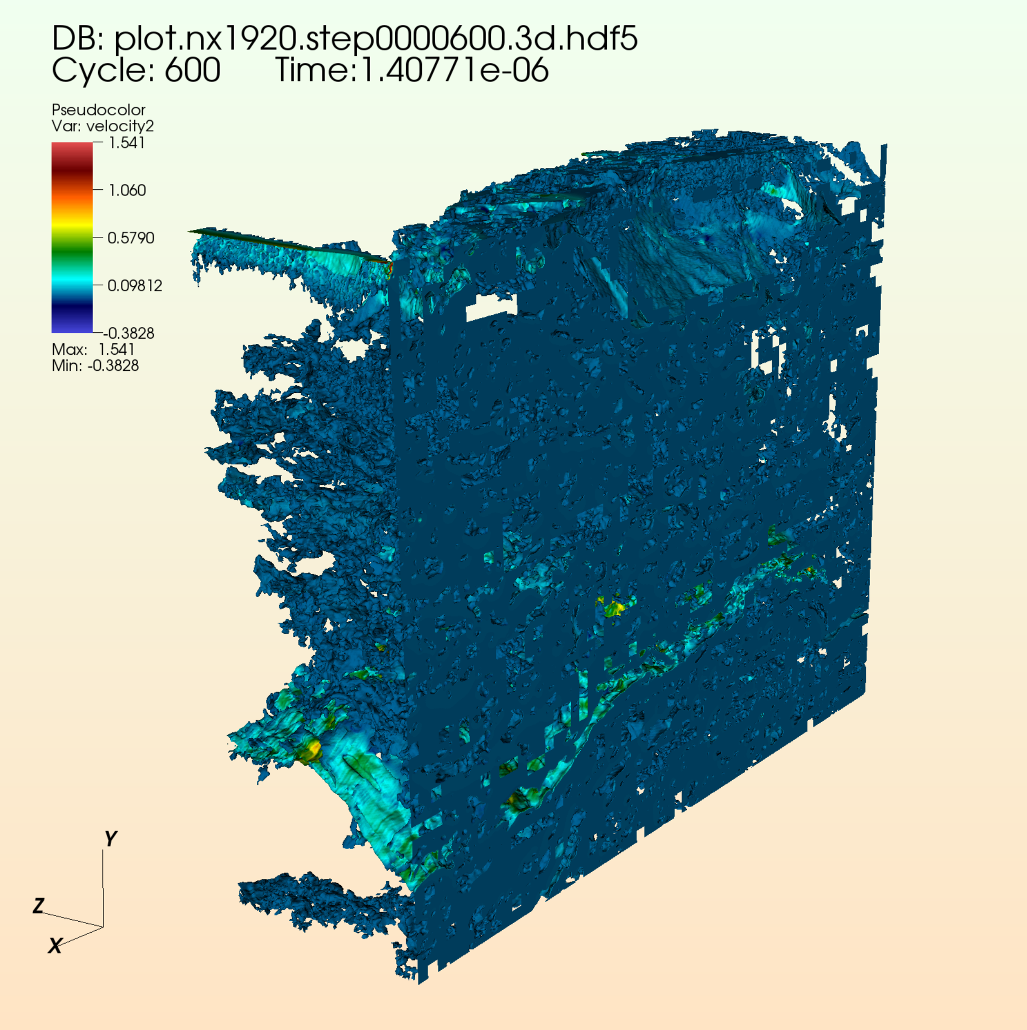
With I/O to Burst Buffer



In-transit visualization (3)

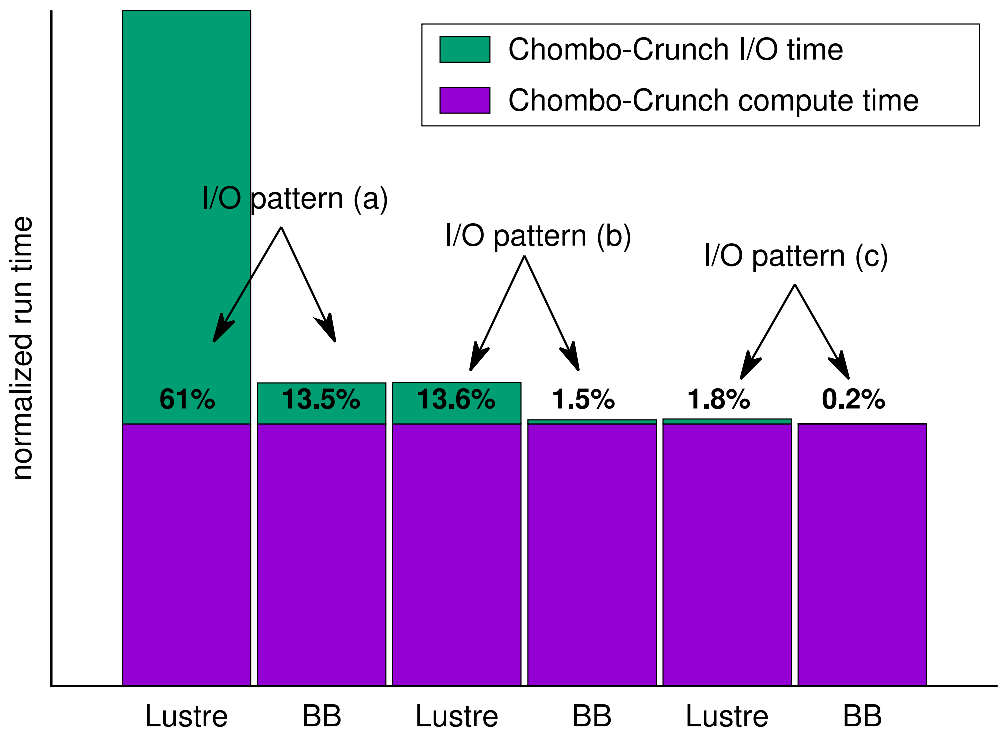
Flow in fractured Marcellus shale

- 0.18 porosity including fracture
- 100 micron block sample
- 48 nm resolution
- 41K cores on Cori Phase 1
- 16 nodes for VisIt
- 144 Burst Buffer nodes
- Plotfile size 290GB



Compute time vs I/O time

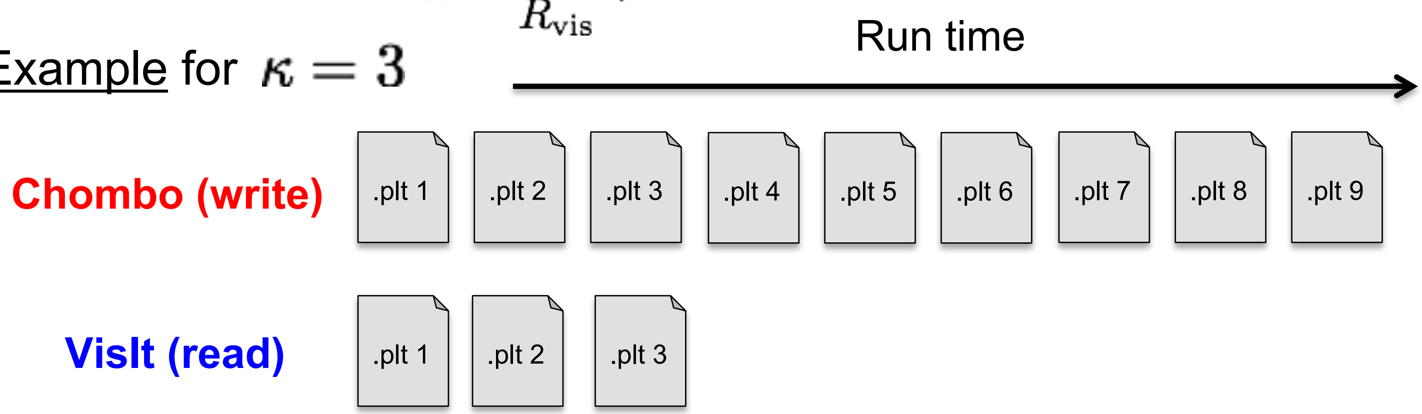
(a) High intensity I/O: plot file every timestep, checkpoint file every 10 timesteps
(b) Moderate intensity I/O: plot file every 10 timesteps, checkpoint file every 100 timesteps
(c) Low intensity I/O: plot file every 100 timesteps, checkpoint file every 500 timesteps



Remaining challenges: i) Load imbalance

Load imbalance when rate of simulation is higher than rate of processing: $\kappa = \frac{R_{\text{sim}}}{R_{\text{vis}}} > 1$

Example for $\kappa = 3$



One will end up with **2/3 of unprocessed plotfiles!**

Solution 1: launch additional $\lceil \kappa \rceil$ VisIt sessions. Use extra job steps (Slurm job arrays) in the same batch script. At the moment it is impossible to kill job step (all nodes will go to idle state).
Solution 2: to use persistent reservation and run additional job(s) for VisIt to process remaining files.

Remaining challenges: ii) Managing BB capacity

BB has a limit size per job. Currently it is 20TB.
Total amount of generated data might overwhelm the required BB capacity.

→ Plot files processed by VisIt should be removed from BB on-the-fly.

Conclusions

- In-transit workflow which couples simulation and visualization has been proposed. DataWarp Burst Buffer has been utilized.

- I/O speedup by utilizing Burst Buffer compared to Lustre file system:
  - **3x-5x** for fixed ratio of compute nodes to BB nodes (16:1)
  - **13x** for peak performance (full BB capacity vs Lustre)

- Burst Buffer allowed Chombo-Crunch to move to every timestep of “data-processing” with minimal changes in the source code.

- **Remaining challenges and ongoing work:**
  - Run-time managing of BB capacity (limit per user will be ~20TB)
  - Dynamic component load balancing
  - Including additional components into workflow:
    - coupling pore-scale with reservoir scale simulation
    - extra VisIt sessions for quantitative analysis (computing flow statistics, reactions rates, pore graph extractor, ...)

Thank you!

Contact: aovsyannikov@lbl.gov