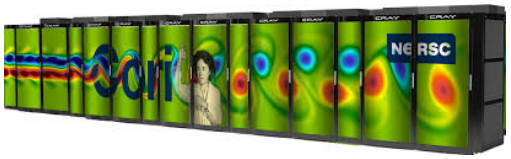


Science with the Burst Buffer: Accelerating Scientific Workflows in Chombo-Crunch

Andrey Ovsyannikov* (NERSC, LBL)
with David Trebotich, Brian Van Straalen (CRD, LBL)

*aovsyannikov@lbl.gov

U.S. DEPARTMENT OF ENERGY | Office of Science

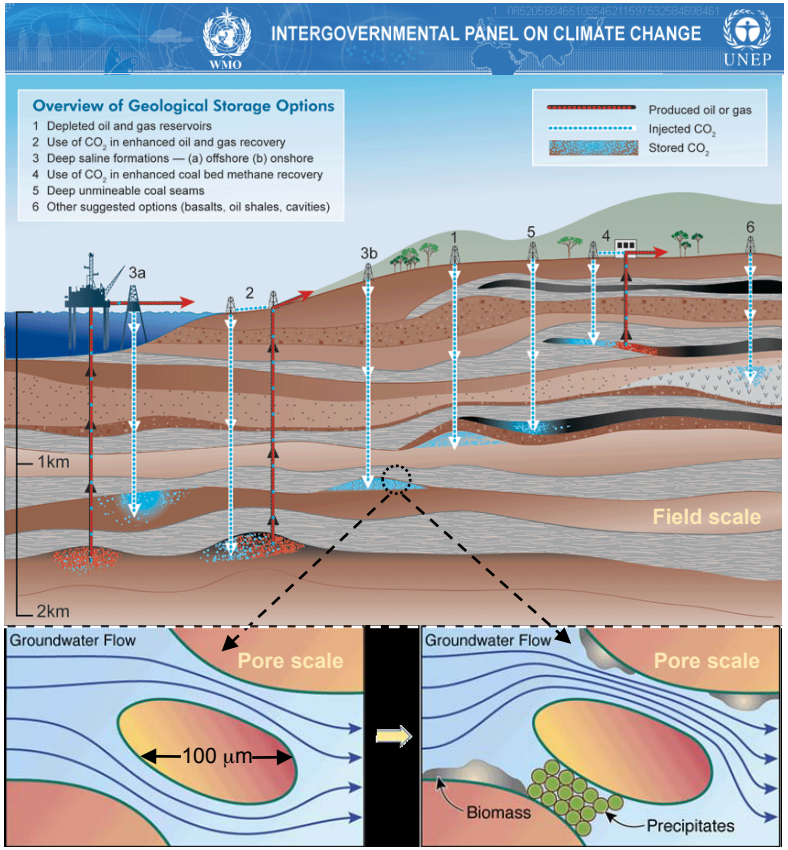
August 22nd, 2016

BERKELEY LAB

Carbon sequestration

Main goal is to enable accurate prediction of the fate of geologically stored CO_2

Flow and transport typically simulated at field scale



CO_2 trapping mechanisms governed by emergent processes at pore (micro) scale

➔ **Need high resolution pore scale reactive transport model**

➔ **Need methods to upscale pore scale information to field scale**

Other porous media applications

Other important Chombo-Crunch applications:

- ❑ Shale gas extraction
- ❑ Used fuel disposition (Hanford salt repository modeling)
- ❑ Lithium ion battery electrodes
- ❑ Paper manufacturing (hpc4mfg)

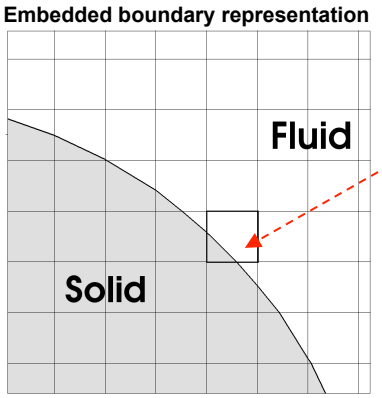
The common feature is ability to perform direct numerical simulation from image data of arbitrary heterogeneous, porous materials.

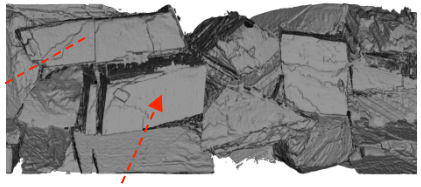
Chombo-Crunch

CFD + single phase multi-component geochemical reactive transport in very complex pore (micro) scale geometries
Adaptive, finite volume methods for advection-diffusion with sources in:

- **Chombo**
  - Accurate reactive surface area using embedded boundaries
  - Dynamic local refinement (AMR)
  - Scalable (100K+ processors)
  - DNS from geologic image data

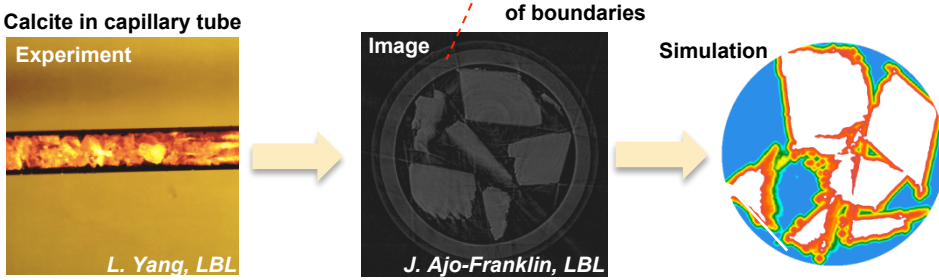
- **CrunchFlow** geochemistry
  - Point-by-point calculation
  - CFL-limited timestep

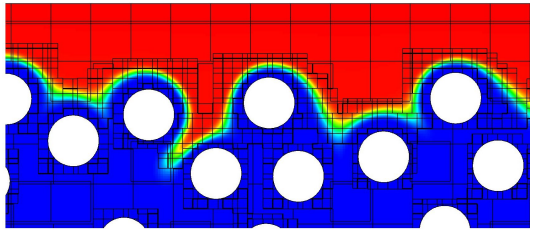




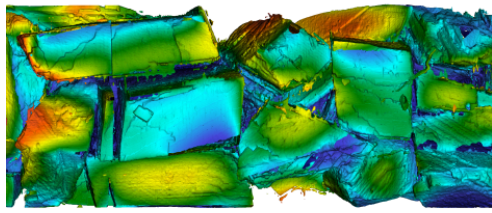
Rate calculated at each water-mineral interface by multiplying by the reactive surface area (RSA)

Image data converted to simulation grid using implicit function representation of boundaries





Adaptive mesh refinement



pH on crushed calcite in capillary tube

Mathematical model at pore-scale

Mathematical model of pore scale flow and reactive transport

- Incompressible flow and advection-diffusion-reaction

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p = \nu\Delta\mathbf{u}, \qquad \nabla\cdot\mathbf{u} = 0$$

Incompressible Navier-Stokes

Conservative transport

$$\frac{\partial \rho c_k}{\partial t} + \nabla\cdot\rho\mathbf{u}c_k = \nabla\cdot\rho D_k\nabla c_k + \rho r_k$$

Reactions

Geochemical System

5 components

9 complexation reactions

Calcite Dissolution

Precipitation

k_1: $CaCO_3(s) + H^+ \rightarrow Ca^{2+} + HCO_3^-$

k_2: $CaCO_3(s) + H_2CO_3^* \rightarrow Ca^{2+} + 2HCO_3^-$

k_3: $CaCO_3(s) \rightarrow Ca^{2+} + CO_3^{2-}$

k_3: $Ca^{2+} + CO_3^{2-} \rightarrow CaCO_3(s)$

$$r_k'[mol\ m^{-2}s^{-1}] = \left(k_1 a_{H^+} + k_2 a_{H_2CO_3^*} + k_3\right)\left(1 - \frac{a_{Ca^{2+}} a_{CO_3^{2-}}}{K_{eq}}\right)$$

Pore scale simulation to improve continuum scale models

Pore scale information can be up-scaled to parameterize continuum scale (Darcy) model

	Pore scale	Continuum scale
Flow	$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \nabla p = \nu \Delta \boldsymbol{u}$ $\nabla \cdot \boldsymbol{u} = 0$	$\boldsymbol{q} = -\frac{\boldsymbol{k}}{\mu} \nabla p$ $\nabla \cdot (\boldsymbol{k} \nabla p) = 0$
Transport	$\frac{\partial c}{\partial t} = \nabla \cdot (D \nabla c) - \nabla \cdot (\boldsymbol{u} c)$	$\boldsymbol{\theta} \frac{\partial c}{\partial t} = \nabla \cdot (\mathcal{D} \nabla c) - \nabla \cdot (\boldsymbol{q} c)$ *where* $\mathcal{D} = \boldsymbol{\theta \tau} D + \boldsymbol{\alpha_L} \lvert \boldsymbol{u} \rvert$
Reaction	$-D \nabla c \cdot \boldsymbol{n} = k a_i^n (1 - Q/K)$	$r = k \boldsymbol{A} a_i^n (1 - Q/K)$

Multiscale methods

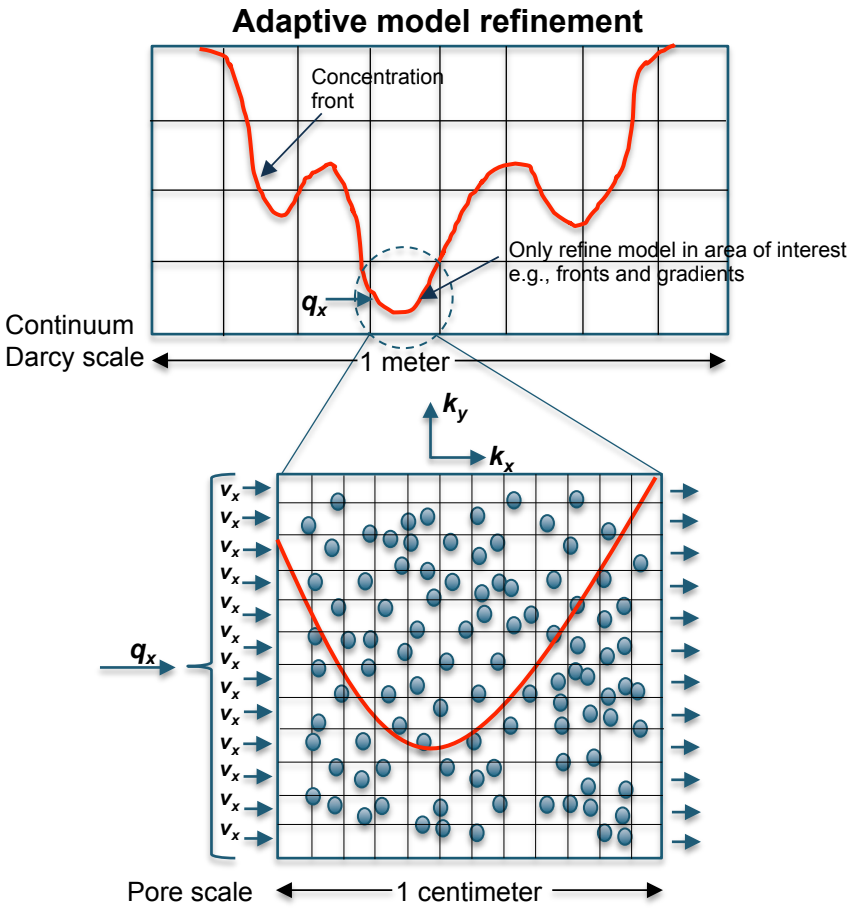
Multiscale approach: Use first-of-its-kind highly resolved pore scale simulation data (i) to inform better parameterizations of permeability, reaction rates and dispersion at continuum Darcy scale, and (ii) to verify multiscale approaches:

Deterministic approach:

- "Brute force": resolve known pore scale domain and upscale everywhere
- Adaptive model refinement: upscale pore scale data to continuum scale *locally* in areas of interest in domain

Stochastic approach:

- Intermediate pdf: characterize random pore space with a probability density function that is fitted to the pore scale data or graph connectivity

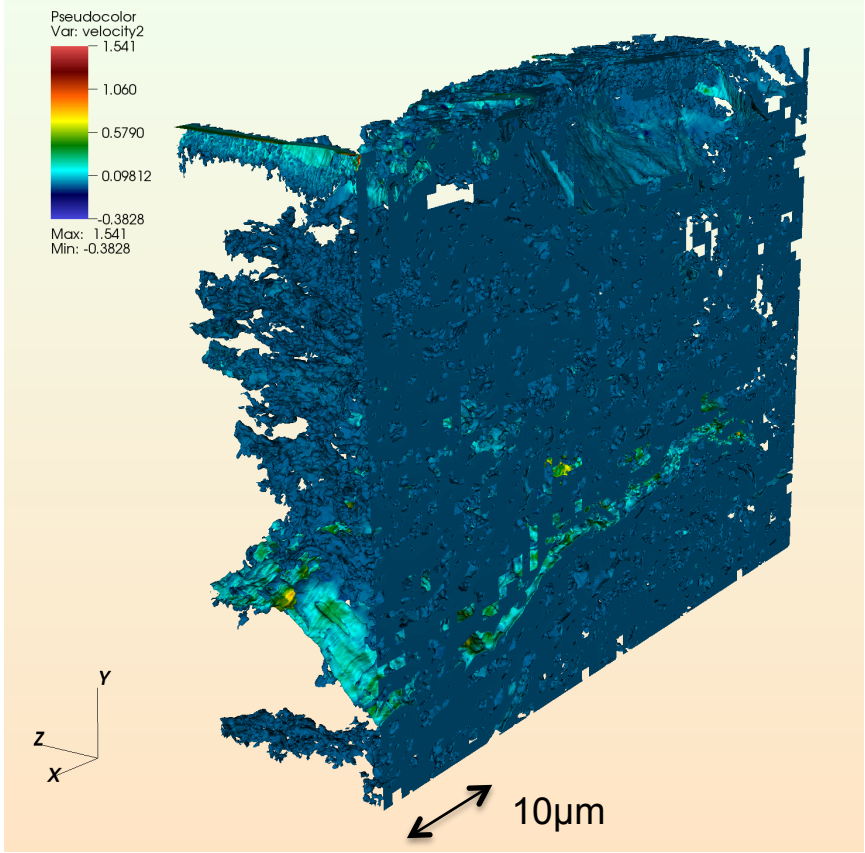


Data-intensive simulation at scale

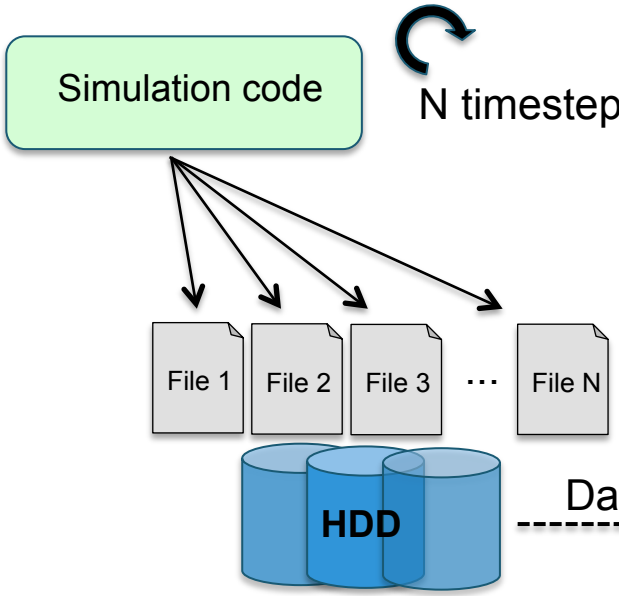
Example: Reactive flow in a shale

- Computational resources: **41K cores**
- Space discretization: **2 billion cells**
- Time discretization: **~1μs;**
  in total 3x10^4 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**

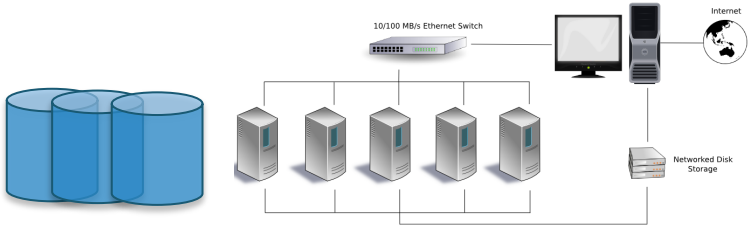
*Assuming that the plotfile is written at every timestep



Traditional post-processing



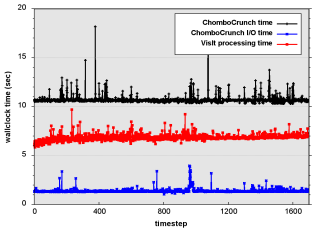
$$\frac{\mathbf{U}^{n+1} - \mathbf{U}^{n}}{\Delta t} = \mathbf{F}(\mathbf{U}^{n}, \mathbf{x}, t^{n})$$

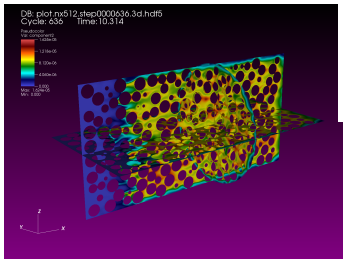


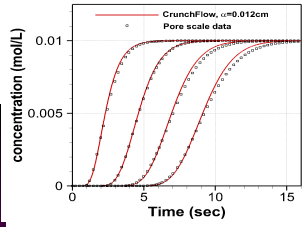
Remote storage: e.g. Globus Online, visualization cluster,...

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

Data analysis/ Visualization



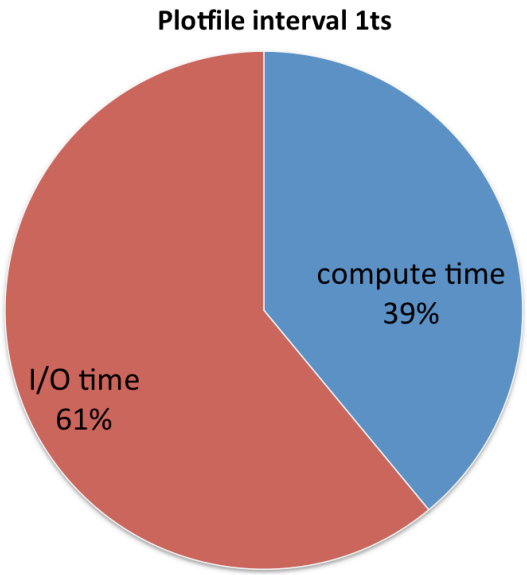


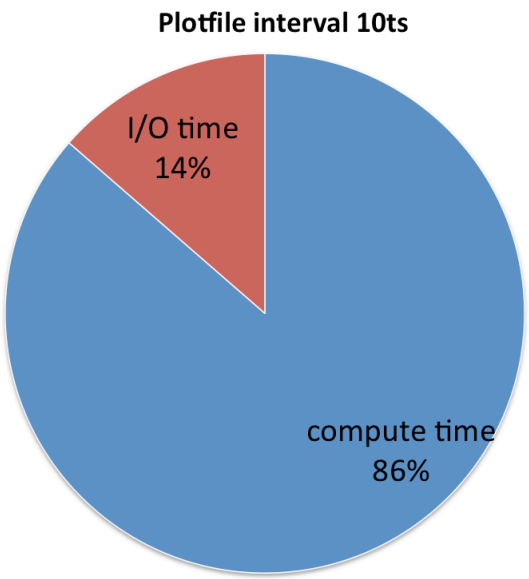


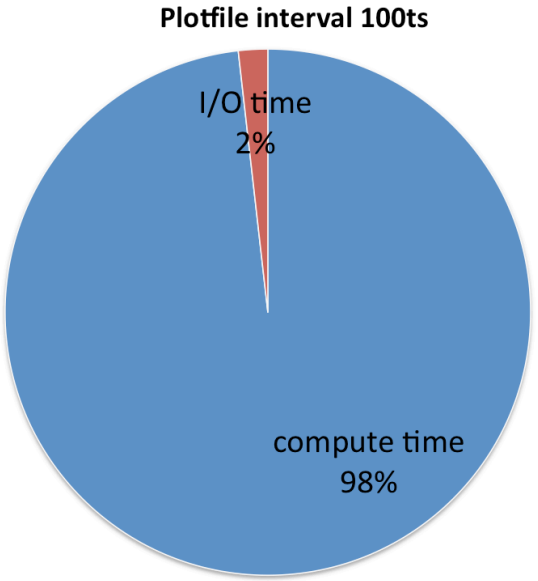
I/O constraint: common practice

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

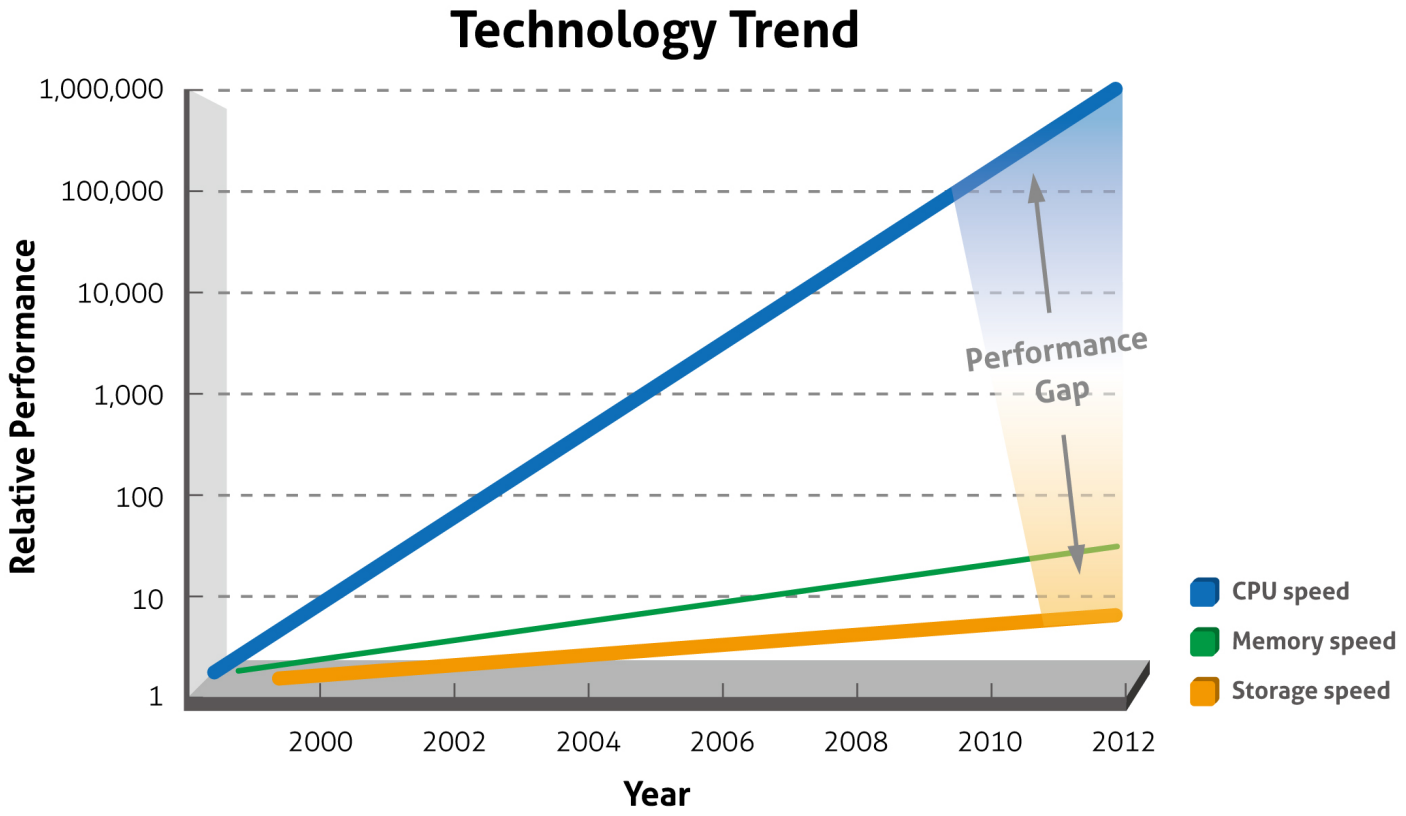
I/O contribution to Chombo-Crunch wall time at different plotfile intervals







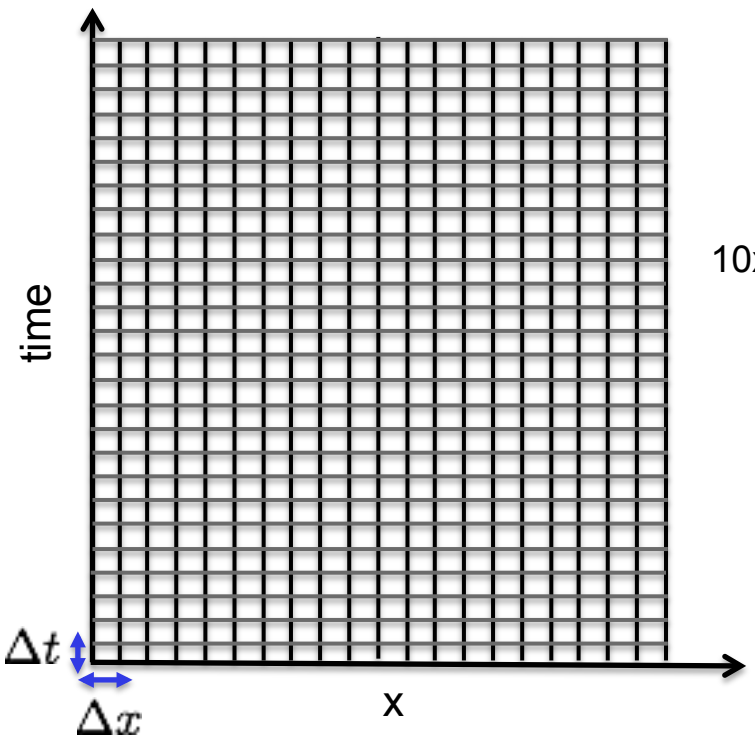
Exascale perspective



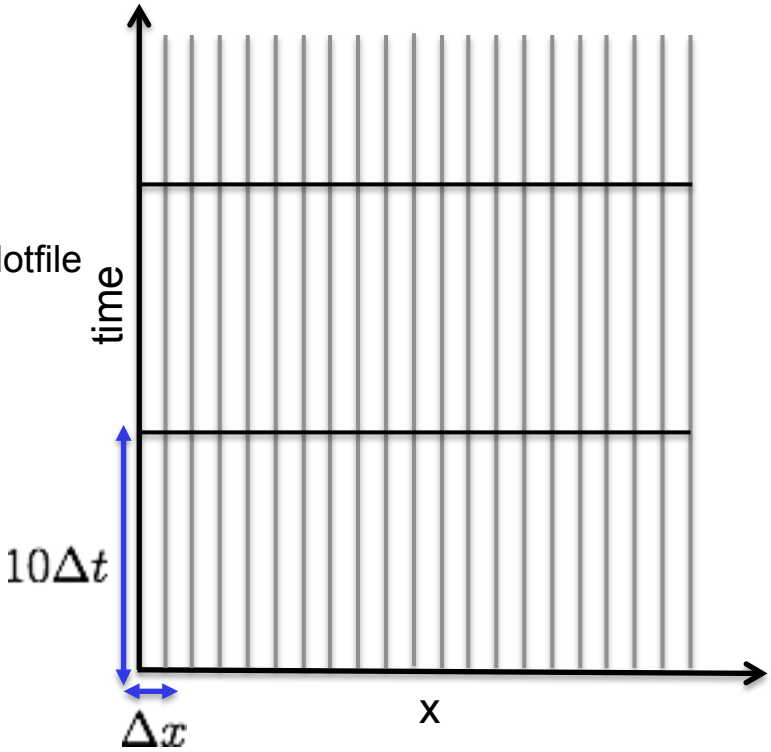
Growing gap between computation and I/O rates.
Insufficient bandwidth of persistent storage media.

Loss of temporal/statistics accuracy

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



10x increase of plotfile interval



Pros: less data to move and store

Cons: degraded accuracy of statistics (stochastic simul.)

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

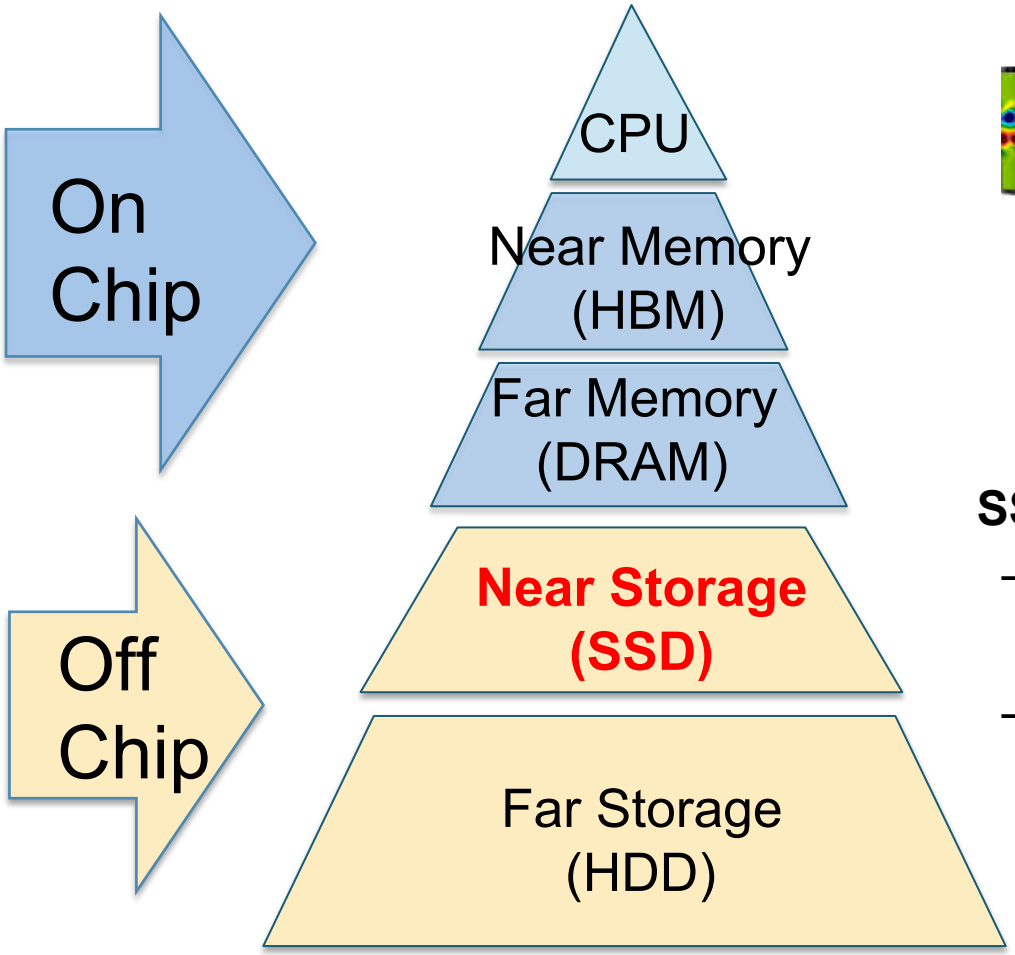
Data processing methods

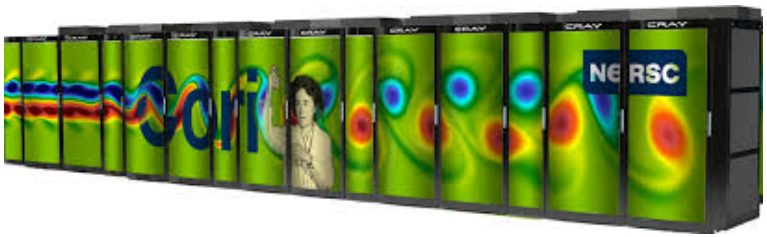
Data processing execution methods (Prabhat & Koziol, 2015)

	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

Burst Buffer

Memory hierarchy





SSD-based Burst Buffer:

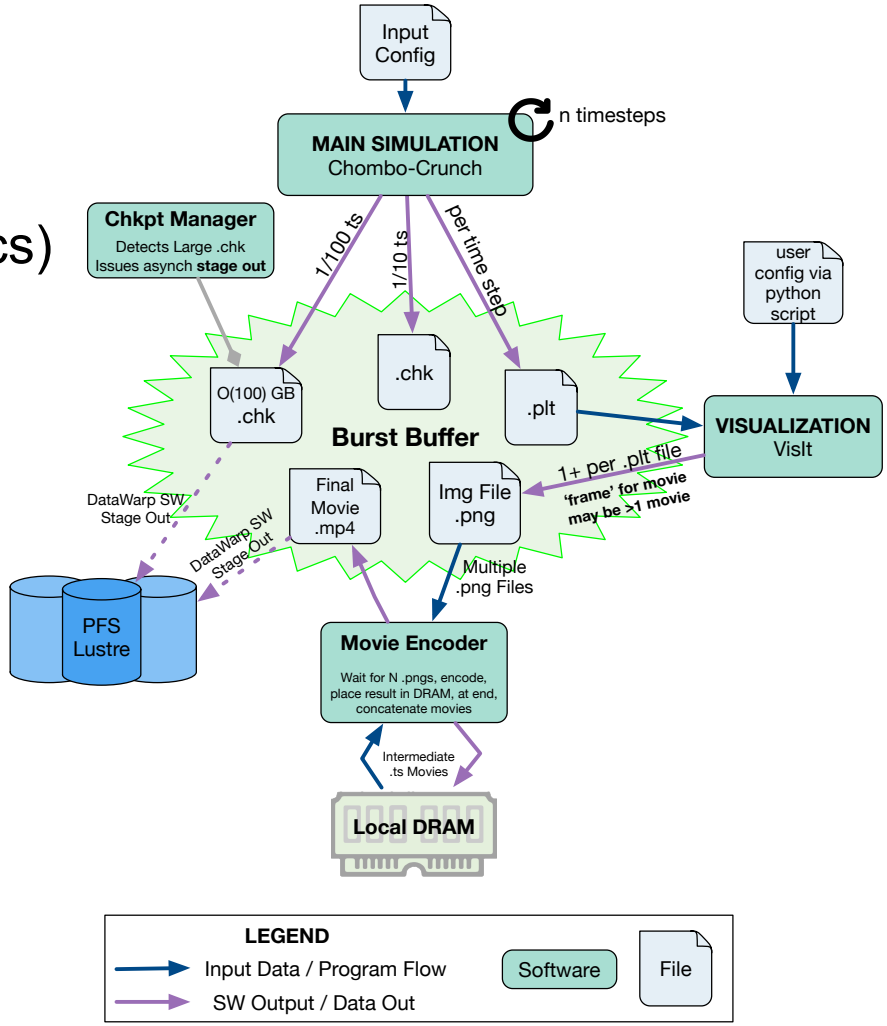
- Lower latency, higher bandwidth of flash-based Burst Buffer than PFS
- Better scalability than large PFS

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 →

```
#!/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
```

run each component →

transfer output product to persistent storage →

DataWarp API

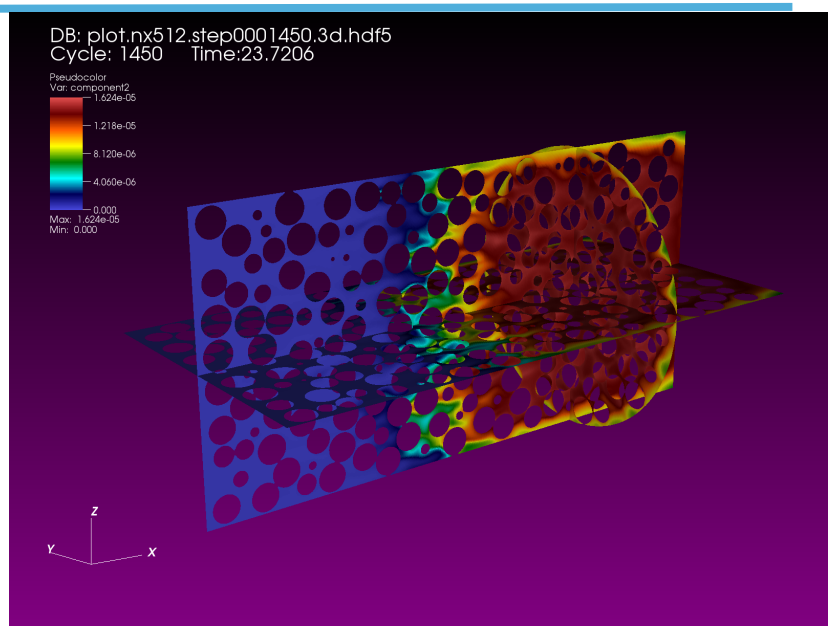
Asynchronous transfer of plot file/checkpoint from Burst Buffer to PFS

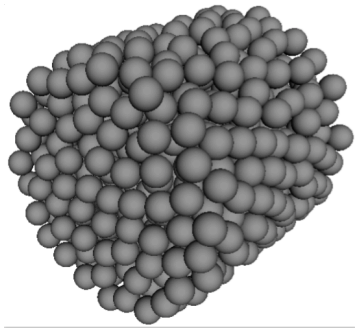
```
#ifdef CH_DATAWARP
// use DataWarp API stage_out call to move plotfile from BB to Lustre
   char lustre_file_path[200];
   char bb_file_path[200];

   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
```

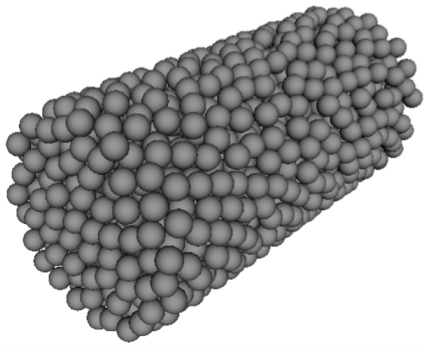
Scaling study: Packed cylinder

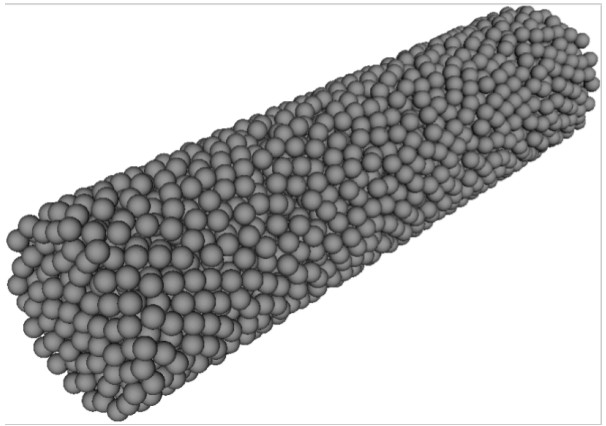
Weak scaling setup (*Trebotich&Graves,2015*)

- 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



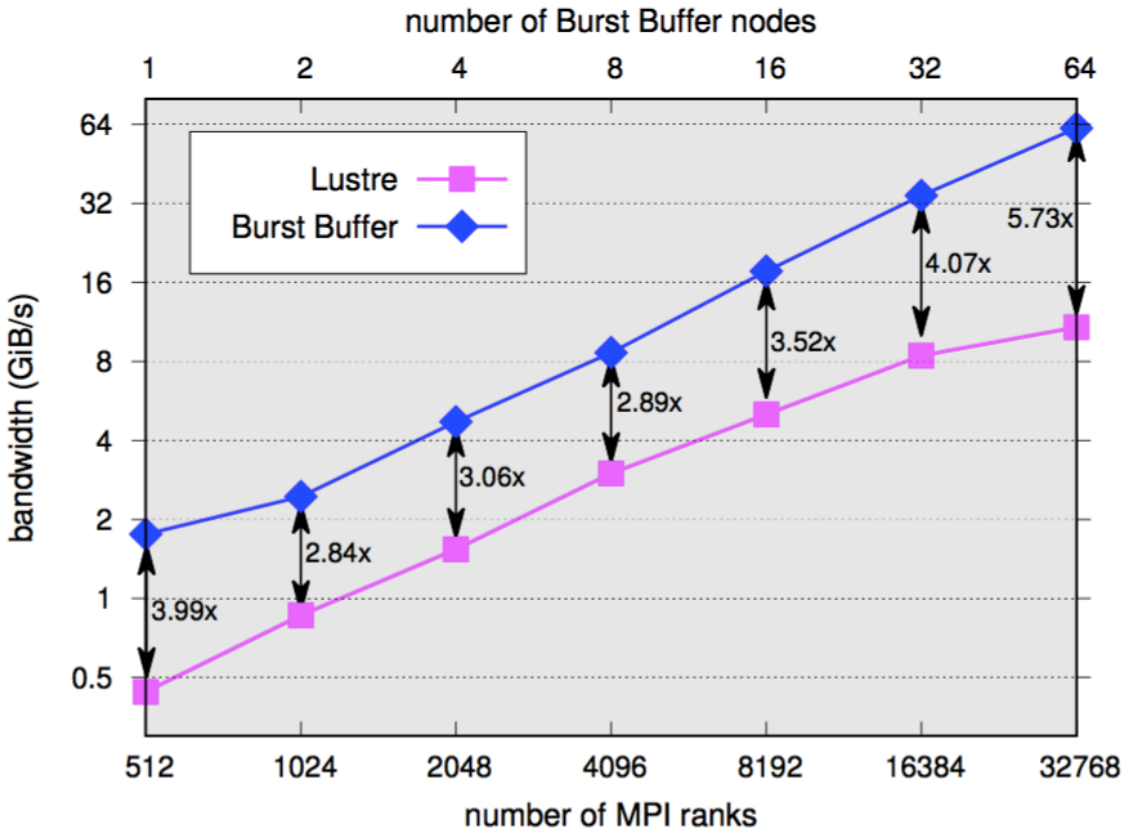






I/O bandwidth study (1)

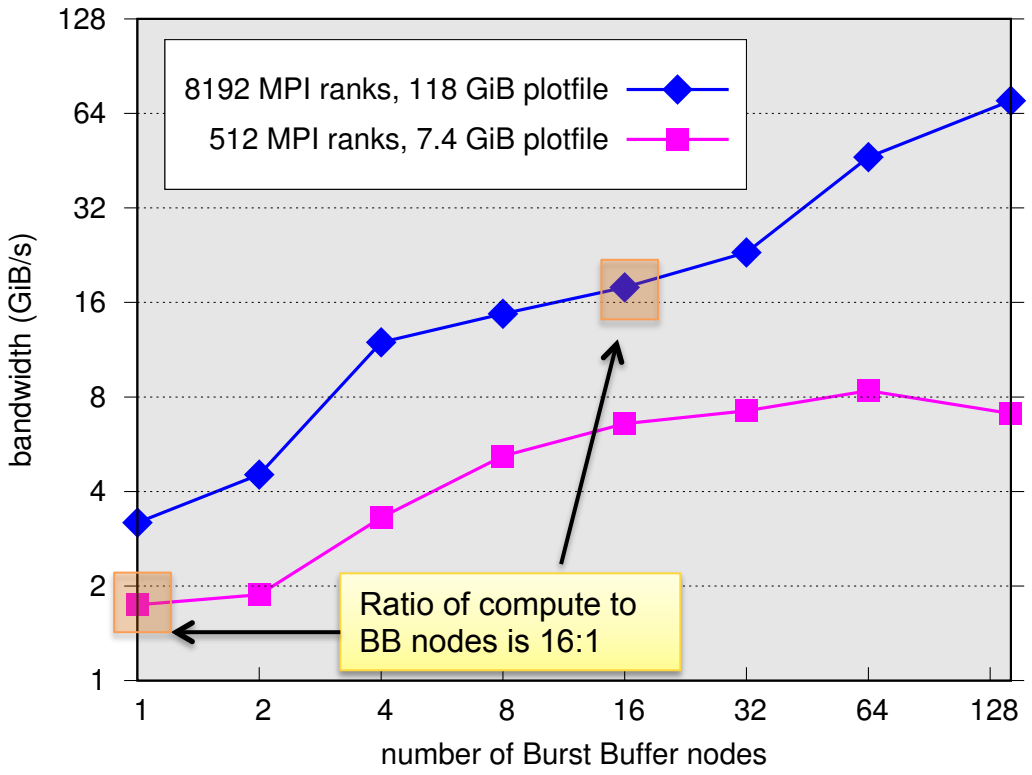
Collective write to shared file using HDF5 library



Scaling study for 16 to 1024 compute nodes on Cori Phase 1.
Number of compute nodes to BB nodes is fixed at 16:1.

I/O bandwidth study (2)

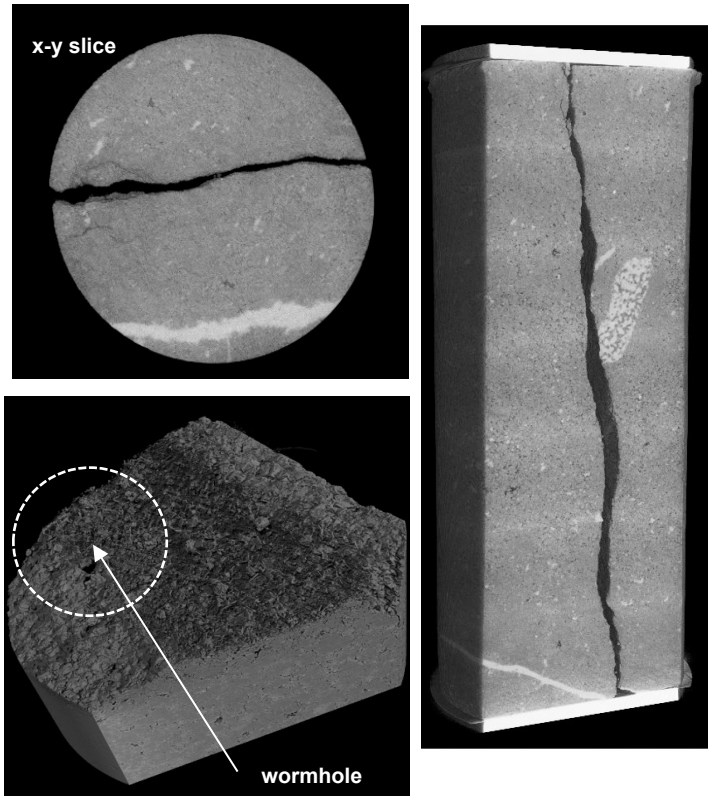
Collective write to shared file using HDF5 library



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

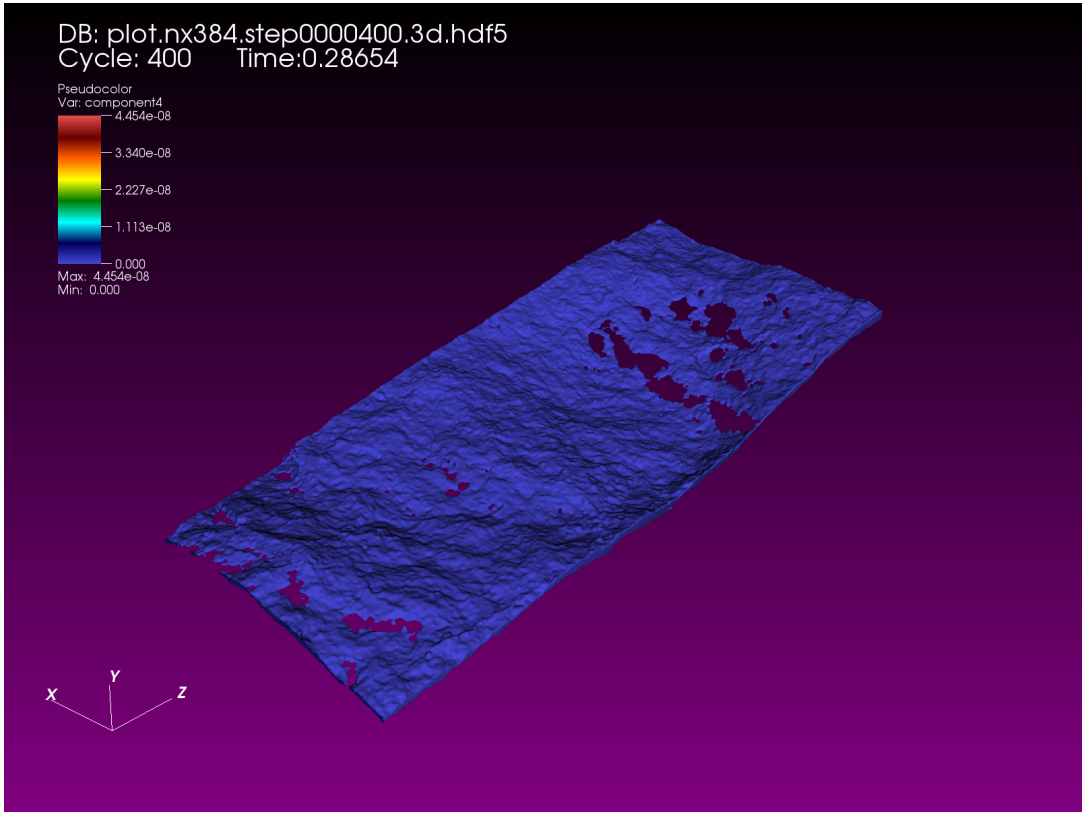
In-transit visualization (2)

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, 4 Burst Buffer nodes for I/O.



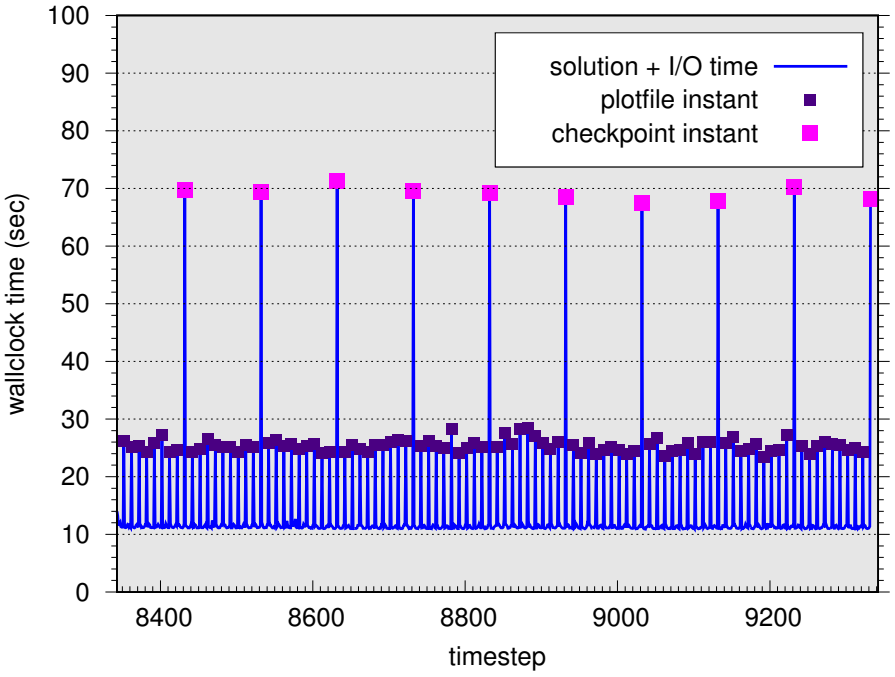
Experimental images courtesy of Jonathan Ajo-Franklin and Marco Voltolini, EFRC-NCGC and LBNL ALS.

Ca^{2+} concentration

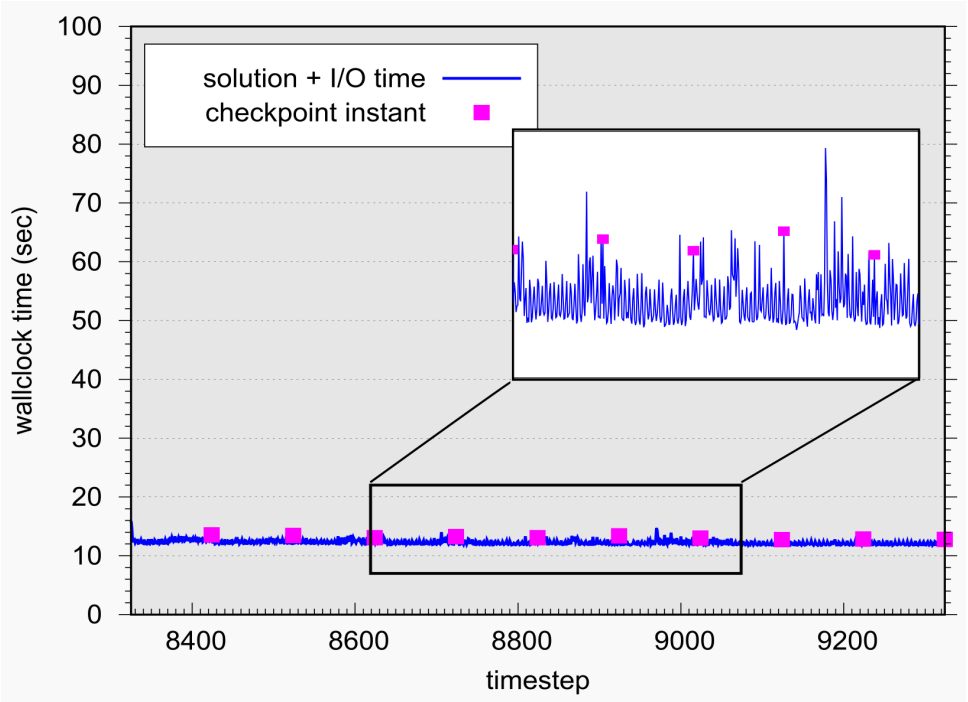


Wall clock time history

With I/O to Lustre PFS



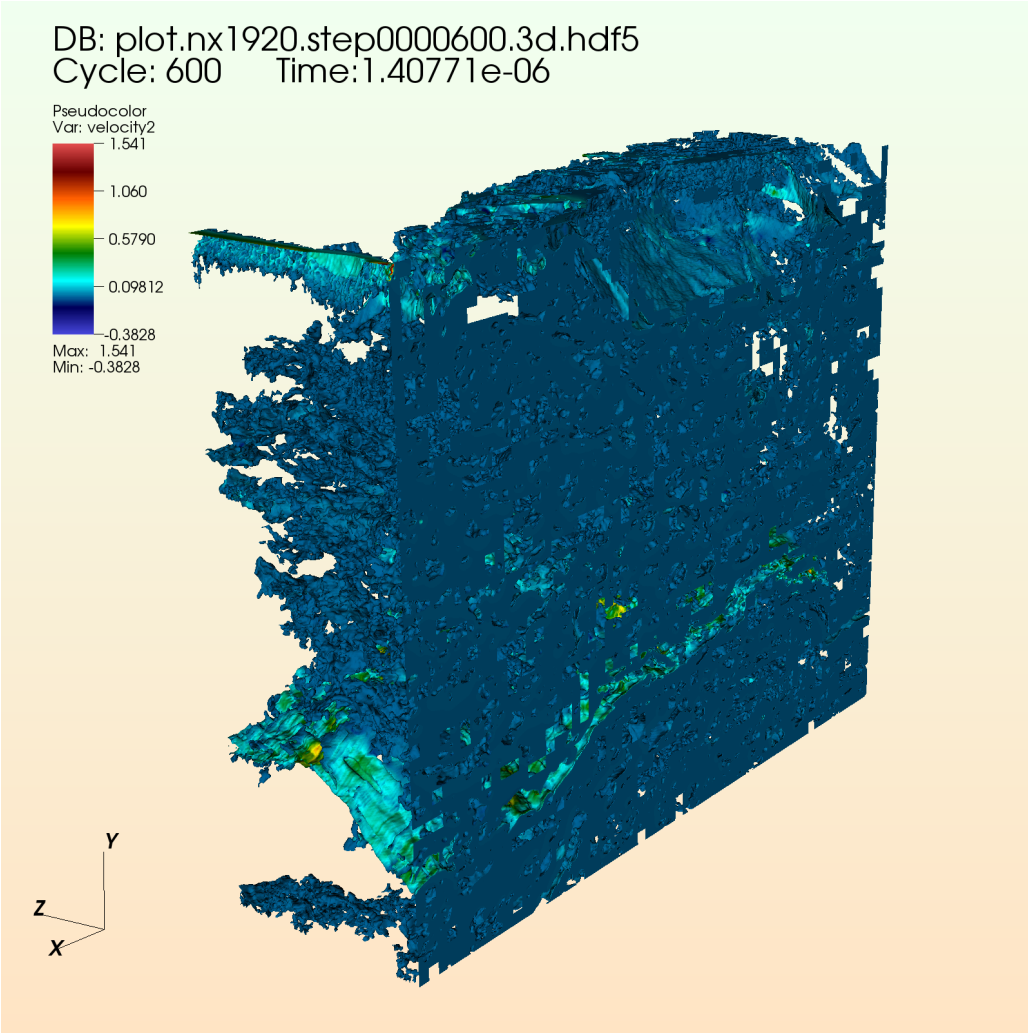
With I/O to Burst Buffer



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

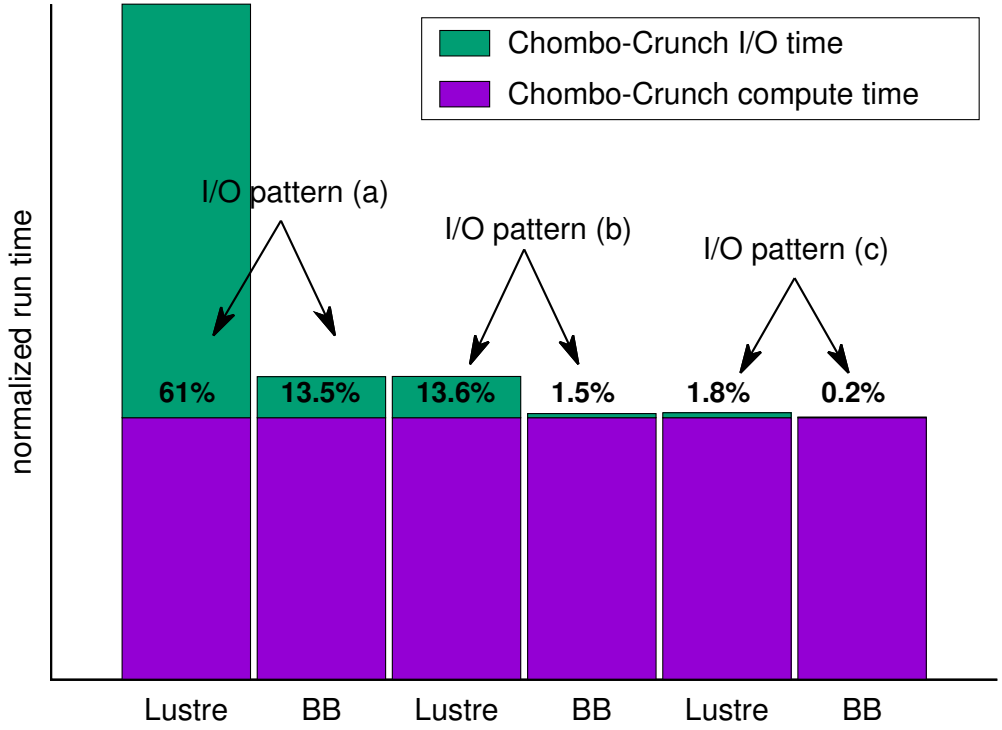


Compute time vs I/O time

(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



Conclusions

- In-transit workflow which couples simulation and visualization has been proposed. The performance has been assessed at large-scale by running Chombo-Crunch simulation of reactive flows in porous media from image data.
- First results showed 3x to 20x I/O speedup for BB in comparison with Lustre file system.
- 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: extra VisIt sessions for quantitative analysis (e.g. reactions rates, pore graph extractor)

Acknowledgments

We are gratefully acknowledge the Burst Buffer Early User Program and in particular:

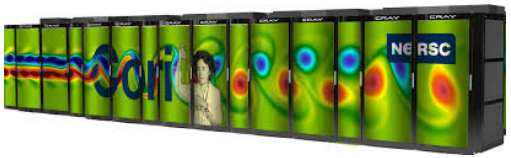
Melissa Romanus, Rutgers U.

Gunther Weber, CRD-LBL

Dave Paul, NERSC-LBL

Debbie Bard, NERSC-LBL

Wahid Bhimji, NERSC-LBL



Thank you!

Science with the Burst Buffer: Accelerating Scientific Workflows in Chombo-Crunch

Andrey Ovsyannikov* (NERSC, LBNL)
with David Trebotich, Brian Van Straalen (CRD, LBNL)

*aovsyannikov@lbl.gov

U.S. DEPARTMENT OF ENERGY | Office of Science

August 22nd, 2016

BERKELEY LAB