Storage and I/O Issues in Large Scale Scientific Computing

David R. O'Hallaron

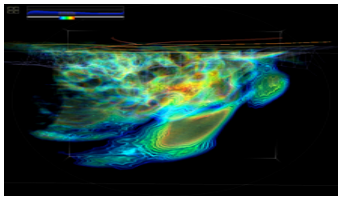
Associate Professor of CS and ECE

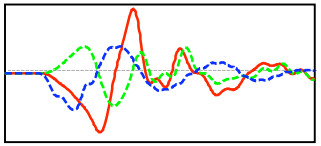
Carnegie Mellon Quake Group

www.cs.cmu.edu/~quake

Funded in part by SCEC, NSF, DOE, and Vanguard Charitable Endowment

Anatomy of an End-to-end Simulation





Parallel supercomputer w/1-10Ks PEs

x20K timesteps

Mesh generation

Solving

Analysis/ visualization

DFS

Velocity model

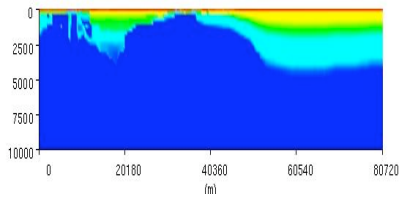
Unstructured mesh

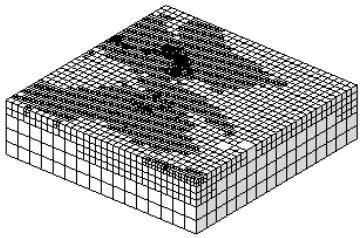
4D vel/dis wavefield

10s MB

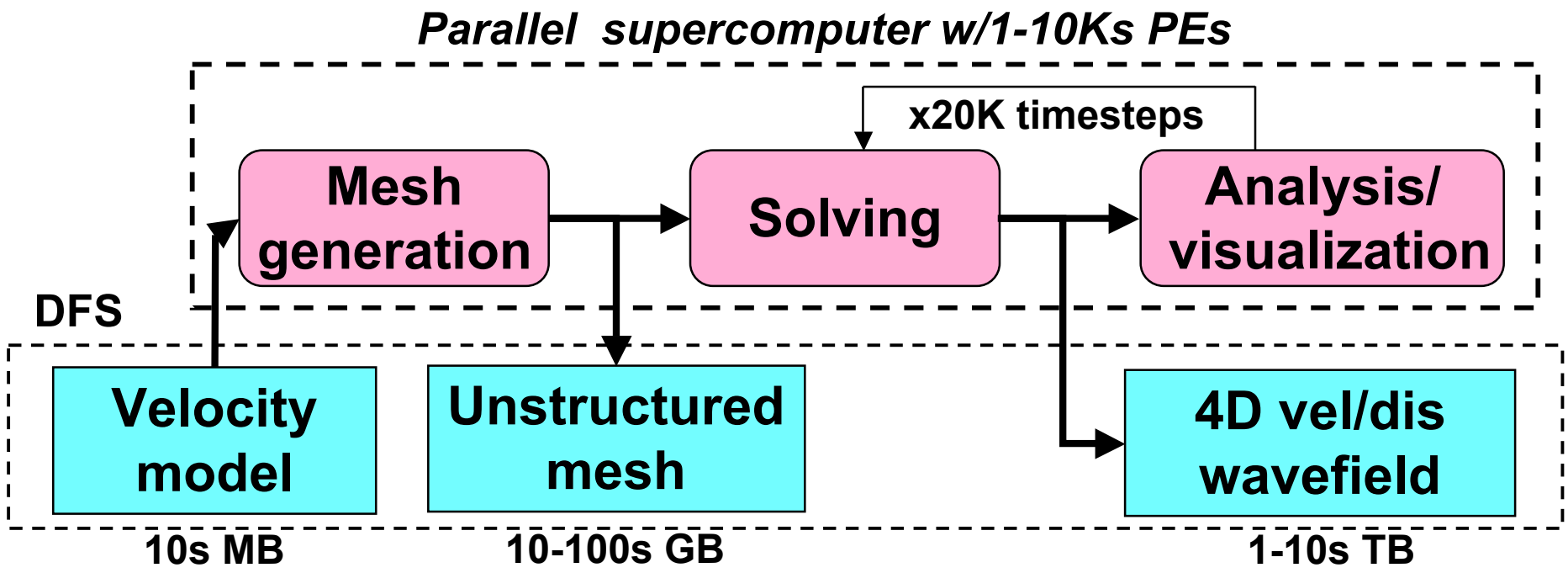
10-100s GB

1-10s TB





End-to-end Simulation I/O Patterns



All PEs, phased partitioned reads.
Each PE reads many small 4KB blocks from its part of input file

1 PE, almost sequential writes.
90% of writes are 4KB appends, 10% are random 4KB writes (b-tree index)

All PEs, partitioned writes.
Each PE writes 1 large disjoint block per timestep x 20K (must avoid lockstep)

Verification and Validation (V&V)

***Verification*: Does the synthetic data match the simulation's specification?**

- **Compare synthetic data to analytical solutions or outputs of other codes.**

***Validation:* Does the synthetic data match real data?**

- **Compare synthetic data to real data.**

Key point: Both verification and validation can be difficult at large scales.

- **SCEC verification effort.**
- **Terrashake verification.**
- **Northridge (1994) validation.**

Why is V&V hard?

1. It's difficult to manipulate large datasets.

Flat files don't cut it anymore!

- **All input and output files should be stored in compressed and indexed databases.**
- **Need new techniques for indexing and querying compressed data.**

Traditional SQL-style point and range queries not sufficient.

- **Need *computational database systems* that:**
  - **Support query operators such as transpose, transform, filter, resample, interpolate, and volume render.**
    - **These are complex parallel HPC apps in their own right.**
  - **Keep track of derived data.**

Why is V&V hard?

2. It's difficult to share large datasets with others.

None of the current options work well:

- Copy the file from the local to the remote site.
- Provide local logins for everyone who needs to access the data.
- Provide remote database access.

Virtual machines (VMs) might provide a solution.

- Remote users package their own VM and ship it to local site.
- VM is placed close to the dataset and then mounted in VM.

Why is V&V hard?

3. It's difficult to compare large datasets.

Quake project Terashake example:

- UCSD and CMU codes differ.
- Are the input datasets equivalent?
- Where (in time and space) do the synthetic datasets diverge?

Need a "diff" command for large (possibly non-local) floating point datasets.

- Datasets are typically populated fields.
- Different codes populate the field differently
  - CMU: values are stored at nodes of an octree mesh.
  - UCSD: values are stored at nodes of regular mesh.

Summary

Current practice for V&V is ~~a disgrace~~ unsatisfactory.

V&V is difficult at scale because:

- **It's difficult to manipulate large datasets**
- **It's difficult to shared large datasets with others**
- **It's difficult to compare large (possibly remote) datasets**

Many V&V issues are storage related, and thus a good focus for PDSI.