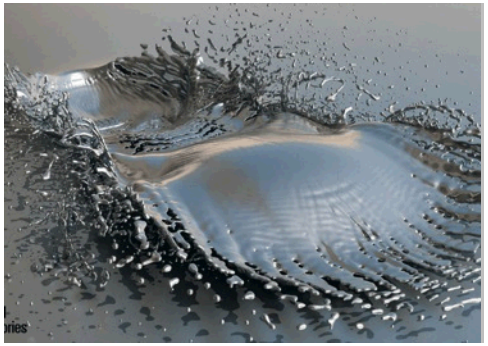
Exceptional service in the national interest

SAND2015-0592PE

Sandia National Laboratories



Enabling Capabilities for Analysis at Extreme Scale

Ron A. Oldfield

Manager, Scalable Analysis and Visualization
Sandia National Laboratories
Albuquerque, NM, USA

JOWOG 34 Applied Computer Science Meeting
February 2015

Approved for public release: SAND2015-XXXX

U.S. DEPARTMENT OF ENERGY · NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Extreme-Scale Computing

- Trends: More FLOPS with comparatively less storage, I/O bandwidth
  - Consequence: A smaller fraction of data can be captured on disk

Oak Ridge National Laboratory

	System Peak	I/O BW
Jaguar (2008)	263 TFLOPS	44 GB/s
Jaguar PF (2009)	1.75 PFLOPS	240 GB/s
Titan (2012)	20 PFLOPS	240 GB/s
Factor Change	**76×**	**5.5×**

Bland, Kendall, Kothe, Rogers, and Shipman. "Jaguar: The World's Most Powerful Computer"
http://archive.hpcwire.com/hpcwire/2012-10-29/titan_sets_high-water_mark_for_gpu_supercomputing.html?featured=top

Argonne National Laboratory

	System Peak	I/O BW
Intrepid (2003)	560 TFLOPS	88 GB/s
Mira (2011)	10 PFLOPS	240 GB/s
Factor Change	**17.8×**	**2.7×**

https://www.alcf.anl.gov/intrepid
https://www.alcf.anl.gov/mira

Lawrence Livermore National Laboratory

	System Peak	I/O BW
ASC Purple (2005)	100 TFLOPS	106 GB/s
Sequoia (2012)	20 PFLOPS	1 TB/s
Factor Change	**200×**	**9.4×**

http://www.sandia.gov/supercomp/sc2002/flyers/SC02ASCIPurplev4.pdf
https://asc.llnl.gov/publications/Sequoia2012.pdf

Sandia National Laboratories

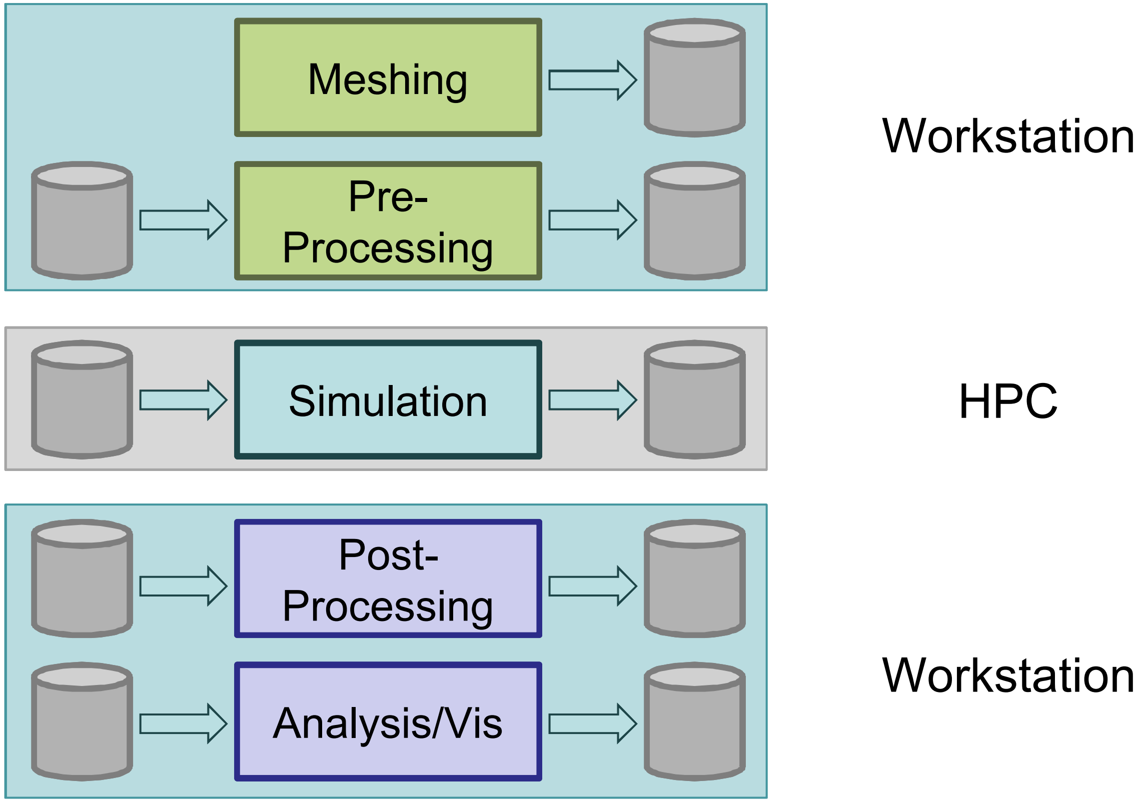
	System Peak	I/O BW
Red Storm (2003)	180 TFLOPS	100 GB/s
Cielo (2011)	1.4 PFLOPS	160 GB/s
Factor Change	**7.8×**	**1.6×**

https://cfwebprod.sandia.gov/cfdocs/CCIM/docs/033768p.pdf
http://www.lanl.gov/orgs/hpc/cielo/

Usage Models Conflict with Trends

App workflows historically use parallel file system for communication

For some use cases, parts of the workflow execute on different platforms (e.g., analyst desktop)

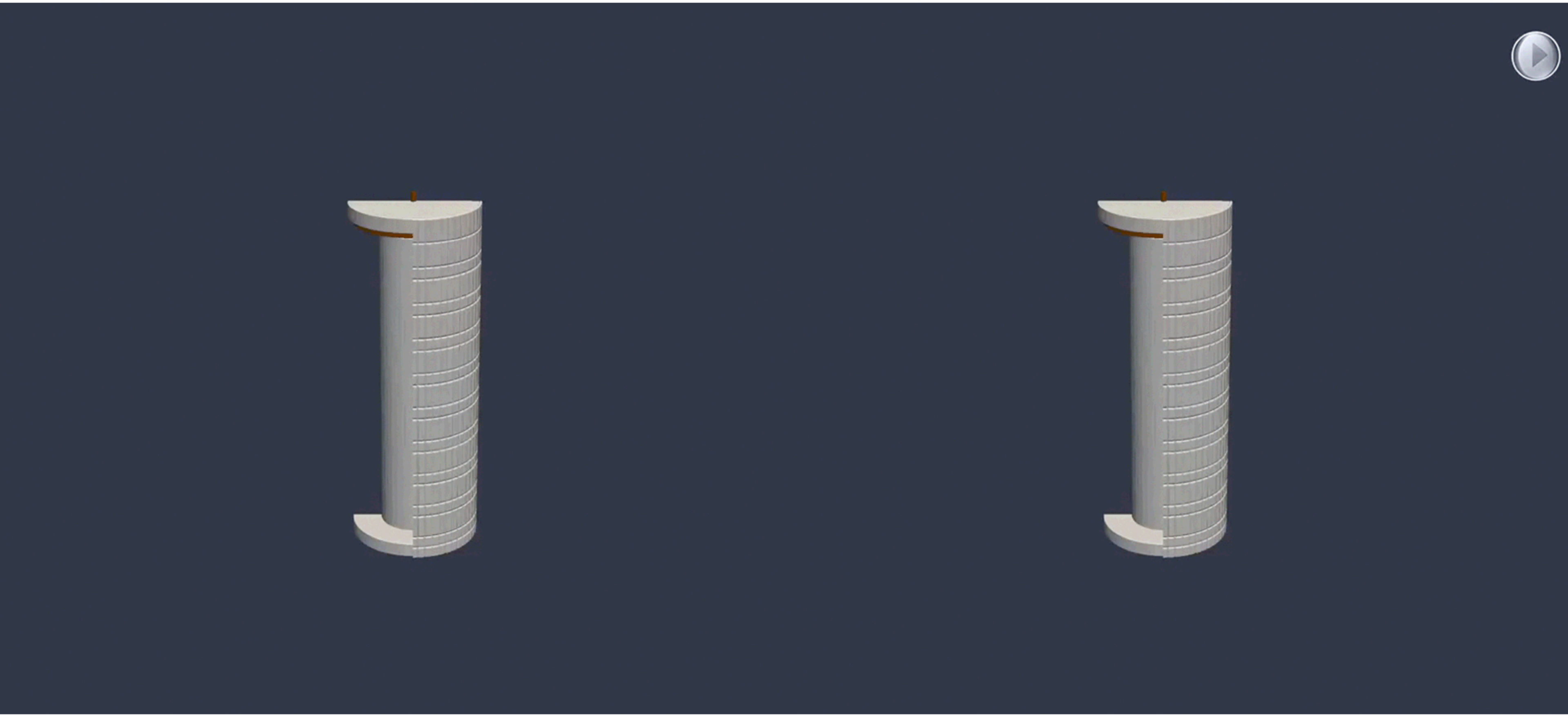


One way to relieve I/O pressure is to integrate components (avoid the FS).

Integrating Simulation and Analysis

A compelling motivation for integration is “resolution of analysis”

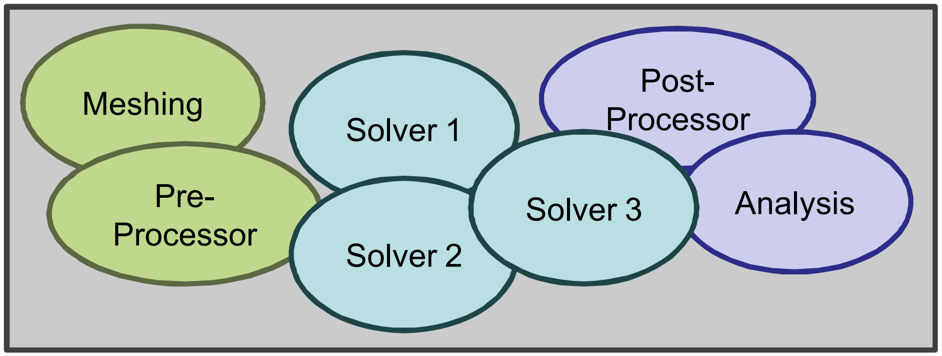
Post-Processing/Offline Analysis | **Integrated Analysis**



Analysis every 100 time steps | Analysis every time step

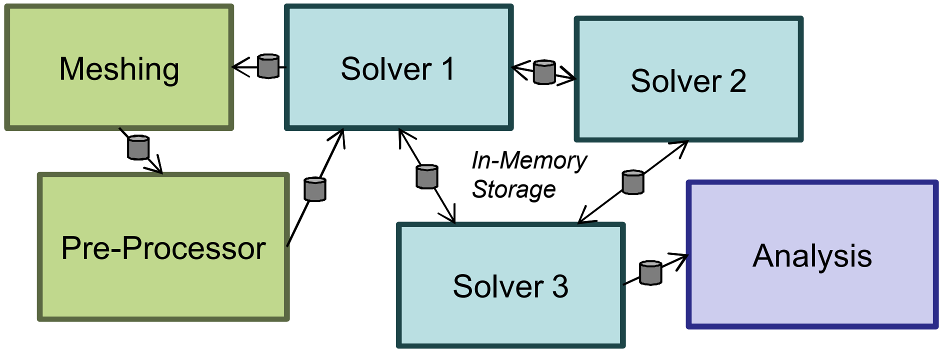
Two Existing Approaches to Integration

Tightly Coupled (In Situ)



- Pros
  - Standard communication (MPI)
  - Supported by HPC runtimes
- Implementation Challenges
  - Configuration/build (lib conflicts)
  - Data structure mismatches
  - Resilience (one fails, they all fail)

Loosely Coupled (In Transit)



- Pros
  - Configuration/build is easy
  - Resilience is easier to manage
- Implementation Challenges
  - Not well supported by runtimes
  - No dynamic scheduling, placement, load balancing, …
  - No standard comm interface

Observations and Projections

- Integration of simulation and analysis is a key to scalability
  - Both *in situ* and in-transit approaches will be important
- Gaps remain before these approaches become “productive”
  - Need portable, fast, memory-efficient mechanisms and interfaces for sharing data
    - POSIX file system is not sufficient
    - Need the right “hooks” into in-memory data structures (avoid copies)
    - Need to deal with data structure mismatches in coupled codes
    - Need to deal with multi-resolution/multi-scale issues
  - Need new definitions for “persistence” of transient data
    - E.g., time windows, data set versioning, …
  - Need new system software that supports integrated workflows
    - Scheduling, load balancing, node and data placement
    - Runtime requirements may differ for coupled components
  - Need resilience…everywhere… nuff said

We've been addressing some of the gaps

- Capabilities for "Integrated" Workflows (Nessie, NNTI – ASC)
  - RPC-based framework for developing data services
  - Portable RDMA abstraction over HPC interconnects (Cray XT/XE, IBM BG, IB)
- Capabilities for data sharing (Kelpie, Sirocco – ASC)
  - Kelpie: In-memory, high-performance key-value store
  - Sirocco: Peer-to-peer like storage system. Supports many media, adaptable and resilient.
- Capabilities for *In situ* Analysis and Visualization (ASC)
  - ParaView/Catalyst (w/Kitware) – focus on modularity, low memory footprint, scalability
  - Dax/VTK-m – Visualization algorithms on advanced architectures
- Resilient integrated workflows (D2T – LDRD)
  - D2T – distributed transaction-based approaches
- OS and Runtime changes to support integrated workflows (ASC and ASCR)
  - Hobbes and Argo – Both ASCR projects
  - Resource management, data sharing, application composition, prog models.