

IO SLEUTHING: ADVENTURES IN BANDWIDTH

ROB LATHAM

Research Software Developer
Argonne National Laboratory

<https://github.com/radix-io/io-sleuthing>

ABOUT ME

- “It’s not Dr”
- Undergraduate at Lehigh University
 - Eternally grateful for the guy down the hall who said “You should try Linux”
- First job at a Linux cluster vendor
 - Big expensive vector supercomputers (e.g. Cray) replaced with cheap commodity racks of Linux nodes.
 - Installed a lot of Argonne-developed software on the machines
- Rob Ross: “hey, we’re looking for developers if you know anyone”
 - I’m still here, and still working for Rob
- Storage and I/O with a focus on application experience
 - Lots of work in MPI-IO, Parallel-NetCDF, tutorials
- 2022 BSSW fellow (which supported this material)



THEMES

- “Do I have an I/O problem?”
 - Performance
- “Where is my I/O problem?”
 - Diagnosis
- “How do I fix my I/O problem?”
 - Tuning
- “I want to learn even more about I/O problems!”
 - Attempting a “living presentation”
 - <https://github.com/radix-io/io-sleuthing>
 - Job scripts, experiments, configurations:
 - experiment yourself: let’s talk about what you find out

“A supercomputer is a device for turning compute-bound problems into I/O-bound problems.”

- Ken Batchner

“There is no physics without I/O.”

– Anonymous Physicist
SciDAC Conference
June 17, 2009

(I think he might have been kidding.)

“Very few large scale applications of practical importance are
NOT data intensive.”

– Alok Choudhary, IESP, Kobe Japan, April 2012

(I know for sure he was not kidding.)

TODAY'S OUTLINE

- Part One
 - Challenges
 - File systems
 - Benchmarking
 - Demo: Lustre + striping
 - Darshan
- Part Two
 - MPI-IO
 - High-level I/O libraries
 - Parallel-NetCDF
 - HDF5
 - Machine Learning workloads

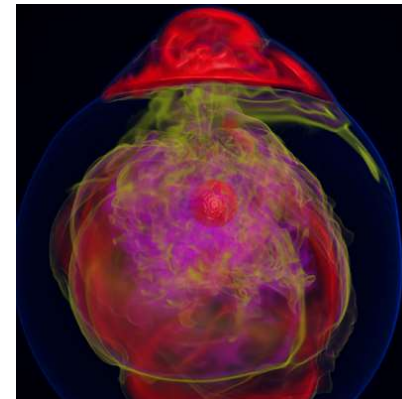
STAGE SETTING: CONCEPTS, TERMS, ETC

COMPUTATIONAL SCIENCE

- Computer simulation as a tool promotes greater understanding of the real world
 - Complements experimentation and theory
- Problems are increasingly computationally expensive
 - Large parallel machines are needed to perform calculations
 - Leveraging parallelism in all phases is critical
- Data access is a huge challenge and includes
 - Using parallelism to obtain performance
 - Finding usable, efficient, and portable interfaces
 - Understanding and tuning I/O



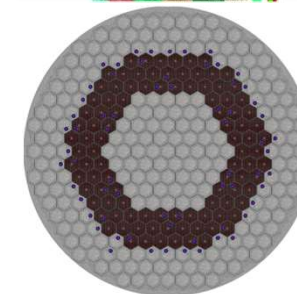
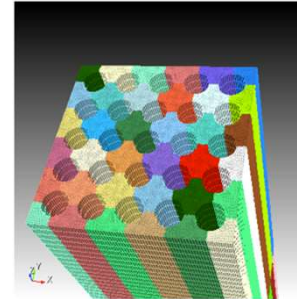
HPE Polaris system at Argonne National Laboratory.



Visualization of entropy in Terascale Supernova Initiative application. Image from Kwan-Liu Ma's visualization team at UC Davis.

APPLICATION DATASET COMPLEXITY VS. I/O

- I/O systems have very simple data models
 - Tree-based hierarchy of containers
 - Some containers have streams of bytes (files)
 - Others hold collections of other containers (directories or folders)
- Applications have data models appropriate to domain
 - Multidimensional typed arrays, images composed of scan lines, records of variable length
 - Headers, attributes on data
- Someone has to map from one to the other!



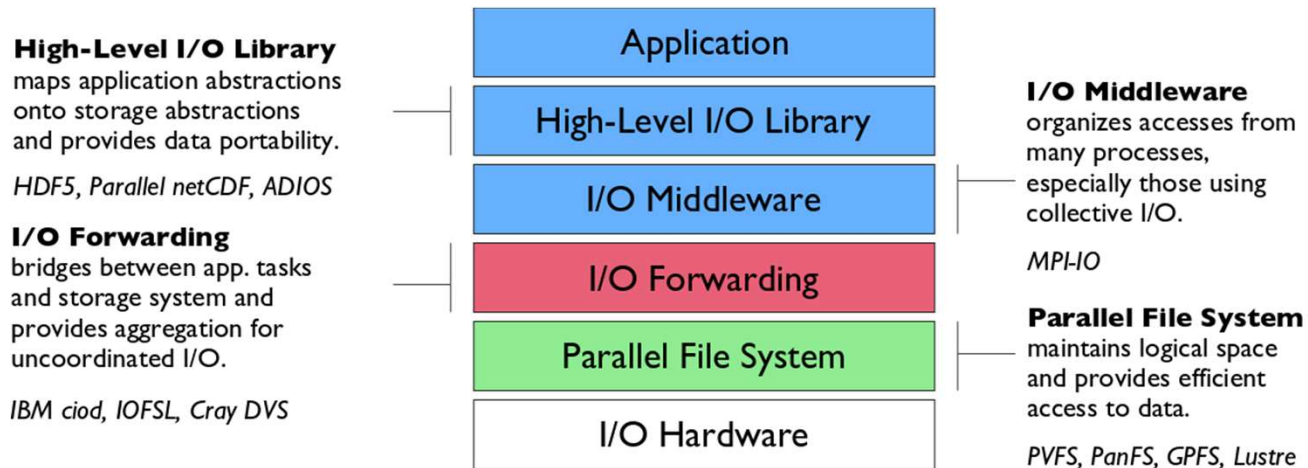
Model complexity:
Spectral element mesh (top)
for thermal hydraulics
computation coupled with
finite element mesh (bottom)
for neutronics calculation



Scale complexity:
Spatial range from the
reactor core, in
meters, to fuel pellets,
in millimeters

Images from T. Tautges (Argonne) (upper left), M. Smith (Argonne) (lower left), and K. Smith (MIT) (right).

I/O FOR COMPUTATIONAL SCIENCE



Additional I/O software provides improved performance and usability over accessing the parallel file system directly. Reduces or (ideally) eliminates need for optimization in application codes.