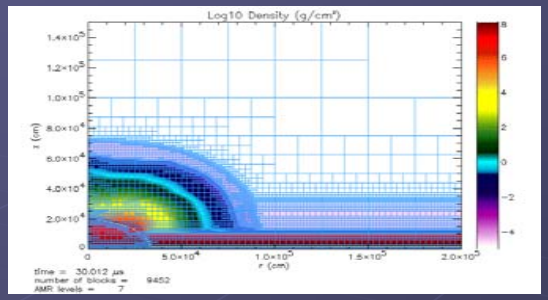
I/O and the DOE Office of Science

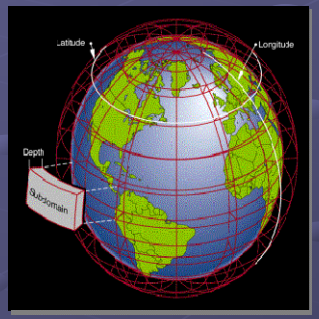
Rob Ross
Mathematics and Computer Science Division
Argonne National Laboratory

Applications

- Simulation (astrophysics, climate, etc.)
  - Input dataset I/O
  - Checkpointing
  - Visualization
- Searching (biology, experimental physics)
  - Growing needs here
  - Less traditional interfaces



Graphic from A. Siegel, ANL

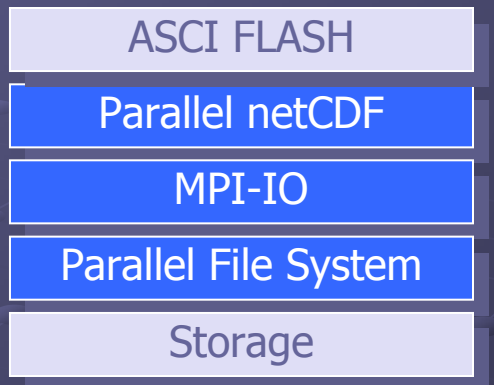


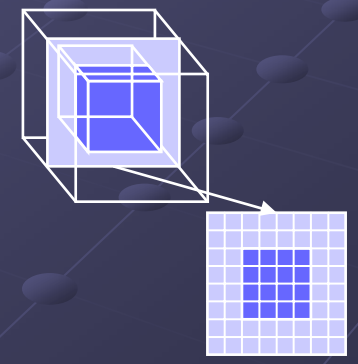
Graphic from J. Tannahill, LLNL

Application	Reading and Generation	Post-processing, Checkpointing	Analysis
Astrophysics	20-200	20-200	20
Supernova	20	2	2
Climate Modeling	2	2	1
Cosmology	5	1	1
Fusion	1,000	1	0.5

Example: ASCI/Alliance FLASH

- FLASH is an astrophysics simulation code from the ASCI/Alliance Center for Astrophysical Thermonuclear Flashes
  - Fluid dynamics code using adaptive mesh refinement (AMR)
  - Runs on systems with thousands of nodes
- Three layers of I/O software between the application and the I/O hardware
  1) Processes write regions of variables using PnetCDF
  2) PnetCDF converts data to portable format and calls appropriate MPI-IO collectives
  3) MPI-IO optimizes writing of data to PFS using whatever interface is available
  4) PFS handles moving and storing data and maintaining file metadata





- Ghost cell
- Element (24 vars)

3D FLASH Block

What Constitutes "Effective" I/O?

- Providing performance is only one piece!
- Three metrics on which we measure success:
  - Usability – How well I/O interfaces map to application data models and access patterns
    - Solutions are unique to HPC
  - Performance and scalability – How well our I/O systems are tuned for common application patterns (e.g. concurrent access, noncontiguous access) and metadata access
  - Reliability and management – How much maintenance our parallel I/O systems require, how well they handle failures

Current Research Efforts

- Scientific Data Management SciDAC
  - ANL, LBNL, ORNL, LLNL, NCSU, NWU, SDSC, others
- PVFS2 file system and ROMIO MPI-IO implementation
  - ANL, Clemson, OSC, OSU, NWU
- Lustre
  - Cluster File Systems, LLNL, PNNL, others?
- LWFS
  - Sandia, UNM

Perceived Needs

- Scaling to ever larger systems
  - Improved caching, read-ahead, write-behind
  - Better collective I/O and data layout
  - Enhanced interfaces to file systems
  - New approaches to name space and metadata management
- Functionality to match new application domains
  - Filtering/processing within the storage system (active storage)
  - Efficient search/query capabilities
- Resiliency and easy management
  - Autonomic storage
  - Enhanced redundancy (while maintaining performance)

Short-Term Directions

- Interfaces
  - POSIX I/O Extensions for HPC
  - NFSv4 and pNFS
- Communication and Intelligence in the I/O system
  - Infrastructure for developing active and autonomic storage
  - Migration, virtualization
- Caching
  - Leveraging other system components (e.g. interconnects, MPI)
  - Where does caching belong?
- Benchmark development
  - Revisit the question of “what do applications do?”
  - Create simulations for use in system R&D
  - HPC I/O Challenge Benchmark?

Long-Term Directions

- Autonomic storage
  - Mechanisms and policies
  - Leveraging vast redundancy scheme work
- Active, special-purpose storage
  - Targeting specific application domains
  - Leveraging high-level library work (for usability)
- New storage organizations
  - Sub-files, (true) object storage, alternatives to tree-based namespaces, …
  - Integration with archival storage