Scientific data management on exa-scale systems: 2020 vision?

Carlos Maltzahn

Trends in High-Performance Distributed Computing
Vrije Universiteit, March 14, 2012

Who am I?

- Systems Research Lab (SRL), **UC Santa Cruz**
- LANL/UCSC Institute for Scalable Scientific Data Management
- Ultrascale Systems Reserch Center (USRC), NM Consortium
- **cs.ucsc.edu/~carlosm**

- **Current Research**
  - **Storage Systems**
    - Scientific Data Management
    - High-performance exa-scale storage
    - Data Management Games
  - **Performance Management**
    - End-to-end storage QoS
    - Efficient Simulation of Large-scale File Systems
- Past Research
  - Enterprise-level Web Proxies
  - Dynamic Workflow Systems
  - Collaborative Help Systems

What is ***not*** going to change

Digital Data will continue to

- grow **exponentially**
- require **active** protection
- outgrow **read speed** of archival storage media
- consume a lot of **power**
- be stored in **byte streams**
- be hard to move or **convert**

What ***is*** going to change

Data access will rely on

- data **structure**
  - Scientific data is already structured
    - HDF5, NetCDF4, climate community's Data Reference Syntax and Controlled Vocabularies
  - Parsing overhead of unstructured data is unaffordable
    - Apache Avro, Binary XML, ProtocolBuffers, Multimedia, ...

What ***is*** going to change

Data access will rely on

- data **structure**
  - Scientific data is already structured
    - HDF5, NetCDF4, climate community's Data Reference Syntax and Controlled Vocabularies
  - Parsing overhead of unstructured data is unaffordable
    - Apache Avro, Binary XML, ProtocolBuffers, Multimedia, ...
- performance **guarantees**
  - Applications do have utilization needs and deadlines: specify them!

What ***is*** going to change

Data access will rely on

- data **structure**
  - Scientific data is already structured
    - HDF5, NetCDF4, climate community's Data Reference Syntax and Controlled Vocabularies
  - Parsing overhead of unstructured data is unaffordable
    - Apache Avro, Binary XML, ProtocolBuffers, Multimedia, ...
- performance **guarantees**
  - Applications do have utilization needs and deadlines: specify them!
- well-known **data models**
  - Enables automatic access optimization
  - Minimizes data movement (due to shared model)

What ***is*** going to change

Data access will rely on

- data **structure**
  - Scientific data is already structured
    - HDF5, NetCDF4, climate community's Data Reference Syntax and Controlled Vocabularies
  - Parsing overhead of unstructured data is unaffordable
    - Apache Avro, Binary XML, ProtocolBuffers, Multimedia, ...
- performance **guarantees**
  - Applications do have utilization needs and deadlines: specify them!
- well-known **data models**
  - Enables automatic access optimization
  - Minimizes data movement (due to shared model)
- **automatic** access optimization
  - Allows declarative querying, updates
  - No need to re-invent optimization for each application

What ***is*** going to change

Data access will rely on

- data **structure**
- performance **guarantees**
- well-known **data models**
- **automatic** access optimization

What ***is*** going to change

Data access will rely on

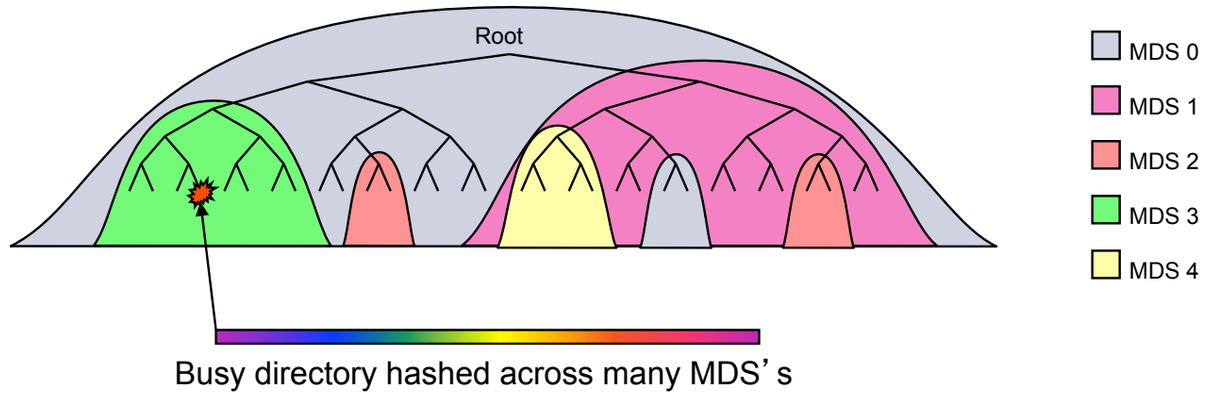
- data **structure**
- performance **guarantees**
- well-known **data models**
- **automatic** access optimization

The database community has thought about this.

Today: The Metadata Bottleneck

- Metadata management is hard to scale
  - hierarchical dependencies
  - distributed rename(), relink()
  - small, but lots of objects and transactions
    - critical to overall system performance
    - skewed workloads, hot spots
- Single-host solutions become bottleneck
- Distributed solutions do exist:

Ceph: Dynamic Subtree Partitioning



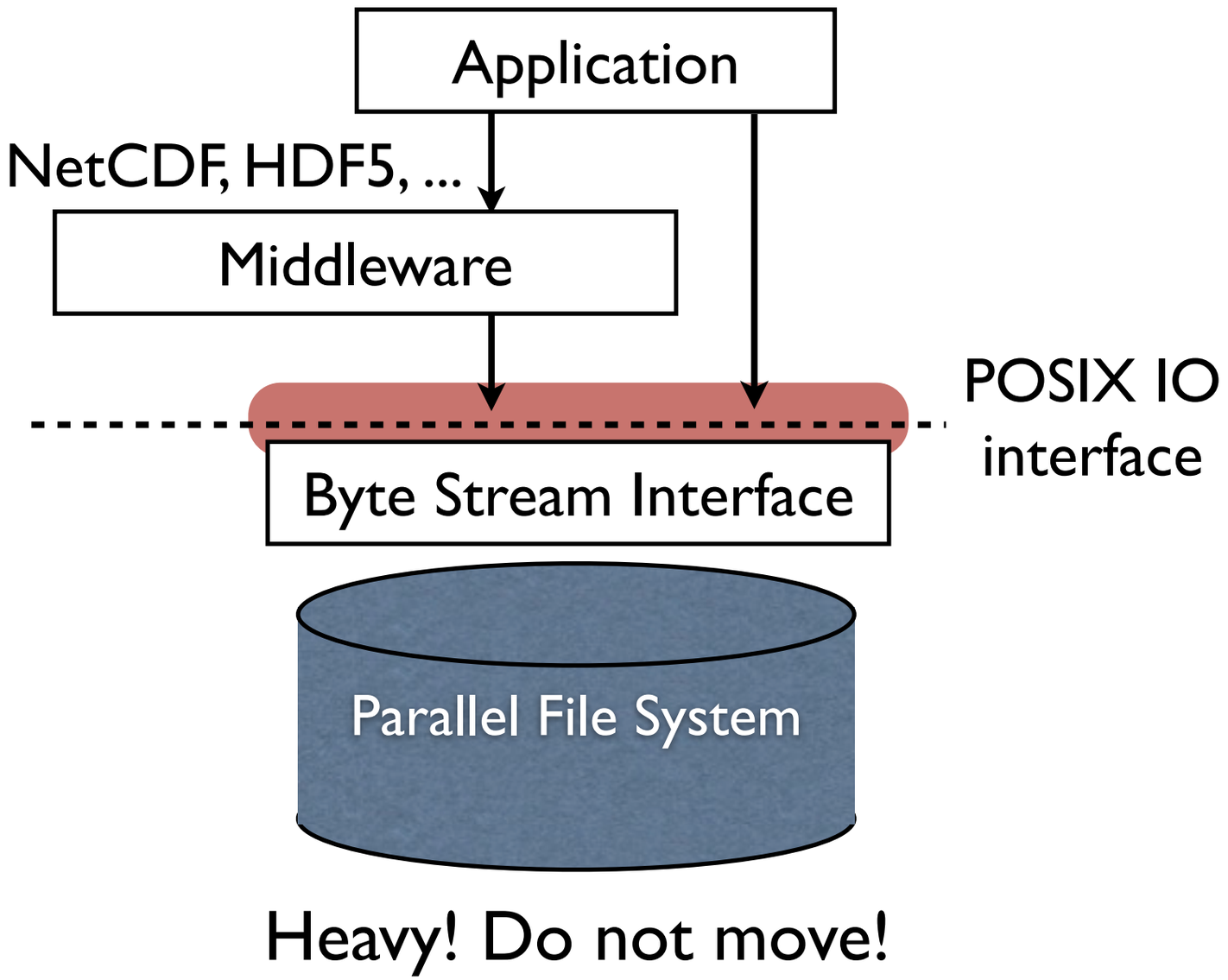
http://ceph.newdream.net

S. A. Weil, K. T. Pollack, S. A. Brandt, and E. L. Miller. Dynamic metadata management for petabyte-scale file systems. In **SC'04**, Pittsburgh, PA, Nov. 2004. ACM.

S. A. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn. Ceph: A scalable, high-performance distributed file system. In **OSDI'06**, Seattle, WA, Nov. 2006.

Today: The POSIX I/O Bottleneck

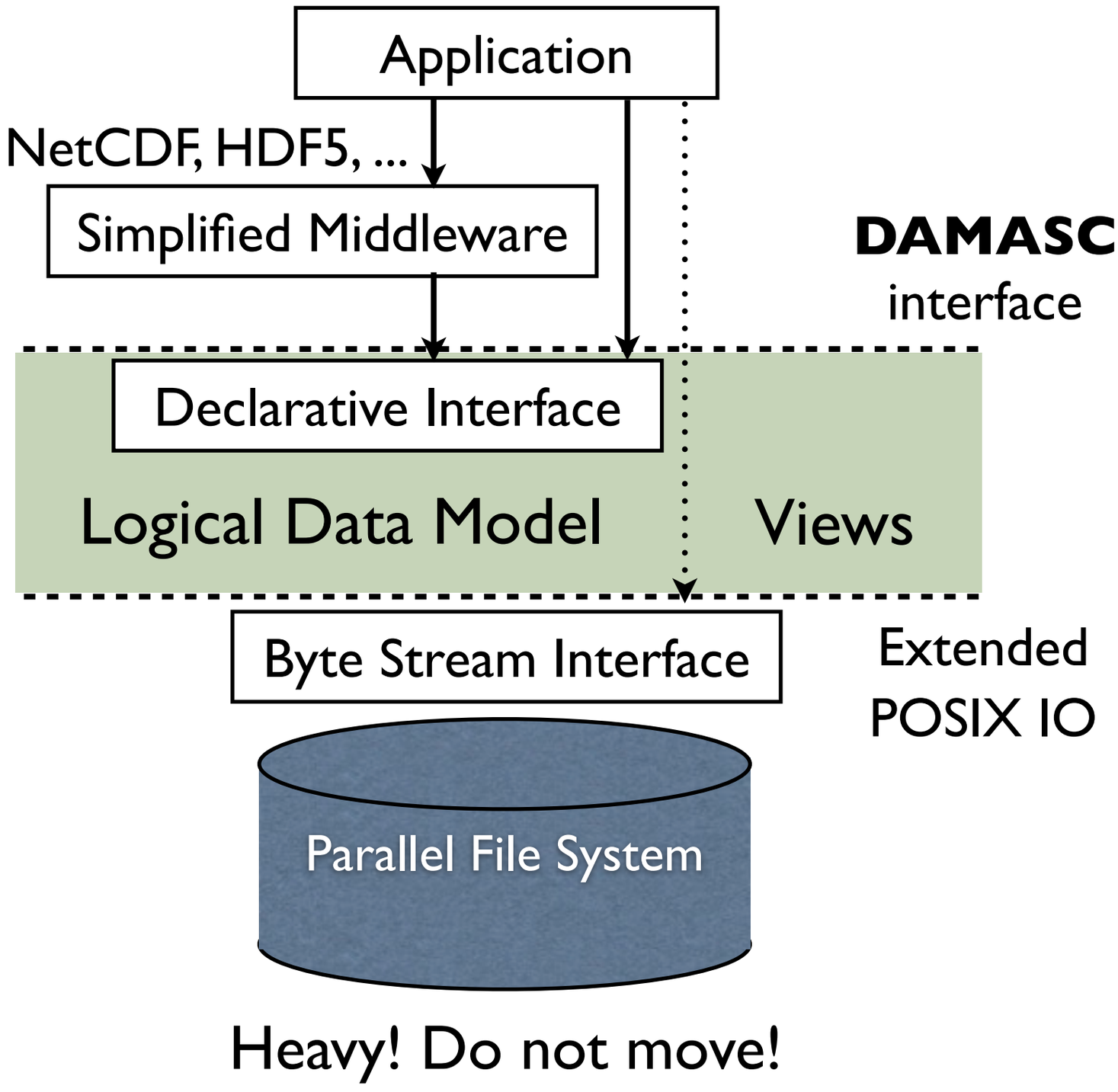
- **POSIX IO** dominates
  File system interfaces
- POSIX IO does not scale
  - **50 years ago**: 100MB
  - **Now**: 100PB (x 1 billion)
- Performance price of POSIX IO is high
  - Workload- & system-specific interposition layers (e.g. PLFS): almost **100 x speed-up**
- Common Workaround
  - **Middleware** tries to make up for limitations
  - Still uses POSIX!



Heavy! Do not move!

DAMASC: DAta MAnagement in Scientific Computing

- Fuse data management services with parallel file systems
  - Declarative **querying**
  - **Views**
  - Automatic content **indexing**
  - **Provenance** tracking
- **Index**, not ingest!
- **Data processing** on storage nodes



S. A. Brandt, C. Maltzahn, N. Polyzotis, and W.-C. Tan. Fusing data management services with file systems. In **PDSW'09**, Portland, OR, November 15 2009

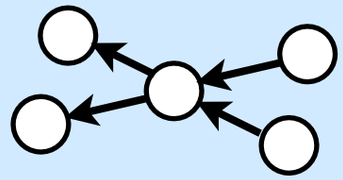
DAMASC: SciHadoop

- All access via **scientific access library** (e.g. NetCDF)
- Task manager **partitions** logical space
  - instantiates mappers and reducers for logical partition
  - **places** mappers and reducers based on logical and physical relationships
- Benefits of structure-awareness
  - reduces **data transfers**
  - reduces **remote reads**
  - reduces **unnecessary reads**

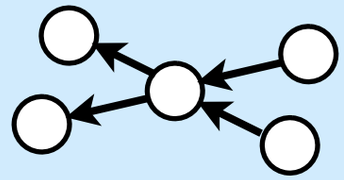
J. Buck, N. Watkins, J. LeFevre, K. Ioannidou, C. Maltzahn, N. Polyzotis, and S. A. Brandt. Scihadoop: Array-based query processing in hadoop. In **SC '11**, Seattle, WA, November 2011.

Exa-scale Stack: Today

Compute Cluster



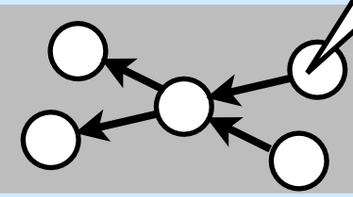
Visualization Cluster



Access Execution Plan

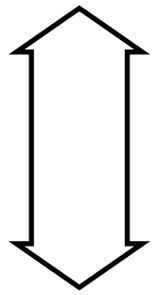
I/O Nodes

High-level I/O Libs, I/O Middleware, I/O Forwarding



POSIX IO

Parallel File System Clients



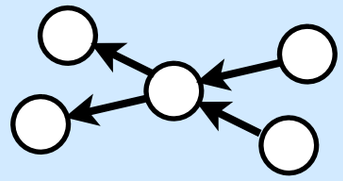
Parallel File System Servers

Storage Cluster

Single Metadata Node

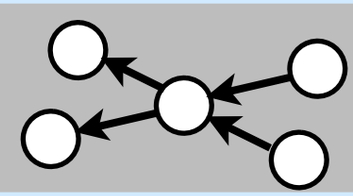
Exa-scale Stack: 2015

Compute Cluster (with *in-situ* Analysis)



I/O Nodes

High-level I/O Libs, I/O Middleware, I/O Forwarding

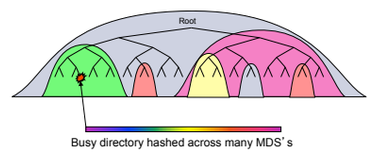


Interposition Layer (PLFS)

Burst Buffer

Parallel File System Clients

Byte Stream Data



Parallel File System Servers

Storage Cluster

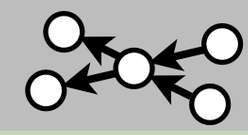
Metadata Nodes

Parallel File Systems: 2018

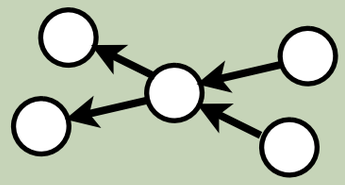
Compute Cluster (with *in-situ* Analysis)

I/O Nodes

High-level I/O Libs



Declarative Query and Update

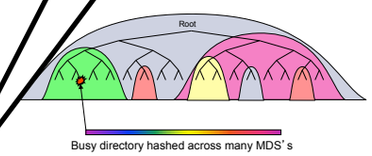


Parallel File System Clients

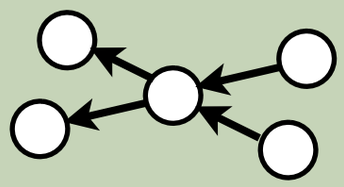
Burst Buffer

Structured Data

Multiple Data Models



Parallel File System Servers



Storage Cluster

Byte streams

Metadata Nodes

Acknowledgements

UCSC: Joe Buck, Noah Watkins, Jeff LeFevre, Kleoni Ioannidou, Alkis Polyzotis, Sott Brandt, Wang-Chiew Tan

LANL: John Bent, Gary Grider, Meghan Wingate, James Nunez, Carolyn Connor, Lucho Ionkov, Mike Lang, Jim Ahrens

LLNL: Maya Gokhale, Celeste Matarazzo, Sasha Ames

UCAR/Unidata: Russ Rew

Funding: DOE/ASCR: DE-SC0005428, NSF: 1018914, ISSDM

Thank you!

systems.soe.ucsc.edu