A RENAISSANCE FOR DATA MANAGEMENT IN HPC?

ROB ROSS Mathematics and Computer Science Division
Argonne National Laboratory
rross@mcs.anl.gov

29th International Conference on Scientific and Statistical Database Management, June 27, 2017

Renaissance – a revival of or renewed interest in something

Trilab SGPFS Requirements

3/07/00

Abstract

The following is intended to serve as guidance for the SGPFS PathForward initiative. It describes ASCI Trilab file system requirements, in particular we focus on the special requirements of ASCI-scale systems. The usual requirements of any file system remain, generally, in place. For example, requirements such as persistence, and stability will be assumed. Beyond that, due to the nature of the machines served by the file system, there are some "usual" requirements with a new or different twist as well as some that are unusual. These requirements are, apparently, outside what the industry has in sight. All requirements are prioritized as either Mandatory, Highly Desired, or Desired.

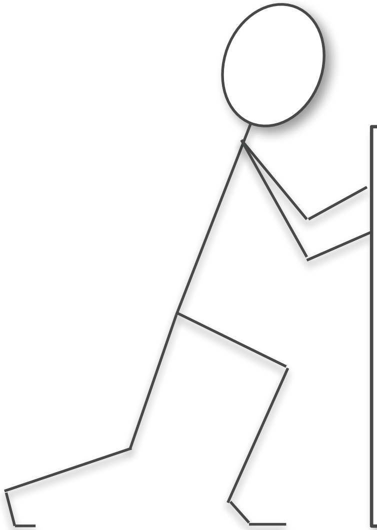
Thanks to G. Grider for digging this up!

Trilab SGPFS Requirements

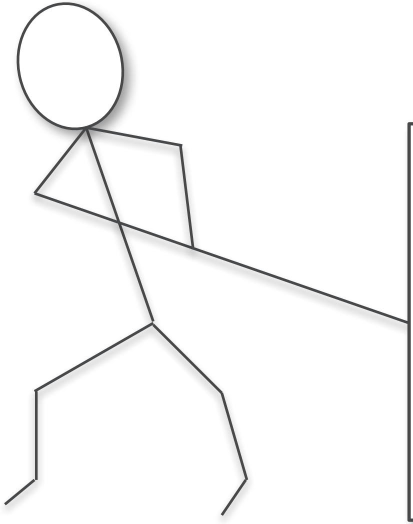
- 2000s and early 2010s, **focus was on the POSIX file system model**
- GPFS and Lustre dominate HPC deployments
- Intellectual **dark age** for data management in HPC:
  - **Philosophy** of maximizing compute
  - **Workload** focus on simulation checkpoint/restart
  - **Architectural model** fixed: PFS on storage nodes with disks
  - (Much) Research focused on **mitigating deficiencies**
  - Novel research got little traction

Thanks to G. Grider for digging this up!

PUSH AND PULL DRIVING CHANGE IN HPC



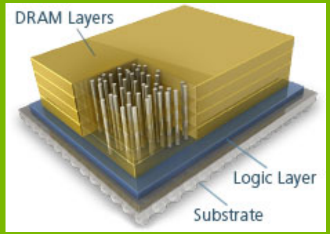
- Hardware technology
- Cloud and enterprise computing software



- New science applications
- New roles for HPC

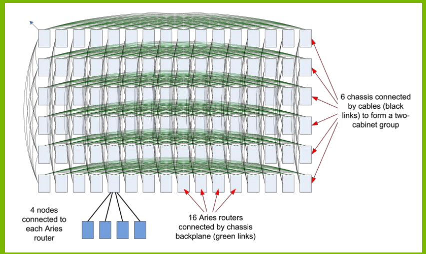
Also, **data as a first class citizen** as a guiding philosophy.

PUSH AND PULL DRIVING CHANGE IN HPC



Micron Hybrid Memory Cube

Cray Aries network (dragonfly)



Metagenomics
High throughput analysis of biological function from DNA sequence data

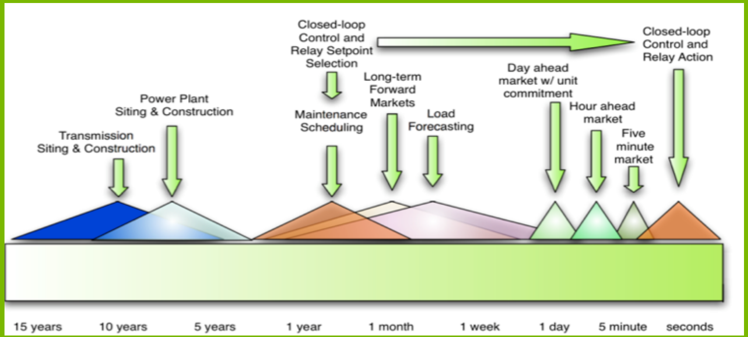
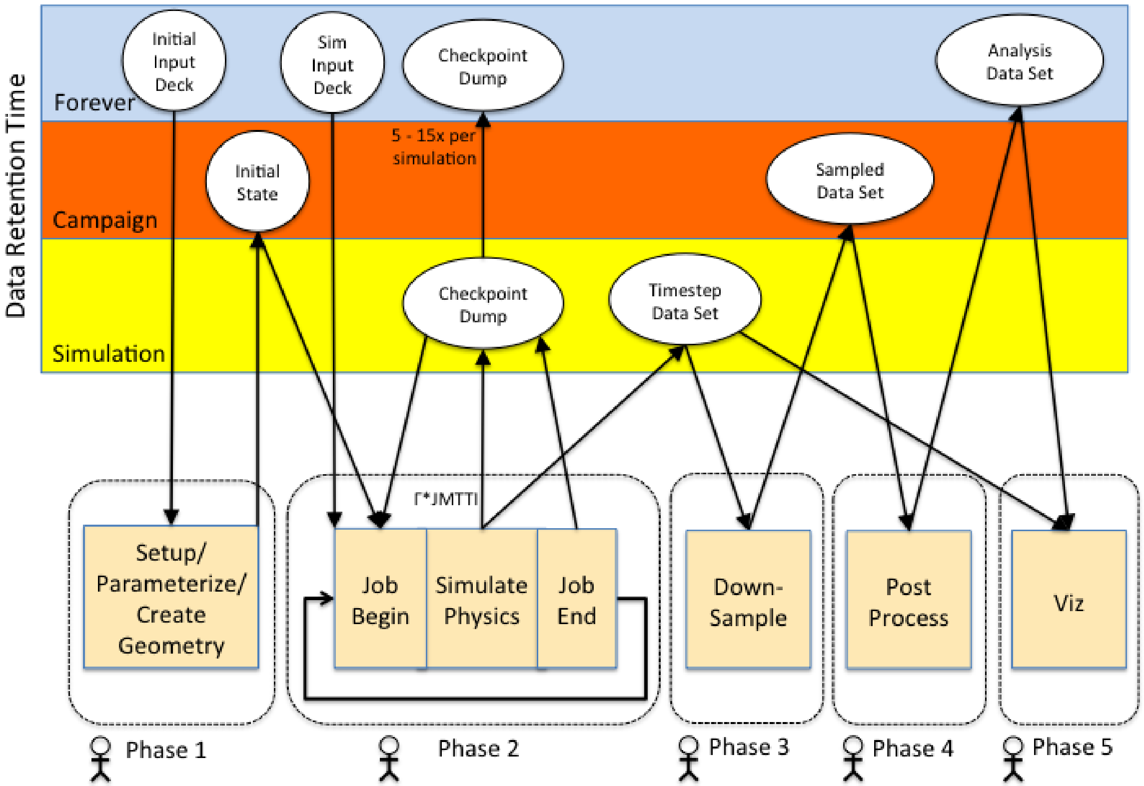


Image courtesy of Chris de Marco (U-Wisconsin).

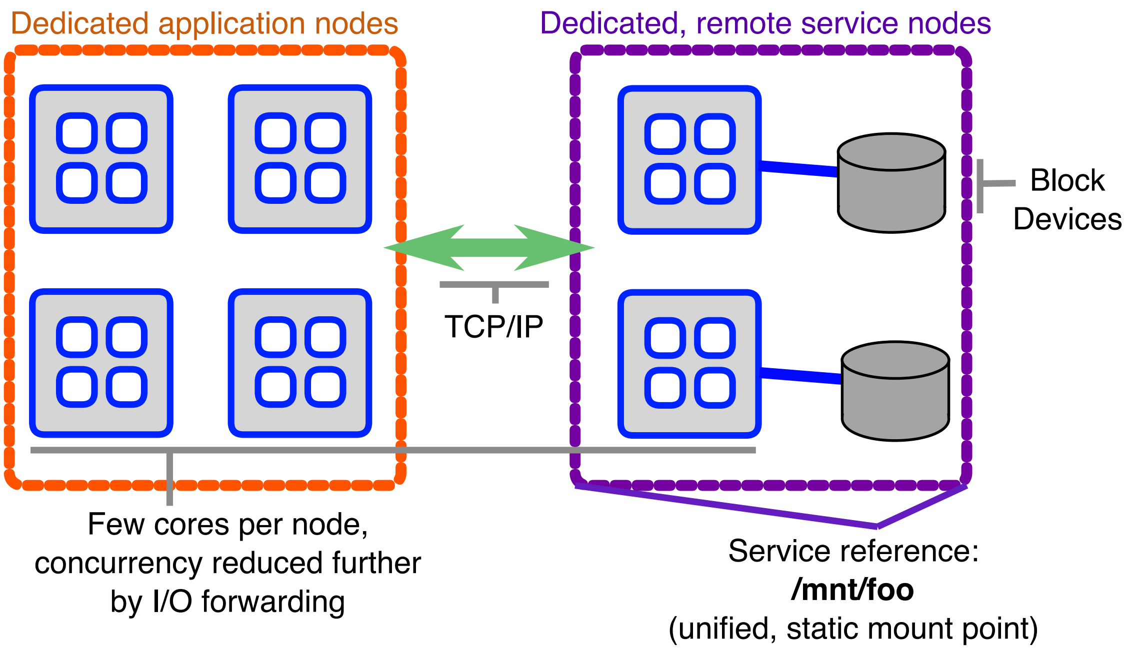
Also, **data as a first class citizen** as a guiding philosophy.

DATA AS A FIRST CLASS CITIZEN



From “APEX Workflows”, LANL, NERSC, SNL, SAND2015-10342 O LA-UR-15-29113, Nov. 24, 2015.

TRADITIONAL DATA SERVICES: IMPLEMENTATION

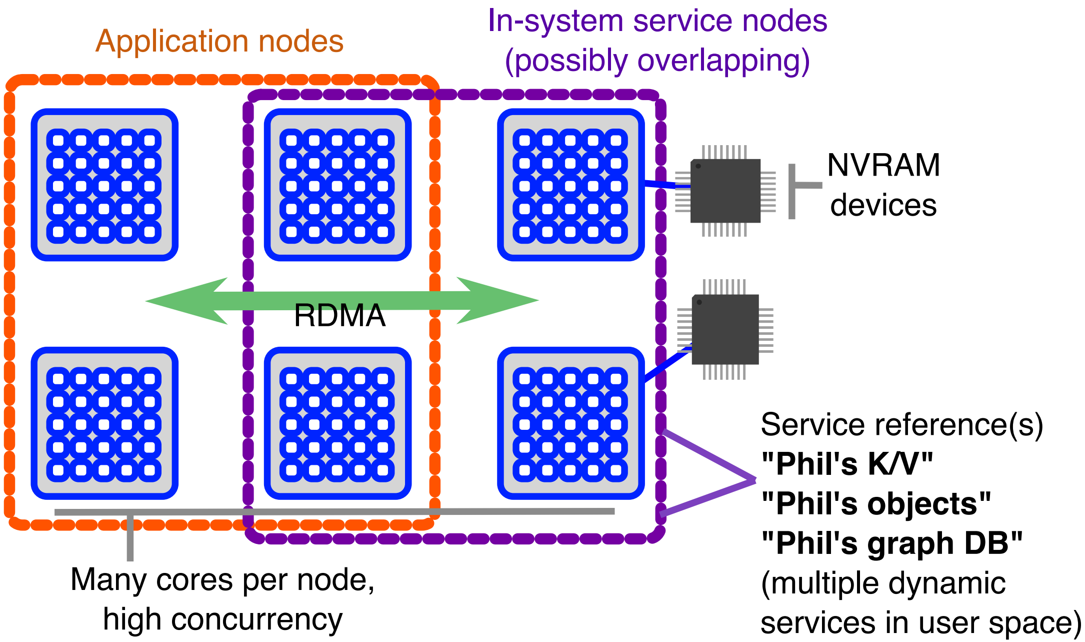


- Has a lot in common with scalable Internet services
- Key technologies: *block devices, sockets, pthreads, kernel drivers*
- Operations take ***milliseconds*** to complete
- Checkpoint/restart workload

Figure courtesy P. Carns.

MODERN DATA SERVICES

Dramatically different deployment environment.



- Key technologies: *NVRAM, RDMA, dynamic services, higher concurrency*
- Latency and jitter are more apparent now than ever
- Many deployment modes
- Dynamic service organization
- Diverse workload

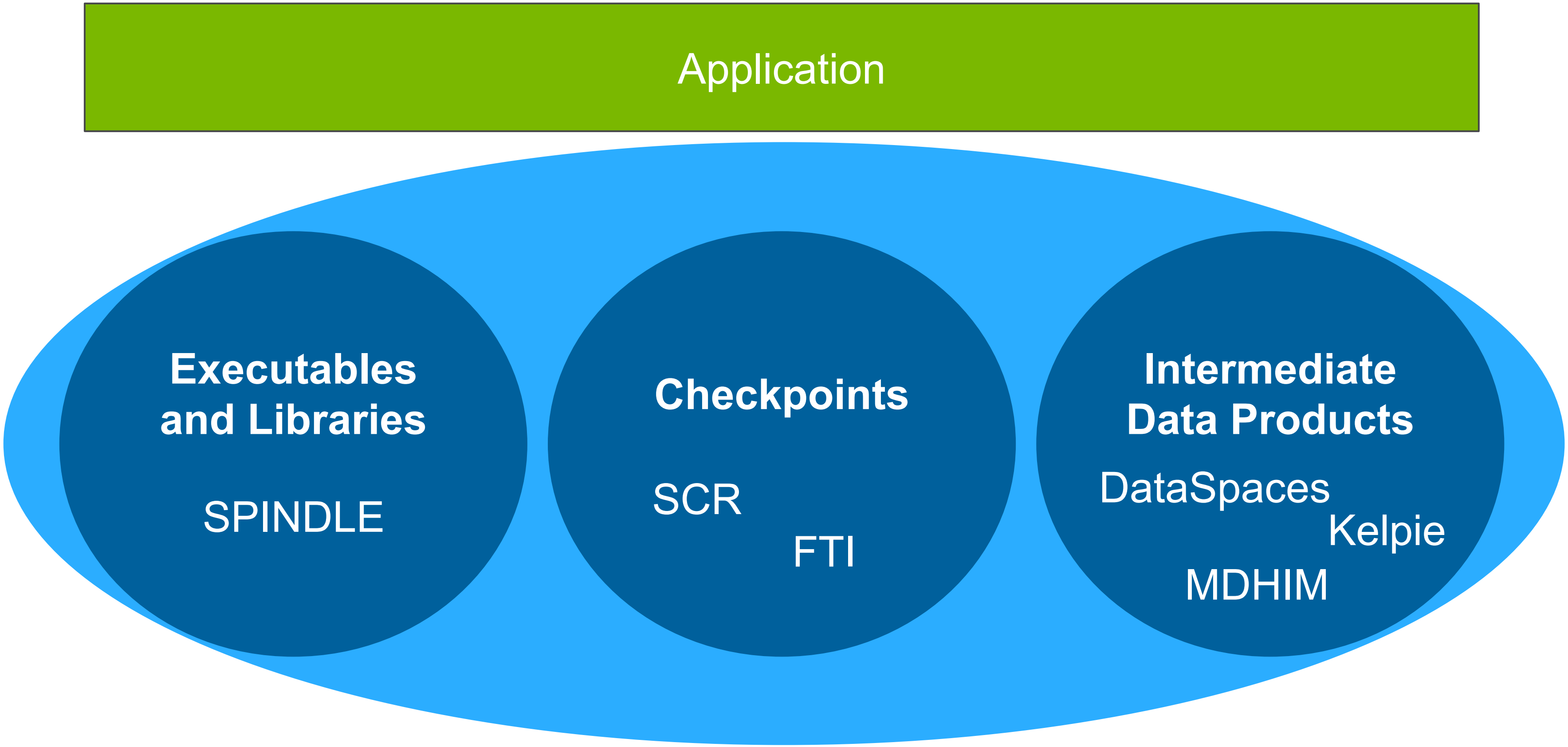
Figure courtesy P. Carns.

RENAISSANCE IN HPC DATA SERVICES?

- Technology has forced a rethink of many service implementations
- Aggregate workload not well suited by any one data abstraction
- Cloud and enterprise have shown that many data services can coexist and be composed to solve important problems

SPECIALIZATION OF DATA SERVICES

SPECIALIZATION OF DATA SERVICES



MANAGING EXECUTABLES AND LIBRARIES

Dynamic libraries are a clean class of data to treat separately.

- Characteristics:
  – Can assume data doesn't change during runtime
  – High degree of sharing across application processes
  – No need for redundancy in service (original stored elsewhere)
- Opportunities:
  – Dramatic reduction in parallel file system traffic
  – Stripping of libraries on load
  – Pre-staging of data (with scheduler integration)
- SPINDLE is a great example of how to manage this data.

Frings, Wolfgang, et al. "Massively parallel loading."
ICS 2013, June 2013.

MANAGING CHECKPOINTS

- Characteristics
  - Typically bulk synchronous
  - Write once, often not read
- Opportunities
  - Latency hiding
  - **Leveraging multiple layers of storage**
  - **Adjusting rate/placement to match fault rates**
- Fault Tolerant Interface (FTI)
  - Simple “snapshot” abstraction
  - Manages all the layers for the user

Local Storage:
SSD, NVM

Partner Copy:
Ckpt. Replication

RS Encoding:
Ckpt. Encoding

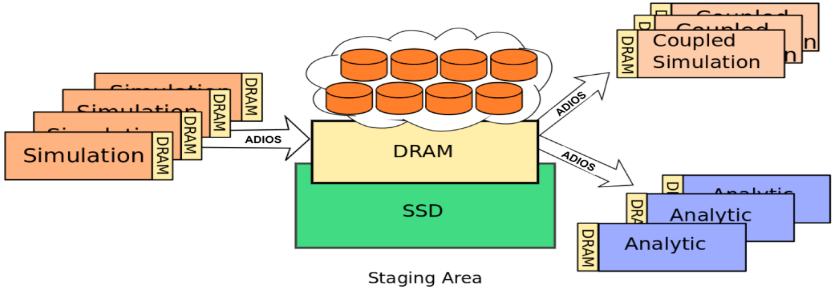
File System:
Classic Ckpt.

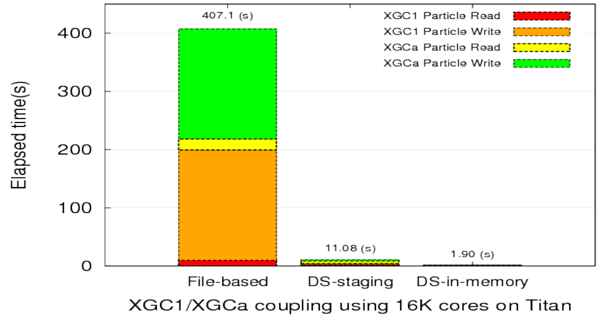
```
int main(int argc, char **argv) {

   MPI_Init(&argc, &argv);
   FTI_Init(“conf.fti”,
      MPI_COMM_WORLD);

   double *grid;
   int i, steps=500, size=10000;
   initialize(grid);
   FTI_Protect(0, &i,   1,
FTI_INTG);
   FTI_Protect(1, grid,
size,FTI_DFLT);

   for (i=0; i<steps; i++) {
      FTI_Snapshot();
      kernel1(grid);
      kernel2(grid);
      comms(FTI_COMM_WORLD);
   }

   FTI_Finalize();
   MPI_Finalize();
   return 0;
}
```

L. Bautista-Gomez et al. "FTI: high performance fault tolerance interface for hybrid systems." SC 2011. November 2011.
S. Di et al. "Optimization of multi-level checkpoint model for large scale HPC applications." IPDPS 2014. 2014.

MANAGING INTERMEDIATE DATA PRODUCTS

- Characteristics:
  - Data leaves application but not the system
  - Variety of different data abstractions
  - Producer-consumer model is common
- Opportunities:
  - Exploiting locality
  - Avoiding data movement off system
  - More efficient synchronization

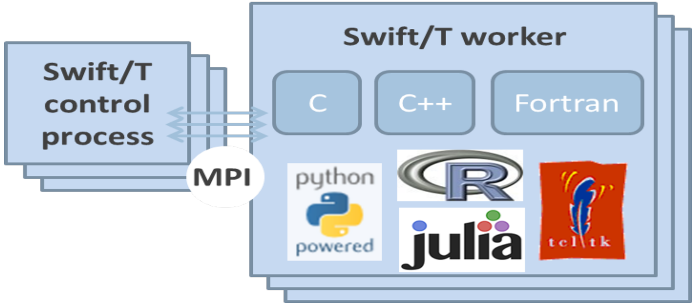




Impact of coupling via ADIOS/DataSpaces on XGC1/XGCa fusion application.
Material from S. Klasky (ORNL).

C. Docan et al. "DataSpaces: an interaction and coordination framework for coupled simulation workflows." *Cluster Computing* 15.2 (2012).
C. Ulmer. “Leveraging In-Memory Key/Value Stores in HPC: Kelpie.” Salishan 2013, April 2013.

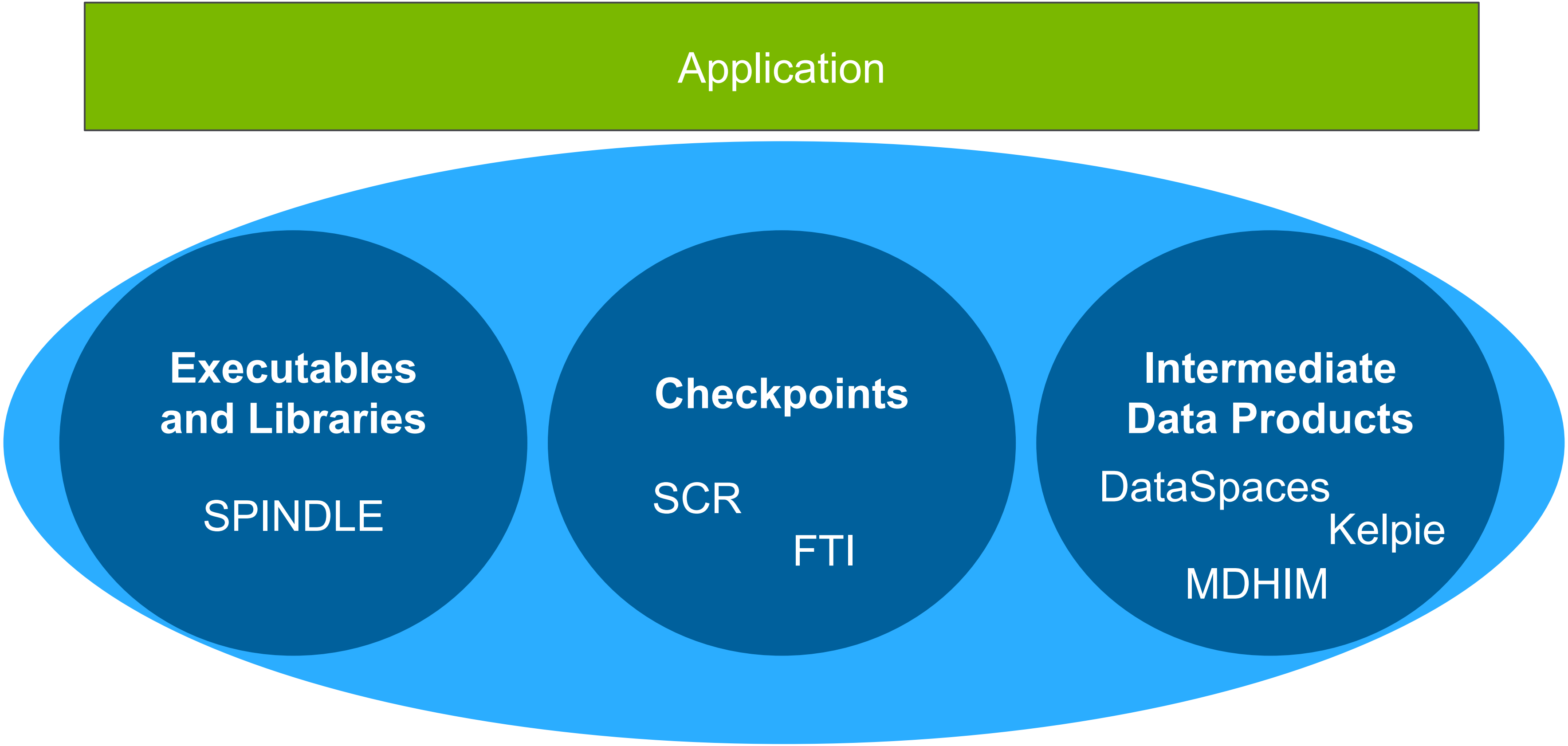
SPECIALIZATION FOR MANY-TASK WORKFLOW



- Swift script controls execution – generates an ADLB program (see below)
  - Tasks can be basically anything (e.g., MPI code)
  - Data dependencies are emitted as run proceeds
- Asynchronous Dynamic Load Balancer (ADLB) manages data and work
  - Distributed, data-dependent work queue
  - Work units have (optional) priorities, types, and **locality constraints**
  - Enables heuristic, coarse-grained data-aware scheduling, mixing user control and automatic decisions
- Applied in materials science, power grid, etc.
  - E.g., transforming TBs of X-ray data from the Advanced Photon Sources, streaming to compute nodes at 100 GB/s

J. Wozniak et al. "Swift/T: large-scale application composition via distributed-memory dataflow processing. CCGrid 2013.
E. Lusk et al. “More Scalability, less pain: a simple programming model and its implementation...” SciDAC Review, 2010.
F. Duro et al. “Flexible data-aware scheduling for workflows over an in-memory object store”. CCGrid 2016.

SPECIALIZATION OF DATA SERVICES



ACCELERATING DATA SERVICE DEVELOPMENT

ROB ROSS, PHILIP CARNS, MATTHIEU DORIER, KEVIN HARMS, ROB LATHAM, AND SHANE SNYDER — Argonne National Laboratory

GARTH GIBSON, GEORGE AMVROSIADIS, CHUCK CRANOR, SAURABH KADEKODI, AND QING ZHENG — Carnegie Mellon University

JEROME SOUMAGNE — The HDF Group

GALEN SHIPMAN, DAVID RICH, AND BRAD SETTLEMYER — Los Alamos National Laboratory

WHAT GOES INTO A DATA SERVICE?

	Provisioning	Comm.	Local Storage	Fault Mgmt. and Group Membership	Security
ADLB *Data store and pub/sub.*	MPI ranks	MPI	RAM	N/A	N/A
DataSpaces *Data store and pub/sub.*	Indep. job	Dart	RAM (SSD)	Under devel.	N/A
DataWarp *Burst Buffer mgmt.*	Admin./ sched.	DVS/ lnet	XFS, SSD	Ext. monitor	Kernel, lnet
FTI *Checkpoint/restart mgmt.*	MPI ranks	MPI	RAM, SSD	N/A	N/A
Kelpie *Dist. in-mem. key/val store*	MPI ranks	Nessie	RAM (Object)	N/A	Obfusc. IDs
SPINDLE *Exec. and library mgmt.*	Launch MON	TCP	RAMdisk	N/A	Shared secret

OUR GOAL

Enable composition of data services for DOE science and systems

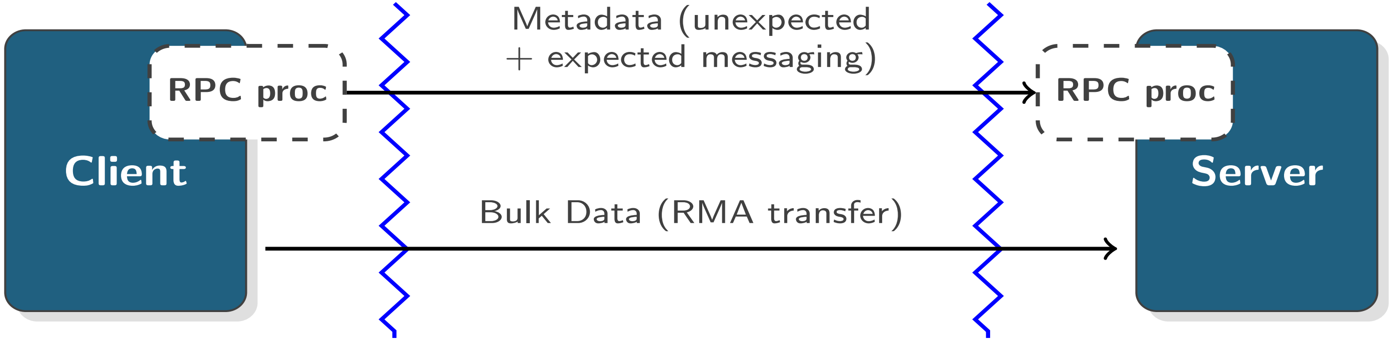
- Application-driven
  - Identify and match to science needs
  - Traditional data roles (e.g., checkpoint, data migration)
  - New roles (e.g., equation of state/opacity databases)
- Composition
  - Develop/adapt building blocks
    - Communication
    - Concurrency
    - Local Storage
    - Resilience
    - Authentication/Authorization
  - Enable rapid development of specialized services

COMMUNICATION: MERCURY

https://mercury-hpc.github.io/

Mercury is an RPC system for use in the development of high performance system services. Development is driven by the HDF5 Group with Argonne participation.

- Portable across systems and network technologies
- Builds on lessons learned from IOFSL, Nessie, lnet, and others
- Efficient bulk data movement to complement control messages



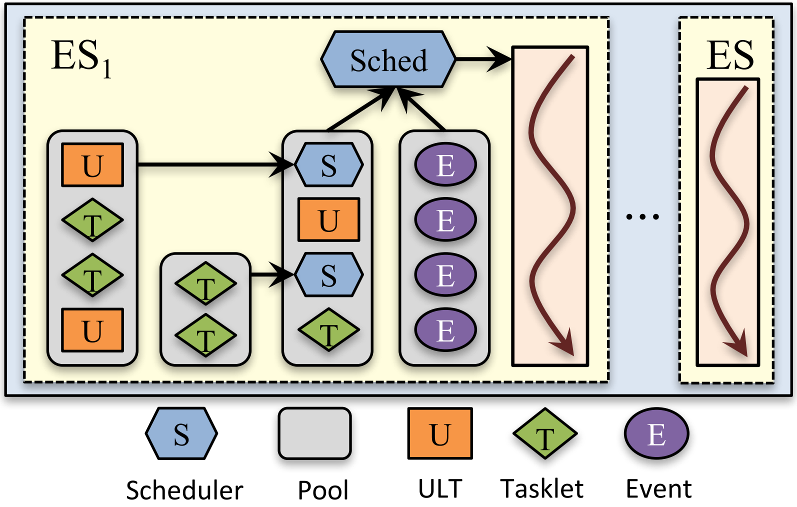
CONCURRENCY: ARGOBOTS

https://collab.cels.anl.gov/display/argobots/

Argobots is a lightweight threading/tasking framework.

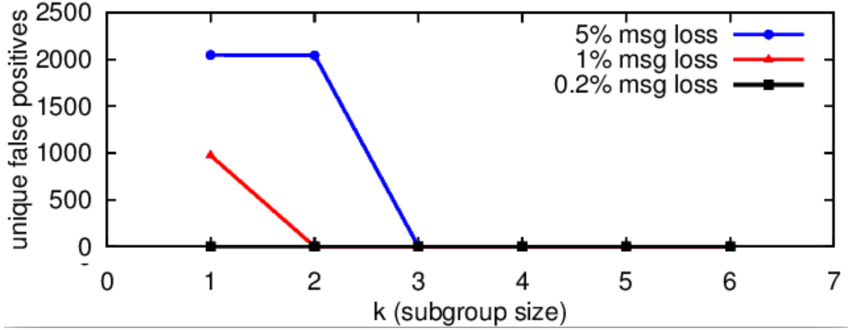
- Features relevant to I/O services:
  - Flexible mapping of work to hardware resources
  - Ability to delegate service work with fine granularity across those resources
  - Modular scheduling
- We developed asynchronous bindings to:
  - Mercury
  - LevelDB
  - POSIX I/O

Argobots Execution Model



GROUP MEMBERSHIP

- Gossip-based detection
  - Scalable, distributes the comm. load
  - SWIM protocol is one example, rolls membership in with detection
  - Could introduce jitter…
- Vendors could provide an “oracle” for specific classes of faults
  - Won’t necessarily know your service is misbehaving
- Replicated state machine for consistent view of membership (if needed)
  - PAXOS, RAFT, Corfu



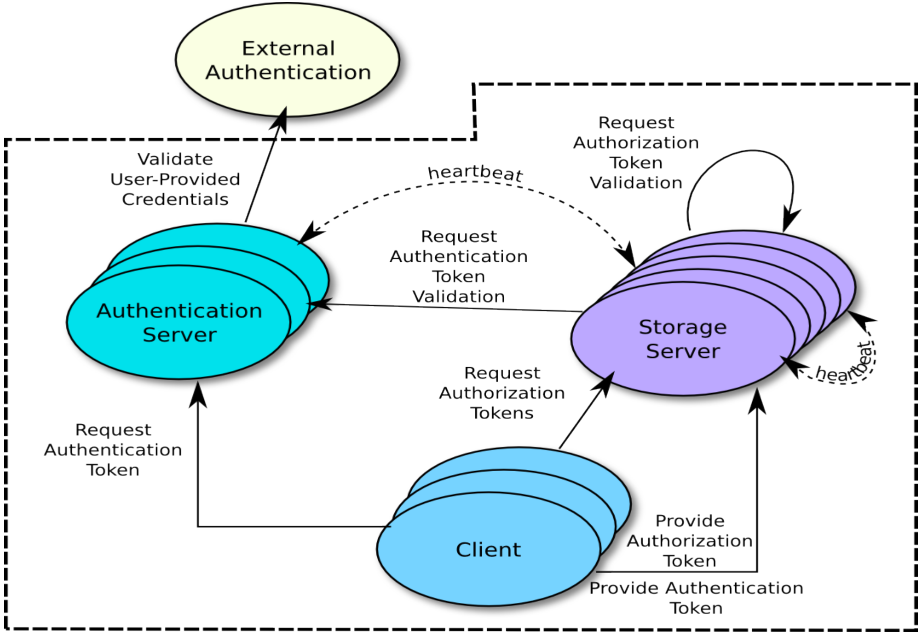
SWIM protocol with 2K nodes, 30 minutes of simulated time. Subgroup size determines number of peers that follow up on a failed ping.

A. Das et al. “SWIM: Scalable weakly-consistent infection-style process group membership protocol.” DSN ’02. 2002.
D. Ongaro et al. "In search of an understandable consensus algorithm.” USENIX ATC 14. 2014.

AUTHENTICATION AND AUTHORIZATION

Services intending to replace parallel file systems must provide (scalable) access control.

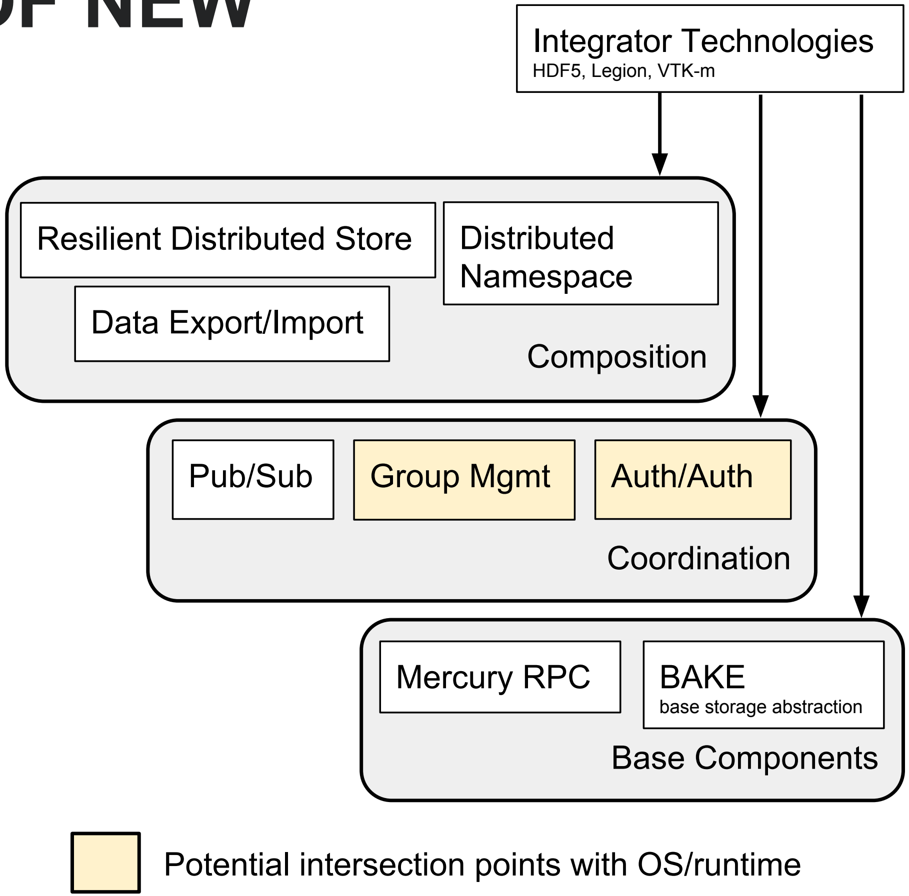
- Integrate with external authentication (Kerberos, LDAP)
- Capability-based approach
  - Caching, delegation to improve scalability
- Building off LWFS work and follow-on activities with L. Ward (SNL) and R. Brooks (Clemson)
  - Mercury prototype



R. Oldfield et al. "Lightweight I/O for scientific applications." Cluster 2006, 2006.

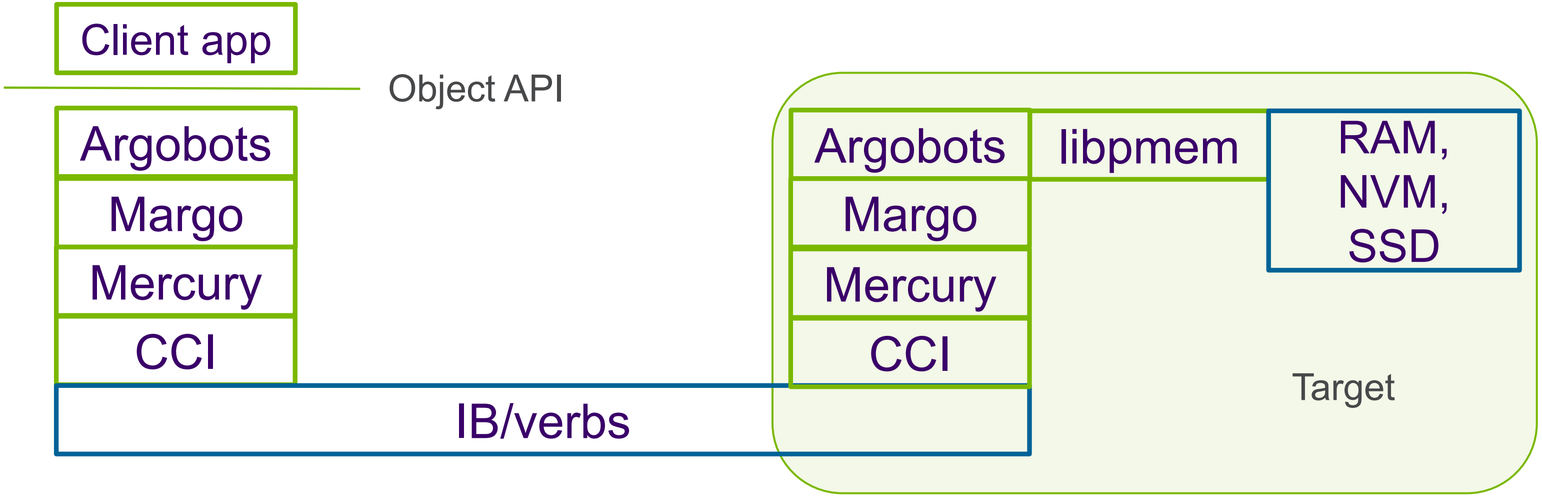
RAPID DEVELOPMENT OF NEW DATA SERVICES

- Provide the building blocks for the next generation of HPC services
- Toolkit of interoperating components
  - Solutions to hard problems
  - Integration with related tech.
- Work with vendors, apps, facilities
- Lower the barrier of entry
  - Teams casually build new services



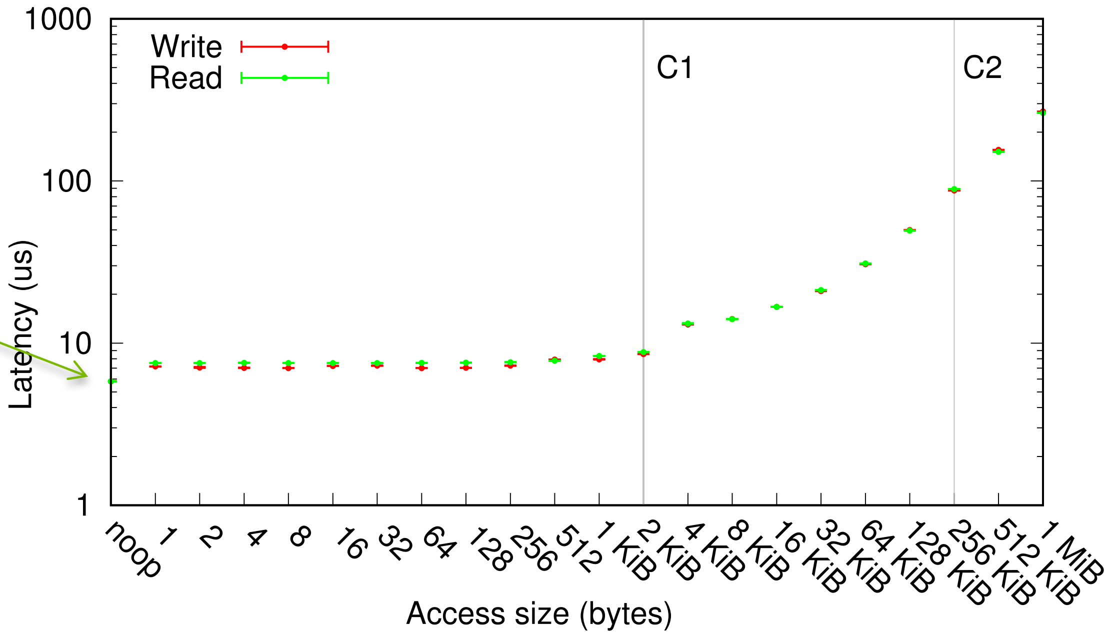
REMOTELY ACCESSIBLE OBJECTS

- API for remotely creating, reading, writing, destroying fixed-size objects/extents
- libpmem for management of data on device
- < 10 usec accesses over FDR IB backed to RAM



P. Carns et al. “Enabling NVM for Data-Intensive Scientific Services.” INFLOW 2016, November 2016.

REMOTELY ACCESSIBLE OBJECTS: HOW MUCH LATENCY IN THE STACK?



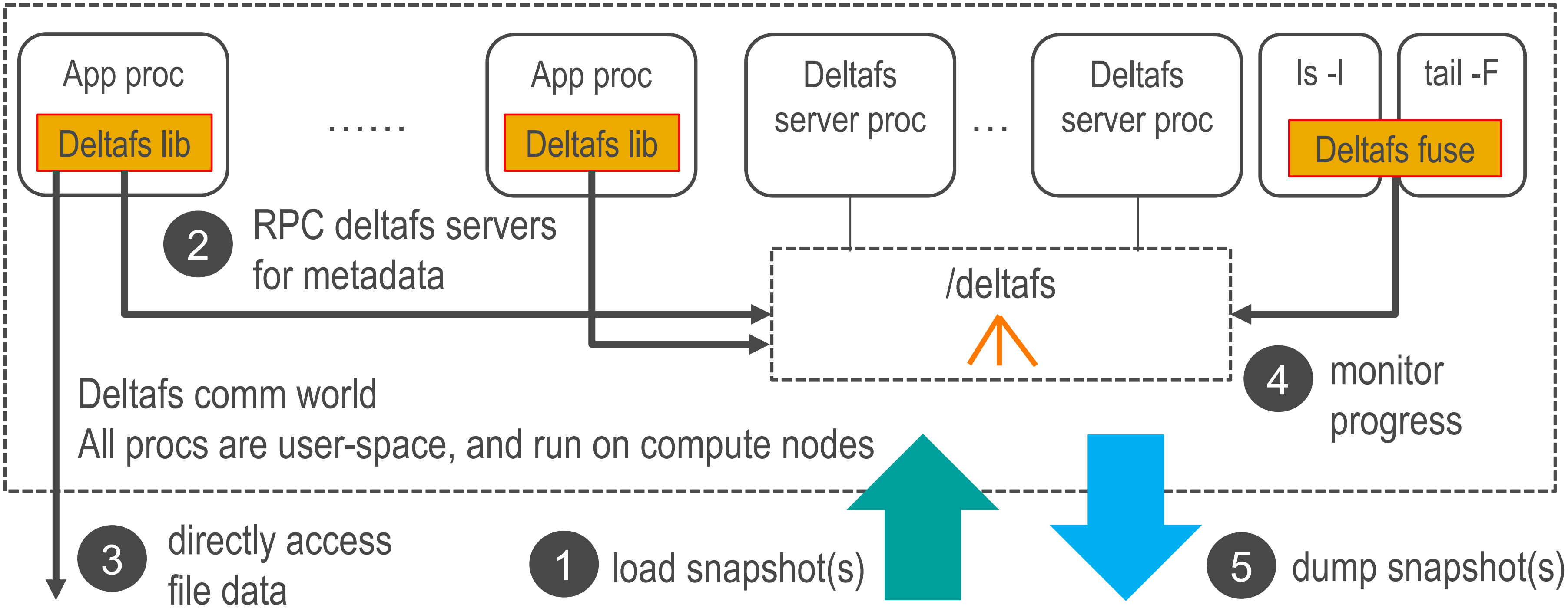
5.8 usec
NOOP

FDR IB, RAM disk, 2.6 usec round-trip (MPI) latency measured separately

Argonne NATIONAL LABORATORY

TRANSIENT FILE SYSTEM VIEWS: DELTAFS

Supporting legacy POSIX I/O in a scalable way.



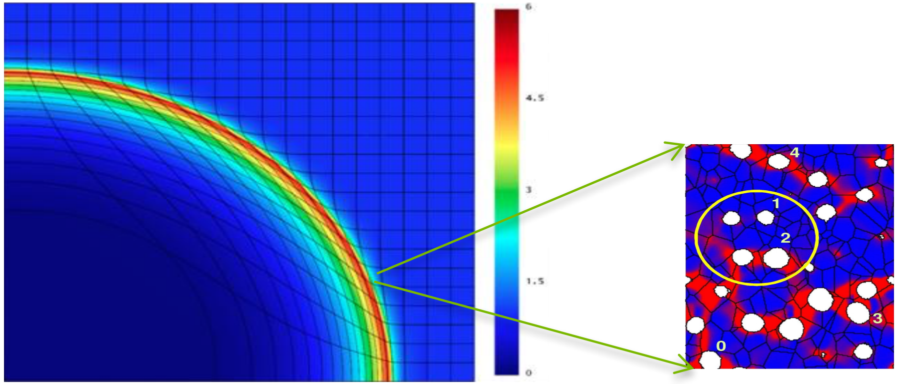
Q. Zheng, et al. "DeltaFS: Exascale file systems scale better without dedicated servers." PDSW 2015. November, 2015.

COMPUTATION CACHING AS A SERVICE

J. Jenkins, G. Shipman, J. Mohd-Yusof, K. Baros, P. Carns, and R. Ross.
“A case study in computational caching microservices for HPC.” IPDRM 2017. June, 2017.

Argonne
NATIONAL LABORATORY

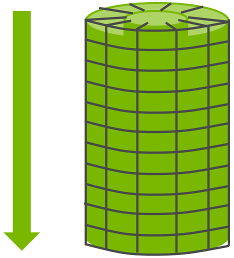
MULTI-SCALE SIMULATION



Coarse-scale model
e.g., Lulesh continuum:
- Lagrangian hydrodynamics
- Unstructured mesh

Fine-scale model
e.g., Viscoplasticity [1]:
- FFT based PDE solver
- Structured sub-mesh

Shockwave

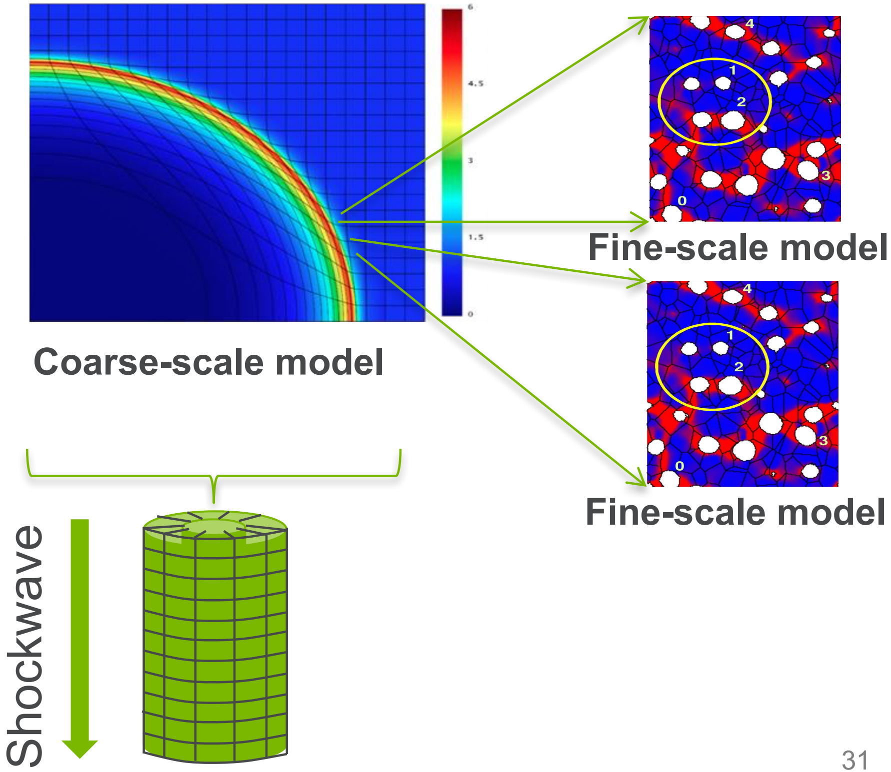


R. Lebensohn et al, Modeling void growth in polycrystalline materials, Acta Materialia, http://dx.doi.org/10.1016/j.actamat.2013.08.004.

- Multi-scale models simulate physical phenomena across multiple time and length scales
- As an example: Loosely coupling continuum scale models with more realistic constitutive/response properties
  - We use the CoEVP proxy application from the ExMatEx project as our case study

Argonne NATIONAL LABORATORY

ACCELERATING SIMULATION WITH COMPUTATION CACHING



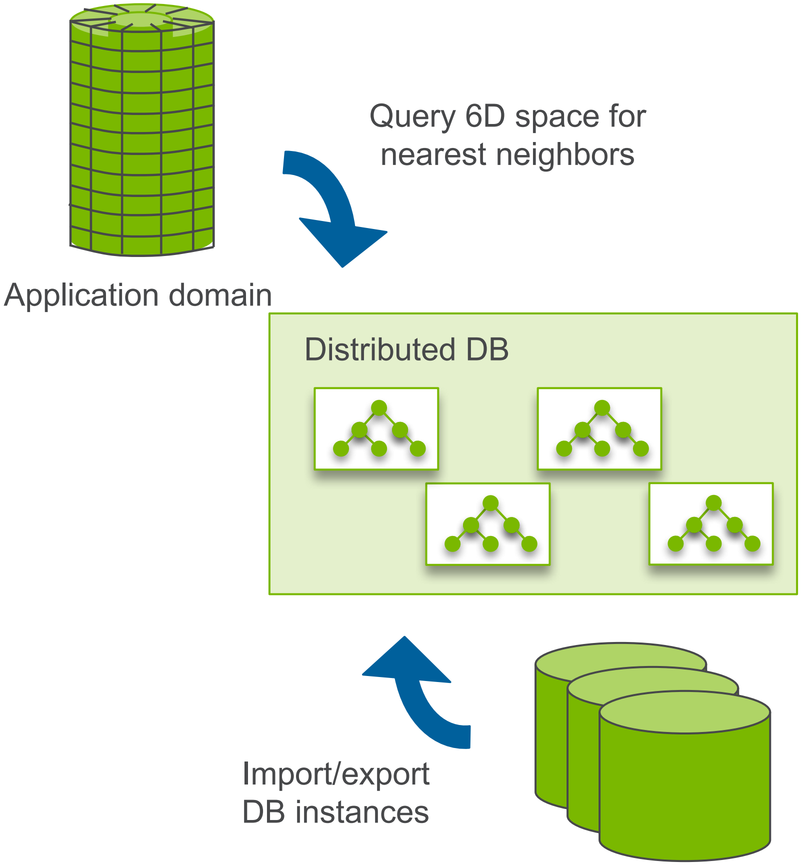
- Phenomena such as shock waves propagate through the coarse-scale model
- Sometimes requires recomputation of similar (or identical) fine-scale models

This is an opportunity for optimization:

If the fine-scale model is expensive, then it may be effective to cache its fine-scale results for later reuse

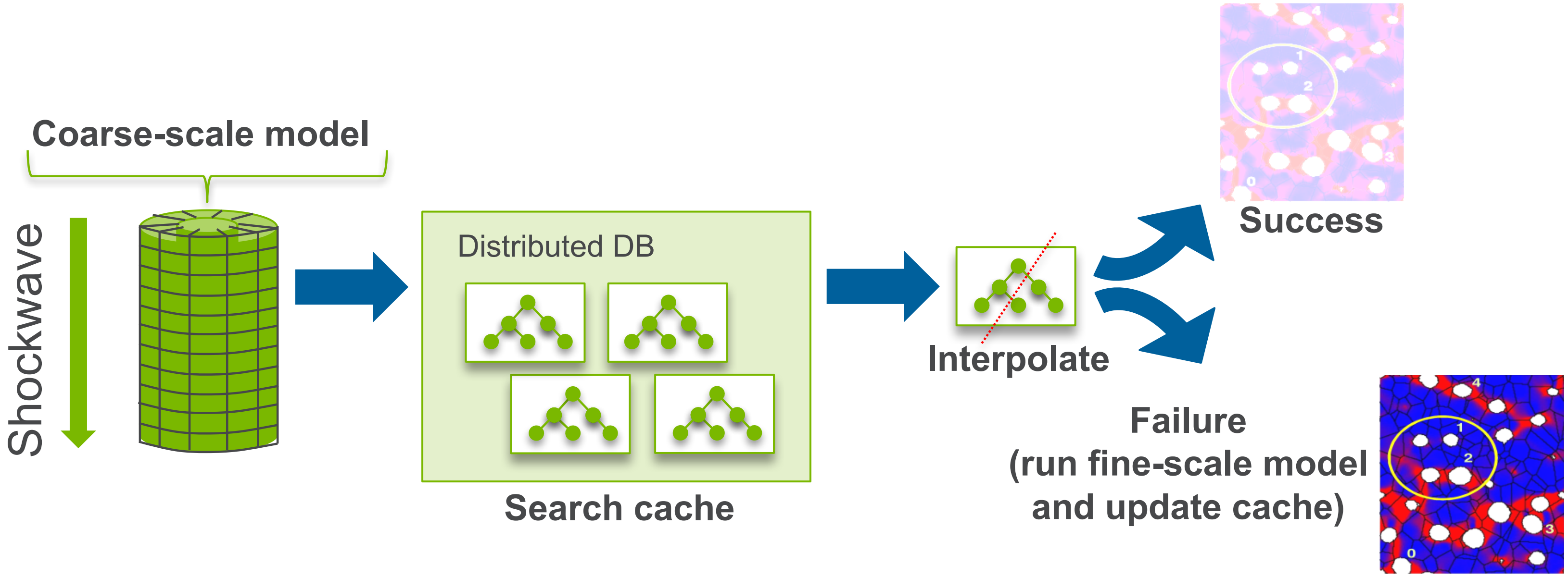
THE CASE FOR A COMPUTATION CACHING SERVICE

- The existing application uses a per-process cache
- Reimagine computation cache as a distributed service
  - Shared cache leads to greater hit rate
  - Sharing computation cache across jobs
  - Possibility of persisting DB
  - More deployment flexibility (e.g. for NVM nodes)
  - Reuse code base in other applications



COMPUTATION CACHING AS A SERVICE

- Search cache for nearest neighbors in multi-dimensional parameter space, interpolate, and check error bounds
- Eventual consistency is a natural fit



WRAPPING UP

SSDBM AND THE RENAISSANCE

Many places where SSDBM supports the renaissance

- Technology
  - Extreme heterogeneity (Nowell)
  - Join operations on GPUs (Rui et al.)
  - Sensor data streams (Gorenflo et al.)
- Data management
  - Publish-subscribe based storage (Qader el al.)
  - In-database linear algebra (Qin et al.)
- Analytics
  - Real-time data analysis (Shein et al.)
  - On-line analytics (panel)
  - Theory guided data science (panel)

A DATA SERVICE ECOSYSTEM

Mochi

Enable broader community to build better, more capable user-level data services than possible today.

- New technologies, architectures, and applications call for new building blocks
- Speed up development and ease maintenance by sharing code
  - Focus development on specifics for use case
  - Specialize/optimize only the performance critical parts
- Share not only low-level building blocks, but microservices:
  - Compose and augment to serve use case
- **http://www.mcs.anl.gov/research/projects/mochi/**
  - Thanks to many at ANL, The HDF Group, LANL, and CMU

Carnegie Mellon University HDF Los Alamos NATIONAL LABORATORY EST.1943 Argonne NATIONAL LABORATORY

THANKS!

THIS WORK IS SUPPORTED BY THE DIRECTOR, OFFICE OF ADVANCED SCIENTIFIC COMPUTING RESEARCH, OFFICE OF SCIENCE, OF THE U.S. DEPARTMENT OF ENERGY UNDER CONTRACT NO. DE-AC02-06CH11357.

Carnegie Mellon University

HDF

Los Alamos NATIONAL LABORATORY EST. 1943

Argonne NATIONAL LABORATORY