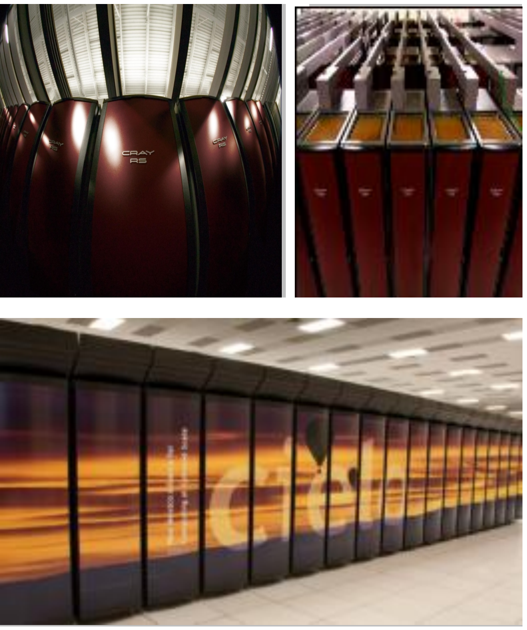


Blurred Lines: You Got Your Memory in My Storage!

Jay Lofstead

Scalable System Software
Sandia National Laboratories
Albuquerque, NM, USA
gflofst@sandia.gov

ROSS@HPDC 2017

June 27, 2017

SAND2017-2916 PE

Exceptional service in the national interest

What is Memory or Storage?

- Things placed in memory have external metadata, generally in program code
  - more compact representation, optimized for interaction with the processors

- Things placed in storage are wrapped in metadata to make them easily usable by other applications
  - file formats to make reading simulation output into visualization tool
  - prescribed (or annotated) endianness.

- What about shared fate? What about wrapping metadata around data in DRAM?

File/Storage Systems Questions

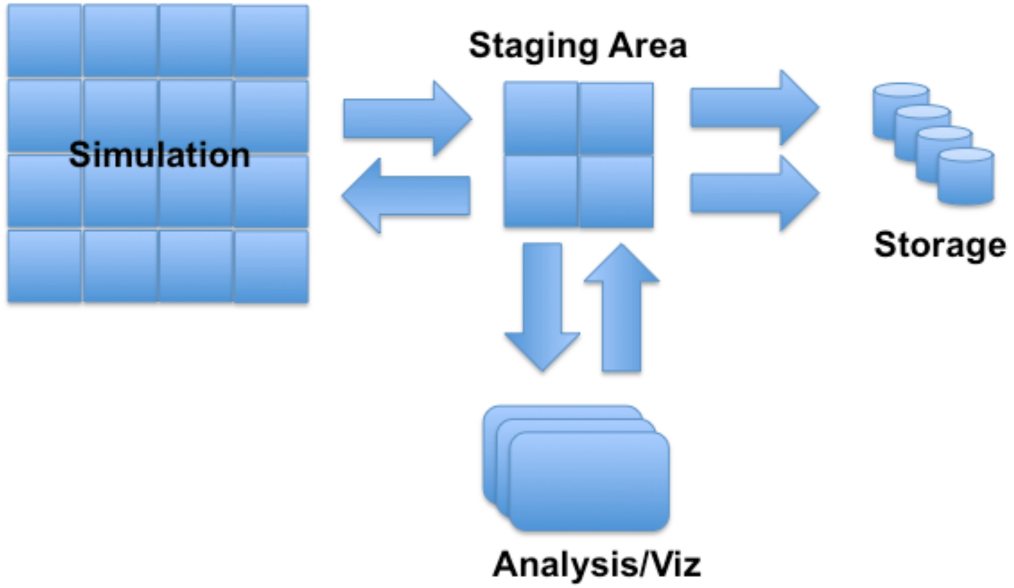
- If POSIX interface is gone, are there files?
  - How do we identify a collection of bytes we want?
- If we use CPU-level get/put instead of block read/write is it storage still?
  - Either directly or via something like libpmem or mmap
- Do we need a storage abstraction for portability anymore?
  - Endian-ness is almost exclusively little endian now.
  - Are there other motivations?
- Are consistency and coherence a programmer or file/storage system responsibility? What about security?
- Since networking people worry about machine instructions, what can storage/IO people afford as service functionality?

OS and Runtime Support Required

- How to support composing simulation, analytics, and viz all online?
  - Traditional approach is push/pull to/from storage between components
- Hobbes (Kitten + Palacios in particular) offers isolation and sharing
  - explicit memory allocation and sharing
  - signaling still required (spin-wait)
- Ok start, but not sufficient. Consider these models...

Phase 1 Architecture

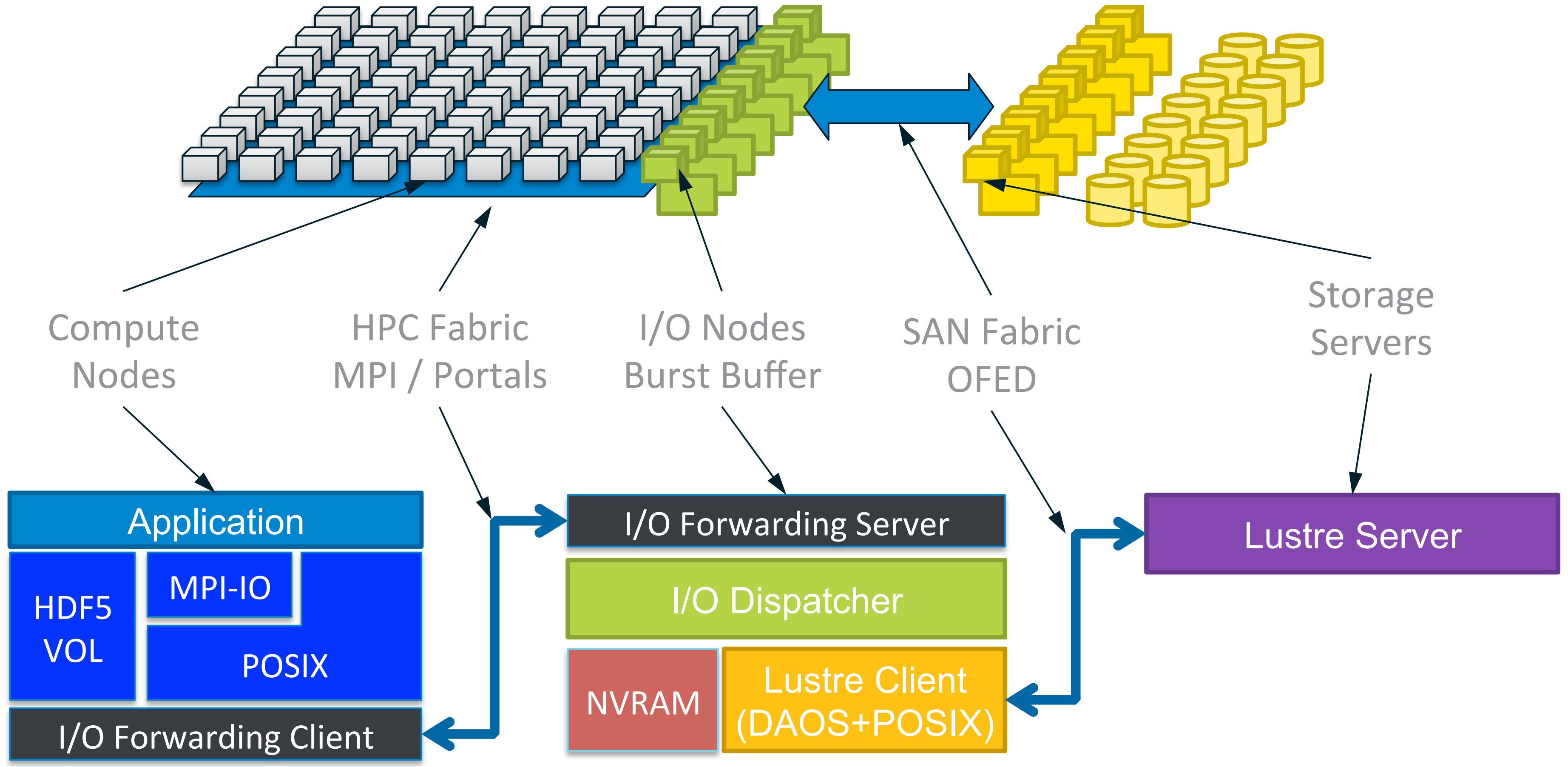
- Use extra compute nodes for their memory
- Data staging work starting in the 1990s, picked up steam in the 2000s.
  - Chain of evidence suggests this is the origin of “burst buffers”, as least in name



Predominant Uses (Phase 1)

- Manually managed IO bursts
  - IO Forwarding nodes on BlueGene
- Offloading communication-heavy operations to fewer nodes with more data each
  - FFT for seismic data
- Offloading independent operation to fewer nodes for asynchronous processing
  - Calculating min/max, bounding box filtering, etc.

Phase 2 Architecture (and Software)



Predominant Uses (Phase 2)

- Offer Flash into or near IO path
  - Some job scheduler support, including rudimentary allocation, data pre-staging, and data draining
- Suggest use for data rearrangement (fast array dimension) and similar processing
  - Not completely though through since these are IOPS bound activities that effectively remove devices from availability slowing aggregate IO bandwidth for the machine.
- If only IO path to storage is through these devices, potential problems abound

Phase 2a Architecture

- Same as Phase 2, except the NVM is on the compute nodes instead of centralized.
- Additional examples, such as Aurora at ANL, will have both models.
- When on compute node only, interference effects can be significant (network, device, potentially memory or disk bus affecting local node use)
- Summit will be a test case for Phase 2a

- SCR attempting to leverage these architecture for checkpoints

Problems Demonstrated for Phase 2

- memchached on persistent memory needs to be reworked (Marathe, et al., HotStorage 2017)

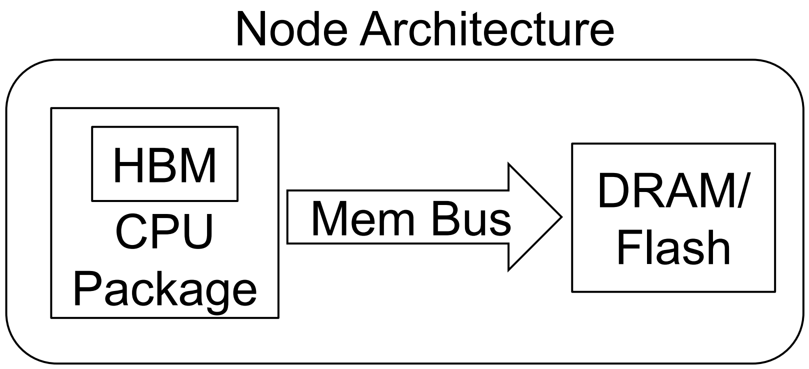
- Inifiniswap (University of Michigan) moving SCR approach into hardware
  - https://www.nextplatform.com/2017/06/12/clever-rdma-technique-delivers-distributed-memory-pooling/

- wear leveling for NVM solutions in abundance

- Kokkos (https://github.com/kokkos) formalizes different kinds of memory for compute purposes.

Phase 3 Architecture

- Nodes gain HBM on package and more memory/storage in the memory bus or PCIe



- Additional node-local storage added
  - 3D XPoint most hyped example
  - node-local Flash/SSDs also possible due to form factor

Phases 2 & 3 Challenges

- Storage devices reach or exceed interconnect speeds
  - Storage stack overheads no longer hidden by device latencies

- Unlike DRAM and disk, NVM has an erase cycle that can take as long as writing. We need to program understanding that overwriting costs 2x writing to clean space (consumer grade devices—less for enterprise grade, but still not free).
  - Some belief background erasure can address this (not me).

- Maintaining coherency and consistency for multi-user, globally shared space

Predominant Use Cases (Phase 3)

- Out of core computations
  - Better support for data analytics workloads as a side benefit
- RDMA access still probably desired, but less interference since memory bus will only be hit when leaving the CPU package
- Do we buy any memory/storage for local memory bus since spending so much for HBM?

Phase 4 Architecture

- Memory-centric Design (Gen-Z Consortium)
  - HPE “The Machine” prototype (160 TiB DRAM + 40 nodes)
- In network (on switches) storage
- DRAM, potentially in the same address space
- Line between memory and storage all but gone

Predominant Use Cases (Phase 4)

- Coherent virtual fat nodes operating on 100s TB
- Persistent storage near/fast enough to “swap” to
- Online workflows become the natural model
  - Lots of places to stash data between compute components
  - Easier programming model to access data since it can be in a shared, directly addressable address space (just pass a pointer).

OS-level “Memory” Tools?

- What can be done to address offering different device access in a meaningful way?
  - IO-500 list thought: ephemeral, persistent, resilient (, and archive?)
- How do we address remote devices in a way that makes sense?
  - Consider intermediate (ephemeral or persistent) data and long term data (resilient or archive?)
- Should the APIs be different since performance differences are far less and hierarchy is so much deeper?

Decaf Project Contributions

- DOE ASCR Data Management Project ending third year
- SmartBlock system demonstrates reusable components
  - Motivating shared system services
  - Exposes OS/Runtime gaps to handle cross-job sharing
- Exploring connectivity API requirements for component to component communication/sharing
- Next steps investigating more system services for hosting runtime deployed code into service infrastructures

Sirius Project Contributions

- DOE ASCR SSIO Project finishing second year
- User level deciding how to split data sets into higher information density chunks
  - ZFP, split doubles at the byte level, striding, combinations, or others
- Data placement management tools
  - writing EVERYWHERE (really objects in essence even though files now)
  - restage months later for reading based on information density (utility)
- Metadata management for querying based on data contents
  - and support QoS needs
- Quality of Service at the storage device level to give reasonable predictions for IO operations
  - reservations, ML-based prediction, and historical timing statistics

SNL ATDM Data Warehouse

- NNSA funded SNL ATDM (somewhat part of ECP)
- Data management for AMT (Dharma from SNL CA)
  - Also investigating coupling with analytics
- Combining Sirius data access/tagging/metadata ideas with Decaf services/API infrastructure

Questions?

Jay Lofstead

gflofst@sandia.gov

Shameless Plug:

Popper – Practical Falsifiable Research

http://falsifiable.us