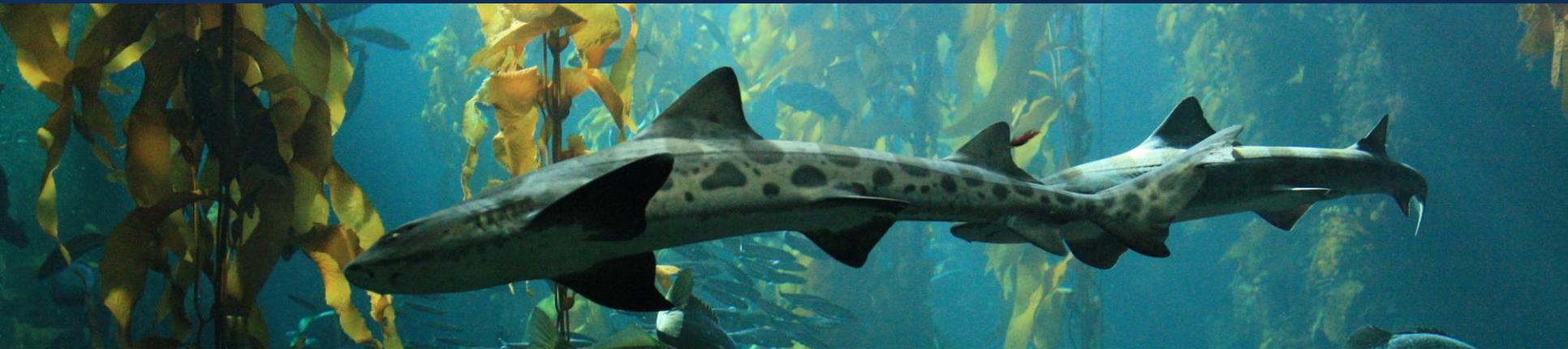


Exceptional service in the national interest



Leveraging In-Memory Key/Value Stores in HPC

Kelpie
Craig Ulmer



Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

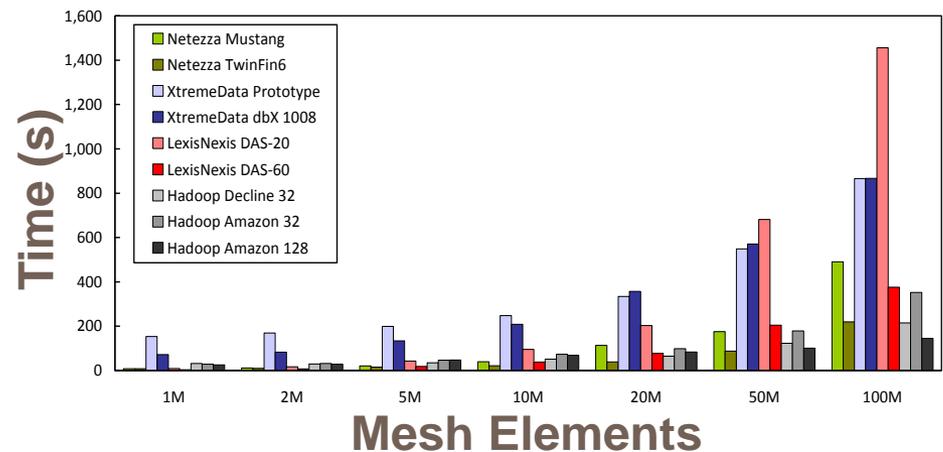
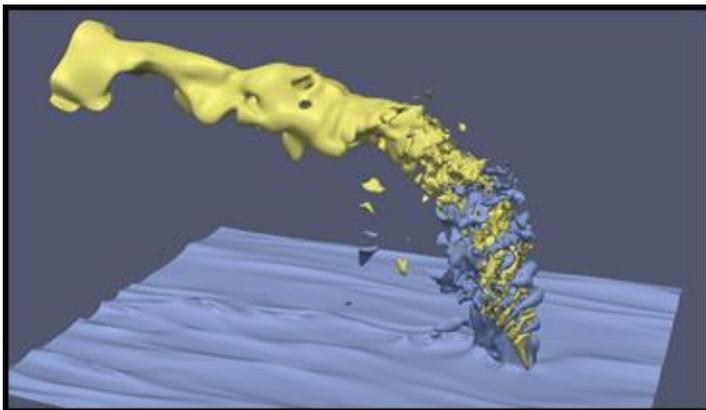
Overview: Leveraging KV Stores

- NoSQL: Dramatic impact on *Out-of-Core* processing
 - Frameworks handle scheduling, placement, and reliability
 - Enables novice users to take advantage of parallel processing
- Good for information apps, but not Scientific Computing
 - NoSQL primary concern is disks, assumes cluster hardware
 - Can we leverage some of the concepts of NoSQL in HPC?
- Distributed In-Memory Key/Value stores are extremely useful
 - **Kelpie**: Experimental 2D KV for HPC built on **Nessie** (portable RDMA)
 - Examples: Caching, Glue, Communication, New Frameworks

LESSONS FROM NOSQL

How did we get here?

- Sandia started investigating NoSQL frameworks in 2008
 - Can we leverage for post-processing on clusters?
 - Make it easier to build one-off analysis
- Ported analysis kernels to different frameworks/stores
 - Hadoop, Pig, Accumulo, Cassandra, MongoDB, SectorSphere...
 - LexisNexis DAS, InfoSphere, XtremeData, Netezza
- Hadoop MapReduce most practical



MapReduce Recap

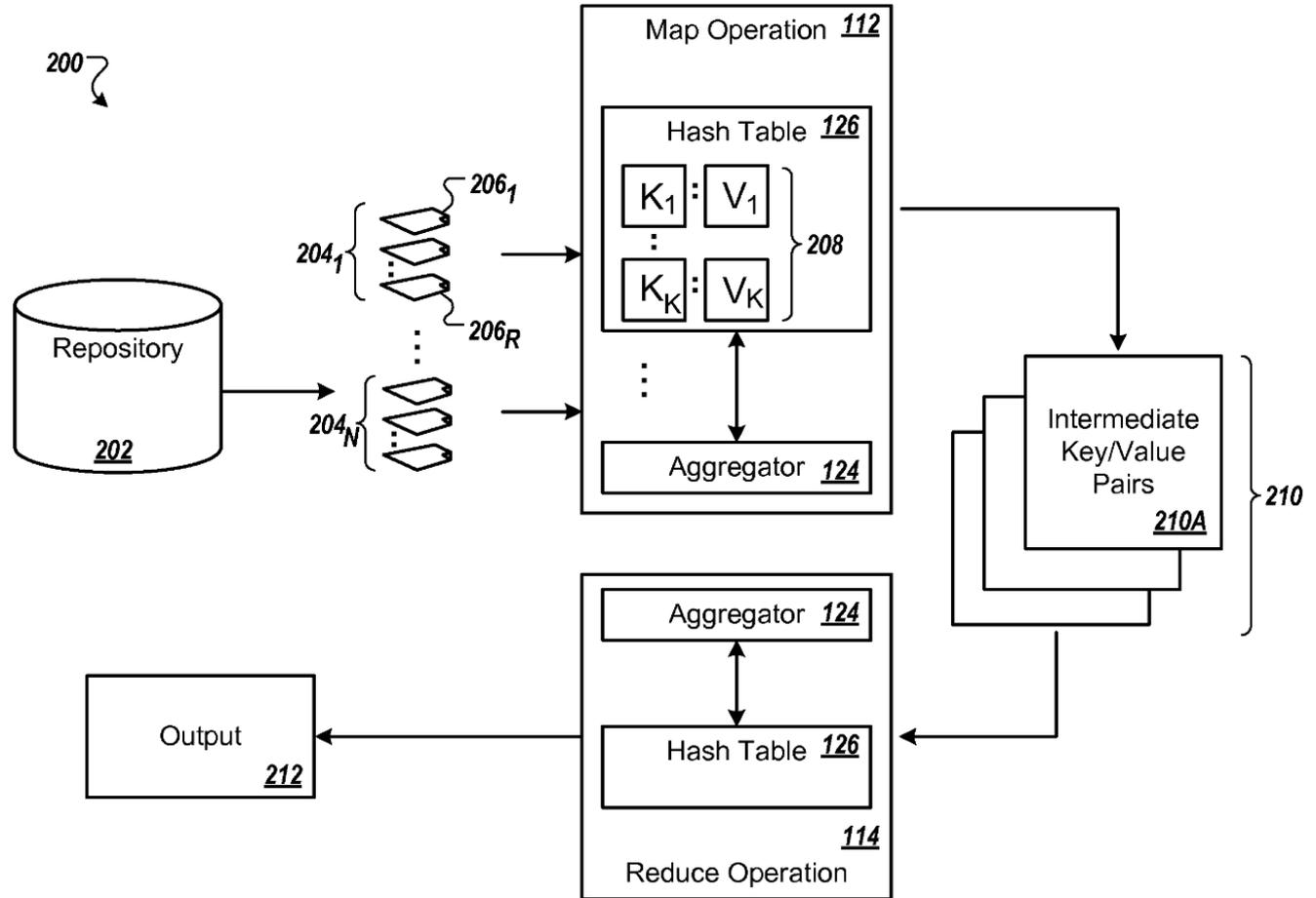
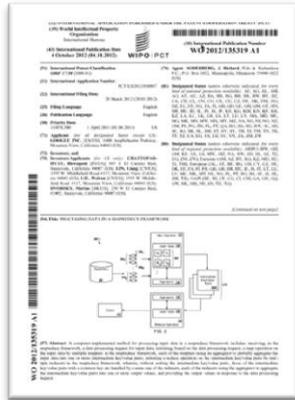


FIG. 2

Good Points of MapReduce

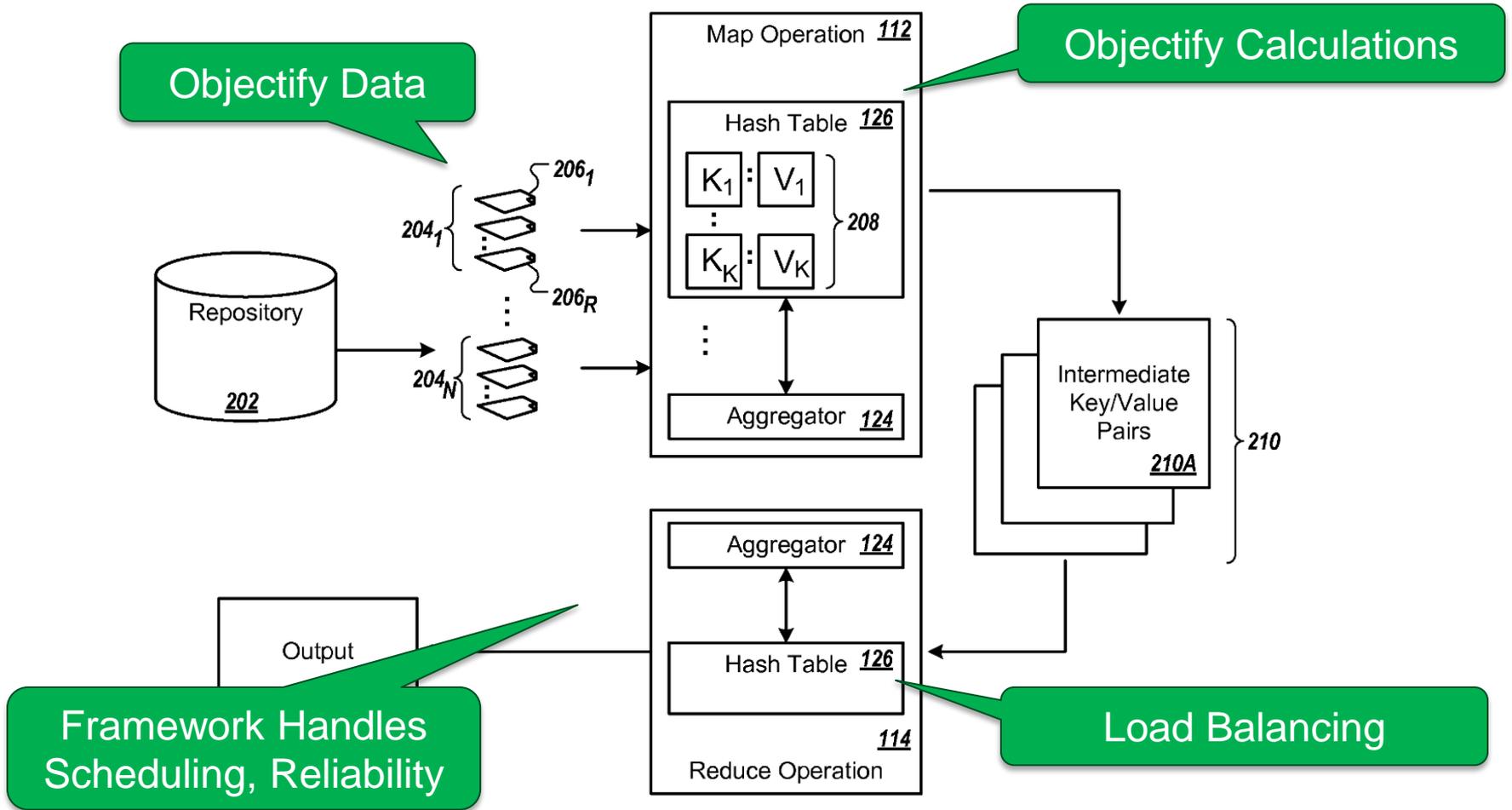


FIG. 2

Bad Points of MapReduce

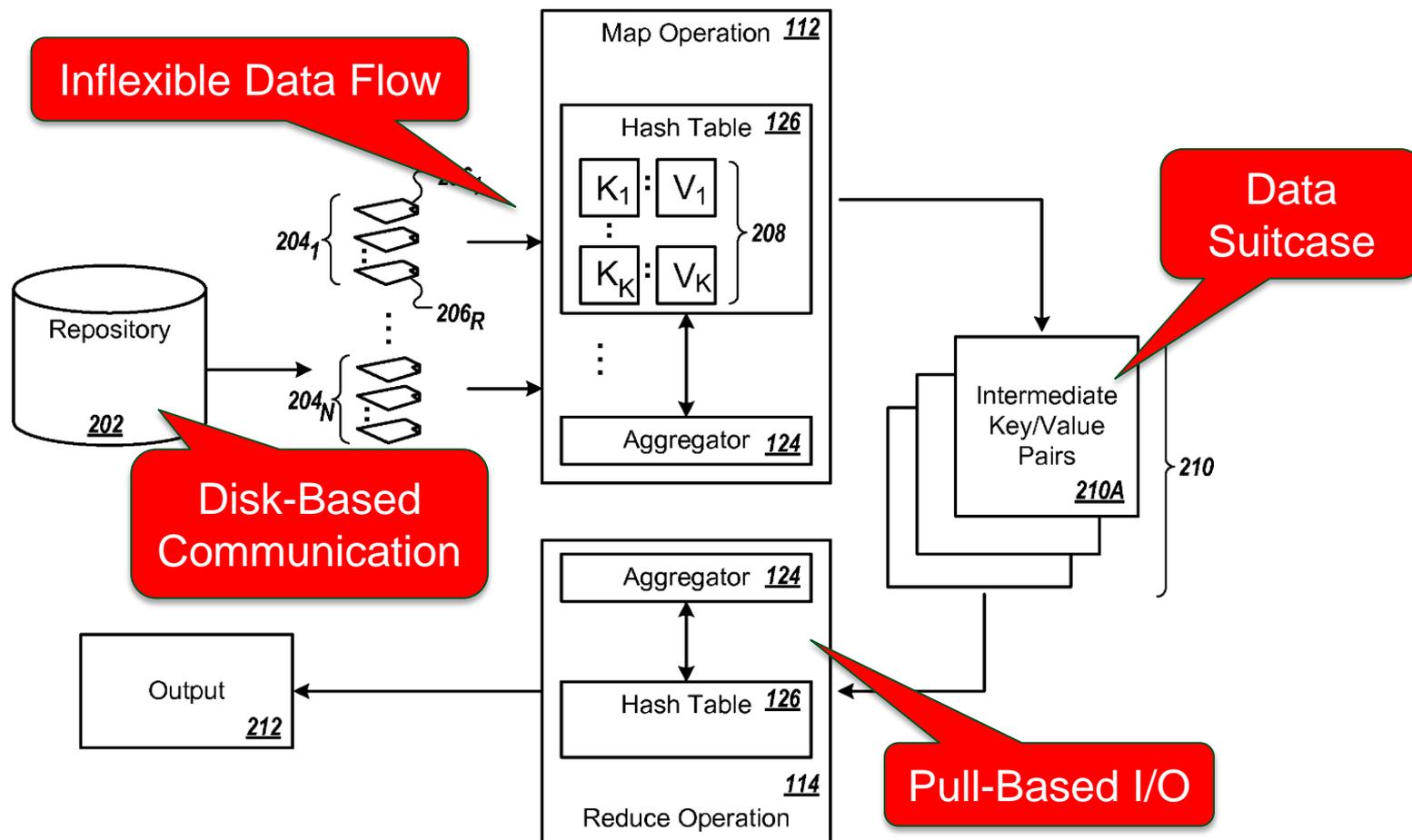


FIG. 2

Other Lessons from NoSQL

- Frameworks trick users into providing useful info
 - Atomic operations, data boundaries
- Indirection is worthwhile
 - Frameworks can do more, Users micromanage less
- Basic functionality is often good enough
 - HDFS: Append-only writes
- 2D Stores > 1D Stores
 - Accumulo: Use a row to build an index generated by multiple writers
- Be open to alternatives
 - SQL vs NoSQL

Meanwhile, Back in HPC

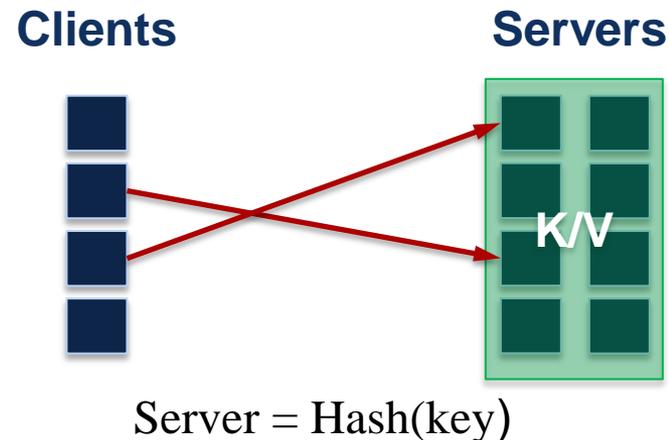
- What are the main concerns in HPC today?
 - Reliability concerns
 - Architecture changes: Fat nodes
 - Application pipelines
 - New programming models
- Are any of the NoSQL ideas relevant to HPC?
- Building blocks: Key/Value Stores



KEY/VALUE STORES

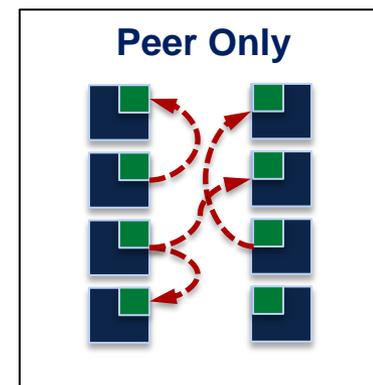
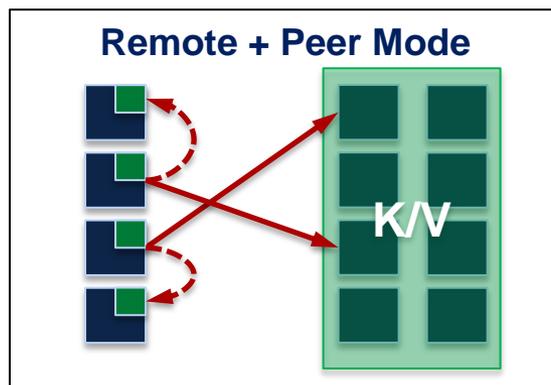
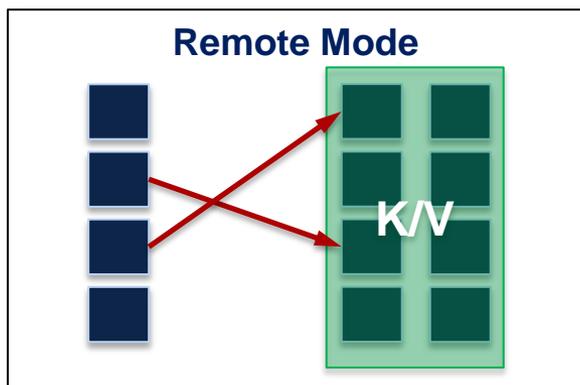
In-Memory, Key/Value Stores

- KV Stores: an old idea with renewed interest
 - Distribute a number of servers to hold values
 - Use a key label to reference data
 - Put/Get interface
- Many existing packages
 - Cluster: Memcached, Redis, Kyoto Cabinet, RAMCloud,...
 - HPC: IBM PIMD



Kelpie: Distributed KV Store for HPC

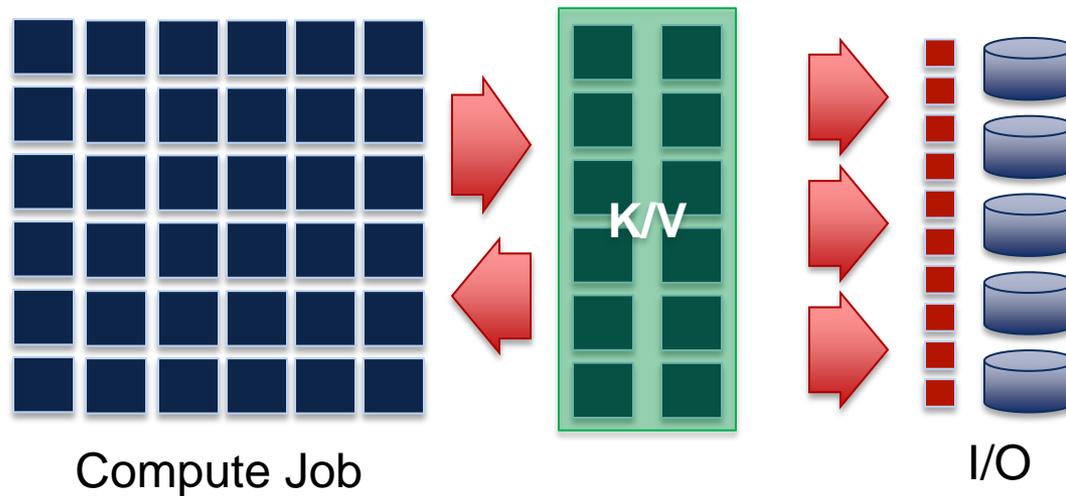
- SNL Developing experimental KV Store for HPC
- Built on top of **Nessie**
 - Portable RPC+RDMA library (IB, Gemini, BG/P, Portals, MPI)
- Typical KV with tweaks
 - 1D or 2D Keys
 - Optional Backing store (local/remote disks)
 - Different modes with *Location Hints*



KV USE STORIES

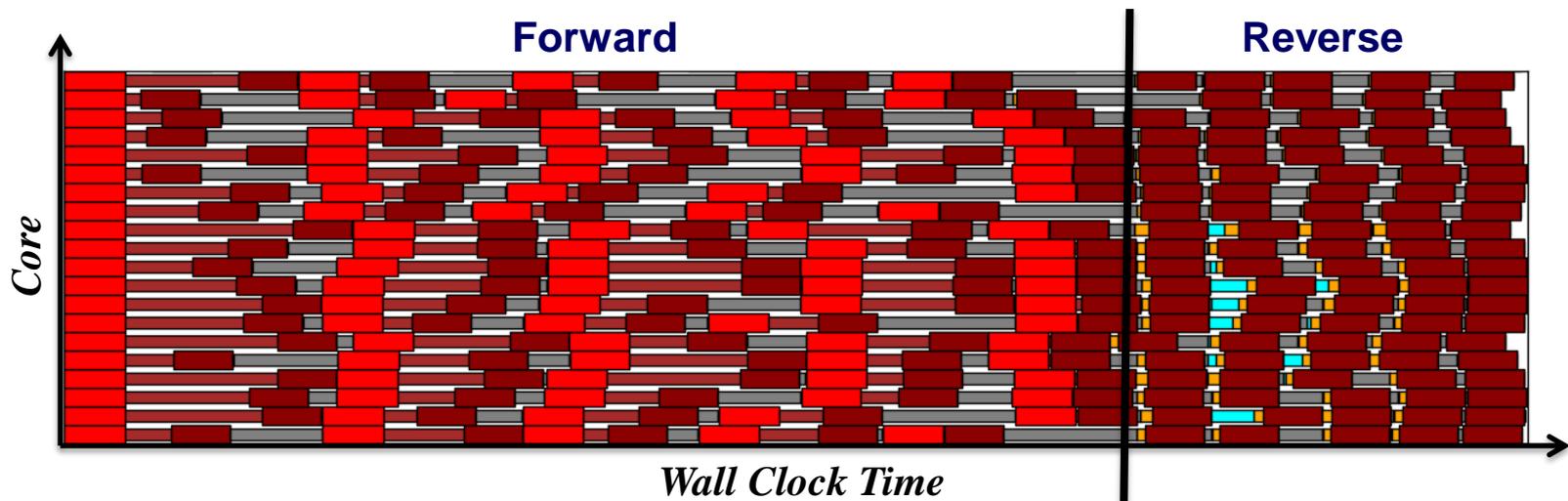
[1/4] Application Caching and I/O

- Leverage K/V store simply as a way to store intermediate data
 - Users put/get blocks of data via references
 - Scale performance tradeoffs by adjusting K/V store size
 - Also useful for I/O swells



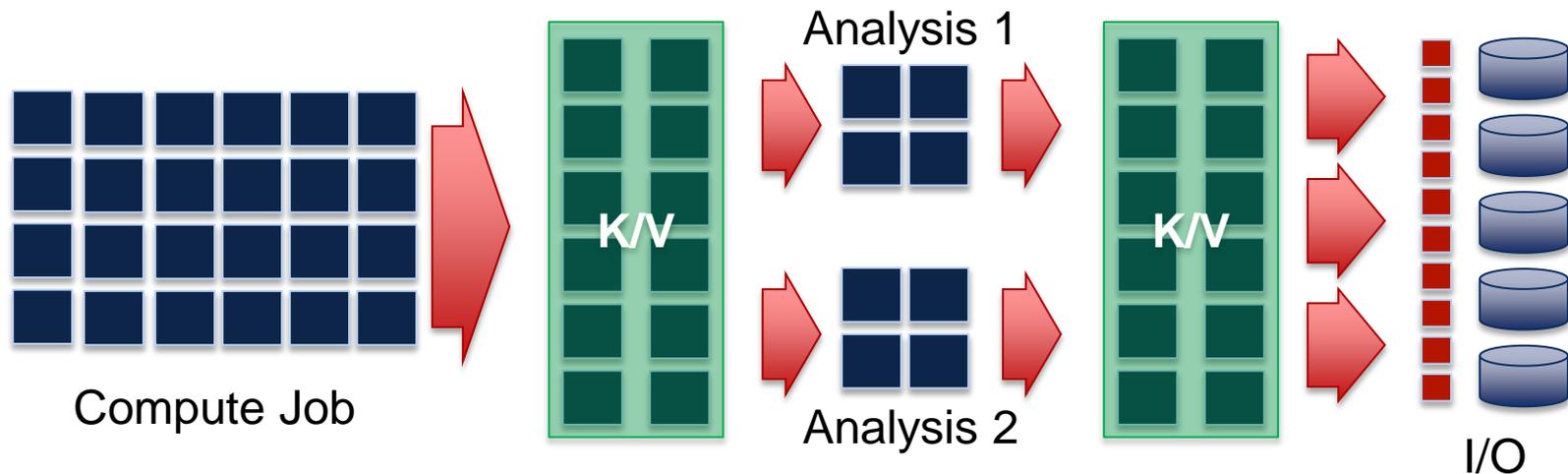
Forward/Reverse Cache

- BSP Code with forward/reverse dataflow
 - During reverse stages, reuse intermediates from forward stages
 - Requires in-memory storage to be practical
 - Regular access pattern makes prefetch easy
- Proxy application for tradeoff studies



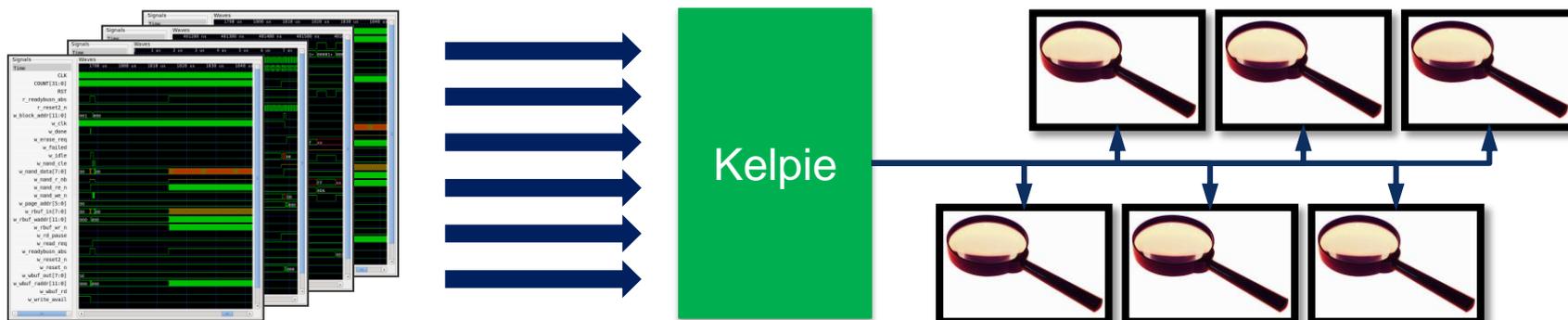
[2/4] Multi-Simulation Glue

- Many users need to join one monolithic job to another
 - Painful to add coupling – some use I/O as the medium
 - KV Stores provide a natural exchange point



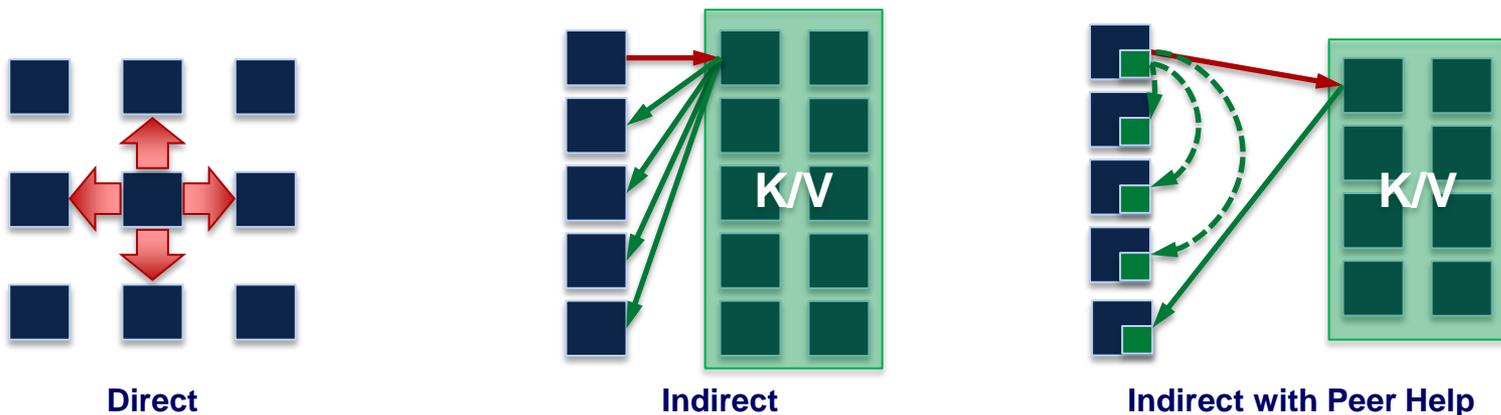
Glue: Hardware Fuzzing

- SNL: Digital hardware fuzzing for reliability testing
 - Flight Recorder Design: Operates in RF-hostile environment
 - Will data corruption lock up the hardware?
- Parameter study: Many long-running VHDL/Verilog sims
 - Previously: Discovered two design problems over multi-month sim
 - Current: Capture waveforms, characterize, look for abnormalities



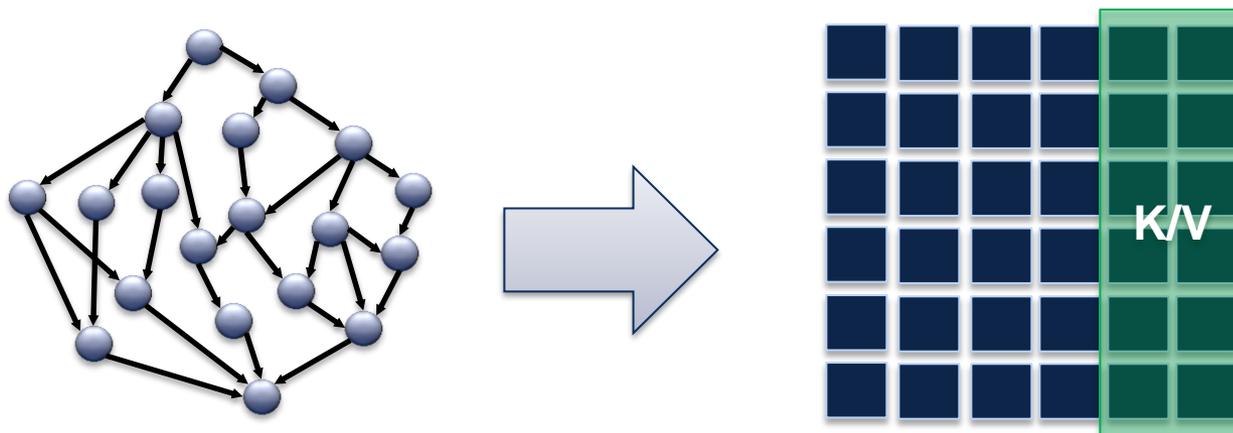
[3/4] Communication without MPI

- Use KV Store for Inter-Process Communication
 - Work at a higher level of abstraction
 - May be of use in workloads with dynamic data flows
 - **Warning:** MPI does IPC very, very well
- Early experiences porting **Mantevo** applications
 - Positive: Feels more like symbolic language
 - Negative: Performance loss due to indirection



[4/4] Frameworks

- Many are working on computational frameworks for HPC
 - ASCR FOX: Fault-Oblivious eXecution
 - SNL: DAG Task Scheduling for Reliability
- KV Stores provide a useful starting point
- Example: Go/MR: GoLang implementation of MapReduce
 - Utilizes Kelpie for storing objects
 - 2D Keys useful for collecting related items



Summary

- NoSQL efforts tested out a variety of interesting concepts
- Key/Value Stores are a valuable service for HPC
- Kelpie
 - Currently tested at small scale on IB, BG/P, and Gemini
 - Expanding our user examples and making more robust
 - Planned external release in June

