High-Performance
Big Data

Designing High-Performance, Resilient and Heterogeneity-Aware Key-Value Storage for Modern HPC Clusters

Dipti Shankar

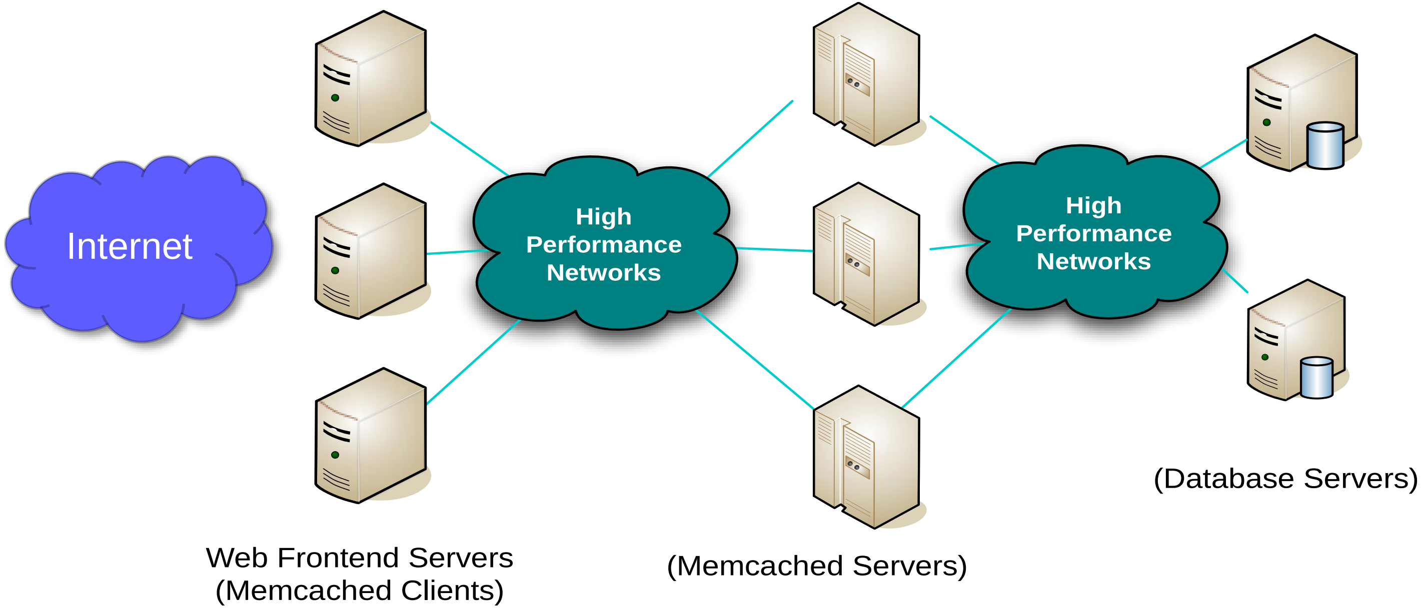
Network-Based Computing Laboratory

The Ohio State University

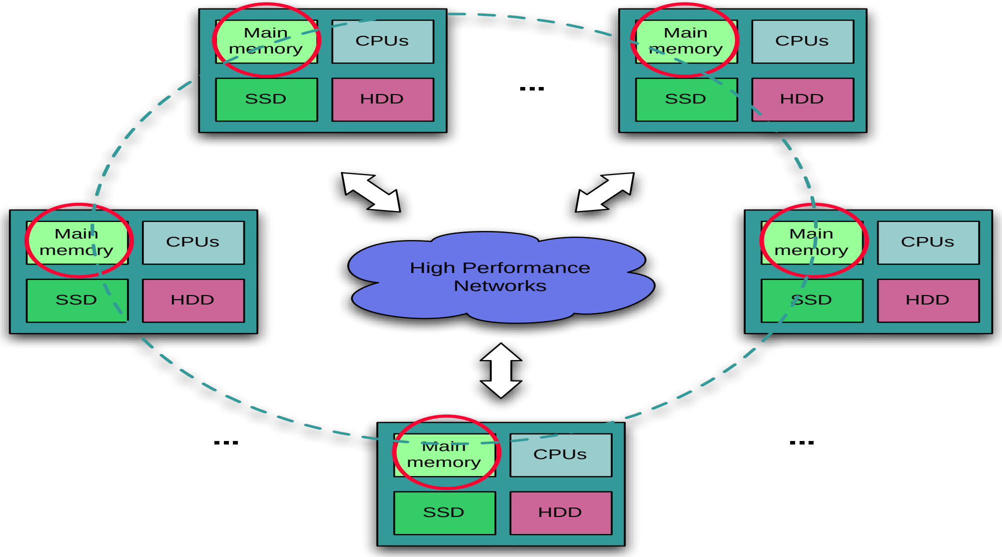
http://hibd.cse.ohio-state.edu/

Overview of Web 2.0 Architecture and Memcached

- Three-layer architecture of Web 2.0
  - Web Servers, Memcached Servers, Database Servers
- Memcached is a core component of Web 2.0 architecture

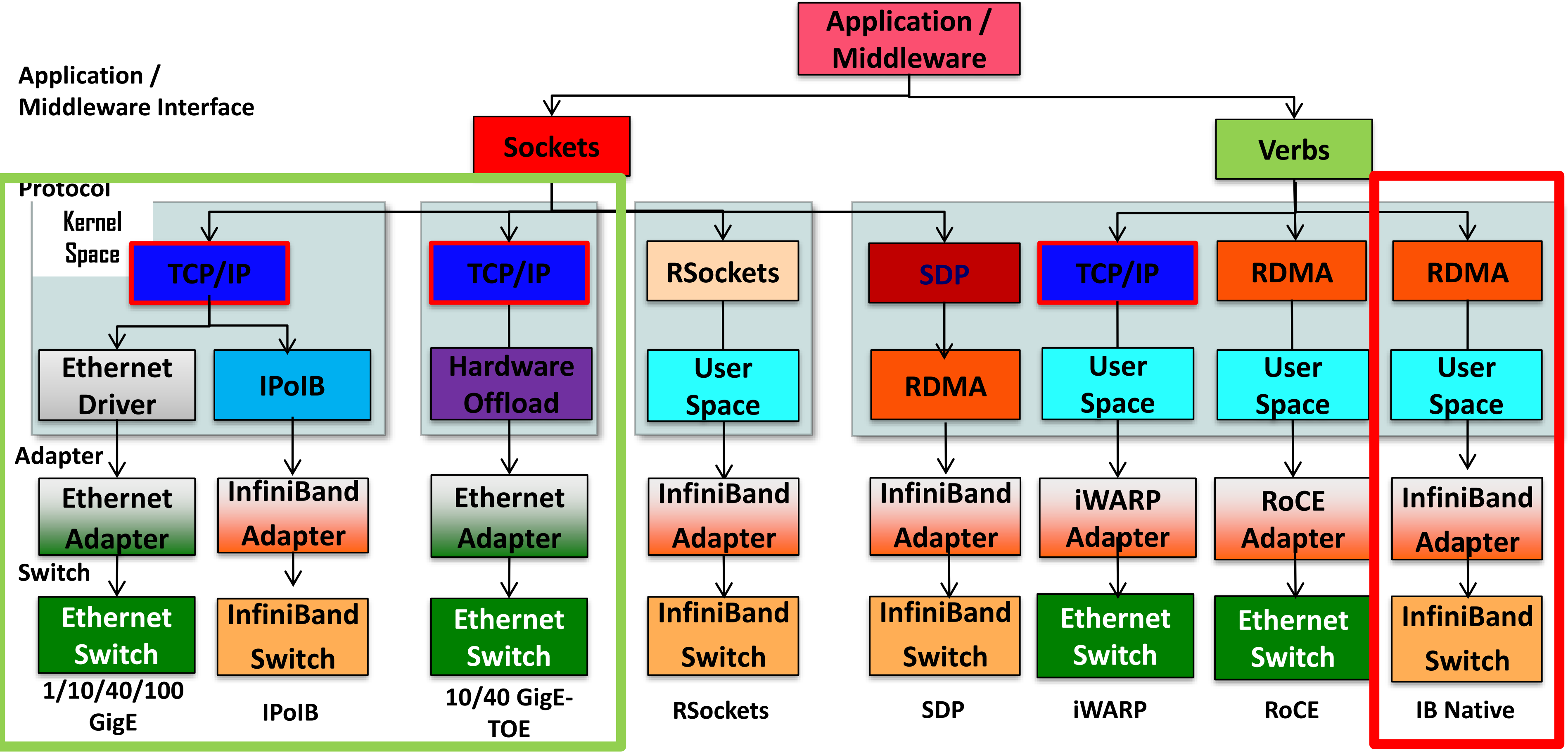


Memcached Architecture



- Distributed Caching Layer
  - Allows to aggregate spare memory from multiple nodes
  - General purpose
- Typically used to cache database queries, results of API calls
- Scalable model, but typical usage very network intensive

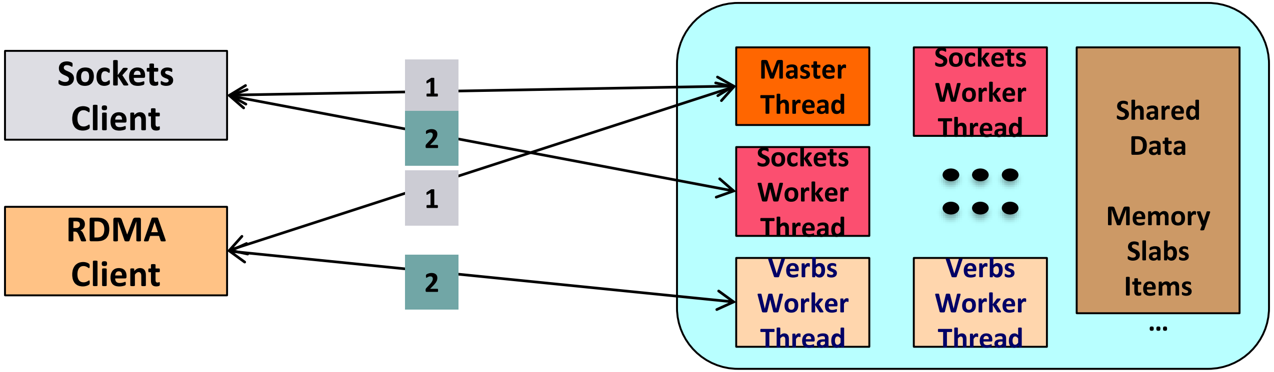
Interconnects and Protocols in OpenFabrics Stack



Open Standard InfiniBand Networking Technology

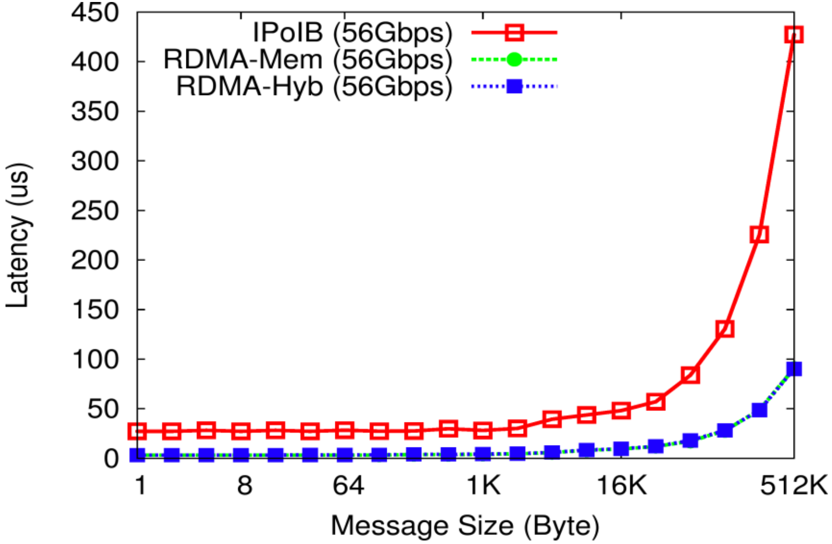
- Introduced in Oct 2000
- High Performance Data Transfer
  - Interprocessor communication and I/O
  - Low latency (<1.0 microsec), High bandwidth (up to 12.5 GigaBytes/sec -> 100Gbps), and low CPU utilization (5-10%)
- Flexibility for LAN and WAN communication
- Multiple Transport Services
  - Reliable Connection (RC), Unreliable Connection (UC), Reliable Datagram (RD), Unreliable Datagram (UD), and Raw Datagram
  - Provides flexibility to develop upper layers
- Multiple Operations
  - Send/Recv
  - RDMA Read/Write
  - Atomic Operations (very unique)
    - high performance and scalable implementations of distributed locks, semaphores, collective communication operations
- Leading to big changes in designing HPC clusters, file systems, cloud computing systems, grid computing systems, ….

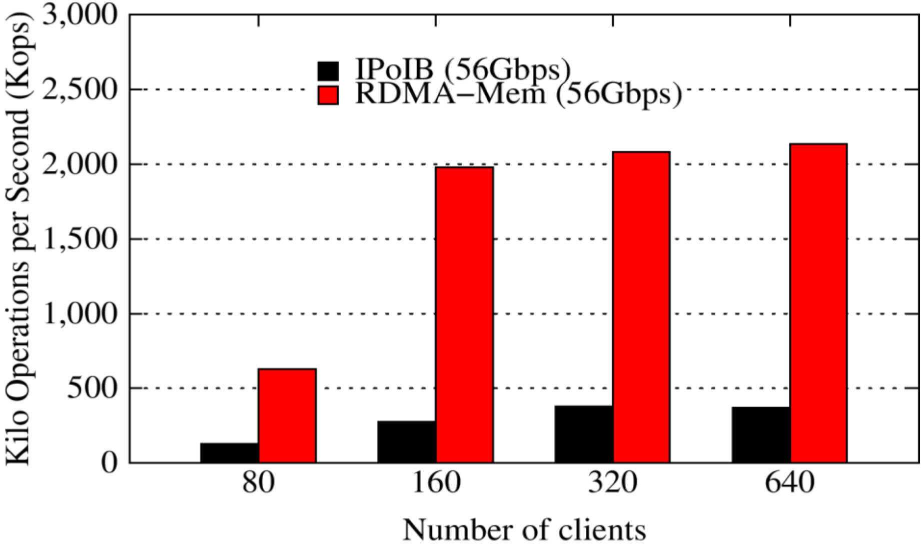
Memcached-RDMA Design



- Server and client perform a negotiation protocol
  - Master thread assigns clients to appropriate worker thread
- Once a client is assigned a verbs worker thread, it can communicate directly and is “bound” to that thread
- All other Memcached data structures are shared among RDMA and Sockets worker threads
- Memcached Server can serve both socket and verbs clients simultaneously
- Memcached applications need not be modified; uses verbs interface if available

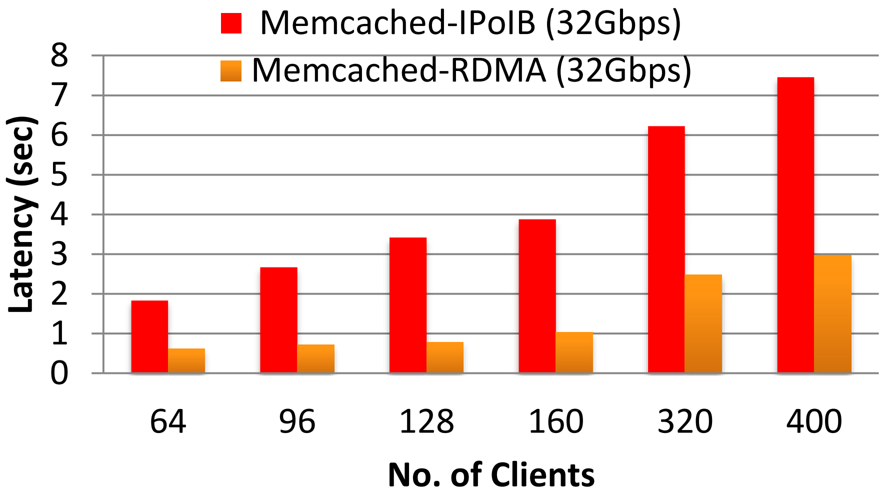
Performance on SDSC-Comet - OHB Latency &YCSB-B Benchmarks

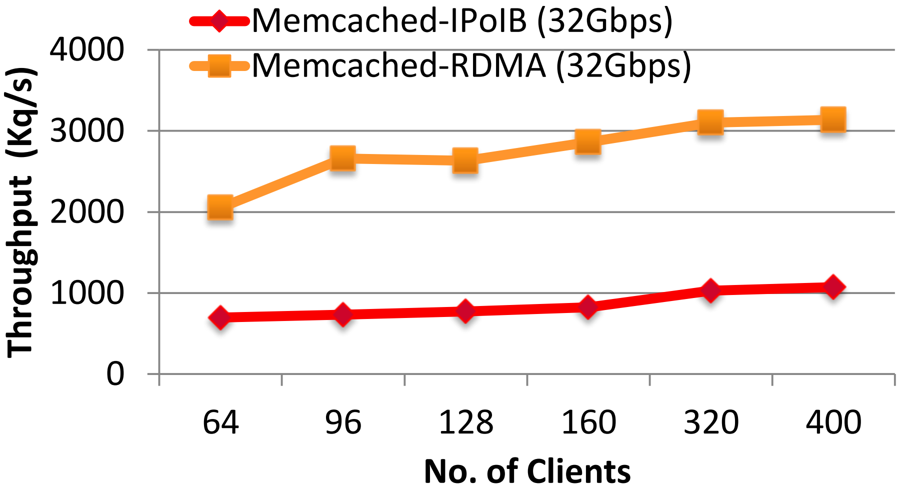




- OHB Latency: End-to-end point-to-point Set/Get latency; Read:Write 90:10 Workload
  - Improves performance by about 71% over Memcached-IPoIB
- YCSB: Three-node Memcached cluster, 64 GB memory/node, 32 compute nodes for clients
  - Improves the overall throughput for Read-Heavy YCSB-B workload by about 5.7X, as compared to default Memcached running over IPoIB

Micro-benchmark Evaluation for OLDP workloads

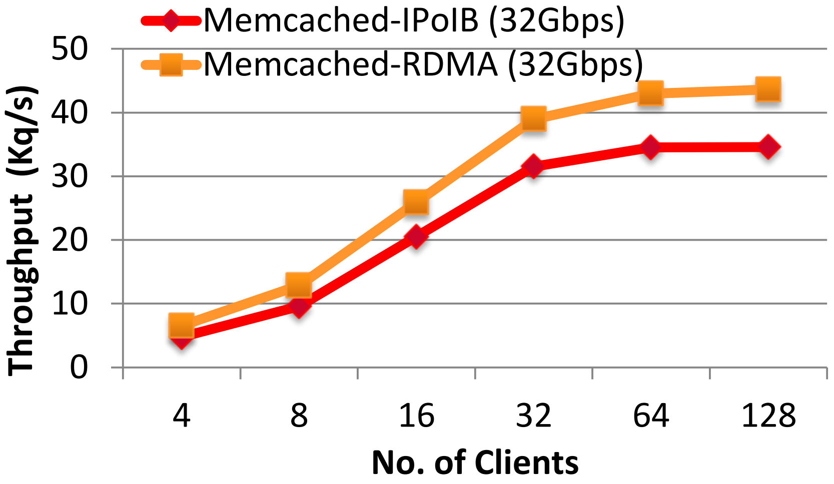


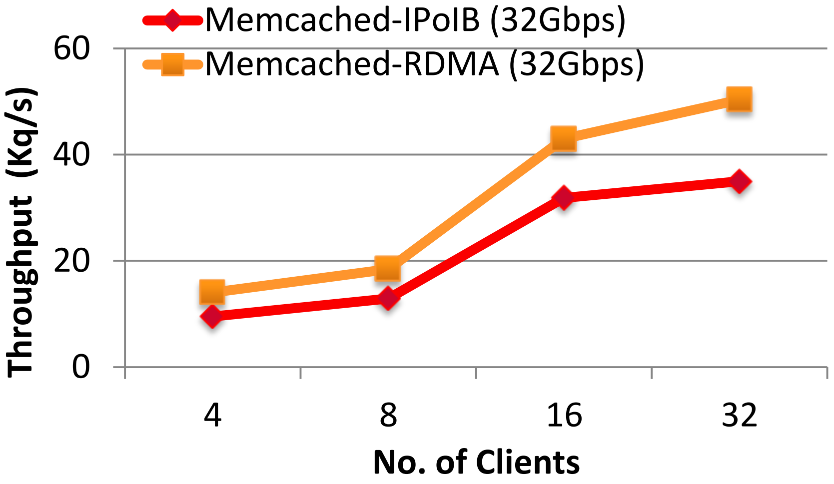


- Illustration with Read-Cache-Read access pattern using modified mysqlslap load testing tool
- RDMA-Memcached can
  - improve query latency by up to 66% over IPoIB (32Gbps)
  - throughput by up to 69% over IPoIB (32Gbps)

D. Shankar, X. Lu, J. Jose, M. W. Rahman, N. Islam, and D. K. Panda, Can RDMA Benefit On-Line Data Processing Workloads with Memcached and MySQL, ISPASS'15

Evaluation with Transactional and Web-oriented Workloads





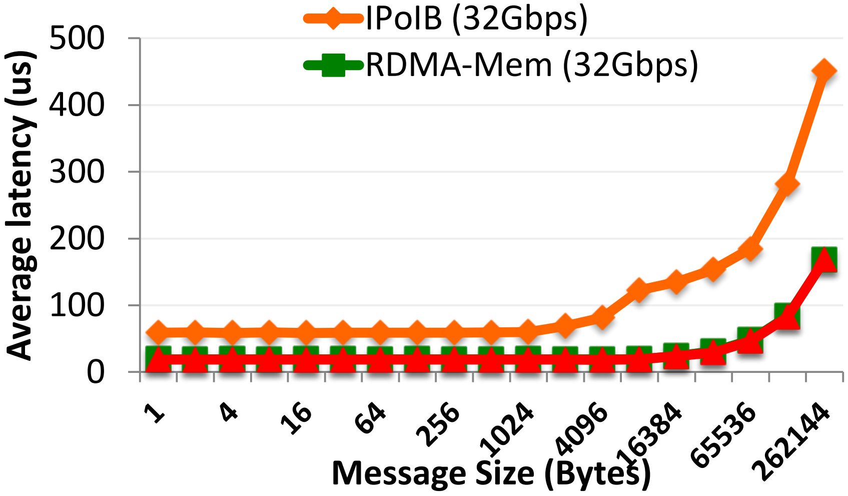
Transactional workloads. Example: TATP

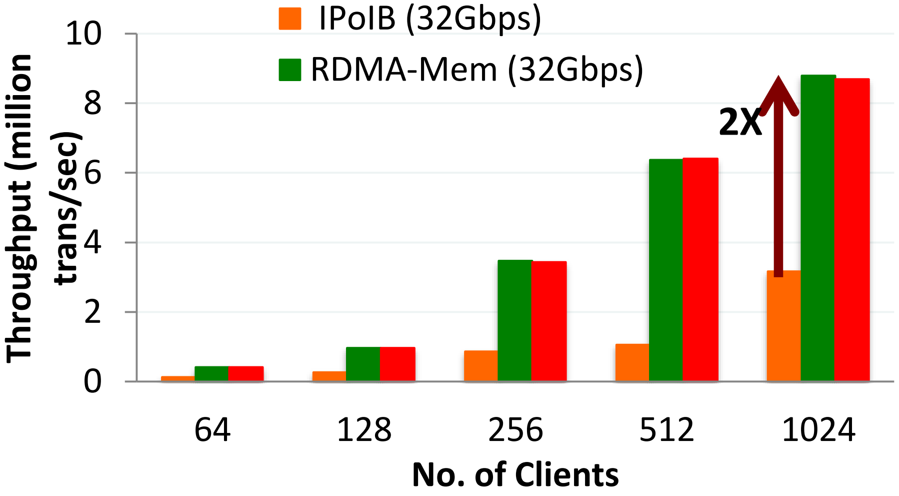
- Up to 29% improvement in overall throughput as compared to default Memcached running over IPoIB

Web-Oriented workloads. Example: Twitter workload

- Up to 42% improvement in overall throughput compared to default Memcached running over IPoIB

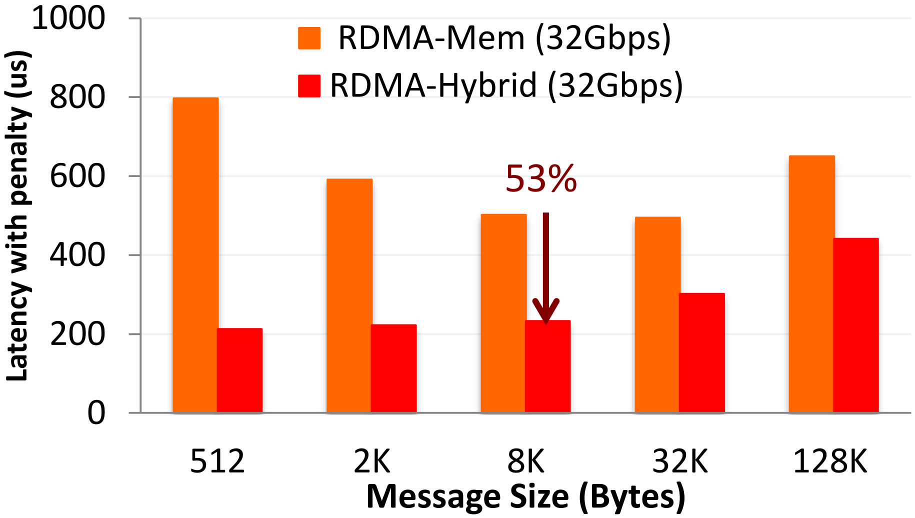
Performance Benefits on SDSC-Gordon – OHB Latency & Throughput Micro-Benchmarks

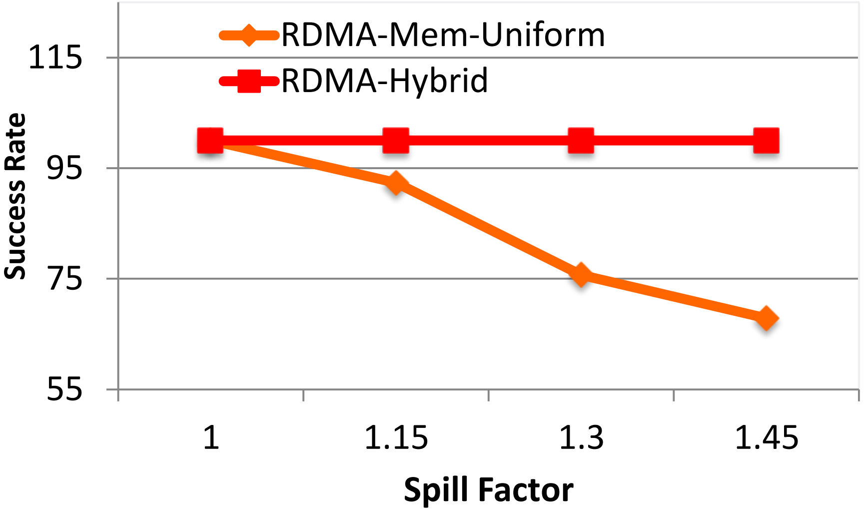




- ohb_memlat & ohb_memthr latency & throughput micro-benchmarks
- RDMA-Memcached can improve query latency by up to 70% over IPoIB and throughput by up to 2X over IPoIB
  - No overhead in using hybrid mode when all data can fit in memory

Performance on OSU-RI-SSD – Hybrid Memcached

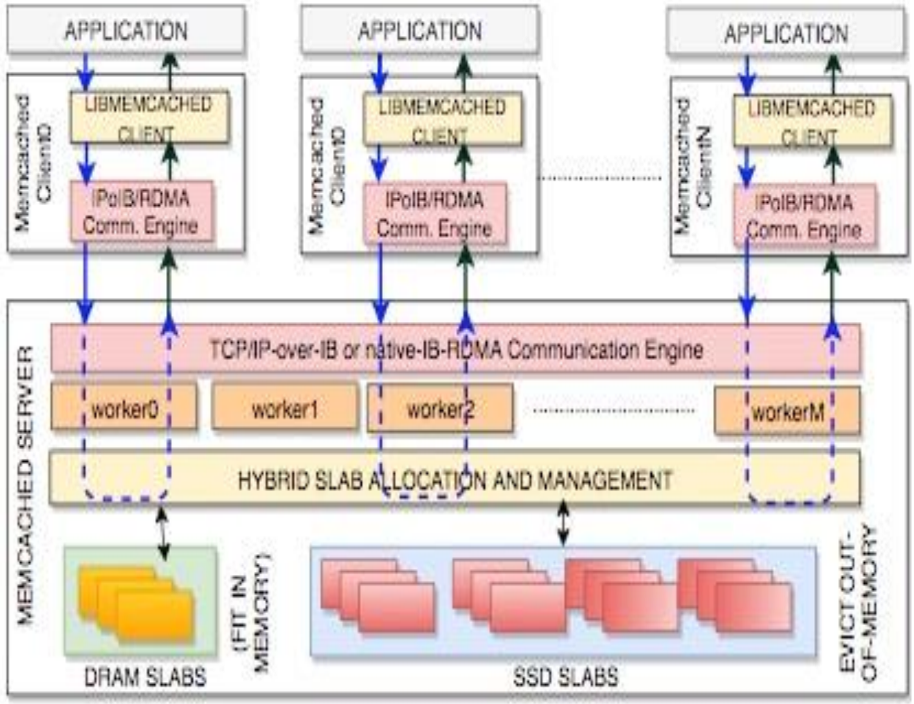




ohb_memhybrid – Uniform Access Pattern, single client and single server with 64MB

- Success Rate of In-Memory Vs. Hybrid SSD-Memory for different spill factors
  - 100% success rate for Hybrid design while that of pure In-memory degrades
- Average Latency with penalty for In-Memory Vs. Hybrid SSD-Assisted mode for spill factor 1.5.
  - up to 53% improvement over In-memory with server miss penalty as low as 1.5 ms

Overview of SSD-Assisted Hybrid RDMA-Memcached

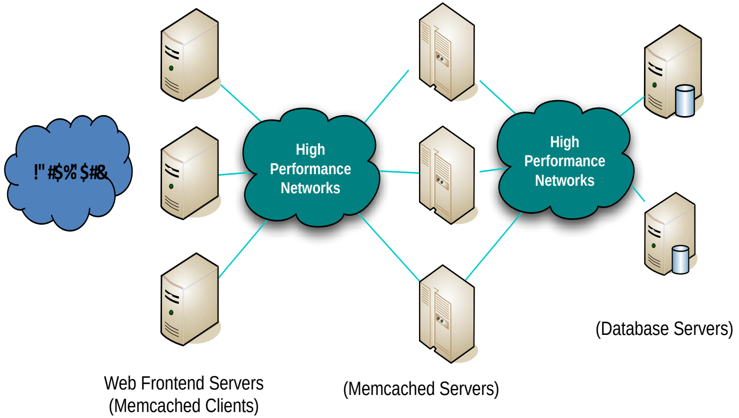


- Hybrid slab allocation and management for higher data retention
- Log-structured sequence of blocks flushed to SSD
- SSD fast random read to achieve low latency object access
- Uses LRU to evict data to SSD

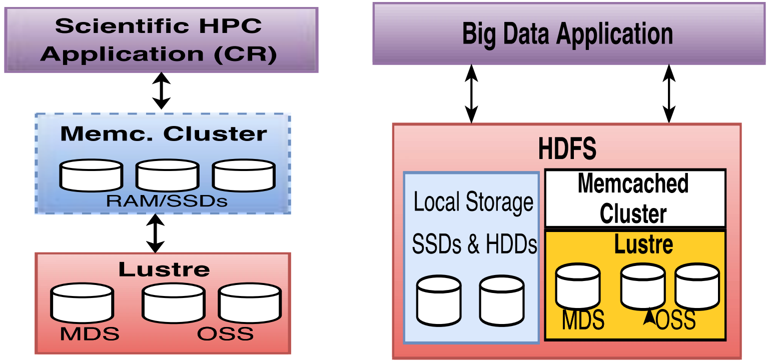
Key-Value Storage in HPC and Data Centers

- General purpose distributed memory-centric storage
  - Allows to aggregate spare memory from multiple nodes (e.g, Memcached)
- Accelerating Online and Offline Analytics in High-Performance Compute (HPC) environments
- Our Basis: Current High-performance and hybrid key-value stores for modern HPC clusters
  - ❖ High-Performance Network Interconnects (e.g., InfiniBand)
    - ❖ Low end-to-end latencies with IP-over-InfiniBand (IPoIB) and Remote Direct Memory Access (RDMA)
  - ❖ 'DRAM+SSD' hybrid memory designs
    - ❖ Extend storage capabilities beyond DRAM capabilities using high-speed SSDs

(Online Analytical Workloads: OLTP/NoSQL Query Cache)



(Offline Analytical Workloads: Software-Assisted Burst-Buffer)



Drivers of Modern HPC Cluster Architectures

High Performance Interconnects

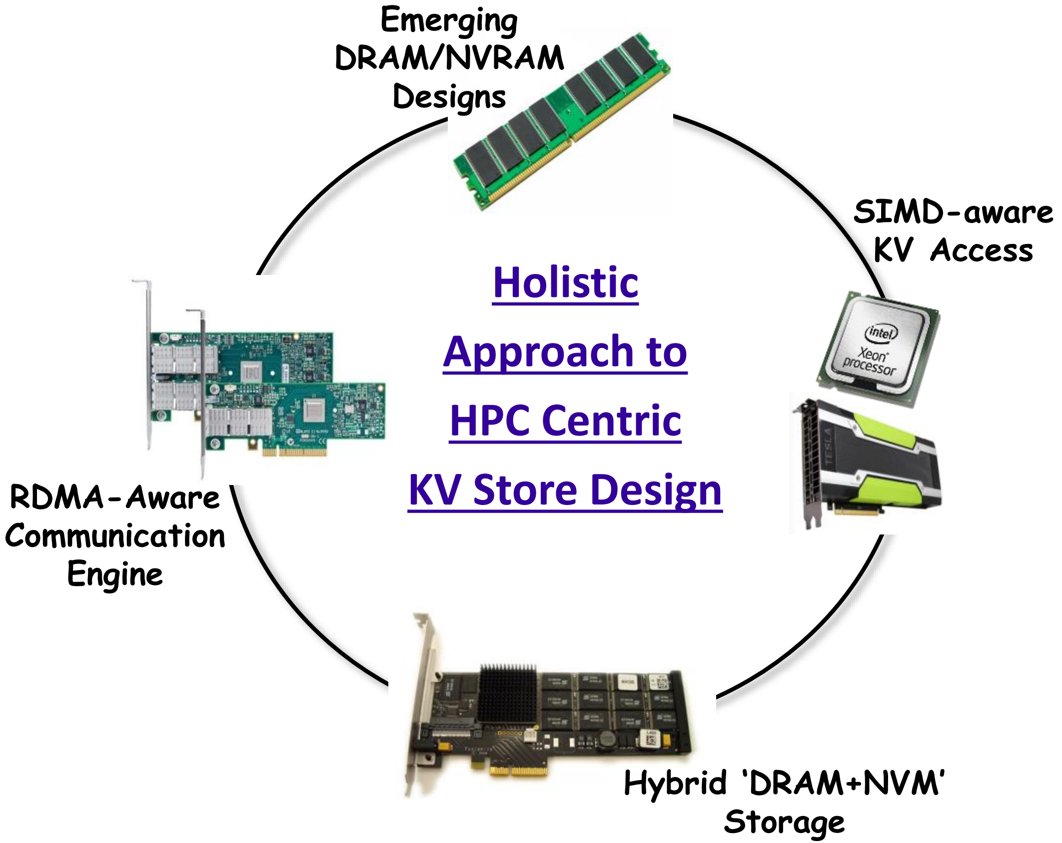
Large Memory Nodes (DRAM)

SSD, NVMe-SSD, NVRAM

Multi-core Processors with vectorization support + Accelerators (GPUs)

- Multi-core/many-core technologies
- Remote Direct Memory Access (RDMA)-enabled networking (InfiniBand and RoCE)
- Solid State Drives (e.g., PCIe/NVMe-SSDs), NVRAM (e.g., PCM, 3DXpoint), Parallel Filesystems (e.g., Lustre)
- Accelerators (e.g., NVIDIA GPGPUs)
- Production-scale HPC Clusters: SDSC Comet, TACC Stampede, OSC Owens, etc.

Designing a High-Performance, Resilient and Heterogeneity-Aware KV Storage



Emerging DRAM/NVRAM Designs

SIMD-aware KV Access

RDMA-Aware Communication Engine

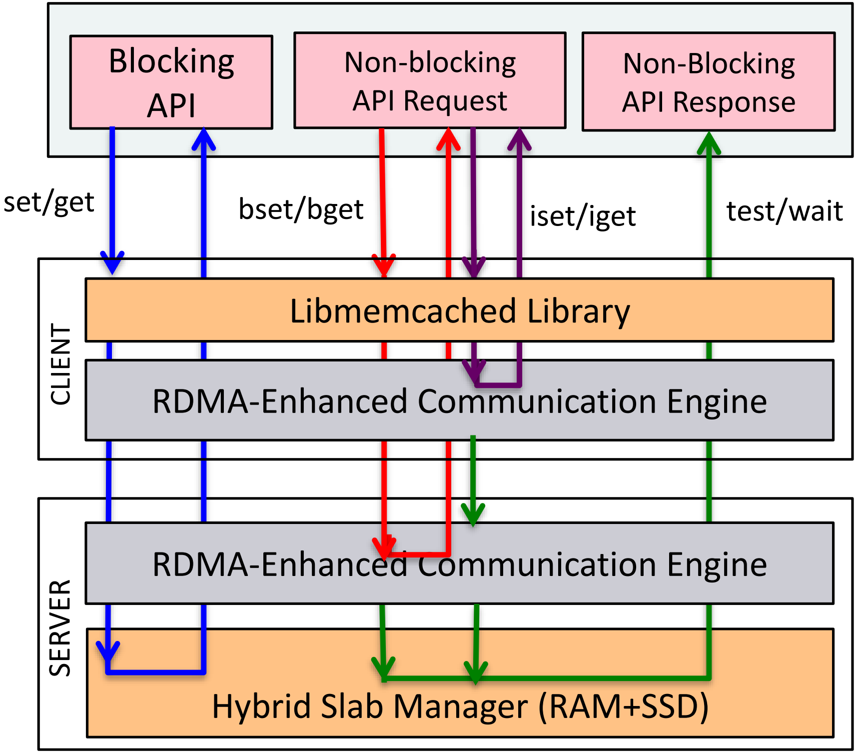
Holistic Approach to HPC Centric KV Store Design

Hybrid 'DRAM+NVM' Storage

- Current and emerging HPC systems
- Goals:
  - Maximize end-to-end performance
  - Exploit all HPC resources (compute/storage/network)
  - Enable HPC Big Data applications to leverage memory-centric KV storage

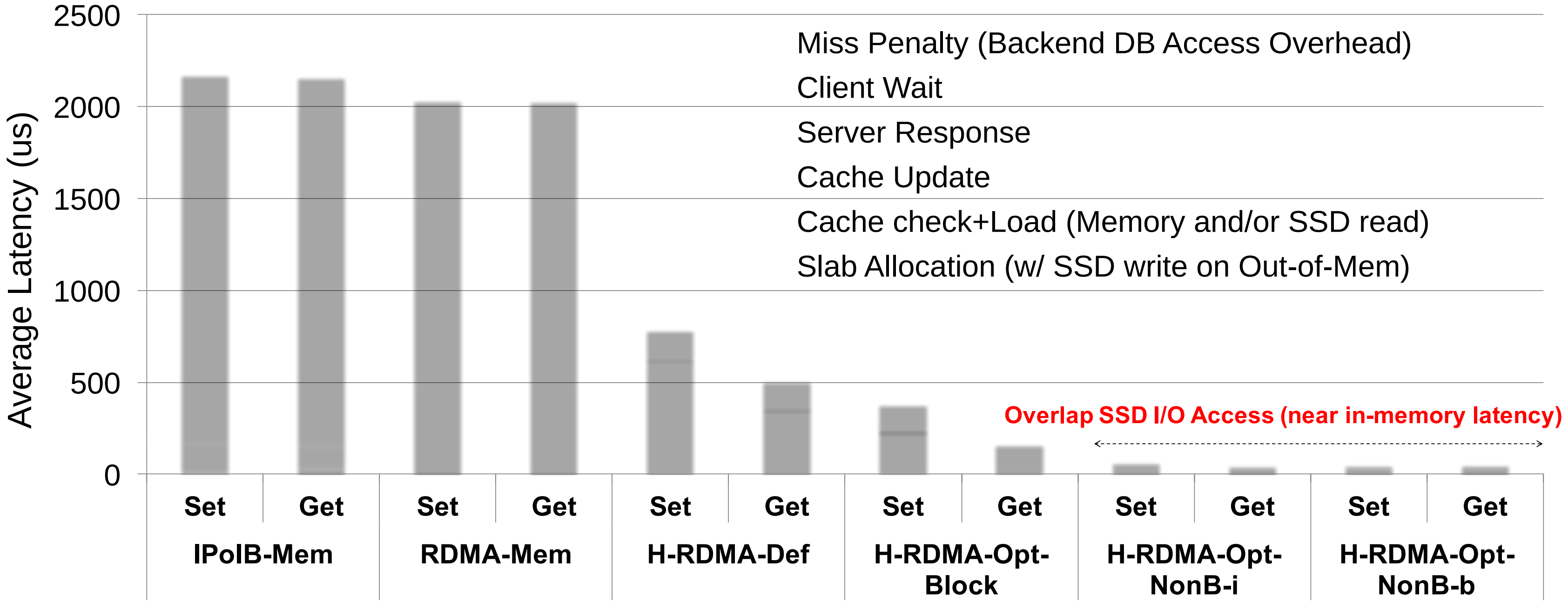
High-Performance Non-Blocking API Semantics

- Heterogeneous Storage-Aware Key-Value Stores (e.g., 'DRAM + PCIe/NVMe-SSD')
  - Higher data retention at the cost of SSD I/O; suitable for out-of-memory scenarios
  - Performance limited by Blocking API semantics
- Goals: Achieve near in-memory speeds while being able to exploit hybrid memory
- Approach: Novel Non-blocking API Semantics to extend RDMA-Libmemcached library
  - memcached_(iset/iget/bset/bget) APIs for SET/GET
  - memcached_(test/wait) APIs for progressing communication



D. Shankar, X. Lu , N. Islam , M. W. Rahman , and D. K. Panda, "High-Performance Hybrid Key-Value Store on Modern Clusters with RDMA Interconnects and SSDs: Non-blocking Extensions, Designs, and Benefits", IPDPS 2016

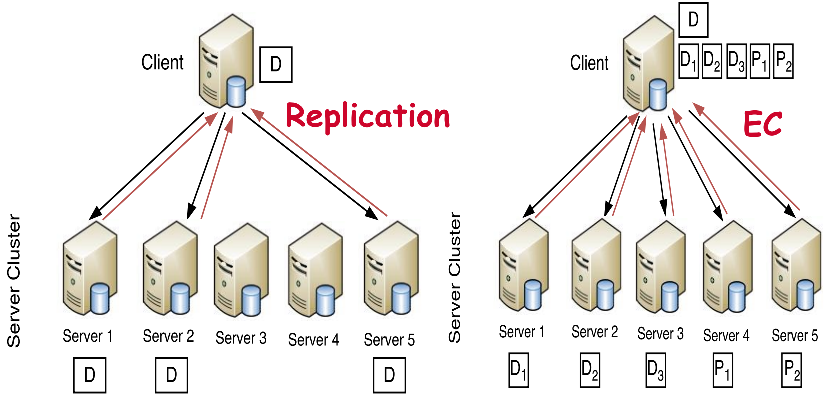
High-Performance Non-Blocking API Semantics



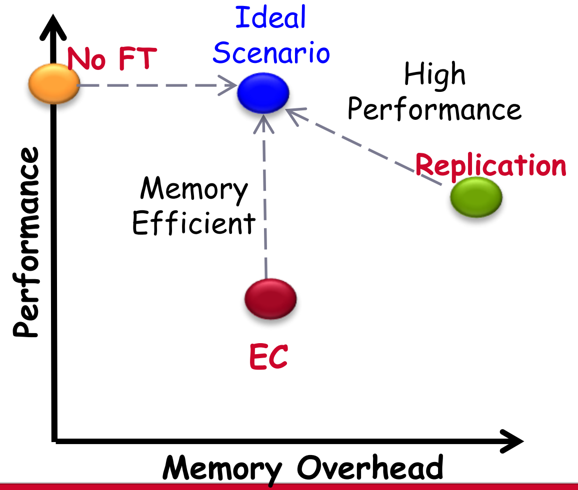
- Set/Get Latency with Non-Blocking API: Up to 8x gain in overall latency vs. blocking API semantics over RDMA+SSD hybrid design
- Up to 2.5x gain in throughput observed at client; Ability to overlap request and response phases to hide SSD I/O overheads

Fast Online Erasure Coding with RDMA

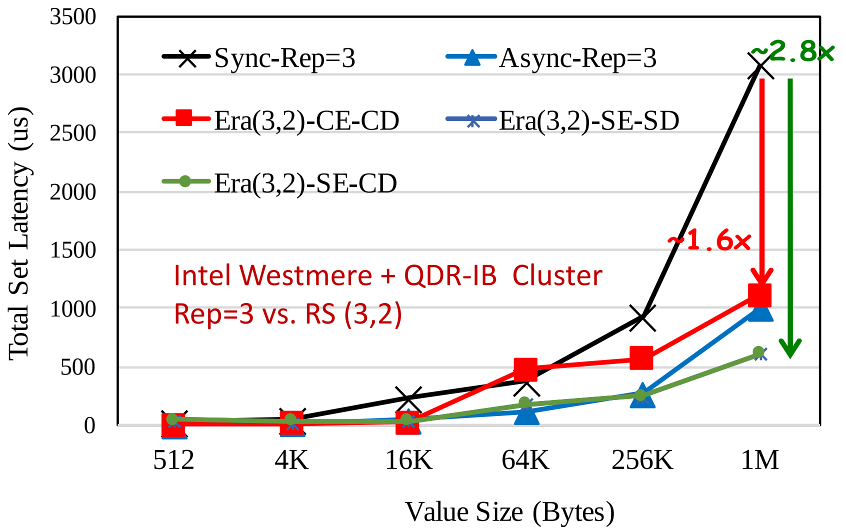
- Erasure Coding (EC): A low storage-overhead alternative to Replication
- Bottlenecks for Online EC:
  - Compute Overhead: Encoding/Decoding
  - New communication overhead: Scatter/ Gather distributed data/parity chunks per-KV request
- Goal: Making Online EC viable for key-value stores
- Approach: Non-blocking RDMA-aware semantics to enable compute/communication overlap
- Encode/Decode offload capabilities integrated into Memcached client (CE/CD) and server (SE/SD)



E.g., Storage Overhead
66% for Reed-Solomon EC vs. 200% of Rep=3

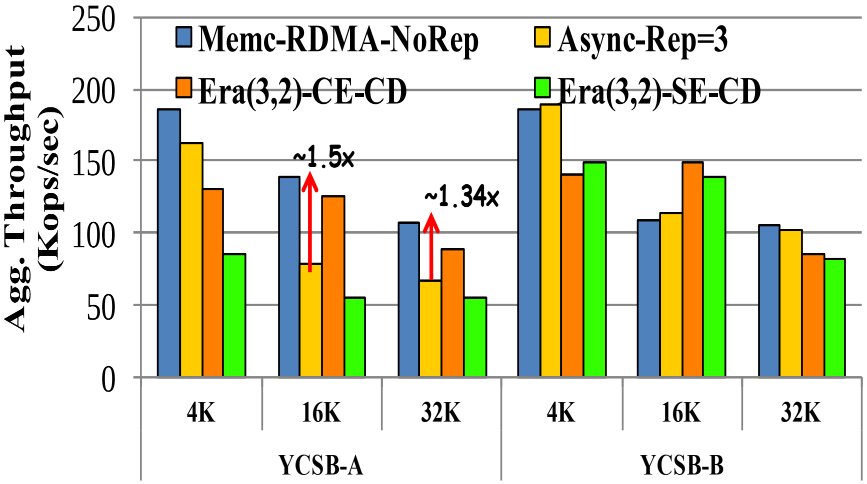


Fast Online Erasure Coding with RDMA



Intel Westmere + QDR-IB Cluster

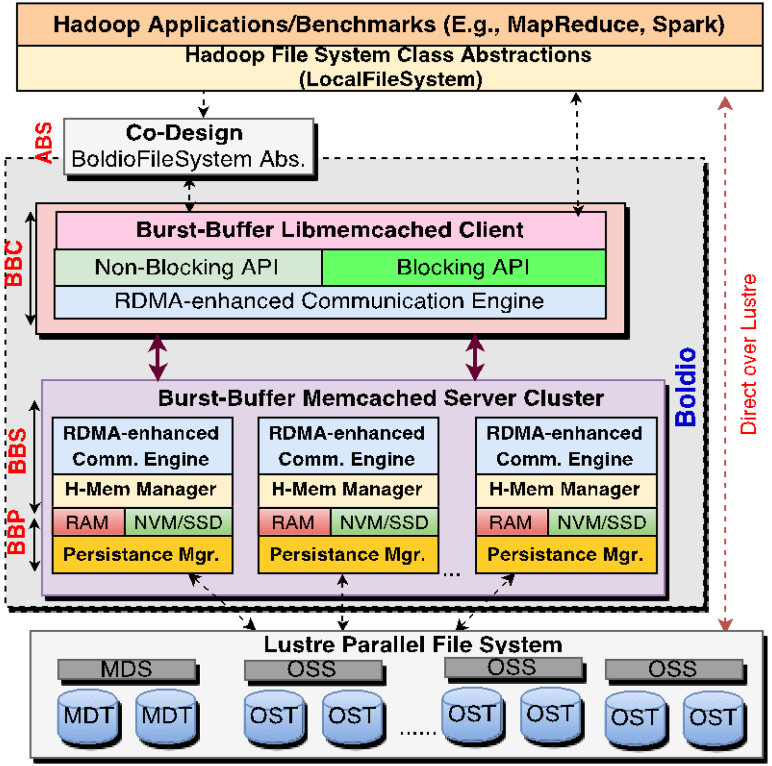
Rep=3 vs. RS (3,2)



- Experiments with YCSB for Online EC vs. Async. Rep:
  - 150 Clients on 10 nodes on SDSC Comet Cluster (IB FDR + 24-core Intel Haswell) over 5-node RDMA-Memcached Cluster
  - (1) CE-CD gains ~1.34x for Update-Heavy workloads; SE-CD on-par (2) CE-CD/SE-CD on-par for Read-Heavy workloads

D. Shankar , X. Lu , and D. K. Panda, "High-Performance and Resilient Key-Value Store with Online Erasure Coding for Big Data Workloads", 37th International Conference on Distributed Computing Systems (ICDCS 2017)

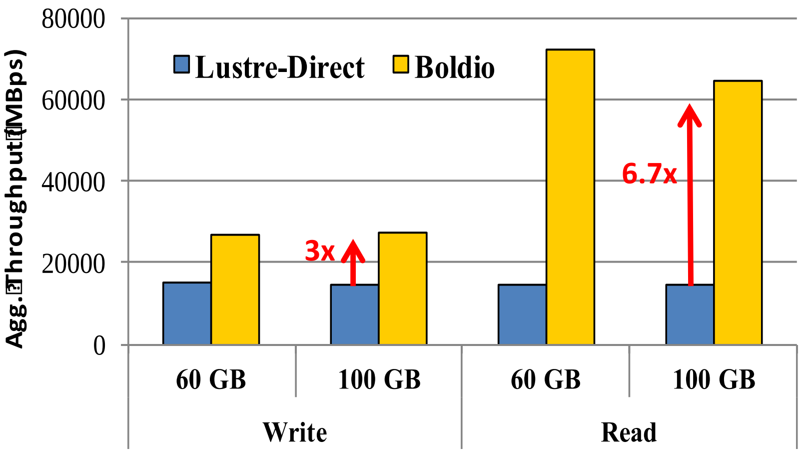
Co-Designing Key-Value Store-based Burst Buffer over PFS

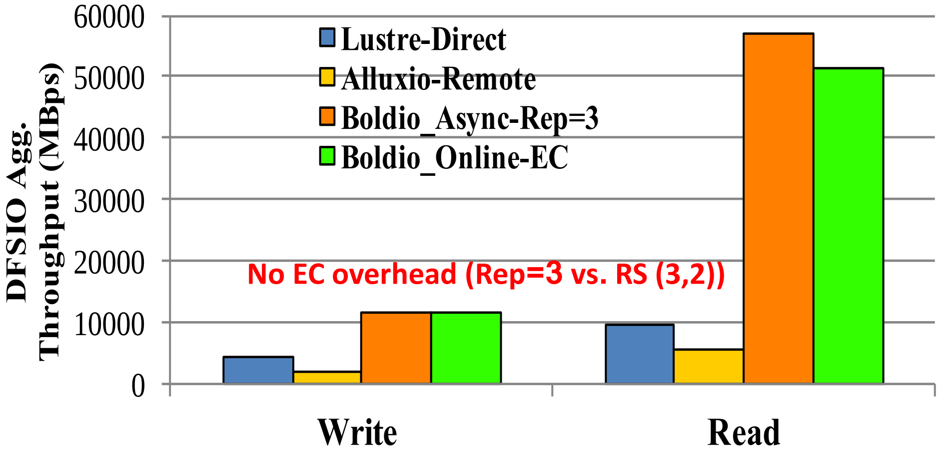


- Offline Data Analytics Use-Case: Hybrid and resilient key-value store-based Burst-Buffer system Over Lustre (Boldio)
- Overcome local storage limitations on HPC nodes; performance of `data locality’
- Light-weight transparent interface to Hadoop/ Spark applications
- Accelerating I/O-intensive Big Data workloads
  - Non-blocking RDMA-Libmemcached APIs to maximize overlap
  - Client-based replication or Online Erasure Coding with RDMA for resilience
  - Asynchronous persistence to Lustre parallel file system at RDMA-Memcached Servers

D. Shankar, X. Lu, D. Panda, Boldio: A Hybrid and Resilient Burst-Buffer over Lustre for Accelerating Big Data I/O, IEEE International Conference on Big Data 2016 (Short Paper)

Co-Designing Key-Value Store-based Burst Buffer over PFS





- TestDFSIO on SDSC Gordon Cluster (16-core Intel Sandy Bridge and IB QDR) with 16-node MapReduce Cluster + 4-node Boldio Cluster
- Boldio can sustain 3x and 6.7x gains in read and write throughputs over stand-alone Lustre

- TestDFSIO on Intel Westmere Cluster (8-core Intel Sandy Bridge and IB QDR); 8-node MapReduce Cluster + 5-node Boldio Cluster over Lustre
- Performance gains over designs like Alluxio (formerly Tachyon) in HPC environments with no local storage

The High-Performance Big Data (HiBD) Project

- RDMA for Apache Spark (RDMA-Spark), Apache Hadoop 2.x (RDMA-Hadoop-2.x), RDMA for Apache HBase
- RDMA for Memcached (RDMA-Memcached)
  - RDMA-aware `DRAM+SSD' hybrid Memcached server design
  - Non-Blocking RDMA-based Client API designs (RDMA-Libmemcached)
  - Based on Memcached 1.5.3 and Libmemcached client 1.0.18
  - Available for InfiniBand and RoCE
- OSU HiBD-Benchmarks (OHB)
  - Memcached Set/Get Micro-benchmarks for Blocking and Non-Blocking APIs, and Hybrid Memcached designs
  - YCSB plugin for RDMA-Memcached
  - Also includes HDFS, HBase, Spark Micro-benchmarks
- http://hibd.cse.ohio-state.edu
- Users Base: 290 organizations, 34 countries, 28,250 downloads

HiBD
High-Performance
Big Data

Network Based Computing
Laboratory

The Ohio State
University

Concluding Remarks

- Presented an overview of Web 2.0 Memcached architecture
- Provided an overview of modern cluster networking technologies and key-value store use-cases
- Presented RDMA for Memcached software under HiBD Project: (1) RDMA-based Communication Engine for InfiniBand clusters; (2) Hybrid design with high-speed SSDs; (3) Non-blocking API extensions to RDMA-LibMemcached
- Presented (1) Fast Online Erasure Coding Resilience with RDMA; (2) Key-Value Store-based Burst-Buffer use-case for Offline Hadoop-based Analytics
- Enabling Big Data processing community to take advantage of modern HPC technologies to carry out their analytics in a fast and scalable manner

Conclusion & Future Avenues

- Holistic approach to designing key-value storage by exploiting the capabilities of HPC clusters for (1) performance, (2) scalability, and, (3) data resilience/availability
- RDMA-capable Networks: (1) Proposed Non-blocking RDMA-based Libmemcached APIs (2) Fast Online EC-based RDMA-Memcached designs
- Heterogeneous Storage-Awareness: (1) Leverage `RDMA+SSD’ hybrid designs, (2) `RDMA+NVRAM’ Persistent Key-Value Storage
- Application Co-Design: Memory-centric data-intensive applications on HPC Clusters
  - Online (e.g., SQL query cache, YCSB) and Offline Data Analytics (e.g., Boldio Burst-Buffer for Hadoop I/O)
- Future Work: Ongoing work in this thesis direction
  - Heterogeneous compute capabilities of CPU/GPU: End-to-end SIMD-aware KVS designs
  - Exploring co-design of (1) Read-intensive Graph-based workloads (E.g., LinkBench, RedisGraph) (2) Key-value storage engine for ML Parameter Server frameworks