15th ANNUAL WORKSHOP 2019

SCALABLE, RESILIENT, AND DISTRIBUTED KEY-VALUE STORE-BASED DATA MANAGEMENT OVER RDMA NETWORKS

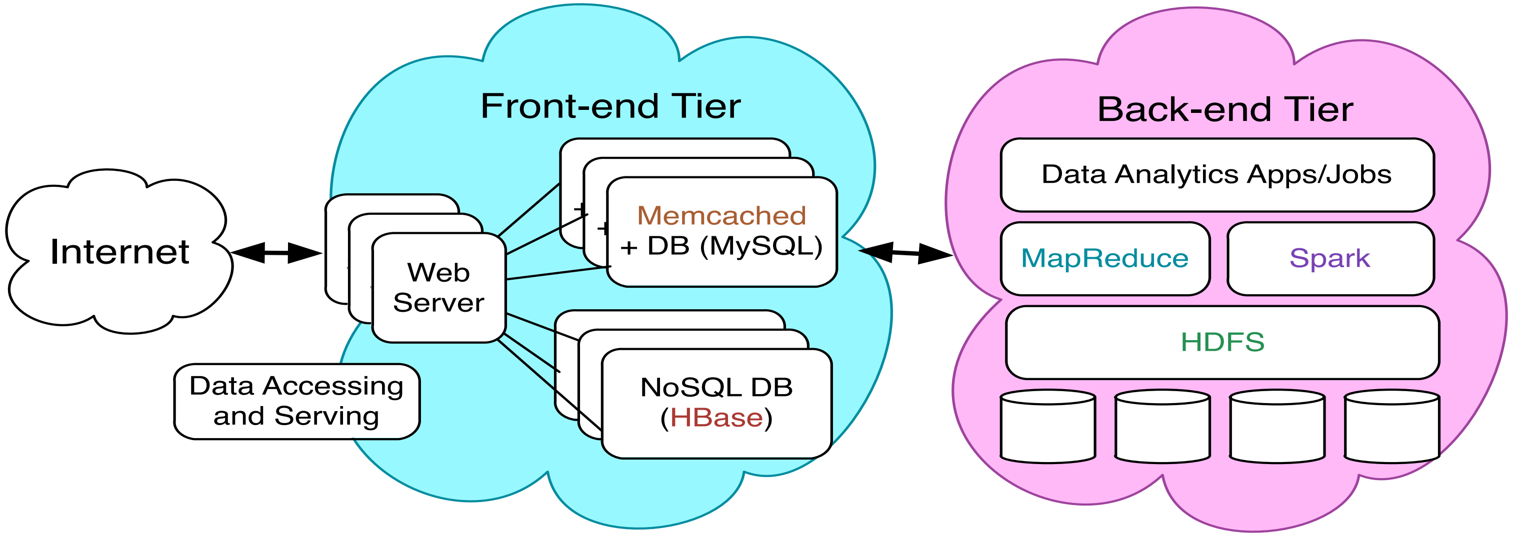
Xiaoyi Lu, Dhabaleswar K. (DK) Panda

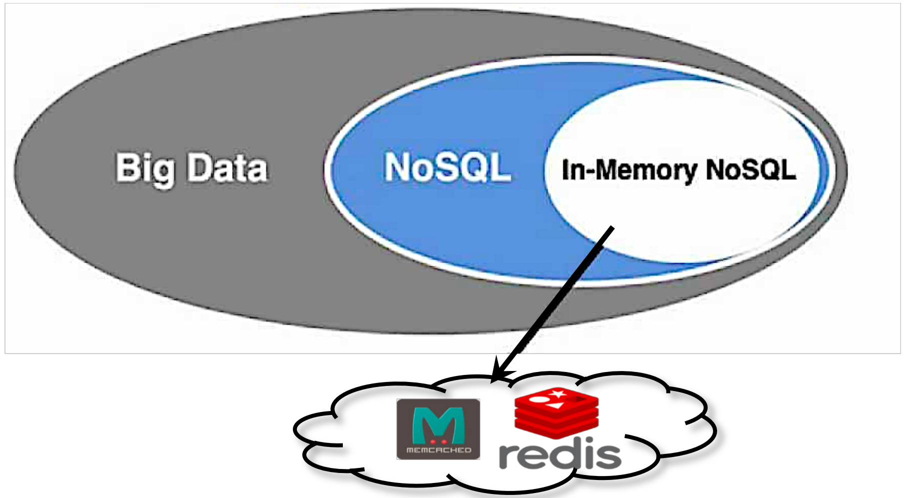
The Ohio State University

[March 20, 2019]

E-mail: {luxi, panda}@cse.ohio-state.edu
http://www.cse.ohio-state.edu/~luxi
http://www.cse.ohio-state.edu/~panda

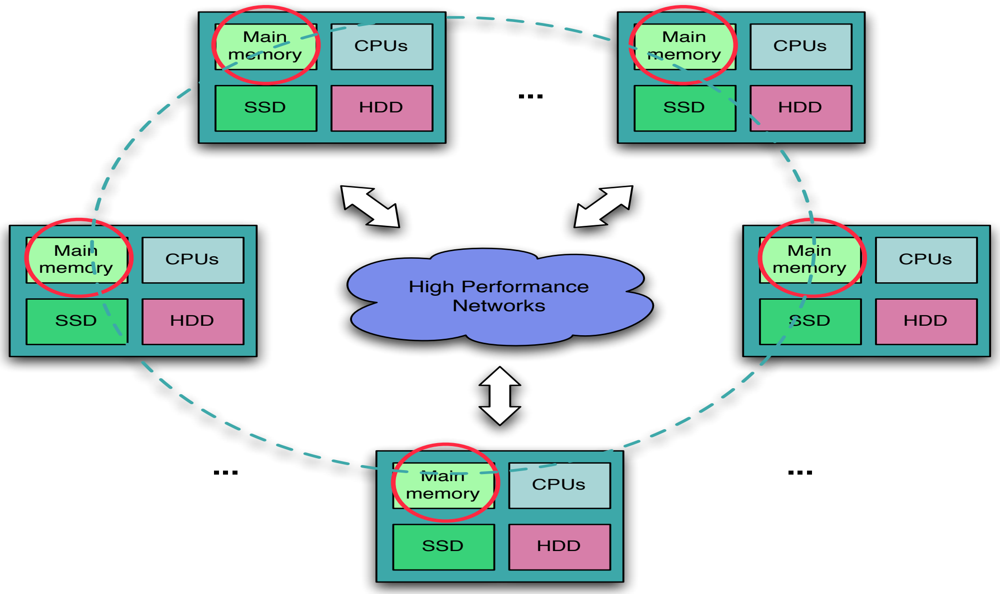
BIG DATA AND IN-MEMORY COMPUTING

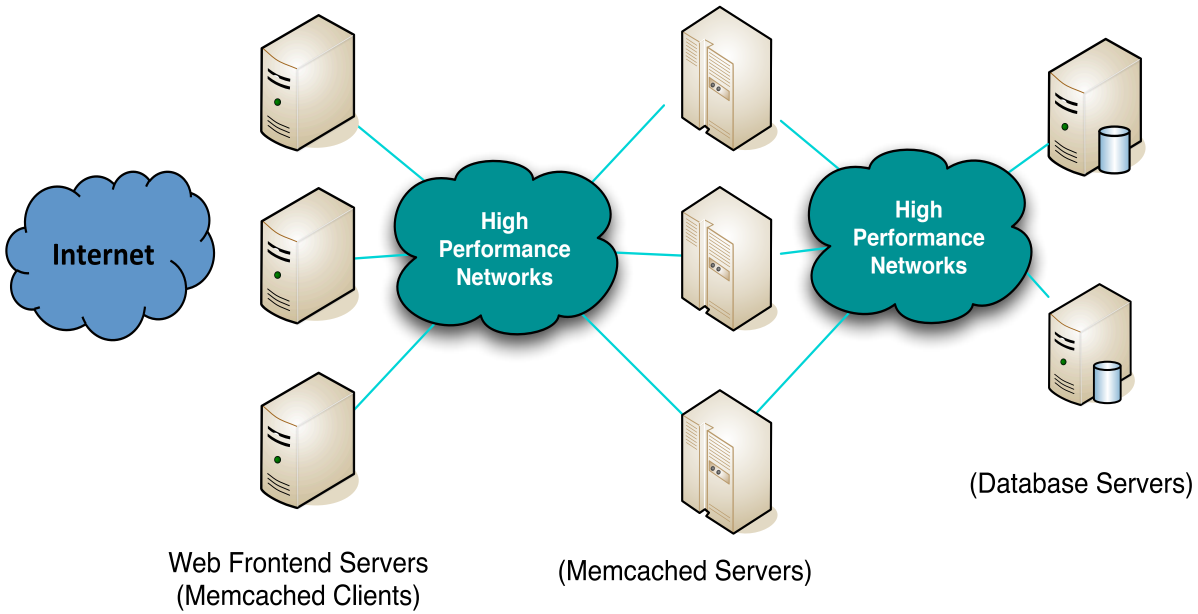




- **Growing popularity of in-memory computing systems for Big Data on HPC and Data Centers**
  - E.g., Spark, Alluxio, Apache Ignite
- **Emergence of memory-centric storage to complement in-memory computing**

DISTRIBUTED KEY-VALUE STORES

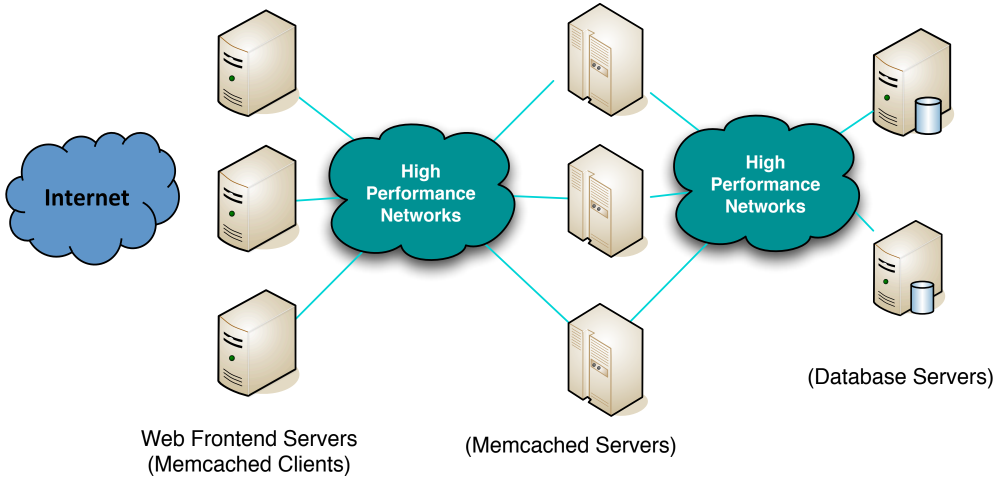




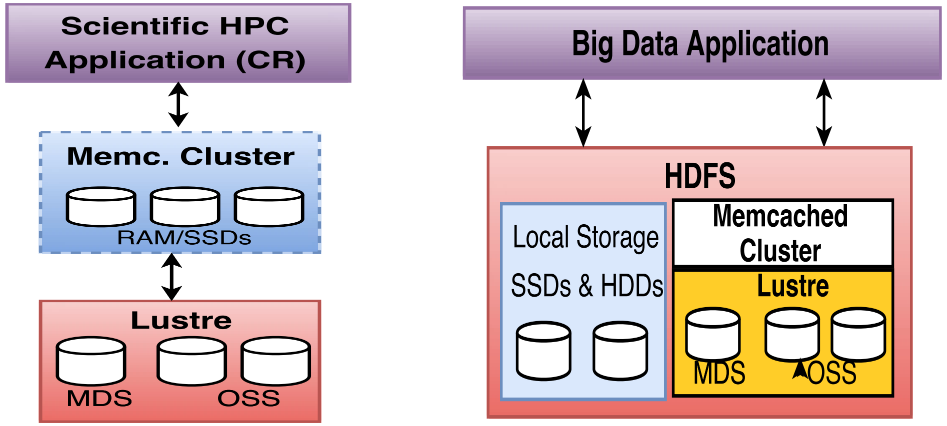
- **Generic distributed memory-centric data storage system; objects stored as (key, value) pairs**
  - E.g., Memcached, Redis
- **Distributed caching layer for**
  - Database queries and results of API calls (e.g., Memcached + MySQL for OLTP),
  - Graph nodes/associations for search (e.g., Facebook TAO)
- **Scalable model, but typical usage very network intensive**

CURRENT AND EMERGING BIG DATA KV STORE WORKLOADS

Online Analytical Workloads: SQL/Graph DB query cache

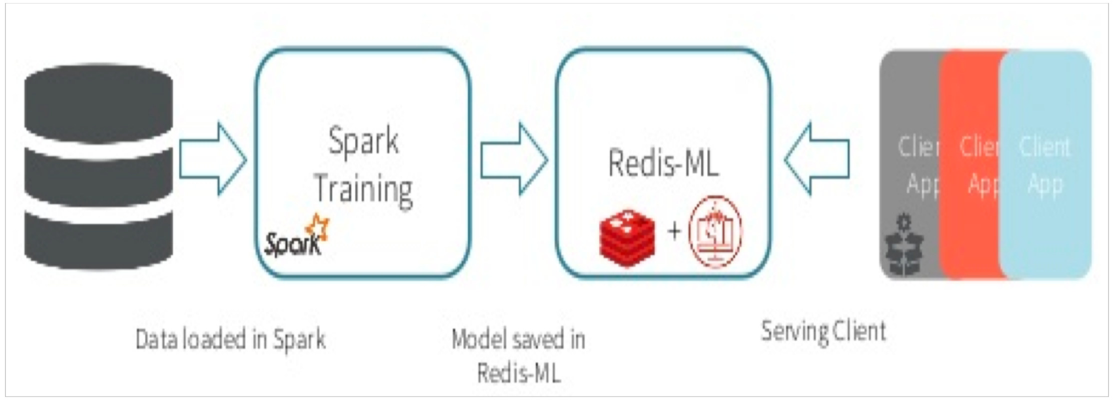


Offline Analytical Workloads: I/O Staging and Caching

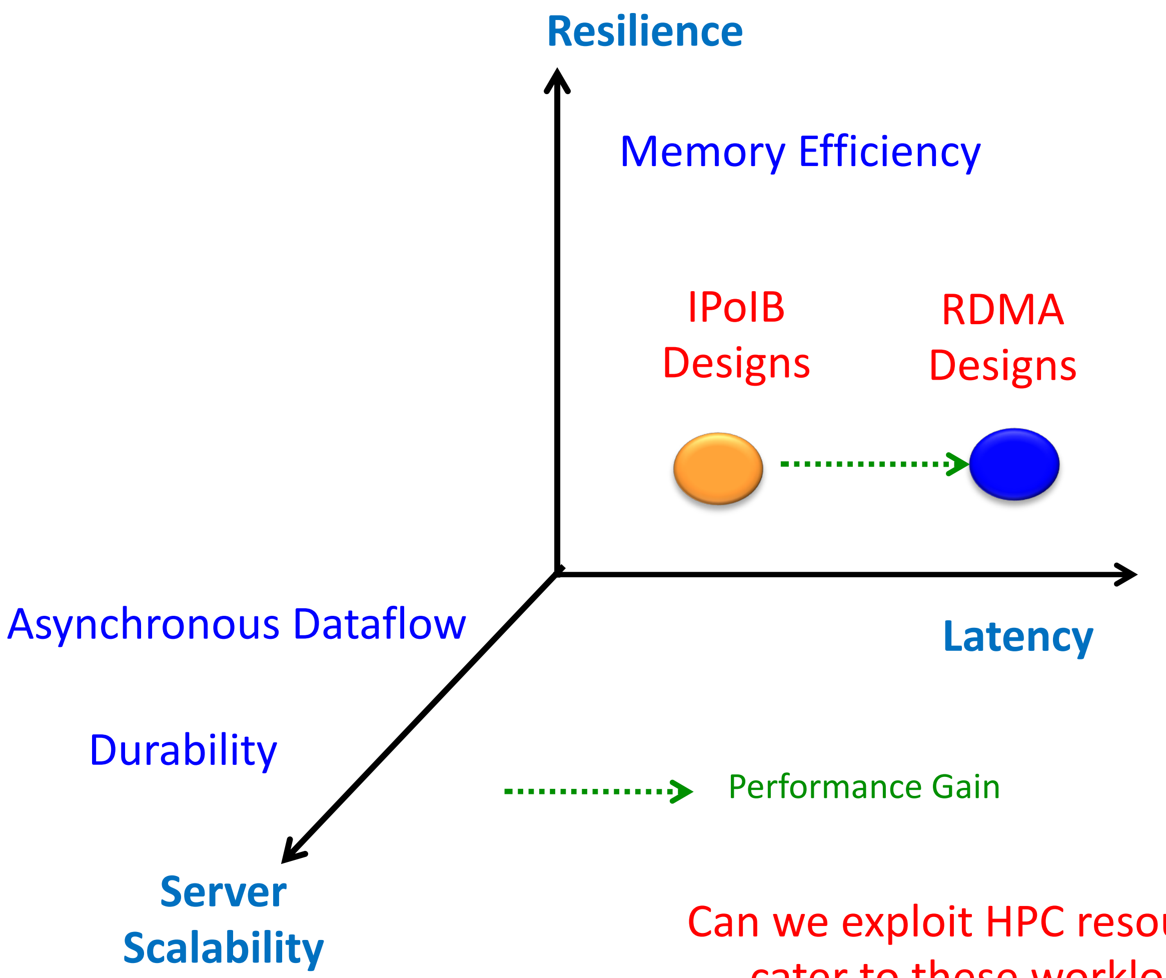


Real-time Analytical Workloads: Pub/Sub Messaging Service

Machine Learning Workloads: E.g., Model Server for Inferencing



NEEDS OF EMERGING KV-CENTRIC WORKLOADS



Can we exploit HPC resources to cater to these workloads?

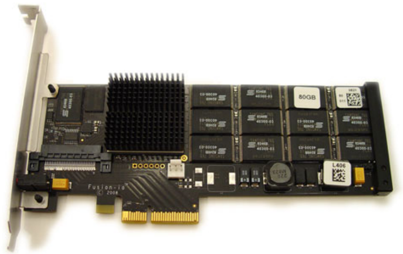
- **What do these new and diverse workloads need? Can we support them in the best manner?**
  - Low latencies still vital
  - Scalable servers
    - Total Agg. Throughput
  - Non-Blocking API Semantics
  - Memory-Efficient Resilience
    - Fault Tolerance / Data Availability
  - High-Performance Durable Stores

DRIVERS OF MODERN HPC CLUSTER AND DATA CENTER ARCHITECTURE

Multi-/Many-core Processors

High Performance Interconnects – InfiniBand (with SR-IOV)
<1usec latency, 200Gbps Bandwidth>

Accelerators / Coprocessors high compute density, high performance/watt
>1 TFlop DP on a chip



SSD, NVMe-SSD, NVRAM

- **Multi-core/many-core technologies**
- **Accelerators (NVIDIA GPGPUs and FPGAs)**
- **Remote Direct Memory Access (RDMA)-enabled networking (InfiniBand and RoCE)**
- **Solid State Drives (SSDs), NVM, Parallel Filesystems, Object Storage Clusters**

SDSC Comet

TACC Stampede

Microsoft Azure

amazon web services EC2

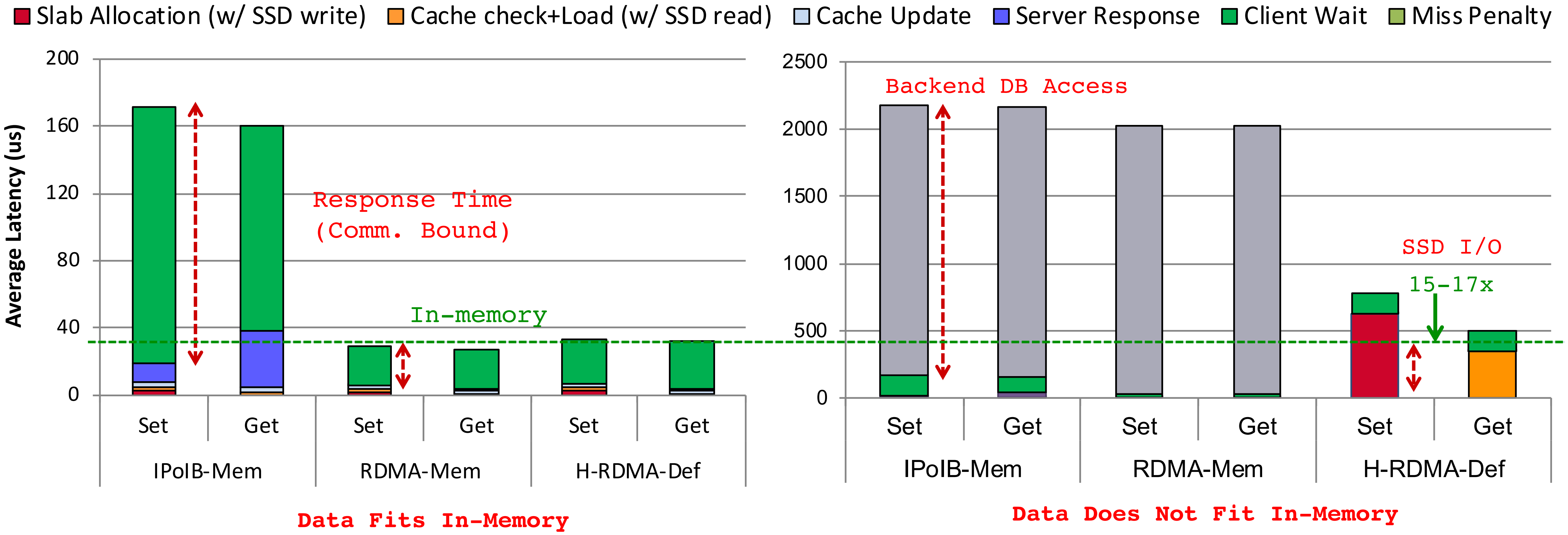
ORACLE Cloud

Chameleon Cloud

OUTLINE

- **Three Approaches**
  - High-Performance RDMA-aware Non-Blocking APIs
  - Fast Online Erasure Coding with RDMA
  - Co-Designing Key-Value Store-based Burst Buffer over PFS
- **Software Release & Impact**
- **Conclusion**

NEEDS OF RDMA AND ASYNCHRONOUS DATAFLOW FOR KVS

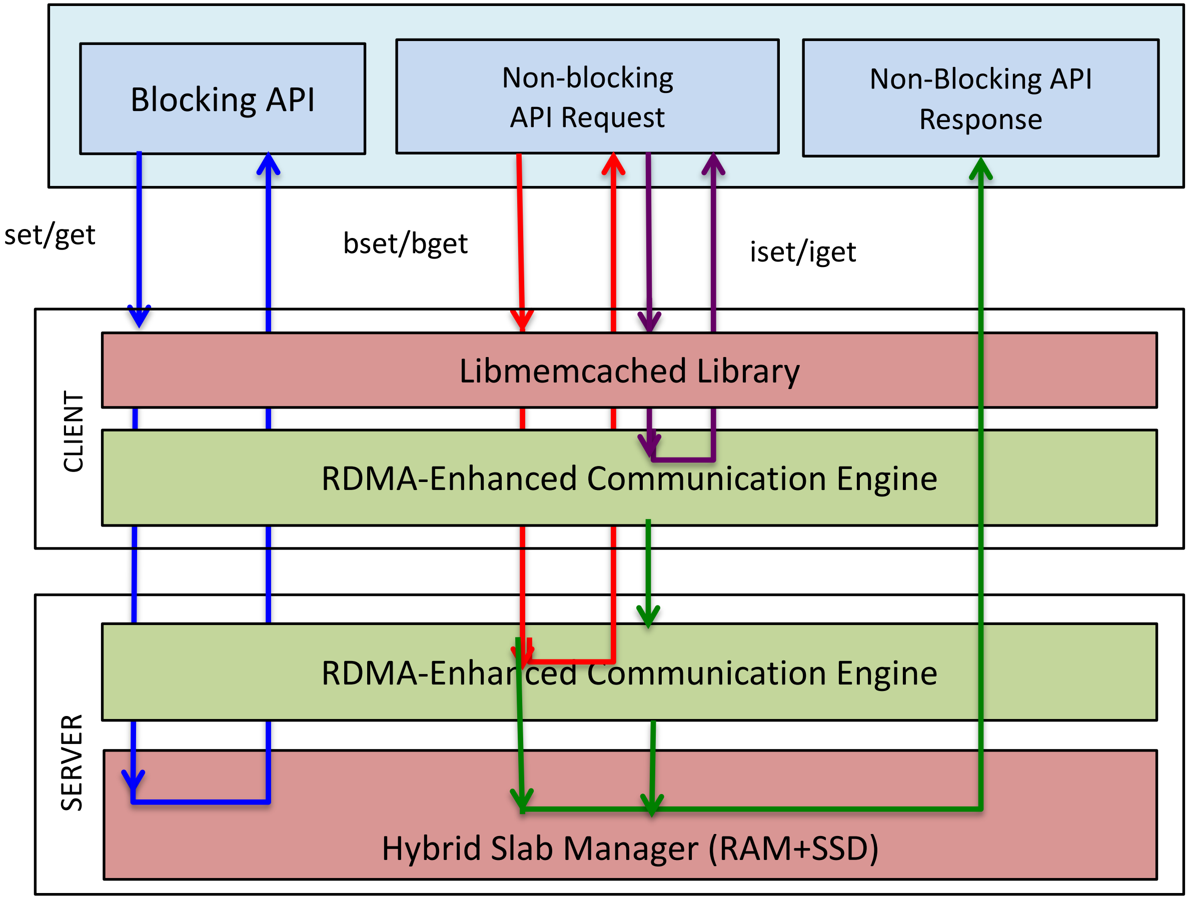


- Single Memcached Client-Server (Pt-2-Pt)
  - 1.5 GB workload + 32 KB KV size
  - Miss penalty of 2ms for Memcached miss

- No backend overhead with hybrid memory design
- Performance degradation of 15-17x due to SSD access vs. In-Memory

RDMA-AWARE NON-BLOCKING SET/GET API SEMANTICS

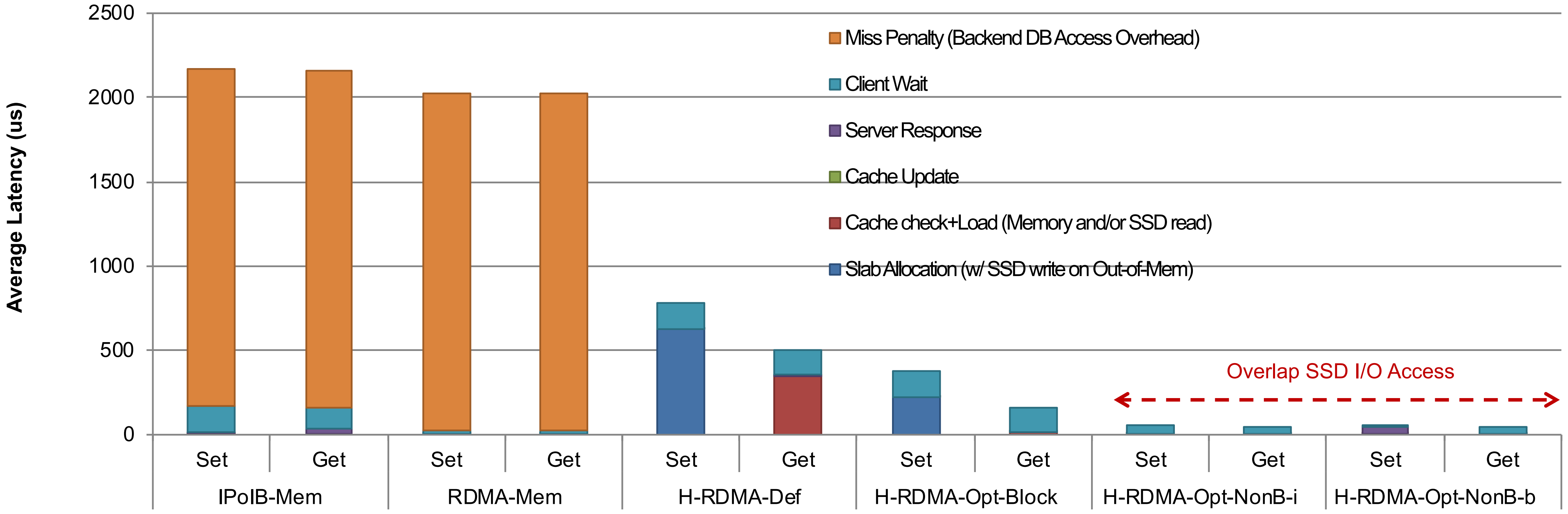
- **Non-blocking RDMA-aware Set/Get Semantics**
  - **memcached_(bset/bget)**
    - Offload to server communication engine
    - Re-use buffer at client
  - **memcached_(iset/iget)**
    - Offload to client communication engine (async. bg progress thread)
    - Need to track client buffers in-use
- **Supporting Non-Blocking Requests: memcached_req**
  - Completion flag (request progress)
  - Server response information
  - Client's 'key' and 'value' buffers
- **Progressing all offloaded requests; with per-request monitoring**
  - **memcached_test**
    - Non-blocking monitor (SUCCESS/FAIL/IN-PROGRESS)
  - **memcached_wait**
    - Blocking monitor (SUCCESS/FAIL)



(RDMA-Libmemcached library extensions)

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

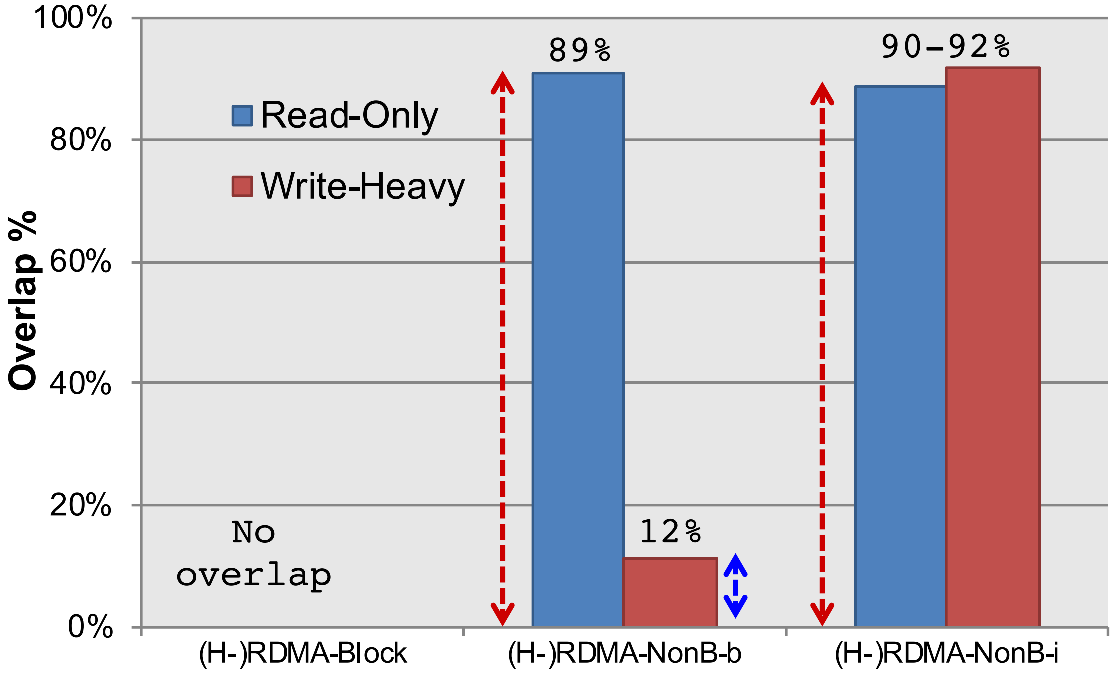
PT-TO-PT ON SDSC COMET: DATA DOES NOT FIT IN-MEMORY



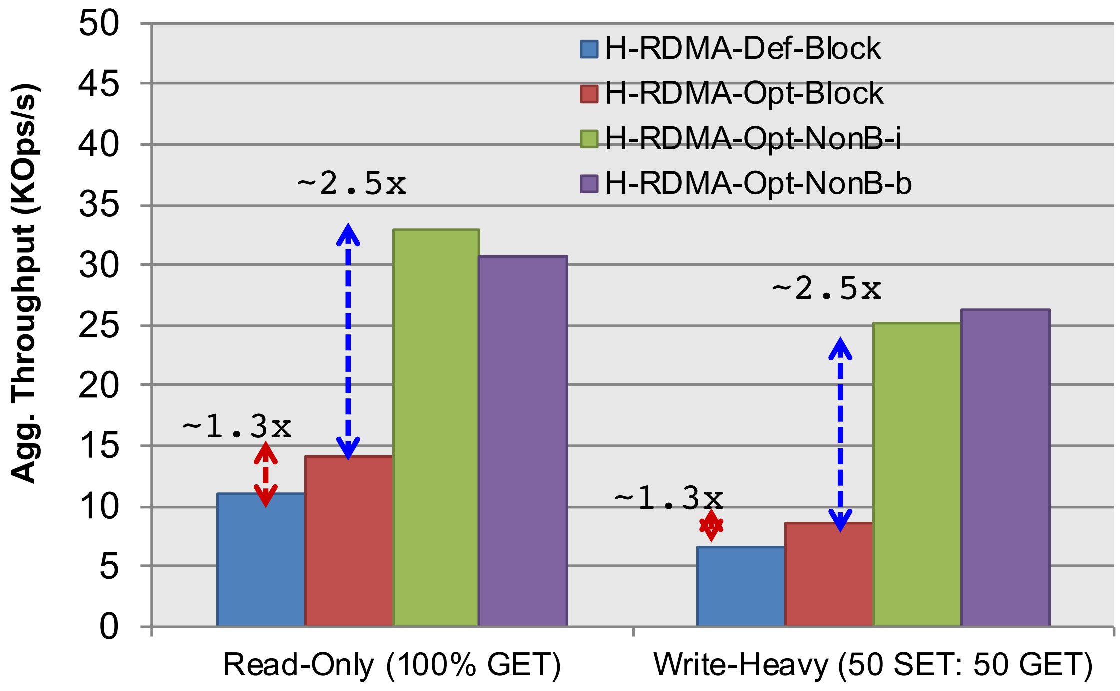
- Up to 8x gain in overall latency vs. blocking API semantics over RDMA+SSD hybrid design
- Ability to overlap request and response phases to hide SSD I/O overheads

H = Hybrid Memcached over SATA SSD
Opt = Adaptive slab manager
Block = Default Blocking API
NonB-i = Non-blocking iset/iget API
NonB-b = Non-blocking bset/bget API w/ buffer re-use guarantee

PERFORMANCE AND SCALABILITY ON SDSC COMET



- 1 Memcached server/client
- Pre-loaded with 1.5 GB data + 32 KB KV size
- Read-Only (100% read) and Write Heavy (50:50 read:write)

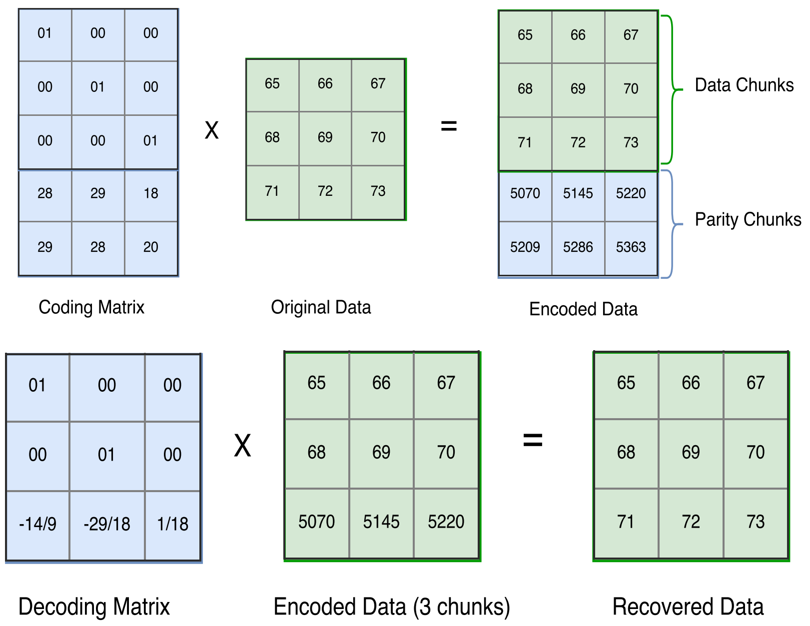


- 4 Memcached servers
- Pre-loaded with 2GB data + 8 KB KV size
- 100 Clients on 32 nodes + Zipf distribution

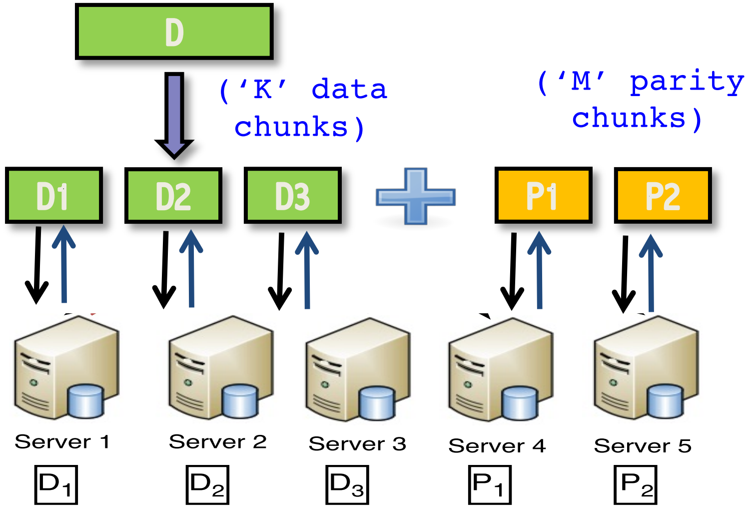
OUTLINE

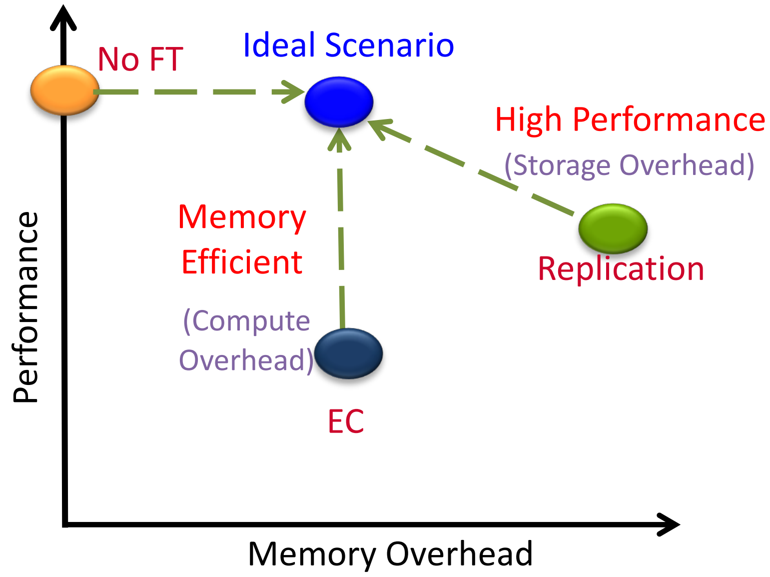
- **Three Approaches**
  - High-Performance RDMA-aware Non-Blocking APIs
  - Fast Online Erasure Coding with RDMA
  - Co-Designing Key-Value Store-based Burst Buffer over PFS
- **Software Release & Impact**
- **Conclusion**

CHALLENGES FOR ONLINE EC IN KV STORAGE



(Reed-Solomon Erasure Coding for K=3 and M=2)



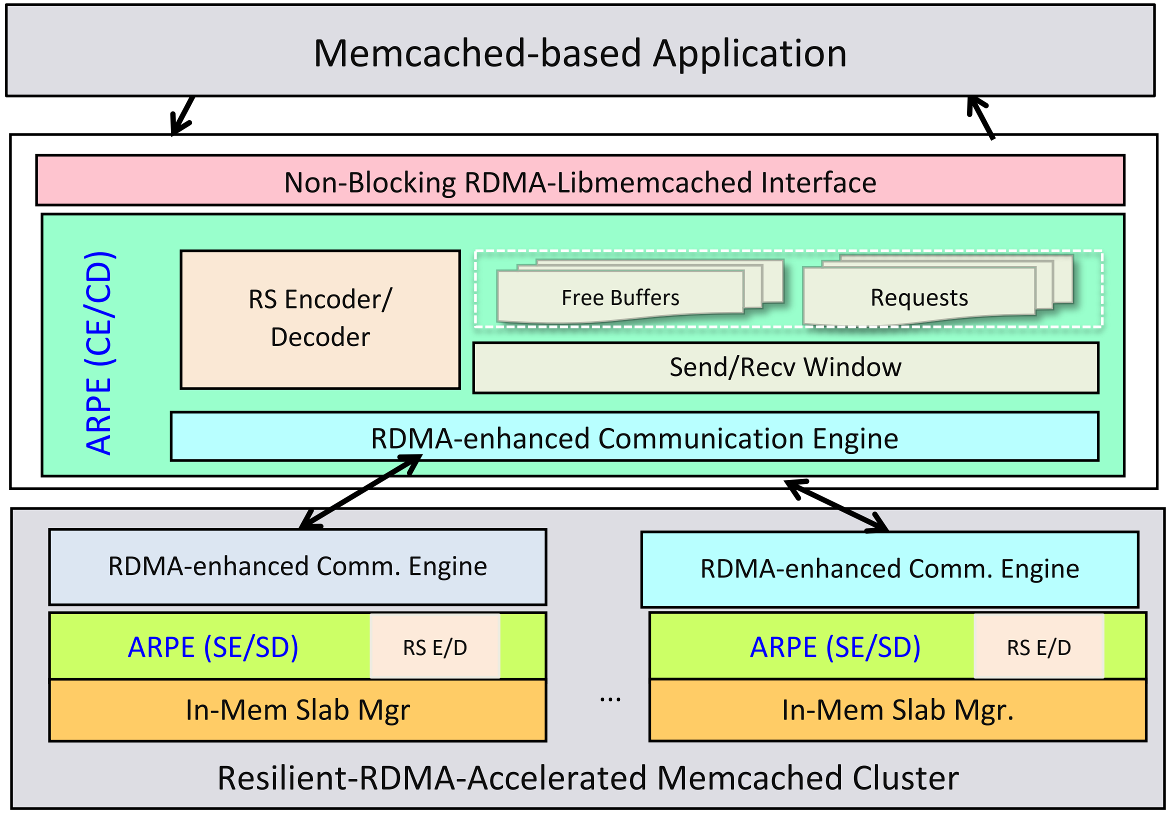


- **Challenges for EC for Resilient KV Storage**
  - Compute Overhead for KV Set/Get request
  - Added communication overheads to fetch/store data from K or N servers

How can we make Online EC viable for KV Storage?

PROPOSED KV STORE WITH ONLINE EC

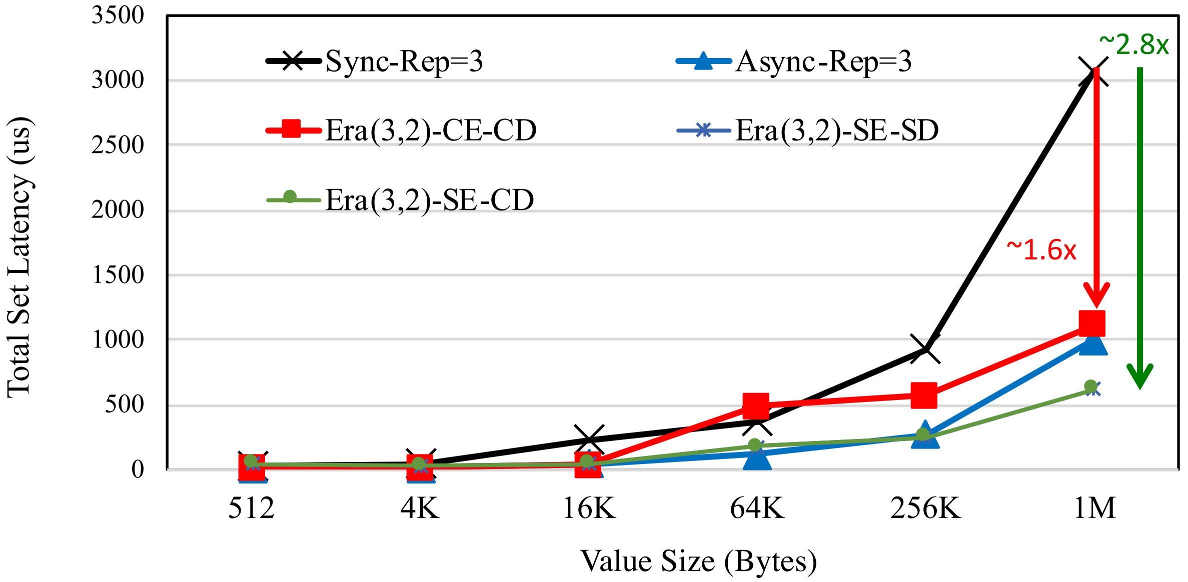
- **Based on RDMA-enhanced Memcached**
  - Non-Blocking RDMA-Libmemcached Set/Get APIs
    - memcached_iset/iget to queue request
    - memcached_wait/test to poll completion
- **Asynchronous RDMA-enabled Request Processing Engine (ARPE)**
  - RS Encoder/Decoder Module designed with the Jerasure Library
  - Pre-registered re-usable buffers + Send/Recv window
  - Client ARPE performs Client-Encode and/or Client-Decode
  - Server ARPE performs Server-Encode and/or Server-Decode



ARPE = Async. Request Processing Engine
CE/CD = Client Encode / Client Decode
SE/SD = Server Encode / Server Decode

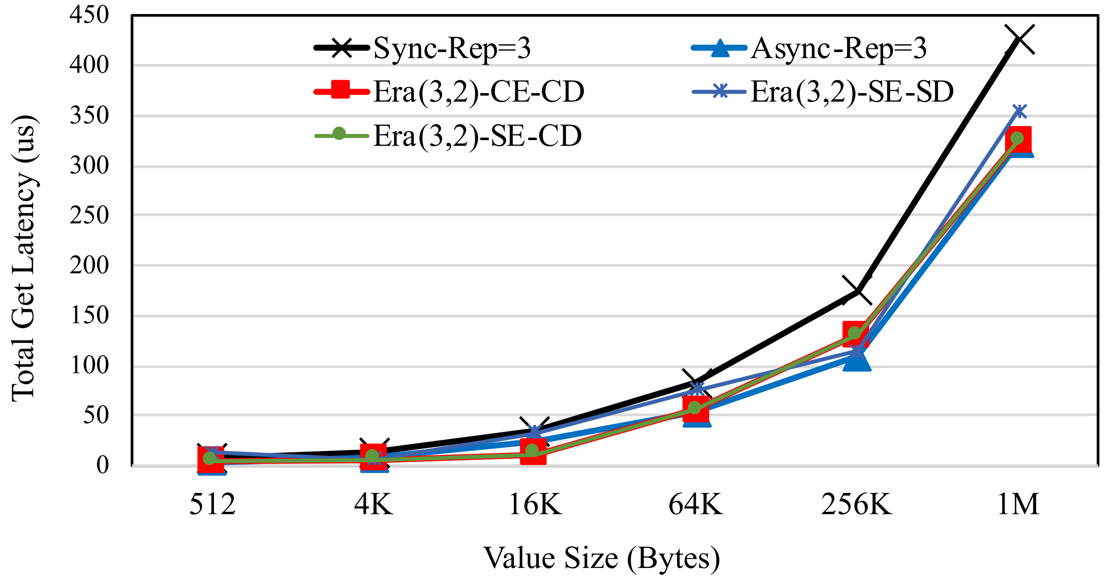
D. Shankar , X. Lu , and D. K. Panda, "High-Performance and Resilient Key-Value Store with Online Erasure Coding for Big Data Workloads", IEEE ICDCS 2017

NON-BLOCKING RDMA-MEMCACHED MICRO-BENCHMARKS



Iset Latency: RS(3,2) vs. Replication=3

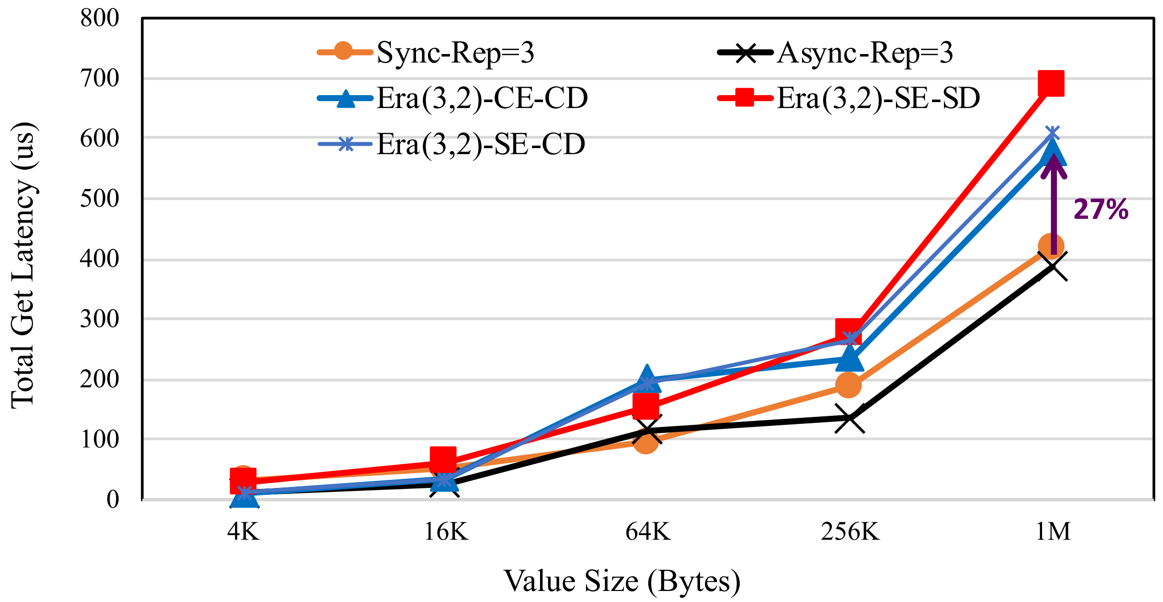
- 5 Memcached servers on Intel Westmere Cluster, 1 Client, 1K ops.
- 1.6x - 2.8x gain for Era-CE-CD vs. Sync-Rep
- Era-CE-CD performs similar to Async-Rep
- Best case latency: Era-SE-CD/SE-SD gains 38% over Era-CE-CD (overlap T_{comm} and T_{encode})

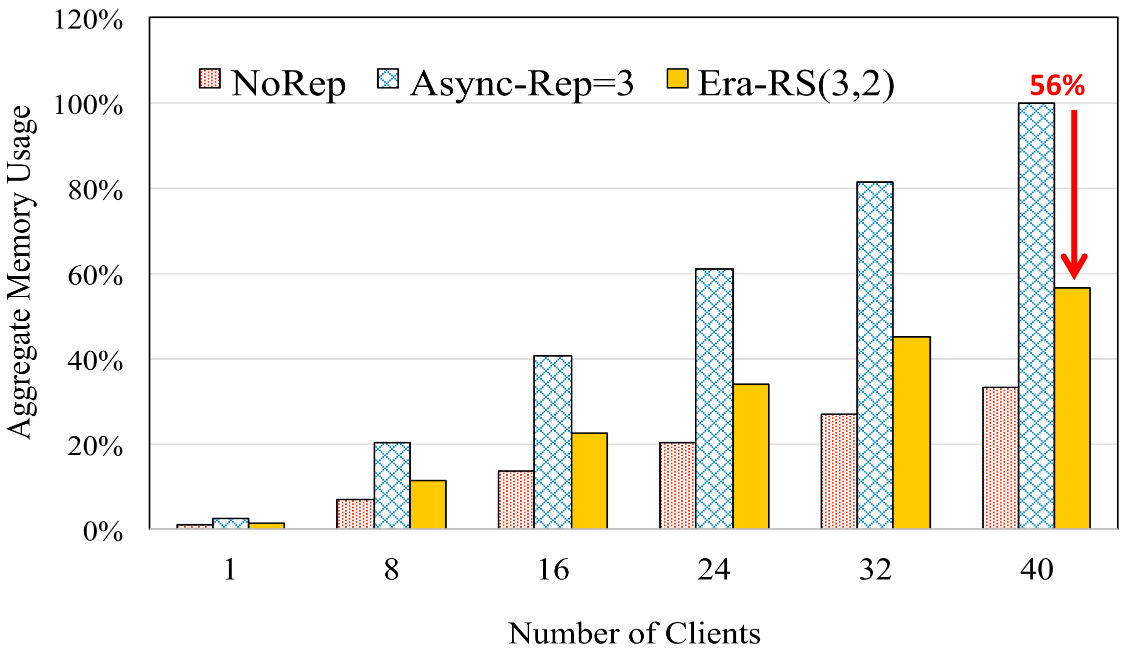


Iget Latency: RS(3,2) vs. Replication=3

- 5 Memcached servers on Intel Westmere Cluster, 1 Client, 1K ops.
- Online Erasure Coding (CE-CD/SE-SD/SE-CD)
  - Performs on-par with Async-Rep
  - Leverages non-blocking RDMA-based get requests

FAULT-TOLERANCE AND MEMORY EFFICIENCY





Recovery during two-node failures (max. failures):

- Get Micro-benchmark on 5 Memcached servers on Intel Westmere Cluster, 1 Client, 1K ops.
- Noticeable degradation vs. Replication
  - Era-SE-CD/Era-CE-CD = 27%
  - Era-SE-SD = 2.2x

Memory Efficiency:

- OHB Set Micro-benchmark on 5 Memcached servers on Intel Westmere Cluster, 1 – 40 Clients, 1 GB/Client
- % usage of total aggregated memory: Online-EC design = 56% and Replication = 100%

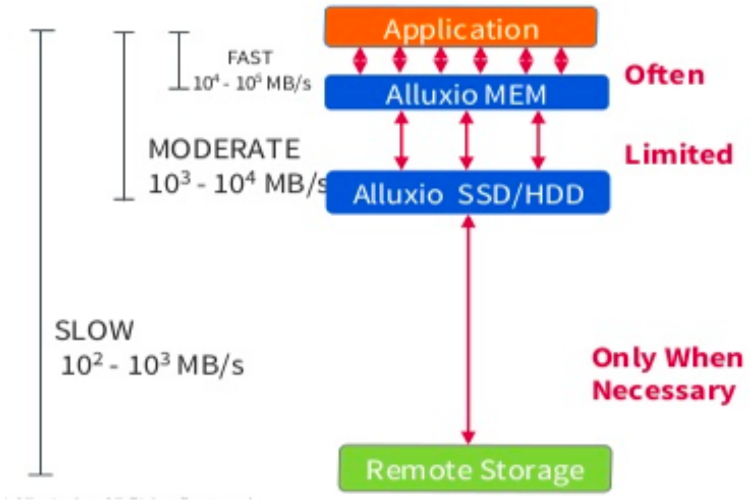
OUTLINE

- **Three Approaches**
  - High-Performance RDMA-aware Non-Blocking APIs
  - Fast Online Erasure Coding with RDMA
  - Co-Designing Key-Value Store-based Burst Buffer over PFS
- **Software Release & Impact**
- **Conclusion**

MOTIVATION OF KVS-BASED BURST BUFFER SYSTEMS

Traditional Big Data I/O

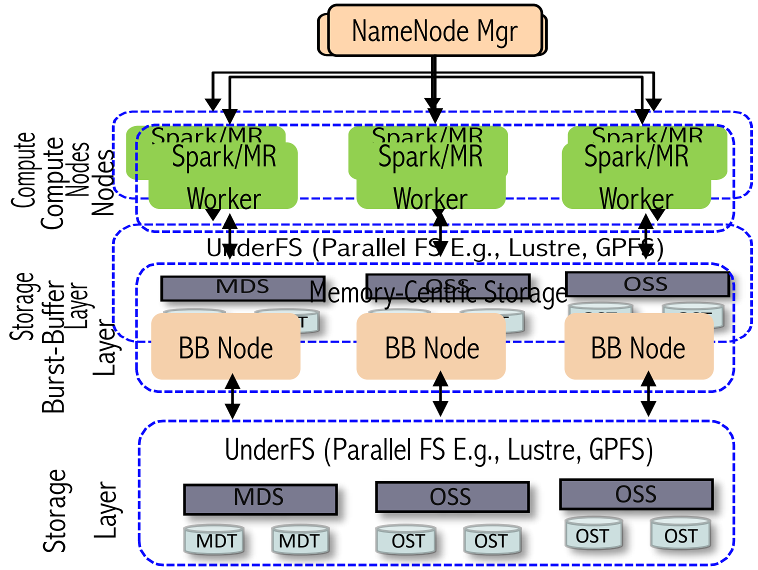
- Aggregate local storage across nodes
- Data Locality
- E.g. of Memory Centric Storage (Alluxio / Tachyon)



*Source: Alluxio, Inc.

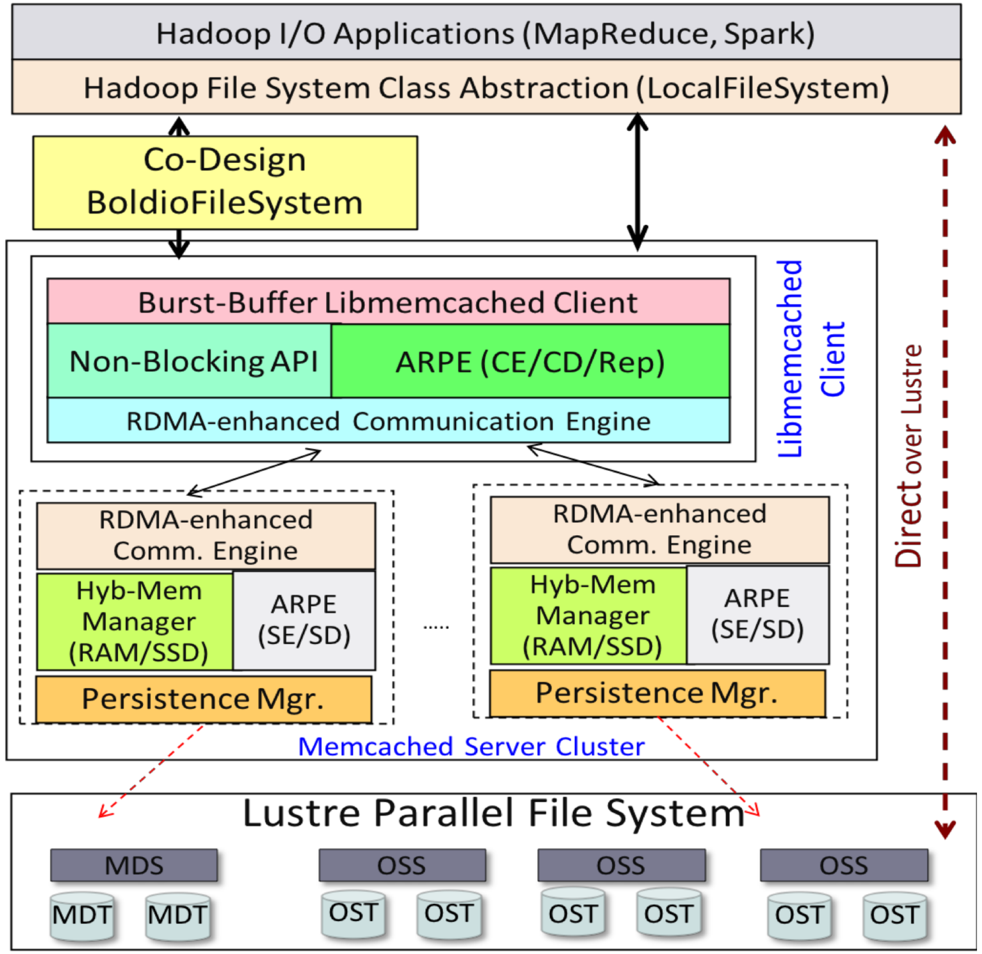
Big Data I/O on HPC

- Heavy reliance on Parallel FS (no-locality)
- Burst-Buffer (BB) systems to application I/O throughput needs



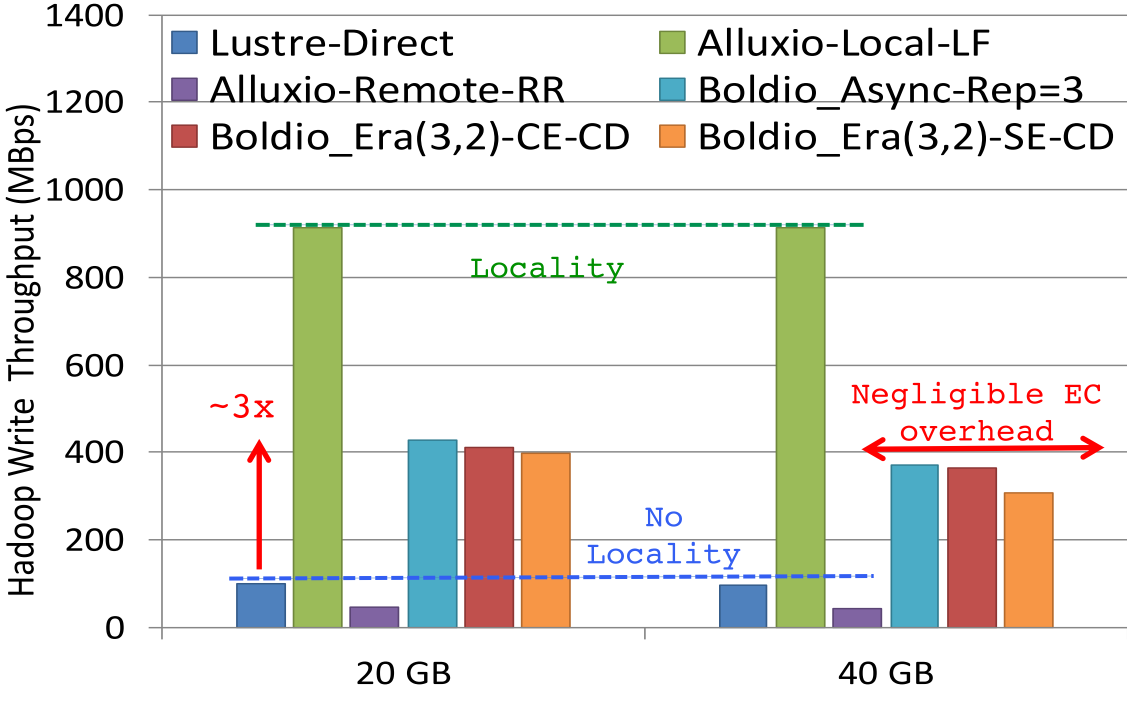
CO-DESIGNING KVS-BASED BURST BUFFER OVER PFS

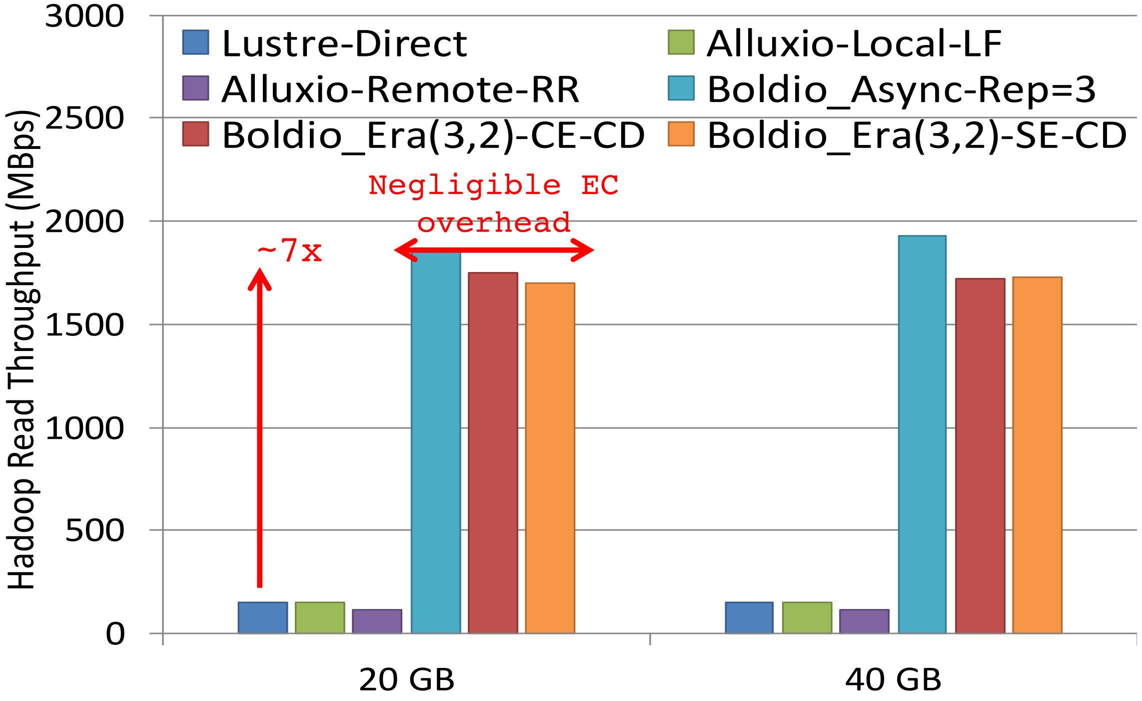
- **Hybrid and Resilient KV Store-based Burst-Buffer system Over Lustre (Boldio)**
- **Overcome local storage limitations on HPC nodes; performance of `data locality'**
- **Accelerating I/O-intensive Big Data workloads**
  - Light-weight transparent interface to Hadoop/ Spark applications
  - Non-blocking RDMA-Libmemcached APIs
  - Resilience via Client-initiated Replication or Online Erasure Coding with RDMA
  - Asynchronous persistence to Lustre from RDMA-Memcached Server Cluster



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

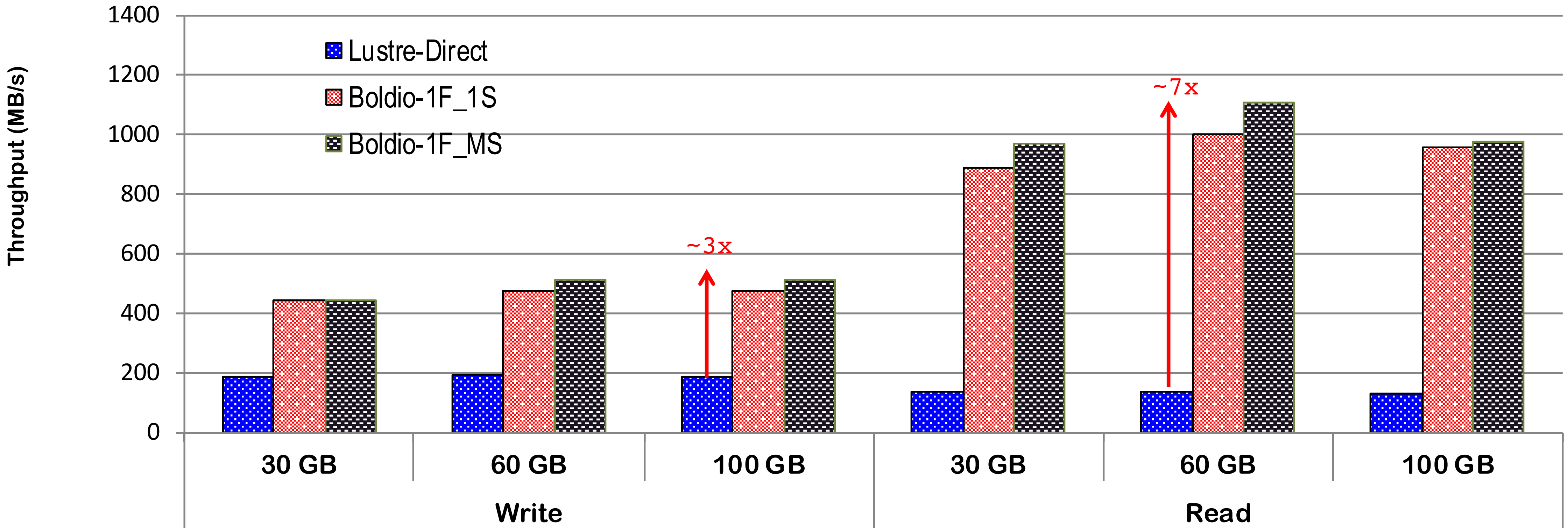
CONTRASTING LOCALITY-AWARE STORAGE AND RESILIENCE





- Intel Westmere Cluster (8-core Intel Sandy Bridge and IB QDR)
- (8-node Hadoop + 5-node Boldio) Cluster vs. 12-node Hadoop Cluster over Lustre
- >10x Performance drop without locality in Alluxio => need for RDMA semantics!!
- Boldio1F_MS/1F_1S improvements: Write = ~3.07x and Read = ~7.3x over Lustre-Direct; Write = ~3.1-5.8x and Read = ~10.8x over Alluxio-Remote

HADOOP I/O THROUGHPUT: SCALABILITY



- TestDFSIO on SDSC Gordon Cluster (16-core Intel Sandy Bridge and IB QDR) with 16-node Hadoop Cluster + 4-node Boldio Cluster
- Boldio can sustain 3x and 7x gains in read and write throughputs over stand-alone Lustre

OUTLINE

- Three Approaches
  - High-Performance RDMA-aware Non-Blocking APIs
  - Fast Online Erasure Coding with RDMA
  - Co-Designing Key-Value Store-based Burst Buffer over PFS
- Software Release & Impact
- Conclusion

SOFTWARE RELEASE & IMPACT

- **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
  - Compliant with libMemcached 1.0.18 APIs and applications
- **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
- **RDMA for Apache Spark (RDMA-Spark), RDMA for Apache Hadoop (RDMA-Hadoop 1.x, 2.x, 3.x), RDMA for Apache Kafka, RDMA for HBase**
- **http://hibd.cse.ohio-state.edu**
- **HiBD Users Base: 300 organizations, 35 countries**
- **As of Mar '19, more than 29,350 downloads have taken place from this project's site**

HiBD
High-Performance Big Data

Network Based Computing Laboratory

The Ohio State University

OUTLINE

- **Three Approaches**
  - High-Performance RDMA-aware Non-Blocking APIs
  - Fast Online Erasure Coding with RDMA
  - Co-Designing Key-Value Store-based Burst Buffer over PFS
- **Software Release & Impact**
- **Conclusion**

CONCLUSION

- **Addresses key issues on building scalable, resilient and distributed key-value stores over RDMA networks**
- **Presents a holistic approach to leverage HPC resources to maximize end-to-performance, scalability, and data availability**
- **Presents RDMA-aware non-blocking designs for KV Storage that vital in today's memory-centric storage systems**
  - Co-design with online and offline data analytical workloads
  - Demonstrates the corresponding benefits on modern HPC clusters
- **Broader outreach through HiBD RDMA-Memcached public releases**

15th ANNUAL WORKSHOP 2019

THANK YOU

Xiaoyi Lu, Dhabaleswar K. (DK) Panda

The Ohio State University

E-mail: {luxi, panda}@cse.ohio-state.edu

http://www.cse.ohio-state.edu/~luxi

http://www.cse.ohio-state.edu/~panda