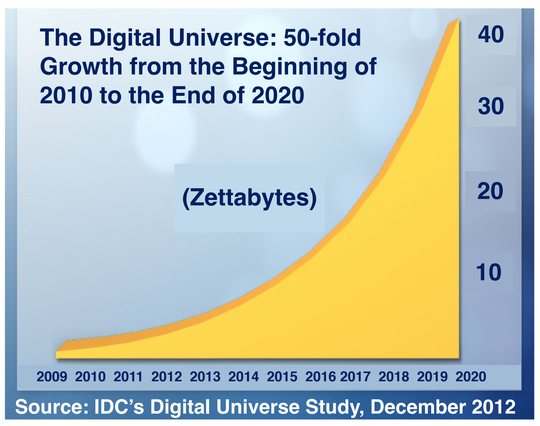
Enabling Efficient Data Infrastructure and Analytics on HPC Systems

Huansong Fu, Advisor: Weikuan Yu
Department of Computer Science, Florida State University
Email: fu@cs.fsu.edu

When Big Data Meets HPC

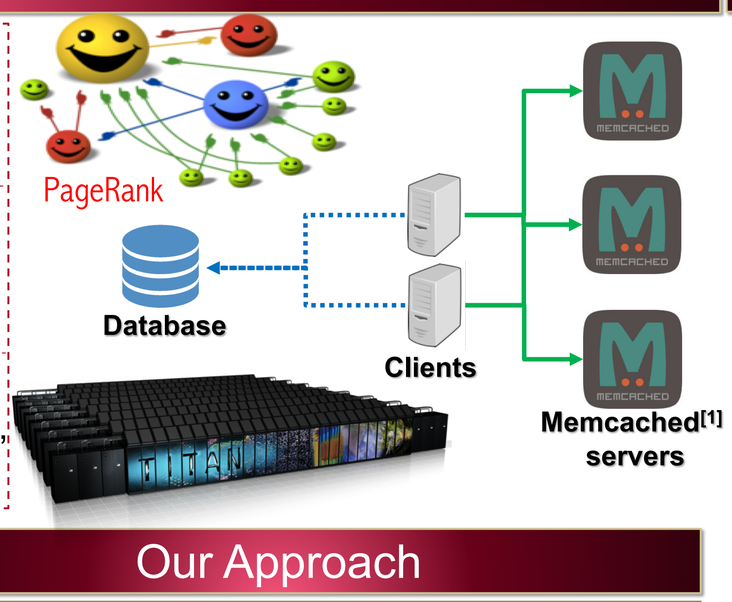
The "Big Data" Era



❖ **Data Analytics**
(What to do with the data?)
e.g. Graph Analytics.

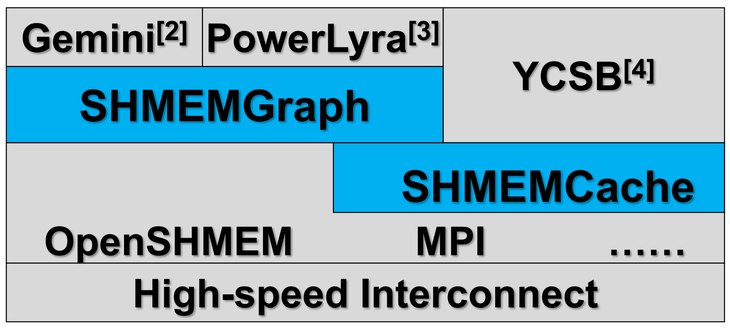
❖ **Data Infrastructure**
(Where to put the data?)
e.g. Key-Value (KV) Store.

❖ **HPC capabilities**
e.g. supercomputers, PGAS, high-speed interconnect



Design Overview

❖ **SHMEMCache**: a distributed KV store with efficient server-bypass access using one-sided communication.
❖ **SHMEMGraph**: a distributed graph processing system with a focus on progress balancing, also using one-sided communication.

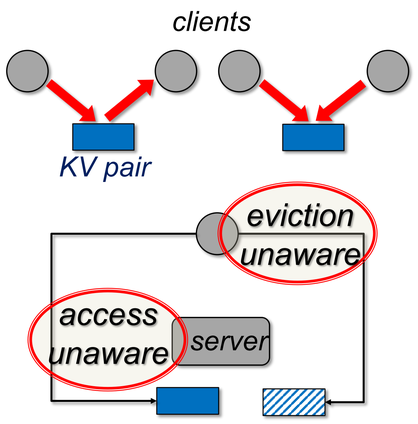


Challenges

❖ **SHMEMCache:**

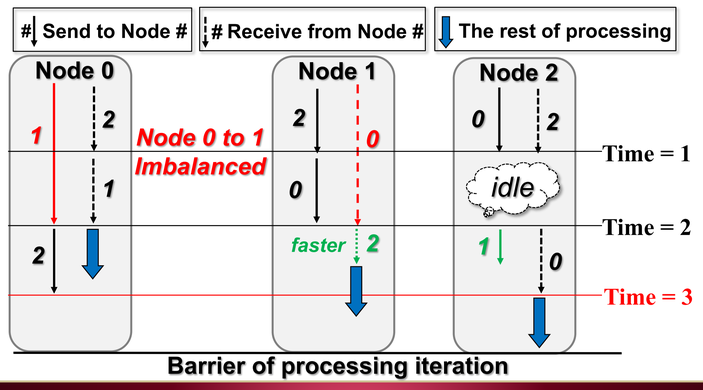
➢ Read-write & write-write races

➢ Unawareness of access & eviction



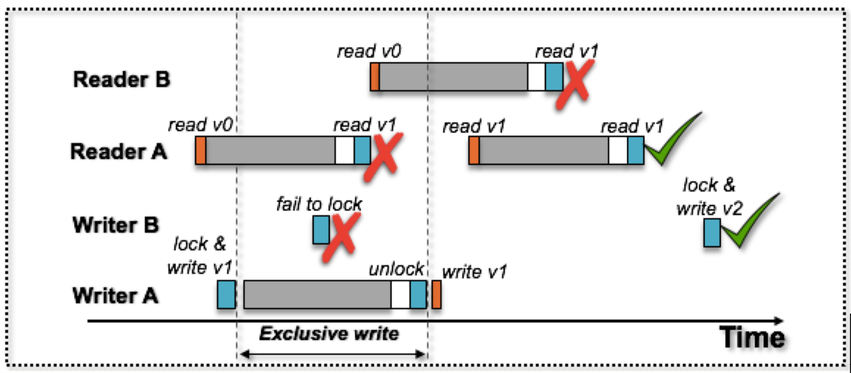
❖ **SHMEMGraph:**

➢ Imbalanced progresses in synchronous model.

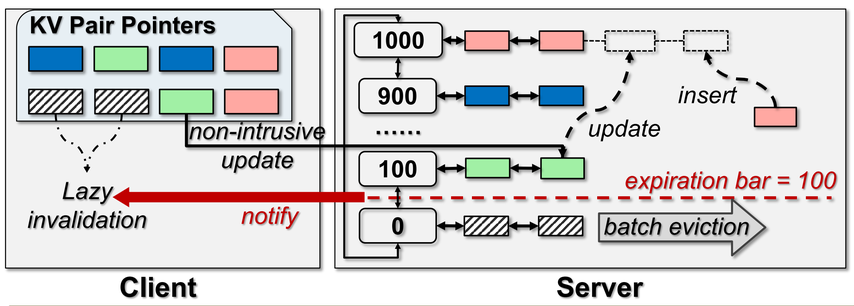


Our Approach

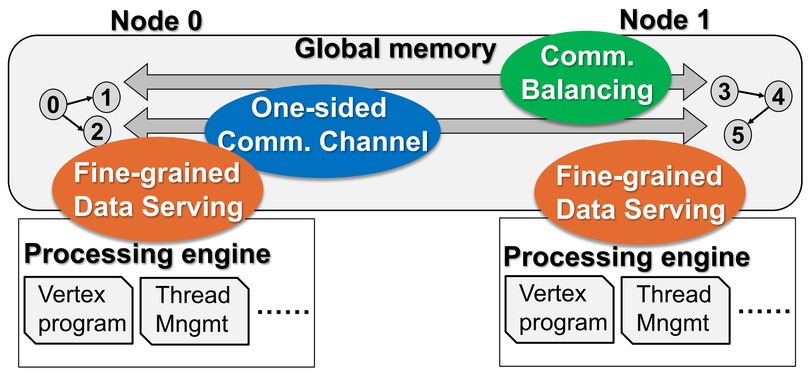
❖ **Key observation 1:** write can invalidate both read and write, so write needs to be done *exclusively*.
➢ Reads are more common, so lock is only for write.
➢ A lightweight versioning concurrency control for read.



❖ **Key observation 2:** we can relaxed *recency* definition from a *time point* (t1) to *time range [t1, t2)*.
➢ Non-intrusive recency update.
➢ Batch eviction of KV pairs with the same recency.
➢ Client lazily invalidate pointers with outdated recency.



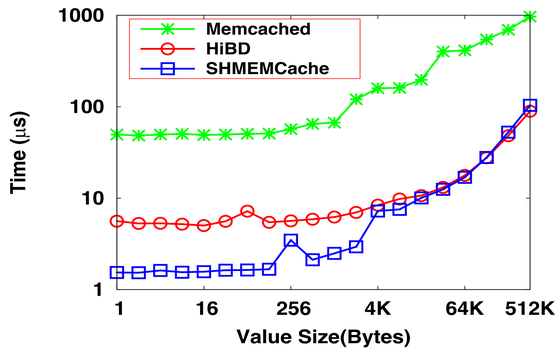
❖ **Key observation 3:** we can exploit the efficiency and flexibility of one-sided communication.
➢ *One-sided communication channel* facilitates direct access of remote vertices in PGAS.
➢ *Fine-grained data serving* reduces long delay caused by imbalance and improves overlapping.
➢ *Communication balancing* dynamically re-balances work between nodes.



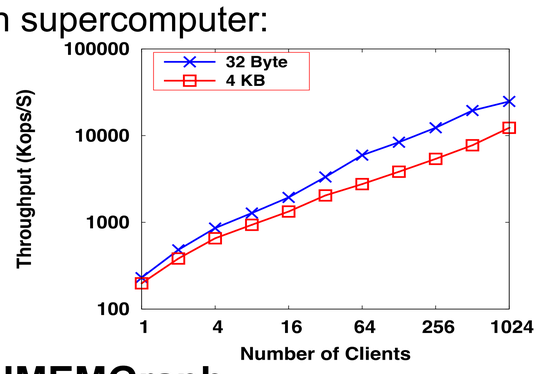
Results

❖ **SHMEMCache**

➢ Improvement to Memcached[1] and HiBD [5] : 128% and 2,534% lower latency for Set (shown below) and 16% and 2,045% for Get.



➢ Scaling SHMEMCache well to 1024 nodes on Titan supercomputer:

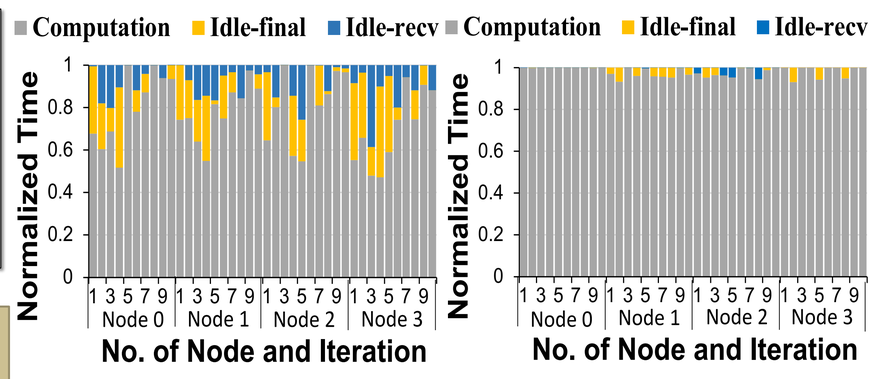


❖ **SHMEMGraph**

➢ Compared to the state-of-the-art Gemini system [2]:

	PR	CC	SSSP	BC	BFS
Gemini	0.43	0.29	0.44	2.25	1.19
SHMEMGraph	0.24	0.21	0.34	1.91	1.05
Improvement	*44.2%*	*26.3%*	*23.1%*	*15.1%*	*11.8%*

➢ Elimination of node idle time (before & after):



Selected Publications

[CCGrid'18] **Huansong Fu**, Manjunath Gorentla Venkata, Shaeke Salman, Neena Imam, and Weikuan Yu. SHMEMGraph: Efficient and Balanced Graph Processing Using One-sided Communication.
[CCGrid'17] **Huansong Fu**, Manjunath Gorentla Venkata, Ahana Choudhury, Neena Imam and Weikuan Yu. High-Performance Key-Value Store On OpenSHMEM.

Reference

[1] Memcached. https://memcached.org/
[2] Zhu, Xiaowei, et al. "Gemini: A Computation-Centric Distributed Graph Processing System." *OSDI'16*.
[3] Chen, Rong, et al. "Powerlyra: Differentiated graph computation and partitioning on skewed graphs." *EuroSys'15*.
[4] Cooper, Brian F., et al. "Benchmarking cloud serving systems with YCSB." *SoCC'10*.
[5] HiBD. http://hibd.cse.ohio-state.edu/.

Acknowledgement

This research was supported in part by a contract from Oak Ridge National Laboratory and the National Science Foundation awards 1561041, 1564647, and 1744336. This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.