A Paradigm Shift of the Computing Model for IO-Intensive Big Data Applications

Dongchul Park, Ph.D.
Computer & Electronic Systems Engineering
Hankuk University of Foreign Studies (HUFS)
May 10, 2019

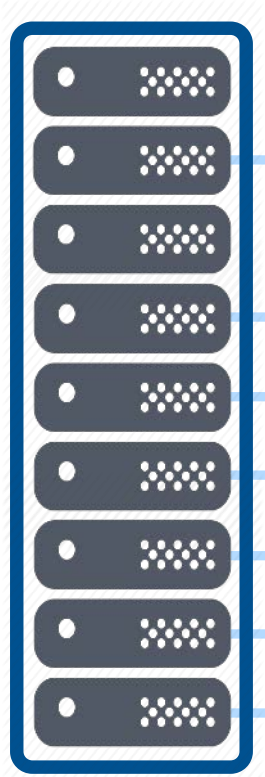
Outline

- ISC for Big Data Processing
- Case Study and Demo
- Performance Analysis
- Summaries and Takeaways

D. Park, et al. “In-Storage Computing for Hadoop MapReduce Framework: Challenges and Possibilities.” *IEEE Transactions on Computers (TC)*

In-Storage Computing for IO-Intensive Big Data Applications

Data Deluge Era



Close to **2.5 BILLION** people are online around the world

A number has grown **566%**

and **70%** of them use the Internet every day

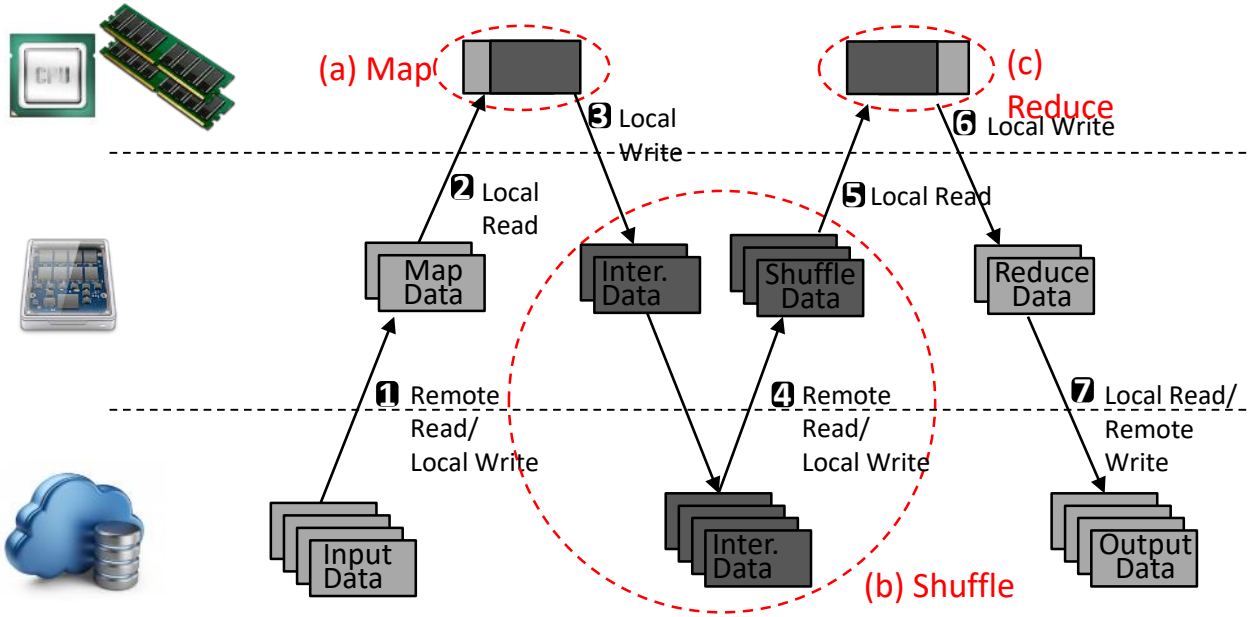
EVERY 60 SECONDS:

- **204 Million** EMAIL MESSAGES ARE EXCHANGED
- **5 Million** SEARCHES ARE MADE ON GOOGLE
- **1.8 Million** 'LIKES' ARE GENERATED ON FACEBOOK
- **350,000** TWEETS ARE SENT ON TWEETER
- **$272,000** OF MERCHANDISE IS SOLD ON AMAZON
- **15,000** TRACKS ARE DOWNLOADED VIA ITUNE

Source: NRDC, Data Center Efficiency Assessment

Heavy Data I/O

- Big data processing framework (e.g., Hadoop MapReduce)



Issue: I/O Performance

- BI and analytics tools: I/O bottleneck

Event	Waits	Time(s)	Avg. wait (ms)	% DB time	Wait class
Direct path read	4,604,339	567.141	123	63.67	User I/O
Direct path read temp	1,955,162	147,298	75	16.54	User I/O
DB CPU		38,874		4.36	
DB file sequential read	117,944	16,399	139	1.84	User I/O
Direct path write temp	597,138	13,507	23	1.52	User I/O

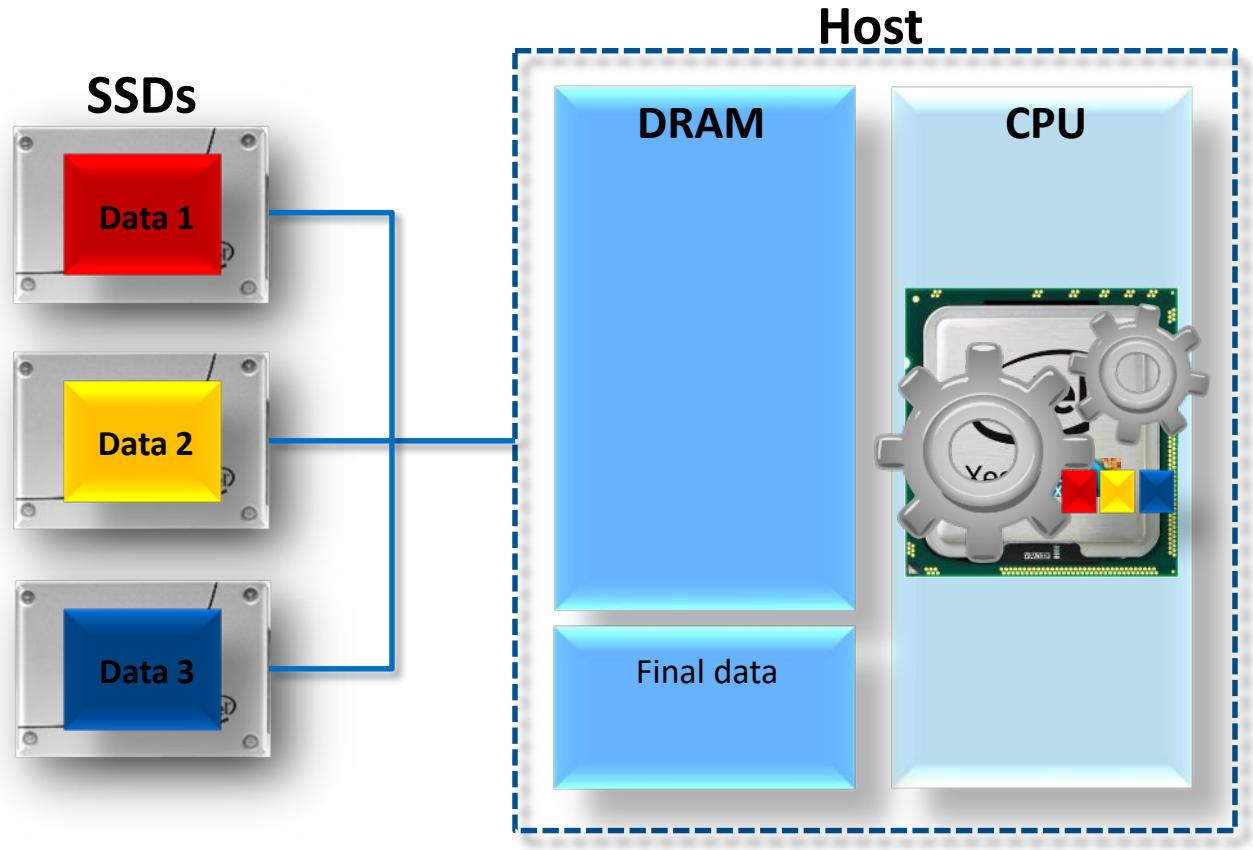
Source: HUAWEI, Accelerate Oracle Performance

Fastest I/O?

Fastest I/O is the I/O that never happened !
(Still, not there yet)

"In-Storage Computing (ISC)"

Legacy CPU-centric Computing



Moving Data is Expensive!

"Moving computation is cheaper than moving data."

Source: HDFS Architecture Guide

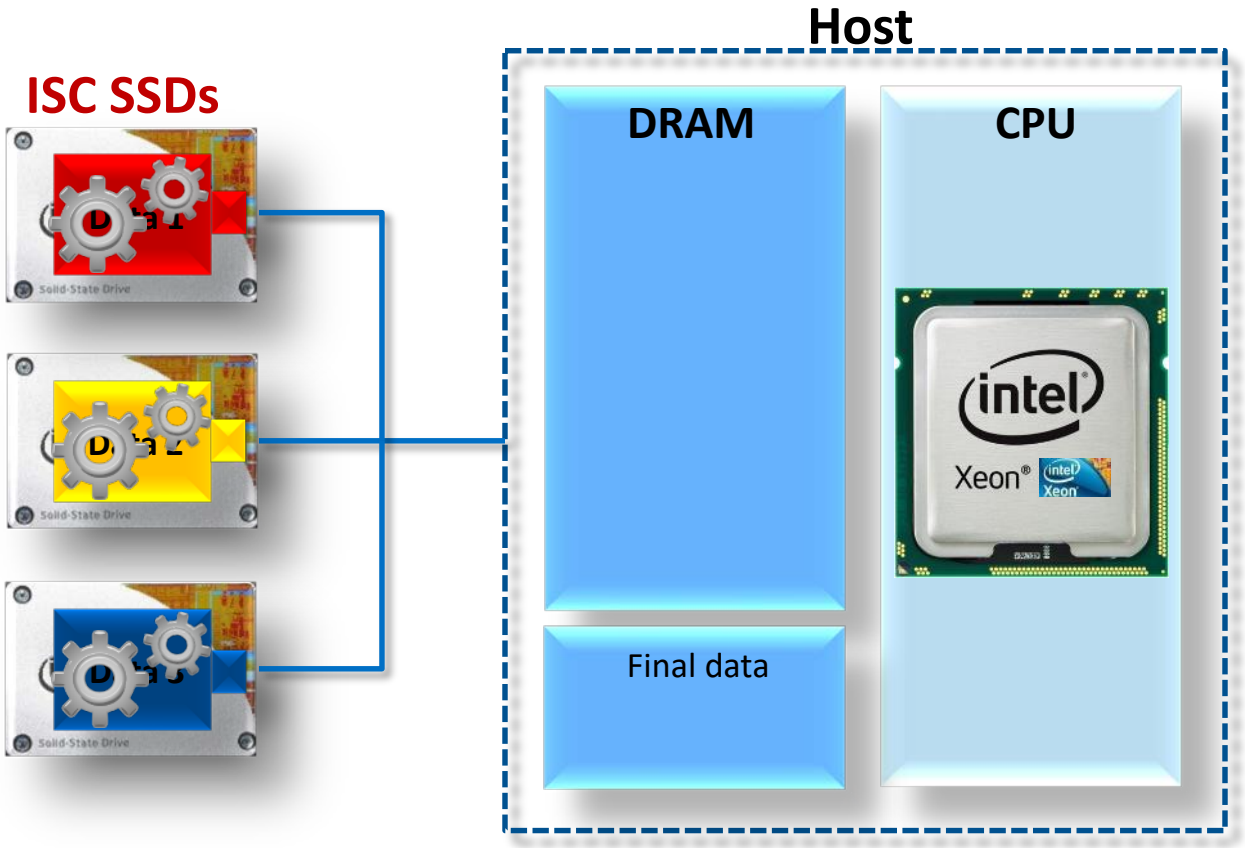
"Reducing data movement can help improve both energy and performance."

Source: USENIX HotPower, 2012

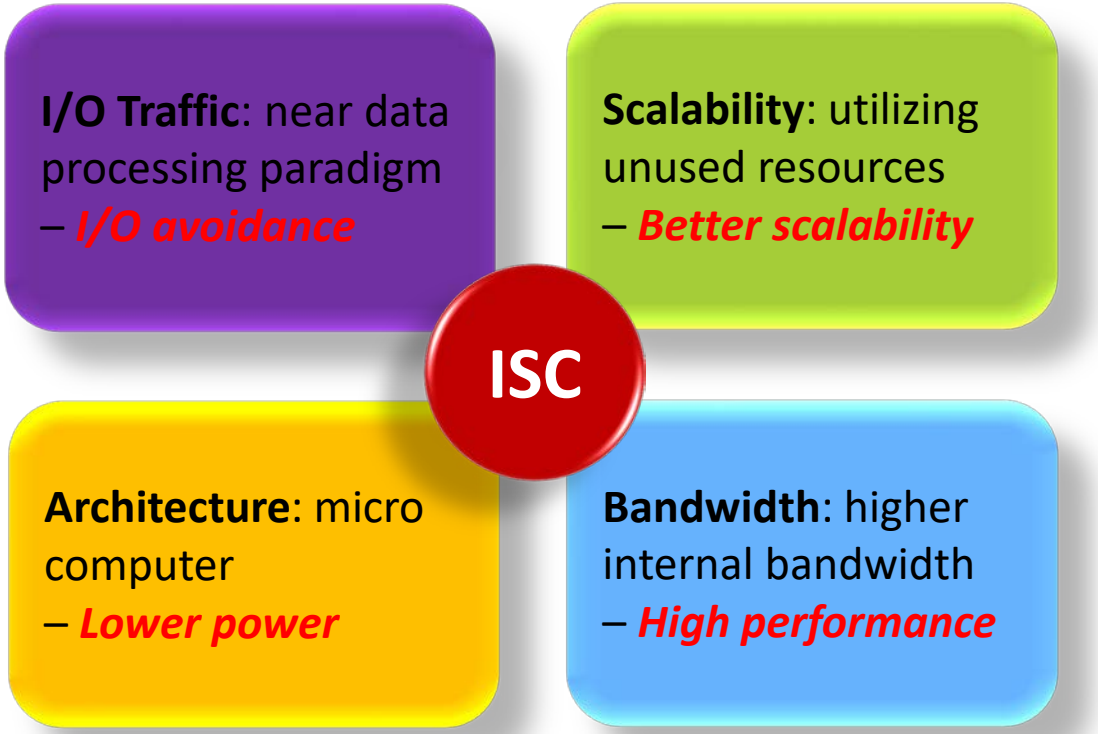
"The energy consumed by data movement is starting to exceed the energy consumed by computation."

Source: High Performance Parallel IO, 2014

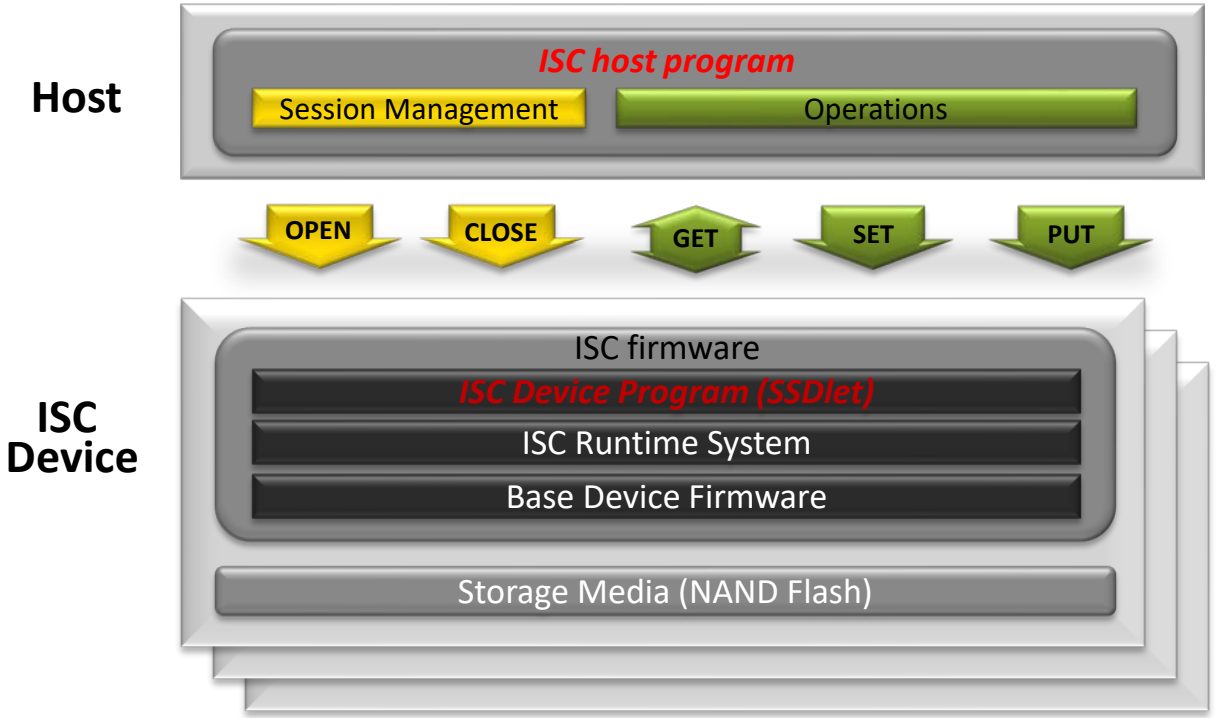
In-Storage Computing (ISC)



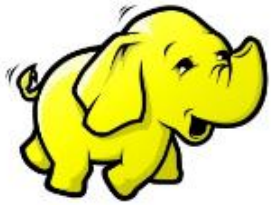
Why ISC (In-Storage Computing)?



ISC software architecture (ISC APIs)

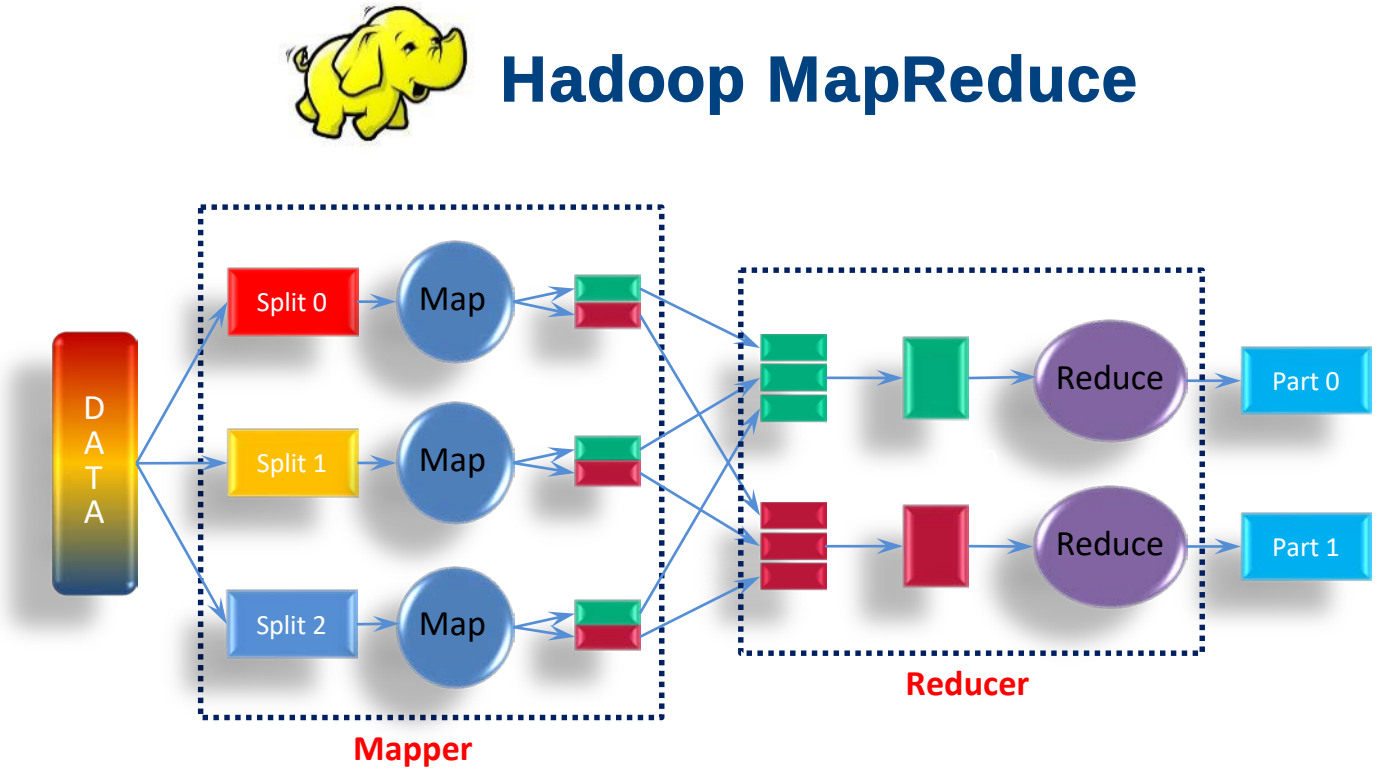


Case Study and Demo

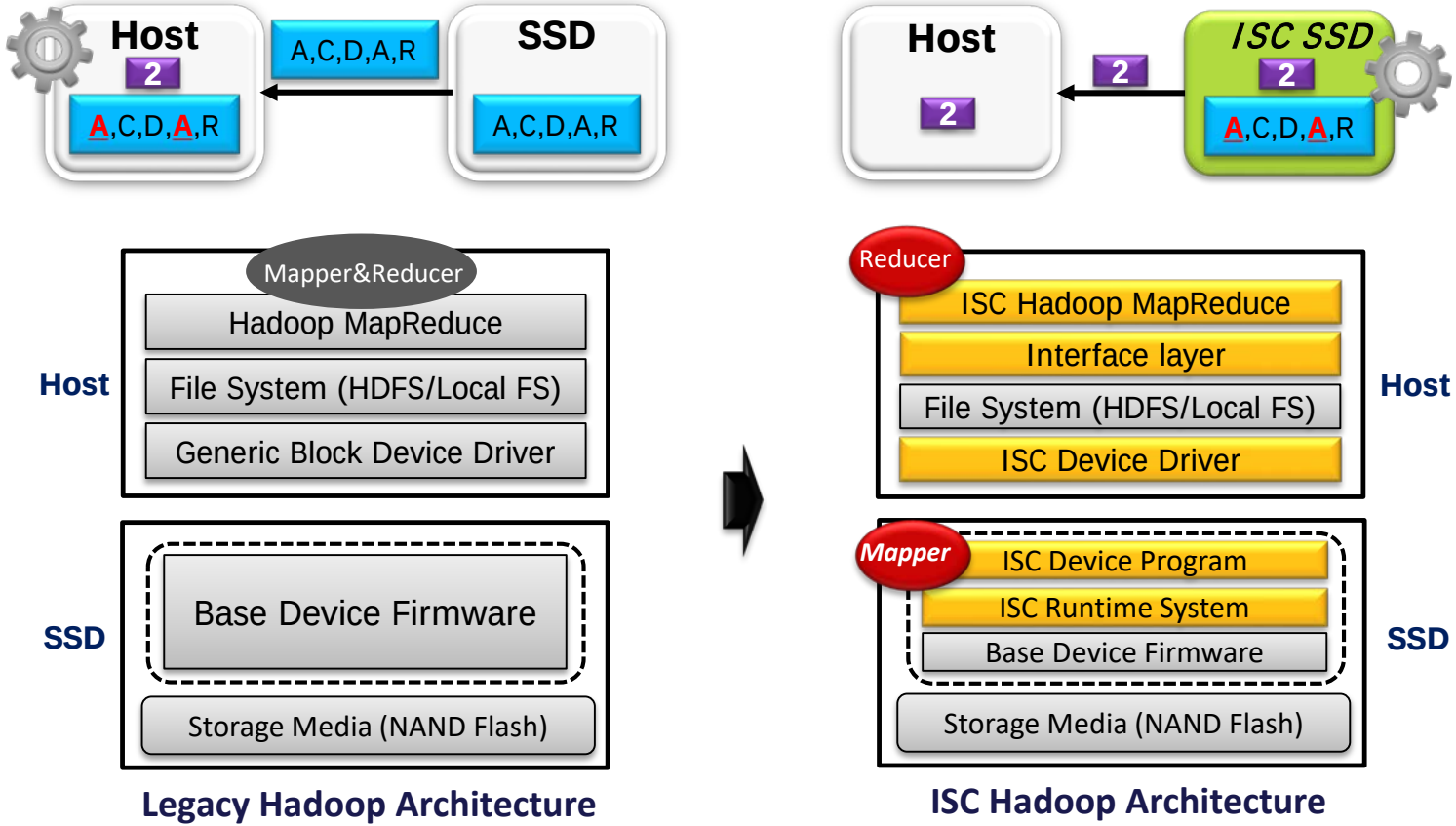


Big Data Processing
(Hadoop MapReduce)

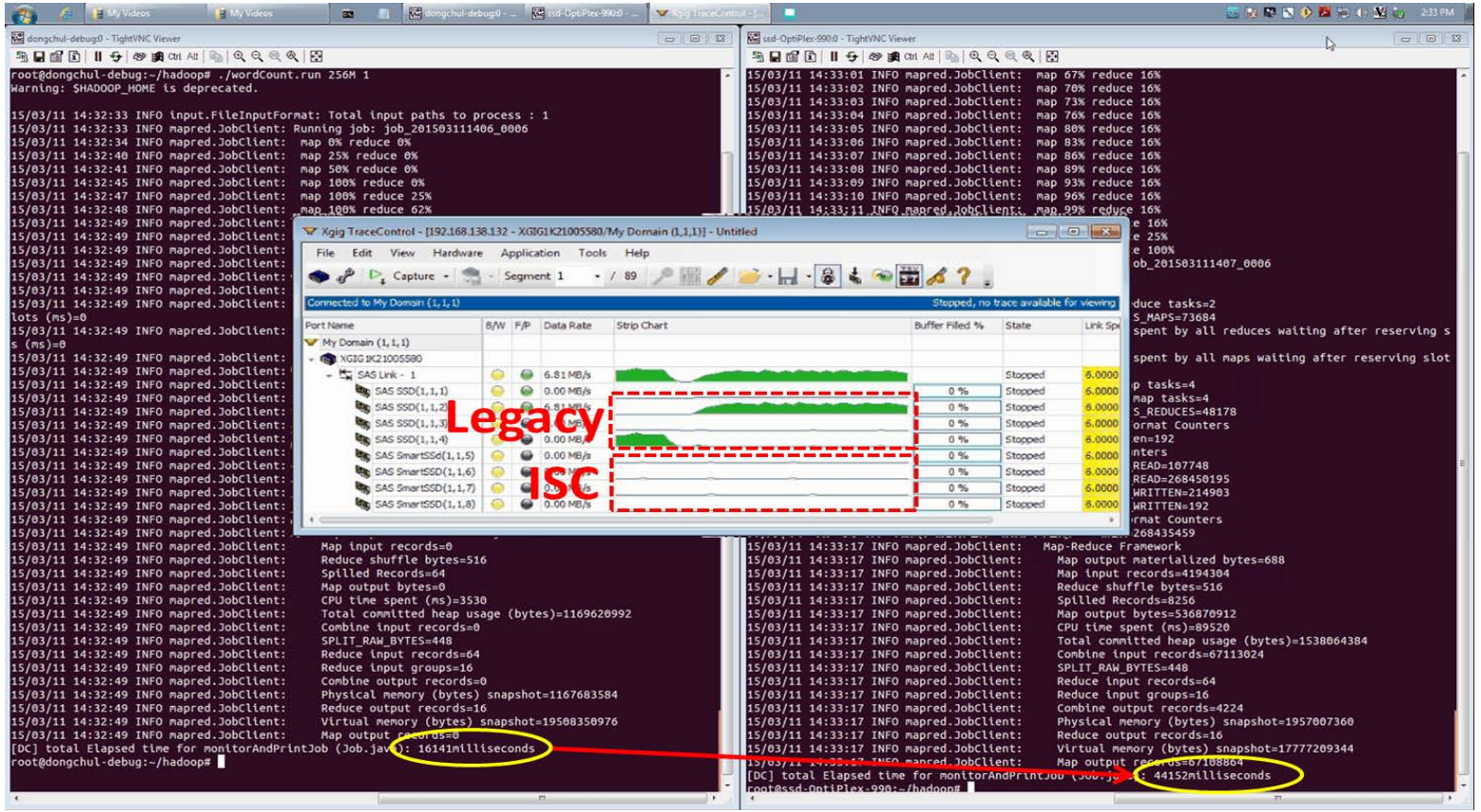
Big Data Processing Framework



ISC Hadoop MapReduce Framework

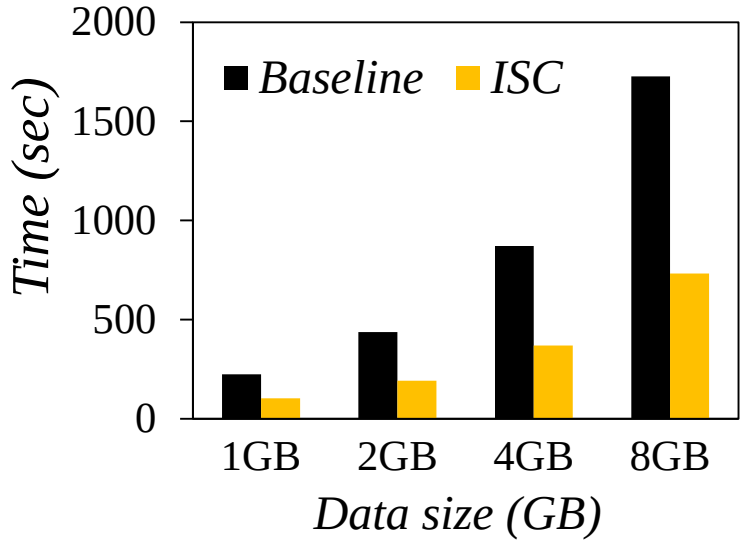


ISC Hadoop Demo Screenshot



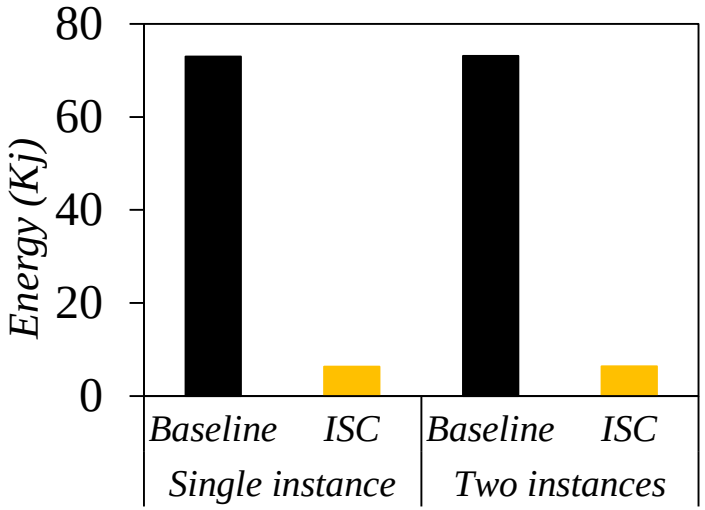
Performance Analysis

Overall Performance



(a) Total elapsed time

2.3X faster !

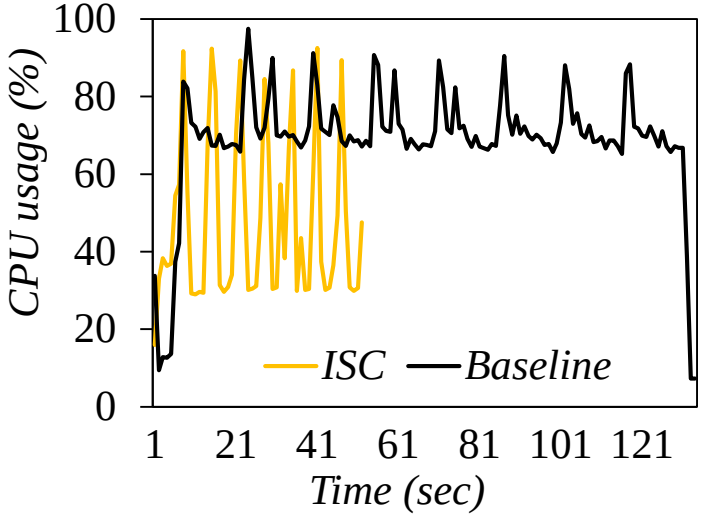


(b) Total energy consumption

11.5X lower energy consumption!

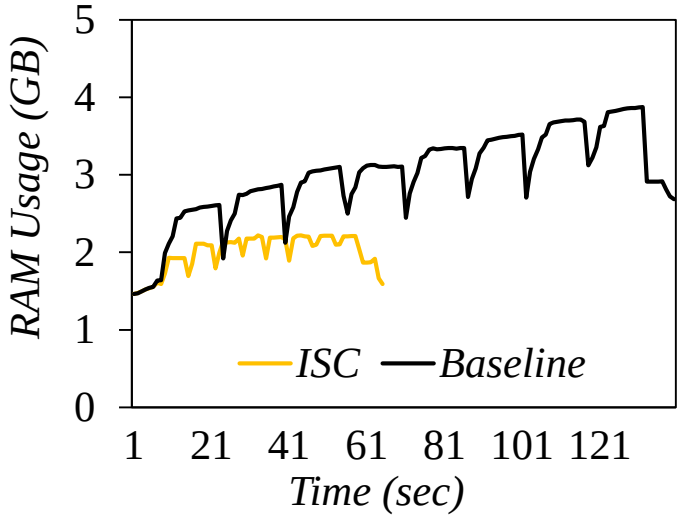
Host Resource Usage (CPU & Memory)

3.7X less



(a) CPU usage over time

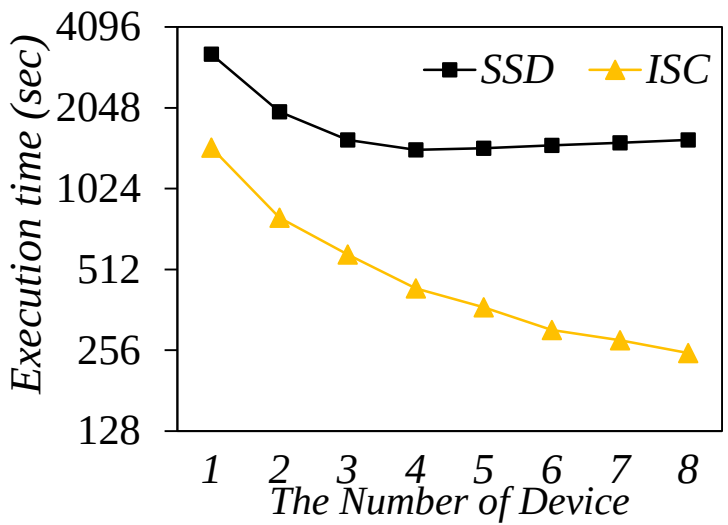
3X less



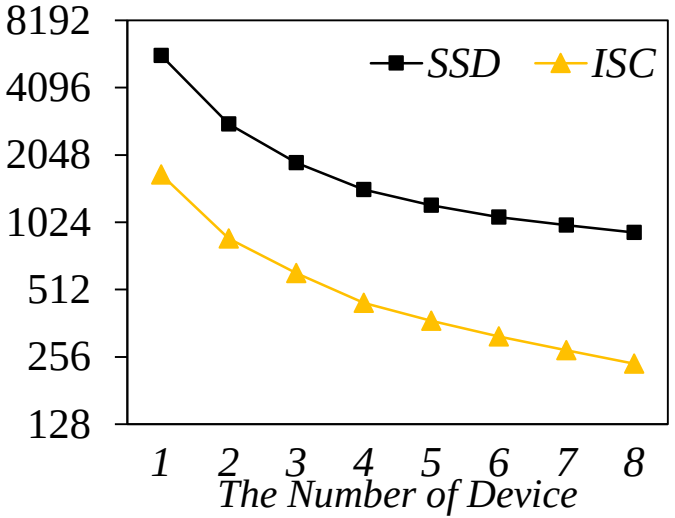
(b) DRAM usage over time

➔ ***ISC Hadoop system consumes even less host resources***

Scalability (PC vs. Server)



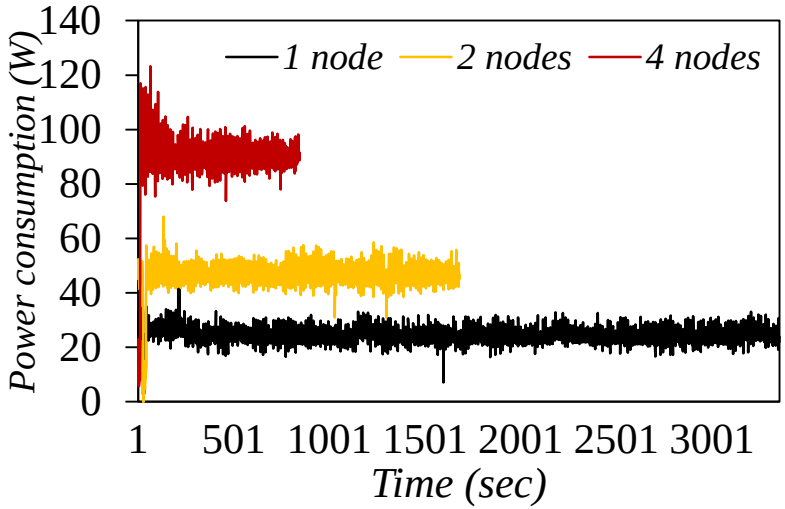
(a) Desktop PC (4 cores)



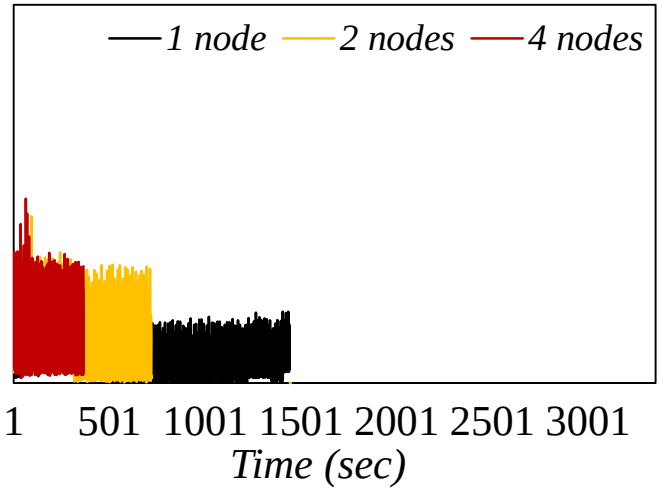
(b) Server (12 cores)

➔ ***ISC Hadoop system scales well even on low-cost desktop.***

Power Consumption (Hadoop Cluster)



(a) Typical Hadoop Cluster



(b) ISC Hadoop Cluster

➔ ***ISC Hadoop Cluster consumes 5X lower power***

Summaries and Takeaways

Challenging Problems for ISC

Discrepancy in data representation

- Host: File systems
- Device: No file systems

Discrepancy in system interfaces

- Different programming interfaces between host and device

Data split

- Logical data can be split in the physical data boundary

Feature offloading

- Which feature should be offloaded?

Takeaways

Computing paradigm shift

- From CPU-centric to data-centric for I/O intensive applications

Big data applications can benefit from ISC

- Utilizing low-power high-performance of embedded processors and SSD's high internal bandwidth

Scalability for data center clusters

- Scale-in: same performance with fewer nodes

ISC prototype performance

- ISC Hadoop: 2.3X faster & 5X less power
- ISC LevelDB: 10X higher throughput

Q&A

Dongchul Park (dpark@hufs.ac.kr)