

HotStorage '14  
June 18, 2014

# Accelerating External Sorting via On-the-fly Data Merge in Active SSDs

*Young-Sik Lee*, Seungryoul Maeng

KAIST

Luis Cavazos Quero, Youngjae Lee, Jin-Soo Kim

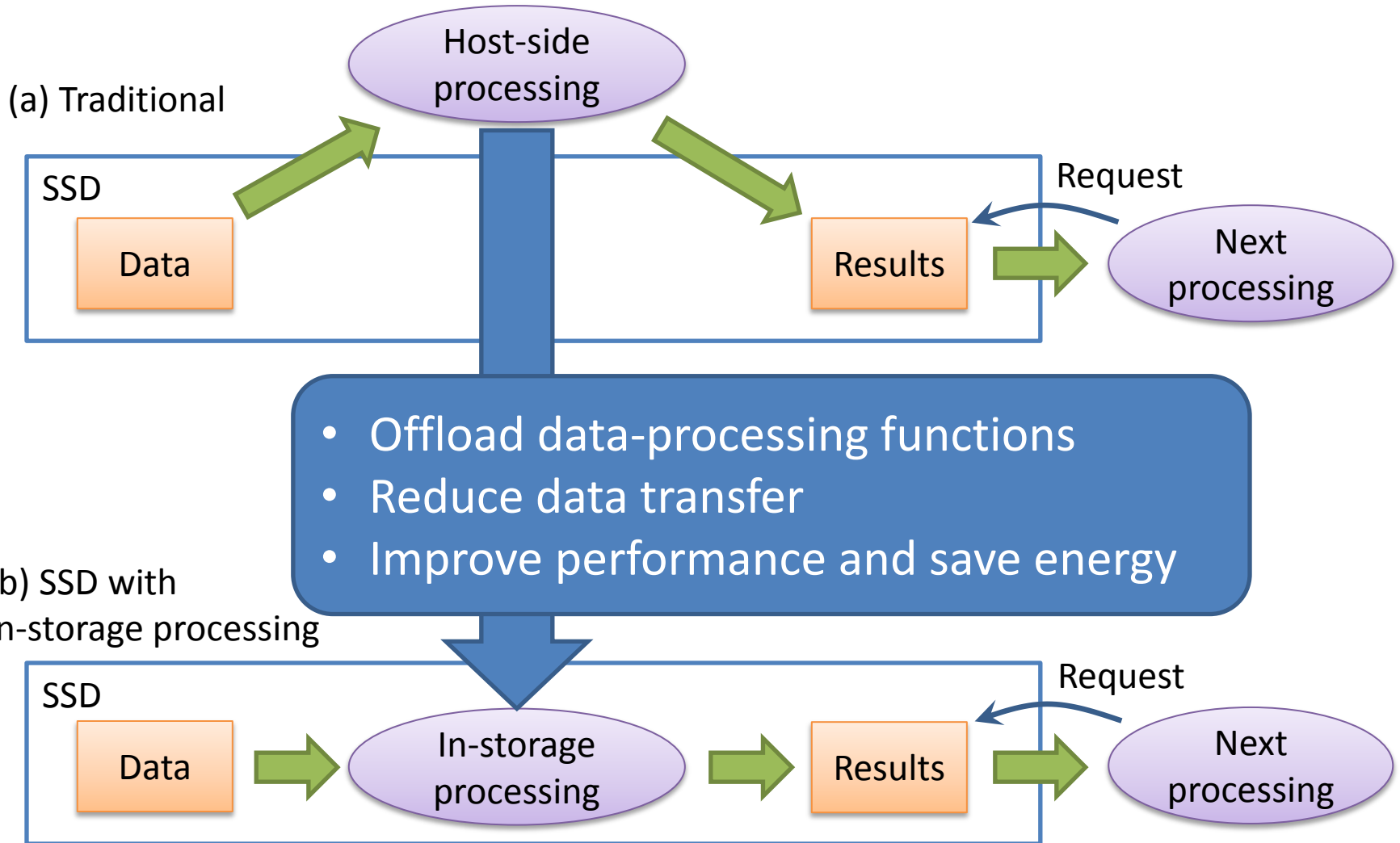
Sungkyunkwan University



# Introduction

- Data-intensive computing
  - Processing large volumes of data → vast I/O
    - Web, SNS, e-commerce, scientific analysis, ...
  - Google MapReduce, Apache Hadoop
- Solid-state drives (SSDs)
  - ***Fast I/O, low power consumption***, small form factor, light weight, and shock resistance

# In-Storage Processing

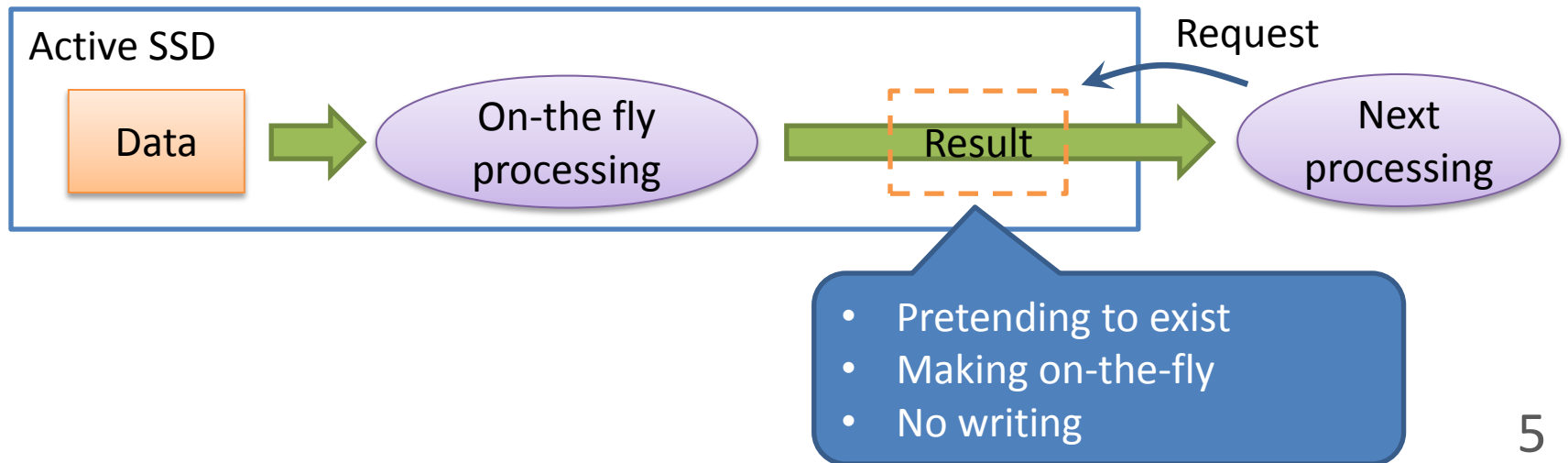


# Related Work

- Fast scan (ADMS'11)
- Intelligent SSD (CIKM'13, ICS'13)
- Active flash (FAST'13)
- Smart SSD (MSST'13)
  
- Large data → small result
  - Aggregate functions: SCAN, SUM, MIN, MAX, COUNT, ...

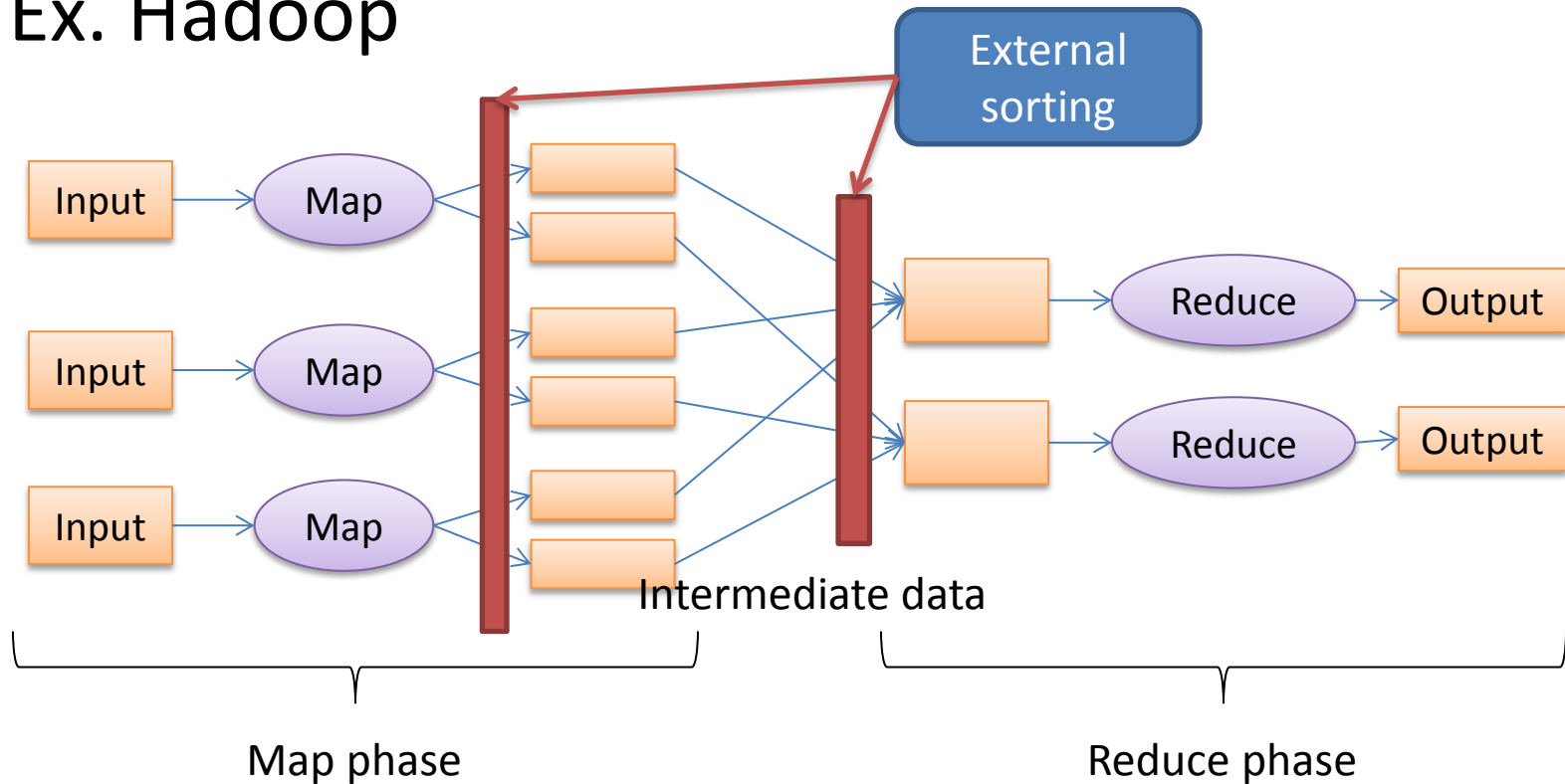
# Our Approach: Active SSDs

- Make results on-the-fly
  - Do in-storage processing when requested
  - Show results, but no writing
  - Eliminate extra data transfer
  - Increase the lifetime of SSDs



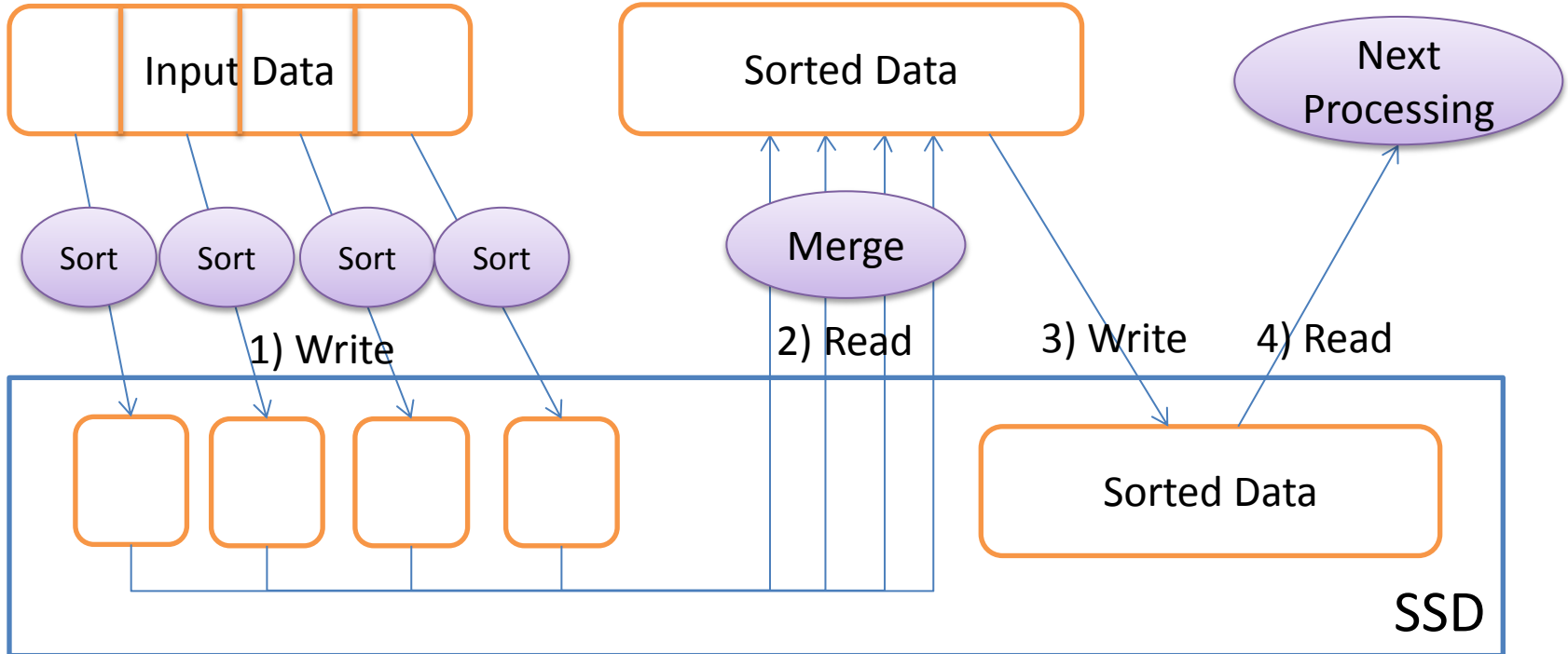
# External Sorting

- Sorting algorithm for large-scale data
- Ex. Hadoop



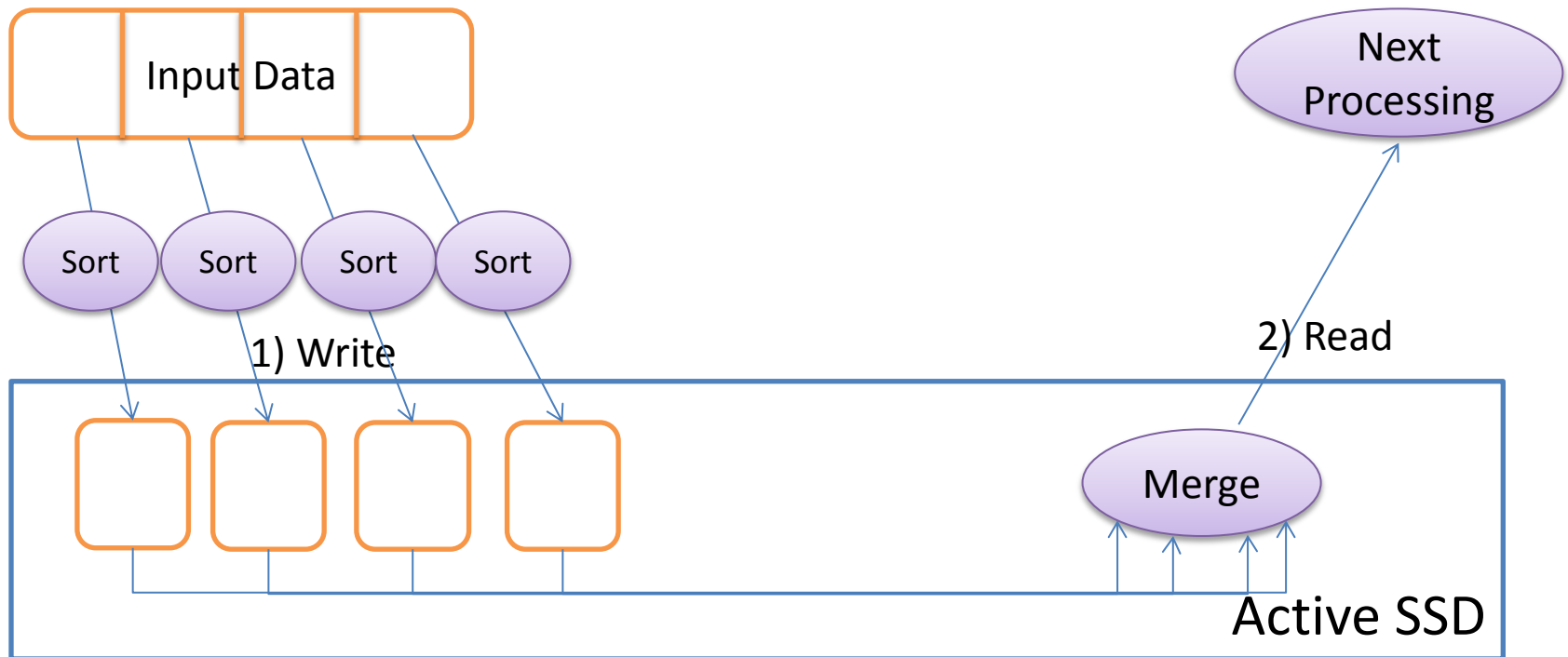
# External Sorting

- Data transfer in external sorting



# ActiveSort

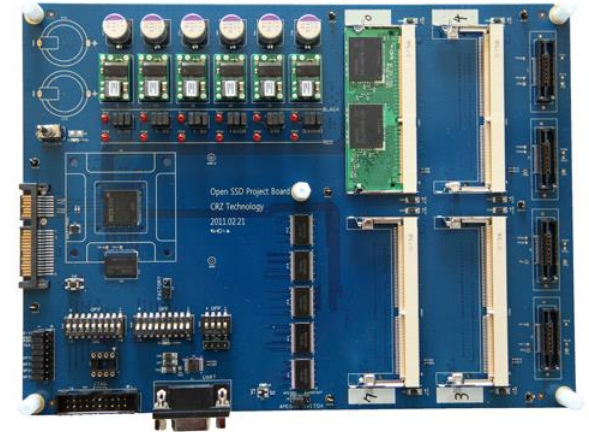
- On-the-fly data merge





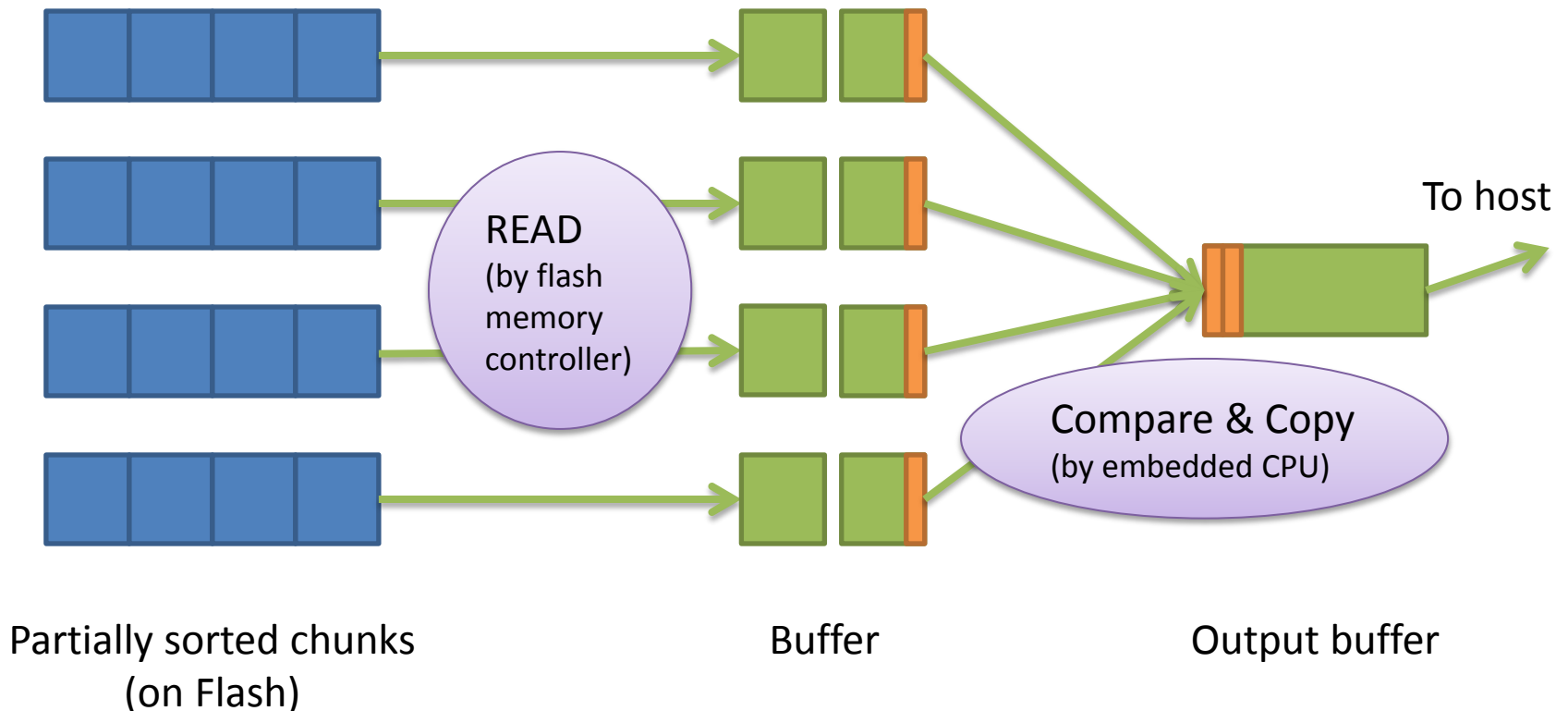
# Implementation

- Host
  - 3.4GHz Intel i5 CPU,  
16GB RAM (1-3GB in experiments)
- OpenSSD platform
  - 87.5MHz ARM CPU, 64MB DRAM,  
4 channels x 32GB flash memory modules, SATA-2
  - Simple page mapping + ActiveSort
- Parameter passing
  - Fix the location of files
    - Partially sorted chunks, final output, key&record conf.



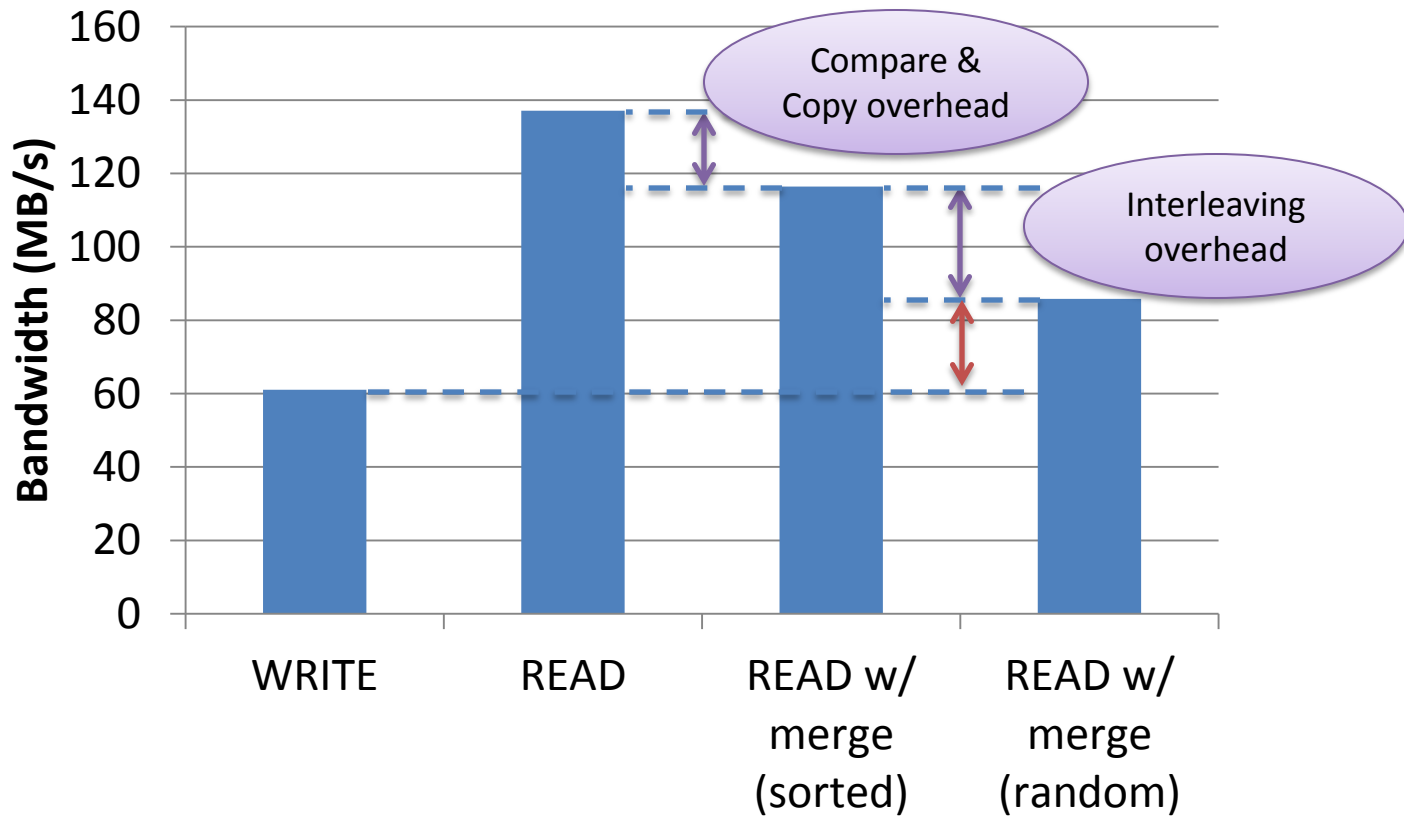
# Inside of Active SSDs

- Implementation of on-the-fly data merge



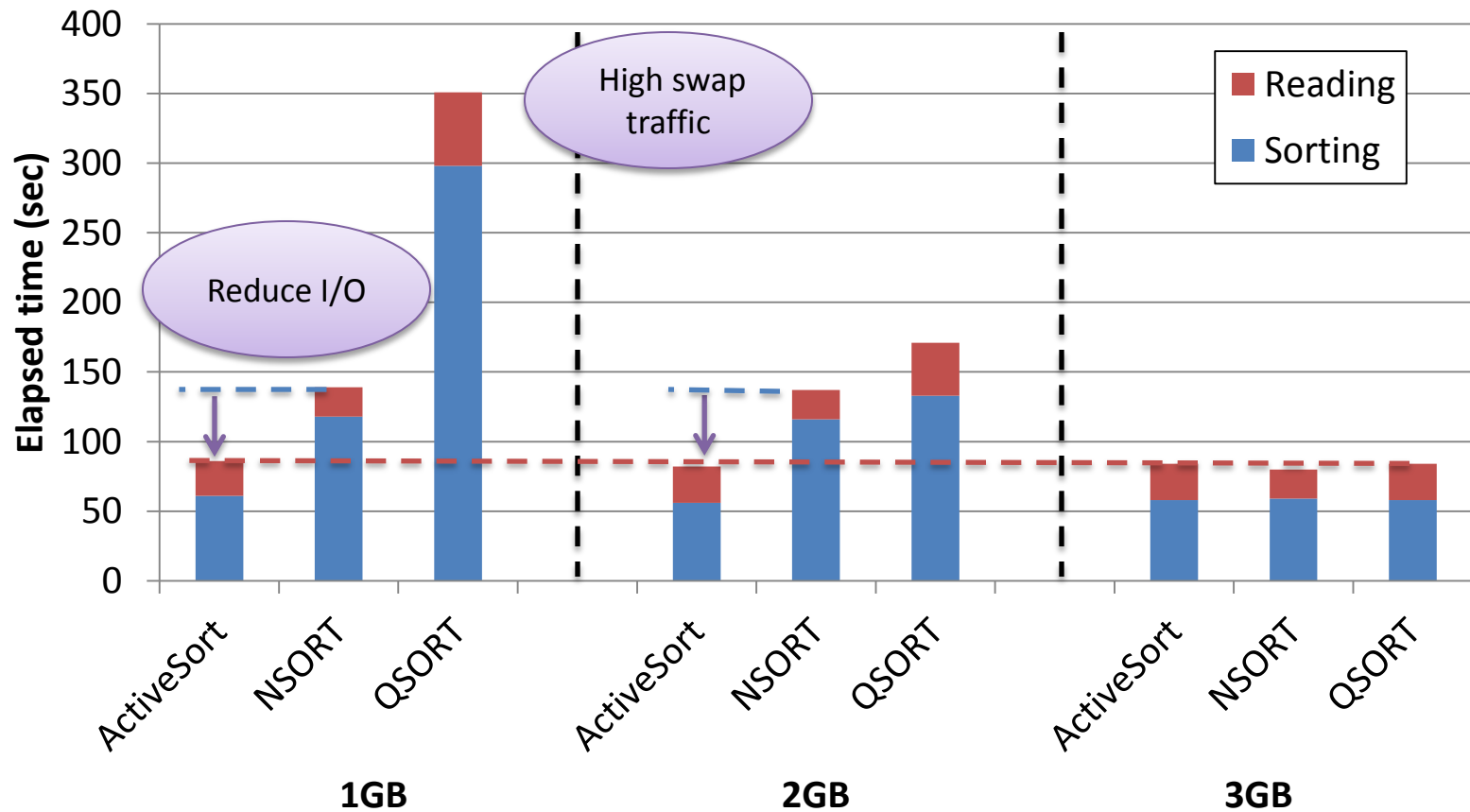
# Base Performance

- Sequential read/write



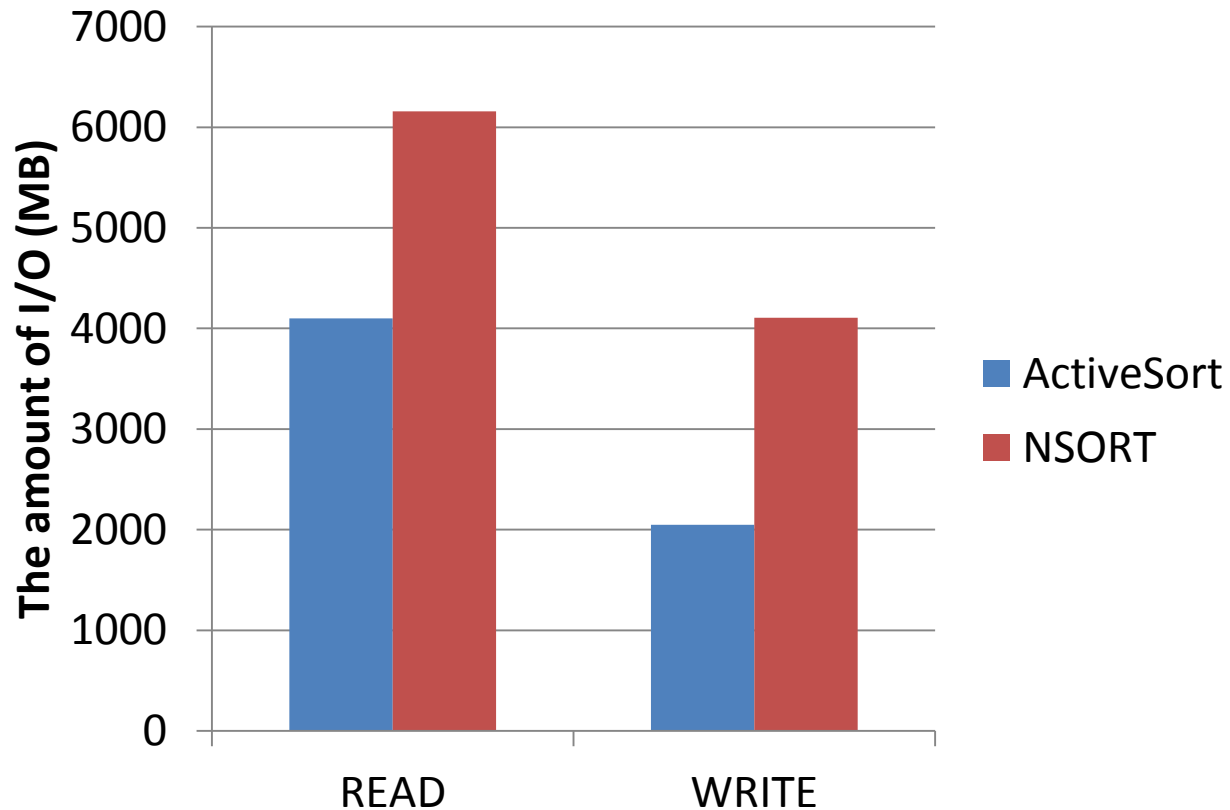
# Sort Benchmark

- Elapsed times of SORT and READ for 2GB data



# Sort Benchmark

- Amount of I/O – 1GB memory size

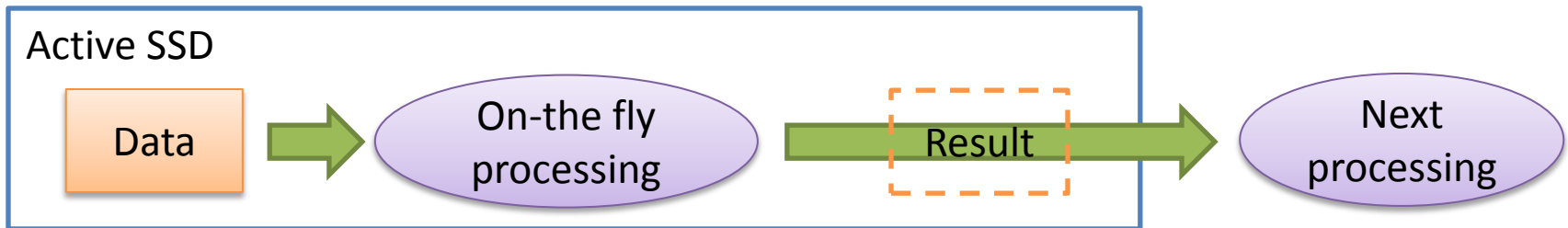


# Discussion

- Optimize on-the-fly merge
  - More cores, more RAM, higher internal bandwidth
  - Specialized hardware support
- Offload entire sorting
- Scale out by multiple active SSDs
- Provide API for parameter passing
  - Object-based interface

# Conclusion

- Active SSDs: in-storage, on-the-fly processing
  - ActiveSort: external sorting via on-the-fly data merge
  - Eliminate extra data transfer
  - Improve the performance and the lifetime of SSDs



- Future work
  - Integrate into Hadoop framework
  - Find another examples for active SSDs

Thank you  
Q & A

[yslee@calab.kaist.ac.kr](mailto:yslee@calab.kaist.ac.kr)