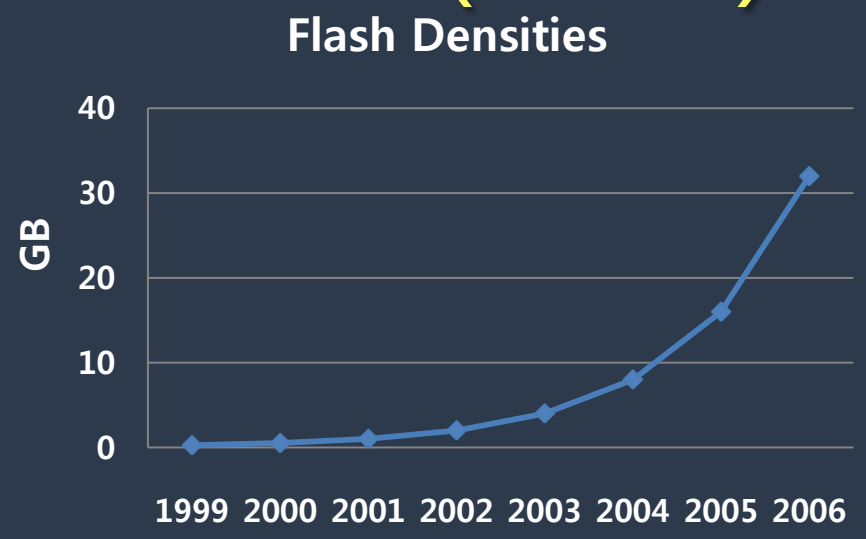
Join Processing for Flash SSDs: Remembering Past Lessons

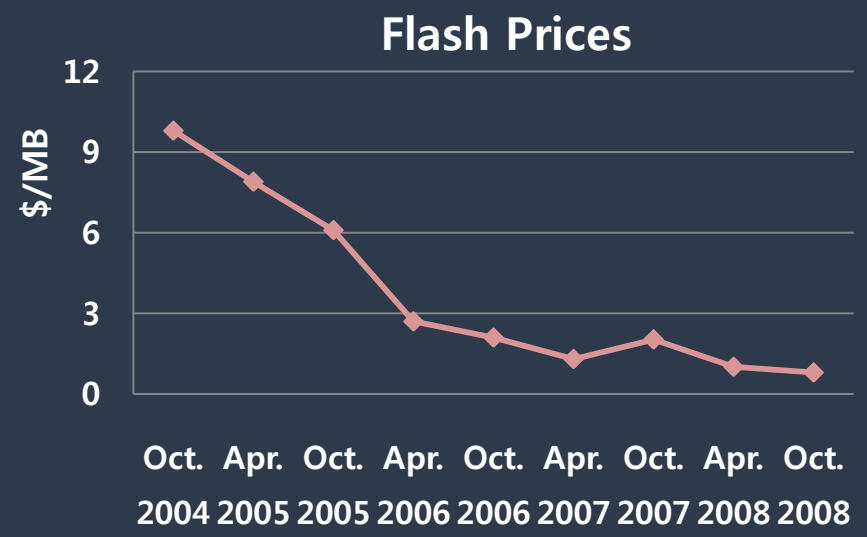
Jaeyoung Do, Jignesh M. Patel

Department of Computer Sciences
University of Wisconsin-Madison

Flash Solid State Drives (SSDs)

- Benefits of Flash SSDs
  - No seek time, small latency
- Flash Densities
  - 1995. 16 MB NAND Flash
  - 2005. 16 GB NAND Flash
  - 2010. 1 TB NAND Flash (predicted by Samsung)
- Flash Prices
  - Decreasing continuously





Flash SSDs for DBMSs

- Many previous works with flash SSDs
  - Flash-based DBMS [Gray ACM Queue 08, Graefe DaMoN07]
  - In-Page Logging [Lee VLDB07]
  - Transaction Processing [Lee VLDB08]
  - B+ Tree Index [Li ICDE09]
  - I/O Benchmarks [Bouganim CIDR09]

- We focus on **Join algorithms**
  - New join algorithms [Shah DaMoN08, Tsirogiannis SIGMOD09]

What lessons learnt for magnetic HDDs still apply to flash SSDs?

Goals

- Demonstrate the importance of recalling past lessons about efficient joins in magnetic HDDs

- Explore the effects of various parameters for joins on flash SSDs

1. Not inventing new join algorithms

2. Providing better insights for join performance on flash SSDs

Our Approach

- Investigate four popular ad hoc join algorithms
  - Block Nested Loops Join (BNL)
  - Sort-Merge Join (SM)
  - Grace Hash Join (GH)
  - Hybrid Hash Join (HH)
- Conduct experiments as varying various parameters
  - Memory buffer pool size
  - Page size
  - I/O unit size

Roadmap

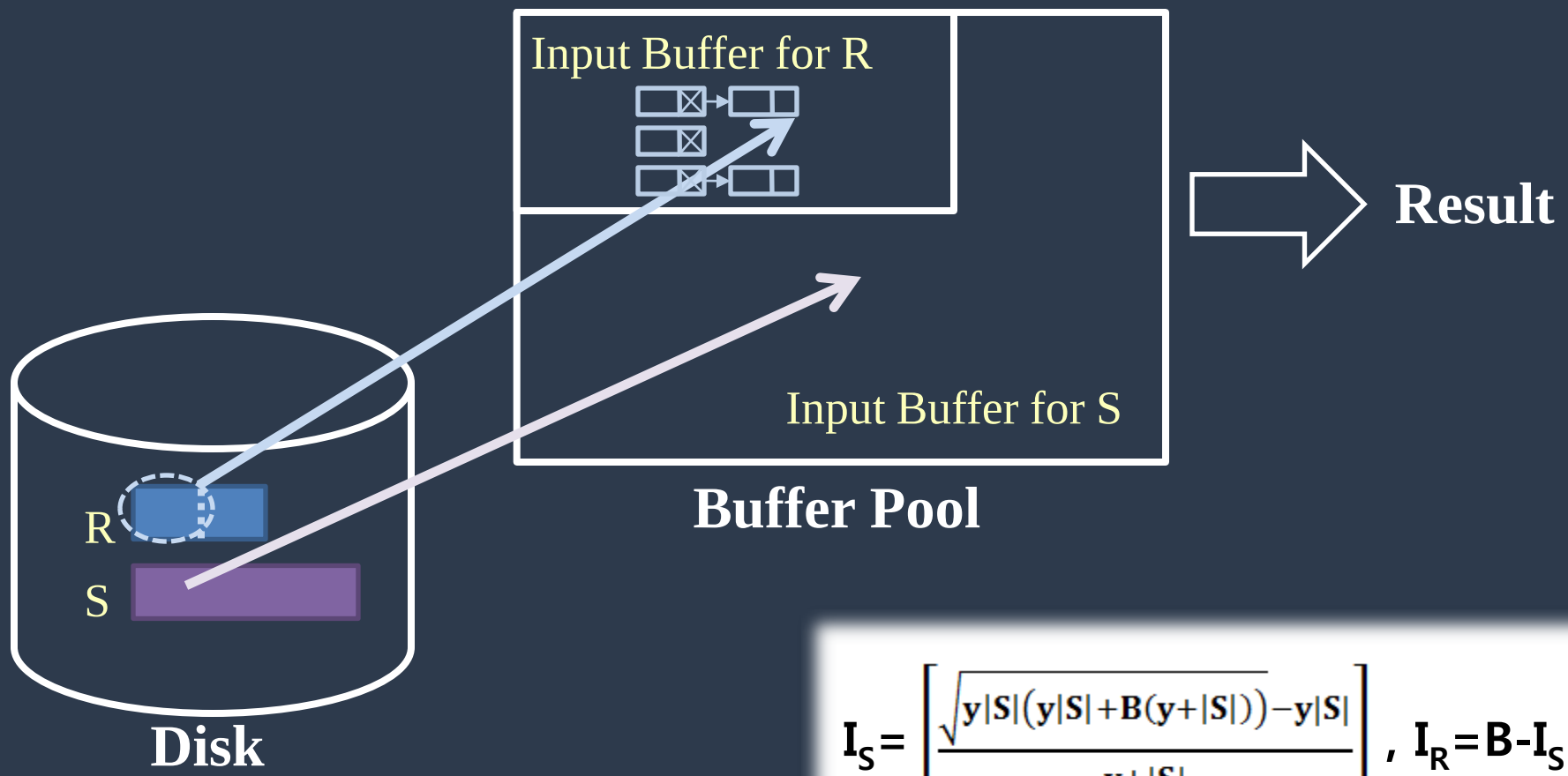
- Introduction / Goals
- Ad hoc join algorithms
- Experimental Results
- Conclusion / Future Work

Assumptions

- Blocked I/O is available
- We use the buffer allocation strategy tuned for magnetic HDDs [Haas et al. VLDB97]

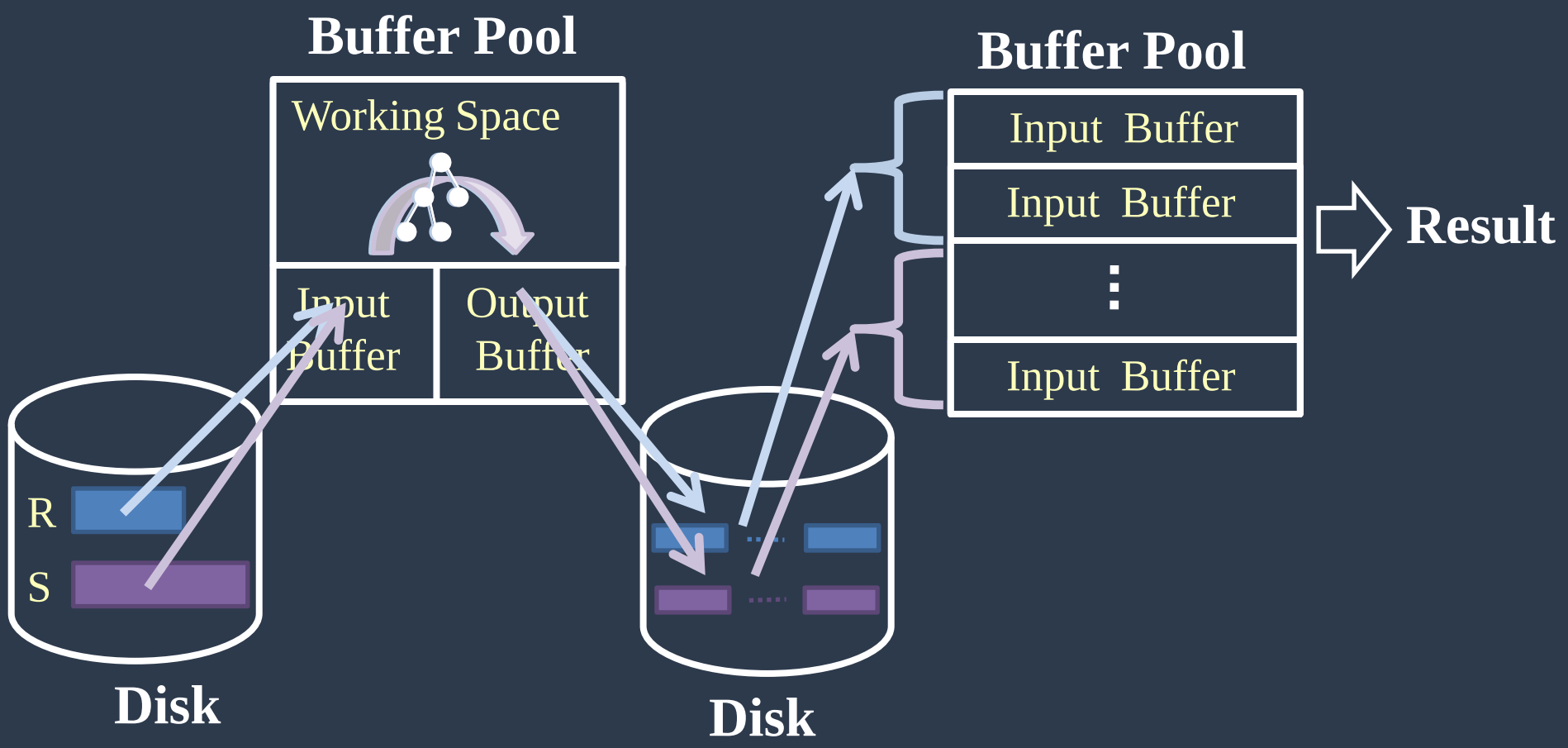
$I_S=$

Block Nested Loops Join

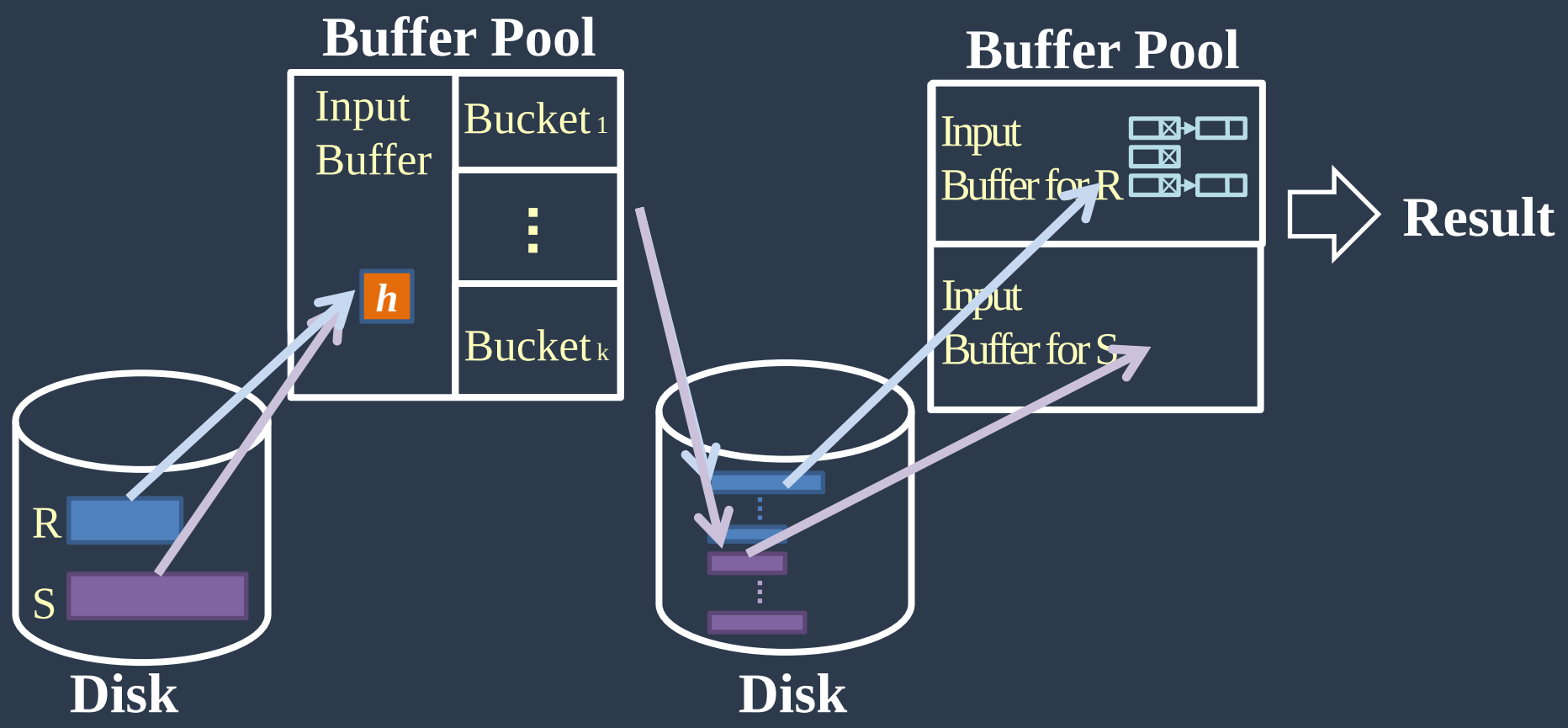


$$I_S = \left\lceil \frac{\sqrt{y|S|\left(y|S| + B(y+|S|)\right)} - y|S|}{y+|S|} \right\rceil, \quad I_R = B - I_S$$

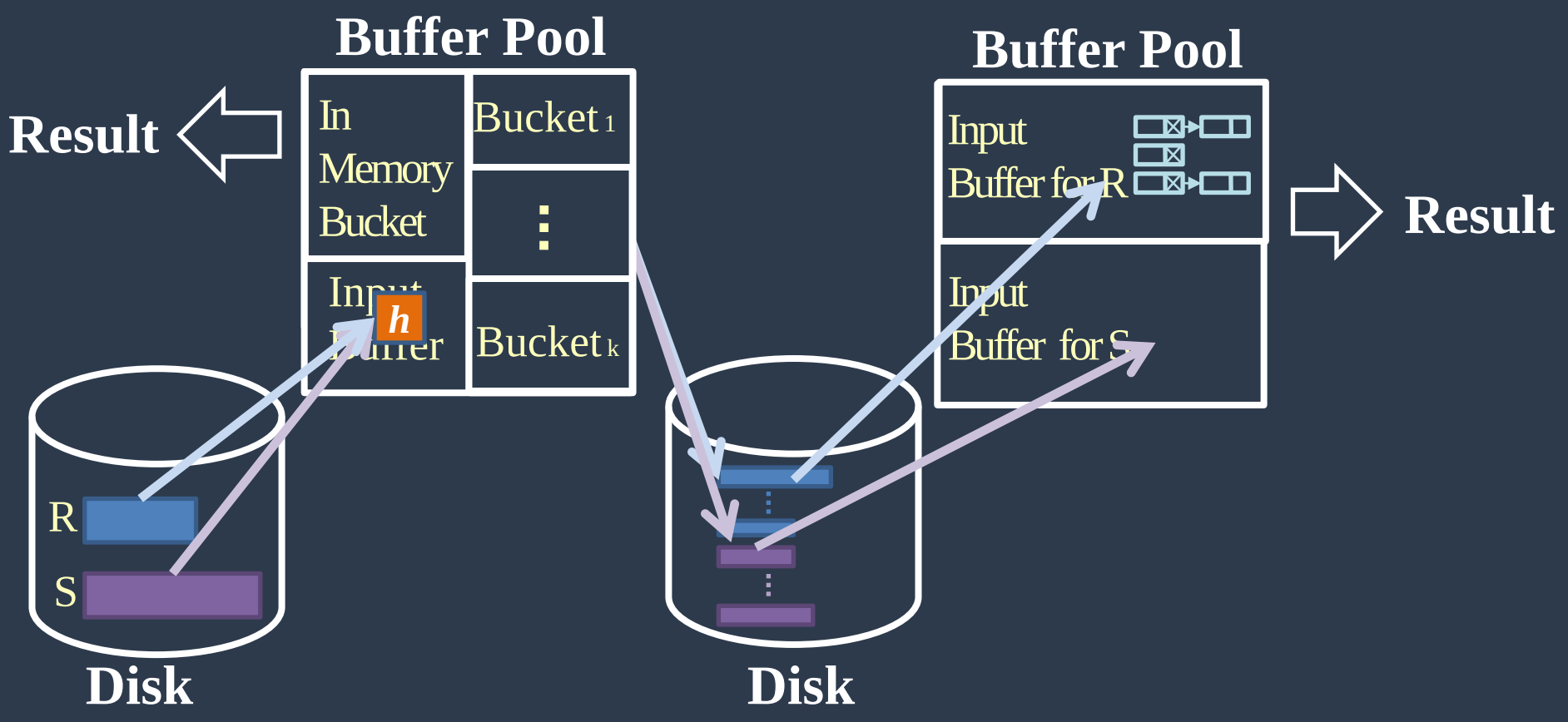
Sort-Merge Join



Grace Hash Join



Hybrid Hash Join



Roadmap

- Introduction / Goals
- Ad-hoc join algorithms
- Experimental Results
- Conclusion / Future Work

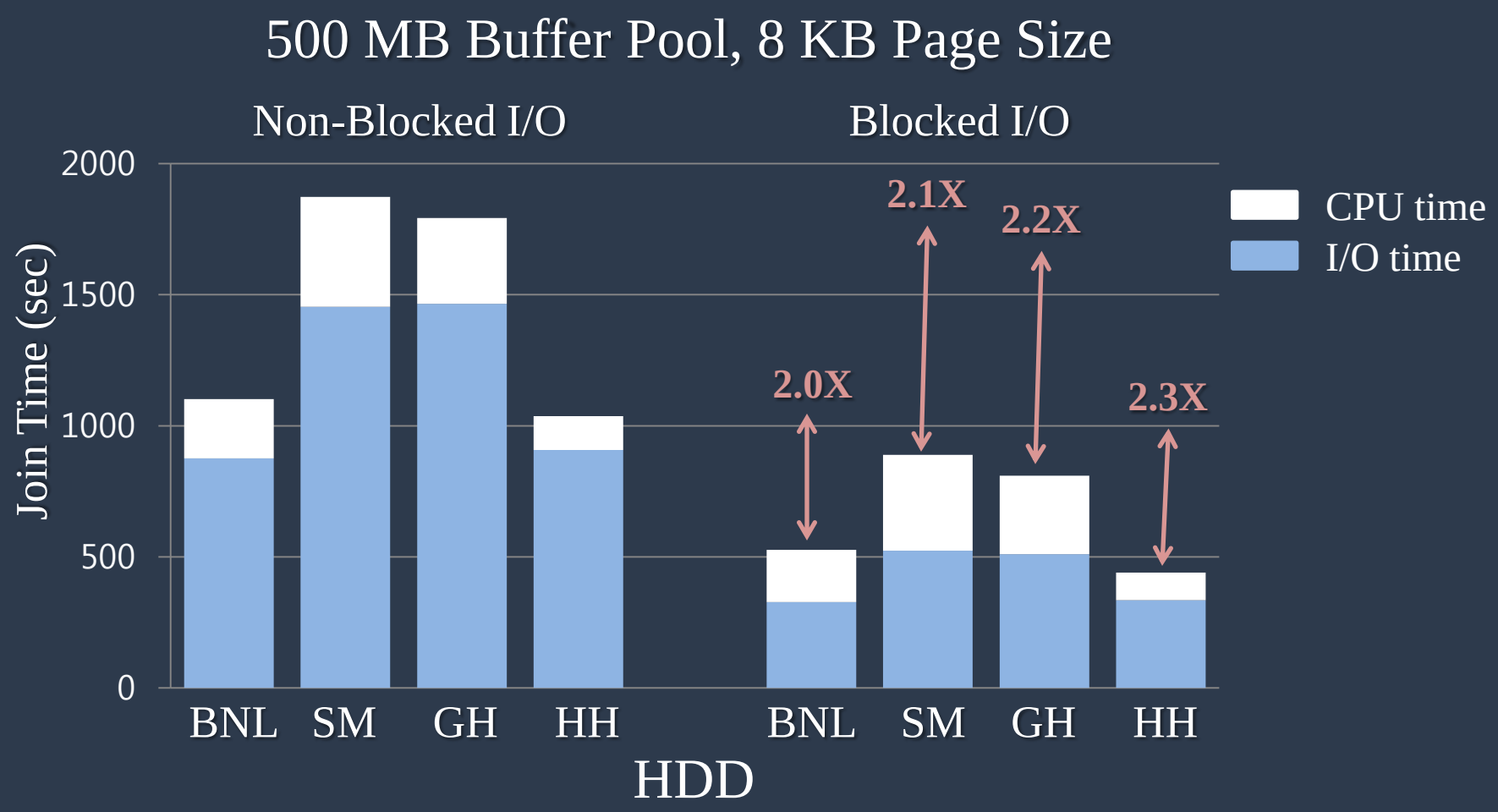
Experimental Setup

- A single-thread and light-weight database engine
- Flash SSD and magnetic HDD
  - OCZ Core Series 2.5" SATA 60 GB
  - TOSHIBA 5400 RPM 320 GB
- Data Set: TPC-H
  - Customer: 730 MB
  - Orders: 5 GB
- Platform
  - Dual Core 3.2 GHz Intel Pentium, Red Hat
  - Max. DB buffer pool size: 600 MB

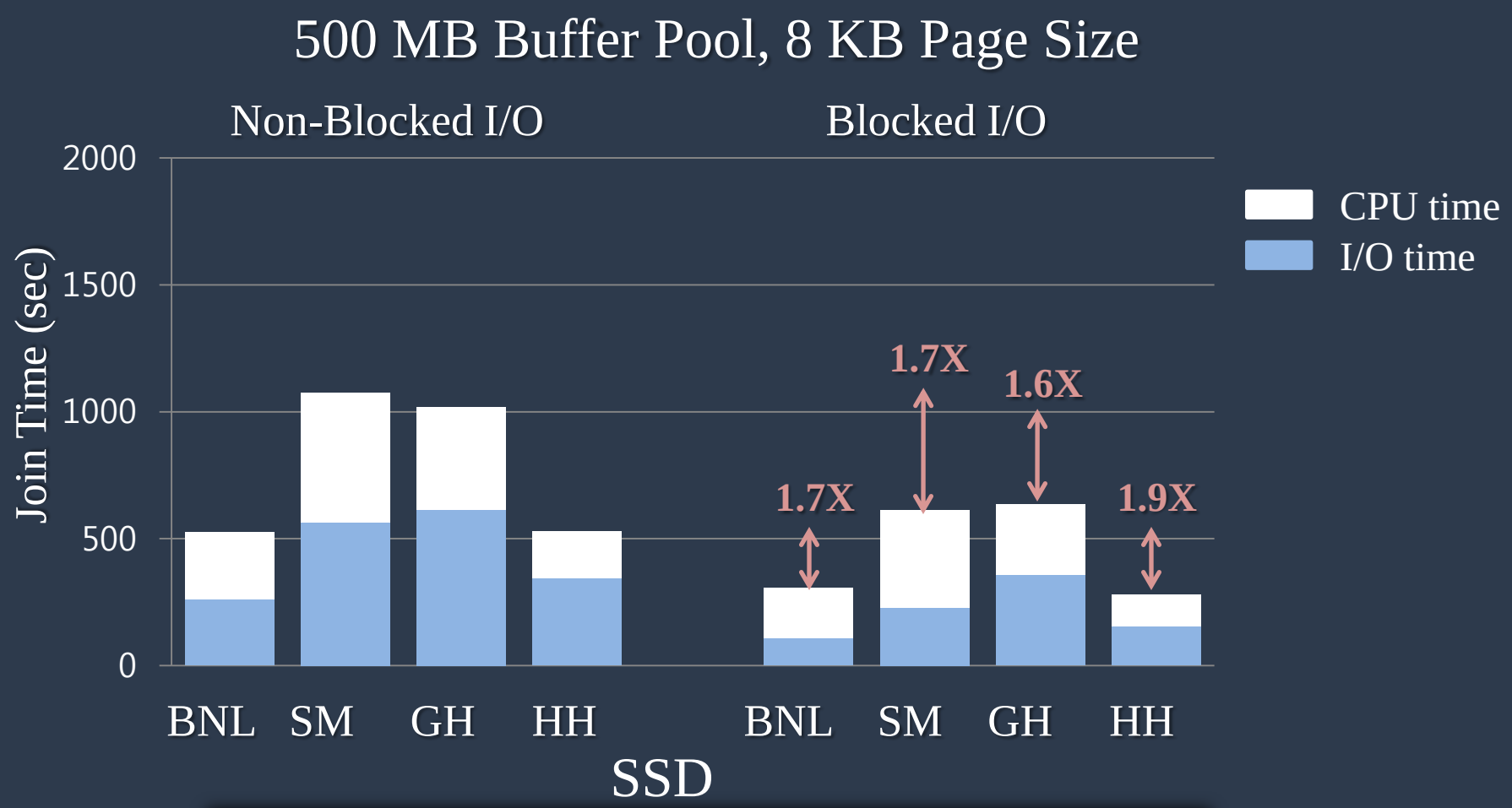
SSD		HDD
None	Avg. Seek Time	12 ms
0.35 ms	Avg. Latency	5.56 ms
120 MB/sec	Read Data Transfer Rate	34 MB/sec
80 MB/sec	Write Data Transfer Rate	34 MB/sec
230.99 $ (3.85 $/GB)	Price	129.99 $ (0.36 $/GB)

Source: OCZ and TOSHIBA

Effect of Blocked I/O (HDD)



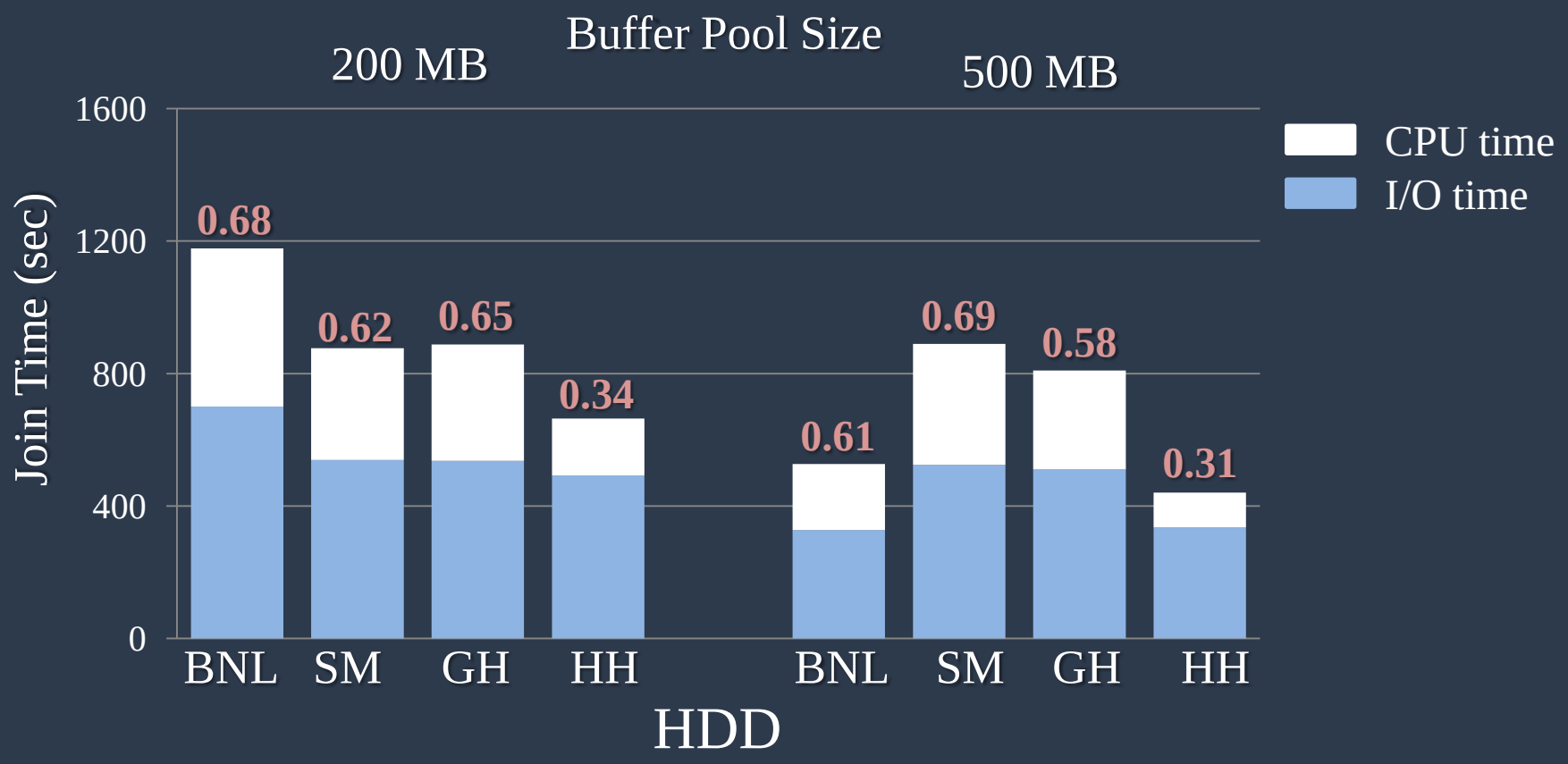
Effect of Blocked I/O (SSD)



Using blocked I/O is critical

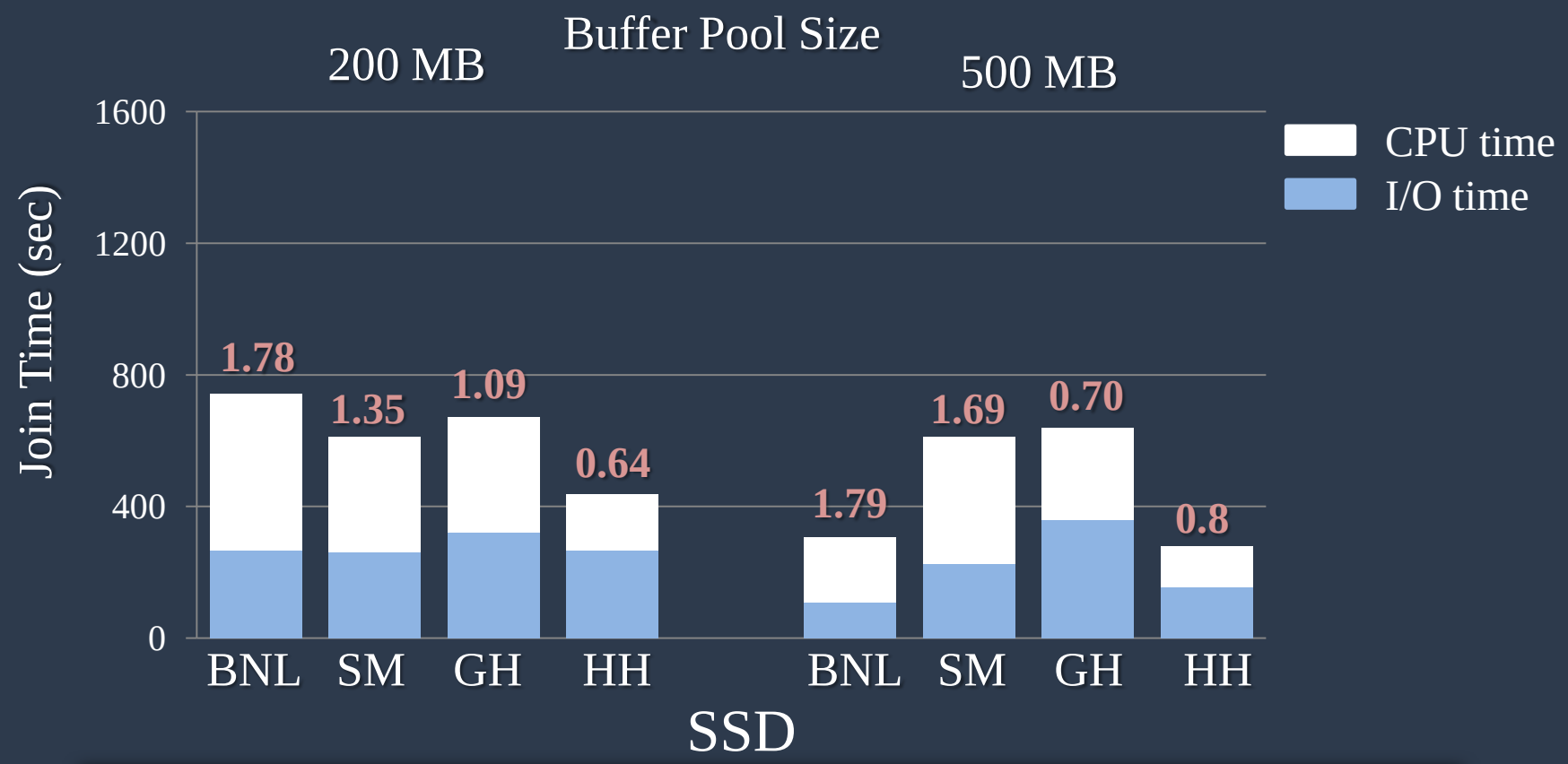
Joins are I/O Bound? (HDD)

8 KB Page Size, Blocked I/O



Joins are I/O Bound? (SSD)

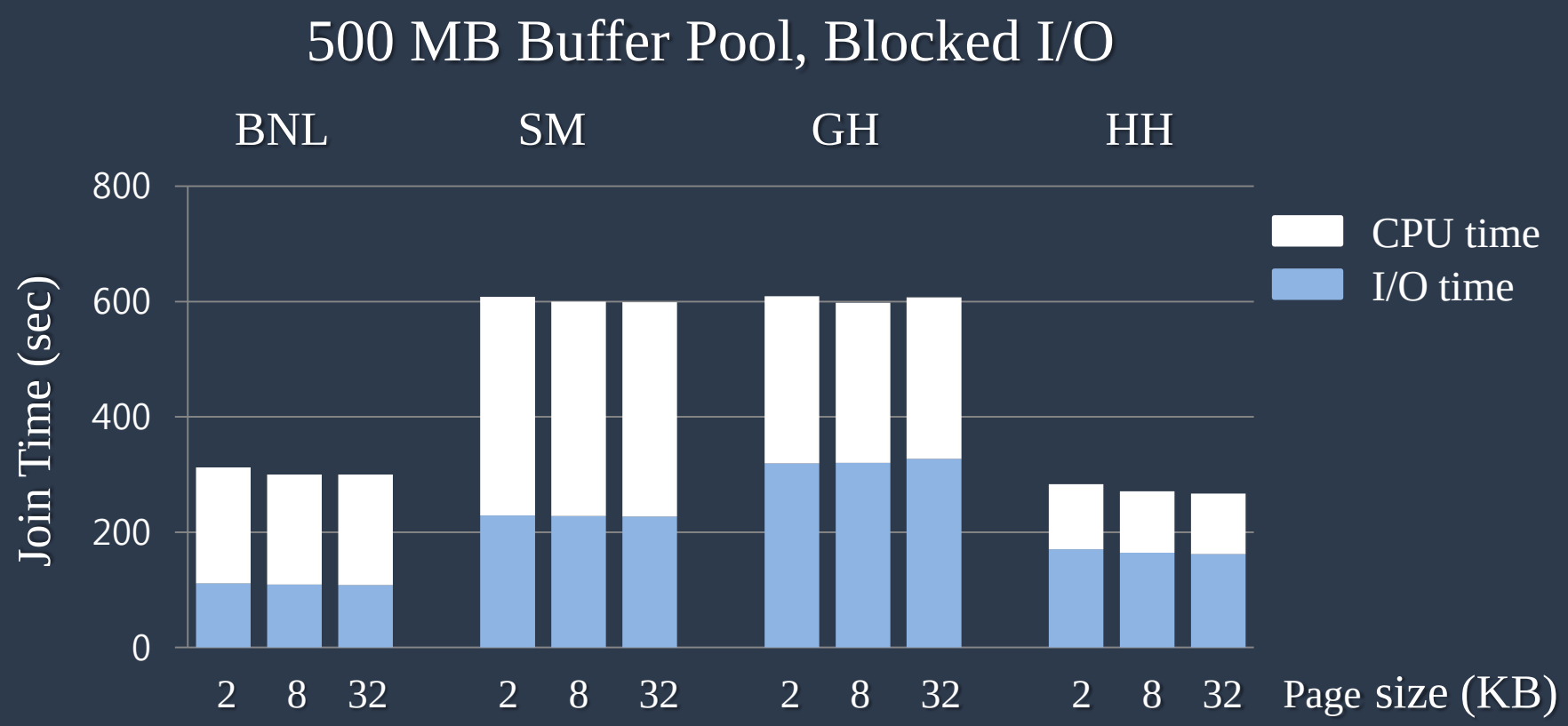
8 KB Page Size, Blocked I/O



Joins may become CPU-bound sooner

More details in our DaMoN'09 paper

Effect of Varying the Page Size (SSD)



When using Blocked I/O, the page size has a small impact on join performance

Conclusions / Future Work

- Traditional Join optimizations continue to be important with flash SSDs
  - Blocked I/O dramatically improves join performance
  - Buffer allocation strategy has an impact on join performance
  - It is even more critical to consider both CPU and I/O costs
- Future Work
  - Expand the range of hardware, and consider other HDD-based configurations
  - Derive detailed cost models for existing join algorithms, and explore the optimal buffer allocations for flash SSDs