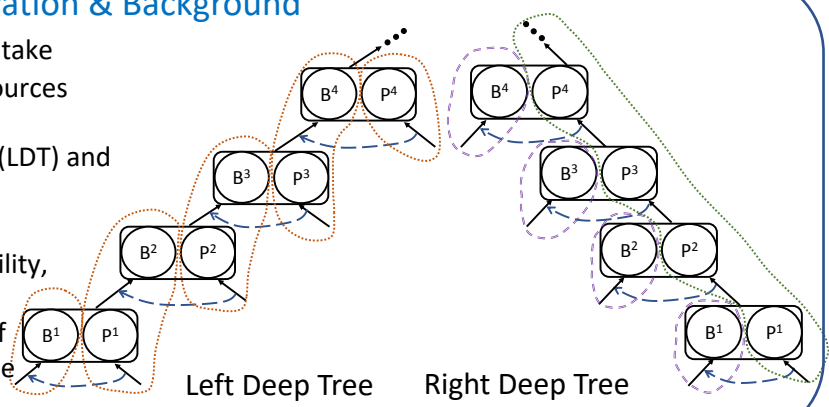
Re-evaluating the Performance Trade-offs for Hash-Based Multi-Join Queries

Shiva Jahangiri
University of California, Irvine
shivaj@uci.edu

Motivation & Background

- The execution tree of multi-Join queries may take many different shapes, each utilizing the resources differently
- We study the performance of Left Deep Tree(LDT) and Right Deep Tree(RDT) query plans, memory distributions for join operators, intra-query concurrency under different memory availability, and storage devices such as HDD and SSD
- AsterixDB is used to re-evaluate the results of one of the early and impactful studies from the 1990s, utilizing both HDD and SSD



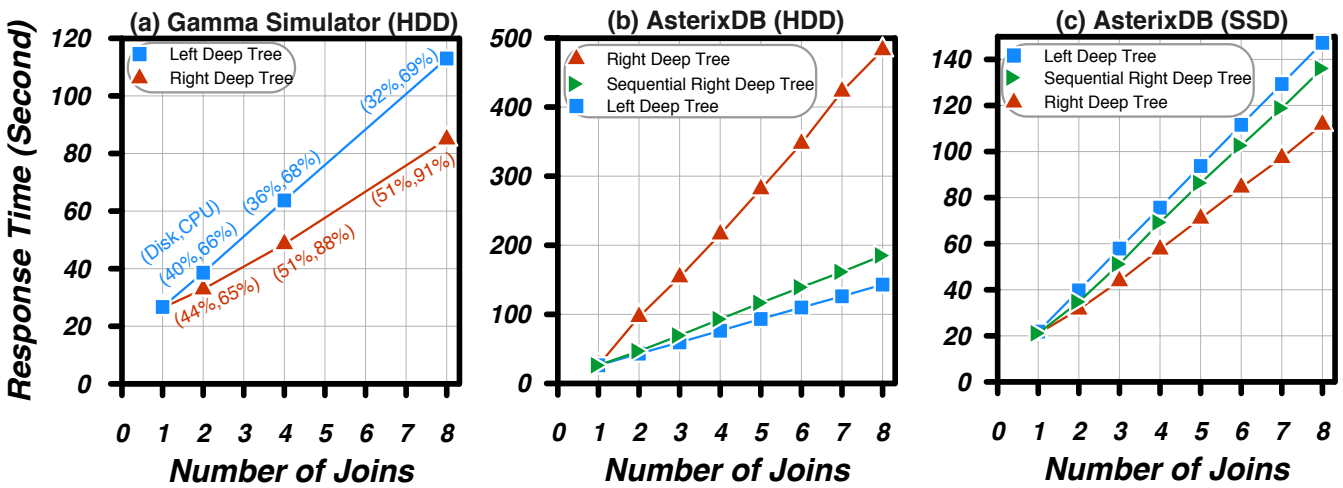
Query Plans:

- LDT: Low memory usage, less concurrent, less disk and CPU utilization
- RDT: High level of parallelism, high memory, disk, and CPU utilization

Memory Distributions & Concurrency Control:

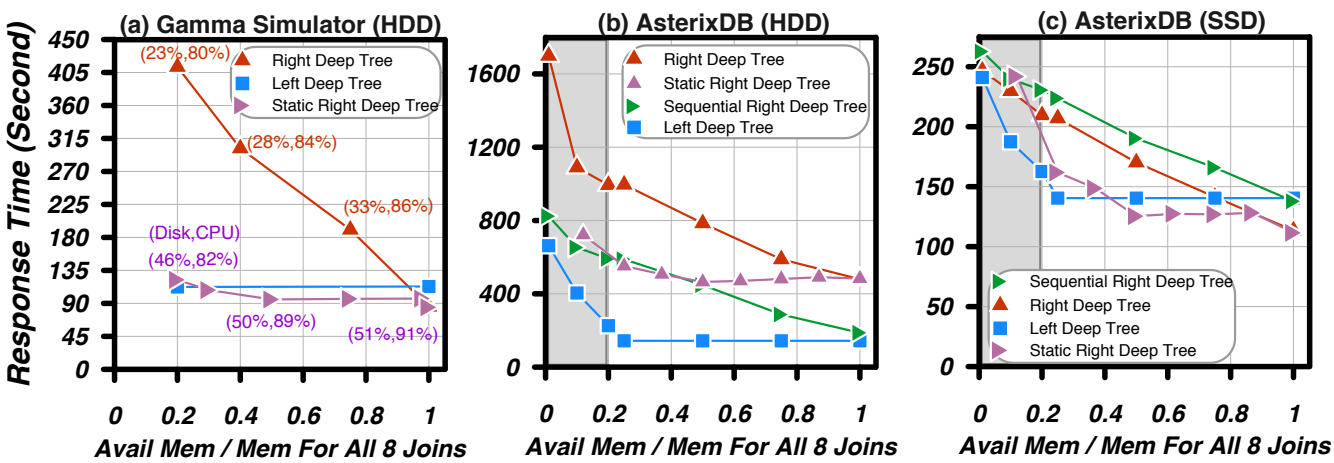
- LDT: Only 2 joins at a time
- RDT: Equal memory distribution, concurrent build phases
- Sequential RDT: Equal memory distribution, sequential builds
- Static RDT: Bottom-up memory distribution, write intermediate results to disk (break the tree) before spilling occurs

Unlimited Memory Experiment



- Lower random disk accesses and lower disk contention are the key to the performance of HDD
- Higher concurrency is the key to the performance using SSD
- Results of Gamma Simulator from 1990 are closer to SSD than HDD (no disk contention simulation)

Limited Memory Experiment



- Dividing memory among all joins in RDT causes more random I/Os for RDT & higher disk contention (Fig. b)
- RDT takes advantage of concurrency in SSD in case of more available memory

Results & Conclusions

- **Importance of underlying storage device:** With HDD, sequential plans have better response times due to lower disk contention while for SSD the higher concurrency leads to better response times due to the absence of arm-related disk contention
- **Importance of verification of simulators:** While useful, simulators may lead to incorrect conclusions if not verified against real systems carefully
- **Importance of re-evaluation of previous studies:** Our study shows that the re-evaluation of previous results is periodically necessary due to improvements in the underlying hardware