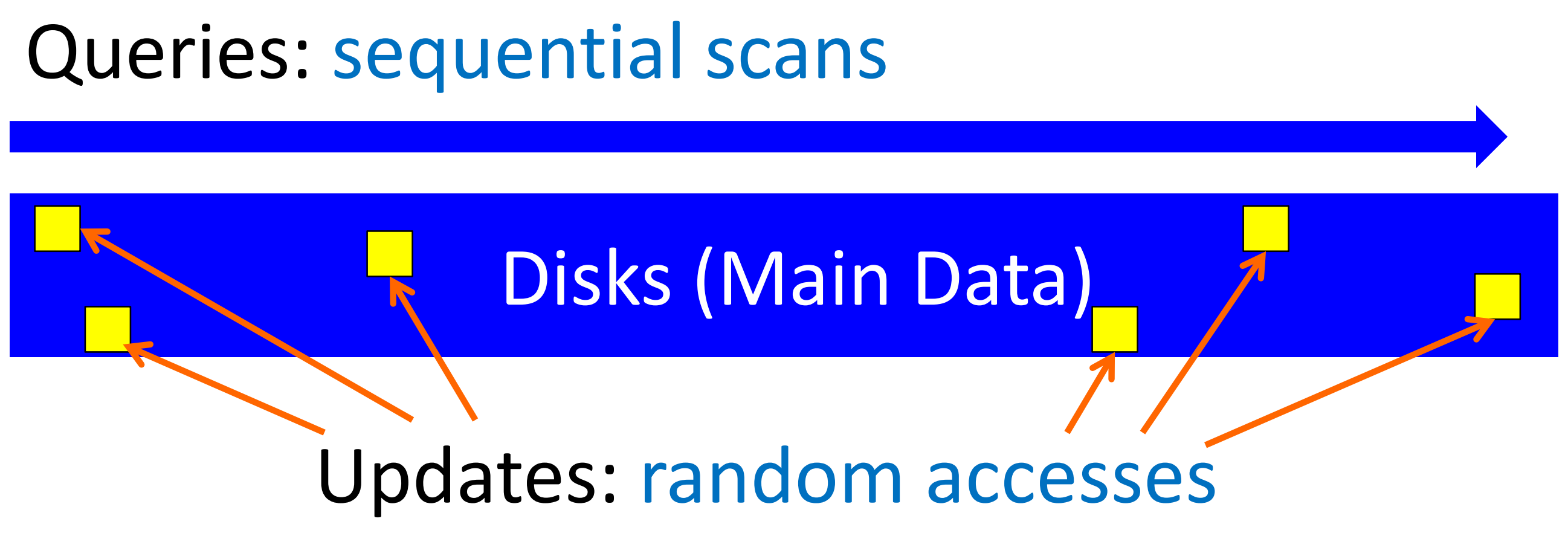
ENABLING ONLINE UPDATES IN DATA WAREHOUSES VIA SSDS

Shimin Chen (HP), Anastasia Ailamaki, Manos Athanassoulis, Radu Stoica (EPFL), Phillip Gibbons (Intel Labs)

MOTIVATION

- Data warehouse and business intelligence
  - Fast growing multi-billion dollar market
  - Traditionally optimized for read-only query performance
  - Allowing only offline updates at night
  - Trade off data freshness for performance
- Online updates increasingly desirable
  - Online & other quickly reacting businesses
  - 24x7 operations for global markets
- Our goal: Enabling online updates without sacrificing query performance

PROBLEM OF CONVENTIONAL APPROACH



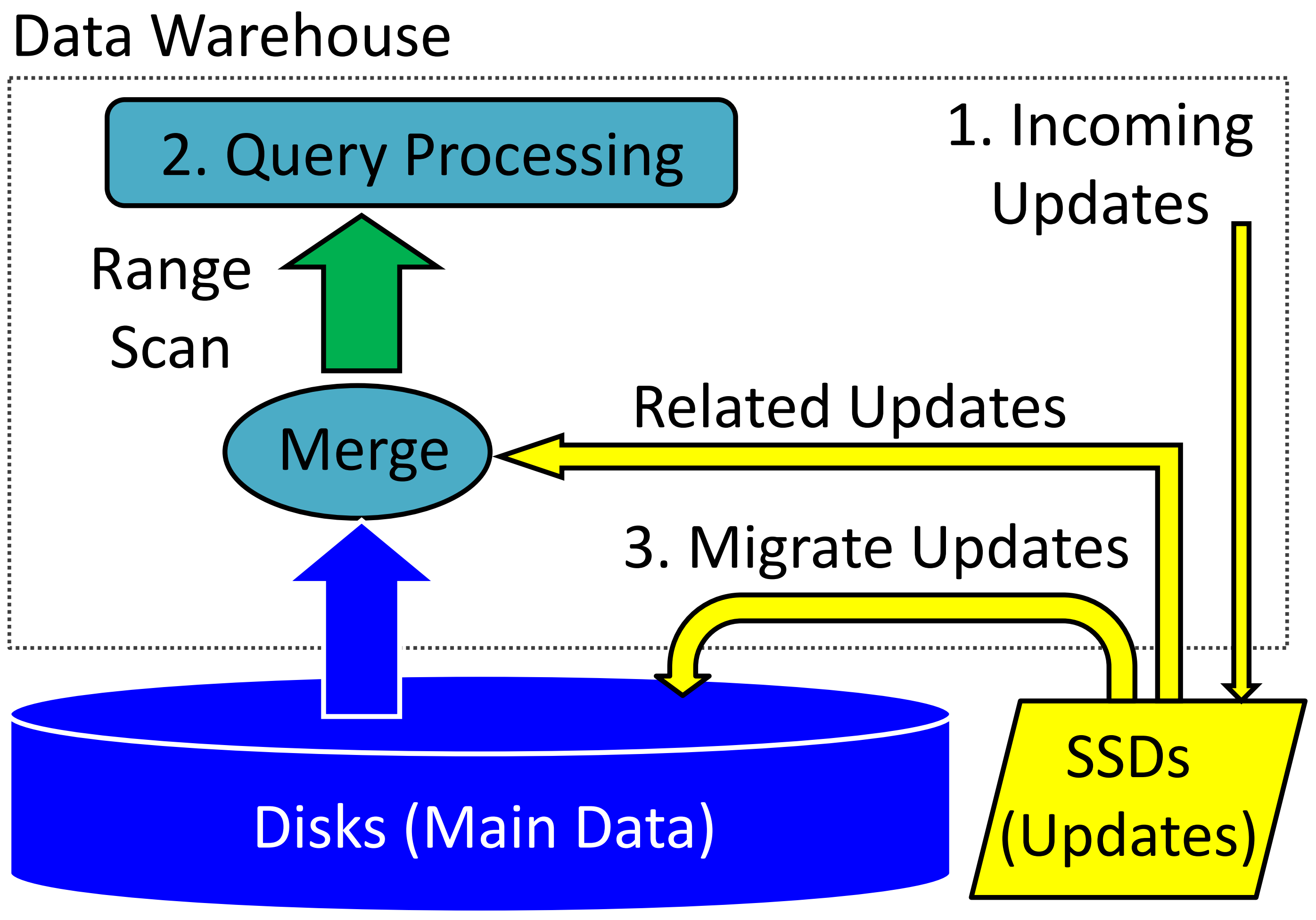
Queries: sequential scans

Disks (Main Data)

Updates: random accesses

Intermixing random updates with queries disturb the good sequential scan patterns of the large data analysis queries

OUR APPROACH



Data Warehouse

2. Query Processing

1. Incoming Updates

Range Scan

Merge

Related Updates

3. Migrate Updates

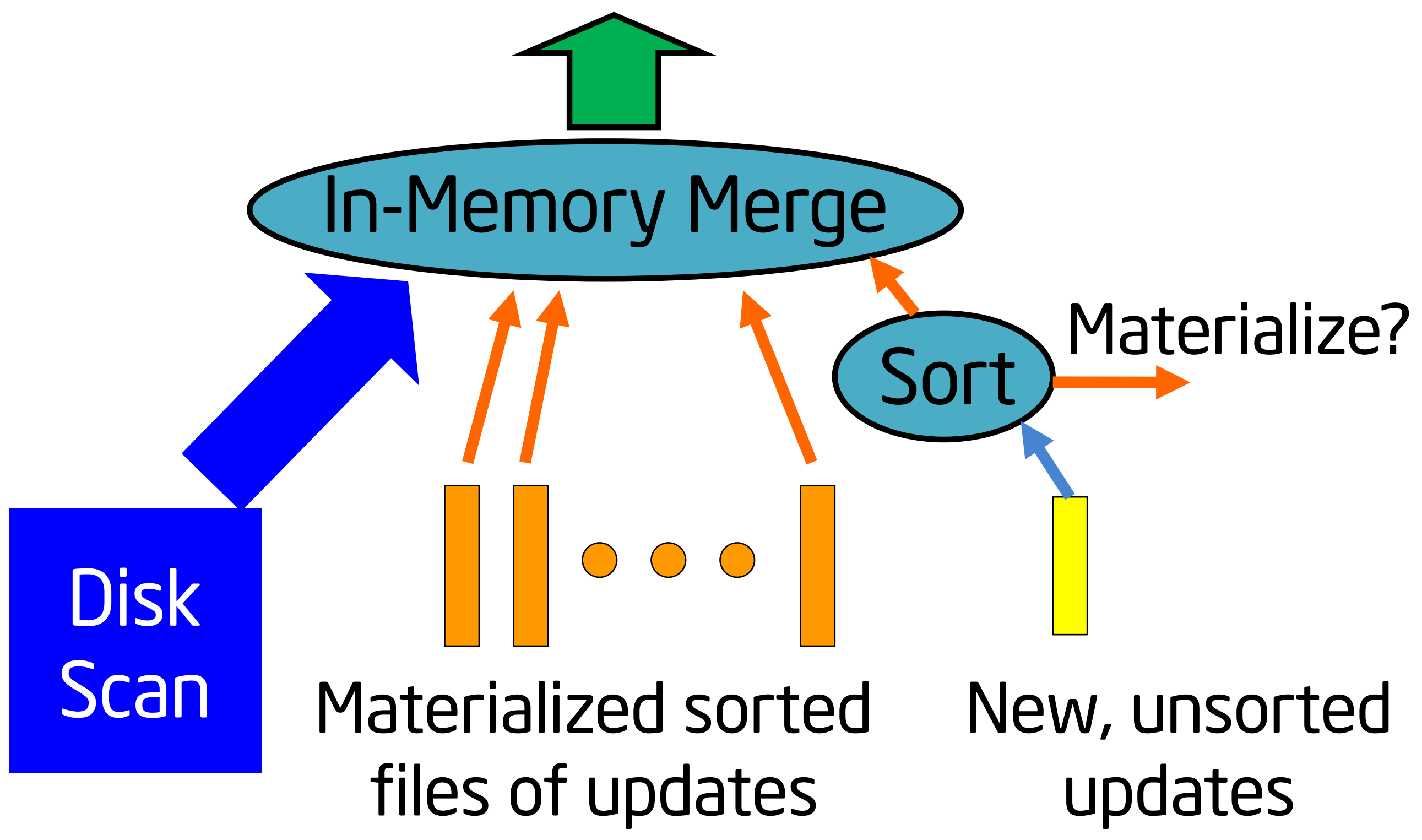
Disks (Main Data)

SSDs (Updates)

SSDs used as non-volatile cache for updates

MaSM ALGORITHM

- Materialized Sort-Merge Algorithm
  - Sort updates in main data order
  - Merge sorted updates with main data
  - Materialize and re-use sorted files



In-Memory Merge

Sort

Materialize?

Disk Scan

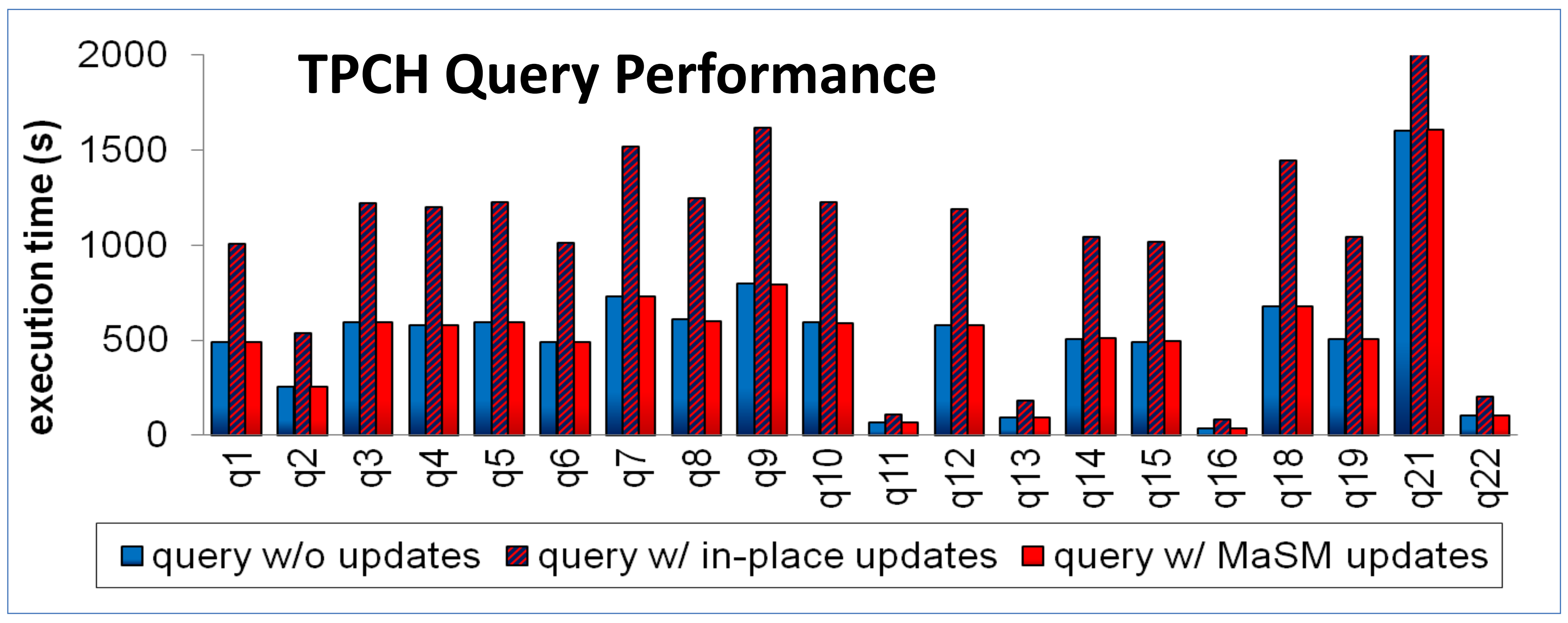
Materialized sorted files of updates

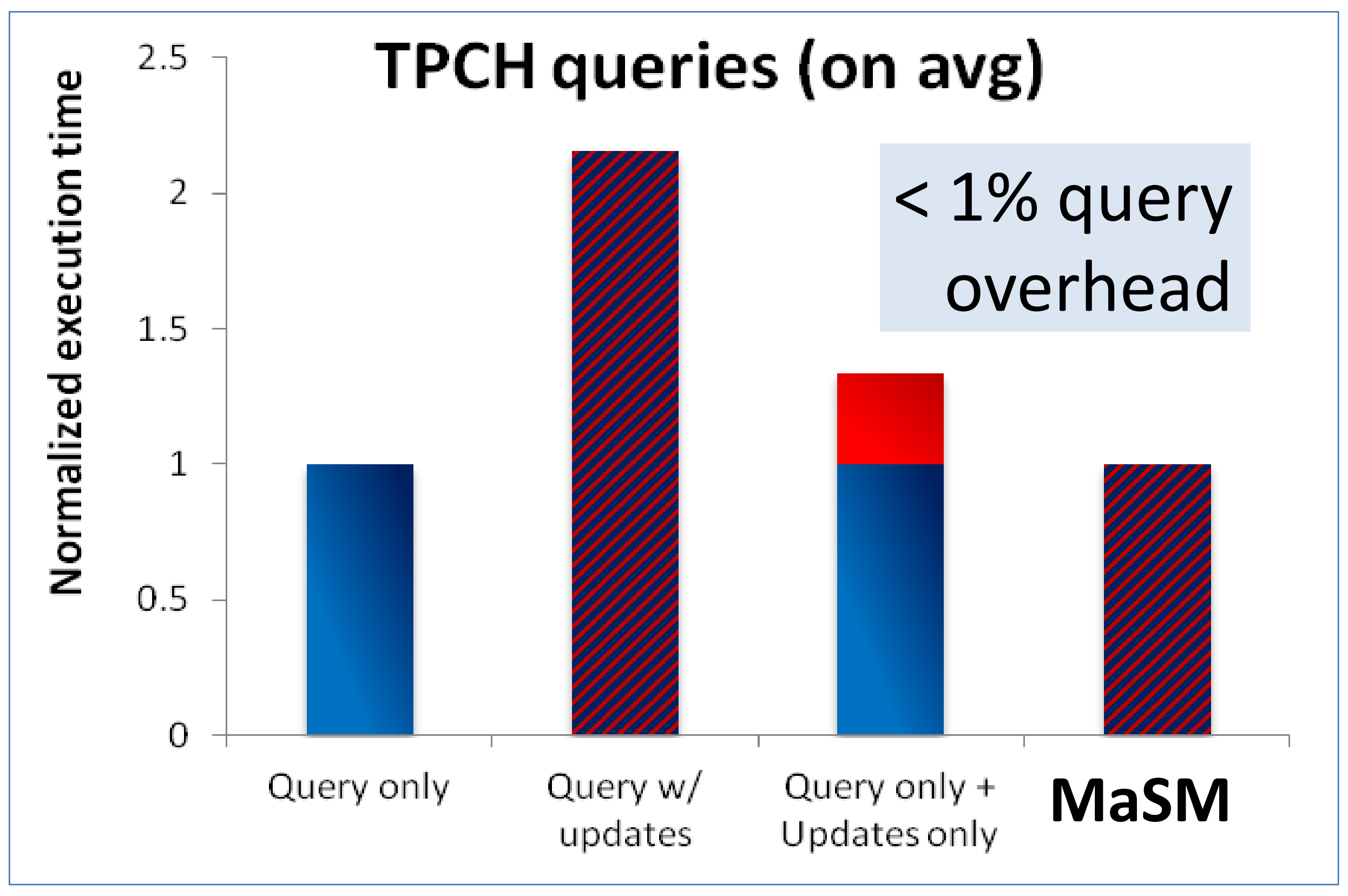
New, unsorted updates

EXPERIMENTAL RESULTS

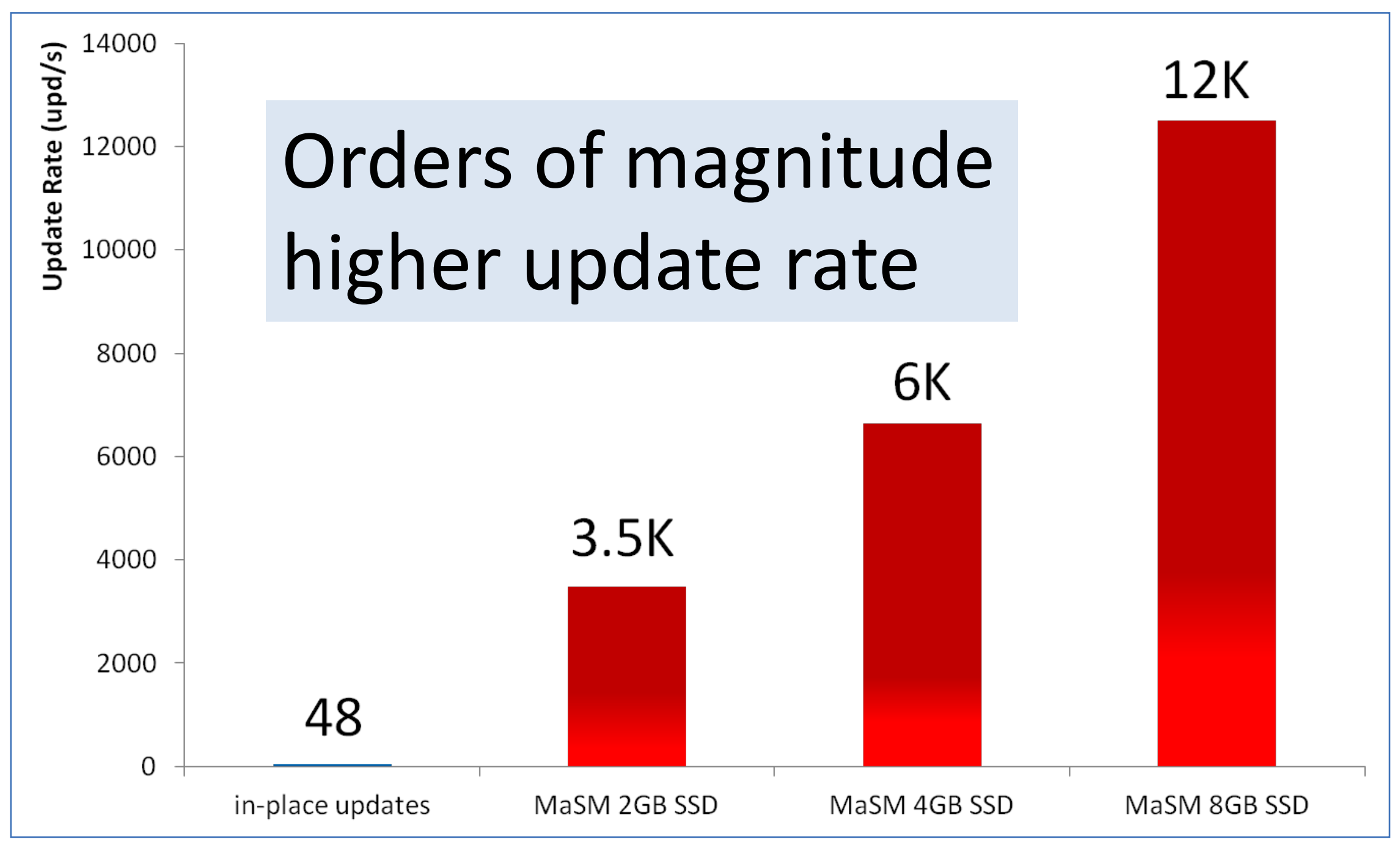
Setup:

- Dell Precision 690
- 100 GB main data on disk
- 4 GB flash space on Intel X25-E SSD
- Replay of TPCH disk trace from commercial DB





< 1% query overhead



Orders of magnitude higher update rate

Update Approach	Freshness	Performance	↓ mem overhead
Batched	X	☺	☺
In place	☺	X	☺
In-memory differential	☺	☺	X
MaSM and SSD	☺	☺	☺

Paper in Sigmod'11