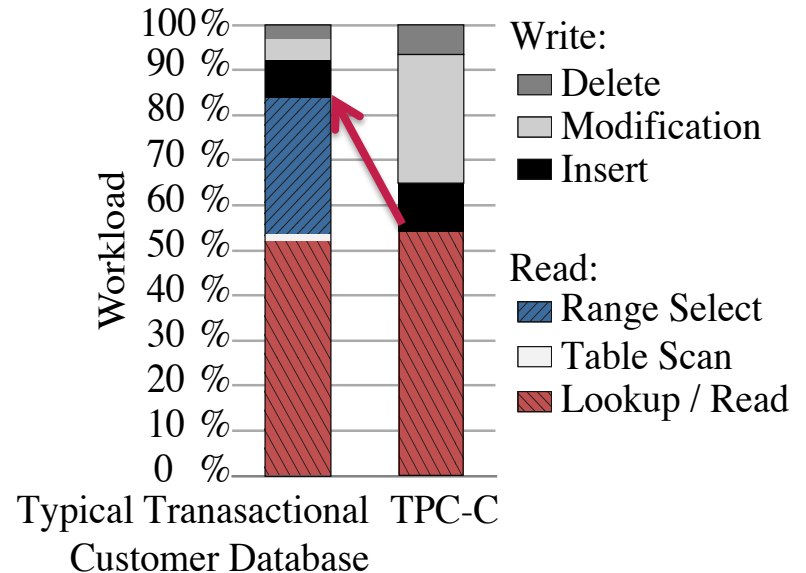# In-Memory Only

Alexander Zeier

Hasso-Plattner-Institute

Motivation

- **Enterprise applications have evolved: not just OLAP vs. OLTP**

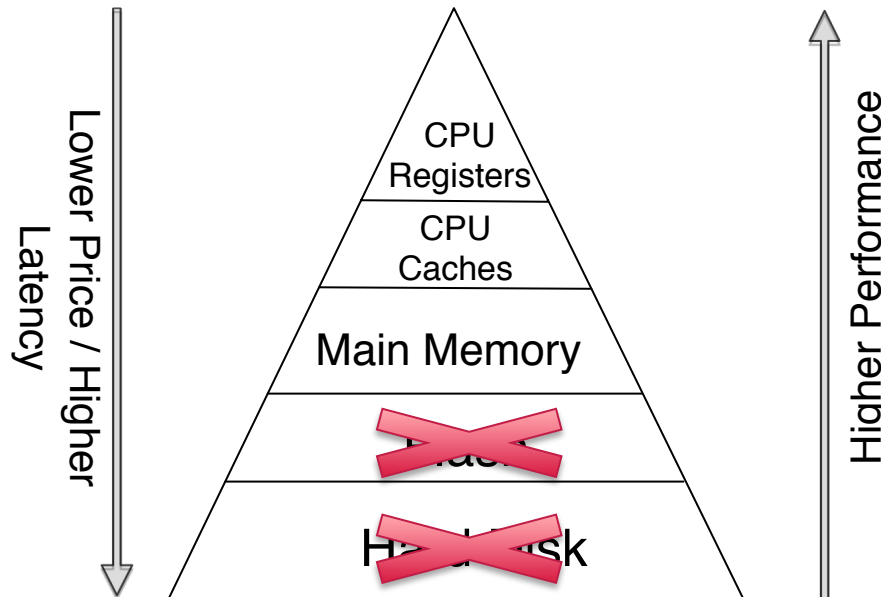 - □ Customer analysis shows a widening "read"-gap between transactional and analytical queries

Write:
- ■ Delete
- ■ Modification
- ■ Insert

Read:
- ■ Range Select
- □ Table Scan
- ■ Lookup / Read

Typical Transactional Customer Database   TPC-C

- **Requirements**

  - □ Demand for real-time analytics on transactional data
    - – More flexible, more dynamic data management

  - □ High throughput transactions

- **Example – Real-time Available-To-Promise Check directly on transactional data during order entry, without materialized aggregates of available stocks.**

- Main Memory becomes <u>cheaper and larger</u>
- Main Memory sub-systems become faster and <u>more scalable</u>

Lower Price / Higher Latency

CPU Registers

CPU Caches

Main Memory

~~Flash~~

~~Hard Disk~~

Higher Performance

- Disk is Tape, Flash is Disk (is Dead), RAM Locality is King[1]
  - <u>No more secondary storage!</u>

[1]In-Memory of Jim Gray

Approach

- Overall data management system changes
  - In-Memory Only
  - Vertically partitioned
  - CPU-Cache Optimized
  - Only one optimization objective – main memory access

- Rethink how enterprise application persistence is build
  - Leaner architecture (less layers)
  - Computational application logic closer to the database