OldSQL vs. NoSQL vs. NewSQL

CMPT 843, SPRING 2018

JIANNAN WANG

OLTP

OnLine Transaction Processing

Workload
High-frequent Updates + Small Queries

OLAP

OnLine Analytical Processing

Workload
Low-frequent Updates + Big Queries

What Happened To OLTP?

OldSQL (1970 – Now)

NoSQL (2000 – Now)

NewSQL (2010 – Now)

OldSQL (1970 – Now)

Traditional SQL vendors

Microsoft SQL Server   **ORACLE DATABASE**   **IBM DB2**   . . .

Still very big market!!!

Limitation 1: Not Scalable

Limitation 2: Pre-defined Schema

The advent of Web 2.0

Read-only Web → Read-write Web

Highly Scalable
◦ Scale to 1,000,000 users and 1000 servers

Highly Available
◦ Available 24 hours a day, 7 days a week

Highly Flexible
◦ Flexible schema and flexible data types

NoSQL Pioneers

Memcached [Fitzpatrick 2004]
◦ In-memory indexes can be highly scalable

BigTable [Chang et al. 2006]
◦ Persistent record storage could be scaled to thousands of nodes

Dynamo [DeCandia et al. 2007]
◦ Eventual consistency allows for higher availability and scalability

NoSQL Categories

NoSQL	Data Model	Example Systems
Key-value Stores	Hash	DynamoDB, Riak, Redis, Membase
Document Stores	Json	SimpleDB, CouchBase, MongoDB
Wide-column Stores	Big Table	Hbase, Cassandra, HyperTable
Graph Database	Graph	Neo4J, InfoGrid, GraphBase

NoSQL Limitations

Low-level Language
- Simple read/write database operators
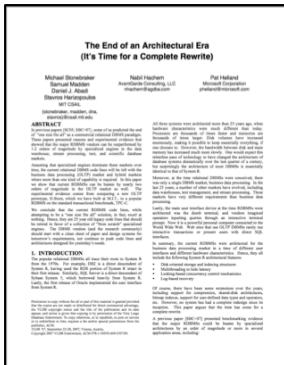
Weak Consistency
- Eventual Consistency

Lack of Standardization
- 100+ NoSQL systems

NewSQL

Strong Consistency + **High Scalability**

The end of an architectural era:(it's time for a complete rewrite)
M Stonebraker, S Madden, DJ Abadi… - Proceedings of the 33rd …, 2007 - dl.acm.org
Abstract In previous papers [SC05, SBC+ 07], some of us predicted the end of" one size fits all" as a commercial relational DBMS paradigm. These papers presented reasons and experimental evidence that showed that the major RDBMS vendors can be outperformed ...
Cited by 580   Related articles   All 55 versions   Cite   Save

90% query time spent on overhead

NewSQL Market

Limitations

- Scalable but not highly scalable
- Available but not highly available
- Flexible but not highly flexible

OLTP Through the Looking Glass, and What We Found There

Stavros Harizopoulos
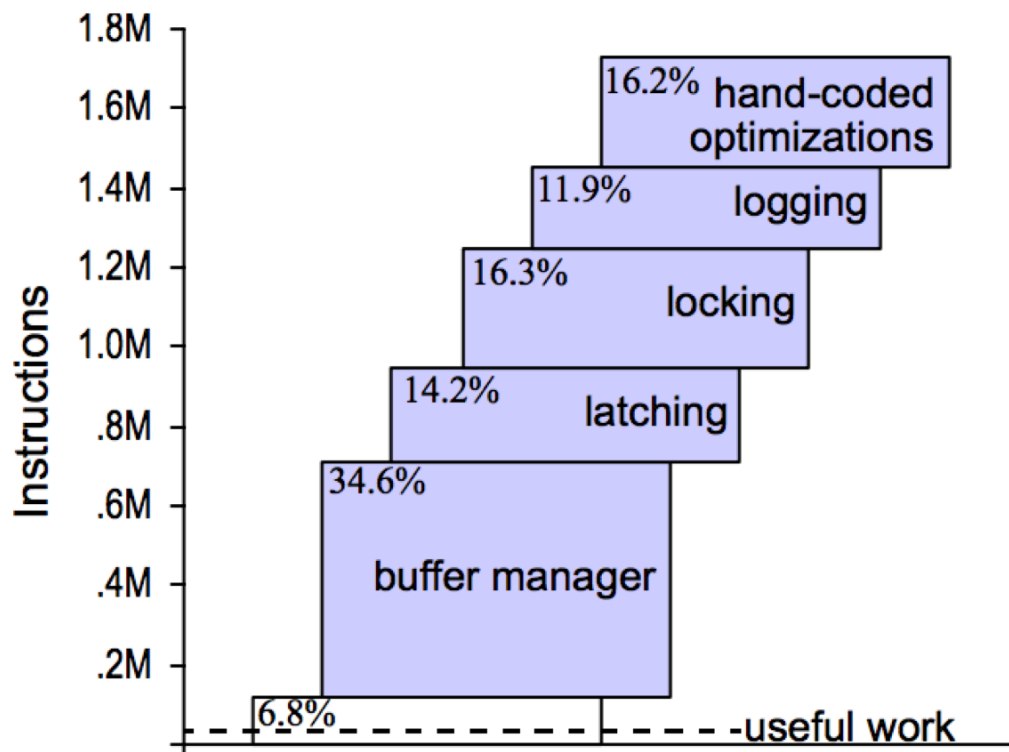HP Labs
Palo Alto, CA
stavros@hp.com

Daniel J. Abadi
Yale University
New Haven, CT
dna@cs.yale.edu

Samuel Madden   Michael Stonebraker
Massachusetts Institute of Technology
Cambridge, MA
{madden, stonebraker}@csail.mit.edu

Hekaton: SQL Server's Memory-Optimized OLTP Engine

Cristian Diaconu, Craig Freedman, Erik Ismert, Per-Åke Larson,
Pravin Mittal, Ryan Stonecipher, Nitin Verma, Mike Zwilling
Microsoft

{cdiaconu, craigfr, eriki, palarson, pravinm, ryanston, nitinver, mikezw}@microsoft.com

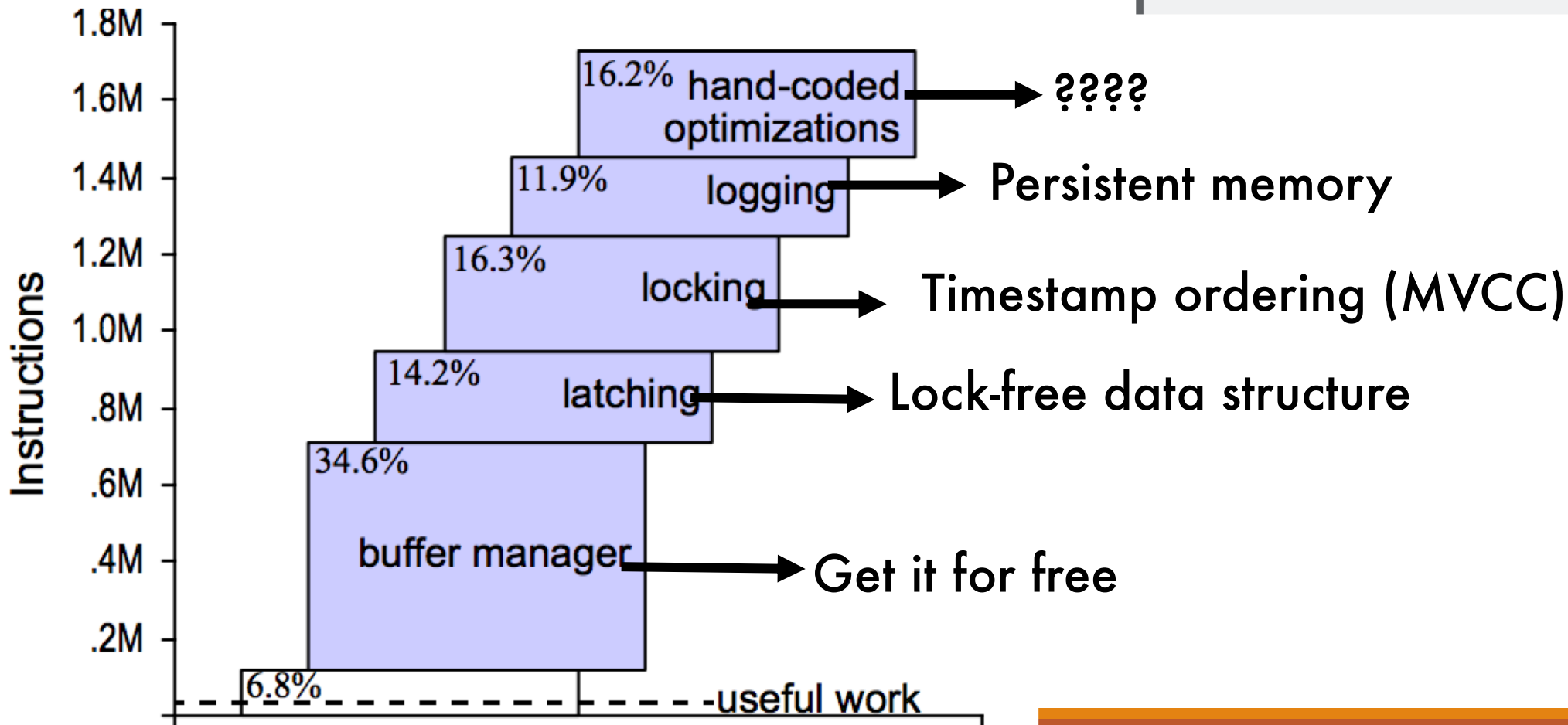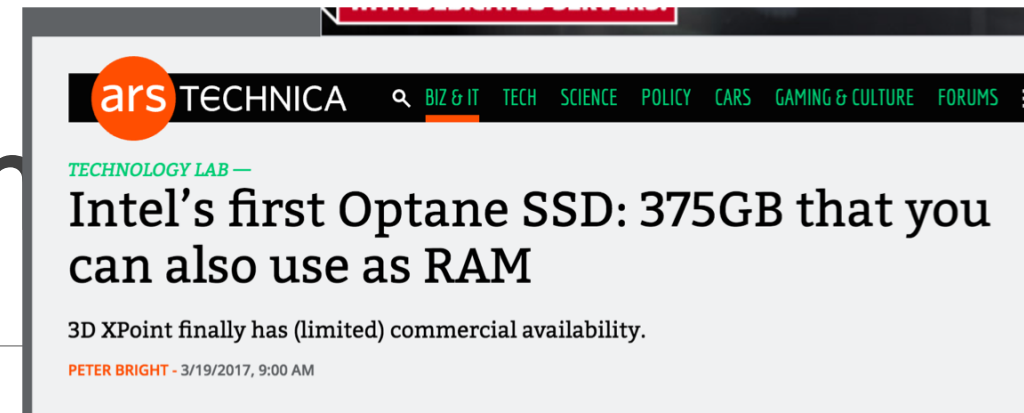## Hekaton: SQL Server's Memory-optimized OLTP Engine (2013)

Inside the **Hekaton**: **SQL Server** 2014's database engine ...

Register - Apr 17, 2014

It's 1996 and Mission:Impossible has just arrived on the cinema screens. RAM is $10 per megabyte and falling. Against this backdrop, Microsoft ...

Can we really achieve th

16.2% hand-coded optimizations → **????**

11.9% logging → **Persistent memory**

16.3% locking → **Timestamp ordering (MVCC)**

14.2% latching → **Lock-free data structure**

34.6% buffer manager → **Get it for free**

6.8% useful work

Summary

Why OldSQL?
Why NoSQL?
Why NewSQL?