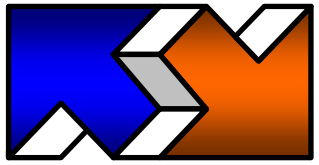
MIDDLEWARE SYSTEMS
RESEARCH GROUP
MSRG.ORG



Solving Big Data Challenges for Enterprise Application Performance Management

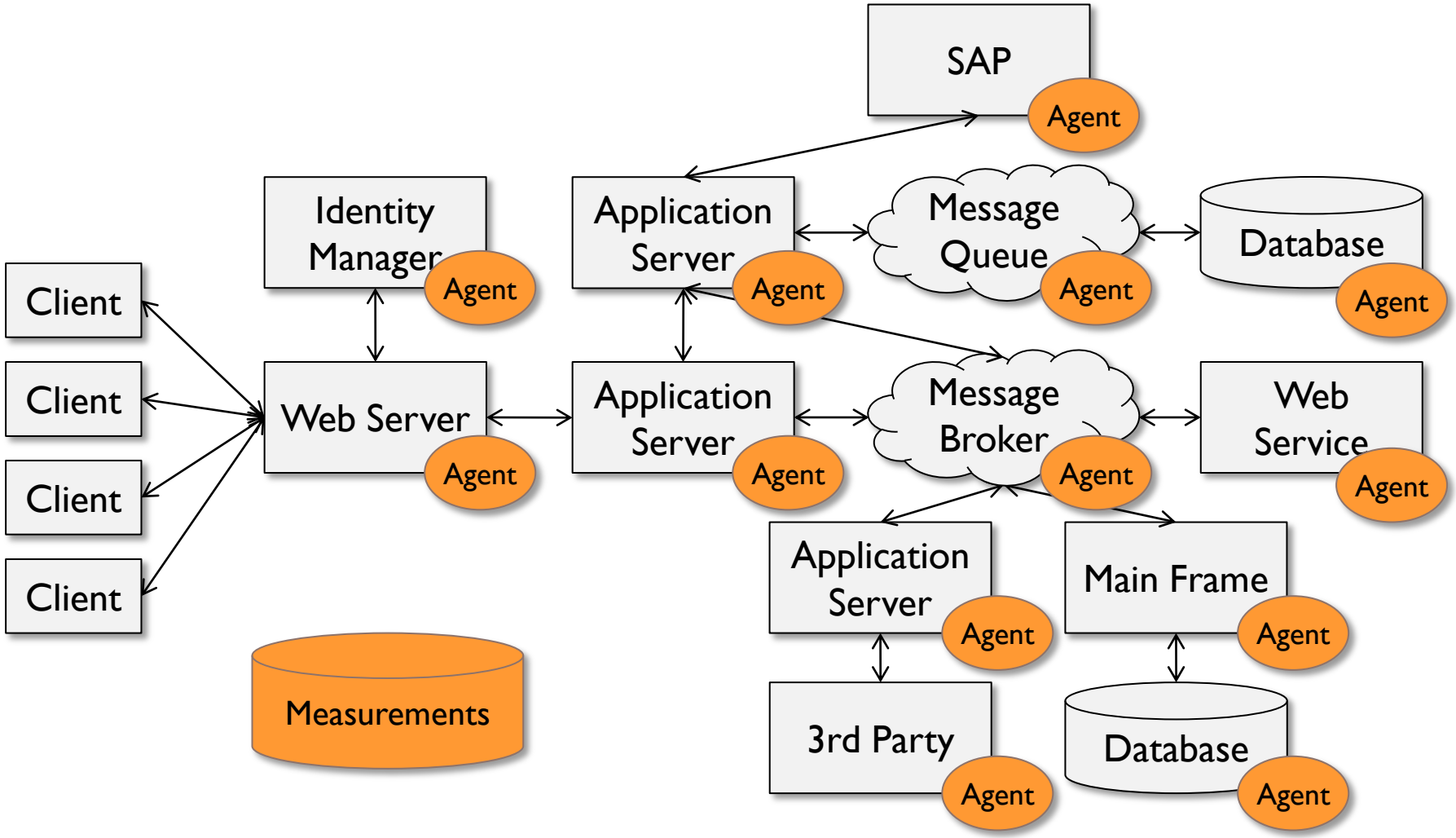
Tilmann Rabl, Mohammad Sadoghi, Hans-Arno Jacobsen
University of Toronto
Sergio Gomez-Villamor
Universitat Politecnica de Catalunya
Victor Muntes-Mulero*, Serge Mankowskii
CA Labs (Europe*)

Agenda

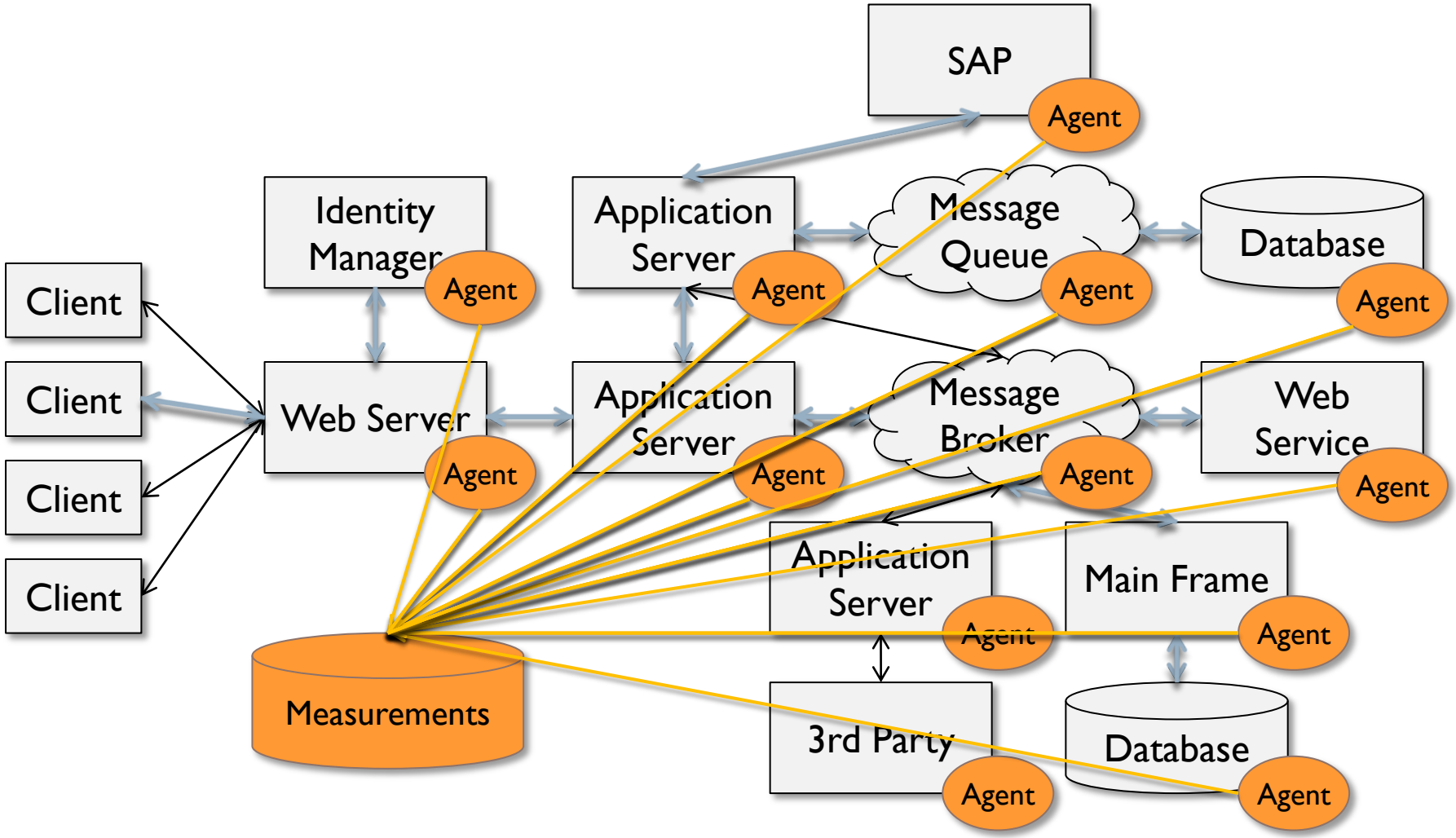
- Application Performance Management
- APM Benchmark
- Benchmarked Systems
- Test Results

Motivation

Enterprise System Architecture



Application Performance Management



APM in Numbers

- Nodes in an enterprise system
  - 100 – 10K
- Metrics per node
  - Up to 50K
  - Avg 10K
- Reporting period
  - 10 sec
- Data size
  - 100B / event
- Raw data
  - 100MB / sec
  - 355GB / h
  - 2.8 PB / y
- Event rate at storage
  - **> 1M / sec**

Metric Name	Value	Min	Max	Timestamp	Duration
HostA/AgentX/ServletB/AverageResponseTime	4	1	6	1332988833	15

APM Benchmark

- Based on Yahoo! Cloud Serving Benchmark (YCSB)
  - CRUD operations
- Single table
  - 25 byte key
  - 5 values (10 byte each)
  - 75 byte / record
- Five workloads

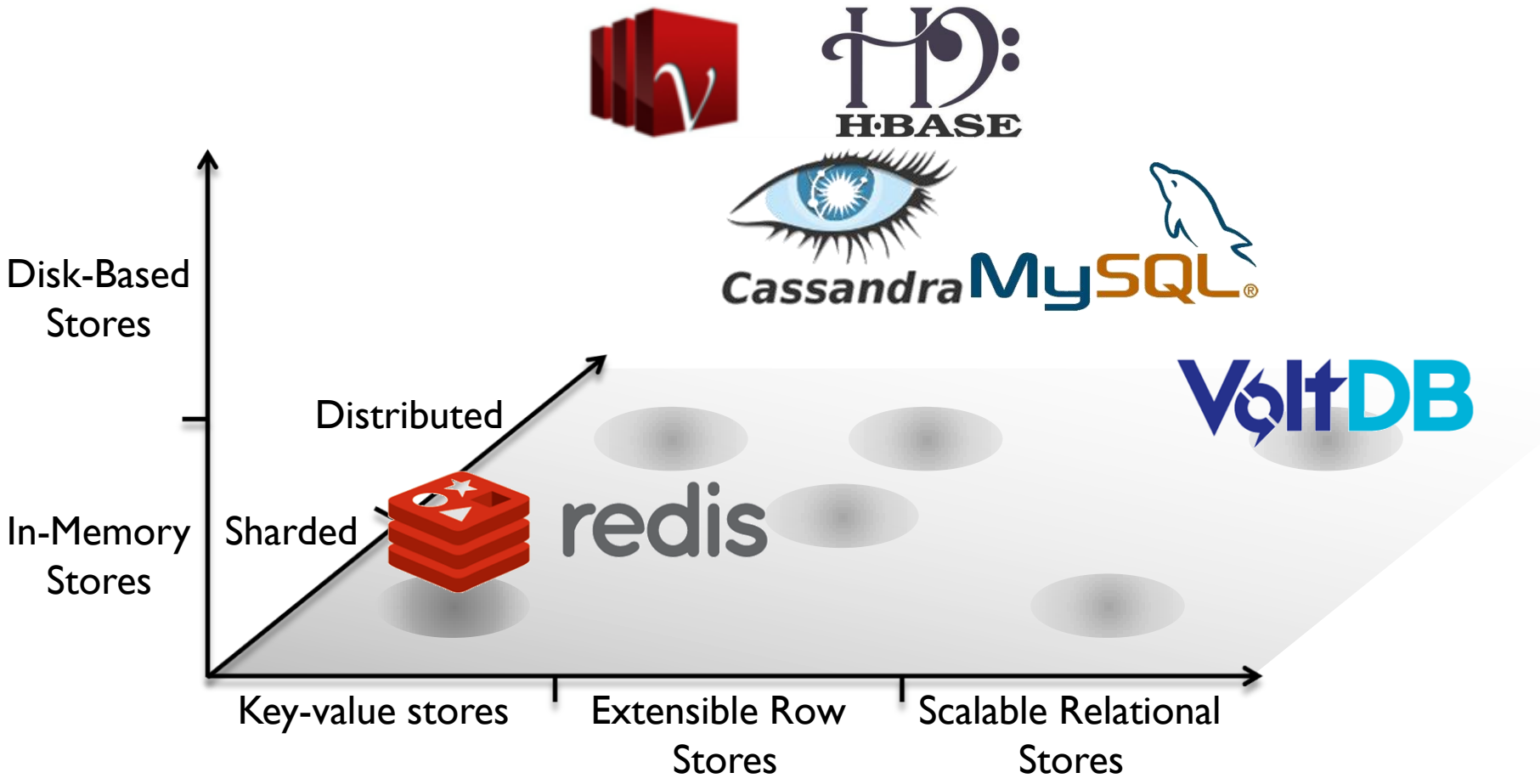
Workload	% Read	% Scans	% Inserts
R	95	0	5
RW	50	0	50
W	1	0	99
RS	47	47	6
RSW	25	25	50

  - 50 rows scan length

Benchmarked Systems

- 6 systems
  - Cassandra
  - HBase
  - Project Voldemort
  - Redis
  - VoltDB
  - MySQL
- Categories
  - Key-value stores
  - Extensible record stores
  - Scalable relational stores
  - Sharded systems
  - Main memory systems
  - Disk based systems
- Chosen by
  - Previous results, popularity, maturity, availability

Classification of Benchmarked Systems



Experimental Testbed

Two clusters

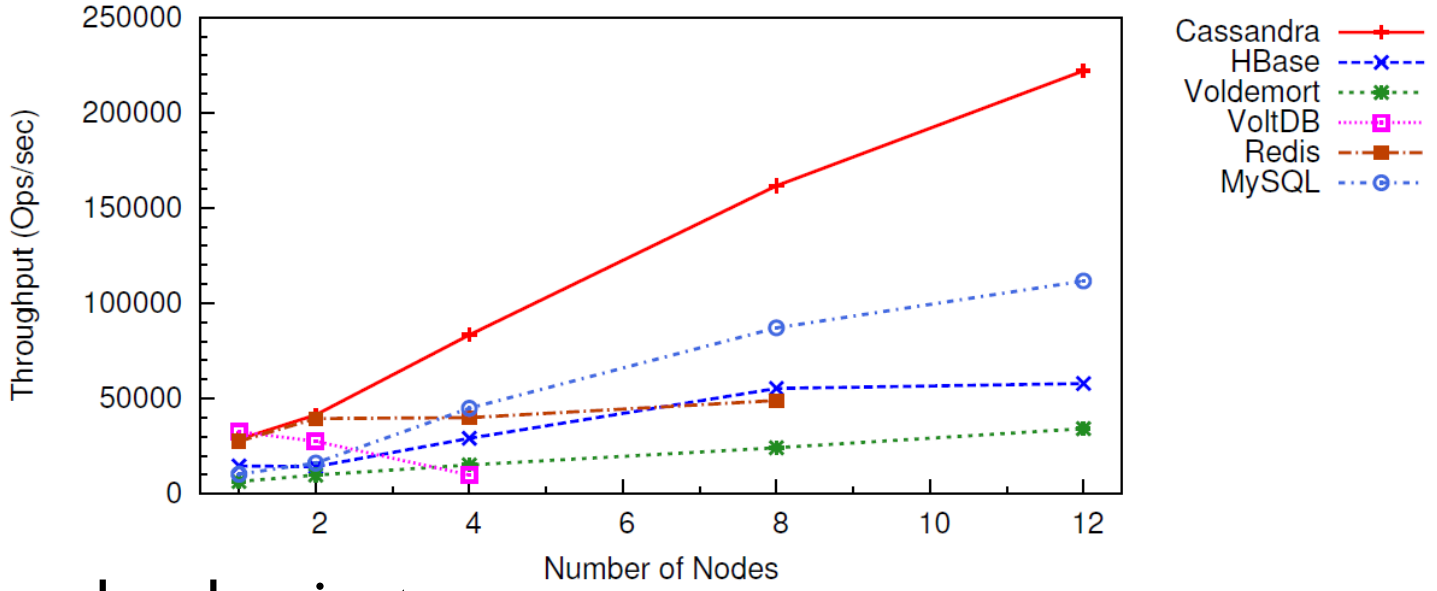
- Cluster M (memory-bound)
  - 16 nodes (plus master node)
  - 2x quad core CPU,16 GB RAM, 2x 74GB HDD (RAID 0)
  - 10 million records per node (~700 MB raw)
  - 128 connections per node (8 per core)
- Cluster D (disk-bound)
  - 24 nodes
  - 2x dual core CPU, 4 GB RAM, 74 GB HDD
  - 150 million records on 12 nodes (~10.5 GB raw)
  - 8 connections per node

Evaluation

- Minimum 3 runs per workload and system
- Fresh install for each run
- 10 minutes runtime
- Up to 5 clients for 12 nodes
  - To make sure YCSB is no bottleneck

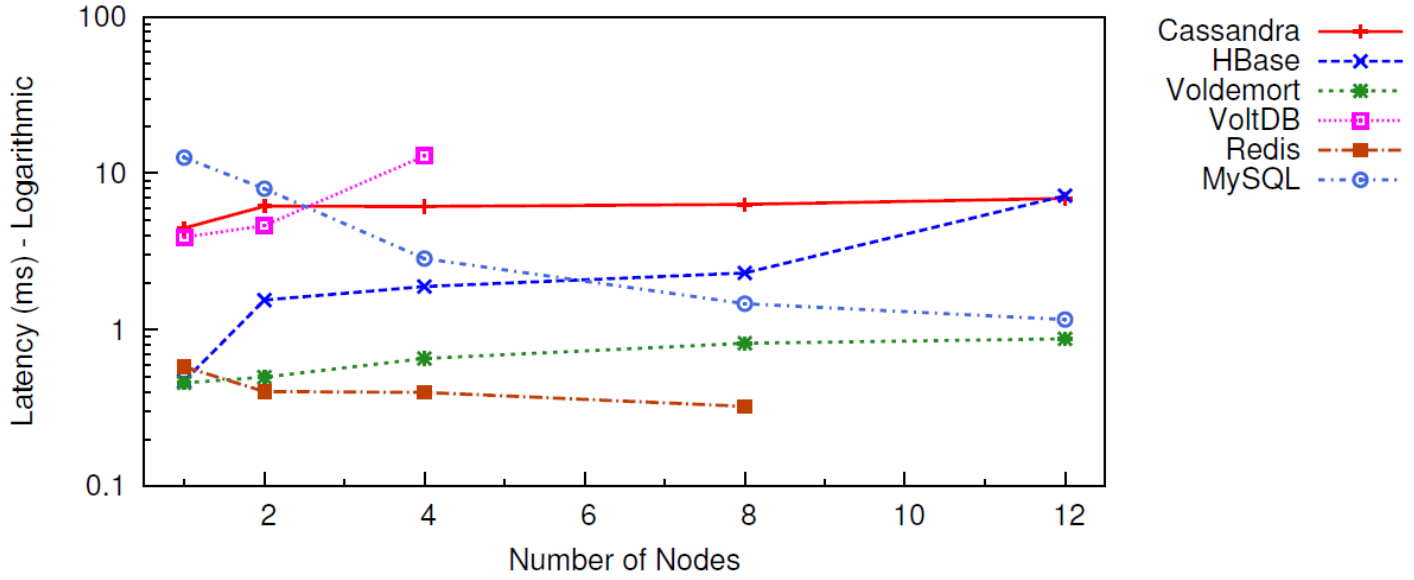
- Maximum achievable throughput

Workload W - Throughput



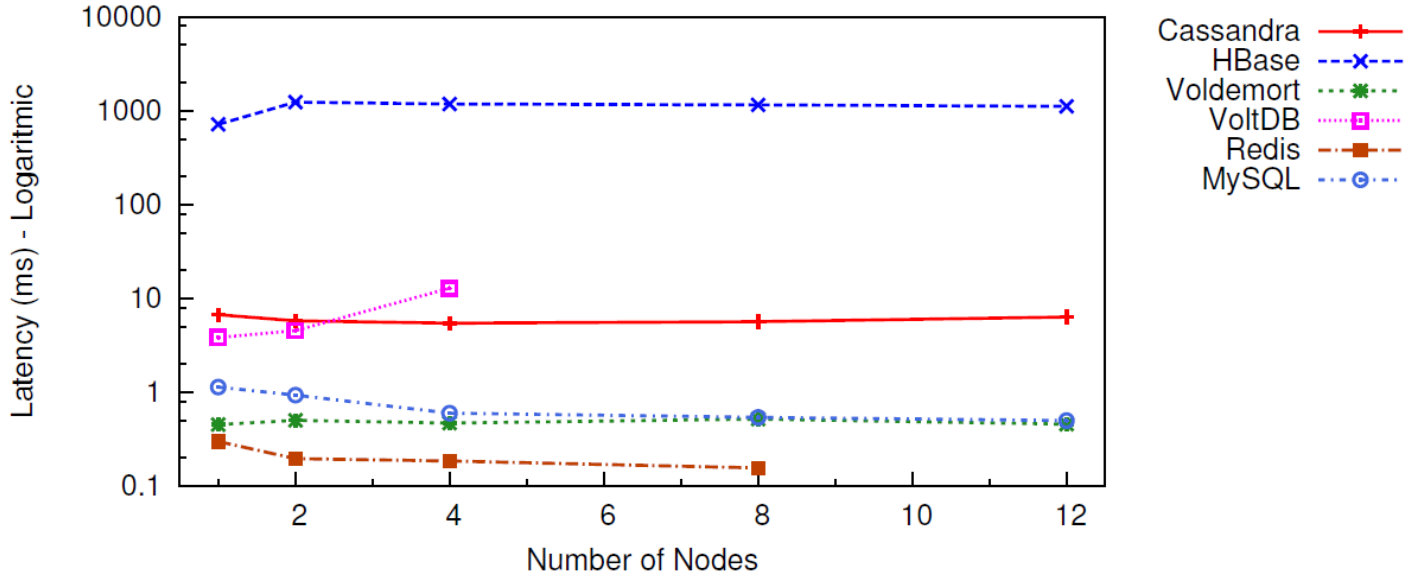
- Cassandra dominates
- Higher throughput for Cassandra and HBase
- Lower throughput for other systems
- Scalability not as good for all web stores
- VoltDB best single node throughput

Workload W – Latency Writes



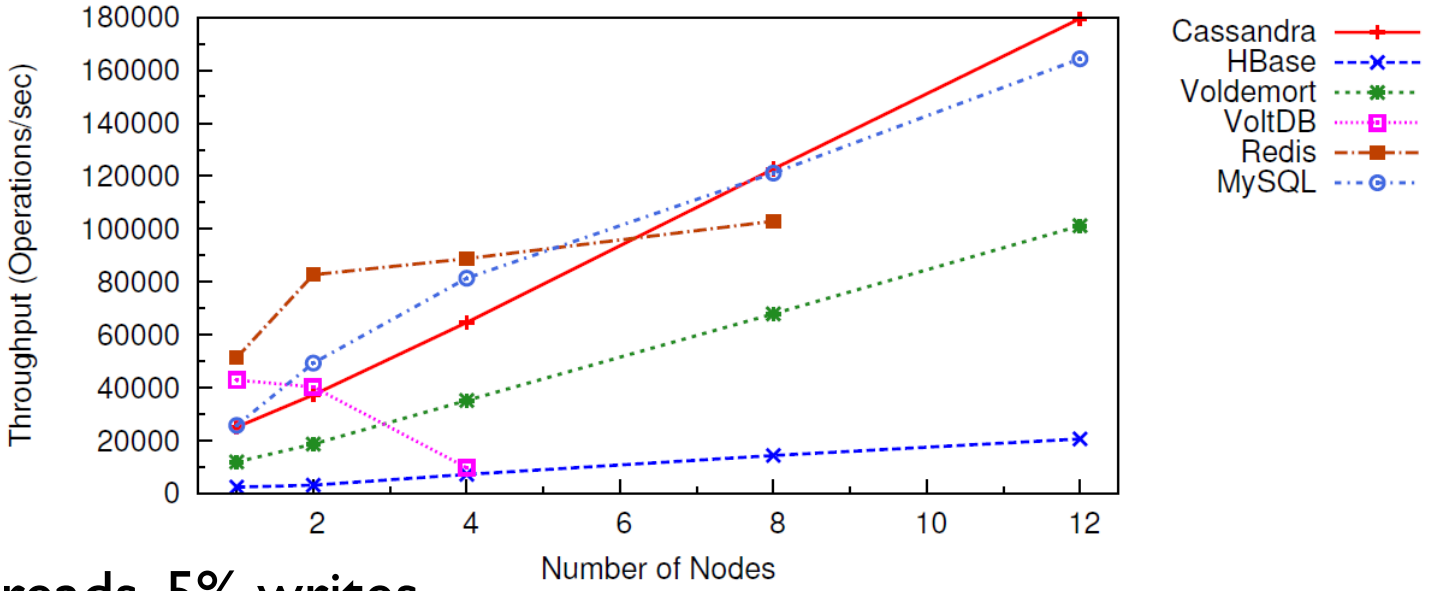
- Same latencies for
  - Cassandra, Voldemort, VoltDB, Redis, MySQL
- HBase latency increased

Workload W – Latency Reads



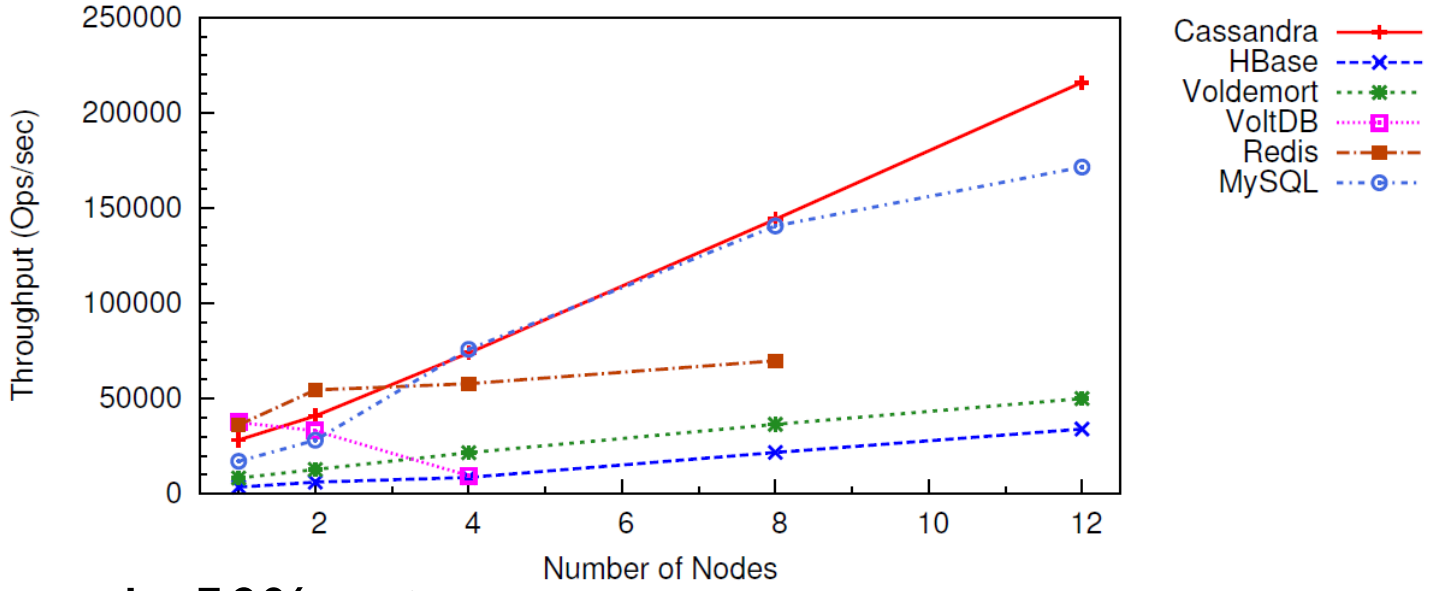
- Same latency as for R for
  - Cassandra, Voldemort, Redis, VoltDB, HBase
- HBase latency in second range

Workload R - Throughput



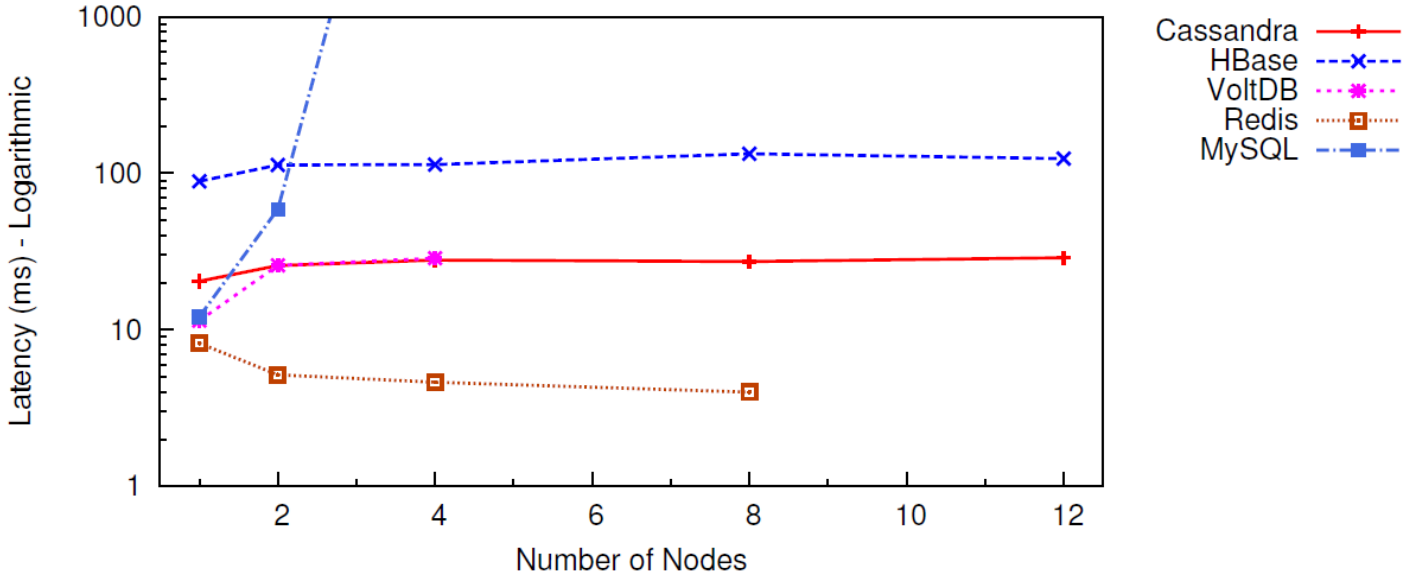
- 95% reads, 5% writes
- On a single node, main memory systems have best performance
- Linear scalability for web data stores
- Sublinear scalability for sharded systems
- Slow-down for VoltDB

Workload RW - Throughput



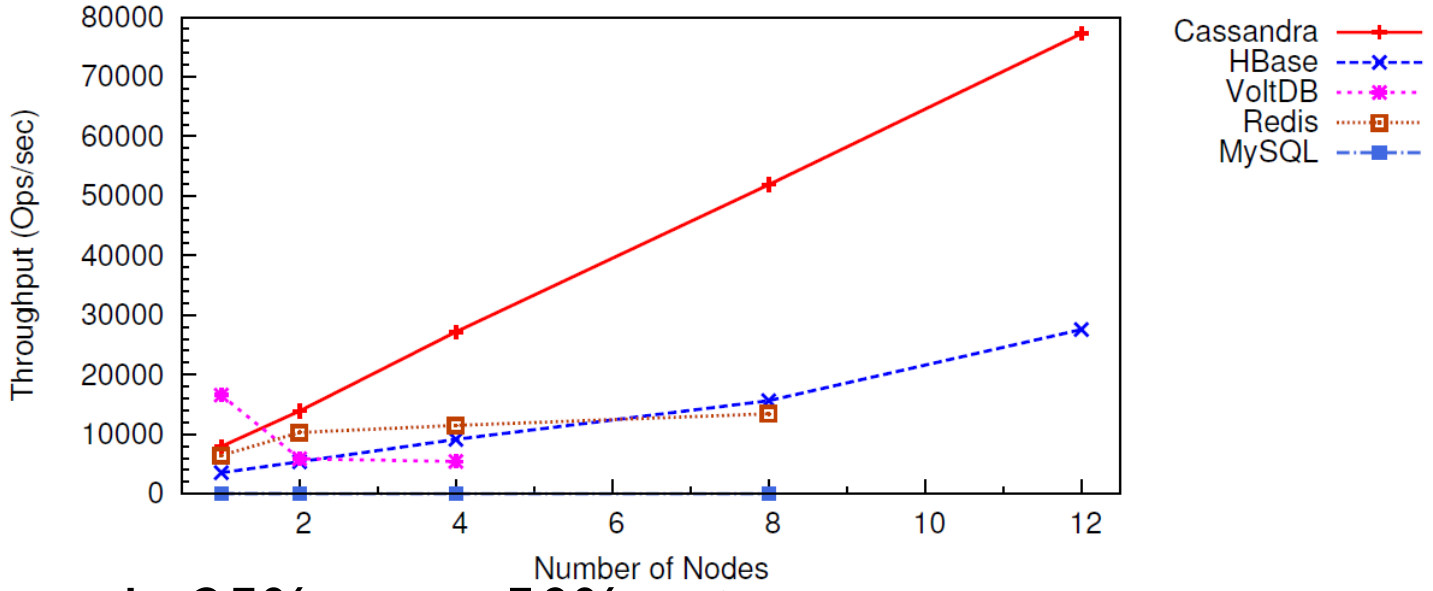
- 50% reads, 50% writes
- VoltDB highest single node throughput
- HBase and Cassandra throughput increase
- MySQL and Voldemort throughput reduction

Workload RS – Latency Scans



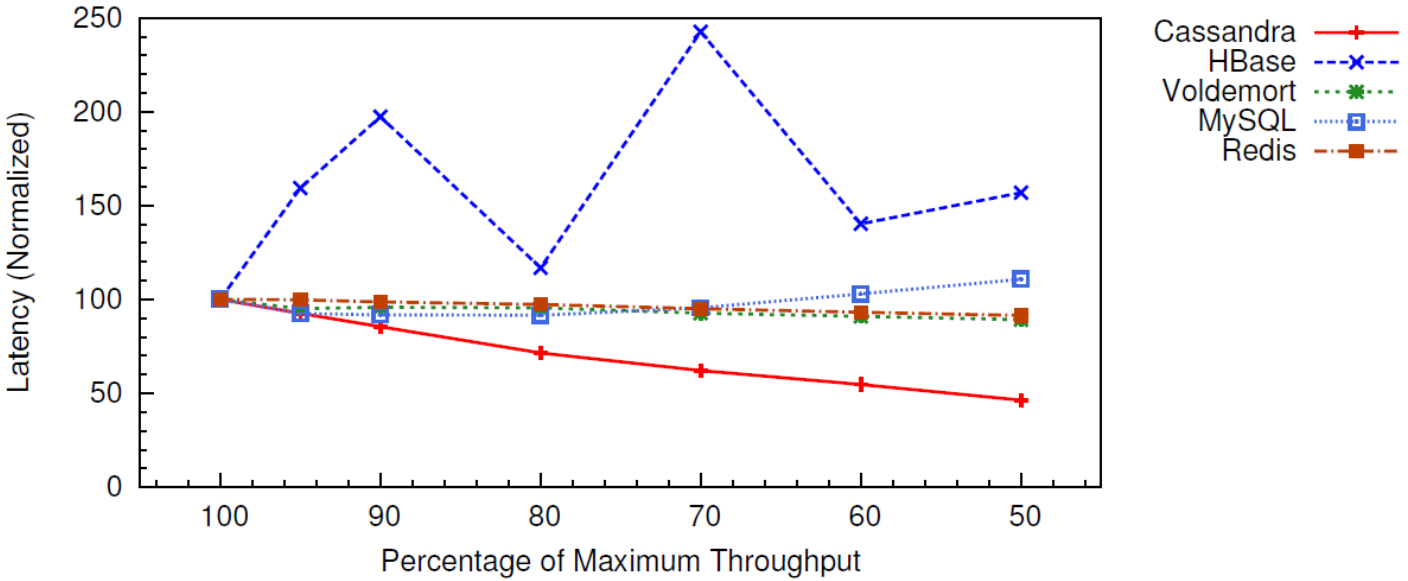
- HBase latency equal to reads in Workload R
- MySQL latency very high due to full table scans
- Similar but increased latency for
  - Cassandra, Redis, VoltDB

Workload RWS - Throughput



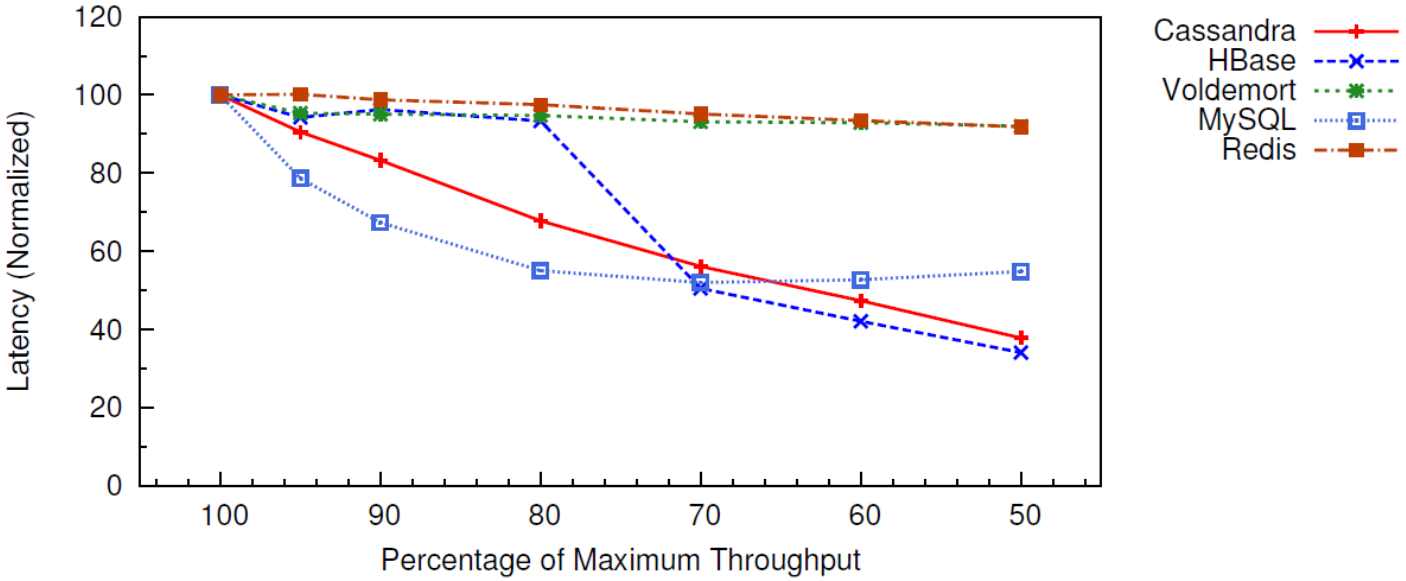
- 25% reads, 25% scans, 50% writes
- Cassandra, HBase, Redis, VoltDB gain performance
- MySQL performance 20 – 4 ops / sec

Bounded Throughput – Write Latency



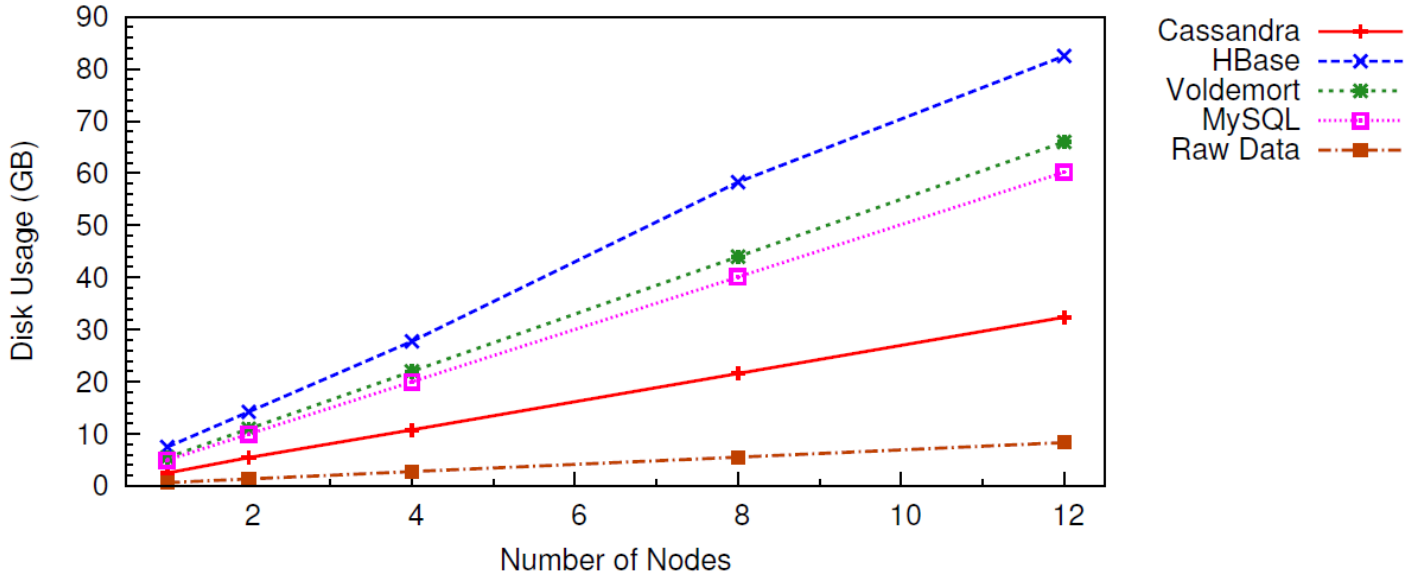
- Workload R, normalized
- Maximum throughput in previous tests 100%
- Steadily decreasing for most systems
- HBase not as stable (but below 0.1 ms)

Bounded Throughput – Read Latency



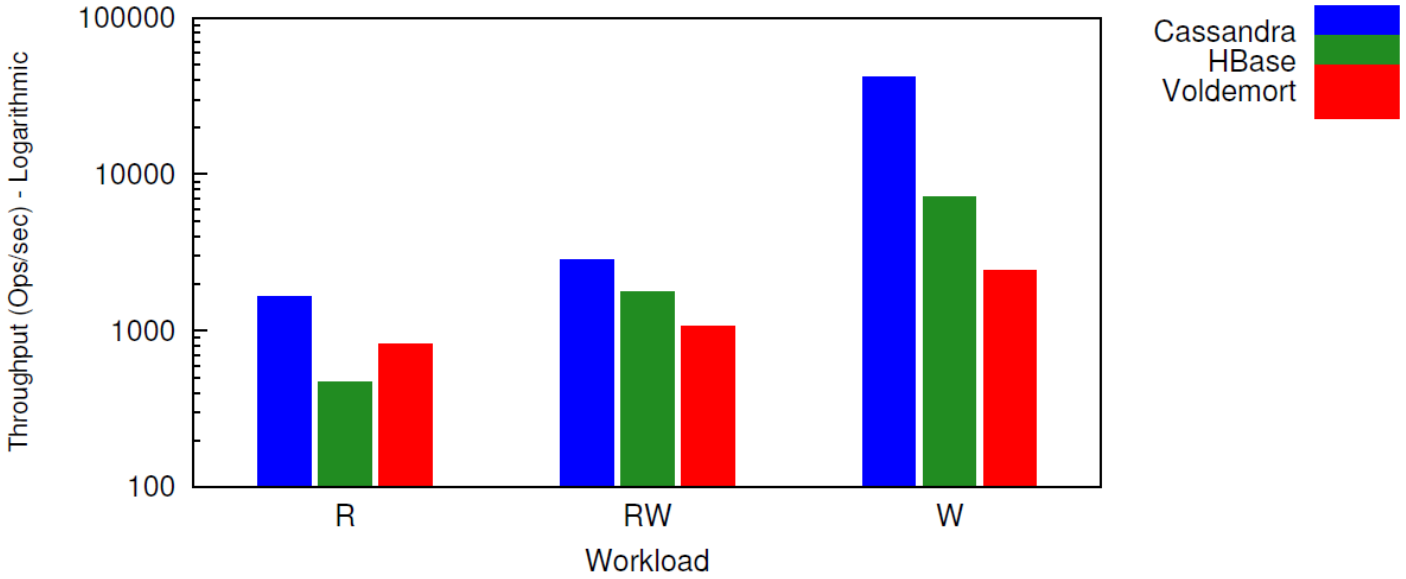
- Redis and Voldemort slightly decrease
- HBase two states
- Cassandra decreases linearly
- MySQL decreases and then stays constant

Disk Usage



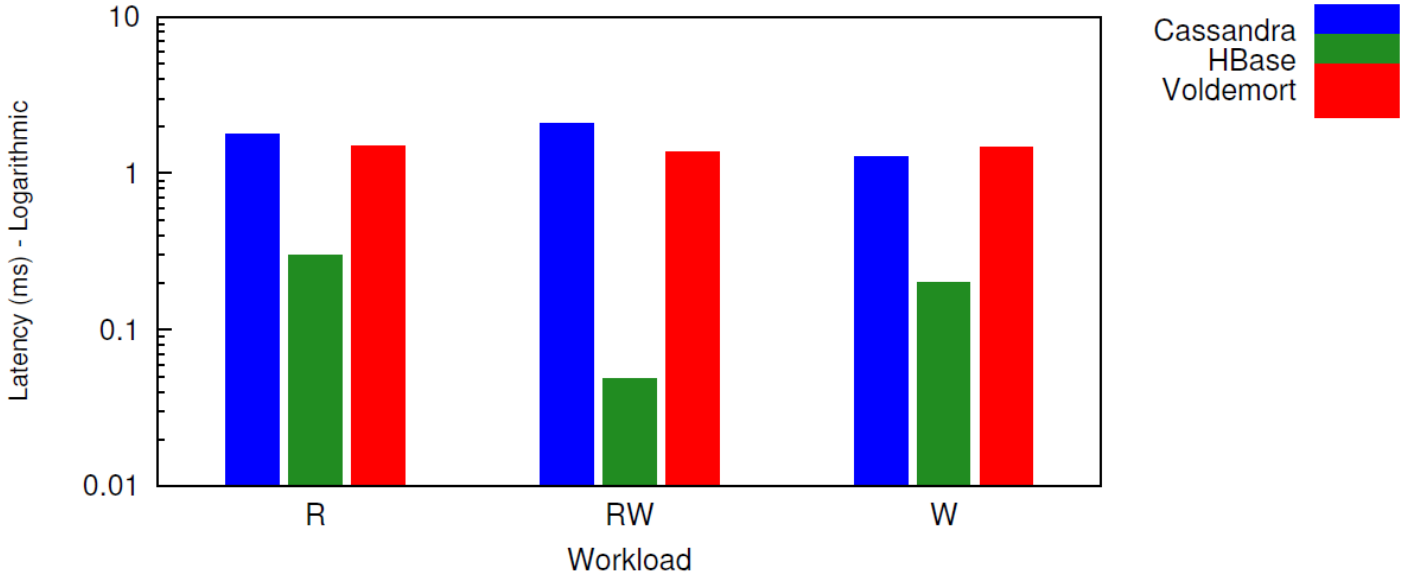
- Raw data: 75 byte per record, 0.7GB/10M
- Cassandra: 2.5GB / 10M
- HBase: 7.5GB / 10M
- No compression

Cluster D Results – Throughput



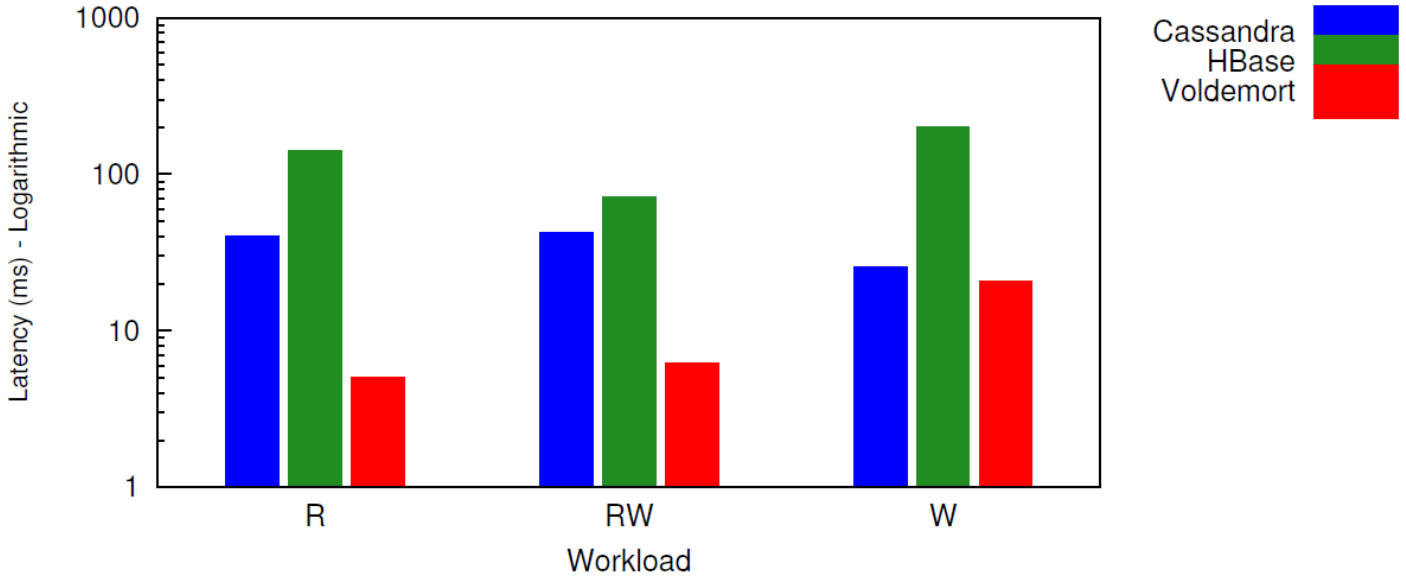
- Disk bound, 150M records on 8 nodes
- More writes – more throughput
  - Cassandra: 26x
  - Hbase: 15x
  - Voldemort 3x

Cluster D Results – Latency Writes

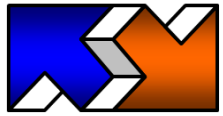


- Low latency for writes for all systems
- Relatively stable latency for writes
- Lowest for HBase
- Equal for Voldemort and Cassandra

Cluster D Results – Latency Reads



- Cassandra latency decreases with more writes
- Voldemort latency low 5-20ms
- HBase latency high (up to 200 ms)
- Cassandra in between



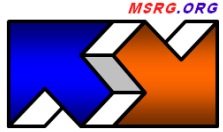
Lessons Learned

- YCSB
  - Client to host ratio 1:3 better 1:2
- Cassandra
  - Optimal tokens for data distribution necessary
- HBase
  - Difficult setup, special JVM settings necessary
- Redis
  - Jedis distribution uneven
- Voldemort
  - Higher number of connections might lead to better results
- MySQL
  - Eager scan evaluation
- VoltDB
  - Synchronous communication slow on multiple nodes

Conclusions I

- Cassandra
  - Winner in terms of scalability and throughput
- HBase
  - Low write latency, high read latency, low throughput
- Sharded MySQL
  - Scales well, latency decreases with higher scales
- Voldemort
  - Read and write latency stable at low level
- Redis
  - Standalone has high throughput, sharded version does not scale well
- VoltDB
  - High single node throughput, does not scale for synchronous access

Conclusions II



- Cassandra's performance close to APM requirements
  - Additional improvements needed for reliably sustaining APM workload
- Future work
  - Benchmark impact of replication, compression
  - Monitor the benchmark runs using APM tool

Thanks!

MIDDLEWARE SYSTEMS RESEARCH GROUP
MSRG.ORG

- Questions?

- Contact: Tilmann Rabl
  tilmann.rabl@utoronto.ca