NetApp 7th Annual University Day
NetApp HQ, Sunnyvale, CA, USA
17 February 2014

Modern Database Systems: NewSQL, NoSQL and Modernized Classic Systems

C. Mohan

IBM Fellow and Former IBM India Chief Scientist
IBM Almaden Research Center, San Jose, CA 95120, USA
cmohan@us.ibm.com http://bit.ly/CMohan

Talk Version of Tutorial Presented First at
39th International Conference on Very Large Data Bases (VLDB)
Riva del Garda, Italy, 29 August 2013
Links to Revised Tutorial Slides and Bibliography at http://bit.ly/CMnMDS

Background + Warnings

- My personal opinions, NOT necessarily IBM's positions/opinions
- Lots of hype on NoSQL (lesser on NewSQL) in industry, academia & VC world – this **Kool-Aid** has intoxicated many ☺
- Been a details oriented, hard core DB person for > 3 decades – I go beyond hype, am not easily swayed by the latest fashion!
- Not a relational bigot – most of my results on systems topics have applied to all sorts of persistent, distributed information systems
- Have dissected many IBM **and** non-IBM systems' internals (including pre- and post-relational ones)
- Don't claim to have all the answers or make definitive assertions
- **Aim**: not exhaustive survey of modern systems or details of too many specific systems – broad brush analysis of good (sensible) and bad (nonsenSQL) of NoSQL/NewSQL, warn of pitfalls – *time to pause and introspect.* Give a few systems' interesting details.
- **NOT against NoSQL** but unhappy with choices on internals, design justifications ("anything goes"), and not learning enough from the past

Some Quotes

- **Jack Clark, The Register, 30 August 2013**: "*The tech world is turning back toward SQL, bringing to a close a possibly misspent half-decade in which startups courted developers with promises of infinite scalability and the finest imitation-Google tools available, and companies found themselves exposed to unstable data and poor guarantees.*"
- **Google Spanner paper, October 2012:** "*We believe it is better to have application programmers deal with performance problems due to overuse of transactions as bottlenecks arise, rather than always coding around the lack of transactions.*"
- **Sean Doherty in Wired, September 2013:** "*But don't become unnecessarily distracted by the shiny, new-fangled, NoSQL red buttons just yet. … Relational databases may not be hot or sexy but for your important data there is no substitute.*"
- **Gartner** estimates RDBMS market at $26B with about 9% annual growth
  Market Research Media Ltd expects NoSQL market to be at $3.5B by 2018

Introduction

- Goal: Broad Survey of Modern Database Systems (MDS)
- Drivers Behind Emergence of MDS, Benchmarks/Performance Studies
- Classes of MDS
  - Evolution of Classical DBMSs (DB2, SQL Server, Oracle, Informix, PostgreSQL)
  - Brand New Systems (NewSQL, NoSQL)
  - Hybrid Systems (Hadoop + SQL)
- Overviews of Specific Systems
  - DB2 BLU, IDAA, MS Hekaton, Oracle Exadata, Informix Warehouse Accelerator, Postgres Plus Advanced Server
  - SAP Hana, Google F1, NuoDB, VoltDB, HyPer, Calvin
  - MongoDB, DB2/Informix NoSQL, Oracle NoSQL, Aerospike, Neo4j, Facebook Tao
- Deployments or Use Case Scenarios: Facebook

Ack: Figures and tables sourced from documentation/papers/web pages!

Links to Slides and Bibliography at http://bit.ly/CMnMDS

Drivers of MDS Features

- Hardware Trends: Multicore, Multi-socket, InfiniBand, DRAMs, SSDs, FPGAs, GPUs, Storage Class Memories (e.g., PCM)
- Big Data (esp. Semi Structured or Unstructured): Compression, Column Handling, Scalability (Sharding), Avoiding Data Loading, Linkage to Analytics Tools, …
- Wider Adoption of Database Technology: Ease of Use, Appliances
- Monetary Constraints: Open Source, Commodity Processors/Disks, Local Disks, Economic Factors in General
- Skills Shortage: DBAs, Admins in General, SQL Knowledge, …
- Social Media: Relationship Graphs, Ultra Low Response Times, Data Heterogeneity/Volume, Mobile Users
- Internet of Things (IOT): Smarter Planet, Sensors, Low Power, …
- Cloud Platforms: Elasticity, Multi-tenancy, Security, Transborder Data Flows

Hardware Trends

- DRAM costs 17X SSD! Dell $20/GB DRAM (16 GB DIMM $315)
- SSD 10X HDD cost ($0.10/GB, 200 IOPS)

  SSD: Samsung 840, 250 GB, $180

  4 drives/chassis: 1 TB, 384K read IOPS, 248K random write IOPS, $720, 0.3 Watts
- DBs under 100 GB: DRAM

  100 GB – 2 TB: SSD

  >1 PB: HDD
- Open source Aerospike Certification Tool for SSDs for real-time use cases

Conclusions from Workload Studies

- EPFL Studies
  - OLTP workloads in traditional DSs don't leverage aggressive micro-architectural features, wasting time in memory stalls, resulting in low instructions per cycle (IPC)
  - L1 instruction misses cause significant stall times with index probes dominating instruction & data stalls
- H-Store: significant time spent in lock/latch, buffer pool fixes/unfixes
- Sort Vs Hash join & NUMA effects revisited in multicore/multi-socket systems

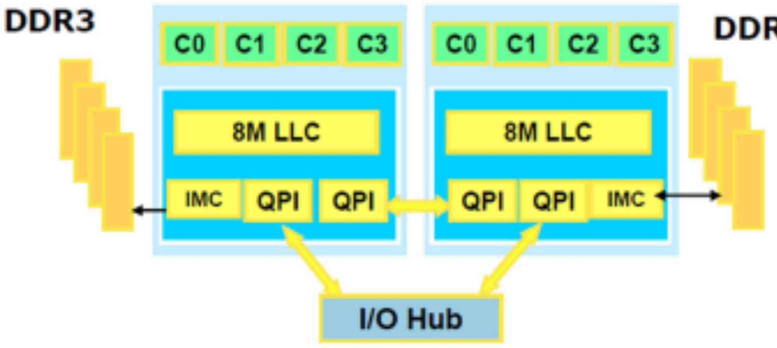
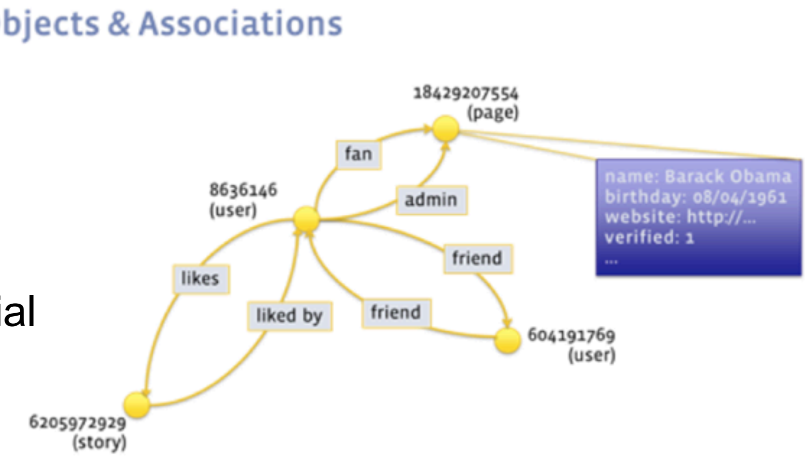


Figure 1: Block diagram of a typical machine. Cores communicate either through a common cache, an interconnect across socket or main memory.

QPI – QuickPath Interconnect
IMC – Integrated Memory Controller

Benchmarks

- **TPC** benchmarks (e.g., TPC-C, TPC-E, TPC-H)
- **LinkBench**: Facebook's Benchmark Based on Social Graph
- **BigBench**: Benchmark for Big Data Analytics
  - Teradata, University of Toronto, InfoSizing , Oracle
  - Product Retailer Scenario
  - Addresses Variety, Velocity & Volume of Big Data
  - Structured (TPC-DS), Semistructured (user clicks), Unstructured (product reviews)
  - Queries Cover Analytics Proposed by McKinsey: Marketing (Cross Selling, Customer Micro-Segementation, Sentiment Analysis, Enhancing Multichannel Consumer Experience), Merchandising (Assortment Optimization, Pricing Optimization), Operations (Performance Transparency, Return Analysis), Supply Chain (Inventory Management), New Business Models (Price Comparison)
  - Feasibility Illustrated via Teradata Aster Database's SQL-MR



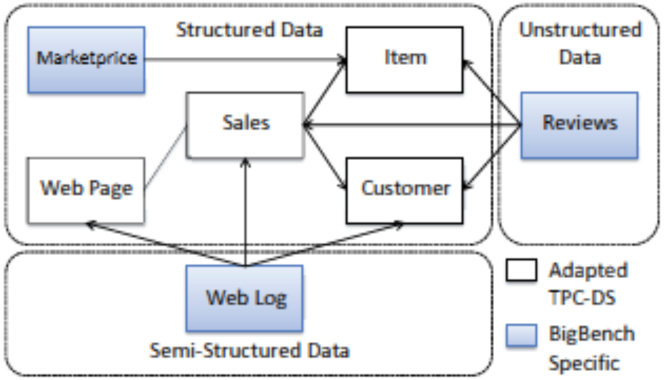
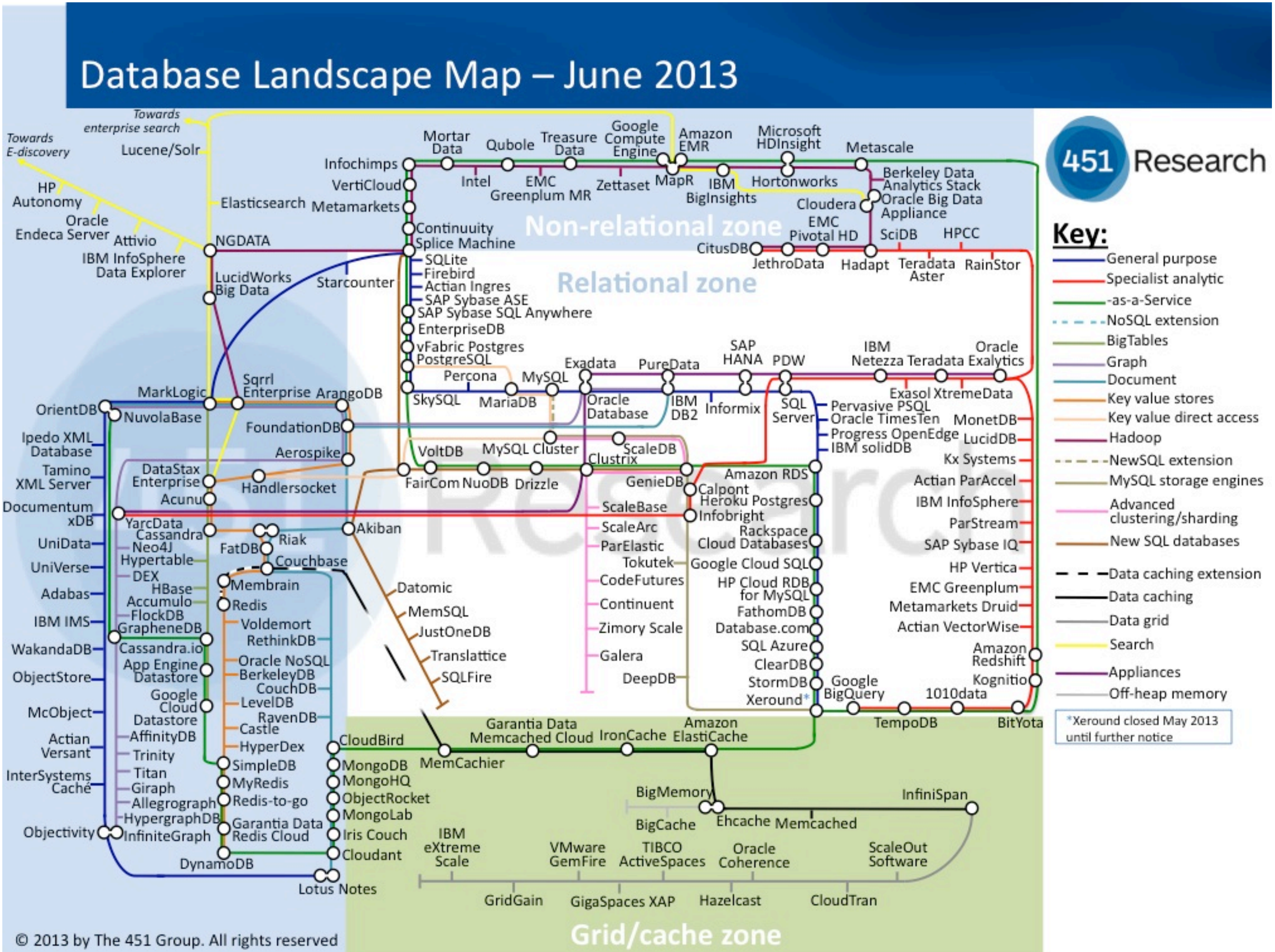
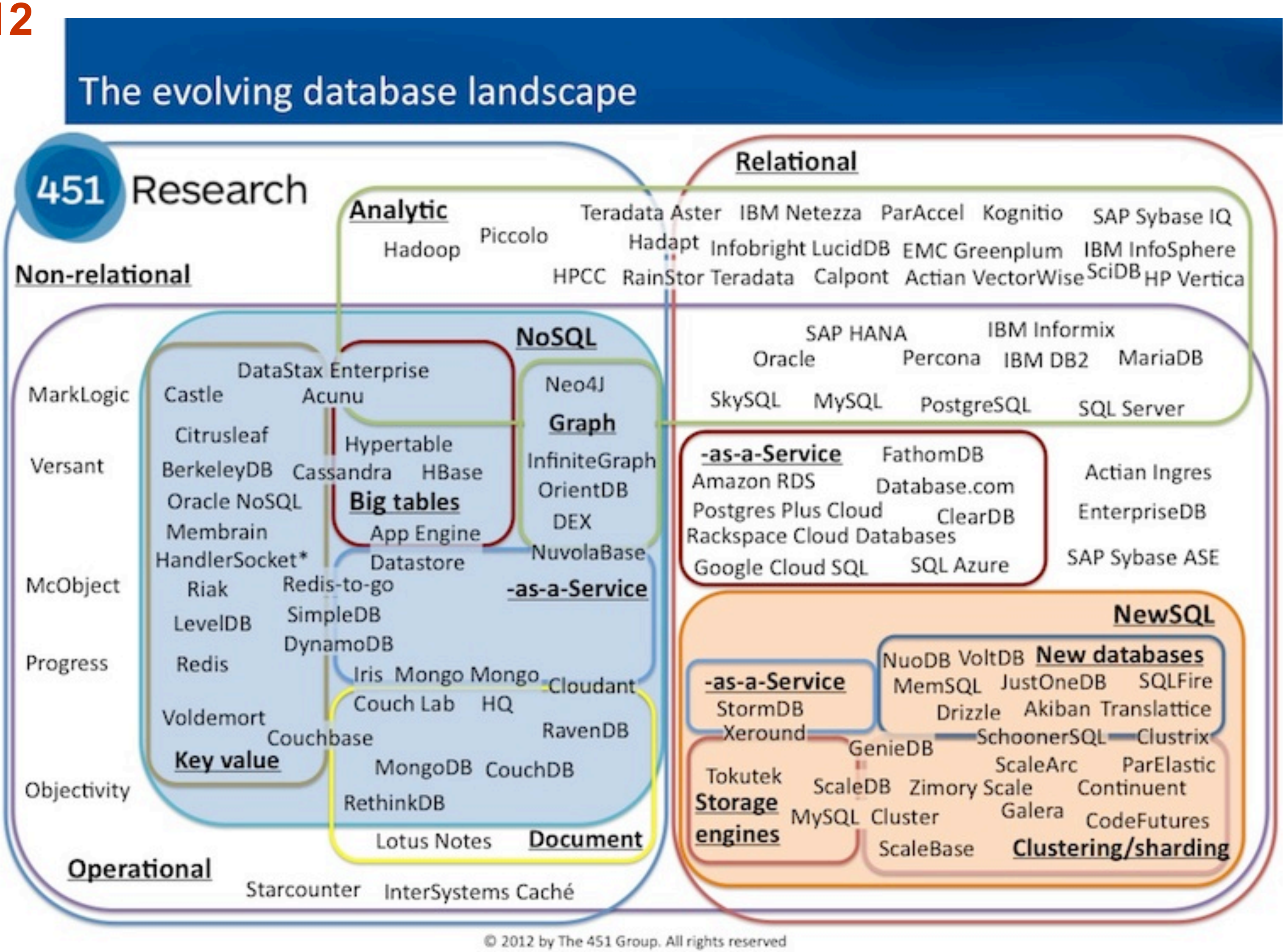


Figure 1: Big Data Benchmark Data Model



https://blogs.the451group.com/information_management/files/2013/06/451db_map_06.13.jpg

Database *Product* Landscape – Matthew Aslett 11-2012

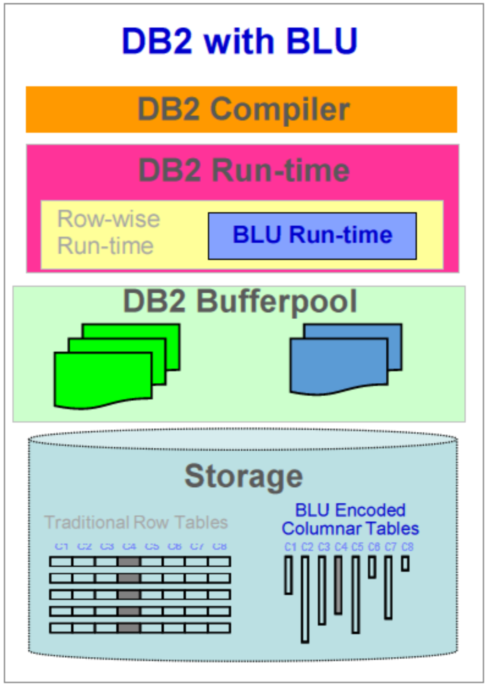


http://blogs.the451group.com/information_management/files/2012/11/DBlandscape.jpg

Evolution of Classical DBMSs

DB2 with BLU Acceleration

- New innovative technology for analytic queries
  - Columnar storage, single copy of the data
  - New run-time engine with vector (aka SIMD) processing
  - Deep multi-core optimizations and cache-aware memory management
  - "Active compression" - unique encoding for storage reduction and run-time processing without decompression
- "Revolution by Evolution"
  - Built directly into the DB2 kernel
  - BLU tables can coexist with row tables, in same schema, tablespaces, bufferpools
  - Query any combination of BLU or row data
  - Memory-optimized (not "in-memory")
- Value : Order-of-magnitude benefits in ...
  - Performance
  - Storage savings
  - Simplicity!
- Generally available since June 2013



Column-store (in) DB2 LUW v10.5

V. Raman et al. *DB2 with BLU Acceleration: So Much More than Just a Column Store*. Proc. VLDB Endowment (PVLDB), 2013.

IBM DB2 Analytics Accelerator (IDAA)

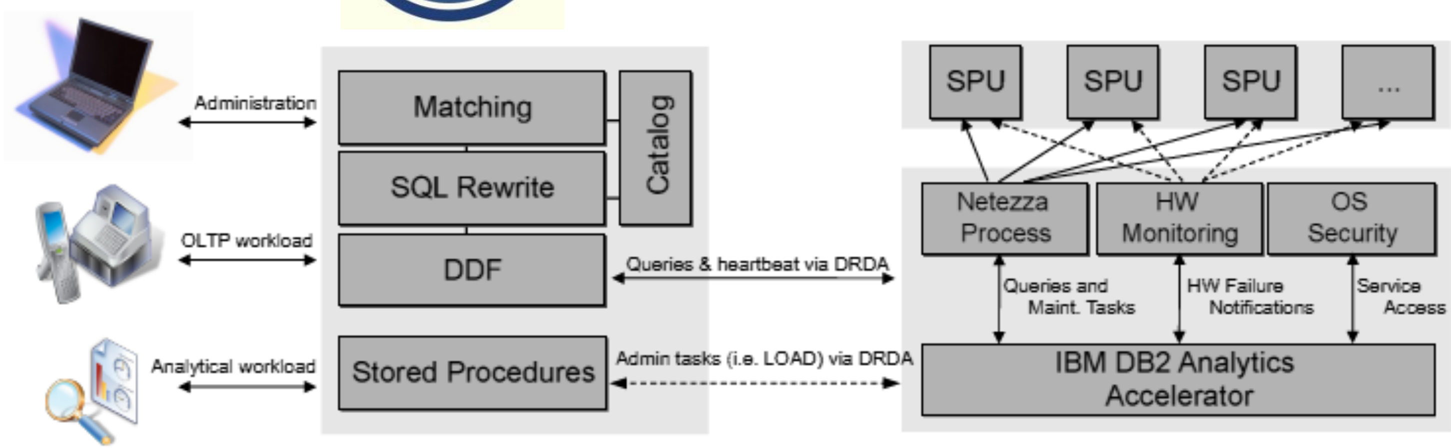


Figure 2: IDAA Architecture

- Data from DB2 z replicated continuously to Netezza
- Offload complex queries from DB2 z to Netezza for more efficient/cheaper execution
- Queries execute with snapshot semantics since Netezza supports MVCC
- Replaced IBM Smart Analytics Optimizer (ISAO) where Blink technology was used
- Evolution of Blink is what is now in DB2 BLU

Martin, D., Koeth, O., Ivanova, I., Kern, J. *Near Real-Time Analytics with IBM DB2 Analytics Accelerator*, Proc. International Conference on Extending Database Technology (EDBT), Genoa, Italy, March 2013.

Hekaton: SQL Server's In-Memory OLTP Engine

- Alternate data manager integrated with SQL Server
- All Hekaton data in memory with lock/latch-free optimistic, multiversion concurrency control
- Declare table to be memory optimized
- T-SQL stored procs accessing only Hekaton tables compiled into machine code
- Hash and range indexes
- Optionally non-durable tables
- Index updates not logged; rebuilt during recovery from checkpoint and logs.

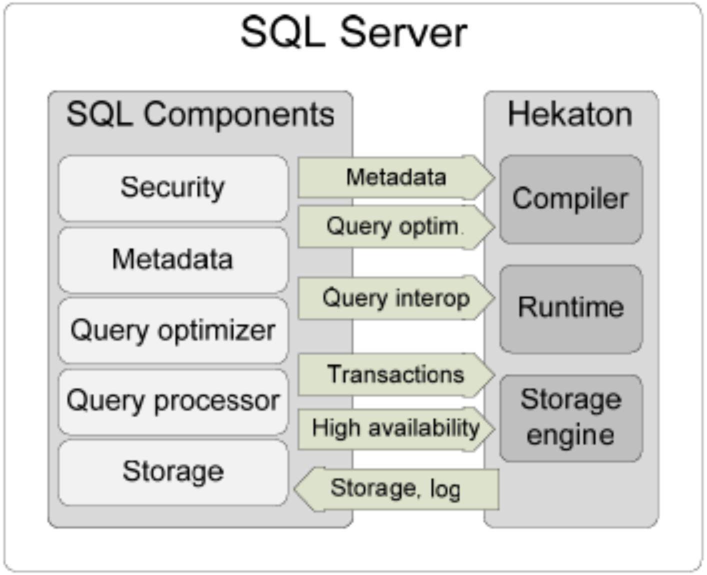
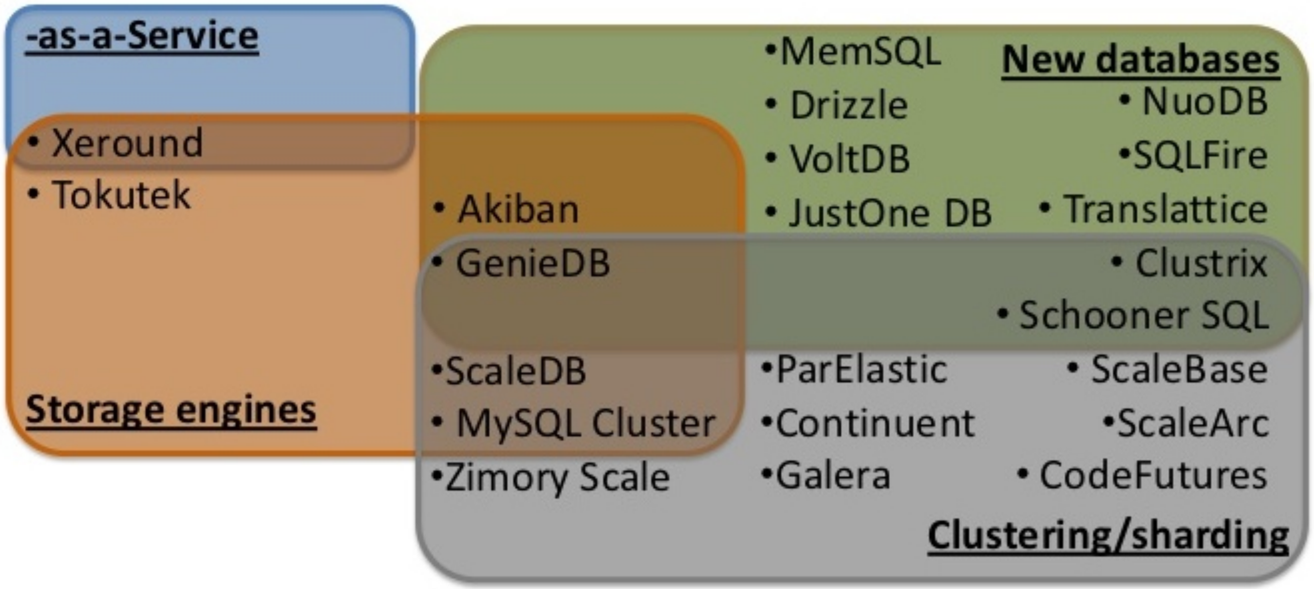


Figure 1: Hekaton's main components and integration into SQL Server.

C. Diaconu, C. Freedman, E. Ismert, P. Larson, P. Mittal, R. Stonecipher, N. Verma, M. Zwilling: *Hekaton: SQL Server's Memory-Optimized OLTP Engine*. Proc. ACM SIGMOD International Conference on Management of Data, New York, USA, June 2013.

NewSQL Systems

NewSQL Ecosystem - Matthew Aslett 2012



SAP Hana (OLTP+OLAP)

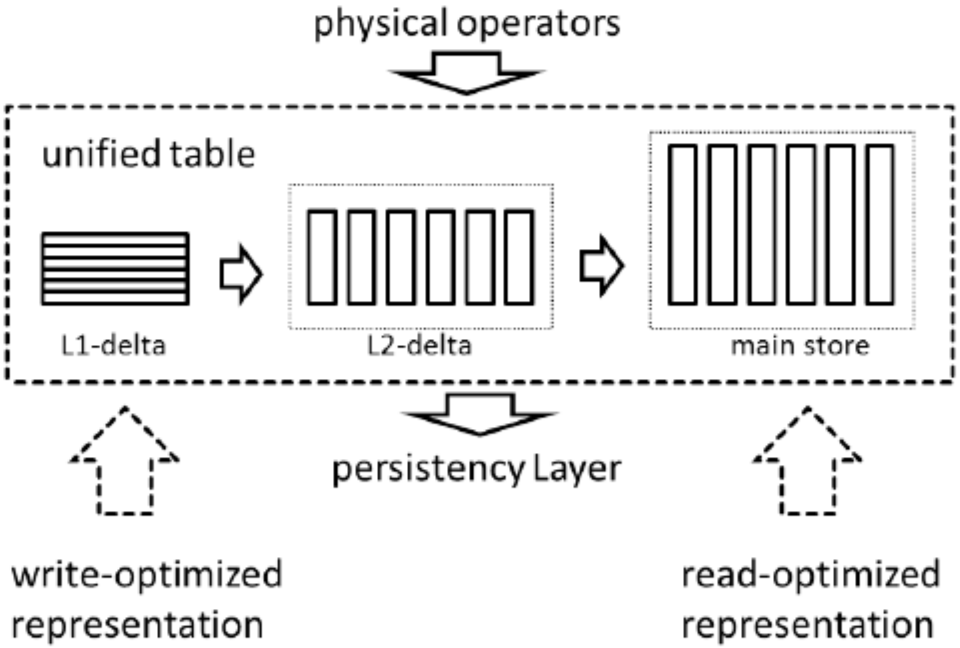


Figure 4: Overview of the unified table concept

L2 delta dictionary encoded, no compression for L1
Main store maximally compressed with many schemes

SAP Hana

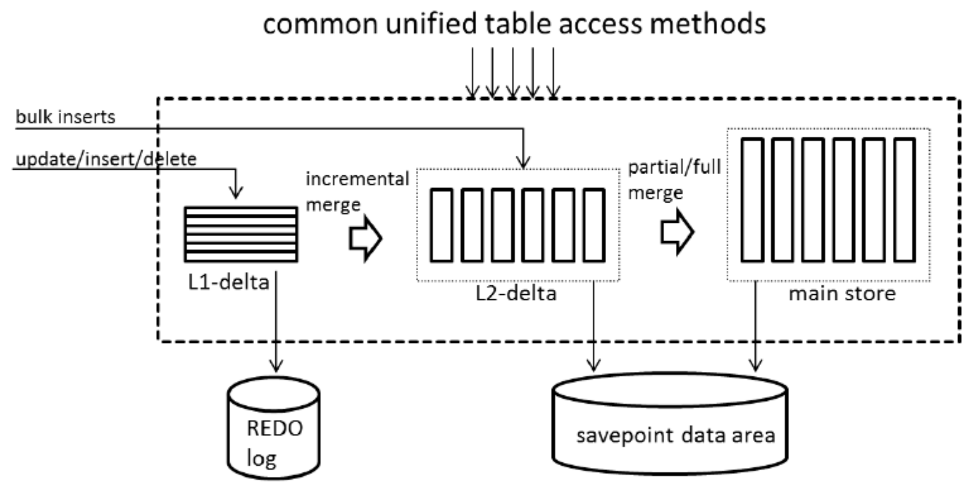


Figure 5: Overview of the persistency mechanisms of the unified table

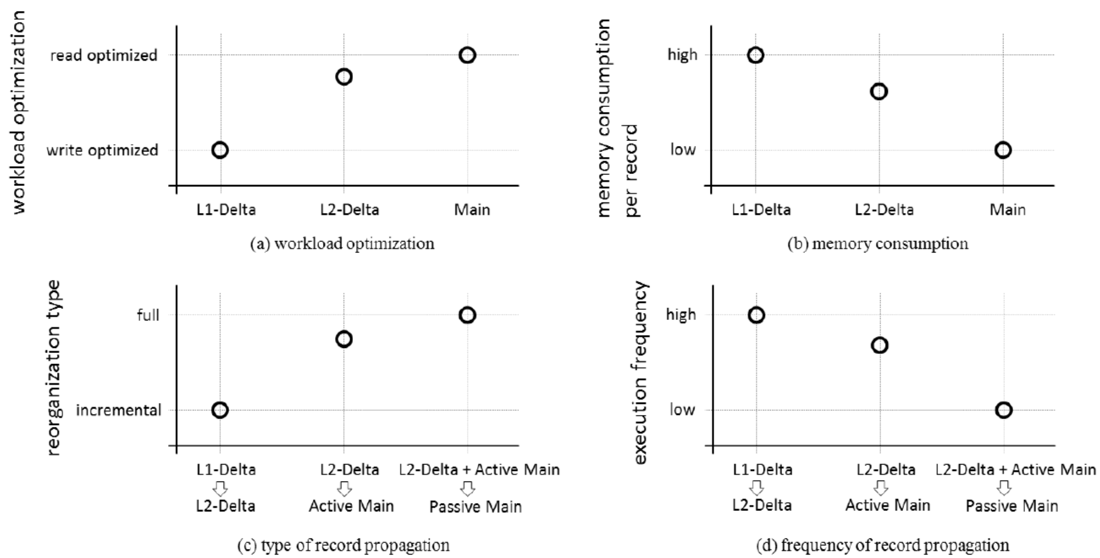


Figure 11: Characteristics of the SAP HANA database record life cycle

Google Spanner

- Globally distributed semi-relational system, SQL-based query lang
- Transactional, relies on TrueTime and Paxos

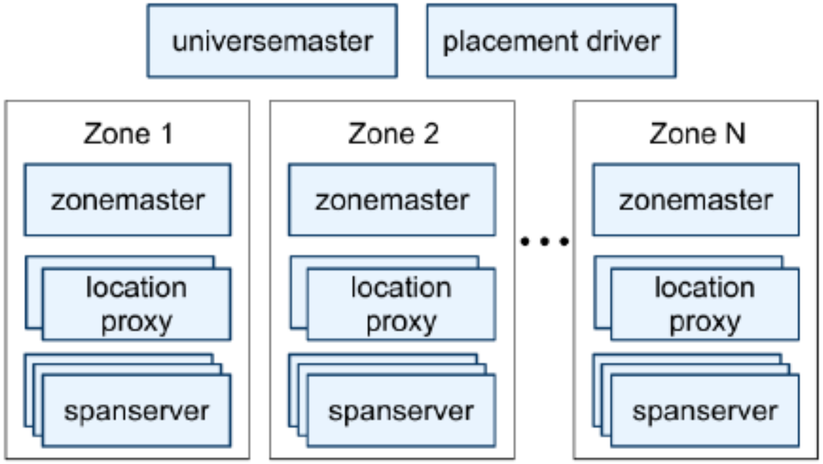


Figure 1: Spanner server organization.

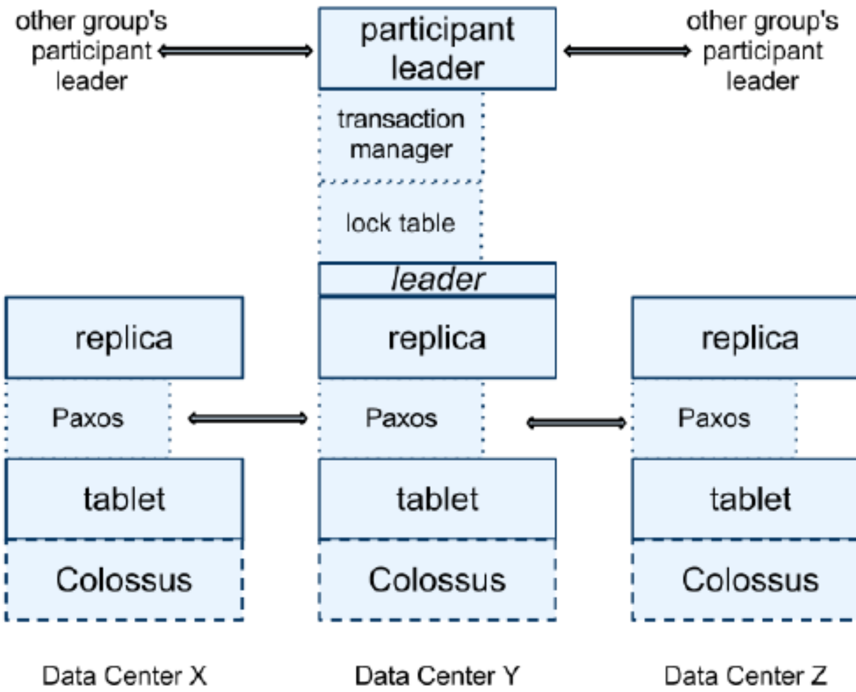


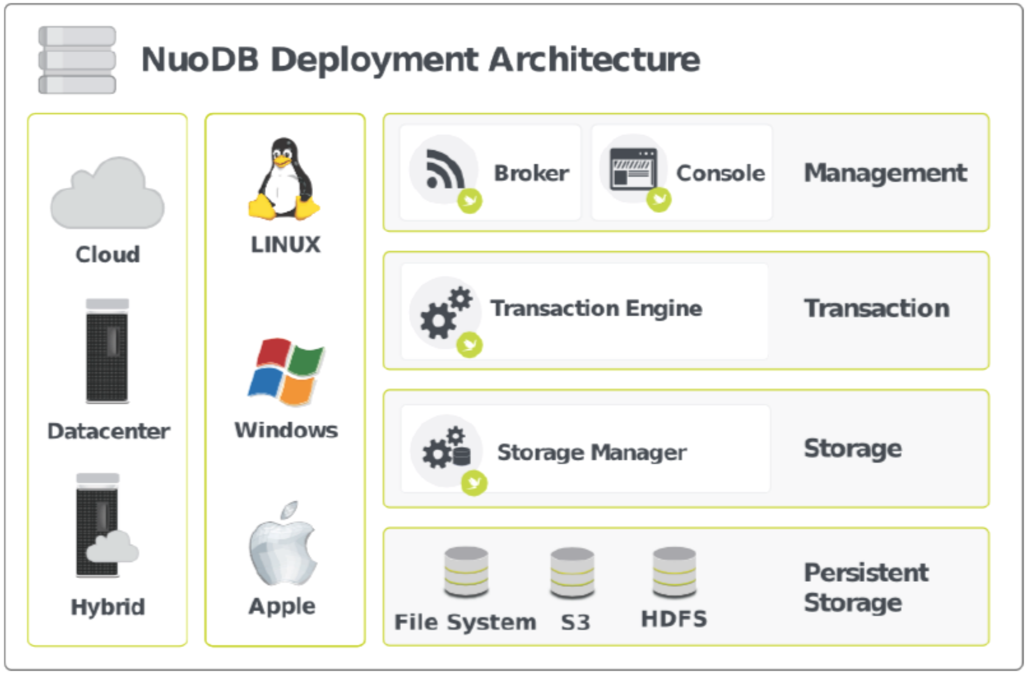
Figure 2: Spanserver software stack.

Google F1

- Developed for use with AdWords platform replacing MySQL
- Relies on Google Spanner
- Distributed SQL queries for OLAP-style queries, using snapshot transactions
- and two phase commit
- Optimistic concurrreny control
- Transaction consistent secondary indexes
- Protocol Buffer as column data type
- Clustered hierarchical tables
- Increased read, write and commit latencies
- 5 copy replication
- Row or sub-row level locking
- Many issues found with inefficiencies of MySQL ORM layer – replaced with F1 ORM
- Multiple client consumers of a single query's results

NuoDB

- Separates transaction management from storage management
  – Transaction Engines and Storage Managers
- TEs manage data via partial, on-demand replication
- Data resides where it is needed and used
- NuoDB Cloud on Amazon AWS
- Uses multi-version concurrency control (MVCC)

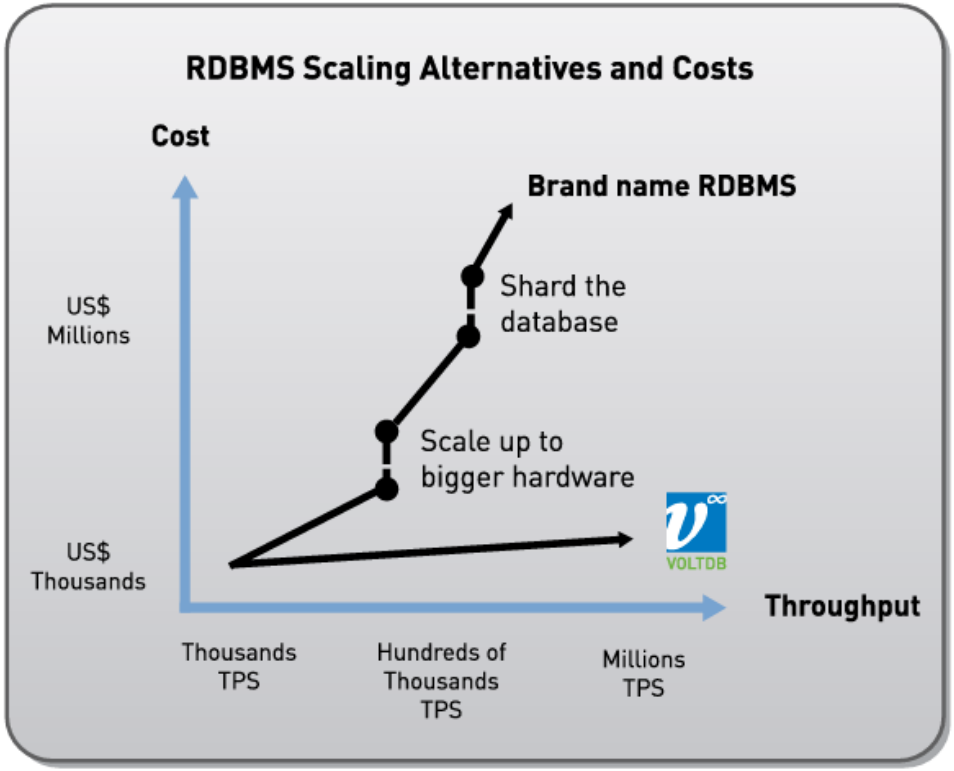


VoltDB (H-Store product)

- H-Store origin
- In-memory, shared nothing RDBMS
- Database is sharded with serial execution of transactions within a shard – user specifies partitioning column for a table
- Synchronous replication with reexecution of transactions at replicas
- Read only remote backup for disaster recovery
- Transaction-level logging rather than ARIES-style page-oriented logging
- Periodically consistent snapshots written to disk and transaction-level log used to do recovery
- Multi-shard transactions are expensive

VoltDB (H-Store product)

	Nodes	VoltDB	DBMSx	VoltDB Advantage
"TPC-C-like" workload (VoltDB lab)	1	53,000 TPS	1,555 TPS	45x better throughput
	12	560,000 TPS	N/A	Near-linear (.9) scaling

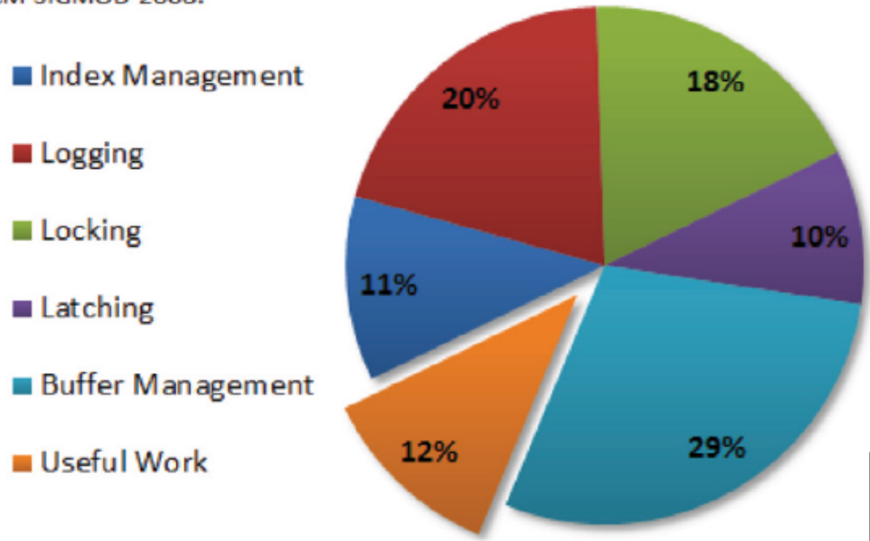


The Hype on it!

VoltDB

General Purpose RDBMS Processing Profile

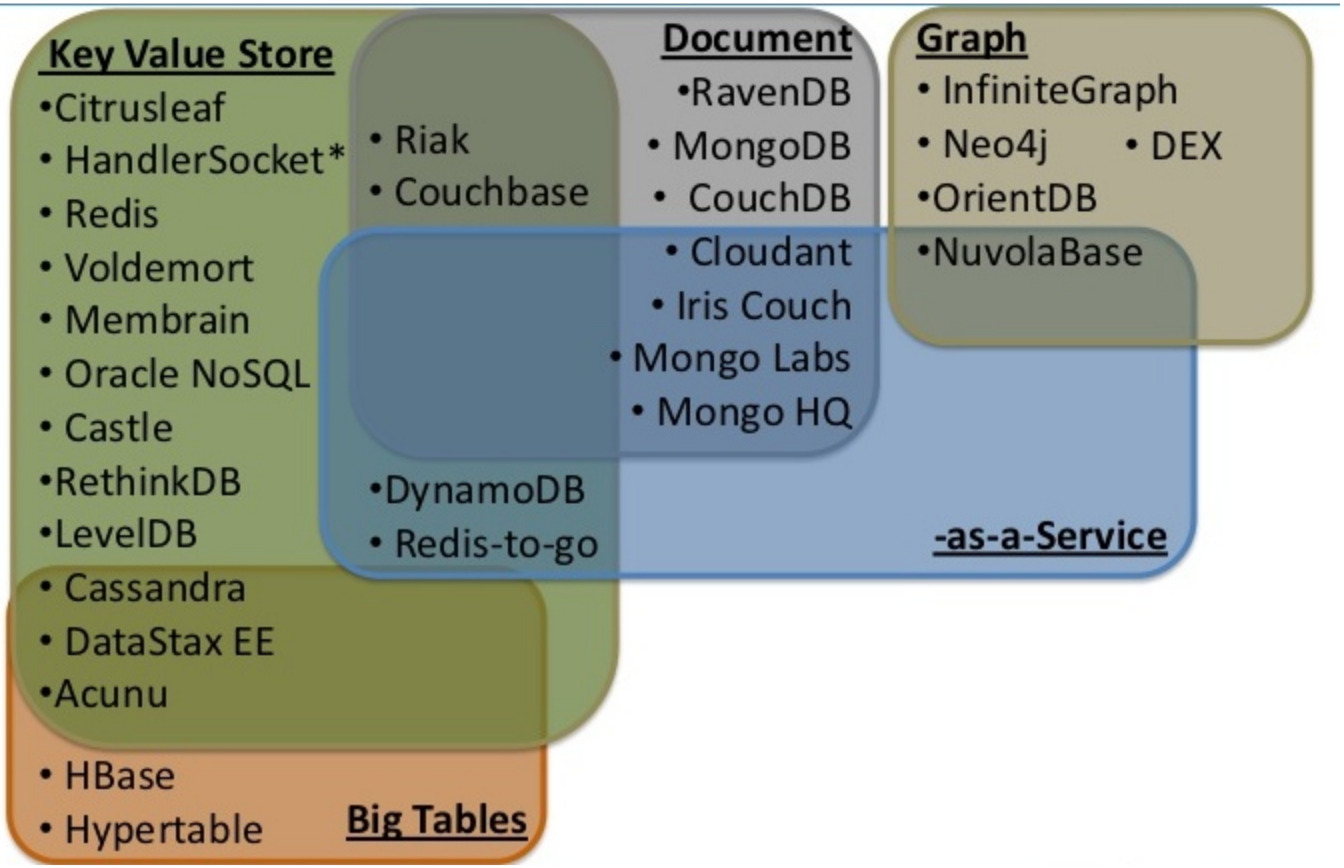
OLTP Through the Looking Glass, and What We Found There
Stavros Harizopoulos, Daniel Abadi, Samuel Madden, and Michael Stonebraker
ACM SIGMOD 2008.



	VoltDB	NoSQL	Traditional RDBMS
Scale-out architecture	✓	✓	
Built-in high availability	✓	✓	
Multi-master replication	✓	✓	
ACID compliant	✓		✓
SQL data language	✓		✓
Cross-partition joins	Automatic	In app code	In app code
Cost at Web scale	$	$$$	$$$$

NoSQL Systems

NoSQL Ecosystem - Matthew Aslett 2012



Motivations for NoSQL Systems

- Need for flexible data model, schema evolution, …
- Many apps (e.g., Web2.0) need fewer features, high scale (data sizes, # of servers), very low response times
- Native support for interchange/storage formats like JSON
- Sharding RDBMS DBs rigid or unfriendly
- High-scale RDBMS too expensive for simple apps
- Need for low-latency, low-overhead API to access data
- Need to scale-out on cheap commodity nodes with locally-attached SATA disks, especially in cloud context
- Increasing use of distributed analytics
- Desire for one programming language for imperative programming and persistent data accesses (ref: POS)
- For some, access to source code for tailoring for their needs
- Flexibility with respect to data consistency

NoSQL Systems

Focus on	Give up
▪ Commodity servers, networking, disks ▪ Easy elasticity and scalability to multiple racks (10s to 100s of servers) ▪ Fault-tolerance and high availability	▪ Relational data model ▪ Standardized SQL APIs ▪ Complex queries (joins, secondary indexes) ▪ Stronger notion of transactions

Two Worlds

Transactional

- Custom high-end OLTP for financial applications
- Scaleout datastores for Cloud/Web 2.0
- Examples
  - MemcacheDB, Cassandra, Dynamo, Voldemort, SimpleDB, Gigaspaces, Websphere eXtreme Scale (WXS)

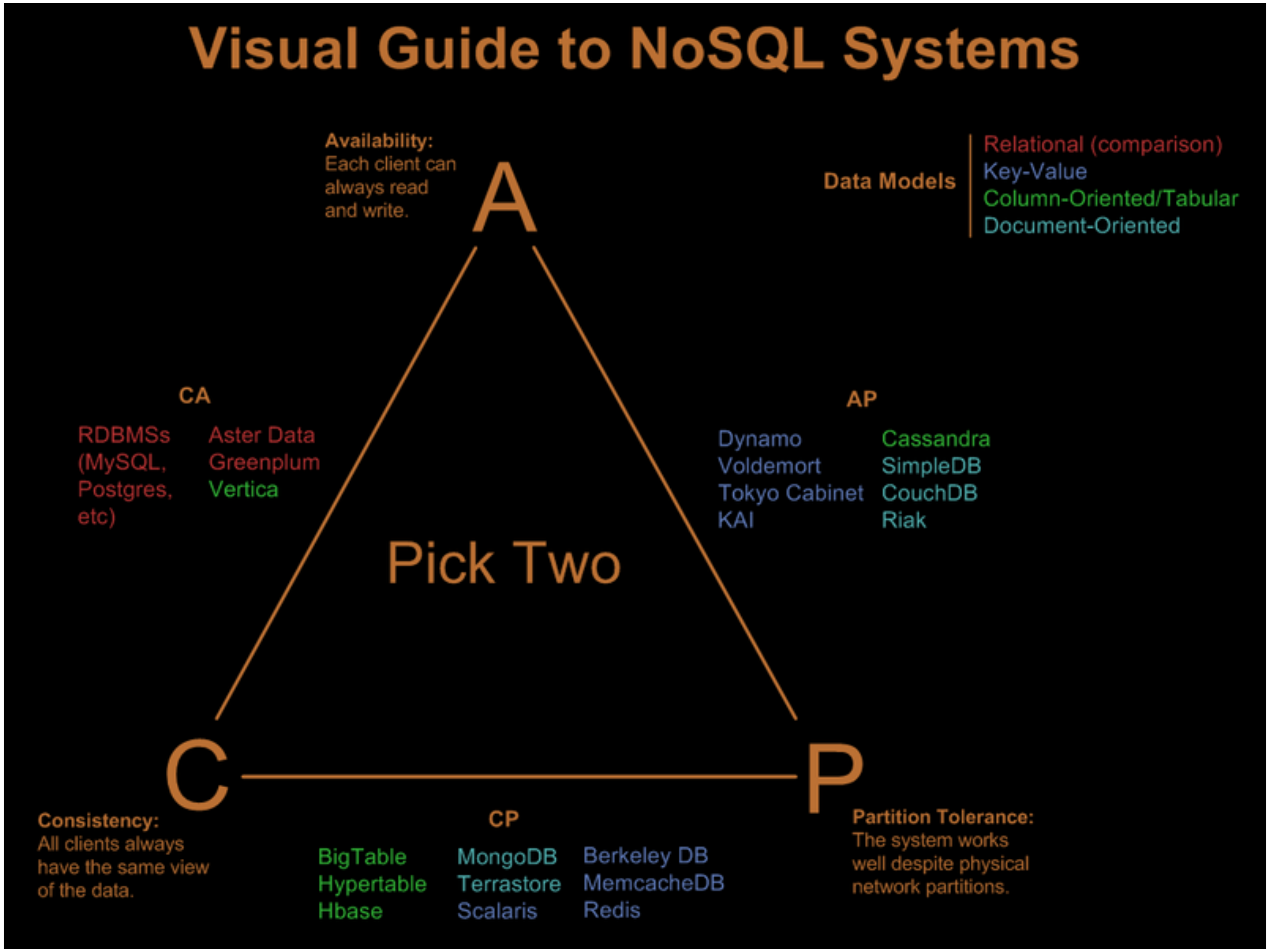
Analytics

- Managing updates
- Support for random access and indexing
- Scaleout content store
- Examples
  - Bigtable, HBase, Hypertable

NoSQL Stores

- Major open source projects
  - Cassandra and HBase
  - Other Projects: Voldemort, MongoDB, Hypertable
- Mushrooming of startups with tremendous interest from VCs (e.g., Aerospike)
- "**Scale-out structured storage**": between raw files and DBMS
- Commercial flex-schema offerings based on relational technologies: DB2/Informix JSON, MonetDB

One View of NoSQL Systems – Nathan Hurst, 3/2010?



http://blog.nahurst.com/visual-guide-to-nosql-systems

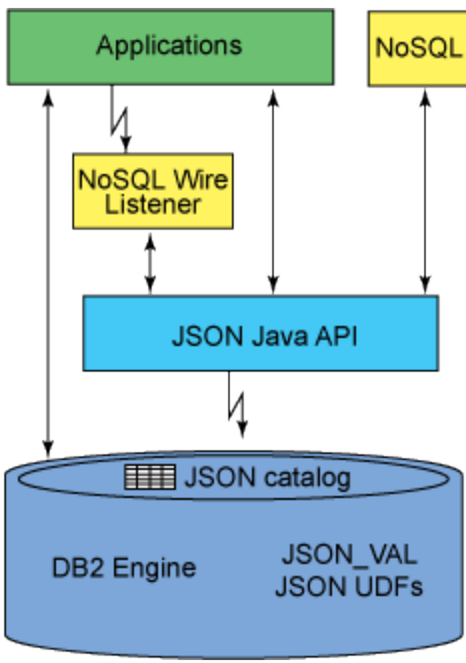
Concerns/Misgivings (NonsenSQL …)

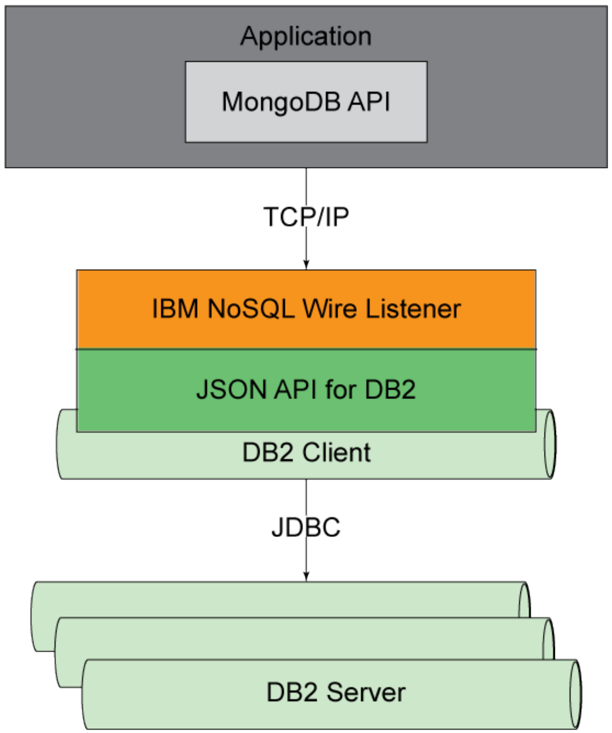
- Throwing baby with bath water analogy comes to mind … many very good developments of last 3 decades discarded … but lot of them coming back … albeit in ad hoc ways on a flaky base
- Many past systems that started simple ran into many scalability and other issues – S/38, AS/400, Lotus Notes/Domino, OODBMSs and RDBMSs with page as smallest unit of locking (e.g., mainframe DB2, Sybase/SQLServer, ObjectStore), Single level stores, …
- Seat of the pant design in newer systems by highly inexperienced people who haven't done their homework AND who aren't documenting enough their designs for others to recognize their deficiencies
- Too many misleading claims on novelty of some design features
- Ridiculous choices like database level lock in MongoDB
- Object granularity as unit of replication locally and for disaster recovery remotely
- Myths about ACID transactions and relaxed notions of consistency

MongoDB

- Open source document database system (similar to Lotus Notes)
- 1st release in 2009, Company behind it: MongoDB, Inc
- Native support for JSON-like docs with dynamic schemas (BSON)
- Indexing, including secondaries
- Master slave replication and high availability
- Auto sharding
- Simple query support
- MapReduce functionality could be invoked for complex aggregation
- Relies on file system for buffering
- GridFS for very large files storage
- Common use cases: operational and analytical big data, content management/delivery, mobile/social infrastructure, user data management and data hub.
- $231M funding, 300 employees, $1.2B evaluation!

DB2 NoSQL JSON





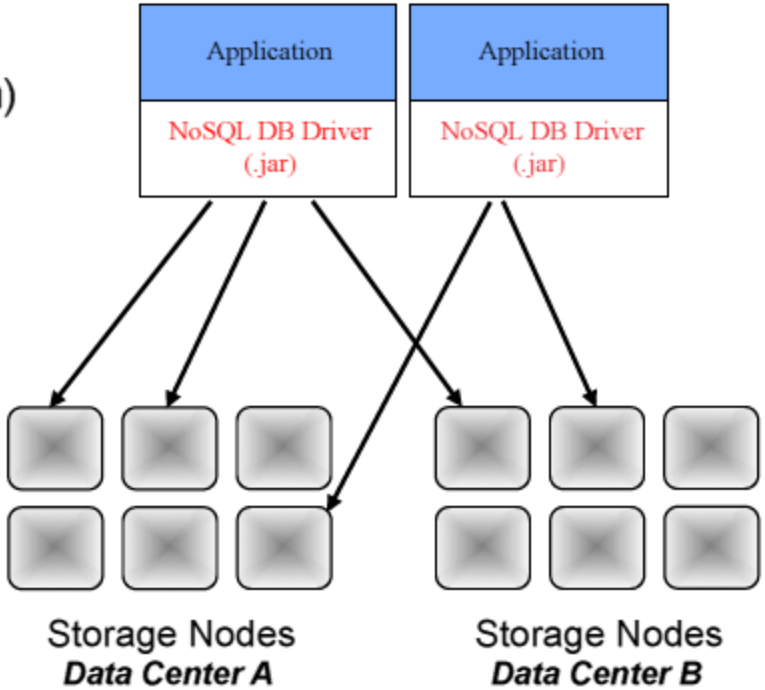
In addition to support of basic features of the MongoDB query language, DB2 JSON also provides DB2 database server-specific extensions. These extensions enable applying more DB2 database features to JSON documents, for example:

- Transaction control to group multiple operations into one commit scope.
- Bi-temporal data to associate business time and system time concepts with your JSON data.

Oracle NoSQL

A Distributed, Scalable Key-Value Database

- Simple Data Model
  - Key-value pair (with major/minor key paradigm)
  - CRUD + iteration
- Scalability
  - Data partitioning and distribution
  - Optimized data access via intelligent driver
- High availability
  - One or more replicas
  - Resilient to partition master failures
  - No single point of failure
  - Disaster recovery through location of replicas
- Transparent load balancing
  - Reads from master or replicas
  - Driver is network topology & latency aware



Lamb, C. Oracle NoSQL Database, Presentation at International Workshop on High Performance Transaction Systems (HPTS), Asilomar, October 2011, http://hpts.ws/papers/2011/sessions_2011/cwl-hpts-for-website.pdf

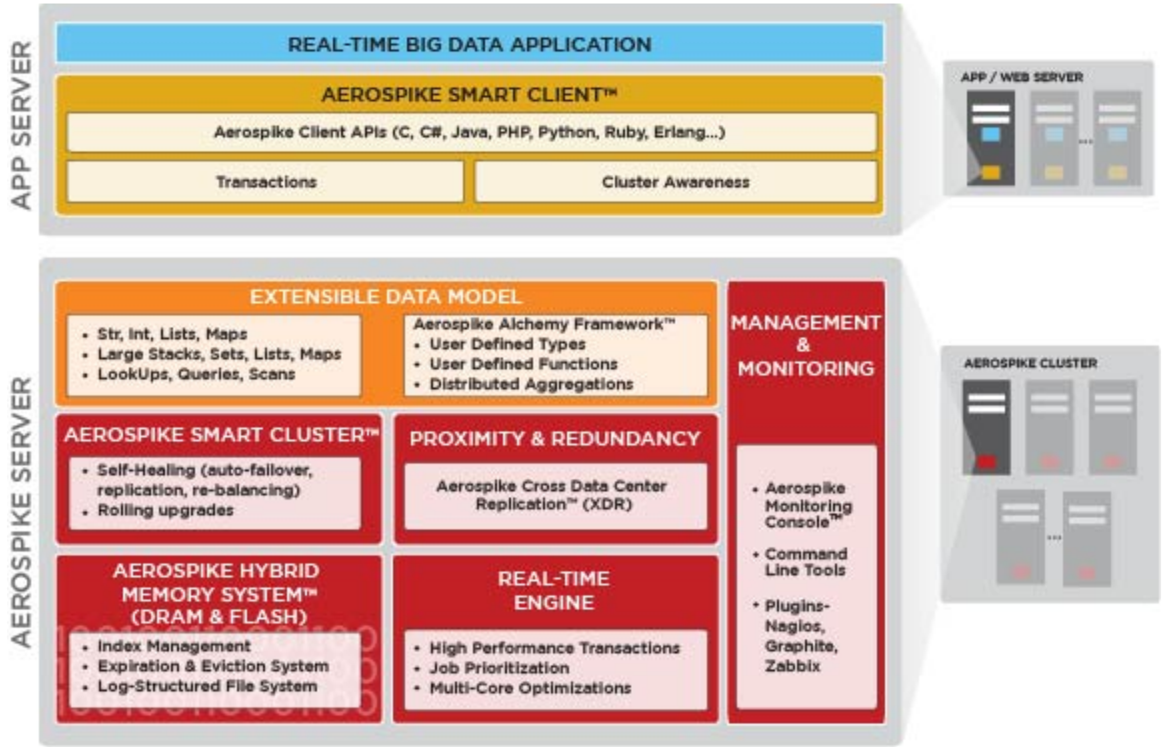
Oracle NoSQL Building Block

Berkeley DB Java Edition

- Robust storage for a distributed key-value database
  - ACID transactions
  - Persistence
  - High availability
  - High throughput
  - Large capacity
  - Simple administration
- Already used in
  - Amazon Dynamo
  - Voldemort (LinkedIn)
  - GenieDB

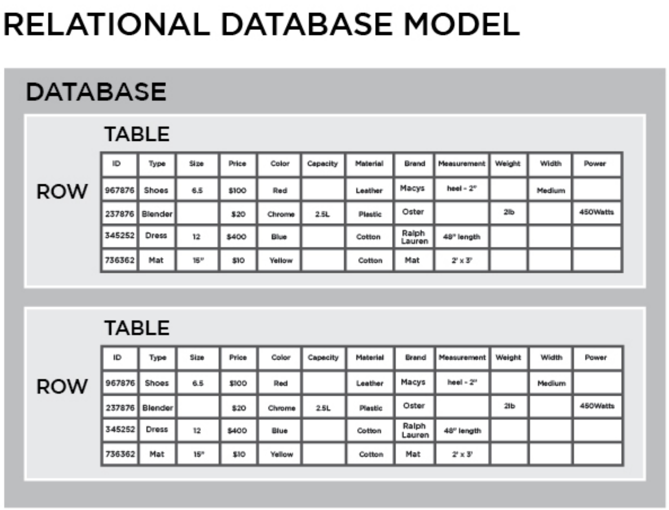
Figure 2: The Oracle NoSQL Database Flexibility Spectrum

Aerospike



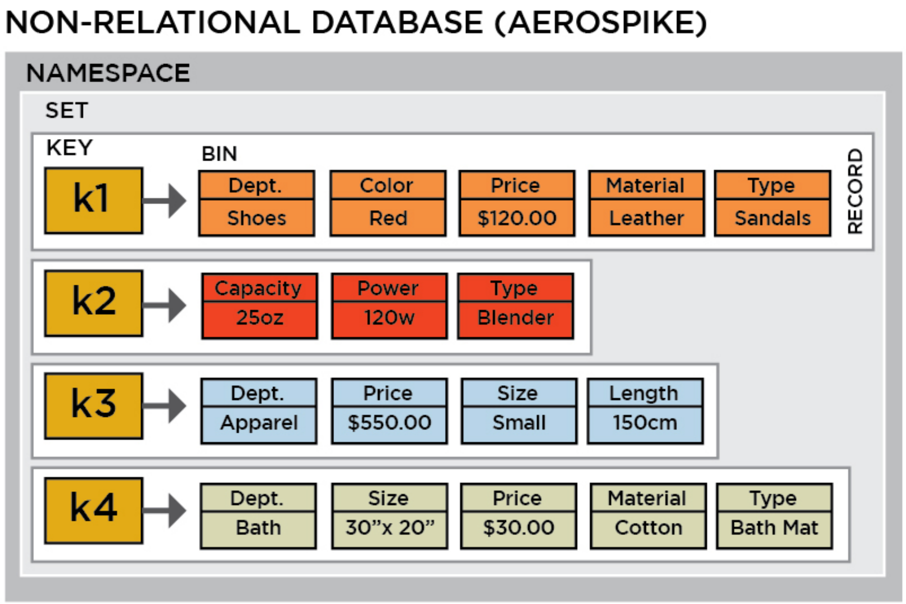
- Advocates a hybrid DRAM & Flash system with indexes stored in DRAM & data in flash, disks for persistence
- Direct access to flash with own LFS, optimized via small block reads and large block writes
- Shared nothing with data distribution using randomized hashing and identical nodes
- Uses Paxos for cluster configuration, not for commit logic: all nodes have to agree, not just a majority
- Implemented in C for speed with an extensible data model
- Master/Slave and Master/Master replication supported
- Acquired alchemyDB (used Redis before) being ported onto Aerospike

Aerospike



ID	Type	Size	Price	Color	Capacity	Material	Brand	Measurement	Weight	Width	Power
967876	Shoes	6.5	$100	Red		Leather	Macys	heel - 2"		Medium	
237876	Blender		$20	Chrome	2.5L	Plastic	Oster		2lb		450Watts
345252	Dress	12	$400	Blue		Cotton	Ralph Lauren	48" length			
736362	Mat	15"	$10	Yellow		Cotton	Mat	2' x 3'			

The data is presented as relations - a collection of tables with each table consisting of a set of rows and columns



The data is presented as flex-schema and provides simplicity of design, horizontal scaling and finer control over availability

- 500K TPS on $2K server
- 1M TPS on $5K server
- $1 for 250 TPS
- Scaling by mitigating network-I/O-soft-interrupt overhead & avoiding unnecessary context switches – isolate NIC-Physical CPU, NUMA sensitive memory allocation, dedicated core for NIC hardware interrupt handling on > 4 core CPUs

Facebook Case Study

- mySQL + memcache for real time access to PBs of data
- TAO for consistency across big geographies
- Haystack for managing photos (3B new ones per week)
- Apache Hadoop for mining intelligence from 100PBs of click logs
- Apache HBase for FB messages
- Data warehouse in Hive
- Scuba for real time event data handling – in-memory

Hybrid Systems: Hadoop + SQL

Hadoop

IBM InfoSphere BigInsights

Hortonworks

MAPR TECHNOLOGIES EASY. DEPENDABLE. FAST.

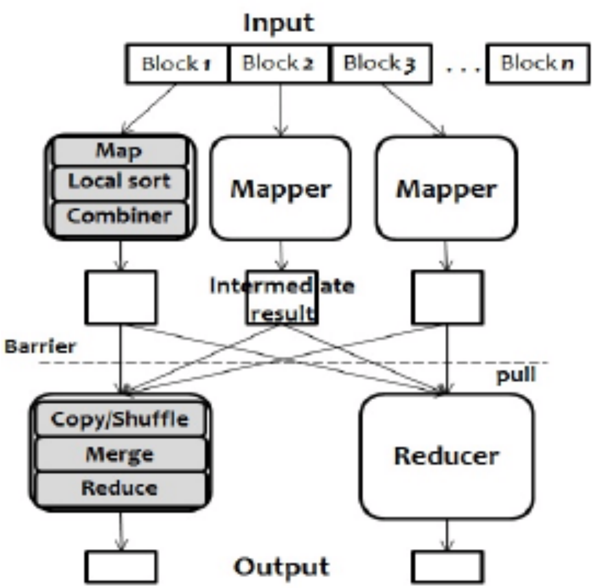
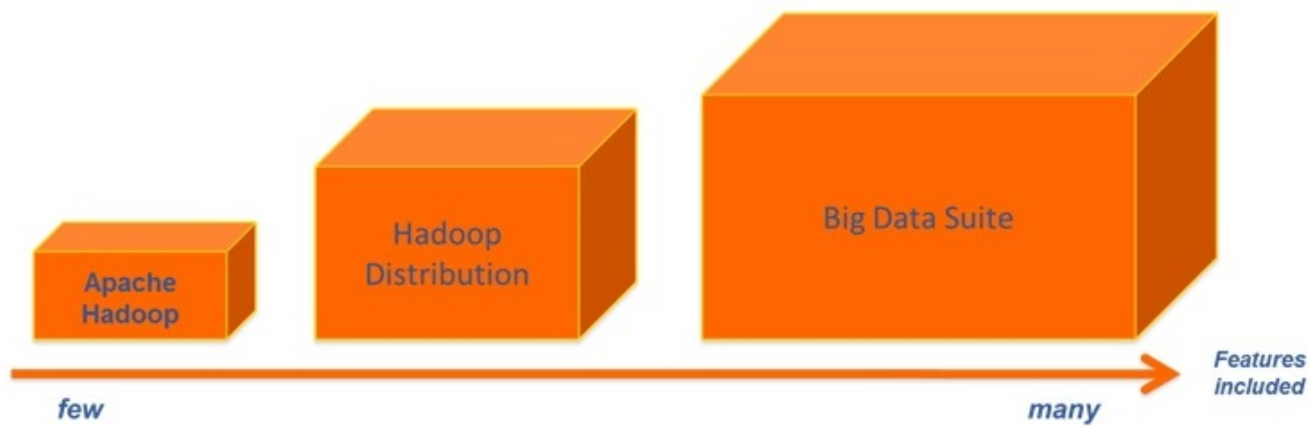


Figure 1: Hadoop Architecture



MapReduce
HDFS
Ecosystem

+

Packaging
Deployment-Tooling
Support

+

Tooling / Modeling
Code Generation
Scheduling
Integration

Hadoop

- Apache Hadoop Modules: Common, Hadoop Distributed File System (HDFS), MapReduce, YARN
- Hadoop Ecosystem: Flume, HBase, Hive, Pig, Sqoop, Zookeeper
- Hadoop Distributions: Amazon Elastic MapReduce, Cloudera CDH, HortonWorks, IBM BigInsights, MapR, Pivotal HD
- Cloudera Impala: Distributed query execution engine for Apache HDFS/HBase data
- Apache Hadoop 2 (Rel 2.2.0) GAed on 15 October 2013
- Largest known production Hadoop environment: >35K nodes at Yahoo
- New: HA and Snapshots for HDFS data, YARN (aka MRv2 – ResourceManager, ApplicationMaster, NodeManager), Federation support (Multiple Independent Namenodes & Namespaces)

Summary

- Good that many shortcomings of RDBMSs began to be addressed by user community (Web 2.0 companies)
- More rigor needed in rationalizing design choices
- Just because data model isn't relational doesn't mean internals have to be different or lessons from the past don't apply
- Let us not rediscover problems and adopt solutions with significant shortcomings
- Longer time issues not considered in addressing more immediate needs (migration, skills, tools, optimization, …)
- Let us not allow history to repeat itself
- Shakeout and market consolidation will happen