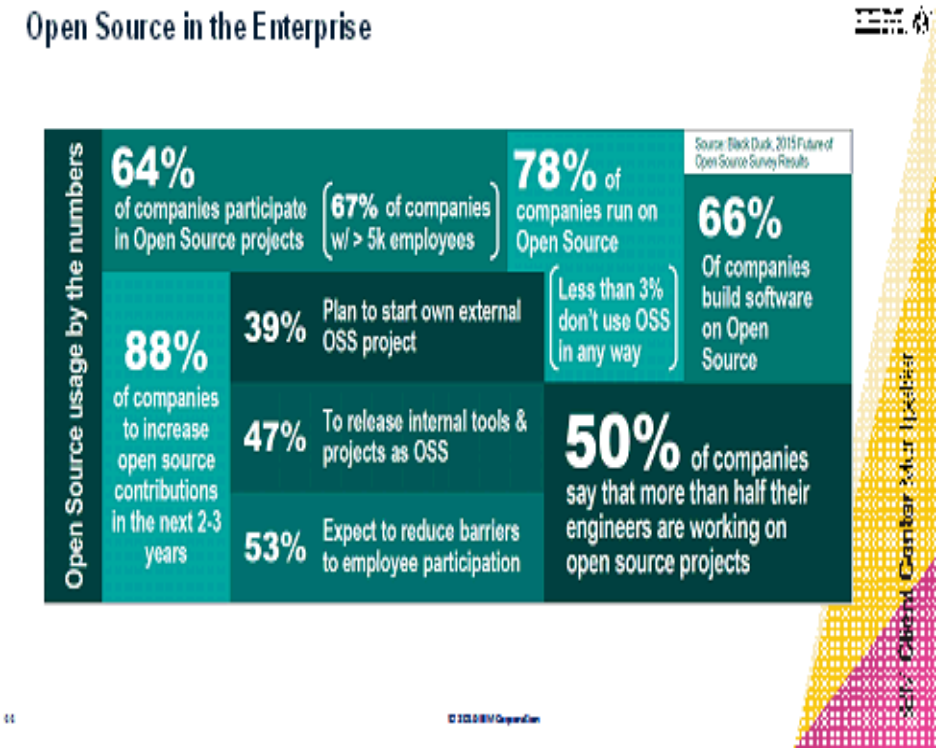
Open Data on Power Linux

Pierluigi Antoniolli
Senior Consultant and IT Architect
Pantoni@us.ibm.com

Open Source Software – enabling digital transformation



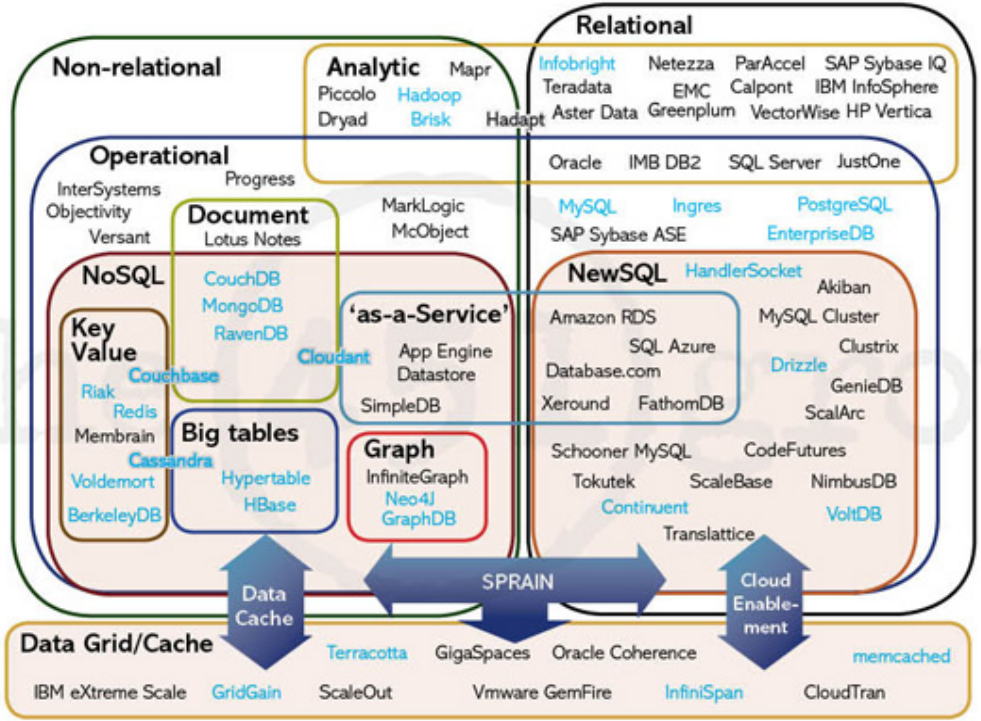
Ed Boyajian, president and CEO of EDB,, "If 80 percent to 90 percent of your IT spend is on maintenance (such as proprietary software licensing and such), that leaves very little budget for innovation."

Open source undoubtedly speeds the digital transformation for most companies. I see this happening in several ways. The factor that most people think of, of course, is money — the less a company spends on proprietary software, the more it can dedicate to other facets of digital transformation efforts. Open source software allows companies to derive benefit not only from their own IT employees, or from the employees of proprietary software vendors, but from the whole open source community. "For example, we've seen this recently in analytics: Apache Spark has emerged very quickly as a faster successor to Hadoop's MapReduce, and I'm sure there will be a new generation coming soon. You don't see that speed of innovation in proprietary software.. **Kelly Stirman, VP,** MongoDB

The rapid adoption of open source technologies across organizations has provided a network-effect of reliability in resiliency, scalability, affordability and adaptability. Not only are these traits critical to open source technology, but they are also necessary for successful digital transformations in business. As a result, there is a strong correlation between open source and digital business models. The two go hand-in-hand.“ **Ritika *Gunnar IBM vice president of data and analytics***

http://www.cmswire.com/digital-experience/how-open-source-guides-digital-transformation/

Modern Database Offering Taxonomy



Relational or NoSQL?

Relational	NoSQL
Queries	Scalability
Transactions	Management
Consistency	Schema Flexibility

More detailed info at the Datafloq site (https://datafloq.com/big-data-open-source-tools/os-home/).

The changing database landscape

The digital economy is driven by big data.

To deal with it, companies require more agile, flexible, and scalable tools.

*"By 2018, more than **70%** of new in-house applications will be developed on an **OSDBMS**."* [1]

– Gartner

The vocabulary of Database software

- **Relational Database Management Systems** – Rows, Tables, Columns, Records, SQL, Keys, Indexes , full table scans, transactions, ACID Compliant.
- **Document DBs – example MongoDB** - Instead of tables, a MongoDB database stores its data in *collections*. A collection holds one or more *BSON documents*. Documents are analogous to records or rows in a relational database table. Each document has *one or more fields*; fields are similar to the columns in a relational database table.
- **Redis** is an open source (BSD licensed), in-memory **key value structure store**, used as database, cache and message broker. It supports data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, hyperloglogs and geospatial indexes
- **Cassandra's** data model offers the convenience of column indexes with the performance of log-structured updates, strong support for materialized views, and powerful built-in caching.
- **Graph DBs (Neo4J)** – Graph DBs contain connected entities (the *nodes*) which can hold any number of attributes (key-value-pairs). Nodes can be tagged with labels representing their different roles in your domain. In addition to contextualizing node and relationship properties, labels may also serve to attach metadata—index or constraint information—to certain nodes. *Relationships* provide directed, named semantically relevant connections between two node-entities. A relationship always has a direction, a type, a *start node*, and an *end node*. Like nodes, relationships can have any properties. In most cases, relationships have quantitative properties, such as weights, costs, distances, ratings, time intervals, or strengths. Note that although they are directed, relationships can always be navigated regardless of direction.
- **Mutli-Model database** - A database supporting more than one format (e.g. Document and Key-Value)

All in the NoSQL Family

NoSQL databases are geared toward managing large sets of varied and frequently updated data, often in distributed systems or the cloud. They avoid the rigid schemas associated with relational databases. But the architectures themselves vary and are separated into four primary classifications.

Document databases
Store data elements in document-like structures that encode information in formats such as JSON. Common uses include content management and monitoring Web and mobile applications.
EXAMPLES: Couchbase Server, CouchDB, MarkLogic, MongoDB

Graph databases
Emphasize connections between data elements, storing related "nodes" in graphs to accelerate querying. Common uses include recommendation engines and geospatial applications.
EXAMPLES: InfiniteGraph, Neo4j

Key-value databases
Use a simple data model that pairs a unique key and its associated value in storing data elements. Common uses include storing clickstream data and application logs.
EXAMPLES: Aerospike, DynamoDB, Redis, Riak

Wide column stores
Also called table-style databases—store data across tables that can have very large numbers of columns. Common uses include Internet search and other large-scale Web applications.
EXAMPLES: Accumulo, Cassandra, HBase, Hypertable, SimpleDB

- **Document databases** are generally useful for content management systems, blogging platforms, web analytics, real-time analytics, ecommerce-applications.

- **Graph databases** are very well suited to problem spaces where we have connected data, such as social networks, spatial data, routing information for goods and money, recommendation engines

- **Key-value databases** are generally useful for storing session information, user profiles, preferences, shopping cart data.

- **Column family databases** are generally useful for heavy write volume such as log aggregation / Internet of Things data

Source: TechTarget

RDBMs and NoSQL system types

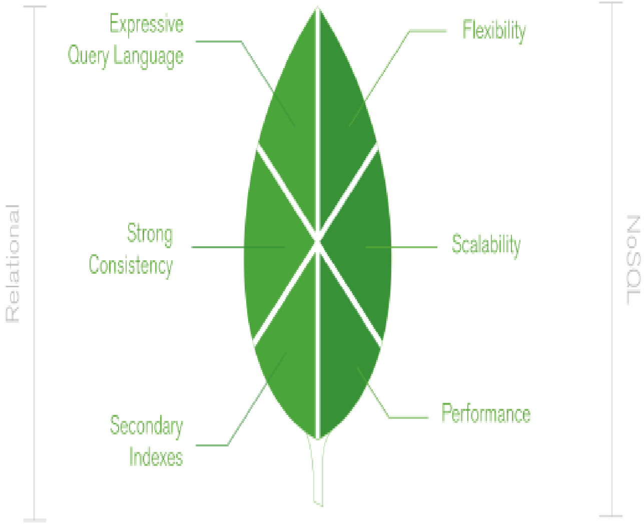
- **Relational database management systems (RDBMS)** support the relational (=table-oriented) data model. A pre-defined schema describes the database structure, which includes tables defined by a table name and a fixed number of attributes with fixed data types. A record (=entity) corresponds to a row in the table and consists of the values of each attribute

- **Document Databases** (eg – MongoDB) store data in Documents, Documents contain one or more Fields. Data can be queried based on any combination of fields in a document. The appeal of these systems is that that are very general purpose, have large application ecosystems and map very nicely to support and enable many of today's object oriented programing styles.

- **Key Value Store Databases** (eg – Redis) are the most basic type of non-relational DBs. They store a Key and associated Values.

- **Wide Column Stores** (example – Cassandra) vary in the number of Columns that are stored. The appeal of these systems is around their very high performance and scalability. For example, some customers running Cassandra and their tables have > 30,000 columns. Oracle RDBMS can have a maximum of 1000 columns per table, 200-300 is more "normal".

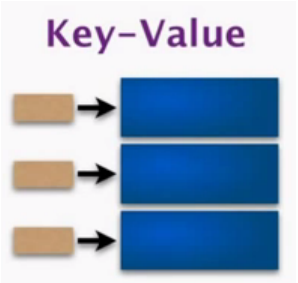
- **Graph Databases** – (eg – Neo4j) focus on storing simple and complex relationships and can be queried to discover simple and more complex relationships between the data.



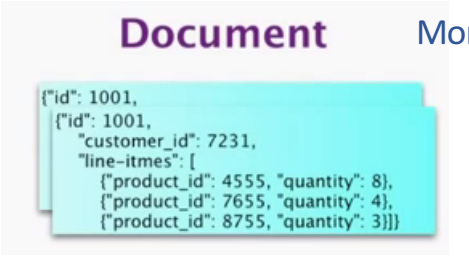
RDBMS usually assume storage is expensive, and NoSQL databases assume storage is cheap

Harumquiae cum qui alit andi blat volor sequia nis quid utet litibus, si ipsandae nos iditate mpores sam, vendigendici necti derorestium del id quatem quatia volorporum imus nestrunti sapiendem fugia imaio officabo. Igendae sunt.

Four main types of NoSQL DBs



Redis



MongoDB

Wide column store

Cassandra

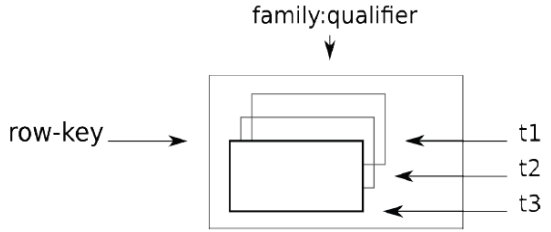
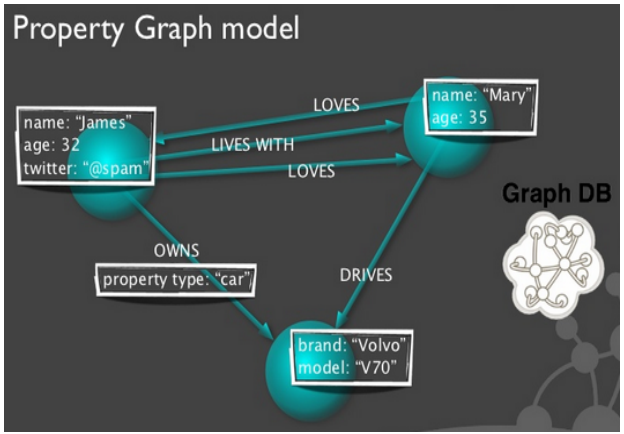


Figure 3.2: A value in BigTable is referenced by its row key, the column key consisting of the column family name and the qualifier and a timestamp. Each row can have any number of columns.



Neo4J

What/When – Open Data on Power

Name	Classification	Optimized for	Lead with for these use cases / data types	Complexity – install and use
MongoDB	NoSQL - Document Store	Document Model, Document stores, semi-structured or unstructured data.	Single view of Customer records, Enterprise content management, catalogs, personalization	Low
Redis	NoSQL - in memory Key Value Store	Data queues, Strings, Lists, Counts, caching, Statistics, Text, session IDs, pictures, videos	Live in memory cache, data queues, User session data, shopping cart data,	Very low
Cassandra	NoSQL - Wide Column Store	NoSQL environments that need Very High Performance and Scalability, Very High data volumes	Messaging, Fraud detection, Internet of Things data – sensor data, log data, telco call detail records	Low
Neo4J	NoSQL - Graph Store	Data stored as edges, nodes, or attributes (Graphs).	Fraud detection, Social Network Analysis, Location aware apps, Master data mgmt., Machine Learning	High
PostGres (Enterprise DB)	Open source Object Relational database	Wide variety of transactional work at lower TCO – relational/structured queries to object store and retrieval	Oracle RDBMs migrations and take-outs	Medium
MariaDB	Open source Relational database	Lower cost transactional SQL based queries and updates	Migrations from Oracle MySQL, Turbo LAMP stack	Medium

Example - E-commerce app leveraging multiple Databases

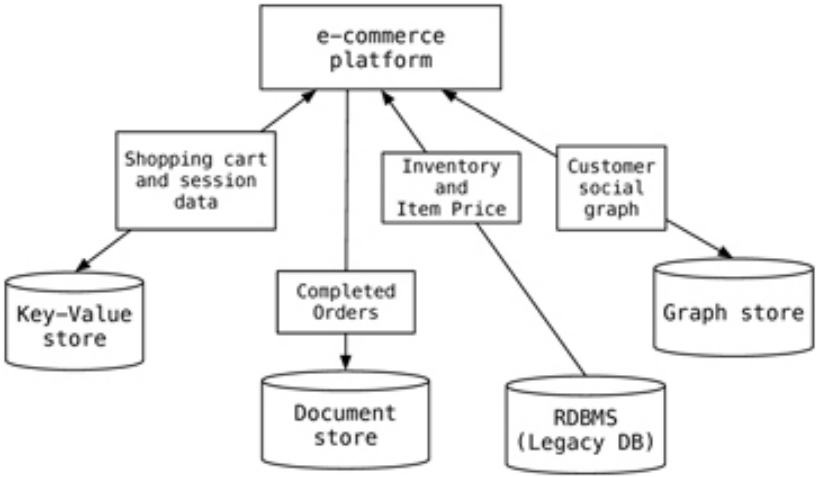
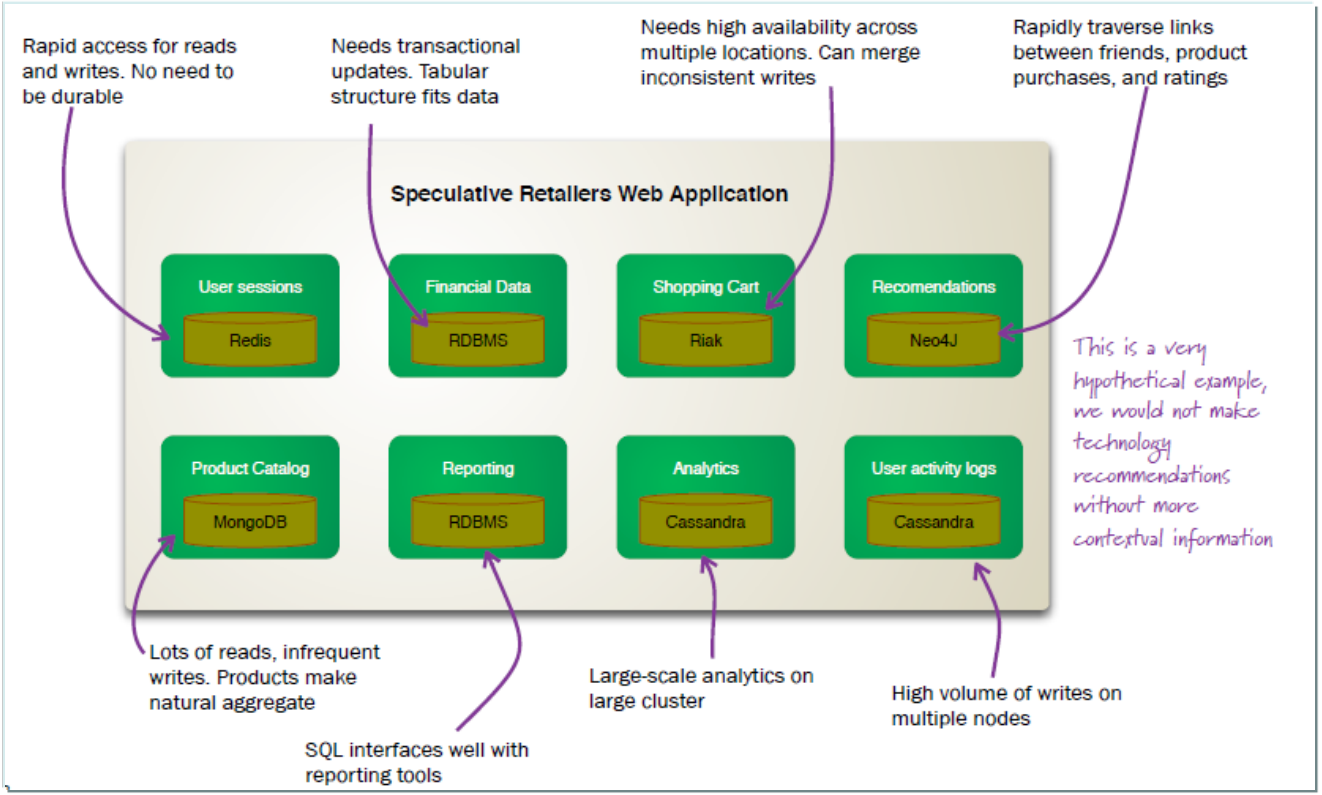


Figure 13.3. *Example implementation of polyglot persistence*

http://i0.wp.com/www.jamesserra.com/wp-content/uploads/2015/04/pp.png

What might a client environment look like – Relational and Non-Relational data stores

From Polyglot Persistence – Martin Fowler & Pramod Sadalage (Thoughtworks)



Open Source Databases on Linux on Power

	Name	Classification	Optimized for	Lead use cases / data types	CAPI-Flash Enabled
RDBMS	PostGreSQL (EnterpriseDB)	Open source Object Relational database	Wide variety of transactional work at lower TCO – relational/structured queries to object store and retrieval	Oracle RDBMs migrations and take-outs PL/SQL compatibility, Oracle SQL syntax and semantics	No
	MariaDB	Open source Relational database	Lower cost transactional SQL based queries and updates	Migrations from Oracle MySQL, Turbo LAMP stack	No
NoSQL	MongoDB	NoSQL - Document Store	Document Model, Document stores, semi-structured or unstructured data.	Single view of Customer records, Enterprise content management, catalogs, personalization	No
	Redis	NoSQL - in memory Key Value Store	Data queues, Strings, Lists, Counts, caching, Statistics, Text, session IDs, pictures, videos	Live in memory cache, data queues, User session data, shopping cart data,	Yes
	Cassandra	NoSQL - Wide Column Store	NoSQL environments that need Very High Performance and Scalability, Very High data volumes	Messaging, Fraud detection, Internet of Things data – sensor data, log data, telco call detail records	No
	Neo4J	NoSQL - Graph Store	Data stored as edges, nodes, or attributes (Graphs).	Fraud detection, Social Network Analysis, Location aware apps, Master data mgmt., Machine Learning	GA in 3Q

MongoDB

- **MongoDB is a NoSQL Open Source Document Database**
- JSON-like documents with dynamic schemas (MongoDB calls the format BSON)
- As of May 2016, MongoDB was the fourth most widely mentioned database engine on the web, and the most popular for document stores.
- ibm.co/29Rr51J - getting started with MongoDB Enterprise on Linux on Power
- MongoDB Enterprise LoP Tuning Guide: ibm.co/28Qv5iK

- **Forbes** gains critical insight into the social sharing of their articles, to capitalize on stories going viral in real-time.
- **Otto,** Europe's second-largest e-commerce company, continually updates its catalog of over 2M products to provide a one-to-one shopping experience for 30M shoppers and drive €2.3B in revenue.
- **Bosch** has built its Internet of Things suite on MongoDB, bringing the power of big data to a new range of Industrial Internet applications including manufacturing, automotive, retail, energy and many others.
- **Expedia** is a virtual concierge – Using MongoDB, it pushes special travel offers to users in real time by tracking their searches and comparisons across its site.

Redis

•**Main points:** Simple values or data structures by keys. Blazing fast
•**Exploits Power 8**: Redis Labs on Power utilizes IBM POWER8 servers, the Flash System, the IBM CAPI-Flash card and the Redis Labs Enterprise Cluster (RLEC) for Flash software.
•**Other features :** Master-slave replication, automatic failover
Best used: For rapidly changing data with a foreseeable database size (should fit mostly in memory).
For example: To store real-time stock prices. Real-time analytics. Leaderboards. Real-time communication.

Redis on Power systems
https://redislabs.com/solutions/redis-labs-and-ibm-power-systems

- **Instagram** – Instagram uses Redis heavily to run their main feed, activity feed and session store. You can check out some of the articles about their infrastructure here and here, see how they scaled in a great presentation by one of the founders, and read about their switch from Cassandra to Redis. Lastly, check out Redis-Faina, a query analyzer that the engineering team at Instagram built.
- **Github** – Github is using Redis for exception handling and queue management. They also use Redis for configuration management, and as a persistent key/value store for routing all kinds of data.
- **Stack Overflow** – Stack Overflow uses Redis as a caching layer for their entire network. They praise the speed with which Redis is able to perform: "In our (admittedly limited) experience, Redis is so fast that the slowest part of a cache lookup is the time spent reading and writing bytes to the network." They report that about 1,300,000 keys are being stored by Redis at any given time, most of which expire within minutes. At most, several hundred read/write operations per second occur within Redis. They use around 6GB of memory and have extremely low CPU usage (1%).
- **Twitter** - Twitter makes heavy use of Redis, and has open-sourced some of the projects they built internally to take advantage of Redis. Twemproxy is a fast proxy for Redis that reduces the connection count on backend caching servers. Manju, the creator of Twemproxy, talked about it during the SF Redis Meetup. Check out the slides!
- **Tumblr** - Tumblr uses Redis to power dashboard notifications for their tens of millions of users. To do so, they built Staircar, a tool that gave vastly better performance than the MySQL setup they were previously using for notifications. Redis a key part of their scalable architecture, as their high scalability interview demonstrated.

http://kkovacs.eu/cassandra-vs-mongodb-vs-couchdb-vs-redis

Cassandra

•**Main point:** Store *huge* datasets , retrieves in "almost" SQL (CQL3)
•**Other features:** CQL3 is the official interface and very similar SQL, but with some limitations that come from the scalability (most notably: no JOINs, no aggregate functions.) Querying by key, or key range (secondary indices are also available).

•Highly scalable and highly available with no single point of failure

•Very high write throughput and good read throughput. Writes can be much faster than reads (when reads are disk-bound). Tunable consistency and support for replication

•Flexible schema. Map/reduce possible with Apache Hadoop
•Very good and reliable cross-datacenter replication

Best used: When you need to store data so huge that it doesn't fit on server, but still want a friendly familiar interface to it.
For example: Web analytics, to count hits by hour, by browser, by IP, etc. Transaction logging. Data collection from huge sensor arrays.

Source - http://kkovacs.eu/cassandra-vs-mongodb-vs-couchdb-vs-redis

Cassandra use cases

There are at least 3,000 companies that we know of using Cassandra in production. Over the past few months we have been digging into the applications that sit upon Cassandra, and a fascinating pattern emerges. More than 80% of the use cases fit into just five classifications of application:

1.Product Catalog/Playlist
2.Recommendation/Personalization Engine
3.Sensor Data/Internet of Things
4.Messaging
5.Fraud Detection

– Product Catalog/Playlist: Whether you are browsing an article on AOL., or queuing up your Spotify playlist to go out for your morning jog, you are using an application built on top of Apache Cassandra. Cassandra is a great fit for this use case due to its high availability across multiple data centers and its ability to scale predictably.

– Recommendation/Personalization Engine: Without these types of systems how on earth would we know what events to buy tickets to, or what other articles we might find interesting? Eventbrite now uses Cassandra instead of MySQL to power their mobile experience, letting users know what events are happening around them that they will be interested in attending. Eventbrite chose Cassandra for its read/write capacity and ease of deployment. Outbrain, a company you use frequently, but may be unfamiliar with, uses Cassandra to power their content discovery platform, helping companies add revenue streams by serving up applicable third-party articles you may find interesting. One of my favorite Cassandra quotes of all time comes from Outbrain just after Hurricane Sandy: "During Hurricane Sandy, we lost an entire data center. Completely. Lost. It. Our application fail-over resulted in us losing just a few moments of serving requests for a particular region of the country, but our data in Cassandra never went offline."

– Sensor Data/Internet of Things: A buzz-worthy space at the moment to be sure. You can't go a day without reading a headline about how the internet of things is transforming the world. Well the internet of things needs a "database of things", and due to its ability to handle high velocity time series data, Cassandra is a great choice. The National Renewable Energy Laboratory (NREL), owners of the world's most environmentally friendly building, use Cassandra to store sensor data, and analyze it to provide ways to save water and energy, while the world's smartest thermostat sits on top of Cassandra and learns about your energy usage patterns, and automatically adjusts settings, even when you aren't there to program it.

– Messaging: I have yet to get my hands on Comcast's new X1 platform, but one of the things I am excited about is its messaging capabilities. I love sports, and the ability to track games that I interest me and get alerts through my television while I am watching something else.

– Fraud Detection: Another headline grabber; security threats are rising, and it seems that companies are playing catch-up on their smart fraud detection capabilities. Fraud detection capabilities get smarter with access to more and more data to find anomalies in the data patterns. Internet Identity uses Apache Cassandra to provide defensive and preventative measures for their customers to ensure their online properties are not compromised and stolen.

Neo4J

•**Main point:** NoSQL Graph database optimized for connected data

•**Exploit Power 8 features:** Neo4j on POWER8 offers 56 TB of extended memory, drastically increasing the size at which realtime graph queries are possible. Real-time graph processing with Neo4j on POWER8 supports both standard operational requirements and analytic insights that normally require offline processing. IBM POWER8 hardware allows Neo4j to scale both up and out for graphs of greater size than ever before.

•**Other features:** HTTP/REST (or embedding in Java)

•Full ACID conformity (including durable data)

•Integrated pattern-matching-based query language ("Cypher")

•Indexing of keys, nodes and relationships

•Advanced path-finding with multiple algorithms

•Optimized for reads

•Has transactions (in the Java API)

•Clustering, replication, caching, online backup, advanced monitoring and High Availability are commercially licensed

Best used: For graph-style, rich or complex, interconnected data.

For example: For searching routes in social relations, public transport links, road maps, or network topologies.

Try it - https://neo4j.com/download/

Contact Neo4J for the Neo4J commercial software for Linux on Power.

Neo4J on IBM POWER8

neo4j

The World's Leading Graph Database

Neo4j on IBM POWER8 – Solutions Data Sheet

The union of Neo4j and IBM POWER8 presents a golden opportunity for today's enterprise leaders and application developers.

Conclusion

The union of Neo4j and IBM POWER8 presents a golden opportunity for today's enterprise leaders and application developers.

With the power of these two technologies, data professionals overcome their big data challenges with a graph database that extracts meaningful, real-time insights from data relationships and the hardware that allows it to scale to massive proportions.

Some enterprise applications that would most benefit from Neo4j on POWER8 include:

- Master datasets connecting organizational, customer and product data together with user activity, inside of a single graph
- Inventory or supply chain management data for global enterprise manufacturers
- Detecting sophisticated fraud across billions of international real-time transactions, whether for insurance, banking or e-commerce
- Monitoring national or international IT networks for outages, dependencies and impact analysis
- IoT applications to manage the vast number of direct and indirect connections between devices, users, locations and transactions

Of course, these are only a few of the many and varied use cases employed by enterprises who harness the strength of Neo4j on POWER8.

Not only does Neo4j on POWER8 deliver real-time graph processing, but the robustness of IBM hardware allows enterprises to scale their graph data to levels never previously imagined.

"You're offloading all the I/O from the CPU…and so if you relieve that work from the CPU, then you can use it for computation and do bigger and more powerful things." - explains Philip Rathle, Neo Technology's vice president of products.

"The result is up to 10x the I/O bandwidth for the graph compared to running on Intel X64 servers or Power without CAPI flash," he says.

IBM Data Engine for NoSQL Overview

Cost Savings for In-Memory NoSQL Data Stores

IBM Data Engine for NoSQL is an integrated platform for large and fast growing NoSQL data stores. It builds on the CAPI capability of POWER8 systems and provides super-fast access to large flash storage capacity. It delivers high speed access to both RAM and flash storage which can result in significantly lower cost, and higher workload density for NoSQL deployments than a standard RAM-based system.

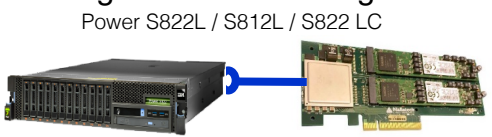
redis neo4j

NEW Q3

External Flash Configuration

Up to 56TB of extended memory with one POWER8 server + CAPI attach FLASH

Integrated Flash Configuration



Up to 8TB of super-fast storage tier on one POWER8 server

- Offers an integrated platform for implementing large and fast-growing NoSQL data stores without the complexity and cost typically associated with x86 implementations
- Provides flexible tuning to optimize high-speed access to RAM and flash storage to help reduce costs while delivering higher workload density
- Enable deployment of larger NoSQL databases on a single or smaller number of servers as well as the ability to consolidate multiple NoSQL databases on a smaller footprint
- Pre-integrated solution with select ISVs available from IBM® Rapid Build program partners

Microservices architecture with Docker Containers

- NoSQL Databases are often deployed as part of a microservices architecture using Docker
- Docker Containers are available for Power with Ubuntu and RHEL
- Provides very high density on Power

10K docker containers created in a 20-core VM on POWER8

Demo at DockerCon Europe 2015

https://developer.ibm.com/bluemix/2015/11/13/docker-insane-scale-on-ibm-power-systems/

Components of Docker

Docker Daemon

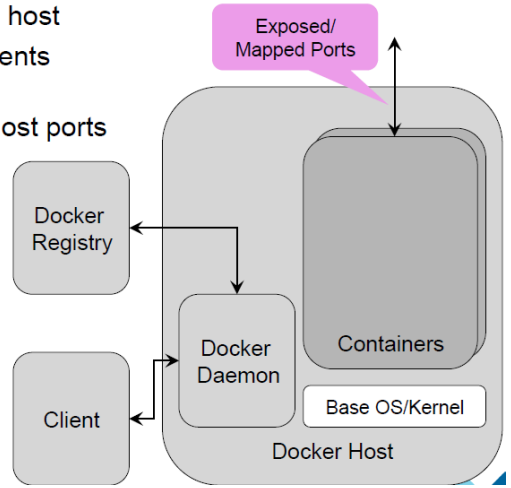
- Manages containers on a host
- Accepts requests from clients
  - REST API
- Maps container ports to host ports
  - E.g. 80 → 3582

Docker Client

- Drives daemon

Docker Registry

- Image DB



Docker Primer on IBM Systems magazine
http://www.ibmsystemsmag.com/Blogs/AIXchange/June-2015/A-Docker-Primer/

Open Source Relational Databases on POWER8

- **MariaDB on POWER8**
  - "Light weight" Relational Database. Limited xml, limited datatypes, etc
  - MariaDB has been created by the founders of MySQL (former SkySQL) as a fork of MySQL and is establishing its growth partly by moving MySQL customers to MariaDB
  - Example POWER8 solution with MariaDB – TURBO LAMP

- **PostgreSQL ('Postgres') on POWER8**
  - **Postgres** an Open Source Relational Database equivalent to Oracle DB
  - **EnterpriseDB** is the company that delivers and supports Postgres via EDB Postgres Advanced Server
  - Used for complex, business logic driven, high performance, transaction based business applications. Data integrity is heavily managed and business logic can be leveraged.
  - Running EDB Postgres Advanced Server on POWER8 offers roughly 2x higher performance over Intel-based systems for OLTP applications, high performance multi-threading, more cache and greater data bandwidth, while little endian mode removes application portability barriers.

POWER8 is designed and optimized for Big Data & Analytics

4X
threads per core vs. x86 (up to 1536 threads per system)

4X
memory bandwidth vs. x86 (up to 16TB of memory)

3X
more cache/socket vs. x86

Processors
flexible, fast execution of analytics algorithms

Memory
large, fast workspace to maximize business insight

Cache
ensure continuous data load for fast responses

Supports growth of users, reports and complex queries

Delivers fast analytics results for real-time decision-making

Handles large volumes of data for better response times

Optimized for a broad range of big data & analytics workloads:

Cognos SPSS

hadoop

Industry Solutions

82X is based on IBM internal tests as of April 17, 2014 comparing IBM DB2 with BLU Acceleration on Power with a comparably tuned competitor row store database server on x86 executing a materially identical 2.6TB BI workload in a controlled laboratory environment. Test measured 60 concurrent user report throughput executing identical Cognos report workloads. Competitor configuration: HP DL380p, 24 cores, 256GB RAM, Competitor row-store database, SuSE Linux 11SP3 (Database) and HP DL380p, 16 cores, 384GB RAM, Cognos 10.2.1.1, SuSE Linux 11SP3 (Cognos). IBM configuration: IBM S824, 24 cores, 256GB RAM, DB2 10.5, AIX 7.1 TL2 (Database) and IBM S822L, 16 of 20 cores activated, 384GB RAM, Cognos 10.2.1.1, SuSE Linux 11SP3 (Cognos). Results may not be typical and will vary based on actual workload, configuration, applications, queries and other variables in a production environment.

Notices and Disclaimers Con't.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. IBM EXPRESSLY DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The provision of the information contained h erein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com, Aspera®, Bluemix, Blueworks Live, CICS, Clearcase, Cognos®, DOORS®, Emptoris®, Enterprise Document Management System™, FASP®, FileNet®, Global Business Services ®, Global Technology Services ®, IBM ExperienceOne™, IBM SmartCloud®, IBM Social Business®, Information on Demand, ILOG, Maximo®, MQIntegrator®, MQSeries®, Netcool®, OMEGAMON, OpenPower, PureAnalytics™, PureApplication®, pureCluster™, PureCoverage®, PureData®, PureExperience®, PureFlex®, pureQuery®, pureScale®, PureSystems®, QRadar®, Rational®, Rhapsody®, Smarter Commerce®, SoDA, SPSS, Sterling Commerce®, StoredIQ, Tealeaf®, Tivoli®, Trusteer®, Unica®, urban{code}®, Watson, WebSphere®, Worklight®, X-Force® and System z® Z/OS, are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

Notices and Disclaimers

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IN NO EVENT SHALL IBM BE LIABLE FOR ANY DAMAGE ARISING FROM THE USE OF THIS INFORMATION, INCLUDING BUT NOT LIMITED TO, LOSS OF DATA, BUSINESS INTERRUPTION, LOSS OF PROFIT OR LOSS OF OPPORTUNITY. IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law

Trademarks and notes

IBM Corporation 2016

- IBM, the IBM logo and ibm.com are registered trademarks, and other company, product, or service names may be trademarks or service marks of International Business Machines Corporation in the United States, other countries, or both. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml
- Other company, product, and service names may be trademarks or service marks of others.
- References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.
- IBM and IBM Credit LLC do not, nor intend to, offer or provide accounting, tax or legal advice to clients. Clients should consult with their own financial, tax and legal advisors. Any tax or accounting treatment decisions made by or on behalf of the client are the sole responsibility of the customer.
- IBM Global Financing offerings are provided through IBM Credit LLC in the United States, IBM Canada Ltd. in Canada, and other IBM subsidiaries and divisions worldwide to qualified commercial and government clients. Rates and availability are based on a client’s credit rating, financing terms, offering type, equipment type and options, and may vary by country. Some offerings are not available in certain countries. Other restrictions may apply. Rates and offerings are subject to change, extension or withdrawal without notice.