CS 516 Project 0

Xinyun Jiang
NetID: xj19
Jan 25 2015

1. **Accumulo:**

   Strengths [1]:

   a. Shared-Nothing (Scalability)
   b. Micro-Batching for Efficient Random I/O
   c. High Concurrency, Low Latency for denormalized Data
   d. Sparse, Flexible Schema supports dynamic and diverse data models
   e. Cell-Level Security promotes sharing

   Weakness [1]:

   a. Sorting induces write multiplication factor
   b. Sparse induces support induces additional storage overhead

2. **Actian Ingres**

   Strengths [8, 29]:

   a. Hardware scalability
   b. Replication and distributed option
   c. Checkpoints (backup) and Rollforward (restores) are simple
   d. High performance and Low Cost
   e. Security
   f. Supports failover
   g. Support partial recovery of inconsistent tables
   h. Can attempt automatic resolution of inconsistencies
   i. Great constraint system
   j. Clustering capabilities
   k. Flexible restore options

   Weakness [8, 29]:

   a. Bad documentation system
   b. Third part support is lacking
   c. Only interface is through Erlang ODBC application

3. **Actian Matrix**

   Strengths [7]:

   a. Fast Analytics
   b. Massive Scalability
   c. Cost Efficient

4. **Actian PSQL**

   Strengths [3,4]:

   a. Large Database Features at an Affordable Price: Cost-effective PSQL offers full security, encryption, management and monitoring tools, plus a host of other features you would expect to see in much more expensive DBMS product
   b. Stability and Reliability: There's no doubt why the Windows desktop accounting market uses PSQL as the database of choice. PSQL's solid reputation is based on rock solid reliability

c. Backwards Compatibility: Actian has over a 30-year history of delivering reliable database upgrades. PSQL delivers years of backward compatibility making moves to new versions a snap

d. SQL extensibility

e. Incorporation of a wide variety of procedural languages

f. Good for embedded applications

g. Easily configurable

Weakness:

a. No default params in PSQL

b. No support bult into PSQL to create web-based procedures

c. Does not work well on windows

5. **Actian Vector**

Strengths (6):

a. Able to run a workload on a server when other databases require a much larger machine

b. Cost Efficient

c. Process data much faster than most other relational databases due to exploits performance features in CPU that most other relational databases do not take advantage of

6. **Actian Versant**

Strengths (5):

a. Does not require implementing any mapping code to persist or retrieve objects

b. Architected for multi-core scalability

c. Minimal administration overhead

d. Efficient direct object storage

e. High speed navigation and retrieval

f. Data integrity

7. **Adabas**

Strengths(2):

a. High performance regardless of platform

b. Cost effective, requiring limited hardware resources because its unique ability to compress data

c. Support a wide variety of data models which can add flexibility and performance

d. Provide direct access to ADABAS data via ODBC compliant desktop tools

8. **Allegrograph**

Strengths (9):

a. Flexible and faster than RDBMSs: Add new predicates without changing any schema, ont-to-many relations are directly encoded without the indirection of tables

b. More efficient data retrieval and extremely efficient range queries

c. Work with multiple programming languages and environments

d. Natural modeling of data

e. Special graph storage structure

f. Efficient graph algorithms

g. Schemaless
h. Support for query languages and operators to query the graph structure

Weakness [10]:

a. Operations on large amounts of data could be slow
b. Could use a lot of space
c. Not widely used in business environment
d. Could describe data inconsistently which reduce usefulness of databse

9. **Altibase HDB**

Strengths [11, 12]:

a. several times faster than the leading in-Memory only databases
b. Highly optimized for in-Memory Computing to avoid latency
c. Short and predictable response time
d. Compatible with all standard interfaces and programming languages
e. Data management efficiency and optimization
f. Reduced point of failure and CPU use
g. Reduced operational costs and human resources

Weakness [13]:

a. Not designed for storing the structured data since based on in-memory database, designed for unstructured data that might cause data loss
b. Product maturity is low: Pre-production versions without any standards

10. **Altibase XDB**

Strengths [14]:

a. Altibase XDB is the world's fastest in-memory only database
b. Highly optimized for in-memory computing to avoid latency
c. Delivers breathtaking data precessing speeds
d. Compatible with all standart interfaces and programming
e. Support for different workloads
f. Support for all common data types

Weakness [13]:

a. Not designed for storing the structured data since based on in-memory database, designed for unstructured data that might cause data loss
b. Product maturity is low: Pre-production versions without any standards

11. **Apache Drill**

Strengths [15]:

a. ability to handle nested data such as JSON, Protobuf, Parquet
b. Highly extensible architecture to provide the widest benefit for multiple communities
c. Advanced cost-based optimizer

12. **Apache Hive**

Strengths [16, 17]:

a. Developer productivity
b. Fits the low level interface requirement of Hadoop perfectly
c. Supports external tables which make it possible to process data without actually storing in HDFS
d. It has a rule based optimizer for optimizing logical plans

e. Supports partitioning of data at the level of tables to improve performance

f. Metastor or Metadata stor is a big plus in the architecture which makes the lookup easy

Weakness (16, 17):

a. The batch nature of Map / Reduce makes Hive perform poorly when need low-latency execution for simple queries

b. Query compiler and optimizer are still not mature

c. No support for update and delete

d. No support for singleton inserts. Data is required to be loaded from a file using LOAD command

e. No access control implementation

f. Correlated sub queries are not supported

13. Apache Storm

Strengths (18):

a. Massive scalability

b. High fault-tolerance

c. Rich array of available sprouts specialized for receiving data from all types of sources

d. Easily interoperate with Hadoop

e. Support for Multilanguage programming

14. Apache Tajo

Strengths (19):

a. Powerful distributed processing architecture

b. Advanced query optimization algorithms and techniques

c. Sophisticated resource management specced for low latency

d. Standard HDFS write overcomes memory constraints without sacrificing speed

e. Easy installation and fast configuration

f. Storage medium agnostic with broad file type compatibility

g. Extensive DBMS system and Hadoop version interoperability

h. Simple Hive migration and Hive metastore sharing

i. Stable, fault-tolerant execution of demanding, real-world queries and ETL jobs

j. Seamless, industry-standard ANSI SQL capabilities

k. Time-saving extensible query re-write engine

l. Compatibility with legacy BI tools via ODBC/JDBC

15. ArangoDB

Strengths (21):

a. Multi-model database - Model your data using flexible combinations of key-value pairs, documents and graphs

b. Sharing - Benefit from ArangoDB's rich feature set even when using distributed collections

c. Convenient querying - AQL is a declarative query language similar to SQL. Other options are REST and querying by example.

d. Extendable through JS - No language zoo: you can use one language from your browser to your backend

e. Space efficiency - ArangoDB takes less space than other document stores due to the idea of “schema-free schemata.”

f. High performance - ArangoDB is fast although it has a rich feature set.

g. Multi-platform - Runs on Windows, Linux, OSX and even Raspberry Pi. Open source and free - ArangoDB is available under the Apache 2 license

Weakness (21):

a. It’s new born baby. Need to wait and watch since there were some interesting active developments going on

16. **AWS DynamoDB**

Strengths (22):

a. Scalable – There is no limit to the amount of data you can store in an DynamoDB table, and the service automatically allocates more storage, as you store more data using the DynameDB write APIs

b. Distributed – DynamoDB scales horizontally and seamlessly scales a single table over hundreds of serves

c. Flexible – DynameDB does not have a fixed schema. Instead, each data item may have a different number of arributes. Multiple data types add richness to the data model

d. Easy administration – Hosted by Amazon and receives Fully managed service from Amazon

e. Cost Effective – a free tier allows more than 40 million database operations/month and pricing is based on throughput

f. Built-in Fault Tolerance – DynamoDB has build-int fault tolerance, automatically and synchronously replicating your data across multiple availability zones in a region for high availability and to help protect your data against individual machine or even facility failures

Weakness (22):

a. Deployable only on AWS

b. Dynamo is an expensive and extremely low latency solution

c. Consistency comes with cost – Read capacity unites are based on strongly consistent read operations, which require more effort and consume twice as many database resources as eventually consistenct reads

d. Queries – Querying data is extremely limited

e. Unable to do complex queries – DynameDB is great for lookups by key, not so good for queries, and abysmal for queries with multiple predicates

f. Joins are impossible – you have to manage complex data relations on your code/cache layer

g. Backup – tedious backup procedure as compared to the slick backup of RDS

h. Latency in availability – When you create a table programmatically, the table does not become available instantly

i. No support for atomic transactions – Each write operation is atomic to an item

j. Additional storage cost for each item – In computing the storage used by the table.
k. No triggers
l. No Foreign Keys

17. **AWS ElastiCache**

Strengths [23]:

a. Easy to setup and use
b. For Java client, it supports a single DNS per cluster

Weakness [23]:

a. It is unreliable – you can lose your data in each one of the following events: a node failure; a scaling event; a process restart; a software upgrade
b. The Jave Client with a single DNS solution uses proxy and adds latency
c. It is expensive, it costs 30% over a plain instance and you need to pay for the entire instance memory even if you use part of it. You also cannot deploy more than one Memcached bucket over the same instance

18. **AWS ElastiCache with Redis**

Strengths [24]:

a. Offers persistence to disk
b. Offers extra data structures
c. Very fast

Weakness [24]:

a. Single-threaded and event driven

19. **AWS EMR**

Strengths [25]:

a. Easy to Use - You can launch an Amazon EMR cluster in minutes. You don't need to worry about node provisioning, cluster setup, Hadoop configuration, or cluster tuning. Amazon EMR takes care of these tasks so you can focus on analysis
b. Elastic - With Amazon EMR, you can provision one, hundreds, or thousands of compute instances to process data at any scale. You can easily increase or decrease the number of instances and you only pay for what you use
c. Low Cost - You can launch a 10-node Hadoop cluster for as little as $0.15 per hour. Because Amazon EMR has native support for Amazon EC2 Spot and Reserved Instances, you can also save 50-80% on the cost of the underlying instances.
d. Reliable - You can spend less time tuning and monitoring your cluster. Amazon EMR has tuned Hadoop for the cloud; it also monitors your cluster —retrying failed tasks and automatically replacing poorly performing instances.
e. Secure - Amazon EMR automatically configures Amazon EC2 firewall settings that control network access to instances, and you can launch clusters in an Amazon Virtual Private Cloud (VPC), a logically isolated network you define.

f. Flexible - You have complete control over your cluster. You have root access to every instance, you can easily install additional applications, and you can customize every cluster. Amazon EMR also supports multiple Hadoop distributions and applications.

20. AWS RDS

Strengths [25]:

a. Less Administration: Given the backup, failover and scaling capabilities of RDS, smaller organizations might postpone the inevitable hiring of an expensive, full time DBA

b. Failover: Multi-Zone RDS provides automatic failover between zones within a few minutes, which is acceptable to most small businesses

c. Scalability: Storage, instance extension

d. Diagnostics, Monitoring: Cloudwatch, graphs, etc

Weakness [25]:

a. Databases don't run themselves: At a certain stage of growth, the performance, storage and management needs of your database will exceed the abilities of your development team; If you choose RDS now to postpone hiring a DBA for 6months, prepare for the eventual necessity of migrating back to a native mysql implementation next year

b. Roadblocks: Advanced diagnostic functionality my be buried with the RDS API, which could hinder an experienced DBA who is comfortable with traditional unix tools such as netstat, df, iostat, tcpdump, lsof, sar, etc.

c. Replication: At this time, region->region replication is NOT supported by Mysql on RDS. Nor is logshipping. zone->zone replication is supported by Multi-AZ internally for inter-zone failover, and read slaves are provided per availability zone to offload read-only queries

d. Application considerations: Major modifications to your app may be costly or time prohibitive. You won't know if your code works until you test it.

21. AWS Redshift

Strengths [26]:

a. Scarily fast

b. Cheap - it's around a 10th of the cost of competitors

c. It's AWS - which works great if you're already on it, like we are at Hailo

d. It's SQL - so you can pretty much cut and carve your data as you like. The fork of postgres only takes a few commands out. We've always been able to do what we want. We can go from thinking of an idea, to writing a query, to answering questions in a couple of minutes. I doubt we'd be able to do this with Hadoop/Hive.

e. Scalability - it's horizontally scalable. Need it to go faster? Just add more nodes.

Weakness [26]:

a. Doesn't enforce uniqueness. This means you'll have to be very diligent about data hygiene. If you're running distributed systems that write to

Redshift, you'll probably have to use some caching system like Redis to check if you've written the data to the database already.

b. Very fast, but not fast enough for most web apps. This means you'll have to pull data into a caching layer, or a vanilla postgres instance if you want to serve web apps with data from Redshift.

c. Loading data - the fastest way to do this to take advantage of Massively Parallel Processing (what a great buzzword) is to load in from S3. Alternatively, can do JDBC inserts, but these can't be parallelised. Ties you into the AWS stack more.

d. Difference from vanilla SQL - I don't necessarily view this as a bad thing, but you'll have to get your head around how to optimise the distribution of your data around the nodes using DISTKEYS, and check that you have appropriate compression on your columns. You also can't alter the datatypes of a column in a table after they've been set, which can be annoying

22. **AWS Simple DB**

Strengths [27]:

a. SimpleDB is not a relational database. Relational databases are too complex and don't scale well. Keeping data access simple is a selling point, not a weakness

b. Low setup costs and pay-as-you-go expansion make it perfect for startups. The price is reasonable given the functionality and the hands off admin

c. Setting up and maintaining a highly available clustered database that is constantly growing is extremely difficult. Building your application on a building block that does all this for you adds a lot of value

d. Setting up a database inside EC2 is a pain. The makes getting basic database functionality trivial. No need to worry about scaling, capacity planning, or partitioning

e. It has a decent query language, which is unusual for this type of data store

f. Data are stored across multiple nodes which supports parallel query execution

g. It's built on Erlang and that's cool

h. You don't need to seek funding to hire a database team and buy hardware

Weakness [27]:

a. No SLA. We don't know how reliable it will be, how fast it will be, or how consistent the performance will be

b. Consistency constraints are relaxed. Reading data immediately after a write may not reflect the latest updates. To programmers used to transactions, this may be surprising, but many people think this is one of the tradeoffs that needs to be made to scale

c. Database is a core competency. If you don't control your database you can't out compete your competition

d. When your database is out of your control you can't guarantee it will work properly. You can't create the proper indexes and other optimizations

e. No join or IN operator. You'll need to do multiple client side calls to simulate joins, which will be slow

f. No stored procedures, referential integrity, and other relational goodies. This is not a professional product

23. **Azure DocumentDB**

Strengths [28]:

a. No Schema: Stored data don't need to have a predefined data

b. Scalable: A database can be spread across multiple machines. In this way we have the ability to scale up the computing power of our database

c. Consistency: All the time, we have 3 copies or our database in the same data center. When we scale up we don't scale only the CU power but also we duplicate our database. This is needed because we don't want our storage to become our bottleneck

Weakness [28]:

a. Transactions cross collections: cannot have a transaction on multiple collections

b. No versioning support

24. **BerkeleyDB**

Strengths [29]:

a. Good for term storage

b. Fast lookups with memory cache

Weakness [29]:

a. Does not support constraints

b. Application involvement needed for replication

c. Limited in size capabilities

d. Lots of things left to the using application

25. **BigCache**

Strengths [30]:

a. In Memory is ultra fast (of course)

b. From local disk is also extremely fast (although much slower than in memory, 10x)

c. Memcached is amazingly fast too, especially considering it's over the network, apparently 2 ms per object (although much slower than in memory or local disk, 20x vs disk)

d. S3 and SimpleDB were similar when retreiving directly by key/id, but both were much slower than any other solution at about 25ms per object, 10x vs memcached)

Weakness [30]:

a. only 10 different objects to see if S3 did any caching on recently retrieved items

b. the following scenario is an alternative that gives you the ultra fast memory/disk combo for your local cache and the slower S3 as the global shared cach

26. **BigMemory**

Strengths [31]:

a. Cross Language Client Support: access to BigMemory data from multiple client platforms
b. Multi-data center support: WAN data replication to keep data in sync across regions while offering support for disaster recovery
c. Support for extended hybrid storage: leverages SSD & Flash technologies in addition to DRAM

27. **BitYota**

Strengths [32]:

a. Source data is not frequently changing
b. Transforms are static
c. Economic constraints prohibit storing raw data and only aggregated data is stored
d. Referential integrity is critical and ETL tools already provide that
e. Reload if needed from the ETL tool's staging store and not the data source
f. Data sources generate large data volumes every minute/hour, and analysis is required on data every hour or several times in the day
g. There are multiple sources with different characteristics in terms of volume, multiple data formats and speed of new data
h. It is critical to discovery value of business metrics over raw data to find new metrics for new initiatives such as marketing promotions, usage of new product features, etc
i. Faster availability of data for analytics

28. **Cassandra**

Strengths [33, 34]:

a. Highly scalable and highly available with no single point of failure
b. NoSQL column family implementation
c. Very high write throughput and good read throughput
d. SQL-like query language and support search through secondary indexes
e. Tunable consistency and support for replication
f. Flexible
g. Broader range of systems in production
h. Richer API which supports values with a dynamic column structure. The columns can evolve independently, meaning that you can update one column without reading the whole structure
i. Optimized for writes
j. Configurable consistency level

Weakness [34]:

a. File format is still in development, changes to the internal structure are likely to happen. Due to the flexibility it provides, the file format is more complex and harder to reason with, especially in terms of performance
b. Requires Clock Synch
c. Reads are more disk-intensive than competitors
d. Doesn't support client conflict resolution, so the latest update always wins

29. **Cloudera**

Strengths [36]:

a. CDH is open source; you have access to the source code and can inspect it for debugging purposes and make modifications as required
b. CDH can be run on a number of public or private clouds using an open source framework, Whirr, so you're not tied to a single cloud provider with CDH, you can move your cluster to dedicated hardware with little disruption when the economics make sense. Most non-trivial applications will benefit from this move.
c. CDH packages a number of open source projects that are not included with EMR: Sqoop, Flume, HBase, Oozie, ZooKeeper, Avro, and Hue. You have access to the complete platform composed of data collection, storage, and processing tools.
d. CDH packages a number of critical bug fixes and features and the most recent stable releases, so you're usually using a more stable and feature-rich product. You can purchase support and management tools for CDH via Cloudera Enterprise.
e. CDH uses the open source Oozie framework for workflow management. EMR implemented a proprietary "job flow" system before major Hadoop users standardized on Oozie for workload management.
f. CDH uses the open source Hue framework for its user interface. If you require new features from your web interface, you can easily implement them using the Hue SDK.
g. CDH includes a number of integrations with other software components of the data management stack, including Talend, Informatica, Netezza, Teradata, Greenplum, Microstrategy, and others
h. CDH has been designed and deployed in common Linux environments and you can use standard tools to debug your programs

30. **CortexDB**

Strengths [37]:

a. Flexibility to change the database schema as required by business departments and software development the system adapts to the process rather than the other way round
b. Extremely high database performance on standard hardware
c. Rapid and agile application development across innovative data services without programming
d. Simple modeling of complex structures
e. Change requests on the fly, enabling self-service usage by business unites

31. **CouchDB**

Strengths [38]:

a. Master-Master Replication – Because of the append only style of commits
b. Reliability of the actual data store backing the DB
c. Mobile platform support. CouchDB actually has installs for iOS and Android

Weakness [38]:

a. HTT REST JSON interaction only, no binary protocol

32. **DataStax Enterprise**

    Strengths [39, 40]:

    a. Internal and external authentication for database access
    b. Granular permission control for database changes
    c. Transparent encryption for enhanced file protection
    d. Advanced data auditing capabilities
    e. Client to node encryption that safeguards data in transit

33. **Datomic**

    Strengths [41]:

    a. Datalog is very enjoyable to work with in this s-expression form. Much, much better than SQL. It's by far my favorite part
    b. The API is intuitive for the most part. When trying something new I can often guess my way through most of i

    Weakness [41]:

    a. The exceptions thrown by Datomic are rubbish. You'll get NullPointerExceptions and IndexOutOfBoundsExceptions when not passing any nils/nulls or performing any indexing. Sometimes it just throws bare Exceptions, even though the source throw had more information (i.e. Lucene's ParseException)
    b. The full-text search is nice to have but Lucene's functionality isn't exposed enough to really take advantage of it. For example, search results aren't ranked
    c. Documentation and examples focus mostly on the Java API. I understand the business reason for doing this, but I'm using it entirely from Clojure. The documentation is generally pretty thin overall.

34. **Drizzle**

    Strengths [42, 43]:

    a. Special defense stat which means it can survive most super-effective special attacks
    b. Better performance on modern machines which means can run more than 4 hardware threads at once
    c. Doesn’t need rain dance
    d. Can run a physical or special set
    e. Works effectively on all rain teams

    Weakness [42]:

    a. Not the greatest movepool
    b. Not great defense stat means it can easily OHKO’d by a physical move

35. **Exasol**

    Strengths [44]:

    a. Fewer collisions with parallel transactions
    b. A transactions view of the database is fixed throughout its lifespan
    c. Every modification can be rolled back

    Weakness [44]:

    a. Some changes cannot be rolled back and immediately impact other users or processes
    b. Higher probability of transaction conflicts

36. **FathomDB**

Strengths [45]:

a. FathomDB leverages MySQL's scaling capabilities across multiple machines and automated backup can make it a reliable Cloud storage. Since standard MySQL is offered as an engine, you do not have to change your application at all, and there is no lock-in or new API to learn.

b. There also should be mentioned FathomDB's risk reduction, high quality level, and time-to-market improvement when it comes to putting a database in the cloud with the ability to scale.

c. FathomDB's performance analysis tools can facilitate the high-level DBA tasks and make them easier.

d. FathomDB allows complex database operations to happen according to user-configurable maintenance schedules; further, that disk storage space is unlimited on all plans and higher memory and CPU shares give you a higher performing database.

e. Finally, FathomDB enables companies to store their database on Amazon or Rackspace servers.

Weakness [45]:

a. there are possible challenges that a company may face when starting database implementation in the Cloud. Particularly, it is the changing of a database instance's size, which may sometimes involve restarting MySQL and even moving to a new serve

37. **Firebird**

Strengths [46]:

a. In testing with smaller databases weve found in most cases Firebird is only slightly slower than SQL server with all hardware being equal. However much depends on the exact operation being performed, how much data is in your database, and your hardware

b. Free with a very solid history in production use

c. Requires little to no administration, once set up it just keeps working

d. Has a very active community of developers

e. Database can be backed up while running with included utilities

f. Can be run on a non Windows computer such as Linux saving the cost of an operating system for a computer being used solely as a database server

g. Just as reliable as Microsoft SQL server in our experience, we have had no issues with either platform

Weakness [46]:

a. Slower under heavy loads: If for example, you were to display a huge list of thousands of records of data in AyaNova rather than filtering it by a date range as is recommended we have found Microsoft SQL server to be noticeably faster. On the other hand when doing normal work in AyaNova entering work orders, printing them etc there is no significant difference

b. It's open source free software, for some that is a bonus, others may find that worrisome. We're not normally great proponents of free software for obvious reasons, but we did look at both commercial and open source alternatives to offer with MS-SQL server and it was technically the best match for the requirements of AyaNova

c. Slower performance has been noted if hyperthreading is used, or dual processors in some cases. Recommended to compare with or without to see what works best for you

38. **FoundationDB**

Strengths [47]:

a. We already use it in several other services (we have experience of it)
b. Better licensing terms (by far) All features free outside production, and production licensing terms essentially means it is currently free for us to use
c. Full ACID compliance
d. Has both Consistency and Availability during Partitioning (assuming it isn't a catastrophic failure)
e. Built in transaction retries
f. Support from FDB team is excellent
g. Excellent read performance both single reads and range reads are only marginally slower
h. Transaction isolation level is serializable
i. Simple to scale horizontally

Weakness [47]:

a. Designed as a Key/Value pair store rather than Document Store
b. Weak .NET support (only 3rd party .NET client wrapper on top of C) with less .NET documentation/support. NET not considered first class citizen of FoundationDB
c. Constraints on deployment - cannot be deployed in IIS, must be self-hosted - loss of IIS specific features such as graceful request handling, automatic app pool recycles/mem management
d. Cannot run in multiple AppDomains on same process
e. Like Raven, also an Alpha product, though we have less concerns about the composition of the development team
f. Does not generate IDs, so a separate "ID Generation Service" would be required (or switch entire platform to GUIDs with the resulting data migrations
g. No nice 'management' interface - you need to roll your own admin tool

39. **Galera**

Strengths [48]:

a. A high availability solution with synchronous replication, failover and resynchronization
b. No loss of data
c. All servers have up-to-date data (no slave lag)
d. Read scalability
e. 'Pretty good' write scalability
f. High availability across data centers

Weakness [48]:

a. It supports only InnoDB
b. With increasing number of writeable masters, the transaction rollback rate may increase, especially if there is write contention on the same dataset. This increases transaction latency.
c. It is possible for a slow/overloaded master node to affect performance of the Galera Replication cluster, therefore it is recommended to have uniform servers across the cluster.

40. **RavenDB**

Strengths [47]:

a. Excellent .NET client API with extensibility points, providing easy
b. Developer learning curve
c. Designed as Document Store
d. Provides Index/Map/Reduce
e. Stores natively as JSON
f. Good Read Performance
g. Fits excellently into integration testing, due to in memory db option designed for testing
h. Automatically generates Ids for records
i. Well presented web based management studio
j. Raven Server has a good console where you can see requests/response times and what indexes are used to resolve queries

Weakness [47]:

a. Cannot test replication, sharding or authenticated access functions without purchasing licenses and licenses are needed for anything other than development (UAT would need licenses)
b. Some concerns over the dependence of the RavenDB project on one key developer
c. Some concerns about the robustness of the testing of the product and its unproven track record in enterprise solutions (posts like this are easy to find
d. Yet another product for devops to support
e. Need to understand the "eventually consistent" model well when designing solutions

41. **GenieDB**

Strengths [49]:

a. Performance: GenieDB provides instantaneous data consistency, so your application always reads the most current data
b. Ease of Use: Providing a well-known SQL interface and automatic node provisioning means spending less time retraining your technical staff and allowing more time for them to solve business critical issues.
c. Resilience: With an automated self-healing algorithm, GenieDB recovers from faults with little effort.
d. Cost Efficiency: You can realize the benefits of GenieDB on your existing hardware without any application changes.

42. **Google App Engine Datastore**

Strengths [49]:

a. No need to buy servers or server space – no maintenance
b. Makes solving the problem of scaling easier
c. Free up to a certain level of consumed resources

Weakness [49]:

a. Java applications cannot create new threads
b. Developers have read-only access to the filesystem on App Engine

43. **Google Cloud Datastore**

Strengths [50]:

a. More sophisticated key-valueish store with built-in transaction support and entity hierarchy
b. Don't need to worry about the capacity of the system
c. Automatically scales to data size and access patterns

Weakness [50]:

a. Documention and language support is not very uniform

44. **Handlersocket**

Strenghts [50, 51]:

a. Stable
b. High Performance
c. Support a lot of query types
d. No duplicate cache
e. Flexible
f. Included in Percona Server

Weakness [50, 51]:

a. No security – Without username and password
b. Do not update MySQL Query Cache – Fixed in recent commits
c. No "Insert…on duplicated key update…"
d. Do not support auto increment column while inserting – Fixed in recent commits

45. **Hbase**

Strenghts [52]:

a. High volumn data random access
b. Scale-out with commodity machine
c. Fault-tolerance
d. Free license

Weakness [52]:

a. No security control
b. Data loss risk
c. Redesign data schema
d. Lacking of aggregate function
e. No commercial support now

46. **Heroku Postgres**

Strenghts [53]:

a. Not having to worry about managing your infrastructure
b. Ease of scale by moving your Dynos slider upEasy integration with 3rd party services such as SendGrid, MongoDB, Redis etc...

c. Savings - most sites don't need maximum horse power all the time so you can achieve massive savings by using a PaaS like Heroku - pay for what you use. Some businesses fit here quite nicely

Weakness [54]:

a. Rarely-accessed free apps take a few seconds (sometimes longer) to start up — upgrade to a paid plan if you need your app to respond quickly even the first time it's accessed in a while
b. Ephemeral file system (files may be written to the app's "tmp/" directory, but can be discarded by the system at any time) — use a cloud storage service instead
c. Not optimal for large static assets — use a CDN instead
d. Runs only on AWS in the US East region (meaning there's no regional redundancy, so an outage is an outage, and traffic from Japan will experience higher latency rates than traffic from the States).

47. **Hortonworks**

Strengths [55]:

a. A spin-off Yahoo product, so it's been tested in the marketplace
b. Lots of documentation and support available from the knowledgebase community
c. The company is continuously working with Yahoo to develop its future products
d. Scalable to meet the demands of specific projects
e. Offers variations and expanded product offerings from partnerships with a number of specialized companies.

Weakness [55]:

a. Product is similar in nature to Cloudera and provides similar features

48. **HP Vertica**

Strengths [55]:

a. Cheap - the cheapest of all three appliances we own
b. Columnar - applies some of the new generation practices and provides full columnar structure
c. Compression - highly compressed database and it automatically chooses the compression algorithm for data.
d. Automating some of the system maintenance activities like purging logically deleted data etc
e. Some of patented analytical functions specifically targeted at Log parsing for web based click stream etc data analysis.
f. Relatively faster loading of data as the data will be sitting in WOS (RAM) first and slowly moves to disk.

Weakness [55]:

a. Product is not as mature as other appliances. With no GUI tools and very limited workload management options
b. Heavily depends on projections to deliver performance for different scenarios. With multiple analysis view points the number of projections for a table will increase (literally maintained as multiple copies of same data) nullifying the benefits of compression

c. Not good for general purpose users with many number of concurrent sessions querying the database. If any user query reaches the RAM availability level in the box, then the performance drop is inevitable

d. Not so good with new generation reporting tools like Microstrategy as tuning options for their query pattern are null

e. Loading into WOS depends on the RAM available. System license audits keep running

f. Indexes and general database practices for optimized querying are not properly utilized. If any query needs tuning options, are limited to duplicating the data in projections.

g. Most of the times segmentation does not help as when data needs group by the data will be redistributed from multiple nodes and that might be a bottleneck

h. No configuration options for limiting database size, resource usage, etc

i. Finally not so great for bigger organizations looking for the single point of source of data warehouse. Can be used for small data marts with small set of questions to be answered with lots of storage at your disposal.

49. HypergraphDB

Strenghts [56]:

a. Implicit graph indexing of everything

b. Given any value, or any property, traverse the graph to find structures that contain it

c. Pattern matching: given a blueprint, fill in the blanks

Weakness [56]:

a. Waste of storage if not many duplicates

b. Slower writes to ensure uniqueness

c. Slow reads on deep trees, with few duplicates and frequent cache misses

50. IBM BigInsights

Strenghts [57]:

a. Offers superior monitoring that can provide a better understanding of data distribution and processing – essential for achieving increased performance

b. A free version is offered, which includes everything except management tools which are only offered in its M5 series products.

c. Excellent technical support and vast quantities of documentation available

Weakness [57]:

a. New to the marketplace so has a limited reputation

b. An expensive solution for small/medium size organizations

c. 24×7 support is only available on the paid version of the application

d. Requires an enormous amount of disk space to install (25GB), compared to similar products

51. IBM DB2

Strengths [58]:

a. Very high degree of availability

b. Increased scalability

c. No need to change applications

Weakness (58):

a. High cost
b. Specialized hardware required
c. Specailized skill needed to implement

52. **InfiniteGraph**

Strengths (59):

a. Extremely fast for connected data
b. Scales out, typically
c. Easy to query
d. Simple data model

Weakness (59):

a. May not support distribution or sharding
b. Requires conceptual shilft, a different way of thinking

53. **InfluxDB**

Strengths (60):

a. Really fast to display
b. Can store all knds of events
c. Grafana is as simple as kibana

Weakness (60):

a. Harder to setup than an ELK stack
b. Need to define all continuous queries to be able to use them after
c. Cluster support not ready yet

54. **Infobright**

Strengths (61):

a. Fastest Bulkloader
b. Very good compression ratio

Weakness (61):

a. Bulkloader swallows errors as a trade off for faster loading
b. Per TB pricing structure

55. **LevelDB**

Strengths (62):

a. richness and a more flexible RAM requirement
b. Since LevelDB is a sorted-order store it can support Riak's secondary indiced
c. Since it does not use a full in-memory keymap it can support a higher number of keys per GB of RAM
d. It's a brand new store, so may still have some bugs left in it.
e. LevelDB indexes need to 'compact' occasionally which MIGHT stall out new writes.
f. You can't just delete the index and rebuild it from the data files like you can with KahaDB
g. KahaDB handles disk corruption much more gracefully, recovering what it can and discarding corrupted records.

Weakness (62):

a. Append mostly disk access patterns improve perf on rotational disk

b. Fewer disk syncs than KahaDB
c. Fewer index entries need to be inserted per message stored
d. Fewer index lookups needed to load a message from disk into memory
e. Uses Snappy compression to reduce the on disk size of index entries
f. Optional Snappy compression of data logs.
g. A send to a composite destination only stores the message on disk once.
h. Faster and more frequent data file GCs.

56. **Loggly**

Strengths [62]:

Whereas Sumo Logic has a strong enterprise and security focus, Loggly is geared more towards helping devops find and fix operational problems. This makes it very developer-friendly. Things like creating custom performance and devops dashboards are super-easy to do. Pricing is also transparent, which makes start of use easier

Weakness [62]:

Don't expect Loggly to scale into a full blown infrastructure, security or analytics solution. If you need forensics or infrastructure monitoring you're in the wrong place. This is a tools mainly for devops to parse data coming from your app servers. Anything beyond that you'll have to build yourself

57. **Lotus Notes**

Strengths [62]:

a. includes diary, calendar, email, to-do list
b. Ability to create databases for workflow management (like public folders in Outlook)
c. Mobile users may access e-mails remotely via VPN or webmail (like Windows remote access)
d. It's very secure (a plus over the more vulnerable Microsoft environment)
e. Domino tech support is reliable

Weakness [62]:

a. Notes is a clunky looking interface similar to Outlook 97
b. Navigation is awkward (think Netscape browser from the 90′ s)
c. Any type of workgroup creation, e-mail addition, or changes requires a technical administrative support person. This is the biggest negative. Out of the box, Lotus needs a lot of manipulation.
d. Domino and Lotus Notes must be constantly updated and backed up for optimal processing (not as crucial with Outlook and Exchange Server).

58. **Lucene/Solr**

Strengths [63]:

a. Bigger Community
b. No split breain situations
c. Response in csv xml json
d. Better language support out of the box
e. Grouping
f. Split shards
g. Spellchecking

Weakness [63]:

a. need a schema
b. No routing magic

59. LucidDB

Strengths [64]:

a. Warehouse Labels
b. Hot backups
c. Very small footprint
d. Strong Mondrian and Pentaho integration
e. Extendible

Weakness [64]:

a. Small company

60. MariaDB

Strengths [65]:

a. XtraDB
b. More advanced join and disk access optimizations
c. Better upgrade scripts than MySQL itself

Weakness [65]:

a. No support for multi-threaded replication
b. No optimizations for ORDER by nidxol LIMIT N
c. New version incompatible with MySQL

61. Memcached

Strengths [66]:

a. Low complexity
b. Simple to configure
c. Few command macros == simple to master
d. Atomic increment and decrement
e. Simple to cluster -- uses a hashing algorithm at the client to find keys in a cluster
f. Runs like a rock -- memcached requires a nuclear strike to fall over
g. Can withstand a member dying
h. Many years in production
i. Every programming language has a memcached library.

Weakness [66]:

a. Doesn’t do anything besides be an in-memory key/value store
b. Caches sharded by client do not scale across AWS zones
c. Unbalanced memcached clusters require a full system restart
d. Adding a member to the pool requires reconfiguring and rebooting the client
e. Seriously doesn't do anything besides be an in-memory key/value store

62. MemSQL

Strengths [67]:

a. Blazing fast
b. You can use all the MySQL client tools out there. Also because it uses MySQL, troubleshooting might not be an issue because it isn’t a new product.

Weakness (67):

a. Huge cache folder! After restoring the database (15Mb) MemSQL creates a folder in the plancache folder which is 5.5GB. So performance comes with a price I think.
b. Unable to create users. When creating the database, I tried to add a new user but this isn't supported. Maybe this is because I'm using the Developer Edition.

63. **Microsoft HD Insight**

Strengths (68):

a. Scalable
b. Easy to integrate
c. Out of the box solution
d. Integration with a lot of external libraries and systems

Weakness (68):

a. Start/Stop features is not yet supported
b. Elastic-scale is not yet supported

64. **MongoDB**

Strengths (69):

a. Schema-less. If you have a flexible schema, this is ideal for a document store like MongoDB. This is difficult to implement in a performant manner in RDBMS
b. Ease of scale-out. Scale reads by using replica sets. Scale writes by using sharding (auto balancing). Just fire up another machine and away you go. Adding more machines = adding more RAM over which to distribute your working set.
c. Cost. Depends on which RDBMS of course, but MongoDB is free and can run on Linux, ideal for running on cheaper commodity kit.
d. You can choose what level of consistency you want depending on the value of the data (e.g. faster performance = fire and forget inserts to MongoDB, slower performance = wait til insert has been replicated to multiple nodes before returning)

Weakness (69):

a. Data size in MongoDB is typically higher due to e.g. each document has field names stored it
b. Less flexibity with querying (e.g. no JOINs)
c. No support for transactions - certain atomic operations are supported, at a single document level
d. At the moment Map/Reduce (e.g. to do aggregations/data analysis) is OK, but not blisteringly fast. So if that's required, something like Hadoop may need to be added into the mix
e. Less up to date information available/fast evolving product

65. **MySQL**

Strengths (69):

a. Somewhat lighter-weight
b. Has commercial support if you want it, like Redhat Enterprise Linux
c. Widespread, plenty of apps that target it

d. Blazing fast performance, particularly when predominantly SELECT queries, especially on MyISAM tables (this is somewhat anecdotal though, and will be highly dependent on the environment and workload)
e. "Enterprise-grade" clustering and some degree of replication is available - this is a very different product from the MySQL Server that most people will be familiar with, so it's not just a drop-in replacement

Weakness [69]:

a. MyISAM tables don't support transactions and have little idea about ACID properties. If a MyISAM table breaks (frequent), you need to run myisamchk. This will take multiple aeons on a large table
b. MyISAM tables only do table-level locking, so concurrency is bollocks
c. This noted, MyISAM is the default if you don't specify an engine at table-creation time, so it's easy for an inexperienced developer to go down the MyISAM path
d. Last I checked, subquery performance in MySQL was pretty awful
e. Triggers are perverse

66. **MySQL Cluster**

Strengths [70]:

a. True HA by design – Fast recovery
b. Supports X-actions
c. Relational Model
d. In-memory architecture – high performance
e. Disk storage for non-indexed data

Weakness [70]:

a. Some semantic limitations on fields
b. Size constraints as wel as hardware limits
c. Higher cost/byte
d. Requires reasonable data partitioning
e. Higher complexity

67. **Neo4J**

Strengths [71]:

a. No O/R impedance mismatch
b. Can easily evolve schemas
c. Can represent semi-structured info
d. Can represent graphs/networks

Weakness [71]:

a. Lacks in tool and framework support
b. No other implementations – potential lock in
c. No support for ad-hoc queries

68. **ObjectRocket**

Strengths [72]:

a. High Availability - Every instance comes automatically configured with a secondary node and free backups. Should a primary node fail, the system seamlessly fails over to the secondary node
b. Fully Managed Service with 24x7 Expert Support - ObjectRocket for Redis is backed by Redis specialists 24x7x365 who proactively monitor and fix

infrastructure issues, assist with data migrations and fine-tune Redis configurations. These specialists can also provide architecture advice and data best practices.

c. Simplified Operations - One click provisioning, standardized configurations and advanced administration capabilities make it easy for developers to focus on building their product while leaving the tasks of maintaining, scaling and supporting Redis to the experts.
d. High Performance - Optimized specifically for Redis, ObjectRocket for Redis uses a containerized approach that eliminates the noisy neighbor issues associated with traditional virtualization. Like all of the company's offerings, ObjectRocket for Redis was built from the ground up for performance, from the network to the OS to Redis itself. ObjectRocket for Redis was designed holistically to provide the best possible performance.
e. High Bandwidth - By directly peering with multiple networks, it doesn't matter where customers host their application -- ObjectRocket for Redis is only a few milliseconds away. So customers can host their applications where they want and leave the database work to ObjectRocket

69. Oracle Coherence

Strengths (73):

a. Elastic. Just add nodes. Auto-discovery. Auto-load-balancing. No data loss. No interruption. Every time you add a node, you get more data capacity and more throughput.
b. Use both RAM and flash. Transparently. Easily handle 10s or even 100s of gigabytes per Coherence node (e.g. up to a TB or more per physical server).
c. Automatic high availability (HA). Kill a process, no data loss. Kill a server, no data loss.
d. Datacenter continuous availability (CA). Kill a data center, no data loss.
e. RESTful APIs available from any language. Native APIs and client libraries for C/C++, C#, .NET and Java.
f. In addition to simple key-value (K/V) caching, also support queries (including some SQL), parallel queries, indexes (including custom indexes), a rich eventing model (for event-driven systems like exchanges), transactions (including MVCC), parallel execution of both scalar (EntryProcessor) and aggregate (ParallelAwareAggregator) functions, cache triggers, etc.
g. Easy to integrate with a database via read-through, read-ahead, write-through and write-behind caching. Automatically refreshes just the changed data when changes occur to the database (leveraging Oracle GoldenGate technology).

Weakness (73):

a. As of Coherence 12.1.2, the cache itself is not persistent
b. Expensive

70. Oracle Exadata

Strengths (74):

a. remote real-time monitor of system

b. proactive maintenance and alerting
c. online patching, almost no down time

Weakness [74]:

a. Expensive

71. Oracle NoSQL

Srengths [75]:

a. Non-Relational means table-less: NoSQL databases are non-relational, hence, very different from SQL databases. This means they are easier to manage and they provide a higher level of flexibility with newer data models
b. Mostly Open Source: The open source nature of NoSQL databases means it is now possible to use large data hungry applications at a lower price point. This is because open source technologies are free of cost.
c. Easier scalability through support for Map Reduce: NoSQL database experts often use elastic scalability as a major selling point of NoSQL. NoSQL databases are designed to function on full throttle even with low-cost hardware
d. No need to develop a detailed database model: The non-relational nature of a NoSQL database allows database architects to quickly create a database without needing to develop a detailed (fine-grained) database model. This saves a lot of development time.

Weakness [75]:

a. Lack of community: NoSQL is relatively new and lacks a healthy community. Meanwhile, MySQL has a seasoned community willing to help you start on your journey into the world of databases
b. Lack of reporting tools: A major problem with NoSQL databases is the lack of reporting tools for analysis and performance testing. However, with MySQL, you can find a wide array of reporting tools to help you prove your application's validity
c. Doesn't conform to ACID properties: Relational database systems function on the ACID paradigm (Atomicity, Consistency, Isolation, Durability). NoSQL databases don't
d. Lack of standardization: In order for NoSQL to grow, it needs a standard query language like SQL. This is a major issue highlighted by researchers at Microsoft, who claim that NoSQL's lack of standardization can cause a problem during migration. Besides this, standardization is important for the database industry to unify itself in future.

72. PostgresSQL

Strengths [76]:

a. Very feature rich: GIS add-on functionality; Flexible full-text search (brought into mainline in 8.2); Multiple replication options to suit your environment and requirements; Powerful server-side procedural languages are available, and can be extended (PL/pgSQL is installed by default, but others like Perl, Python, Ruby, TCL, etc are available); Writing your own extensions is pretty easy

b. Uses multi-version concurrency control, so concurrent performance rocks
c. Fully ACID compliant
d. Commercial support through multiple third-parties is available
e. Well-documented
f. Strong access-control framework

Weakness [76]:

a. Less-mature replication software

Strengths [76]:

Weakness [76]:

73. Oracle NoSQL

Strengths [40]:

a. Commercial-Grade Software and Support: Oracle NoSQL Database is fast, scalable, reliable, and easy to install and configure. Backed by Oracle Support, Oracle NoSQL Database also offers customers a single point of contact for trouble-shooting issues and maintaining continuous availability of critical database instances
b. Scalable throughput and bounded latency: Oracle NoSQL Database scales almost linearly, providing customers with reliable, high performance under a wide range of workloads and use cases
c. Simple Programming and Operational Model: Simple major + sub key and value data paradigm and more relaxed support of ACID transactions provides extreme flexibility of programming for large datasets in horizontally scaled environments
d. Easy Management: With a Web-based console and accessible APIs, Oracle NoSQL Database is easy to manage. The offering automatically manages and monitors topology, load balancing, performance, events and alerts

Weakness [13]:

a. Few firms offring support for NoSQL
b. NoSQL databases offer few facilities for ad-hoc query and analysis
c. Extremely difficult to find competent NoSQL experts

74. Impala

Strengths [20]:

a. Lighting speed and promise near real time adhoc query processing
b. The computation happen in memory, that reduce enormous amount of latency and Disk IO

Weakness [20]:

a. No fault tolerance for running queries
b. If a query failed on a node, the query has to be reissued, it can not resume from where it fails

75. Presto

Strengths [20]:

a. Lighting fast and promise near real time interactive querying
b. Used extensively in Facebook, so it is proven and stable
c. Open Source and there is a strong momentum behind it ever since it’s been open sourced

d. It is also using Distributed query processing engine. So it eliminates all the latency and DiskIO issues with traditional MapDeduce

e. Well documented, Perhaps this is the first open source software from Facebook that got a dedicated website

Weakness [20]:

b. it's new born baby. Need to wait and watch since there were some interesting active developments going on

c. As of now support only Hive managed tables

d. Though the website claim one can query hbase also, the feature still under development

e. Still no UDF support yet, This is the most requested feature to be added

76. Teradata Aster

Strengths [35]:

a. Relatively new but seems to be the first one to bring Map-reduce (hadoop etc) world's best features into SQL

b. It is in hands of Teradata now and is getting I/O level control of user resource and work load usage etc and is getting connectivity to Teradata for seam less integration of queries through unity application

c. Brings flexibility and ease of use with in-built map reduce functionality and features for relatively new buzz in DW space - BIG DATA problem

d. Uses Massively parallel processing capabilities of hardware in intelligent fashion

e. Established and arguably best in the data warehouse appliances

f. Performance degradation is minimal even with exponential raise of number of users and work load (best for thousands of users)

g. With the advent of Teradata 14, Teradata closed in on all new generation technology approaches.

h. Hybrid storage with option for storing data in both row based and column based storage for same table (Teradata 14's main feature).

i. Block level compression using specialized hardware along with normal compression for columnar with facility to users to write their own compression algorithms.

j. Best toolset with constantly updated GUIs and tools.

k. Wide variety of options for configuring and controlling the workload management with TASM and related toolset (I can't write down all the available tools' names).

l. highly tuned custom-made fully supported massively parallel hardware+software architecture.

m. Best loading performance along with querying performance even for active data warehousing.

n. I cant list out all the advantages in services and support availability you have in choosing the best in business with established tool set over a relatively newbies and cheap technologies.

o. Finally, If you are looking for stable and highly performing single point of data warehouse for your full organization needs (and especially if your organization is growing exponentially) Teradata is the best fit. In data warehouse environment in last 5-10 years there were many newcomers with exciting features, but Teradata is still the one to beat.

Weakness [35]:

a. Costly for small organizations
b. Some of the developers find it difficult to understand the system level functionality of Teradata, but if used properly, Teradata will beat any one in the performance
c. Dependency on primary index to achieve the top gear in performance, but looks like Teradata 14 is trying to avoid that by not allowing to have columnar on primary index tables
d. Was behind in introducing new technologies but they made a mark with rapid introduction of new features with Teradata 13.10, 14 versions
e. Not many to count, If I am missing any, please add (if anybody observed some other cons for the same)

Reference:

1. Data Modeling and Indexing for Apache Accumulo, Sqrrl Data, Inc. Confidential and Proprietary
2. Adabas VS DB2: An Evaluative Study by Butler Bloor Group (from www.storrconsulting.com)
3. Official Website of Actian: http://www.actian.com/products/operational-databases/psql/
4. Advantages and disadvantages of PSQL: http://forums.devshed.com/postgresql-help-21/advantages-disadvantages-postgresql-106805.html
5. Official Website of Actian Versant: http://www.actian.com/products/operational-databases/versant/
6. Actian Vector: The Revolutionary High Performance Analytics Database http://www.odbms.org/wp-content/uploads/2014/08/WP01-ActianVector-0424.pdf
7. Actian Analytics Database – Matrix: http://wwwcdn.actian.com/wp-content/uploads/2014/06/DS06-ActianMatrix.pdf
8. Ingres Database: Performance and availability for today's mission critical environments http://www.slideshare.net/Actian/ingres-database-ps
9. AllegroGraph Introduction: http://franz.com/agraph/support/documentation/current/agraph-introduction.html
10. Your Database is so retro: old data, new databases, by Michael Sherman, August 14, 2014: http://www.texasenterprise.utexas.edu/2014/08/14/innovation/your-database-so-retro-old-data-new-databases

11. Altibase HDB – Hybrid DB: http://altibase.com/in-memory-database-hybrid-products/hdbtm-hybrid-dbms/
12. What are the benefits of altibase HDB's hybrid architecture? http://altibase.com/faq/benefits-altibase-hdbs-hybrid-architecture/
13. Altibase vs NOSQL Database Comparison: http://altibase.com/in-memory-database-comparisons/nosql/#toggle-id-5
14. Altibase XDB (Extreme In-Memory Database): http://altibase.com/in-memory-database-hybrid-products/xdb/
15. Responding to the Need for SQL on Big data: Apache Drill https://www.mapr.com/blog/responding-to-the-need-for-sql-on-big-data-apache-drill#.VMR1dorF8mc
16. What are the advantages of Hive over SQL http://www.quora.com/What-are-the-advantages-of-Hive-over-SQL
17. Apache Hive Review http://www.gise.cse.iitb.ac.in/wiki/images/2/26/Hive.pdf
18. Real-time business interlligence is going mainstream thanks inpart to the Storm and Spark open source projects. Here's how to Choose between them http://www.infoworld.com/article/2854894/application-development/spark-and-storm-for-real-time-computation.html
19. All of the advantages of the Hadoop ecosystem http://gruter.com/technology/tajo/
20. Overview of Big data, Hadoop and Microsoft BI http://www.slidesearch.org/slide/overview-of-big-data-hadoop-and-microsoft-bi-version1
21. World of the NoSQP databases http://leopard.in.ua/2013/11/08/nosql-world/
22. Dynamo DB http://www.slideshare.net/saniyakhalsa/dynamo-db-pros-and-cons
23. What are the advantages and disadvantages of using AWS Elasticache over self-hosted Memcached http://www.quora.com/What-are-the-advantages-and-disadvantages-of-using-AWS-Elasticache-over-self-hosted-Memcached
24. What are pros and cons of using AWS ElastiCache Redis versus Memcached http://www.quora.com/What-are-pros-and-cons-of-using-AWS-ElastiCache-Redis-versus-Memcached
25. AWS: RDS: MYSQL Limitations http://oldtimeynerd.net/2012/11/29/aws-rds-mysql-limitations-draft/
26. What are the pros and cons of using Amazon Redshift http://www.quora.com/What-are-the-pros-and-cons-of-using-Amazon-Redshift
27. The Current Pros and Cons List For SimpleDB http://highscalability.com/current-pros-and-cons-list-simpledb
28. Vunvulea Radu Tech Wall (Programming Issues) http://vunvulearadu.blogspot.com/2014/12/documentdb-day-5-of-31.html

29. Evaluation of Massively Scalable Database Management Systems for Erlang, Fifth ACM Sigplan Erlang Workshop Portland, Sept 16th, 2006 by Emil Hellman
30. BigCache Overview https://code.google.com/p/bigcache/
31. BigMemory MAX http://terracotta.org/products/bigmemorymax
32. Bityota Official Website http://www.bityota.com/elt-not-etl/
33. Consider the Apache Cassandra database http://www.ibm.com/developerworks/library/os-apache-cassandra/
34. Choosing a key-value storage system (Cassandra vs. Voldemort) http://blog.medallia.com/product-and-innovation/choosing-a-key-value-storage-system-cassandra-vs-voldemort/
35. Vertica vs Aster Data vs Greenplum vs Netezza vs Teradata http://stackoverflow.com/questions/4528934/vertica-vs-aster-data-vs-greenplum-vs-netezza-vs-teradata
36. What are the pros and cons of running cloudera's distribution for hadoop vs Amazon Elastic MapReduce Service http://nosql.mypopescu.com/post/20582276970/what-are-the-pros-and-cons-of-running-clouderas
37. Positioning CortexDB in Hadoop Data Architectures – Extending the Data Warehouse and beyond http://www.odbms.org/wp-content/uploads/2014/10/Cortex_ResearchNote_2014.pdf
38. NoSQL databases pros and cons http://www.slideshare.net/fabiofumarola1/nosql-databases-pros-and-cons
39. DataStax Exterprise: Key Advantages http://www.datastax.com/what-we-offer/products-services/datastax-enterprise/advantages
40. Oracle Big Data ApplianceOracle Announces Availability of Oracle NoSQL Database http://www.oracle.com/us/corporate/press/519708
41. Why Datomic? http://www.reddit.com/r/Clojure/comments/18ty8j/why_datomic/
42. Castform vs Drizzle http://pokemondb.net/pokebase/227964/castform-vs-drizzle-politoed
43. What are the benefits of Drizzle? http://www.quora.com/What-are-the-benefits-of-Drizzle
44. Transactions https://www.exasol.com/portal/display/SOL/Transactions
45. Cloud Tools: Creating Efficient Cloud Storage with FathomDB http://blog.altoros.com/creating-efficient-cloud-storage-with-fathomdb.html
46. SQL vs Firebird http://www.ayanova.com/WhichDatabaseShouldIUse.htm
47. Devers Development http://pauldevenney.blogspot.com/2014/07/pros-and-cons-comparing-ravendb-and.html
48. What are the pros and cons of Galera? http://support.severalnines.com/entries/20567341-What-are-the-pros-and-cons-of-MySQL-Galera-Replication-

49. Pros and Cons of Google App Engine http://stackoverflow.com/questions/1306279/pros-cons-of-google-app-engine
50. HandlerSocket – A NoSQL plugin for MySQL http://www.slideshare.net/jnlin/handlersocket-a-nosql-plugin-for-mysql
51. MySQL + HandlerSocket = NoSQL https://prezi.com/5ws8tsf7cxfx/mysqlhandlersocketnosql/
52. HBase's pros and cons http://comments.gmane.org/gmane.comp.java.hadoop.hbase.user/10579
53. What are the pros and cons of using either Heroku or Amazon EC2 for launching a ROR web application http://www.quora.com/What-are-the-pros-and-cons-of-using-either-Heroku-or-Amazon-EC2-for-launching-a-ROR-web-application
54. What are the potential downsides of using Heroku http://www.quora.com/What-are-the-potential-downsides-of-using-Heroku
55. Vertica vs Aster Data vs Greenplum vs Netezza vs Teradata http://stackoverflow.com/questions/4528934/vertica-vs-aster-data-vs-greenplum-vs-netezza-vs-teradata
56. HyperGraphDB: Model and Applications http://www.cse.unsw.edu.au/~iwgdm/2012/Slides/HyperGraphDB.pdf
57. Hosted Hadoop-based Applications Review http://blog.monitis.com/2012/04/05/hadoop-based-applications-review/
58. Implementing high availability with DB2 http://www.ibm.com/developerworks/data/library/techarticle/dm-0807wright/
59. An Introduction to Graph Databases http://www.slideshare.net/infinitegraph/an-introduction-to-graph-databases
60. Which tool to use to make marketing dashboard http://odolbeau.fr/blog/which-tool-to-use-to-make-marketing-dashboards.html
61. My very dodgy col store database benchmarking http://pentahomusings.blogspot.com/2010/12/my-very-dodgy-col-store-database.html
62. The 7 log Management Tools Jave Developers Should Know http://blog.takipi.com/the-7-log-management-tools-you-need-to-know/
63. Why we chose Solr instead of ElasticSearch http://www.ymc.ch/en/why-we-chose-solr-4-0-instead-of-elasticsearch
64. My very dodgy col store database benchmarking http://pentahomusings.blogspot.com/2010/12/my-very-dodgy-col-store-database.html

65. Oracle MySQL vs Percona Server vs MariaDB
    http://www.sobstel.org/blog/mysql-like-databases-comparison/
66. Memcached vs Redis?
    http://www.bigdatalittlegeek.com/blog/2014/3/25/memcached-vs-redis
67. Taking MemSQL for a spin
    http://manuel.manuelles.nl/blog/2012/06/20/taking-memsql-for-a-spin/
68. Vunvulea Radu Tech Wall
    http://vunvulearadu.blogspot.com/2014/12/azure-hdinsight-day-25-of-31.html
69. MySL vs PostgreSQL comparison
    http://www.anchor.com.au/hosting/dedicated/mysql_vs_postgres#head-a550fa51208d6e20cd2a7db1372ac3c4308f3ac1
70. Paypal Big Data and MySQL Cluster
    http://www.slideshare.net/matkeep/paypal-mysql-cluster
71. Neo4j – or why graph dbs kick ass
    http://www.slideshare.net/emileifrem/neo4j-presentation-at-qcon-sf-2008-presentation
72. ObjectRocket by Rackspace Adds High Performance Redis Data Store
    http://openstack.sys-con.com/node/3148355
73. What are some good distributed caching systems besides Memcached
    http://www.quora.com/What-are-some-good-distributed-caching-systems-besides-Memcached
74. What are the pros and cons of Oracle Exadata
    http://www.quora.com/What-are-the-pros-and-cons-of-Oracle-Exadata
75. The key differences between MySQL and NoSQL DBs
    http://blog.monitor.us/2013/05/cc-in-review-the-key-differences-between-mysql-and-nosql-dbs/
76. MySQL vs. PostgreSQL comparison
    http://www.anchor.com.au/hosting/dedicated/mysql_vs_postgres#head-e1573b42fddb2f6189d92926583b54b2a4f3ef51