The Internet of Things: Your application will be access by wide range of devices. Like mobile, computer, cloud.

The business world is shifting towards Digital Economy powered by the Internet and other technologies the cloud, mobile, social media and big data to support large customers and to be loyal to them. The technologies need to have following character.

* Support large numbers of concurrent users (in millions).
* Deliver highly responsive experiences to a globally distributed base of users.
* Be always available – no downtime.
* Handle semi and unstructured data.
* Rapidly adapt to changing requirements with frequent updates and new features.

|  |  |
| --- | --- |
| **Digital Economy Trend** | **Requirements** |
| **#1: More customers are going online** | * Scaling to support thousands if not millions of users * Meeting UX requirements with consistent high performance * Maintaining availability 24 hours a day, 7 days a week |
| **#2: The Internet is connecting everything** | * Supporting many different things with different data structures * Supporting hardware/software updates, generating different data * Supporting continuous streams of real-time data |
| **#3: Big Data is getting bigger** | * Storing customer generated semi-structured/unstructured data * Storing different types of data from different sources, together * Storing data generated by thousands/millions of customers/things |
| **#4: Applications are moving to the cloud** | * Scaling on demand to support more customers, store more data * Operating applications on a global scale - customers worldwide * Minimizing infrastructure costs, achieving a faster time to market |
| **#5: The world has gone mobile** | * Creating “offline first” apps - network connection not required * Synchronizing mobile data with remote databases in the cloud * Supporting multiple mobile platforms with a single backend |

What is NoSql?

NoSql: it’s a kind of distributed storage system, basically it means we can spread data on multiple machines which are call nodes and all of them collectively behave as a single database.

Why does it requires? Because of performance when your data is low then performance is not a consideration or bottleneck because every things loads as quickly or just work fine. But when your data grows means Big Data + Big Users then performance get impacted/ hampered.

Performance: greater/higher read and write throughput.

Scalability: When data grows big and number of concurrent users grow big then term comes as scalability.

**Why relational databases fall short**

Relational databases were born in the era of mainframes and business applications – long before the Internet, the cloud, big data, mobile and now, the Digital Economy. These databases were engineered to run on a single server – the bigger, the better. The only way to increase the capacity of these databases was to upgrade the servers – processors, memory, and storage – to scale up.

Cloud computing: Most of the cloud commuting provide commodity hardware.

Commodity hardware means you should have minimum configuration like below

16- 32 GB ram, at least 1 TB HDD, couple of quad (4)-processor and 1GB Ethernet card and the price should be around 1k to 2k.

So cloud provide commodity hardware if you want to scale you scale it horizontally.

Scalability: we have to choice to scale application

* 1. Scale up – Vertically.

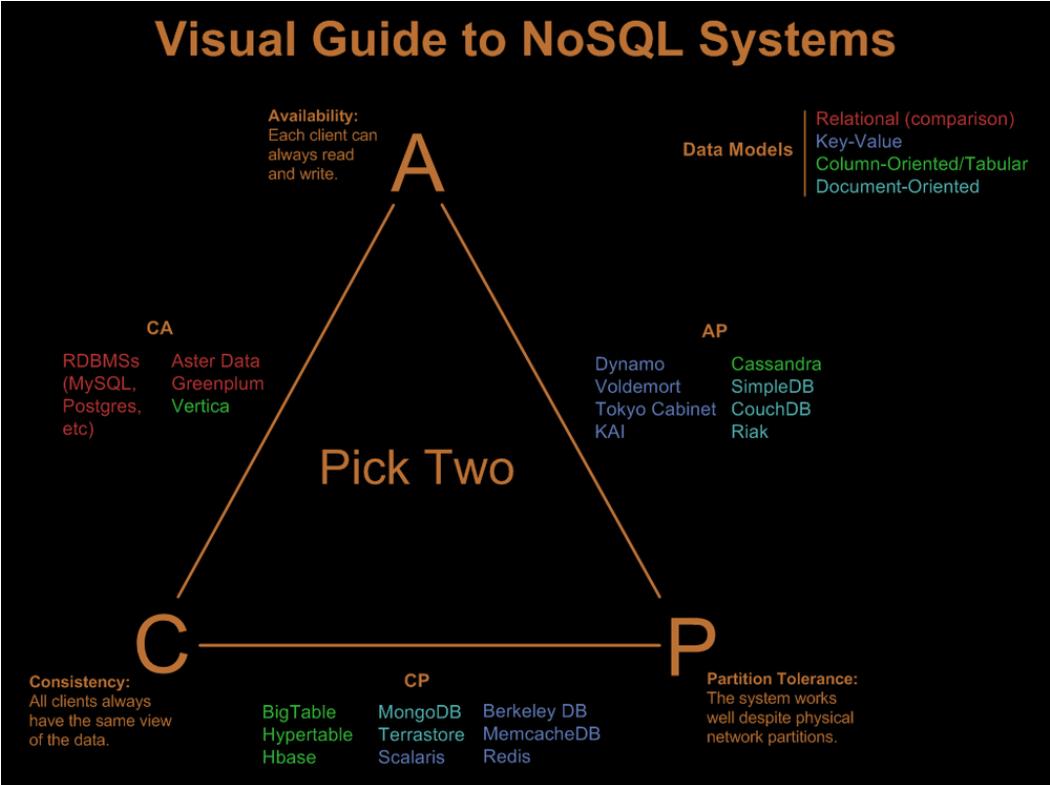
It’s mostly happened with relational database as your data/load grows you purchase better server and put your database on that.

* 1. Scale out- Horizontally.

NoSql takes different approach of horizontally scaling by using the basic traits (mechanism)

* Sharding: spread data across multiple machines and called as node.
* Replication: You replicate data for fail over situation.

CAP theorem: All distributed/big data system driven by CAP theorem.



Consistency: All of our clients see the same data at any given moment.

Availability: If one of our node goes down even though our application should still be able to function through the replication servers.

Partition Tolerance: It’s related to network partition. Suppose you have two nodes. Node A and B. If both of them are running but communication between them is broken. It means sync between A and B doesn’t happens. Then we have two choices.

Take an application down and say I am maintaining strong consistency. Or continue functioning as it is and say your user whatever you see right now is stale data but as soon as we resolve our network issue we and sync everything up and be consistence.

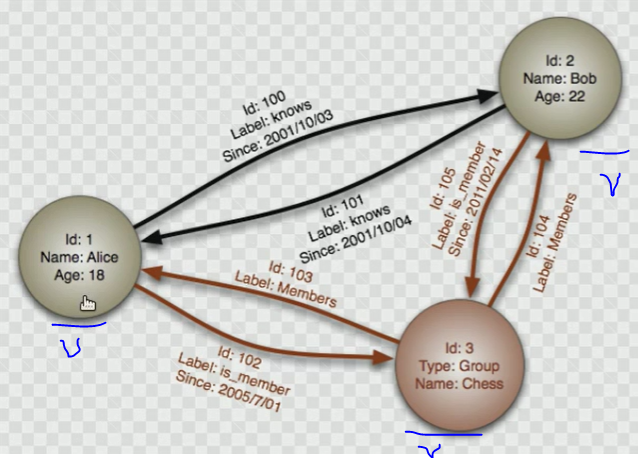
For production environment Linux file system.

Extensible file system (XFS) or B-Tree file system is much better in written and reading data form the disk which help to attend better performance.

What are possible types of NoSql data types?

NoSql categories in four different types

* 1. Key Values-(redis, Memcached, Riak). Key values pair data types such as in .net HashTable or in java HashMap. In this we have unique key for each object and value can be the complex data. These are highly performed well because you look by key.
  2. Document- (json MongoDB, CouchDB, Couchbase): document oriented databases store data in the json format.
  3. Columnar-(Cassandra, Hbase): In relation database we store data in row fashion but in columnar database we store data in column. These make perfect for running aggregates.
  4. Graph – (Edges and vertices). The graph databases are somewhat different than other database it consists of concept of edges and vertices



In this database every entity is treated as vertex (e.g. Alice, Bob or Group). In these all traits of entity e.g. Age, Id and salary are store in vertex and relationship are stored as edges e.g. know, is member, member

Alice knows bobs since 2001/10/03.

This kind of database help you where your business domain structure is too complex and this is mainly use recommendation. Like amazon recommendation for product.