

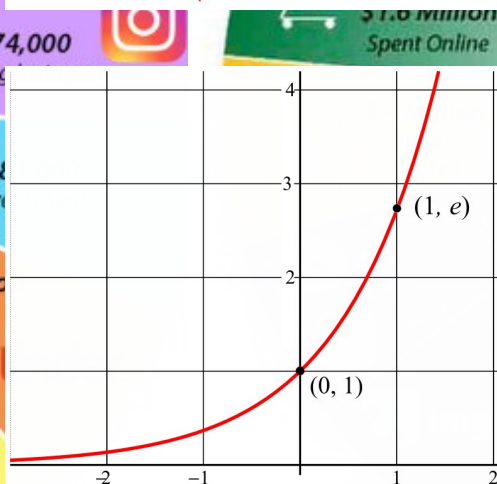
Introduction

Mag. Thomas Griesmayer

2018 This Is What Happens In An Internet Minute → 2021 This Is What Happens In An Internet Minute



NEW



Created By:
[@LoriLewis](#)
[@OfficiallyChad](#)

Created By:
[@LoriLewis](#)
[@OfficiallyChad](#)

NoSQL

- The term NoSQL stands for "Not Only SQL" and refers to a class of database systems that do not follow the traditional relational model (RDBMS).
- NoSQL databases are particularly optimized for handling:
 - large
 - distributed and
 - unstructured datasets.
- Characteristics:
 - Schema-free or flexible schema
 - Horizontal scalability
 - High performance
 - Different data models - requirements

Main Types

Document-oriented Databases (e.g., MongoDB, CouchDB)

- Store data as JSON or BSON documents.
- Flexible schema for different data types.

Key-Value Stores (e.g., Redis, Riak)

- Simple storage of key-value pairs.
- Particularly fast for caching and session data.

Column-family Stores (e.g., Apache Cassandra, HBase)

- Store data in columns instead of rows.
- Suitable for analytical applications with large datasets.

Graph Databases (e.g., Neo4j, ArangoDB)

- Designed for modeling and querying relationships between data.
- Useful for social networks, recommendation systems, etc.

Three Vs

Volume – The sheer amount of data generated and stored.

Big Data involves massive datasets, often measured in terabytes, petabytes, or even exabytes. Examples: Social media posts, IoT sensor data, transaction records, etc.

Velocity – The speed at which data is generated and processed.

Data streams continuously from sources like social media, stock markets, and real-time sensors. Requires fast processing technologies (e.g., Apache Kafka, Spark) to analyze data in real time.

Variety – The different types and formats of data.

Includes structured data (relational databases), semi-structured data (JSON, XML), and unstructured data (videos, images, text). Handling diverse data formats requires flexible storage and processing solutions.

Veracity – Data quality and reliability.

Value – The usefulness of the data for decision-making.

Big Data analysis

Social Media Sentiment Analysis

Example: Analyzing Twitter posts to determine public opinion on a new product.

How it works:

Collect tweets using hashtags (#NewPhone).

Use Natural Language Processing (NLP) to classify tweets as positive, negative, or neutral.

Businesses use this to adjust marketing strategies.

Customer Purchase Prediction

Example: Amazon recommends products based on past purchases.

How it works:

Collect data on what customers buy.

Use machine learning to find patterns.

Suggest similar or frequently bought-together items.

Big Data analysis

Fraud Detection in Banking

Example: A bank detects unusual credit card transactions.

How it works:

Track transaction history (location, amount, time).

If a sudden large withdrawal happens in a different country, flag it as suspicious.

The bank alerts the customer or blocks the transaction.

Traffic Flow Optimization

Example: Google Maps predicts traffic congestion.

How it works:

Collect real-time GPS data from millions of users.

Identify slow-moving areas.

Suggest faster routes.

Big Data analysis

Healthcare Disease Prediction

Example: Predicting disease outbreaks using patient records.

How it works:

Analyze hospital records, online searches (e.g., "flu symptoms"), and wearable device data.

Identify trends and predict disease outbreaks in certain regions.

Authorities prepare vaccines and resources in advance.

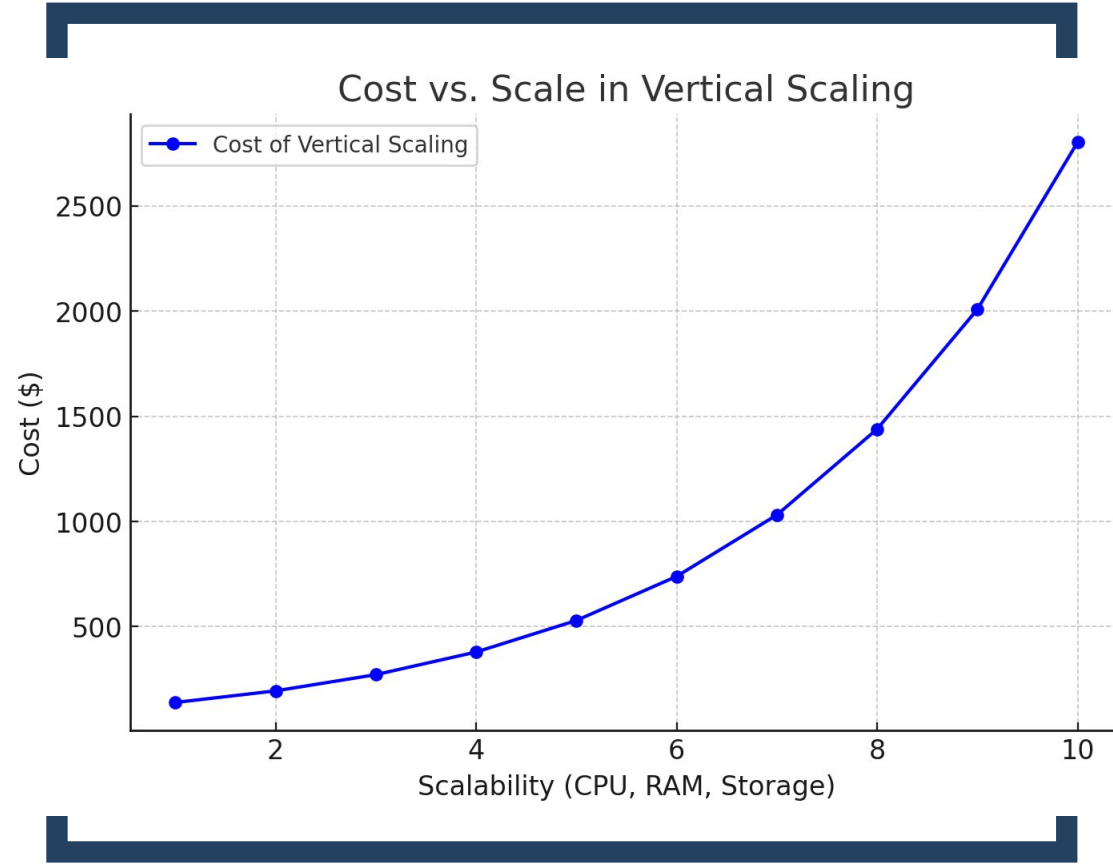
Vertical Scaling - Scaling Up

Advantages:

- Easy to implement (no complex distributed system)
- No software changes required
- Good for database-heavy applications like Oracle, MySQL

Disadvantages:

- Physical limits (there is a maximum capacity)
- More expensive than horizontal scaling (high-end hardware is costly)
- No automatic failover – if the server crashes, the system goes offline



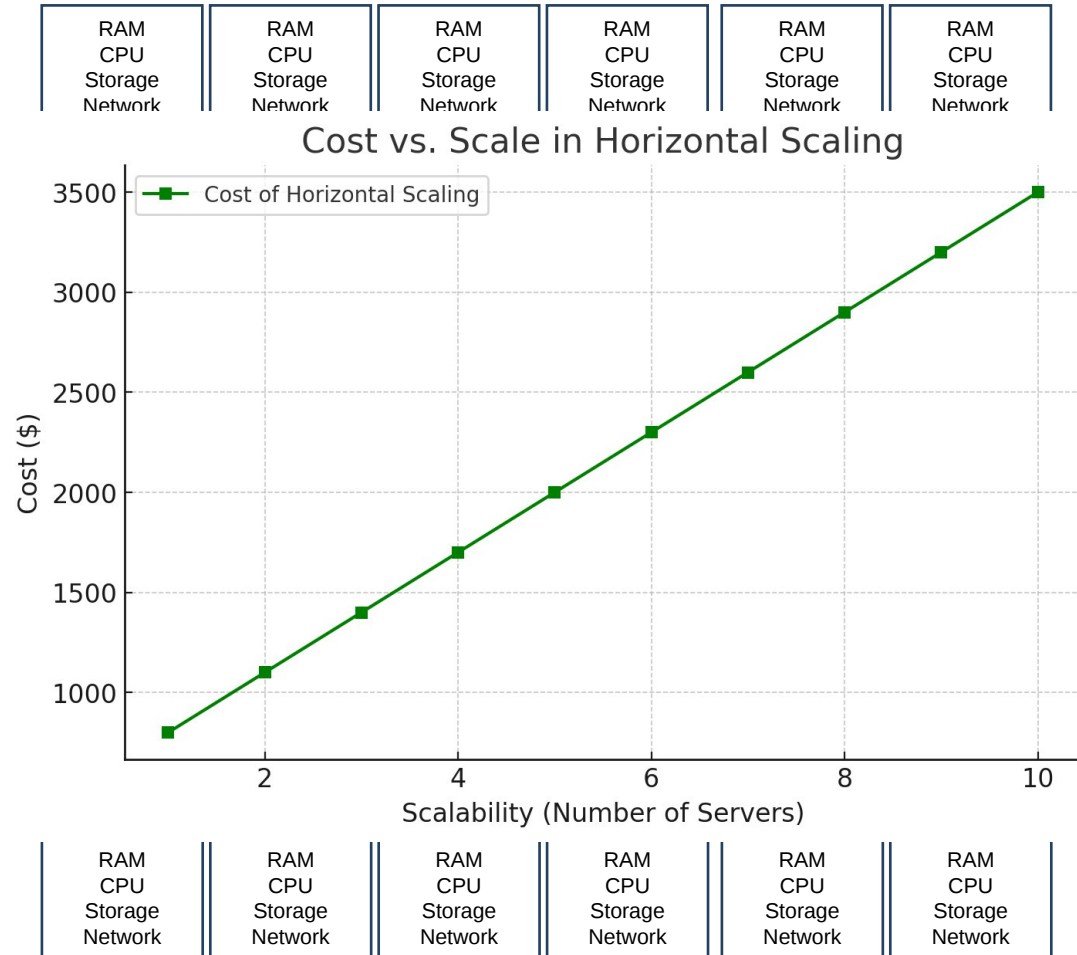
Horizontal Scaling - Scaling Out

Twitter (X Corp)
operates several global data centers in locations like Atlanta, Sacramento, Portland, Europe and Asian.

Initially relied heavily on vertical scaling (powerful servers with more RAM/CPU). Later moved towards horizontal scaling (more distributed servers for handling high traffic).

Twitter generates terabytes of data per day (tweets, images, videos).

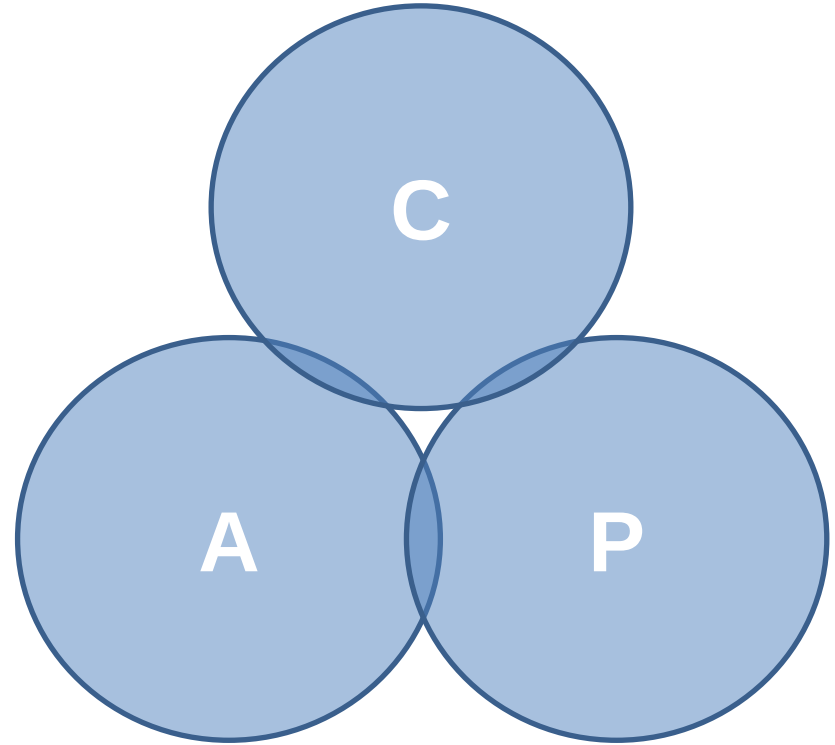
Twitter ensures 99.9% uptime using replicated servers.



CAP Theorem (Brewer's Theorem)

SELECT, DELETE, INSERT and UPDATE

- Consistency (C) – Every read gets the most recent write (all nodes show the same data at the same time).
- Availability (A) – Every request gets a response, even if some nodes fail.
- Partition Tolerance (P) – The system continues to work despite network failures (partitions).



CAP Theorem: Bank Transactions

Imagine a bank with two data centers:

Berlin			Munich		
1	Susi	430€	1	Susi	430€
5	Max	115€	5	Max	115€
9	Alex	10€	9	Alex	10€

Network partition

The bank must ensure that a customer does not withdraw more money than they have.
The account balance is synchronized.

C&P – Consistency & Partition Tolerance

Berlin			Munich		
1	Susi	430€	1	Susi	430€
5	Max	65€	5	Max	115€
9	Alex	10€	9	Alex	10€

Example: Max withdraws €50 in Berlin.

The server in Berlin updates the data and synchronizes with Munich.

But: If the connection between Berlin and Munich fails (partition), the system blocks all further transactions to maintain consistency.

Balance remains correct, but customers must wait.

Lower availability – The customer cannot access their money in Munich until the connection is restored.

A&P – Availability & Partition Tolerance

Berlin		
1	Susi	430€
5	Max	65€
9	Alex	10€

Munich		
1	Susi	430€
5	Max	45€
9	Alex	10€

Example: Max withdraws €50 in Berlin.
At the same time, Max withdraw €70 in Munich.

Due to network issues (partition), the two servers cannot synchronize.
Result: Both transactions are allowed, and the account incorrectly shows 65€ and 45€, because both servers thought money was available.
High availability – The customer can withdraw money anywhere.
Data may be inconsistent (temporary double spending).

C&A – Consistency & Availability

Berlin

1	Susi	430€
5	Max	115€
9	Alex	10€

Munich

1	Susi	430€
5	Max	115€
9	Alex	10€

The customer withdraws money, and the system is always synchronized.

But: If the network fails, no customer can access their account.

Data is always correct & available (as long as there are no network failures).

Only works if all servers are always reachable → not realistic for large distributed systems.

ACID vs. BASE



Das ACID-Prinzip von relationalen Datenbanken stellt, in Zusammenhang mit Transaktionen, folgendes sicher:

- **Atomicity (Atomarität):**

Eine Transaktion wird nach dem „Alles-oder-nichts“ Prinzip entweder vollständig oder gar nicht ausgeführt. Wird eine atomare Transaktion abgebrochen, ist das System in einem unveränderten Zustand.

- **Consistency (Konsistenz)**

In einem konsistenten Datenbanksystem führt eine Folge von Datenbankoperationen wieder zu einem konsistenten Zustand

ACID vs. BASE



Das ACID-Prinzip von relationalen Datenbanken stellt, in Zusammenhang mit Transaktionen, folgendes sicher:

- **Isolation**

Parallel ausgeführte Transaktionen beeinflussen sich nicht gegenseitig

- **Durability (Dauerhaftigkeit)**

Die Auswirkungen von Transaktionen müssen dauerhaft im System gespeichert werden – insbesondere bei Systemabstürzen.



ACID vs. BASE



NoSQL – Datenbanken ? BASE

- Basically Available
- Soft State
- Eventual Consistency


Das Akronym „BASE“ ist gekünstelt um einen einprägsamen Gegensatz zu „ACID“ darzustellen (engl. Lauge / Säure). Für „Basically Available“ und „Soft State“ gibt es keine präzise Definition.

Vielmehr steht „BASE“ für ein Design Prinzip, das das Konzept der „absoluten Konsistenz“ aufgibt, statt dessen die Verfügbarkeit des Systems erhöht und dadurch zwischenzeitlich in ***einem etwas undefinierten Zustand*** sein kann.

SQL vs. NoSQL (1) - Allgemein



SQL Datenbanken

- sind mächtig und können für die meisten Datenbankprobleme herangezogen werden
- Aufgrund langjähriger (Weiter-)Entwicklung wird die Arbeit mit SQL-Datenbanken immer einfacher
- Vielzahl an Anbieter, Schulungen, Support, Manpower etc...
- Einheitliche, sehr mächtige Abfragesprache  SQL

SQL vs. NoSQL (2) - Skalierung



NoSQL Datenbanken

- Horizontale Skalierung ? DIE Stärke von NoSQL Datenbanken
 - Bei SQL Datenbanken ist horizontale Skalierung zwar möglich, jedoch nur mit wesentlich höherem Verwaltungsaufwand und nur begrenzt (ab einem gewissen Punkt führen die Vorteile von SQL ins Negative)
 - Aufgrund des einfachen Schemas (bzw. keines Schemas) sind NoSQL Datenbanken für hohe Skalierbarkeit geschaffen.
 - Die Skalierbarkeit bleibt auch bei sehr hohen Datenvolumina erhalten

SQL vs. NoSQL (3) - Performance



NoSQL Datenbanken

- Sind performanter als Relationale Datenbanken ☐ besonders bei hohen Datenvolumen bemerkbar
- Relationale Datenbanken stoßen bei sehr großen Datenvolumina an ihre Grenzen (ist ein Grund weshalb NoSQL Datenbanken erfunden wurden)
- Unabhängig ob Lese- oder Schreibzugriff, NoSQL Datenbanken sind den SQL Datenbanken voraus

SQL vs. NoSQL (4) - Konsistenz



SQL Datenbanken

- Aufgrund der ACID-Eigenschaften besitzen Relationale Datenbanken eine bessere Konsistenz ? eigentlich eine absolute Konsistenz
- NoSQL Ansatz „eventually consistency“ („schlussendlich konsistent“) ? es wird nicht garantiert, dass nach einem Update immer derselbe Wert zurückgegeben wird. Eine Reihe von Bedingungen müssen erfüllt sein, bis alle denselben Wert bekommen.
- Grundsätzlich muss der Nutzer entscheiden ob Performance im Vordergrund steht, oder ob eine gute Konsistenz notwendig ist.

SQL vs. NoSQL (5) - Beziehungen



- Beziehungen sind eine der schwierigsten und ressourcen-intensivsten Dinge, die man mit Hilfe von SQL-Datenbanken erstellen kann.
- Das Speichern von vernetzten Informationen oder zusammenhängenden Objekten kann man bei SQL-Datenbanken sehr schwer realisieren (bzw. nur mit sehr hohem Aufwand).
- Graphen-Datenbanken sind NoSQL Datenbanken, welche darauf spezialisiert sind, vernetzte Informationen zu speichern.

SQL vs. NoSQL (6) - Fazit



- Aufgrund des zunehmenden Datenvolumens, der Notwendigkeit von steigender Performance und der zunehmenden Wichtigkeit, Beziehungen in Datenbanken zu definieren, werden NoSQL Datenbanken immer beliebter.
- Sie werden SQL Datenbanken jedoch **nicht** Ablösen. Beide Systeme werden parallel existieren und **einander ergänzen**.
- Je nach Verwendungszweck muss zwischen den beiden Systemen gewählt werden.