

NoSQL

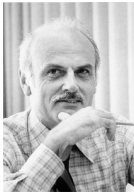
Gerardo Rossel



2017

Motivación y Orígenes

- 1975 ACM SIGMOD(Special Interest Group on Management of Data). **Debate: Ted Codd y Charles Bachman**



- Nada tan complicado como la propuesta de DBTG puede ser la forma correcta de administrar datos
- Las consultas orientadas a conjuntos son dificultosas.
- El modelo de red no tiene fundamentos formales en teoría matemática



- El modelo relacional es demasiado matemático
- No se puede construir una implementación eficiente del modelo relacional
- Las aplicaciones necesitan procesar los datos de a un registro por vez.

Contexto Histórico

- Finales de los 1970-Actualidad: Modelo Relacional
 - 1974-1978 en IBM se desarrolló *System R* (Se transformó posteriormente en SQL/DS y finalmente en DB2)
 - 1973 to 1977 Michael Stonebraker y Eugene Wong (U.C. Berkely) desarrollaron INGRES DBMS
 - 1976 (Marzo) Peter Chen: *The Entity-Relationship Model: Toward a Unified View of Data*. ACM Trans. on Database Systems
- 1990: Orientación a Objetos y Objeto-Relacional. ORMs.
- 2005-Actualidad: Nueva generación. NoSQL y NewSQL

RDBMS

- Soporte de independencia física y lógica
- Fuerte fundamento teórico
- Soporte de transacciones ACID¹
- SQL: Lenguaje de consulta estándar

RDBMS

Al hablar de Sistemas de Gestión de Bases de Datos Relacionales en general se está refiriendo a bases de datos que implementan el *modelo relacional*, soportan transacciones *ACID* y usan *SQL* como lenguaje de consulta y manipulación de datos.

¹Jim Gray. *The Transaction Concept: Virtues and Limitations*. Proceedings of 7th International Conference on Very Large Databases, Sept.1981.

En los 90

- The Object-Oriented Database System Manifesto (1989)²
- ObjectStore (en 2015 adquirida por Ignite)
- Third-Generation Database System manifesto (1990)³
 - *A counterweight to the spreading object-oriented enthusiasm is needed.*
 - *Object-oriented databases is not the solution to the challenges found in using 2nd generation DBMSs*
- The Third Manifesto 1998 ⁴

² Atkinson, DeWitt, Maier, Bancilhon, Dittrich, y Zdonik

³ The Committee for Advanced DBMS Function - Stonebraker, Lawrence A. Rowe y otros

⁴ H. Darwen, C. Date Foundation for Object/Relational Database Systems: The Third Manifesto

SGBD

En los últimos 35 años...

- Nuevas Arquitecturas de aplicaciones.
- Nuevos Paradigmas de programación.
- Nuevas herramientas para desarrollo de Software.

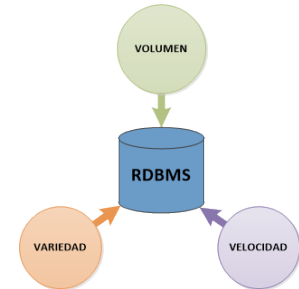
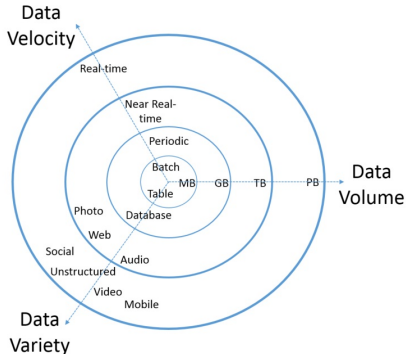
Pero..

- Los RDBMS y el modelo relacional siguen siendo los predominantes
- En el período 1995 a 2005 no se introdujeron nuevas BD significativas

Big Data

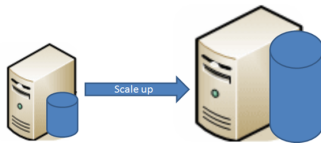
Nuevos desafíos

La diferencias de las arquitecturas de las aplicaciones entre la era cliente/servidor y la las aplicaciones web masivas crearon una presión sobre las bases de datos relacionales que no podía ser aliviada mediante innovación incremental



Las tres V: Volumen-Variedad y Velocidad.

El problema de escalar



Escalamiento vertical



Escalamiento horizontal

Comienzos

- **Teorema CAP.** Eric A. Brewer. *Towards robust distributed systems*. Proc. of the 19th annual ACM symposium on Principles of distributed computing, 2000.
 - Seth Gilbert y Nancy Lynch: *Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services*. ACM SIGACT Volume 33 Issue 2, June 2002
- **MapReduce:** *Simplified Data Processing on Large Clusters*. Jeffrey Dean and Sanjay Ghemawat - 2004
- Amazon publica sobre **Dynamo**
 - DeCandia, Hastorun, Jampani, et al. *Dynamo: Amazon's Highly Available Key-Value Store*. Proceedings of the 21st ACM Symposium on Operating Systems Principles, Stevenson, WA, October 2007.
- Google publica la arquitectura de **Bigtable**
 - Chang, Dean, Ghemawat, et al. *Bigtable: A Distributed Storage System for Structured Data*. ACM Trans. Comput. Syst. 26, 2, Article 4 (June 2008)

Características de NoSQL

- Escalabilidad
- Disponibilidad, Replicación y consistencia eventual
- Alta performance en el acceso a datos
- ACID vs BASE.
- Sin esquema o esquema flexible



Definición

Definiciones

Dan McCreary -Ann Kelly

NoSQL is a set of concepts that allows the rapid and efficient processing of data sets with a focus on performance, reliability, and agility.

Shashank Tiwari

NoSQL is used today as an umbrella term for all databases and data stores that don't follow the popular and well established RDBMS principles and often relate to large data sets accessed and manipulated on a Web scale. This means NoSQL is not a single product or even a single technology. It represents a class of products and a collection of diverse, and sometimes related, concepts about data storage and manipulation.

Definiciones

HOW TO WRITE A CV



Leverage the NoSQL boom

Fowler -Sadalage

NoSQL is an accidental neologism. There is no prescriptive definition— all you can make is an observation of common characteristics.

El origen del nombre

Johan Oskarsson - 2009

To make the most of the flight money I'm putting together a free meetup about "open source, distributed, non relational databases" or NOSQL for short.

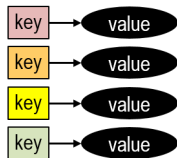
It's taking place on the 11th of June, the day after the Hadoop summit in San Francisco. CBS interactive have been kind enough to provide us with both a venue and free lunch!

- Intro session - Todd Lipcon, Cloudera
- Voldemort - Jay Kreps, LinkedIn
- Cassandra - Avinash Lakshman, Facebook
- Dynomite - Cliff Moon, Powerset
- HBase - Ryan Rawson, Stumbleupon
- Hypertable - Doug Judd, Zvents
- CouchDB - Chris Anderson, couch.io
- VPork - Jon Travis, Springsource
- MongoDB - Dwight Merriman, 10gen
- Infinite Scalability - Jonas S Karlsson, Google

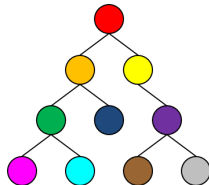


Not Only SQL

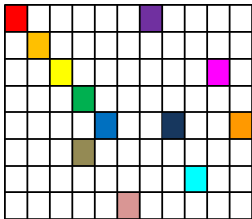
Tipos de Bases No-SQL



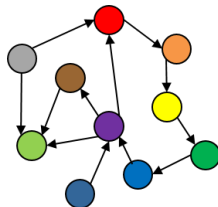
Key-Value



Document



Column Family



Graph Database

Consistencia

Consistencia

Modelos de consistencia

Un modelo de consistencia determina la visibilidad y el orden aparente de las actualizaciones.

Modelos de Consistencia - Tanenbaum

Un modelo de consistencia es básicamente un contrato entre los procesos y el almacén de datos. Este contrato dice que si los procesos aceptan obedecer ciertas reglas, el almacén promete funcionar correctamente.

Consistencia

Consistencia Fuerte

Todas las operaciones de lectura deben devolver datos de la última operación de escritura completada.

Consistencia

Consistencia Fuerte

Todas las operaciones de lectura deben devolver datos de la última operación de escritura completada.



Consistencia

Consistencia Fuerte

Todas las operaciones de lectura deben devolver datos de la última operación de escritura completada.



Consistencia Eventual

Las operaciones de lectura verán conforme pasa el tiempo las escrituras. En un estado de equilibrio el sistema devolvería el último valor escrito. A medida que $t \rightarrow \infty$ los clientes verán las escrituras.

Consistencia no Estricta

- Read Your Own Writes (RYOW) Consistency
- Consistencia de Sesión
- Consistencia Causal
- Consistencia de lectura monotónica.
- Consistencia Eventual.

Read Your Own Writes

El cliente lee sus actualizaciones inmediatamente después de que hayan sido completadas. Independientemente si escribe en un server y lee de otro. Las actualizaciones de otros clientes no tienen porque ser visibles instantáneamente.

Consistencia no Estricta

- Read Your Own Writes (RYOW) Consistency
- **Consistencia de Sesión**
- Consistencia Causal
- Consistencia de lectura monotónica.
- Consistencia Eventual.

Sesión

Es un poco más débil que RYOW. Porvee RYOW sólo si el cliente esta dentro de la misma sesión. Usualmente sobre el mismo server.

Consistencia no Estricta

- Read Your Own Writes (RYOW) Consistency
- Consistencia de Sesión
- **Consistencia Causal**
- Consistencia de lectura monotónica.
- Consistencia Eventual.

Causal

Si el evento ***b*** es causado o influenciado por un evento previo ***a***, la causalidad requiere que todos los demás eventos vean primero a ***a***, y después a ***b***.

Escrituras que potencialmente están relacionadas por la *causalidad*, deben ser vistas por todos los procesos en el mismo orden. Las escrituras concurrentes pueden verse en un orden diferente en diferentes máquinas.

Consistencia no Estricta

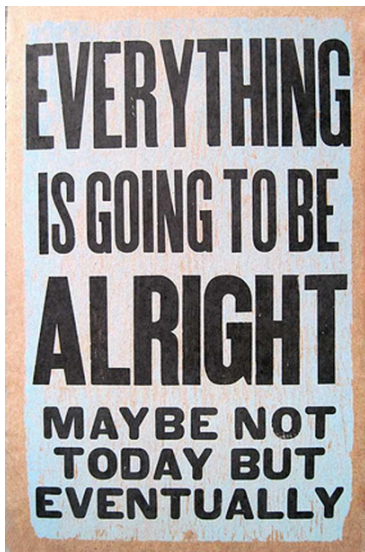
- Read Your Own Writes (RYOW) Consistency
- Consistencia de Sesión
- Consistencia Causal
- **Consistencia de lectura monotónica.**
- Consistencia Eventual.

Monotonic Read

Si un proceso lee el valor de un elemento de datos x , cualquier operación de lectura sucesiva sobre x que haga ese proceso devolverá siempre el mismo valor o un valor más reciente.

Consistencia no Estricta

- Read Your Own Writes (RYOW) Consistency
- Consistencia de Sesión
- Consistencia Causal
- Consistencia de lectura monotónica.
- **Consistencia Eventual.**



El teorema CAP

- C** Consistency: Todos ven los mismos datos al mismo tiempo
- A** Availability: Si se puede comunicar con un nodo en el cluster entonces se pueden leer y escribir datos. ⁵
- P** Partition tolerance: El cluster puede sobrevivir a roturas de comunicación que lo dividan en particiones que no pueden comunicarse entre ellas.

⁵“every request received by a non failing node in the system must result in a response”

El teorema CAP

- C** Consistency: Todos ven los mismos datos al mismo tiempo
- A** Availability: Si se puede comunicar con un nodo en el cluster entonces se pueden leer y escribir datos. ⁵
- P** Partition tolerance: El cluster puede sobrevivir a roturas de comunicación que lo dividan en particiones que no pueden comunicarse entre ellas.

Teorema CAP

Dado C, A y P: sólo se puede tener un máximo de dos de estas propiedades para cualquier sistema de datos compartidos

⁵“every request received by a non failing node in the system must result in a response”

CAP - Falacias de la computación distribuida

- La red es fiable.
- La latencia es cero.
- El ancho de banda es infinito.
- La red es segura.
- La topología no cambia.
- Hay un administrador.
- El costo del transporte es cero.
- La red es homogénea.

CAP - Falacias de la computación distribuida

- La red es fiable.
- La latencia es cero.
- El ancho de banda es infinito.
- La red es segura.
- La topología no cambia.
- Hay un administrador.
- El costo del transporte es cero.
- La red es homogénea.

ATENCIÓN

Fallos de red le suceden a su sistema y no puede elegir cuando se producen.

CAP - Falacias de la computación distribuida

ATENCIÓN

Fallos de red le suceden a su sistema y no puede elegir cuando se producen.

CAP - Falacias de la computación distribuida

ATENCIÓN

Fallos de red le suceden a su sistema y no puede elegir cuando se producen.

- Hay que tolerar particiones!
 - CP - Consistency/Partition Tolerance
 - AP - Availability/Partition Tolerance

CP vs AP

No es una decisión binaria. Un poco de uno a costa del otro.

Eric Brewer: BASE

Mito

¿La plata es importante, así que los bancos debe usar transacciones para mantener la plata segura y consistente?

Eric Brewer: BASE

Mito

¿La plata es importante, así que los bancos debe usar transacciones para mantener la plata segura y consistente?

Realidad

Las transacciones bancarias son inconsistentes, particularmente para ATM (Cajeros Automáticos). ATM es diseñado para tener un comportamiento en modo normal y otro en modo partición. En modo partición la *Availability* es elegida por sobre la *Consistencia*

Realidad

La industria financiera históricamente no tuvo consistencia porque no tenía comunicación perfecta.

Realidad

ATMs eligen operaciones conmutativas así el orden en el cual se aplican no importa.

Relidad ATMs

Si hay una partición el cajero puede seguir funcionando, luego al volver a recuperar la partición se envían las operaciones y el saldo final sigue siendo correcto.

Idea

Lo que se está haciendo es delimitar y administrar el riesgo. ATMs son rentables aún a costa de alguna pérdida.

- **Durabilidad vs. Performance.**

- Durabilidad de replicación

Aplicar Cambios a Memoria y cada Δt enviar los cambios a disco (Redis)
A menudo, se puede especificar las necesidades durabilidad por cada llamada, de tal manera que las actualizaciones más importantes pueden forzar su escritura a disco.



- Durabilidad vs. Performance.
- **Durabilidad de replicación**

Un nodo puede procesar una actualización y falla antes de que se replique en otros nodos.

Maestro/Esclavo: Se puede mejorar esperando por algunas réplicas antes confirmar al cliente.

