

Eventual Consistency

Problema

- Réplica de datos.
- Posibles diferencias entre lo que almacena cada nodo.
- Asegurar que todos se encuentran en el mismo estado requiere mucho cómputo y la complejidad aumenta aún más cuando se agregan más nodos.

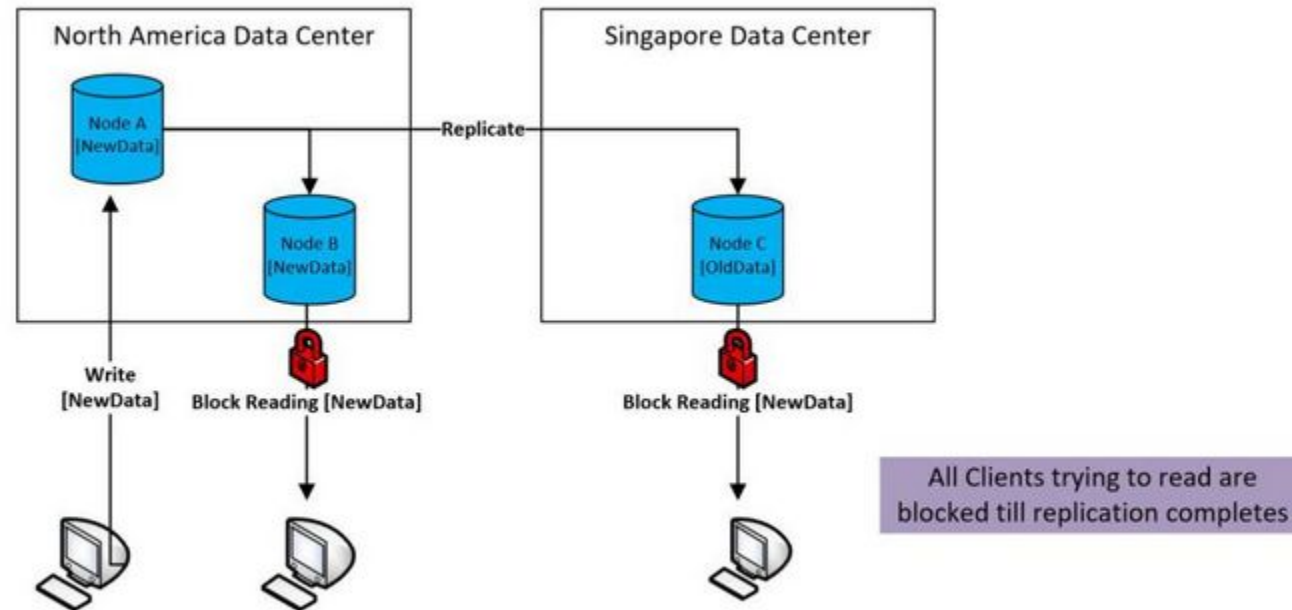
Teorema CAP

- No se puede tener consistencia, disponibilidad y tolerancia a particiones simultáneamente.
- Debe hacerse un trade-off dependiendo de la aplicación.

ACID

- Atomicity
- Consistency
- Isolation
- Durability

Strict consistency



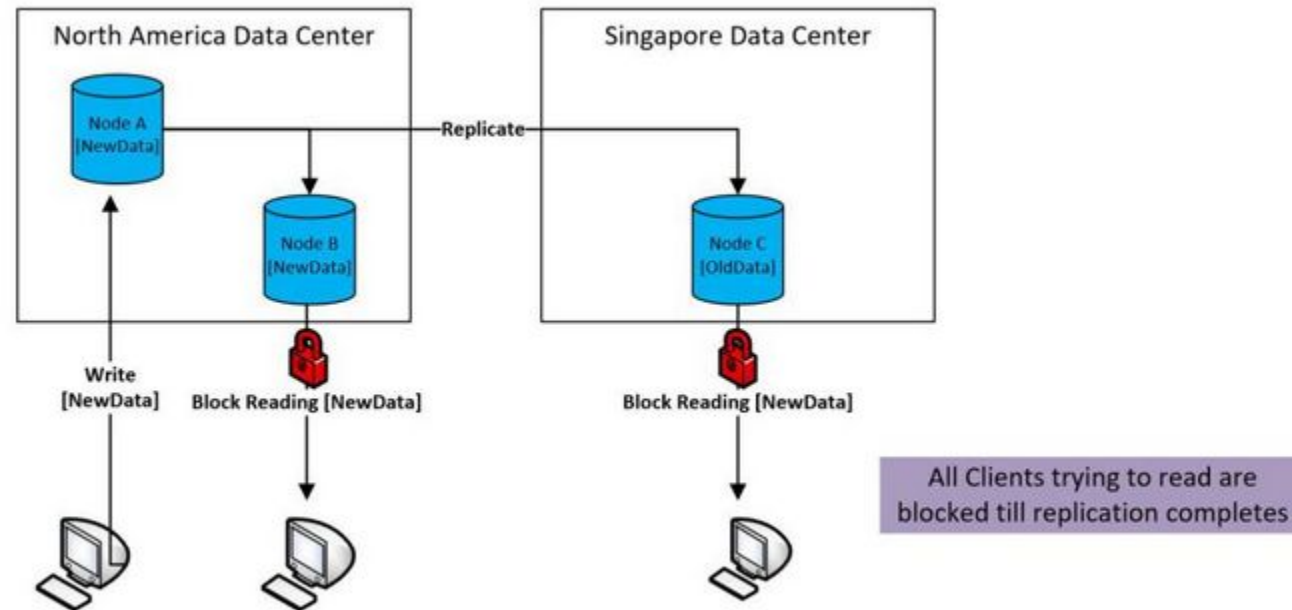
BASE

- **Abandona consistencia a favor de disponibilidad y tolerancia a particiones (CAP).**
- **B**asically **A**vailable
- **S**oft state
- **E**ventual consistency

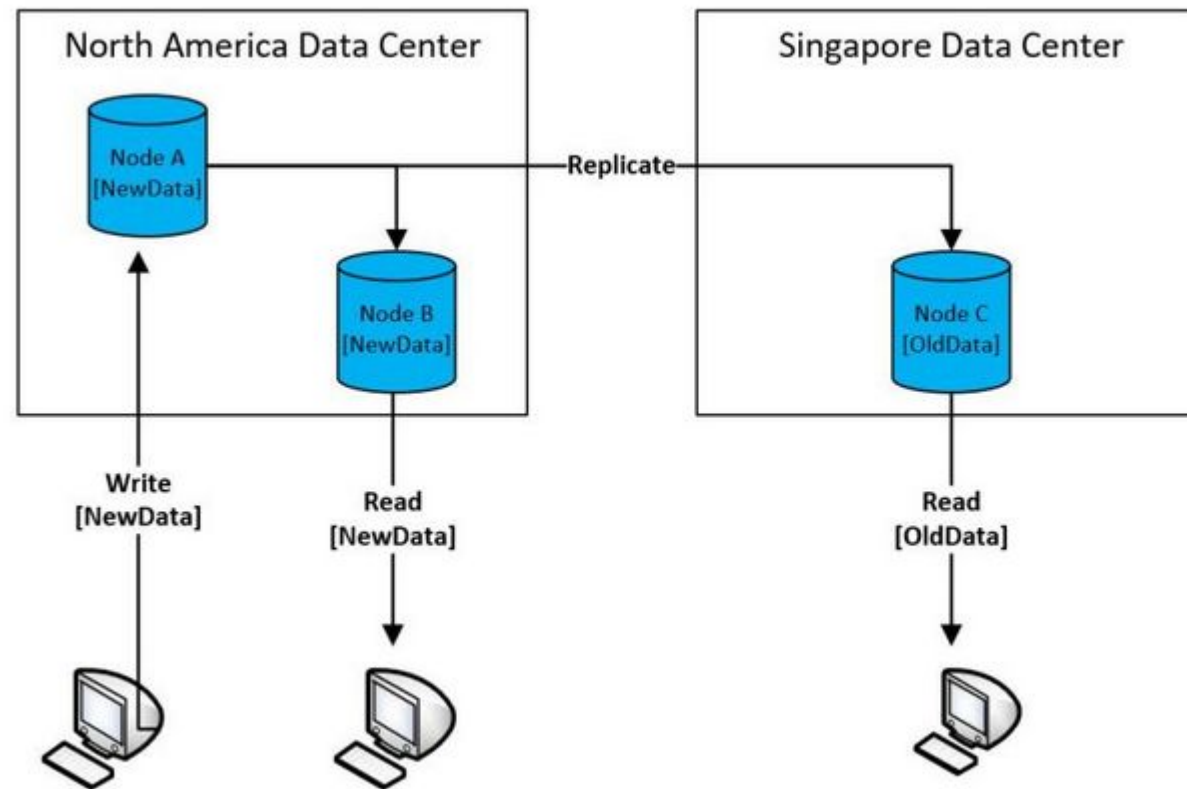
Eventual Consistency

Modelo de consistencia que asegura que, si cierto elemento no es modificado, eventualmente todos los accesos a dicho ítem obtendrán el mismo valor.

Strict consistency



Eventual Consistency



Eventual Consistency

¿Cuándo conviene?

- Necesidad de escalar.
- Tareas tolerantes a fallas.

¿Cuándo conviene?

Eventual Consistency Is Just Caching

Sunday, February 21st, 2010.

People seem to make a big deal about eventual consistency. Those same people don't seem to think twice about using a cache to improve the performance of their system when the database can't handle it. That cache holds a copy of data that isn't 100% synchronized with the database. That's eventual consistency. Get over it.

Desafíos - Convergencia

- Detectar diferencia en versiones.
- Seleccionar estado final. Timestamp y vector clocks pueden ser usados para analizar concurrencia.

Sistemas de bases de datos



Sistemas de bases de datos

348 systems in ranking, November 2018

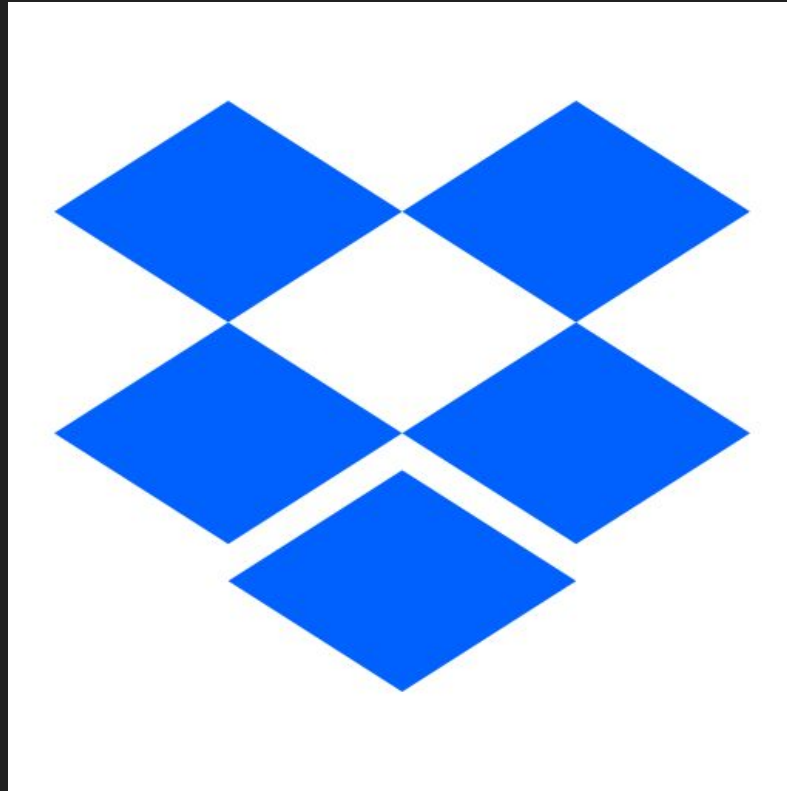
Rank			DBMS	Database Model	Score		
Nov 2018	Oct 2018	Nov 2017			Nov 2018	Oct 2018	Nov 2017
1.	1.	1.	Oracle	Relational DBMS	1301.11	-18.16	-58.94
2.	2.	2.	MySQL	Relational DBMS	1159.89	-18.22	-162.14
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1051.55	-6.78	-163.53
4.	4.	4.	PostgreSQL	Relational DBMS	440.24	+20.85	+60.33
5.	5.	5.	MongoDB	Document store	369.48	+6.30	+39.01
6.	6.	6.	IBM Db2	Relational DBMS	179.87	+0.19	-14.19
7.	7.	9.	Redis	Key-value store	144.17	-1.12	+22.99
8.	8.	10.	Elasticsearch	Search engine	143.46	+1.13	+24.05
9.	9.	7.	Microsoft Access	Relational DBMS	138.44	+1.64	+5.12
10.	11.	11.	SQLite	Relational DBMS	122.71	+5.96	+9.95
11.	10.	8.	Cassandra	Wide column store	121.74	-1.64	-2.47
12.	13.	15.	Splunk	Search engine	80.37	+3.48	+15.50
13.	12.	12.	Teradata	Relational DBMS	79.31	+0.67	+1.07
14.	14.	18.	MariaDB	Relational DBMS	73.25	+0.12	+17.96
15.	16.	19.	Hive	Relational DBMS	64.57	+3.47	+11.32

Desafíos - Convergencia

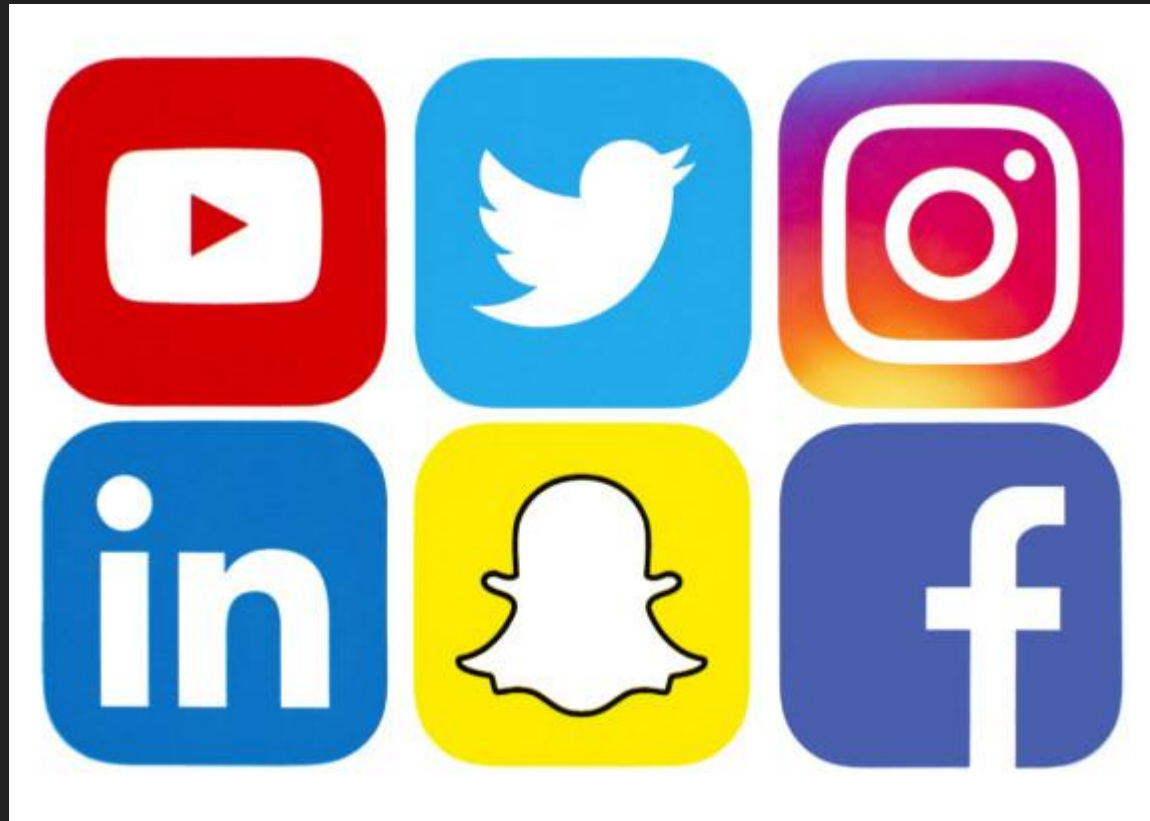
- Cassandra DB.
- Factor de replicación = Copias de un dato en el cluster.
- Para validar una lectura, si N nodos están de acuerdo con un valor se considera válido.
- N es modificable. N = factor de replicación -> Strict consistency.

Casos de uso

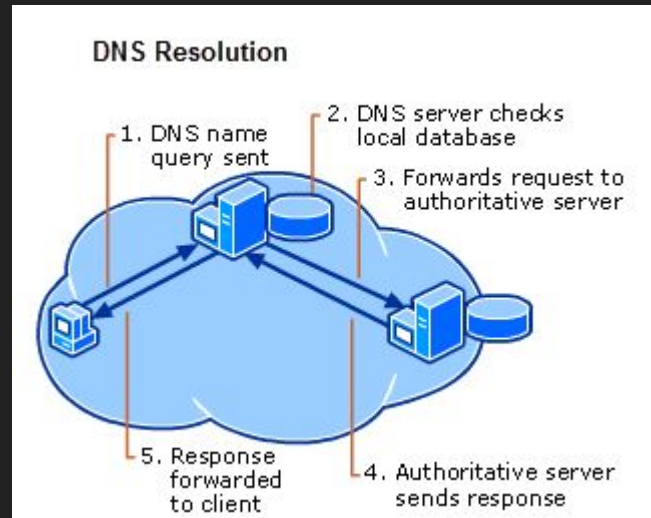
Casos de uso - Dropbox



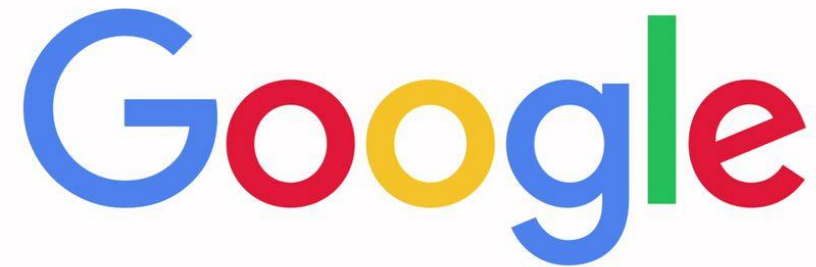
Casos de uso - Redes sociales



Casos de uso - DNS



Casos de uso - Google



Referencias

<http://udidahan.com/2010/02/21/eventual-consistency-is-just-caching/>

<http://www.acodersjourney.com/2018/08/eventual-consistency/>

<https://dzone.com/articles/eventual-consistency>

<http://blog.cs.miami.edu/burt/2016/12/09/dns-and-eventual-consistency/>

<https://cloud.google.com/blog/products/gcp/why-you-should-pick-strong-consistency-when-ever-possible>

<https://mariadb.org/eventually-consistent-databases-state-of-the-art/>

<https://db-engines.com/en/ranking>

<https://web.stanford.edu/class/cs345d-01/rl/eventually-consistent.pdf>

