

“NoSQL”



Que es “NoSQL”

“NoSQL” == “Bolsa de Gatos”

Es un término totalmente genérico que agrupa motores de bases de datos que no implementan el estándar SQL.

Generalidades

(Generalmente) las bases de datos “NoSQL”:

- No tienen modelo ni esquema predefinido.
- No representan relaciones.
- Su utilización es programática.
- No siguen ningún estándar (depende del motor).
- No siguen un protocolo común (depende del motor).
- No garantizan autenticación (depende del motor).

Generalidades

(Generalmente) las bases de datos “NoSQL” son:

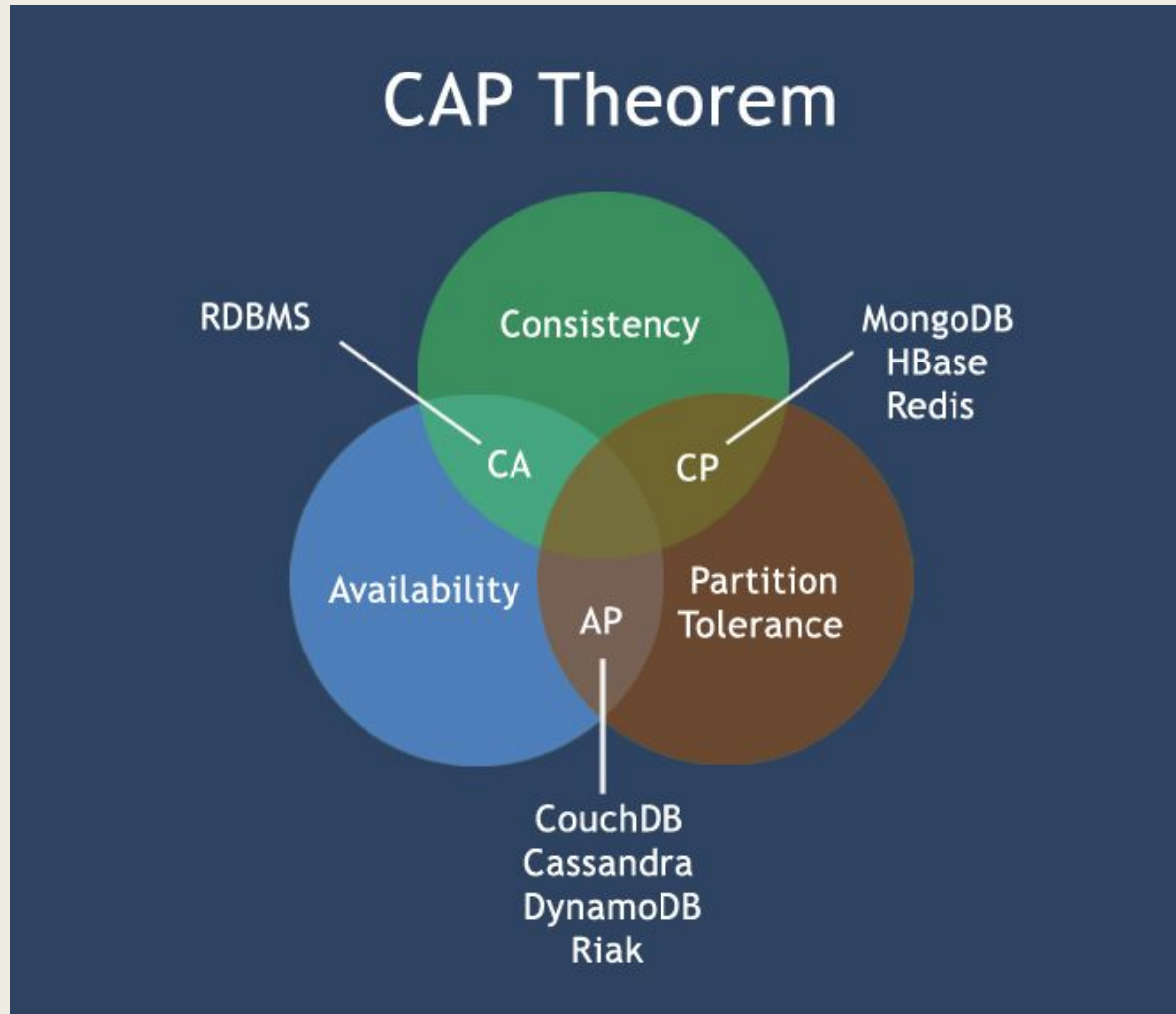
- De casos de uso específicos.
- Al día de hoy, “bleeding edge”:
 - No son probadas y estables como las DBs relacionales.
 - Pueden cambiar su especificación libremente.
 - Pueden desaparecer o ser abandonadas.

Teorema de CAP (o de Brewer)

Para un sistema informático distribuído, es totalmente imposible garantizar a la vez:

- **Consistency** (consistencia)
 - Todos los nodos ven lo mismo al mismo tiempo.
- **Availability** (disponibilidad)
 - Garantía de que todo pedido recibe respuesta.
- **Partition Tolerance** (tolerancia al particionamiento)
 - Garantía de que el sistema sigue operando incluso si fallan algunos nodos de la partición.

Teorema de CAP (o de Brewer)



Consistencia Eventual

La **consistencia eventual** es la garantía **informal**, para un set de datos en un sistema distribuido, de que si éste no recibe actualizaciones durante un tiempo suficiente, todos los nodos **convergerán eventualmente, siendo mutuamente consistentes entre sí.**

(muchas bases de datos “NoSQL” aseguran poseer **consistencia eventual**)

Para qué se usan?

- Persistencia de documentos sin esquema definido
- High-availability, Fault-tolerance
- Distribución, Escalabilidad
- Caching
- Volúmenes inmensos de datos (partiendo de terabytes)
- Sharding
- Message queues
- Persistencia de alta velocidad
- MapReduce
- Propósitos específicos (GIS, Graphs, Nodes, etc).

Cada caso de uso depende del motor.

Cada motor **se focaliza en un par de casos de uso específicos.**

No hay solución genérica.

Para qué ***no*** se usan?

- Aplicaciones donde **consistencia y disponibilidad** son los factores clave.
- Sets de datos fuertemente basados en **relaciones**.
- Reemplazar de una DB relacional **sin razón justificada**.
- Como **solución genérica** de persistencia.



Regla General
Leer el manual, **no el folleto.**



(Tristemente) Típico

“Sin esquema? Bueno, serializo en JSON y lo guardo en un campo TEXT en MySQL. Al leer, deserializo. Total, ya uso MySQL para todo lo demás”

- Imposible buscar
- Imposible indexar
- Imposible optimizar
- Imposible validar

Soluciones

Document Stores, o Key-value stores:

- MongoDB, Cassandra, Redis, CouchDB, Hbase, Riak, Postgres' Hstore, etc.

Optimizado, Buscable, Indexable, Validable.

Además: Integrado y Distribuido.

<http://en.wikipedia.org/wiki/NoSQL>

<http://kkovacs.eu/cassandra-vs-mongodb-vs-couchdb-vs-redis>

Soluciones

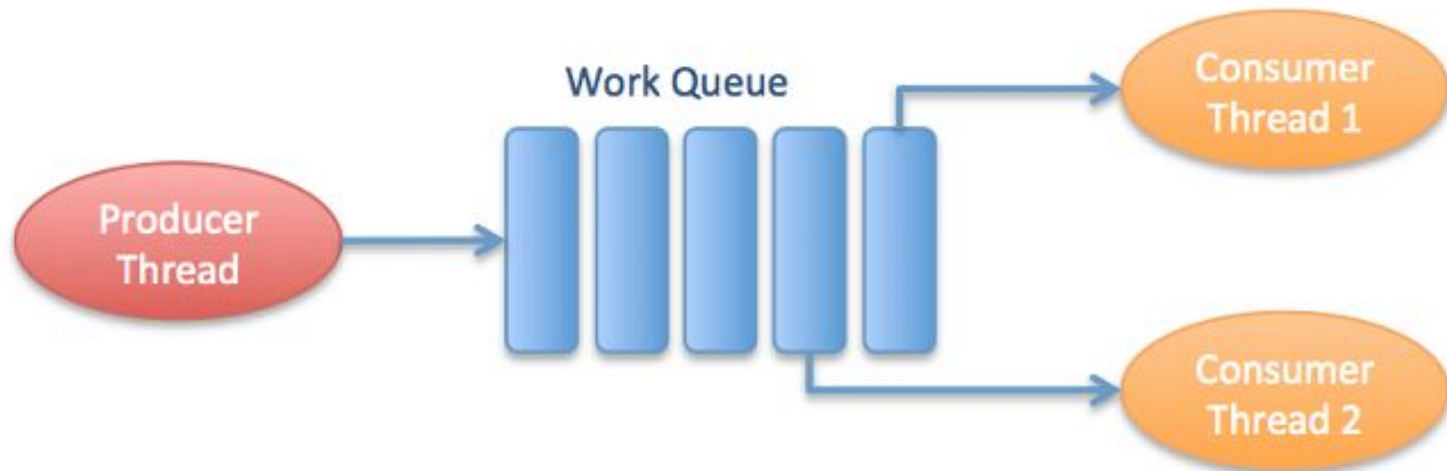


CouchDB

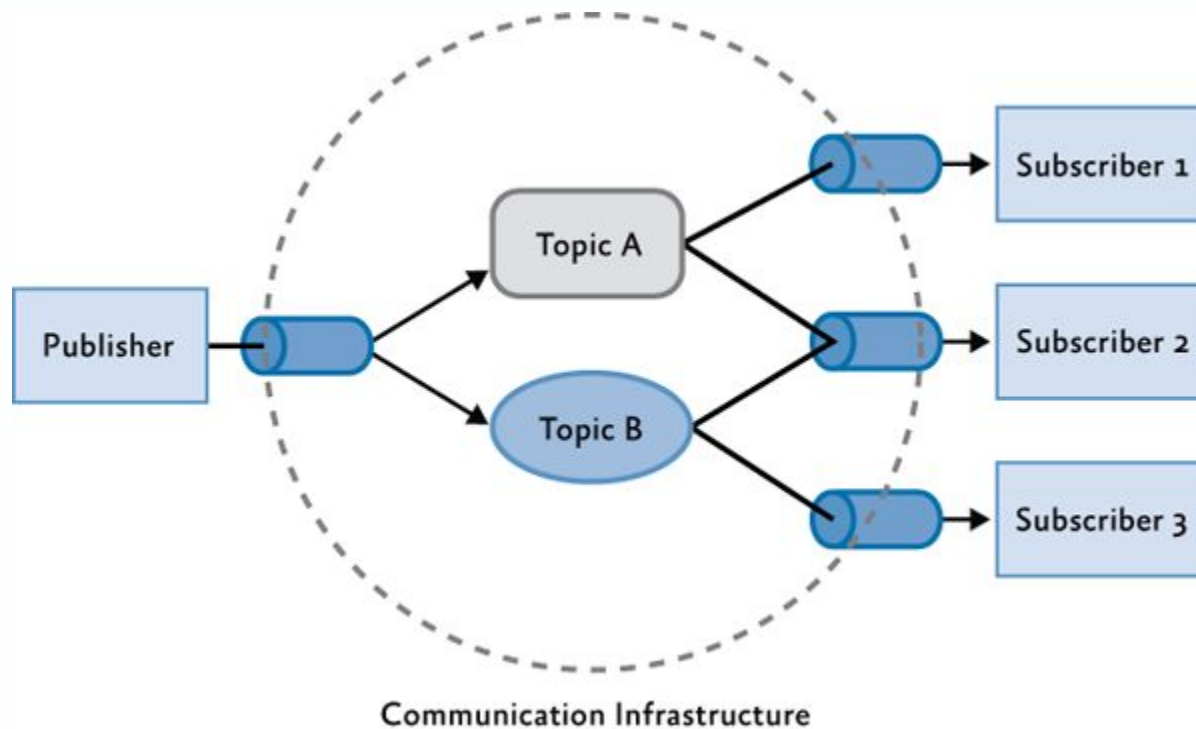


redis

Producer-Consumer



Publisher-Subscriber



Preguntas

