Grandes de Bases de Datos

Paradigma

No-SQL

¿SQL o No – SQL?

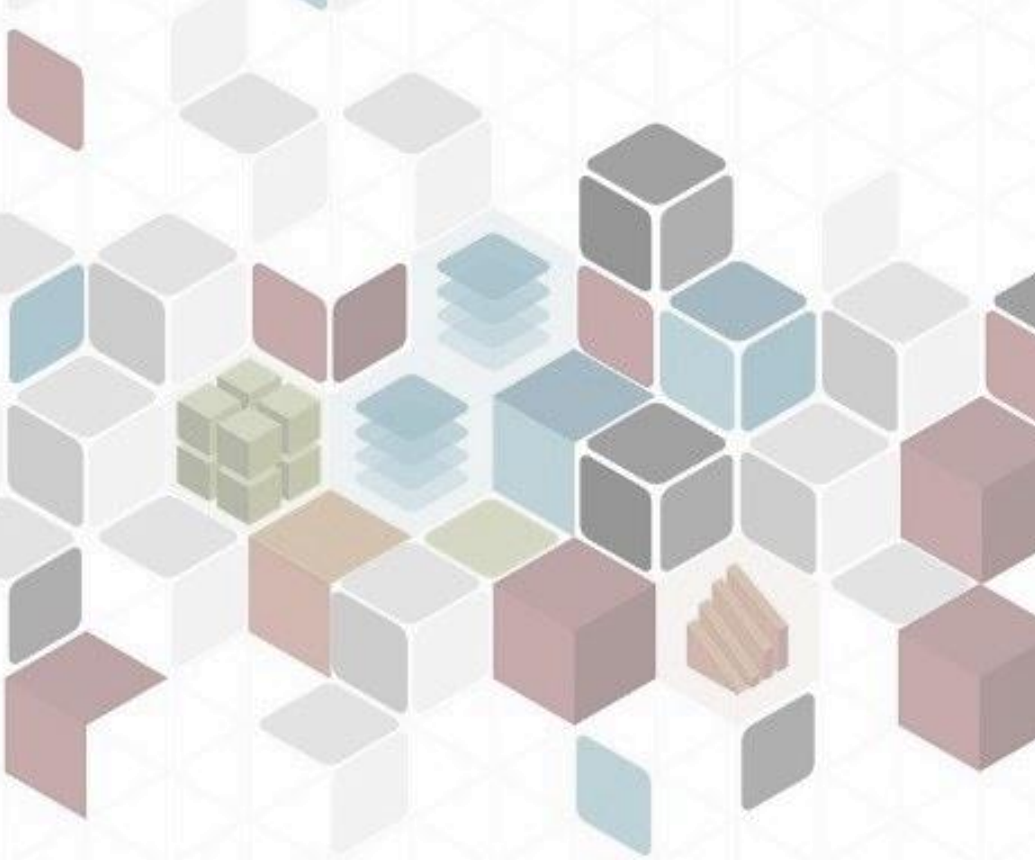
- SQL ha sido la implementación del modelo relación más ampliamente difundido durante los últimos 35 años
- Representó la difusión de dicho modelo como ***el*** modelo de almacenamiento de información por excelencia.
- No es el único modo de guardar información
- **¡Por que no toda la información es igual!**

¿SQL o No – SQL?

- Estructura
  - Estructurados
  - Semi estructurados
  - No estructurados
- Disponibilidad
  - ***ACID complaint synchronized***
  - ***ACID complaint non-synchronized***
  - ***NON ACID complaint***

¿SQL o No – SQL?

- Google construye una infraestructura escalable masiva para su principal producto:
  - Google Search
  - Google Maps
  - Google Earth
  - Gmail
  - Google Finance



¿SQL o No – SQL?

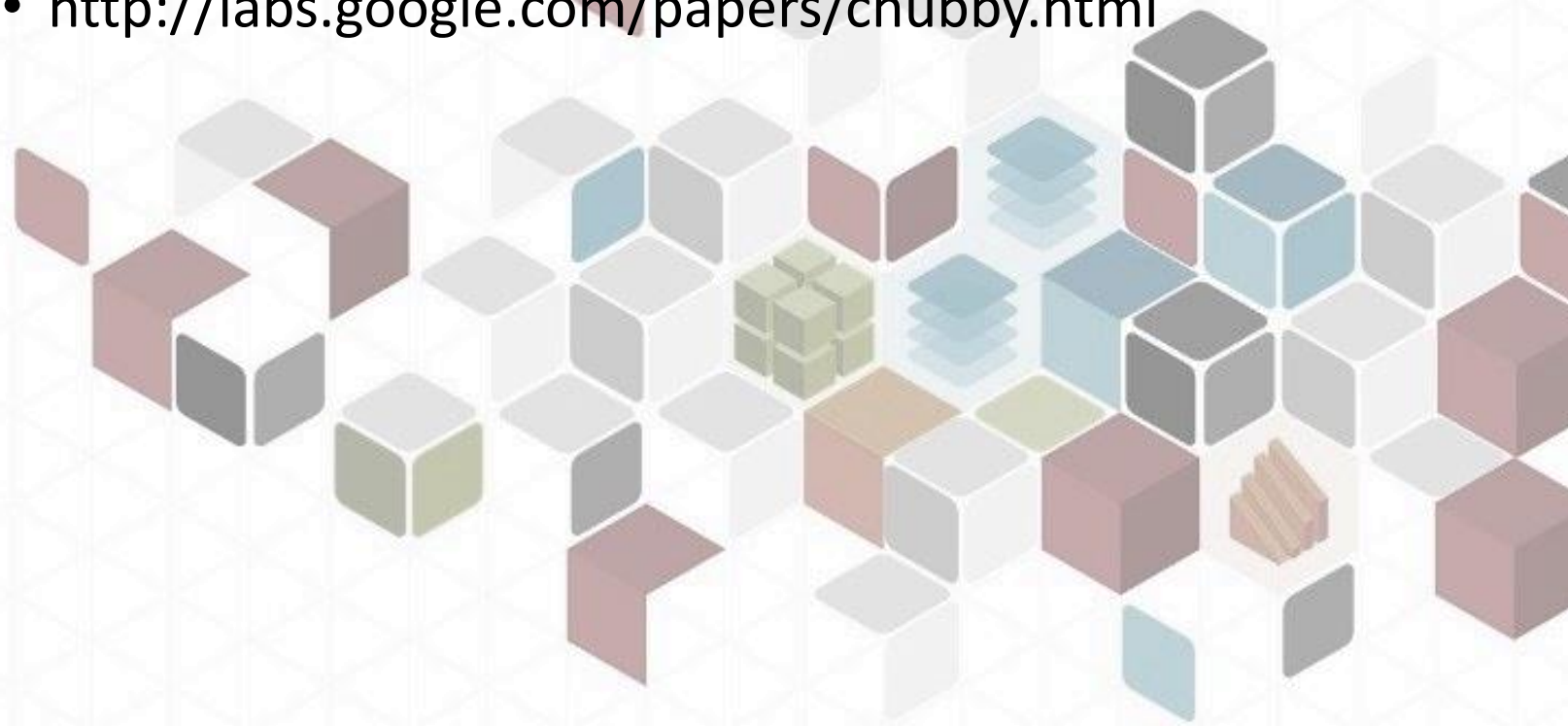
- La idea inicial es procesar gran cantidad de información
- Utilizó:
  - Sistema de archivos distribuidos
  - Almacenamiento orientado a columnas
  - Sistema de coordinación distribuida
  - Ejecución paralela de procesos – ***Map-Reduce***

¿SQL o No – SQL?

- Los principales documentos que publica, son:
  - "The Google File System"
    - http://labs.google.com/papers/gfs.html
  - "MapReduce: Simplified Data Processing on Large Clusters"
    - http://labs.google.com/papers/mapreduce.html
  - "Bigtable: A Distributed Storage System for Structured Data"
    - http://labs.google.com/papers/bigtable.html

¿SQL o No – SQL?

- Los principales documentos que publica, son:
  - “The Chubby Lock Service for Loosely-Coupled Distributed Systems”;
    - http://labs.google.com/papers/chubby.html



¿SQL o No – SQL?

- Se crea el motor de búsqueda "***open-source***", **Lucene** (replica la infraestructura de Google)
- Se unen a **Yahoo**, en conjunto crean "***Hadoop***"
- **Amazon** libera información sobre "***Dynamo***", su medio de almacenamiento altamente distribuido

¿SQL o No – SQL?

- Escalabilidad
- Habilidad de un sistema de incrementar su rendimiento (“***throughput***”) al agregar recursos para manejar una carga adicional
  - Aumentar los recursos internos
  - Aumentar los recursos externos – Nodos
- Procesar datos en clúster horizontalmente escalados, es complejo…

MapReduce

MapReduce

- Modelo de programación en paralelo
- Patentado por Google… pero copiado por implementaciones "*open-source*"
- Deriva de conceptos de programación funcional
  - ***Map*** – Aplica una función u operación a una lista
  - ***Reduce*** – Aplica una función a los elementos de una lista y retorna un elemento

¿Cómo se guarda la información?

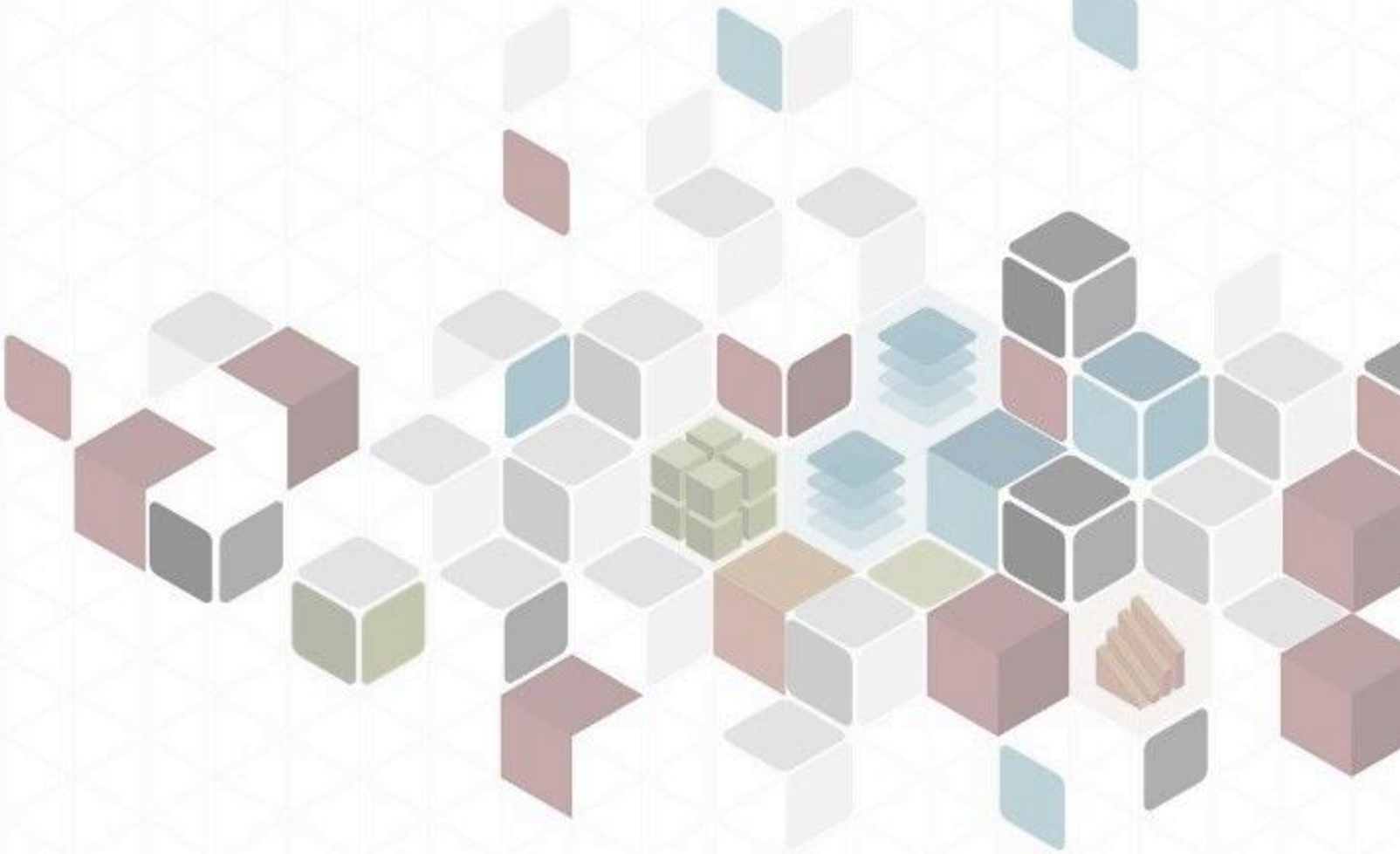
- Almacenamiento ***Columnar-Tabular***
- Almacenamiento ***Orientado a documentos***
- Almacenamiento ***Key-Value***

Almacenamiento Columnar

- Almacenamiento ***Columnar-Tabular***

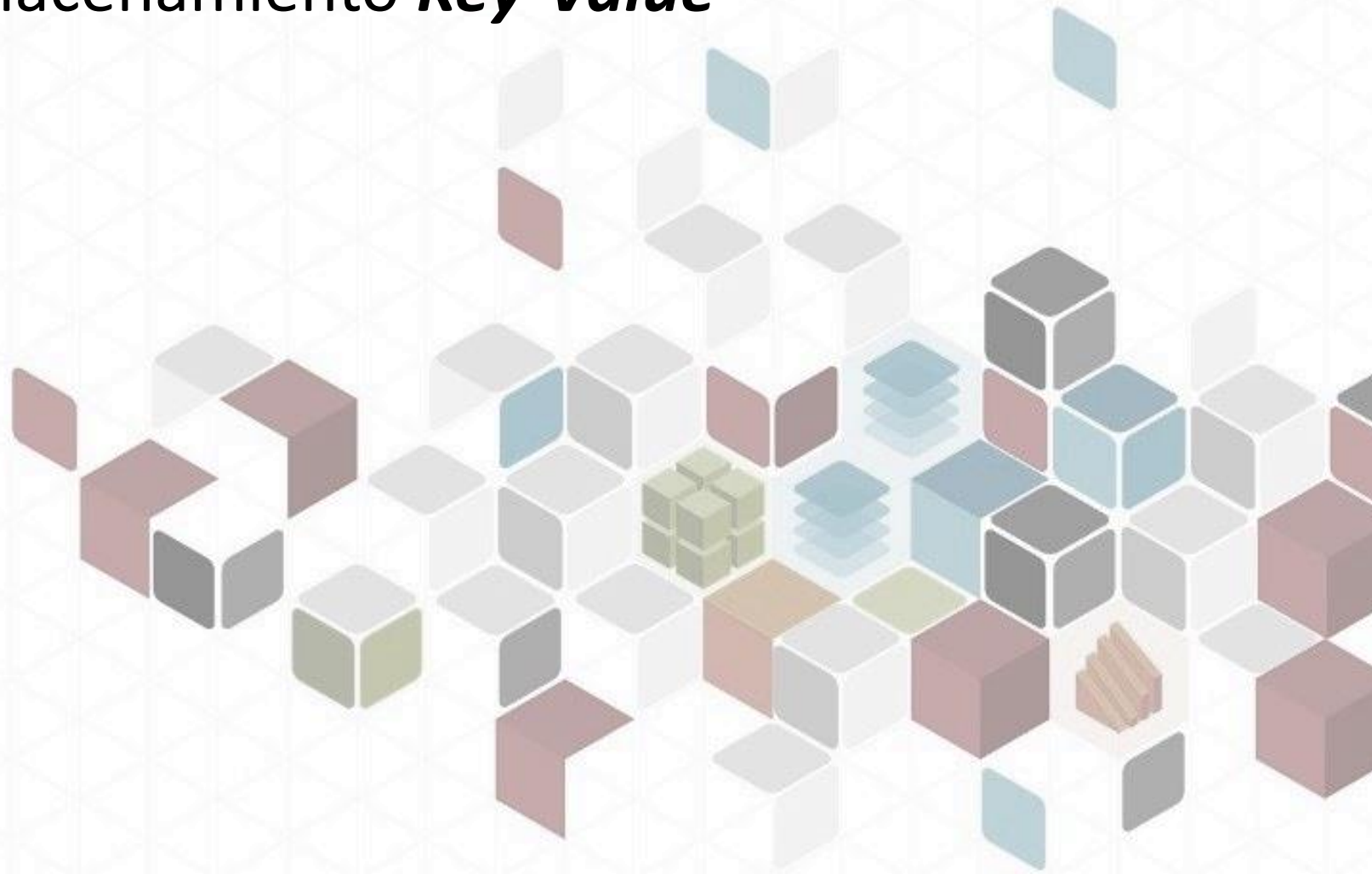
Almacenamiento Doc-Oriented

- Almacenamiento ***Orientado a documentos***



Almacenamiento Key-Value

- Almacenamiento ***Key-Value***



¿Quién utiliza esto?

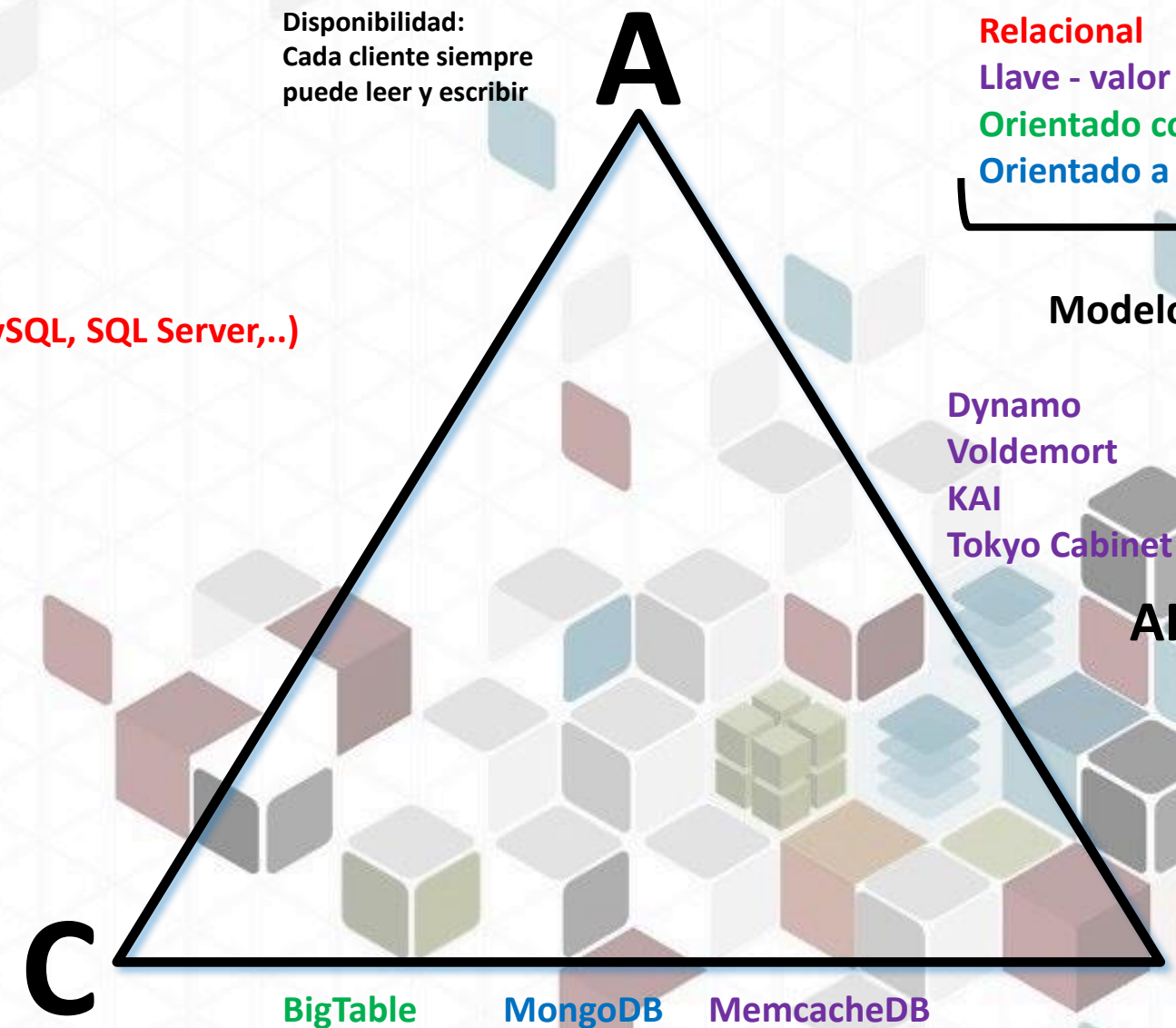
CAP - Teorema

- **C**onsistencia
  - Todos los nodos comparten los mismos datos al mismo tiempo
- Disponibilidad (***Availability***)
  - Las fallas en los nodos no previenen a los restantes de continuar operando
- Tolerancia a particiones (***Partition Tolerance***)
  - El sistema continua operando a pesar de la pérdida arbitraria de mensajes

Solamente se pueden escoger 2 de los 3

Eric Brewer

Panorama general: NoSQL



Disponibilidad:
Cada cliente siempre puede leer y escribir

Relacional
Llave - valor
Orientado columnas – Tabular
Orientado a documentos

Modelo de datos

CA

SMBDR (mySQL, SQL Server,..)
Greenplum
Aster Data
Vertica

Dynamo
Voldemort
KAI
Tokyo Cabinet
Cassandra
SimpleDB
CouchDB
Riak

AP

BigTable
Hypertable
Hbase
MongoDB
Terrastore
Scalaris
MemcacheDB
Redis
Berkeley

CP

Consistencia:
Todos los clientes siempre tienen la misma vista de los datos

Tolerancia a partición:
El sistema funciona a pesar de particiones físicas de la red

Panorama general: NoSQL

- CA
  - La corrupción es posible si nodos activos no pueden comunicarse
- CP
  - El sistema es inaccesible si algún nodo se pierde
- AP
  - El sistema siempre esta disponible, pero posiblemente no siempre contenga datos consistentes

Consistencia débil

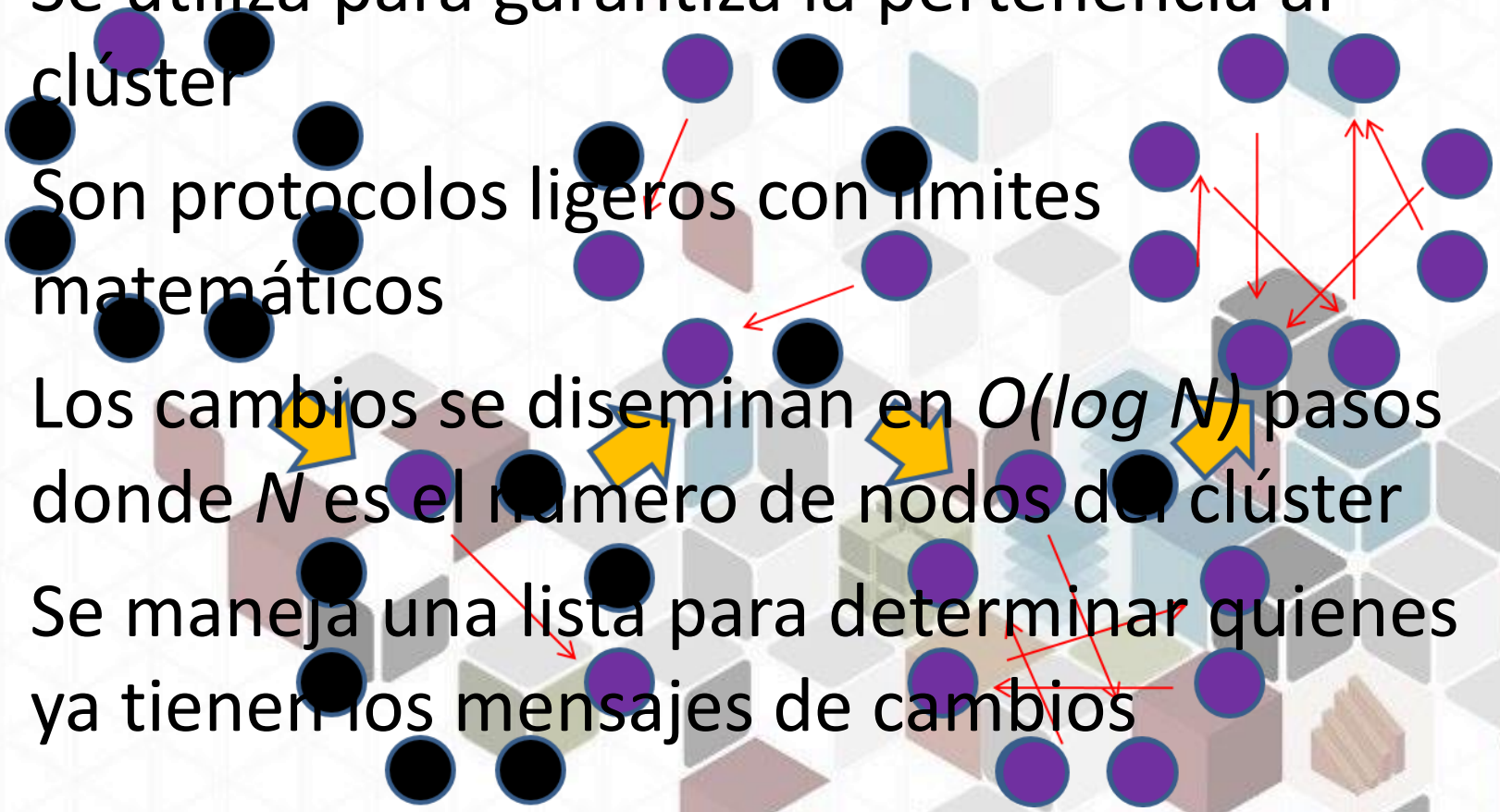
- La consistencia débil permite que sistemas de datos distribuidos, se mantengan en estados “similares”
- Cuando no ocurren actualizaciones por un periodo largo de tiempo, todas las actualizaciones se propagarán a través del sistema
- Las replicas se vuelven consistentes

Protocolos “***gossip***”

- ¿Cómo se propaga la información? Los chismes son la mejor opción!! ☺

Protocolos "***gossip***"

- Se utiliza para garantiza la pertenencia al clúster
- Son protocolos ligeros con límites matemáticos
- Los cambios se diseminan en *O(log N)* pasos donde *N* es el número de nodos del clúster
- Se maneja una lista para determinar quienes ya tienen los mensajes de cambios

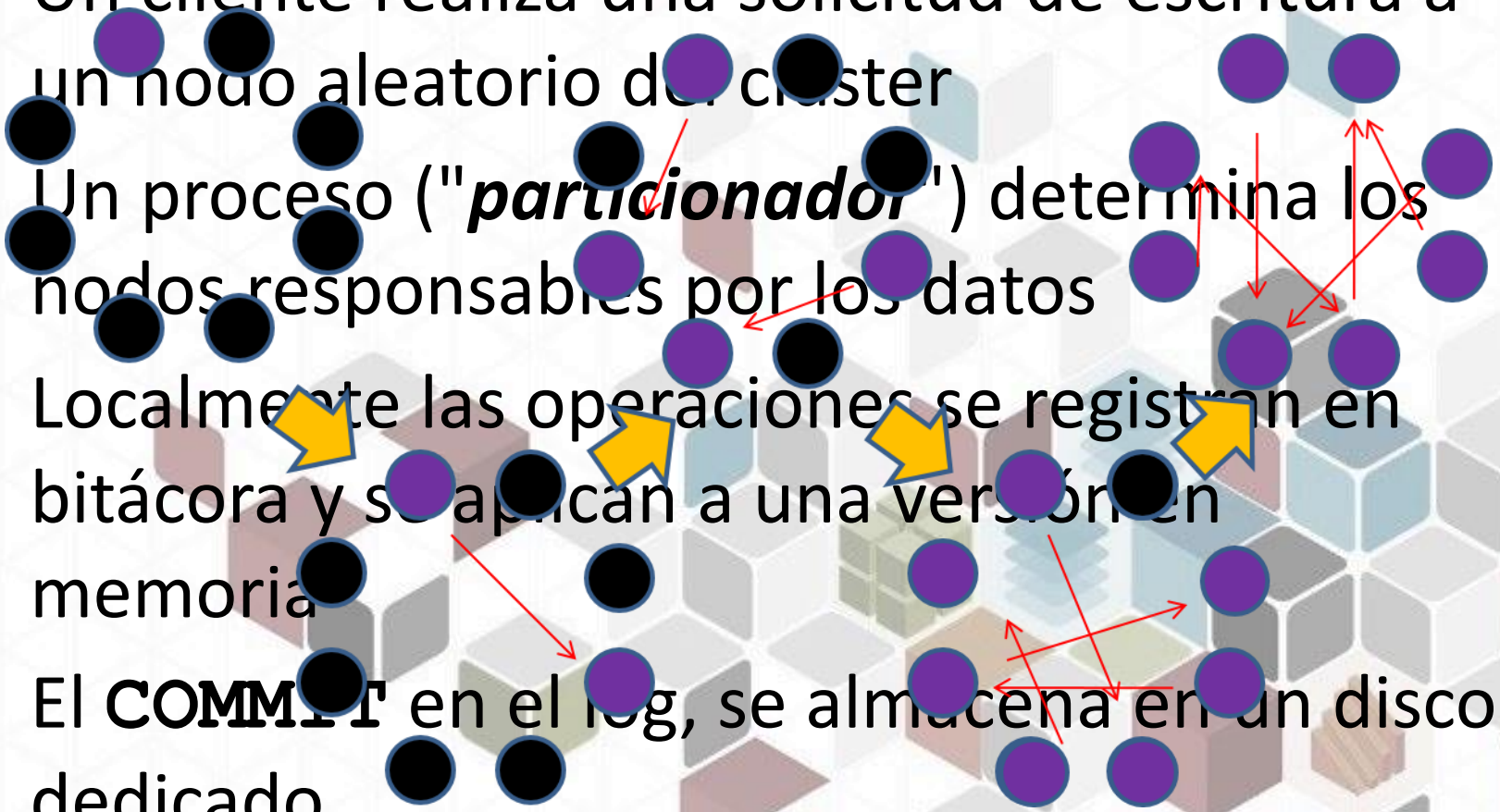


Protocolos “*gossip*”

- Cada T segundos cada miembro del clúster incrementa su contador de red interna y selecciona otro miembro para comunicarse con él (enviar la lista)
- Un miembro une su lista (local) con la lista que recibe

Escrituras

- Un cliente realiza una solicitud de escritura a un nodo aleatorio del clúster
- Un proceso ("***particionador***") determina los nodos responsables por los datos
- Localmente las operaciones se registran en bitácora y se aplican a una versión en memoria
- El **`COMMIT`** en el log, se almacena en un disco dedicado



Escrituras - Propiedades

- No se obtienen bloqueos
- Se tienen accesos secuenciales
- Se comporta como escrituras ***write through cache*** (como en los CPUs)
- Se garantiza atomicidad, al menos en CF
- Siempre se puede escribir (en algún nodo, después se sincronizan)

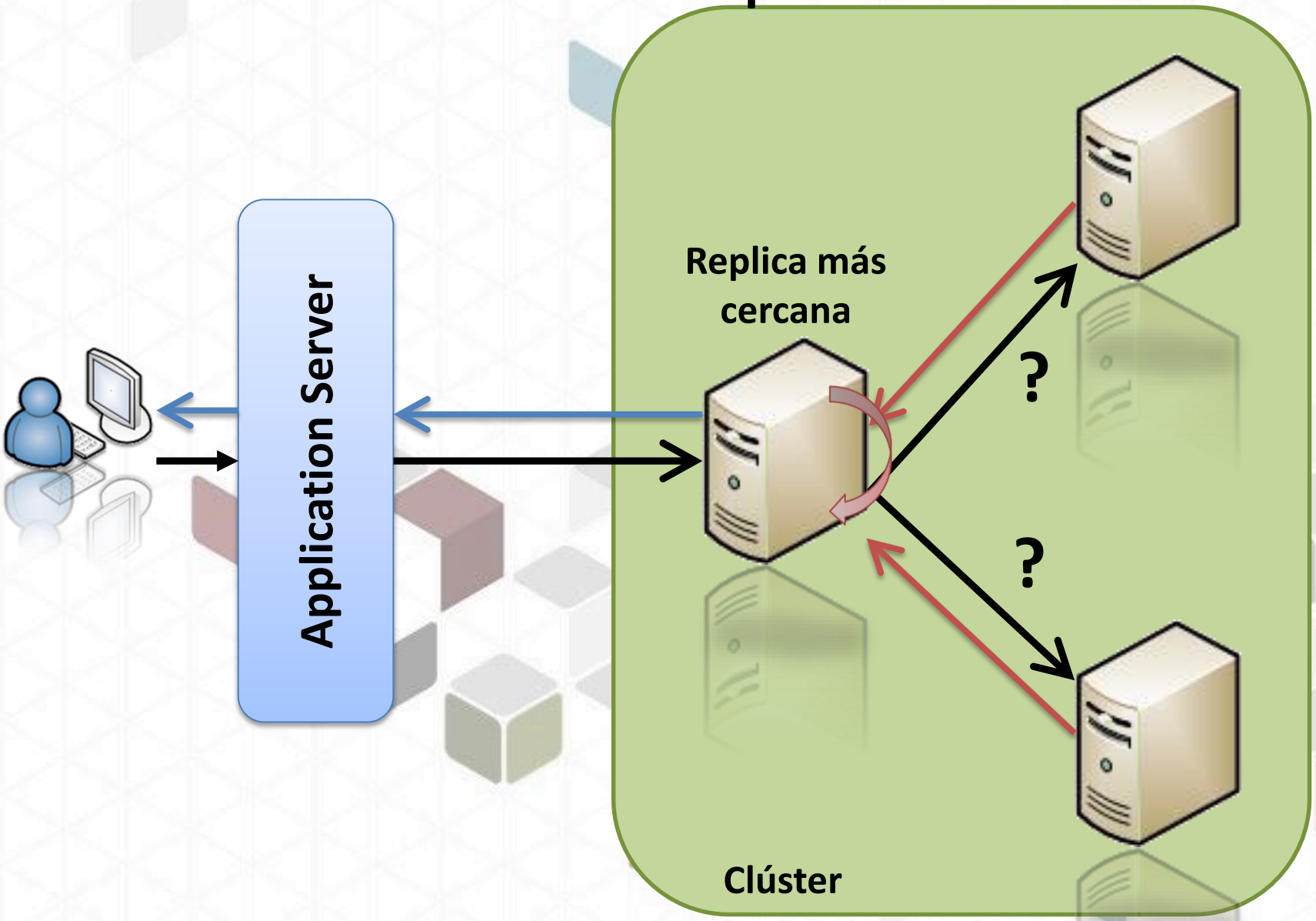
Lecturas

- Un cliente realiza una solicitud de lectura a un nodo aleatorio, el cual la reenvía a los n nodos
  - Espera por R respuestas
  - Espera por N-R respuestas y realiza lecturas reparadas
- Primero lee de la estructura conocida como ***memtable***, si está incompleta entonces utiliza las ***SSTables***

Lecturas - Propiedades

- Desempeño similar a las escrituras
- Se pueden mitigar grandes búsquedas secuenciales con más RAM (hay mas elementos en las ***memtable***)
- Se escala a billones de registros

Lecturas - Operación



… mas. …