TÉCNICO LISBOA

Bases de Dados

T27 - Big Data & NoSQL

Prof. Flávio Martins
Prof. Daniel Faria

Sumário

- Big Data
- Object Databases
- Key–Value Stores
- Wide-Column Stores
- Time Series Databases
- Document Stores
- Distributed File Systems
- Streaming Databases
- Graph Databases

TÉCNICO LISBOA

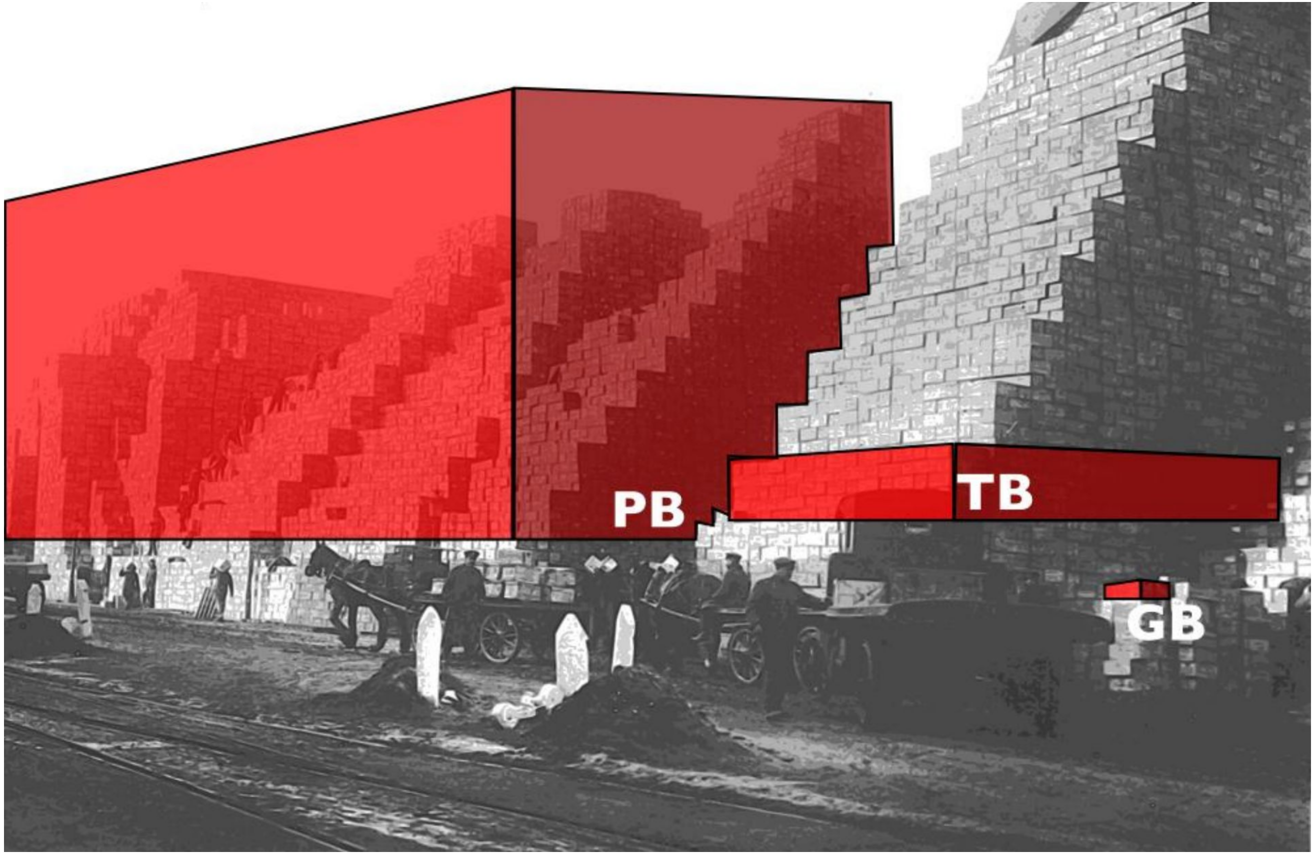
Big Data

Big Data

Differentiated from data handled by earlier generation databases:

- **Volume**: much larger amounts of data stored
- **Velocity**: much higher rates of insertions
- **Variety**: many types of data, beyond relational data

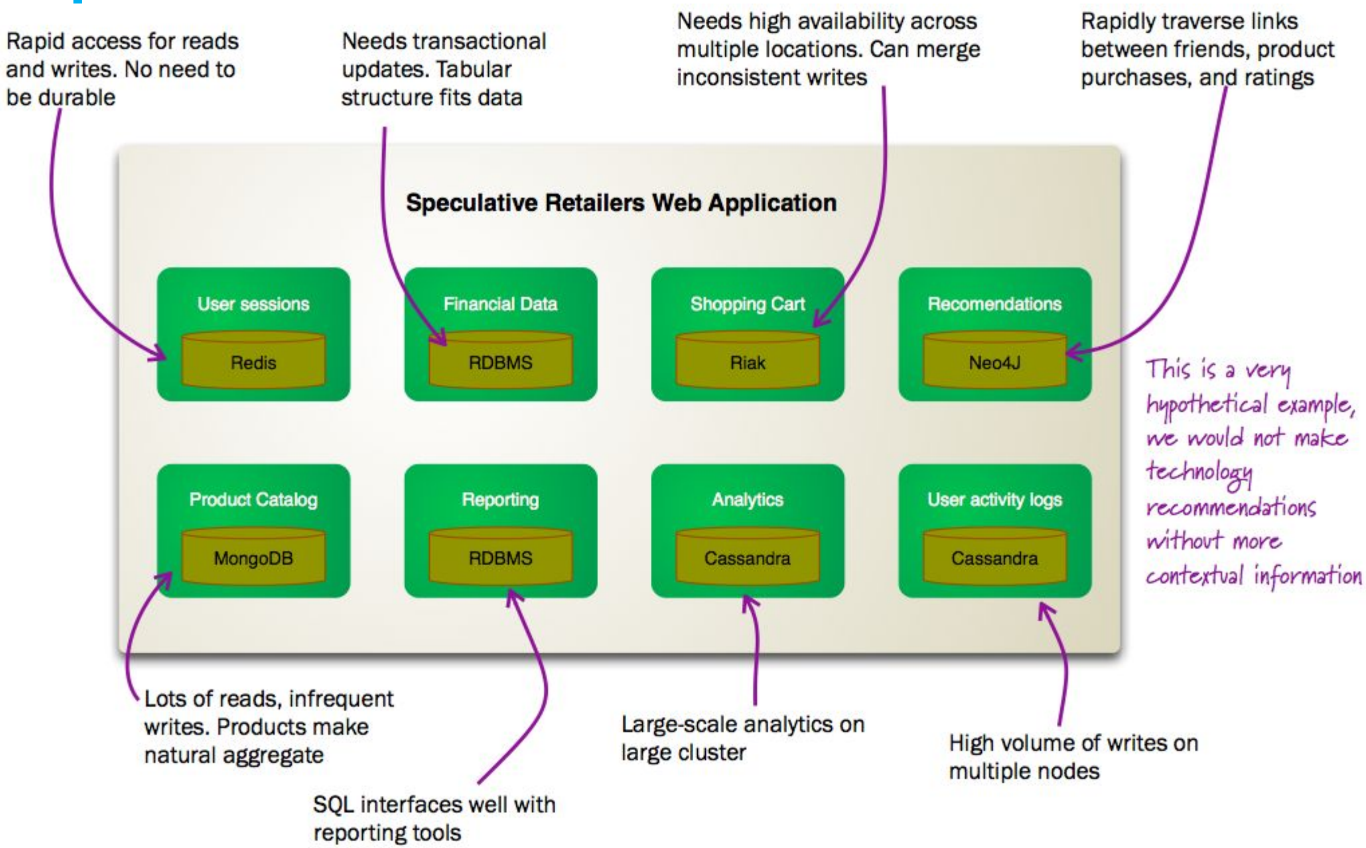
Big Data



Motivação

- Web Advertising
- Social media and real-time analytics
- Internet of Things (IoT)

Exemplos



NoSQL Distilled (http://martinfowler.com/bliki/NosqlDistilled.html)

Object Databases

Object Databases

- Information represented as objects, as in object-oriented programming
  - Support for hierarchical relations (i.e. an object can de defined as a subclass of another)
  - Direct relation between objects (e.g., Person will contain a pointer to an Address in memory, rather than a foreign key to join the two tables)

Examples: Objectivity/DB, Perst, ZopeDB, db4o, GemStone/S, InterSystems Caché, JADE, ObjectDatabase++, ObjectDB, ObjectStore, ODABA, Realm, OpenLink Virtuoso, Versant Object Database, ZODB

IndexedDB (API)

- IndexedDB is an indexed key-object database
- Provides a way for you to persist data in a browser. It can store files/blobs, JSON data and iterate over its indexes in local storage
- Available since Firefox 4, Chrome 11, IE10, Safari 8, and Edge 12
- Web SQL Database was a prior API developed by Apple using SQLite. But Firefox argued against it becoming the standard for browsers because it would codify the *quirks* of SQLite. Web SQL was deprecated in favor of IndexedDB
- Widely supported and stable, version 3.0 is now a First Public Working Draft

Key–Value Stores

Key–Value Stores

- Data is stored in associative arrays (a.k.a. dictionaries or hash maps)
  - Data records contain fields which can vary from record to record
  - Records are stored and retrieved by a <u>key</u> that identifies the record
- Often use less memory than relational databases, as optional values are not represented by placeholders or input parameters
- Data is a single opaque collection—cannot be queried
- Variants: ordered KV stores, distributed KV stores

Examples: Azure Cosmos DB, ArangoDB, Amazon DynamoDB, Aerospike, Couchbase, Oracle NoSQL Database, Riak, Voldemort, FoundationDB, InfinityDB, LMDB, MemcacheDB

Remote Dictionary Server

redis

- Open-source KV Store developed by Redis (Labs)
- Written in the C programming language
- Data structure store, used as a distributed, in-memory key–value database, cache and message broker, with optional durability.
- Fields: Used to store data at Twitter, GitHub, Snapchat, Cragslist, StackOverflow, etc.

●▲★

```
redis> PING
"PONG"
redis> HSET user:1 name antirez vocation artist
(integer) 2
redis> SET e 2.71
"OK"
redis> INCRBYFLOAT e 0.43
"3.14"
redis> RENAME e pi
"OK"
redis>
```

https://redis.io/

TÉCNICO LISBOA

RocksDB

- Open-source persistent KV Store developed by Facebook
- Written in the C++ programming language, it was forked from Google's open source LevelDB, to to exploit fast storage (SSDs)
- High performance <u>embedded</u> database for key-value data.
- Fields: operations monitoring, application metrics, Internet of Things sensor data, and real-time analytics

```cpp
#include <assert>
#include "rocksdb/db.h"

rocksdb::DB* db;
rocksdb::Options options;
options.create_if_missing = true;
rocksdb::Status status =
  rocksdb::DB::Open(options, "/tmp/testdb", &db);
assert(status.ok());
...
```

```cpp
std::string value;
rocksdb::Status s = db->Get(rocksdb::ReadOptions(), key1, &value);
if (s.ok()) s = db->Put(rocksdb::WriteOptions(), key2, value);
if (s.ok()) s = db->Delete(rocksdb::WriteOptions(), key1);
```

Wide-Column Stores

Wide-Column Stores

- Use tables, rows, and columns, but unlike a relational database, the names and format of the columns can vary from row to row in the same table
  - Can be interpreted as a two-dimensional KV store

Examples: Azure Cosmos DB, Amazon DynamoDB, Bigtable, Cassandra, Google Cloud Datastore, HBase, Hypertable, ScyllaDB

Apache Cassandra

- Free and open-source, distributed, wide-column store
- Originally designed for Facebook to enable search in Inbox
- Designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure
- Support for clusters spanning multiple datacenters, with asynchronous masterless replication allowing low latency operations for all clients
- Cassandra Query Language, heavily inspired by SQL

TÉCNICO LISBOA

Time Series Databases (TSDB)

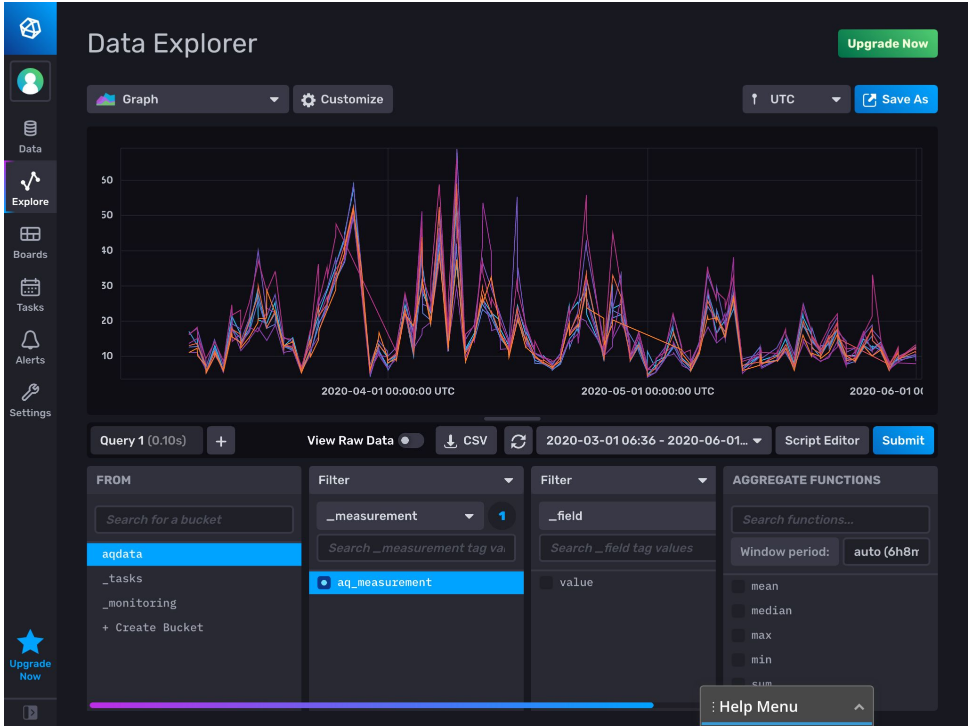
Time Series Databases

- Database management systems optimized for storing and serving time series through associated pairs of time(s) and value(s)

Examples: Apache IoTDB, eXtremeDB, InfluxDB, MongoDB, Prometheus, TimescaleDB

influxdb

- Open-source TSDB developed by InfluxData
- Written in the Go programming language
- Storage and retrieval of time series data
- Fields: operations monitoring, application metrics, Internet of Things sensor data, and real-time analytics



https://github.com/influxdata/influxdb

TÉCNICO LISBOA

Timescale

- Open-source TSDB developed by Timescale Inc
- Written in C and extends PostgreSQL
- Supports standard SQL queries and is a relational database
- Additional SQL functions and table structures provide support for time series data oriented towards storage, performance, and analysis facilities for data-at-scale
- Time-based data partitioning provides for improved query execution and performance when used for time oriented applications.

https://github.com/timescale/timescaledb

TÉCNICO LISBOA

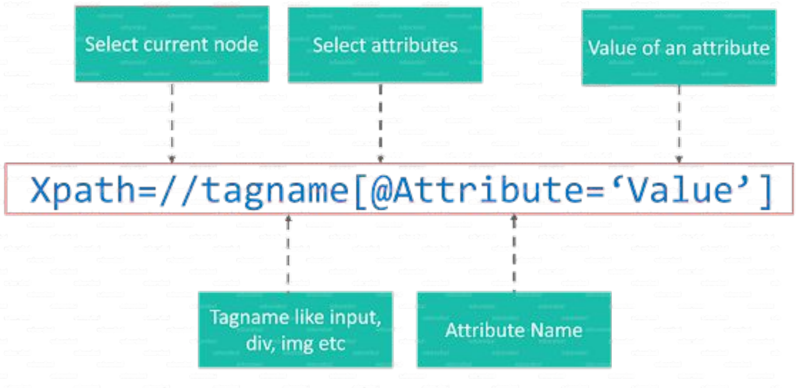
Document Stores

Document Stores

- Data storage system designed for storing, retrieving and managing document-oriented information, also known as semi-structured data
  - Inherently a KV store where the data records are documents
- Key difference from a traditional KV store is that metadata can be extracted from the internal structure of the document for the database engine to uses for optimization
  - Data is not fully opaque
- Document encodings in use include: XML, YAML, JSON, as well as binary forms of JSON like BSON

Examples: Azure Cosmos DB, ArangoDB, BaseX, Clusterpoint, Couchbase, CouchDB, DocumentDB, eXist-db, IBM Domino, MarkLogic, MongoDB, Qizx, RethinkDB, Elasticsearch, OrientDB eXistdb

- Open source NoSQL database management system built on XML technology
- It is both a NoSQL document-oriented database and a native XML database
- Provides support for XML, JSON, HTML and binary blobs
- Provides XQuery/XPath and XSLT as its query languages (Not SQL)



https://github.com/exist-db/exist/

mongoDB

- Source-available cross-platform document-oriented database management system developed by MongoDB Inc. and licensed under the Server Side Public License (not quite public/free)
- NoSQL document-oriented database system based on JSON-like documents with optional schemas
- Ad-hoc querying with support for field, range query, and regular-expression searches
- Very popular because it also provides Full-text Search

MongoDB vs. SQL

SQL Terms/Concepts	MongoDB Terms/Concepts
database	database
table	collection
row	document or BSON document
column	field
index	index
table joins	`$lookup`, embedded documents
primary key Specify any unique column or column combination as primary key.	primary key In MongoDB, the primary key is automatically set to the _id field.
aggregation (e.g. group by)	aggregation pipeline See the SQL to Aggregation Mapping Chart.

http://docs.mongodb.org/manual/reference/sql-comparison/

Distributed Filesystems

Distributed (Clustered) File Systems

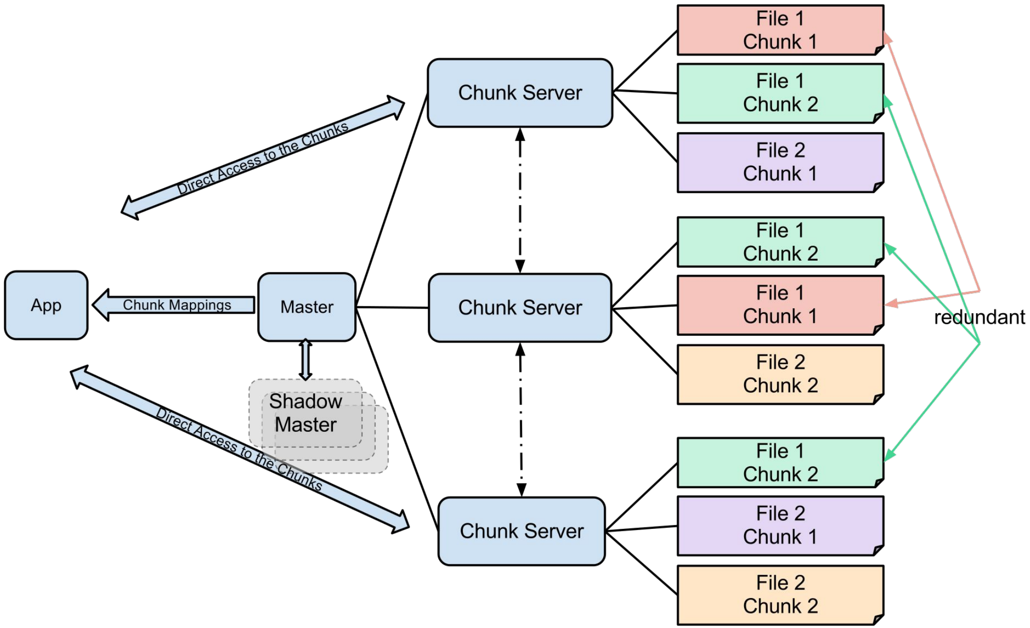
- File system which is shared by being simultaneously mounted on multiple servers and is accessed through a network protocol

- Can restrict access to the file system depending on access lists or capabilities on both the servers and the clients, depending on the protocol design.

Examples: Apache Hadoop, Distributed File System (Microsoft), Google File System & Google Colossus

TÉCNICO LISBOA

Google File System (GFS)



Ghemawat, S.; Gobioff, H.; Leung, S. T. (2003). "The Google file system". doi:10.1145/945445.945450.

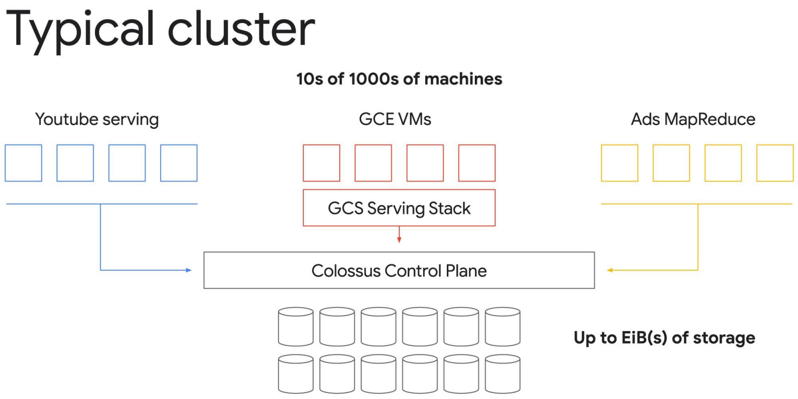
https://en.wikipedia.org/wiki/Google_File_System

TÉCNICO LISBOA

Google Colossus

Successor of the Google File System

- Example:
  - Instances accessing Cloud Storage from Compute Engine VMs, YouTube serving nodes, and Ads MapReduce nodes all share the same file system
  - Shared storage pool that is managed by the Colossus control plane provides the illusion that each has its own isolated file system.

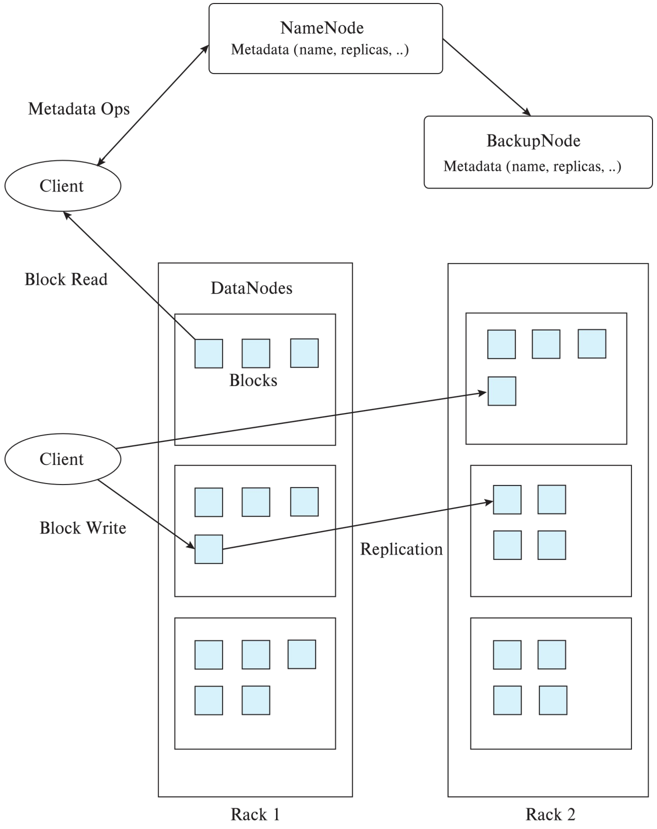


https://cloud.google.com/blog/products/storage-data-transfer/a-peek-behind-colossus-googles-file-system

TÉCNICO LISBOA

HDFS (Apache Hadoop)

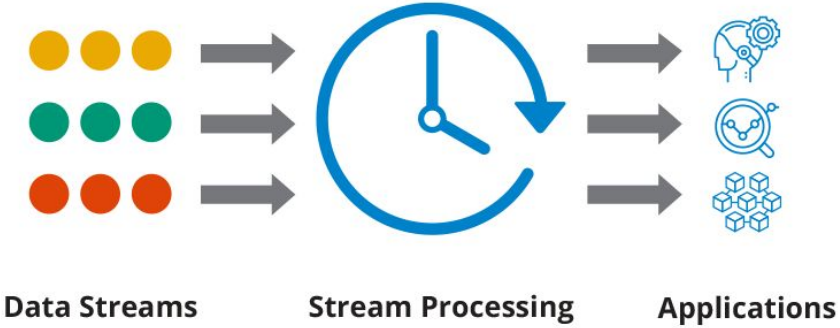
- NameNode
  - Maps a filename to list of Block IDs
  - Maps each Block ID to DataNodes containing a replica of the block
- DataNode: Maps a Block ID to a physical location on disk
- **Data Coherence**
  - **Write-once-read-many access model**
  - **Client can only append to existing files**



Streaming Databases

Streaming Databases

- Data stores designed to collect, process, and/or enrich an incoming series of data points (i.e., a data stream) in "real time"
- Not a discrete class of DBMS
    - Can be implemented in NoSQL databases, NewSQL databases, time-series databases, in-memory databases, or in-memory data grids

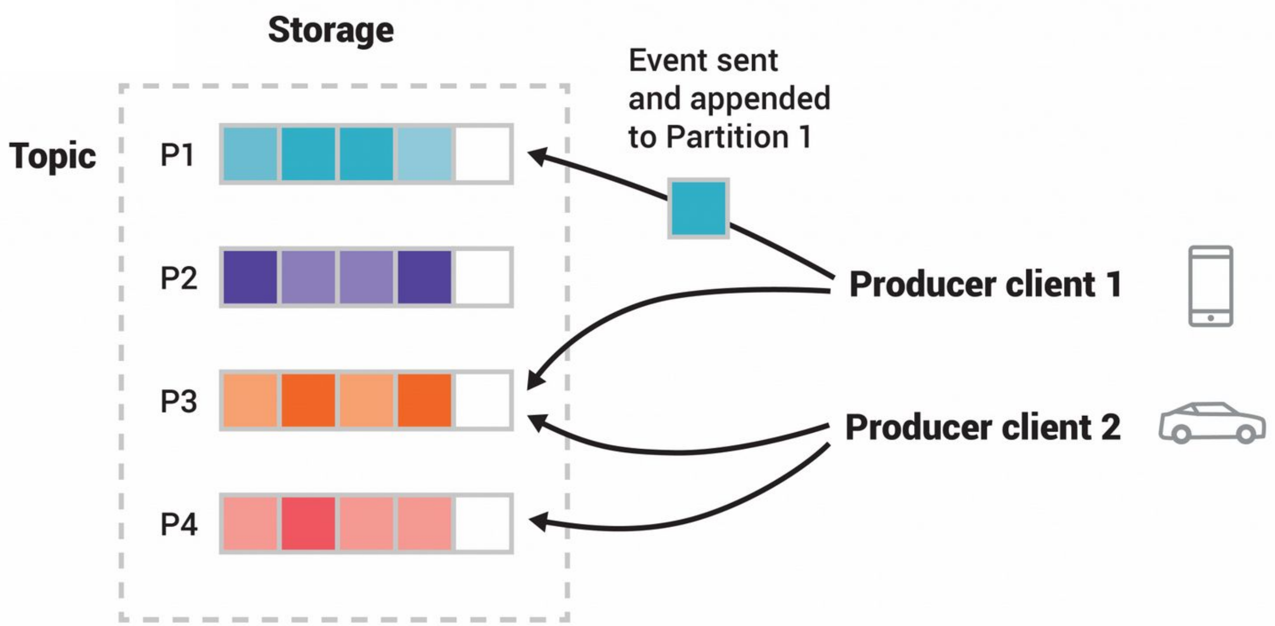


TÉCNICO LISBOA

- Distributed event streaming platform
- Pub/sub platform at its core, a messaging framework commonly used for asynchronous communication between services
- Kafka combines three key capabilities so you can implement your use cases for event streaming end-to-end with a single battle-tested solution:
  - To **pub**lish (write) and **sub**scribe to (read) streams of events, including continuous import/export of your data from other systems.
  - To store streams of events durably and reliably for as long as you want.
  - To process streams of events as they occur or retrospectively.

https://kafka.apache.org/intro

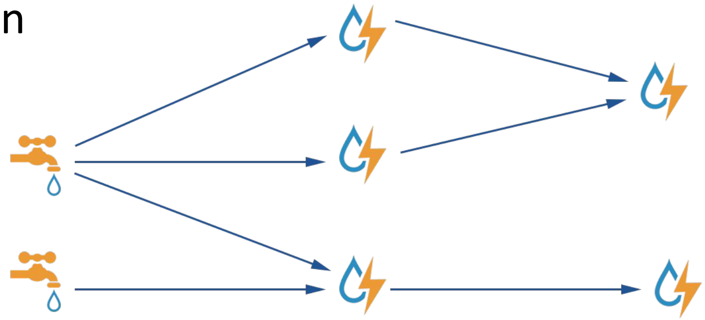
Apache Kafka



https://kafka.apache.org/intro

Apache Storm

- Distributed stream processing computation framework
- Open sourced after acquisition by Twitter
- A Storm application is designed as a "topology" in the shape of a directed acyclic graph with spouts and bolts acting as the graph vertices
    - Edges on the graph are named streams and direct data from one node to another
    - Together, the topology acts as a data transformation pipeline



- Data processing engine to store and process data in real-time across many machines in a cluster using simple constructs akin to MapReduce
- Performs both batch processing as well as stream processing
- Spark uses the concept of Resilient Distributed Datasets (RDDs) to reduce the chances of failures and supports many interfaces
    - Batteries included: Spark implementations of most popular algorithms
    - SPark SQL: SQL queries
    - MLlib for machine learning
-

Apache Spark

TÉCNICO LISBOA

Graph Databases

Graph Databases

- Data is represented as nodes, edges (or relations) and properties (or attributes)
- **Labeled-property graph**
  - Set of nodes, relationships, properties, and labels. Both nodes of data and their relationships are named and can store properties represented by key–value pairs
- **RDF graph / Knowledge Graph**
  - Set of nodes and arcs, under the RDF data model

Examples: Azure Cosmos DB, AllegroGraph, Amazon Neptune ArangoDB, InfiniteGraph, Ontotext GraphDB, MarkLogic, Neo4J, OrientDB, Oracle Database, Virtuoso

Resource Description Framework

- A W3C standard originally designed as a data model for metadata
- Adopted as a general method for description and exchange of graph data
- Directed graph composed of triple XML statements
  - <subject> <predicate> <object>

    OR
  - <subject> <predicate> literal object
- Simple and flexible data model but with a lot of expressive power
- The basis for the Semantic Web languages RDFS and OWL that add semantics on top of the RDF data model

TÉCNICO LISBOA

Triplestores

- Data is stored and retrieved in the form of RDF triples
- It can be queried semantically with the use of specialized query languages (e.g. SPARQL)
- Edge-centric view of RDF graph data
- Data can be perceived as rows in a single three-column table
- Lack features of “proper” RDF graph databases such as index-free adjacency (i.e. nodes keep direct pointers to their neighbors)

Graph Query Languages

SQL (Relational Database)

```
SELECT p2.name FROM people p1 JOIN friend USING (person_id) JOIN people p2
ON (p2.person_id = friend_id) WHERE p1.name = 'Jack';
```

[Cypher](#) (Property Graph)

```
MATCH (p1:person {name: 'Jack'})-[:FRIEND_WITH]-(p2:person)
RETURN p2.name
```

[SPARQL](#) (RDF Graph)

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name
WHERE { ?s foaf:name      "Jack" ;
           foaf:knows     ?o .
           ?o foaf:name   ?name .   }
```

TÉCNICO LISBOA

neo4j

- Graph database management system developed by Neo4j, Inc. and available in a non-open-source "community edition" licensed with a modification of the GNU General Public License
- ACID-compliant transactional database with native graph storage and processing
    - Stores nodes, edges connecting them, and attributes of nodes and edges
- Implemented in Java, but accessible from software written in other languages using the Cypher query language (via a transactional HTTP endpoint) or through the binary "Bolt" protocol

Search

elasticsearch

- Source-available cross-platform distributed, multitenant-capable full-text search engine (not quite public/free). Note: Amazon provides a fully open-source version (OpenSearch).
- Information retrieval (IR) systems use a simpler data model than database systems
    - Information organized as a collection of documents
    - Documents have no schema (i.e., can upload JSON with fields title and content)
    - Core data structure is the Inverted Index (i.e., índice remissivo de palavras)
- Locates relevant documents, on the basis of user input such as keywords or example documents (e.g., find documents containing the words “database systems”)
- Web search engines are the most familiar example of IR systems
    - Google uses 300+ signals on top of an initial retrieval stage (e.g., PageRank, LLMs)

TÉCNICO LISBOA