

# Architectures for massive data management

## Key-value stores and Redis

**Ioana Manolescu , Silviu Maniu**

INRIA Saclay & Ecole Polytechnique

Université Paris-Sud

[ioana.manolescu@inria.fr](mailto:ioana.manolescu@inria.fr) , [silviu.maniu@lri.fr](mailto:silviu.maniu@lri.fr)

<http://pages.saclay.inria.fr/ioana.manolescu/>

<https://silviu.maniu.info/>

M2 Data and Knowledge, 2017/2018

Université de Paris Saclay

# Key-value stores

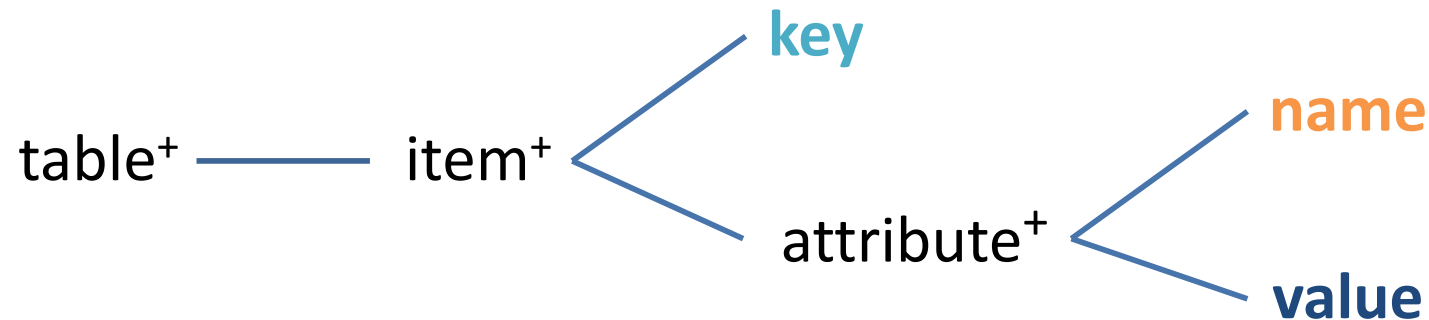
- Relatively recent class of systems, developed as part of the « NoSQL » movement
- Main idea:
  - Trade simplicity for speed and scale
- Extremely simple data model
  - **key**=short byte sequence / integer
  - **value**=byte sequence (may recognize integers)
- No QL. Operations: **PUT(k, v)** and **GET(k,v)**
- **ACID** properties depending on the system; at least atomic PUT and GET
  - Some are in-memory thus no durability at all

# Key-value data models

- Simplest model:
  - One key – one value
- Extensions:
  - **Organization:** key-value pairs belong to « collections » or « databases » or « tables »
  - **Multiplicity:** set or list of values
  - **Internal structure:**
    - One key – a list of *attributes*
    - Each attribute has a *name* and a *value / set of values*

# Sample key-value data model: DynamoDB

- Provided by Amazon Web Services (AWS)



- Naming may vary (there is no standard). See doc.
- Although it is called « table », *items in the same table may have nothing in common!*
- The interface is very similar to the so-called « Big Tables » (to be seen)

# Redis: one of the most popular key-value stores

- **Data model:**
  - Hash (a set of key-value pairs on the same key)
  - List
  - Set
  - Values cannot be lists nor sets (no nesting!)
  - Databases
- **Operations:**
  - Put, get
  - Set operations (union, intersection)
  - List operations: left/right push/pop (→ queue / stack)
  - Arithmetic operations (attempts type conversion to integers)

# Redis: first lab

- Install Redis, launch it
- Follow the tutorial
- Write in Redis a toy application of a database of books which can be borrowed in a library.
  - All books have an ISBN, a title and an author.
  - Books may also have other properties, e.g. language, publication year, edition...
  - Make books expire after a while (if no one borrows the book)
  - Make a Redis client subscribe to books by a certain author
  - **(Task)** With the help of a programming language interface to Redis, index a newly published item by the words in its description. Make a Redis client subscribe to items having certain keywords.