[ ] Aisha Siddiqa, Ahmad Karim, Abdullah Gani. Big data storage technologies: a survey.

Frontiers of Information Technology & Electronic Engineering, 2017, **18**(8):1040-1070.

https://doi.org/10.1631/FITEE.1500441

Big data storage technologies: a survey

Key words: Big data; Big data storage; NoSQL databases; Distributed databases; CAP theorem; Scalability; Consistency-partition resilience; Availability-partition resilience

Corresponding author: Aisha SIDDIQA
E-mail: aasiddiqa@gmail.com
ORCID: http://orcid.org/0000-0002-1016-758X

Motivation/Main idea

- The structural shift of the storage mechanism from traditional data management systems to NoSQL technology is due to the intention of fulfilling big data storage requirements, i.e., fast-growing volume, velocity, and diversity
- NoSQL technologies introduce flexible data models, horizontal scalability, and schema-less data models. These databases aim to provide ease in scalability and management of large-volume data
- However, consistency, scalability, and availability are still challenging issues in existing NoSQL solutions for big data
- The literature does not report an in-depth survey of storage technologies available for big data

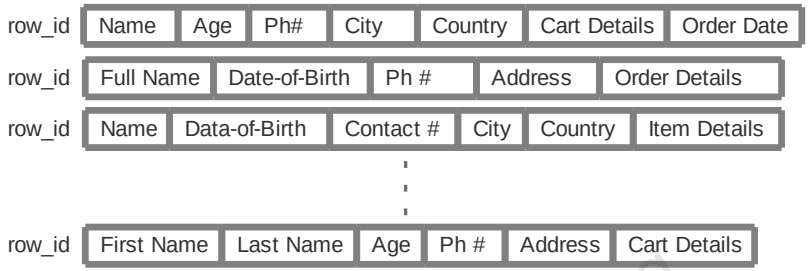
Contribution

- A comprehensive investigation of state-of-the-art storage technologies available for big data
- A well-defined taxonomy of big data storage technologies
- A comparison and analysis of the existing approaches using Brewer's CAP theorem
- A presentation of future research challenges

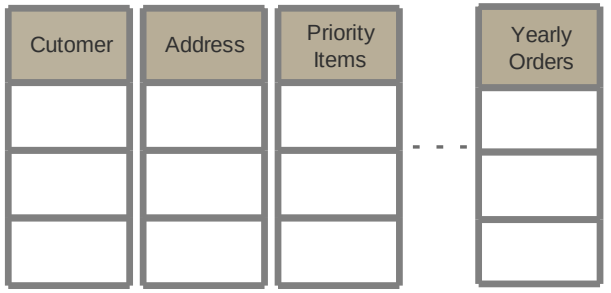
Table 1 SWOT analysis of relational databases and big data storage systems

	Traditional database systems	Big data storage systems
Strengths	Support highly structured data stored and processed over an auxiliary server Vertical scalability with extendible processing on a server Specialized data manipulation languages Specialized schema	Support heterogeneous structured data Horizontal scalability with extendible commodity servers Support data-intensive applications Simultaneous accessibility Reliability and high availability High fault tolerance Eventual consistency
Weaknesses	Performance bottleneck Processing delays Increased deadlocks with growth of data Limited storage and processing capacity Co-relations which hinder scalability Expensive join operations for multidimensional data	No compliance with ACID due to scalability and performance
Opportunities	Support complex queries Atomicity in complex transactions Built-in deployment support	Improved query response times Simplicity in storage structures Data-intensive
Threats	Extensive volume of data for storage with dynamic growth Frequently changing schema Complex data structures More concurrent access needs Frequent I/O needs Real-time processing needs Consistency of a large number of storage servers	Large number of small files Deployment may need community support

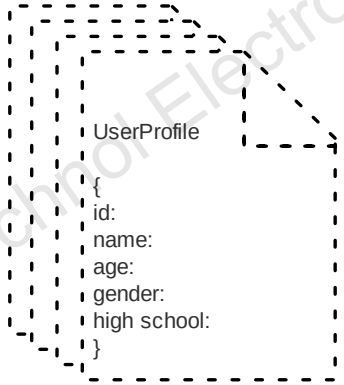
Data models



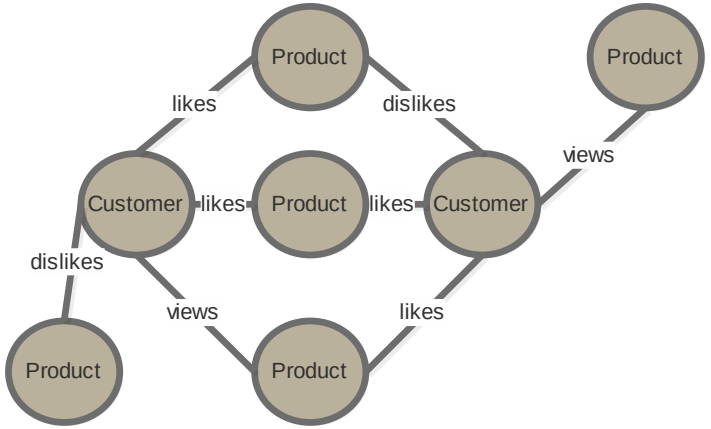
I. Key-value



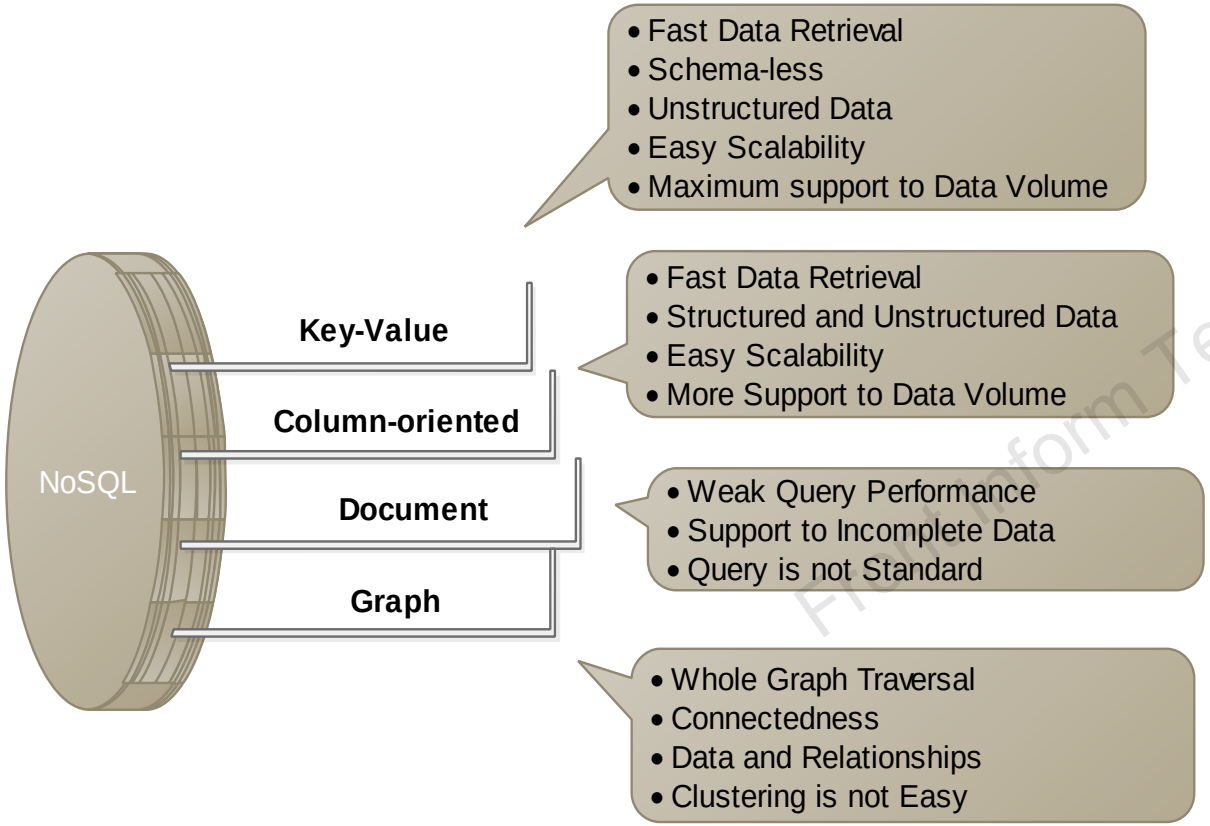
II. Column-Oriented



III. Document-Oriented



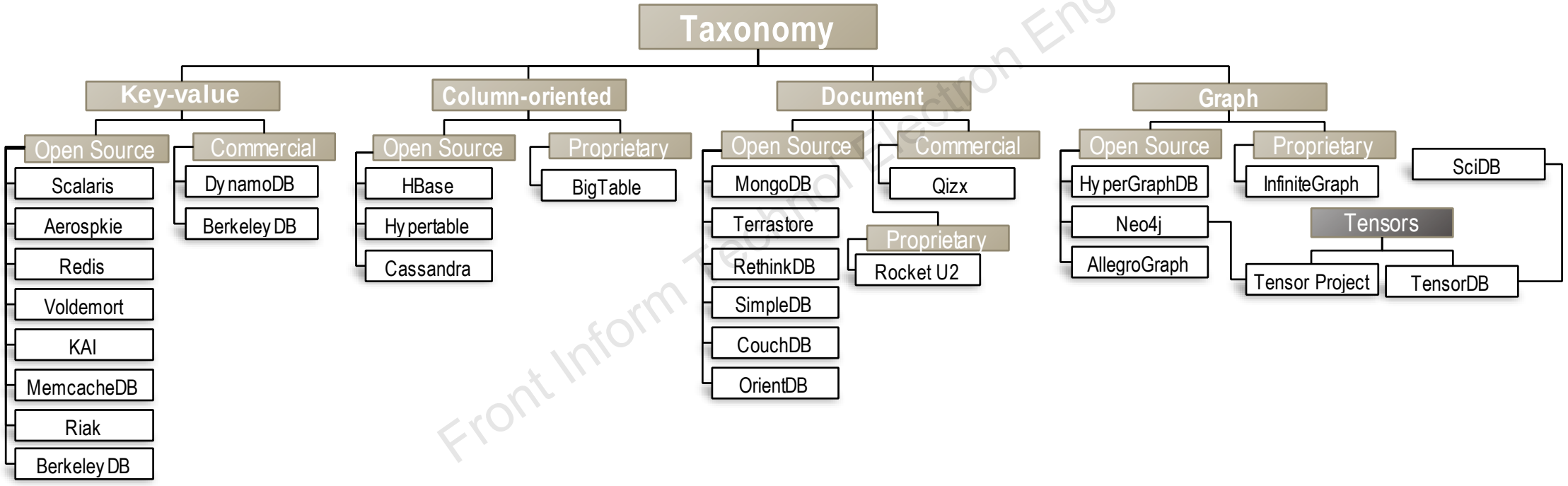
IV. Graph



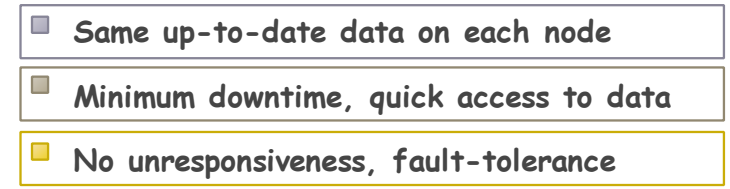
Examples of data models

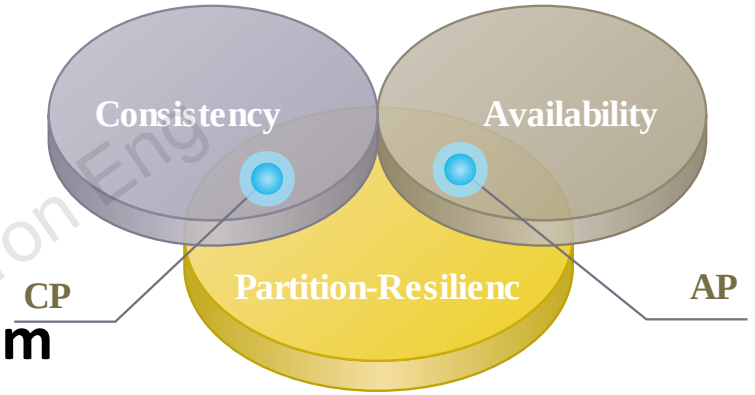
Data models of NoSQL databases

Taxonomy of big data storage technologies

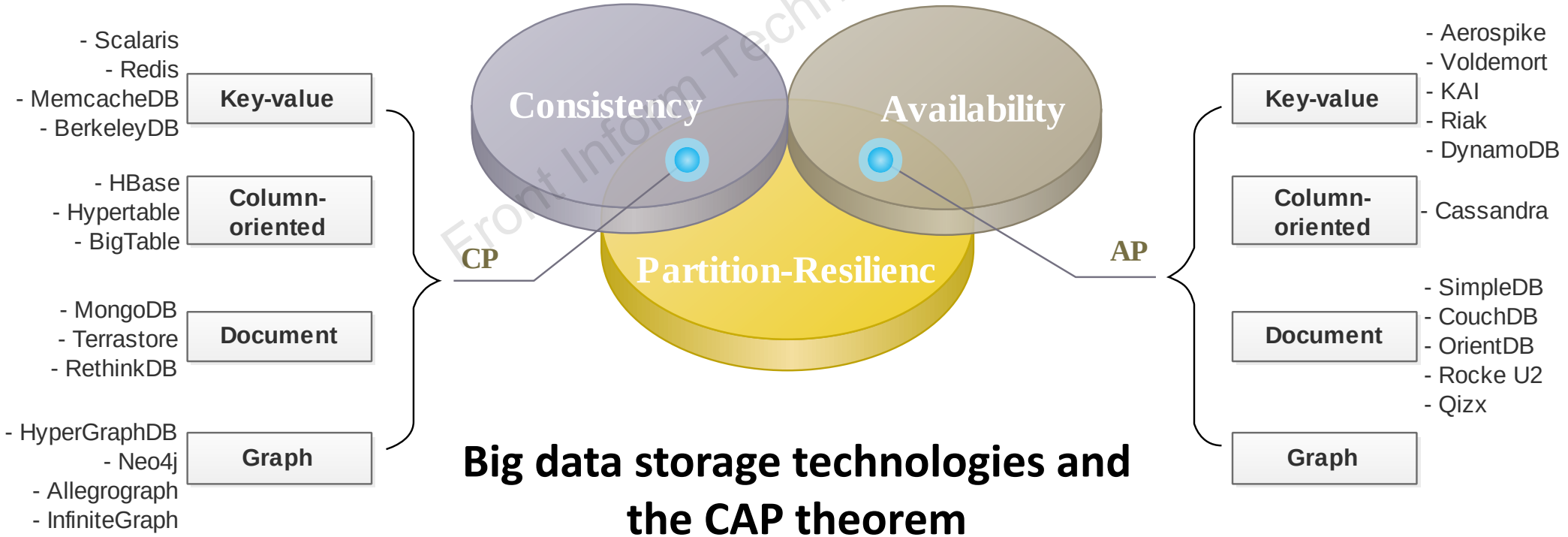


Analysis based on the CAP theorem





Brewer's CAP theorem



Big data storage technologies and the CAP theorem

Future research challenges

- **Frequent data update and schema change:** The update rate is mostly very high and the volume of data is growing very rapidly. In case of unstructured data, change in schema is also very common. Although available storage technologies are scalable, the need to be efficient in data updates and schema is still under consideration.
- **Partitioning method:** Horizontal or vertical partitioning is applied on data based on access patterns. Data may be required to be analyzed by the features or records. The prediction of access pattern might be wrong or change during execution. This poses a critical research challenge on existing data models.

Future research challenges Contd.

- **Replication factor:** Data are replicated over multiple sites to achieve fault tolerance and high availability to its users. Although this concept makes the storage very efficient to improve access performance, it compromises data consistency and does not suit in frequent data changing. This leads to poorer access performance if frequent consistency locks are experienced.
- **User expertise:** The user space is broadened by enterprises, so that users from different domains can execute queries on data according to their problems. To achieve improved performance, sometimes these databases integrate DBMS platforms, which undoubtedly meets the expectations, but the implementation and configuration process becomes complex for non-expert users.

Discussion

- NoSQL technologies, when analyzed using the CAP theorem, are either consistent or available. However, all of them have support to partition resilience.
- Although key-value databases are mostly suitable for unstructured data, structured data can be presented with these databases if record-based data retrieval and analysis needs to be performed on it.
- Most of the key-value and column-oriented databases have open source license
- DynamoDB is the only key-value database that is available commercially.
- Likewise, BigTable is the only column-oriented database that has proprietary license.
- Column-oriented databases such as HBase, Hypertable, Cassandra, and BigTable are mostly suitable for structured data with enough support to unstructured data.
- These databases apply vertical partitioning on the data and store each column as a separate chunk of data, and performing queries on attributes as well as attribute-based analysis of data is easier with this data model.

Discussion Cont’d.

- Document-oriented databases also have the key-value data structure. However, the value identifies a document instead of a record.
- Compared to the key-value data model, document databases have less support to scalability and unstructured data.
- Databases having document structure are, on average, prone to availability (i.e., MongoDB, Terrastore, and RethinkDB) and consistency (i.e., SimpleDB, CouchDB, OrientDB, Rocket U2, and Qizx).
- Graph databases are well-structured databases, where analyzing data as well as their relationships is significant.
- Although graph databases do not have good support to scalability and clustering, they offer complex data structures.
- All graph databases in this survey are CP-type systems.