Writing a Technical Document (I)

- Observation: Structure of a research paper (in computer science) is always the same.

- General structure
  - <Title>
  - <Affiliation>
  - Abstract
  - 1 Introduction
  - 2 Related Work
  - 3 … n <one or several chapters, “the meat of the paper”>
  - <n+1> Conclusions and Future Work
  - References

- General rule: Write from the reader’s perspective and not from your perspective

Writing a Technical Document (II)

- Title
  - should be interesting, catchy, and concise
  - should summarize the whole paper in a few lines

- Abstract
  - should summarize the whole paper in a single paragraph
  - should indicate the problems covered
  - should describe the goals aimed at
  - should sketch the solutions proposed

Writing a Technical Document (III)

- Introduction
  - Purpose: summarize the paper, attract the reader to further read the paper
  - First paragraph: introduce the paper from a larger perspective, name application areas that are relevant for the paper
  - Second paragraph: narrow down to the problems covered by this paper
  - Third paragraph: describe the goals of the paper
  - Fourth paragraph: indicate the solutions
  - Fifth paragraph: description of the contents of the next sections

Writing a Technical Document (IV)

- Related Work
  - Result of a literature study
  - Describe the state-of-the-art documented in the literature
  - Define literature categories
  - Classify the literature according to the categories
  - For each literature category
    - For each reference
      - Summarize it briefly
      - Describe the strengths but especially the weaknesses of its concepts
      - Compare the strengths and weaknesses with those of your concepts
      - Show how your new concepts solve the weaknesses

Writing a Technical Document (V)

- "The meat of the paper"
  - Usually presented in one or more sections
  - Carefully explains the concepts and solutions of the problems
  - Determines the scientific quality of the paper
  - Impossible to describe general principles
- Conclusions and Future Work
  - Conclusions
    - summarize what the paper was about
    - say what the reader should have learned from the paper
  - Future work
    - gives an outlook to extensions of the paper
    - indicates open, related research problems

Writing a Technical Document (VI)

- References
  - are the result of a literature study
  - list the most important literature that is relevant for the topic of the paper
  - are all formatted according to certain style descriptions
  - number depends
    - on the kind of paper (e.g., journal article, conference paper)
    - on the number of available pages

Advanced Database Systems (I)

Data model

- A collection of tools for describing
  - Data
  - Data relationships
  - Data semantics
  - Data constraints

- Examples of past and current data models
  - Relational model
  - Entity-Relationship data model (mainly for database design)
  - Object-based data models (object-oriented and object-relational)
  - Semi-structured data model (XML)
  - Other older models:
    - Network model
    - Hierarchical model

Advanced Database Systems (II)

Possible classification of advanced database systems (DBS) with respect to the underlying data model (in alphabetical order) provided by DB-Engines.com

- Content stores
- Document stores
- Event Stores
- Graph DBMS
- Key-value stores
- Multivalue DBMS
- Native XML DBMS
- Navigational DBMS
- Object oriented DBMS
- RDF stores
- Relational DBMS
- Search engines
- Time Series DBMS
- Wide column stores

[Source: DB-Engines.com, accessed on August 28, 2016]

Calculating Ranked Popularity Scores for DBS (I)

- Ranking of database systems measured by their popularity
- Measuring the popularity involves the following parameters
  - Number of mentions of the system on websites
    - Measured as number of results in search engine queries
    - Search engines used: Google, Bing, Yandex
    - In order to count only relevant results, the search string is “<system name> database”, e.g. "Oracle database".
  - General interest in the system
    - Frequency of searches in *Google Trends*.
  - Frequency of technical discussions about the system
    - Number of related questions and the number of interested users on the well-known IT-related Q&A sites *Stack Overflow* and *DBA Stack Exchange*

[Source: DB-Engines.com, accessed on August 28, 2016]

Calculating Ranked Popularity Scores for DBS (II)

- Measuring the popularity involves the following parameters (*continued*)
  - Number of job offers, in which the system is mentioned
    - Number of offers on the leading job search engines *Indeed* and *Simply Hired*
  - Number of profiles in professional networks in which the system is mentioned
    - Use of *LinkedIn* and *Upwork* as the internationally most popular professional networks.
  - Relevance in social networks
    - Number of *Twitter* tweets in which the system is mentioned.

[Source: DB-Engines.com, accessed on August 28, 2016]

Calculating Ranked Popularity Scores for DBS (III)

- Measuring the popularity of database systems
  - Standardizing and averaging of the individual parameters
  - Mathematical transformations are performed in a way so that the distance of the individual systems is preserved
    - Meaning: When system A has twice as large a value in the DB-Engines Ranking as system B, then it is twice as popular when averaged over the individual evaluation criteria.
  - The popularity score is always a relative value and should be interpreted in comparison with other systems only.
  - The DB-Engines Ranking does not measure the number of installations of the systems, or their use within IT systems
    - One can assume that an increase of the popularity of a system (e.g. in discussions or job offers) precedes a corresponding broad use of the system by a certain time factor.
    - Thus, the DB-Engines Ranking can act as an early indicator.

[Source: DB-Engines.com, accessed on August 28, 2016]

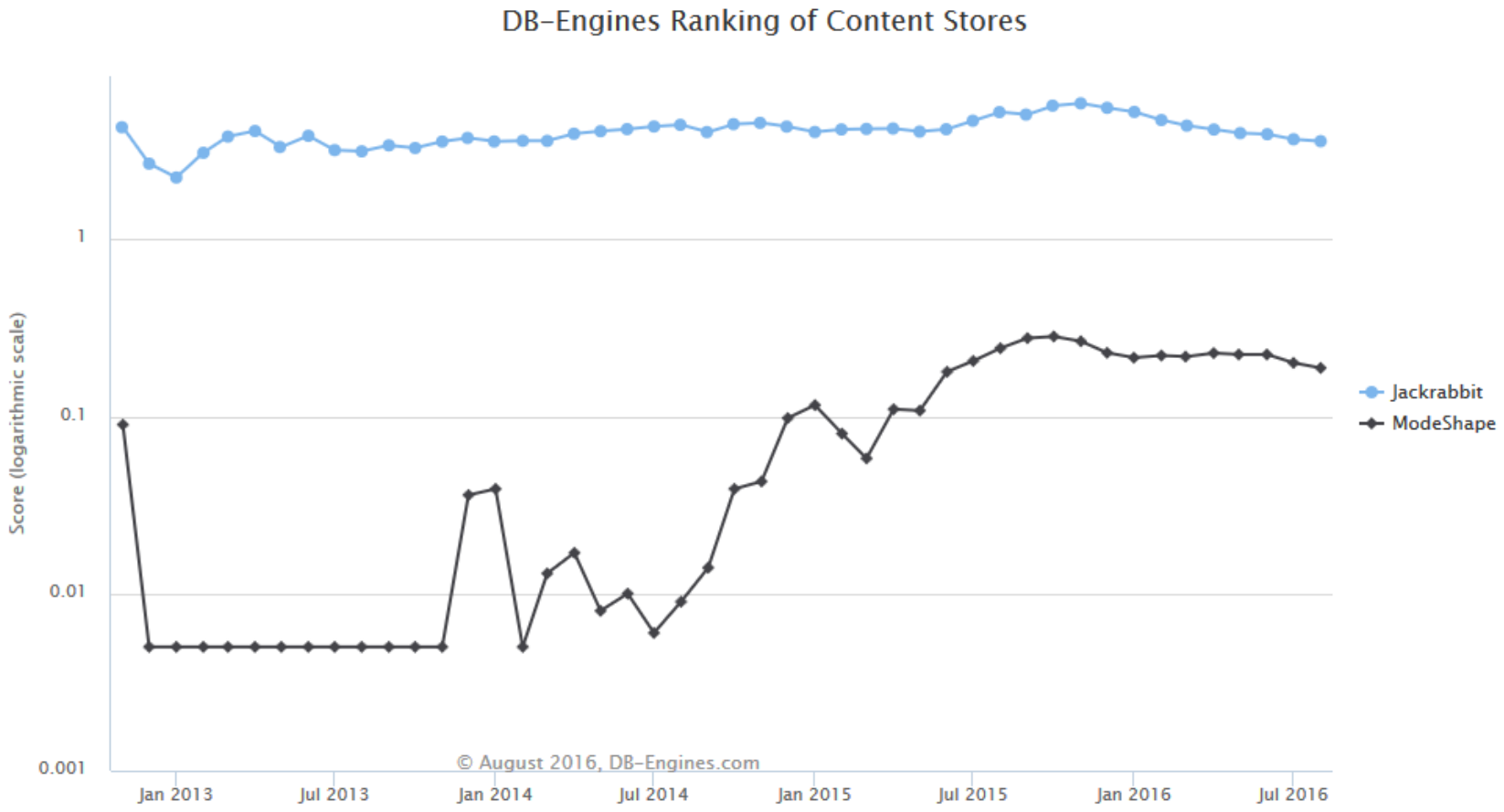
Content Stores (I)

- Also called content repositories
- A content repository is a hierarchical content repository with support for structured and unstructured content (text, pictures, videos), full text search, versioning, transactions, and observations.
- They store data plus metadata
- Full implementation of the Content Repository for Java Technology API (JCR)
- Designed as a foundation of modern world-class web sites and other demanding content applications

[Source: DB-Engines.com, accessed on August 28, 2016]

Content Stores (II)

- 2 Systems in ranking



[Source: DB-Engines.com, accessed on August 28, 2016]

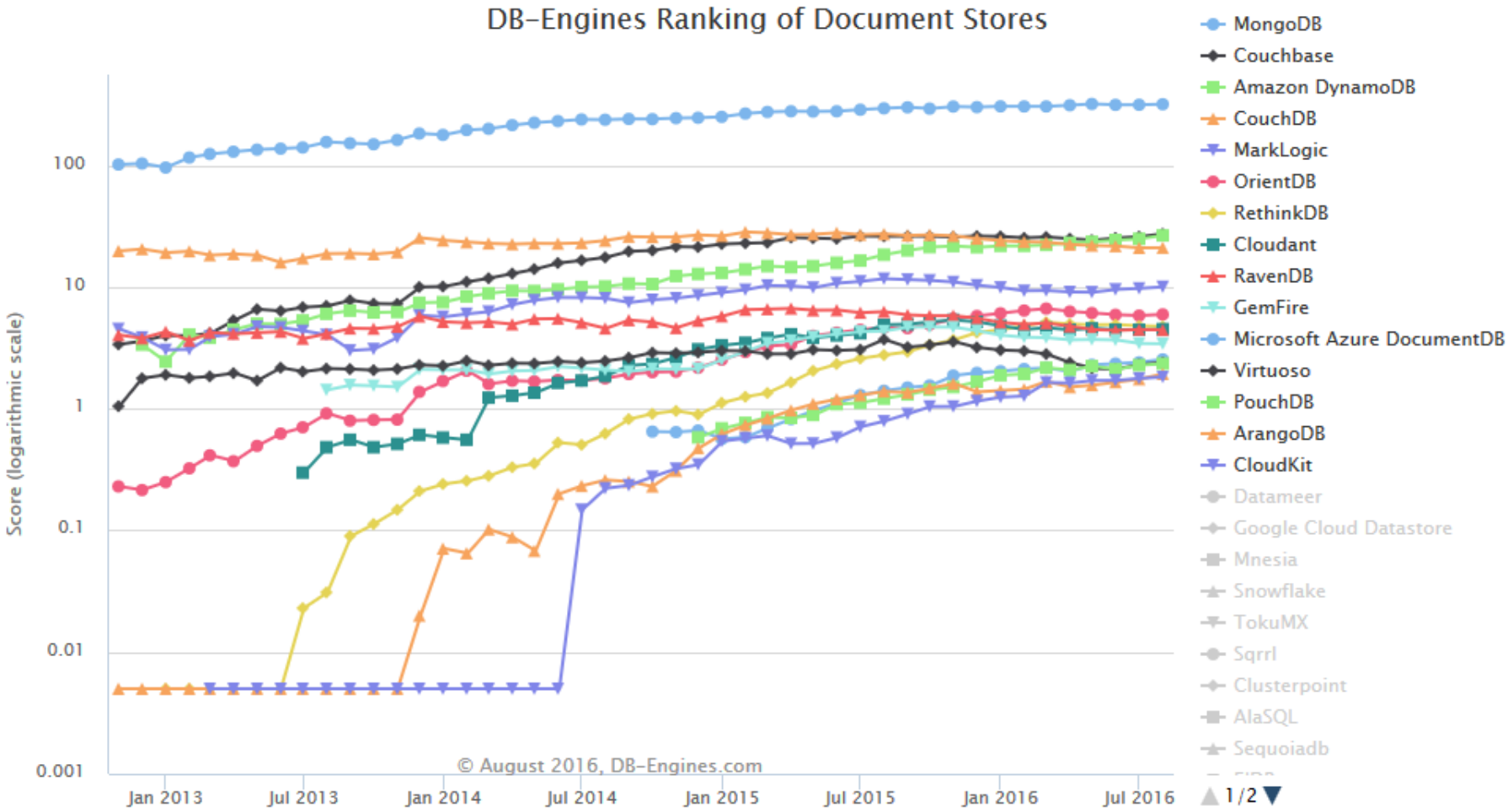
Document Stores (I)

- Also called document-oriented database systems
- Characterized by their schema-free organization of data
  - Records do not need to have a uniform structure, i.e., different records may have different columns.
  - The types of the values of individual columns can be different for each record.
  - Columns can have more than one value (arrays).
  - Records can have a nested structure.
- Document stores often use internal notations, which can be processed directly in applications, mostly JSON
- JSON (JavaScript Object Notation) is an open-standard format that uses human-readable text to transmit data objects consisting of attribute-value pairs.
- NoSQL databases: no or limited use of SQL, no table-based, relational database structure

[Source: DB-Engines.com, accessed on August 28, 2016]

Document Stores (II)

- 42 systems in ranking, only the first 15 systems are shown graphically



[Source: DB-Engines.com, accessed on August 28, 2016]