Advanced Topics in Databases

Exercise 9

1. Follow up of Exercise 8 Task 9

   (a) Give the set of key names (1st level properties (no nested object), no duplicate key names) for all research papers stored in the database! You may use map-reduce for this purpose! Additionally, give your query statement.

2. Despite differences in their data model, document stores share many conceptual ideas and internal structures with relational systems. However, due to a focus on scalability that heavily benefit from the denormalization of stored records, document stores typically have significant differences at storage level.

   (a) **[Group 10]** Consider the append-only storage of CouchDB, and the update-in-place storage of MongoDB. Explain how database modifications (inserts and updates) are handled, and discuss benefits and drawbacks of each approach.

   (b) **[Group 11]** Recap the default storage engine of MongoDB since version 3.2, WiredTiger. Explain where B+-tree structures are used.

3. Document records are physically organized by a variety of data formats, that result from a diversity of applications (such as fast parsability, understandability, or expressibility).

   (a) **[Group 12]** Justify the statement "There is no One-Size-Fits-All format for representation of semi-structured data" w.r.t. the diversity of application requirements, and discuss this statement for at least two formats not already mentioned in the lecture.

   (b) Clone the libcarbon repository from GitHub [1], and checkout the branch teaching/atdb/2019. In this branch you will find the directory ds/, which contains excerpts of pre-processed datasets. The GitHub Repository API Excerpt dataset is the one you will work with. Analyze the (disk) size requirements for the following formats on the GitHub Repository API dataset snapshot (that you must download from our FTP server) by converting the snapshot into each format, and compare them to the plain-text JSON format.

      - Universal Binary JSON (UBJSON)
        Tip: Find a library or tool under ubjson.org/libraries/ that matches your preferences
      - Binary JSON (BSON)
        Tip: Study the toolchain of MongoDB, which provides an export to BSON

[1] $ git clone https://github.com/protolabs/libcarbon.git && cd libcarbon &&
git checkout -b teaching/atdb/2019 origin/teaching/atdb/2019

- Columnar Binary JSON(CARBON)
  Tip: Build carbon-tool from sources in the branch teaching/atdb/2019 from our repository github.com/protolabs/libcarbon (see README.md), and run in your bash:

  $ build/carbon-tool convert –size-optimized –no-string-id-index github-repo-api.carbon ds/github-repo-api/snapshot-excerpt.json
  (conversion with carbon-tool may take XXX min for this dataset)

Tip: Use a Linux distribution or macOS as operating system to match the building tools and tool chains.

(c) **[Group 15]** Revisit your results from sub task 2 for each format (including the given plain-text JSON format). Speculate on the reason why your see differences in the file sizes for each format.

Good Luck!