University of Zurich^UZH

Department of Informatics

University of Zürich
Department of Informatics
Binzmühlestr. 14
CH-8050 Zürich
Phone. +41 44 635 43 11
Fax +41 44 635 68 09
www.ifi.uzh.ch/dbtg

UZH, Dept. of Informatics, Binzmühlestr. 14, CH-8050 Zürich

Prof. Dr. Michael Böhlen
Professor
Phone +41 44 635 43 33
Fax +41 44 635 68 09
boehlen@ifi.uzh.ch

Zürich, 19. Februar 2019

MSc Basismodul
Topic: Implementation of RPP indexes in Apache Jackrabbit Oak

Hierarchical databases come with property-and-path (PP) indexes that include information about not only values but also the structure of the data. Paths that index the same property and value appear in distinct parts of the PP index and can be updated concurrently. This can lead to path conflicts and transaction aborts when insertions/deletions into different paths propagate to common ancestor nodes.

The Robust PP (RPP) [2] index was proposed to reduce the number of path conflicts and their ensuing transaction aborts. An RPP index identifies *volatile nodes* in the index, i.e., nodes that are repeatedly inserted/deleted and which are a frequent source of path conflicts between concurrent transactions. Volatile nodes are not deleted from an RPP index anymore to avoid concurrent writes to the same nodes. This reduces the number of path conflicts, but grows the size of the index. As a result, the query time is increased. This tradeoff can be controlled with the volatility threshold τ, which determines after how many insertions and deletions of a node it is classified as volatile. The smaller τ, the quicker a node is classified as volatile and more path conflicts can be prevented. On the other hand, query times increase for small τ, because the index is bigger.

The goal of this project is to study and implement RPP indexes. In an experimental evaluation the impact of volatility threshold τ on the abort ratio and the query performance should be studied. RPP indexes should be implemented in the hierarchical distributed database system Apache Jackrabbit Oak [1].

Tasks

1. Literature study on PP and RPP indexes [2].
2. Implement an RPP index in Apache Jackrabbit Oak [1].
   - Implement the volatility computation for the document node store.
   - Implement the RPP index. An RPP index on property k must support the following interface:
     - `insert(value, path)`
     - `delete(value, path)`
     - `query(value, path)`
3. Evaluate your implementation of an RPP index experimentally.
   - Use the DELL dataset provided to you.
   - Evaluate the abort ratio in terms of the volatility threshold τ.
   - Evaluate the query performance in terms of the volatility threshold τ.
4. Summarize your findings in a short report.

Optional Tasks

1. Execute the following additional experiments
   - Evaluate the throughput of your RPP index implementation.
   - Evaluate the RPP index for different write and read skews.

References

[1] Apache. Apache Jackrabbit Oak. `https://jackrabbit.apache.org/oak/`, 2018. [Online; accessed November 2018].

[2] K. Wellenzohn, M. Böhlen, S. Helmer, M. Reutegger, and S. Sakr. Workload-Aware Contention-Management in Indexes for Hierarchical Data. to be published.

Supervisor: Kevin Wellenzohn (wellenzohn@ifi.uzh.ch)

Oral exam date: 02.04.2019, 3pm

University of Zurich
Department of Informatics

Prof. Dr. Michael Böhlen
Professor