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

Timothy Pescatore

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

Zürich, 15 March 2018

Bachelor's Thesis: Integration of Ongoing Integers into PostgreSQL

Data that are associated with a valid time interval are common in applications that deal with, e.g., employment contracts and software bugs. Ongoing time points such as now are used to denote that a contract is valid from the start point onward. The duration of valid time intervals is used to determine how long an employee works at a company or how long a software bug is open since its discovery.

Ongoing time points change as time passes by. For instance, ongoing time point now instantiates to time point $2018/2$ at reference time $2018/2$ and to time point $2018/3$ at reference time $2018/3$. Thus, the duration of ongoing time intervals, i.e., intervals whose start or end point is ongoing, changes as time passes by. For instance, the duration of ongoing time interval $[2018/1, now)$ is 1 at reference time $2018/2$ and 2 at reference time $2018/3$. To get results that remain valid as time passes by, we represent the result of the duration as an *ongoing integer*. The duration of $[2018/1, now)$ results in ongoing integer $\{\mathrm{s}[(0 \nrightarrow 0, +0) \text{ at } [-\infty, 2018/1)], \mathrm{s}[(0 \nrightarrow \infty, +1) \text{ at } [2018/1, \infty)]\}$. It instantiates to 0 up to reference time $2018/1$ and afterwards to integers that increases by one for each reference time increase (1 at reference time $2018/2$, 2 at reference time $2018/3$, and so forth).

In this project, the student integrates ongoing integers as well as the duration function for ongoing time intervals and the aggregate functions addition, minimum and maximum for ongoing integers into the widely-used, open-source database system PostgreSQL. Then, the duration of ongoing time intervals can be stored in relations and thus, used in queries whose result is used as materialized view. Supporting functions on ongoing integers allows using the result of the duration function further in other queries.

Tasks

1. Literature study [1, 3, 2] in order to understand the concept of ongoing time points, the concept of ongoing integers, and the motivation, challenges, and the approach for getting results on ongoing values that remain valid as time passes by.
2. Integrate ongoing integers in PostgreSQL:
   (a) Create a new, variable-length data type *ongoingInt*.
   (b) Use a list of ongoing integer segments [2] that are non-overlapping and sorted in ascending order as representation.
   (c) Determine and implement the functions and predicates that PostgreSQL requires for a data type (e.g., hash and sorting functions).
   (d) Determine the advantages and disadvantages of your representation of ongoing integers.
   (e) Optional: develop a more space-efficient/runtime-efficient representation for ongoing integers.
3. Implement the following functions whose result are ongoing integers in PostgreSQL. Make the functions callable in SQL statements.
   - Duration function on ongoing time intervals.
   - Aggregate functions addition, minimum, and maximum whose input are two ongoing integers.
4. Evaluate the performance of your approach. Consider space consumption and runtime in your evaluation. Compare your implementation with the state-of-the-art approaches, which instantiate ongoing time points before evaluating the predicates [1, 4]. Determine the cases in which your approach performs well and the cases in which it does not and explain your findings.
5. Write the thesis (approximately 50 pages).
6. Present your thesis in a DBTG meeting (25 minutes presentation).

References

[1] J. Clifford, C. Dyreson, T. Isakowitz, C. S. Jensen, and R. T. Snodgrass. On the Semantics of Now in Databases. *ACM Transactions on Database Systems*, 1997.

[2] Y. Mülle and M. H. Böhlen. Durations in Ongoing Databases. to be published.

[3] Y. Mülle and M. H. Böhlen. Query Results over Ongoing Databases that Remain Valid as Time Passes By. to be published.

[4] K. Torp, C. S. Jensen, and R. T. Snodgrass. Modification Semantics in Now-Relative Databases. *Information Systems*, 29(8):653–683, Dec. 2004.

Supervisor: Yvonne Mülle (muelle@ifi.uzh.ch)

Start date: 15 March 2018

End date: 14 September 2018

Presentation date: TBD

University of Zurich
Department of Informatics

Prof. Dr. Michael Böhlen
Professor