

Master Project: In-Memory Data Management and Enterprise Application Development

Traditional enterprise systems use two separate database systems, one for Online Transaction Processing (OLTP) and another for Online Analytical Processing (OLAP). This separation was introduced because of performance reasons. With the current hardware (80 CPU cores / 2TB main memory per machine; possibly multiple servers) and software developments, this separation is not necessary anymore [P09].

All data is stored in an in-memory database, resulting in predictable query execution times [S11a, S11b]. This change from unpredictable disk-based database systems to a predictable in-memory database system opens a wide range of opportunity for optimizations. These optimizations also affect how enterprise applications will be developed in the future.

Description

The goal of this project is to analyze, design and implement new strategies for in-memory data management and future enterprise application development. The team will work on one or more of the following research topics.

In-Memory Data Management Optimizations

Given a database table T with N columns and the workload that affects T , we already know how to optimize the physical representation of T into a row-store, a column-store or a hybrid store. In a mixed OLTP-OLAP setting, a hybrid physical storage is often favorable [G12]. Furthermore, we know the influence of logical database design on query processing performance [B12].

Research question 1: How should workload-aware logical database schemas be designed for an in-memory database system if the conceptual data model and the respective workload are known?

The runtime of most database operations correlates with the number of tuples in the database. We experience that only a fraction of database tuples is touched by day-to-day queries.

Research question 2: How to separate active and passive data automatically in order to improve query performance on active data?

In-memory data management facilitates on-the-fly calculations of aggregates, which had to be materialized in disk-based systems for performance reasons. We already have a cost model for aggregation operations in the database.

Research question 3: How to decide if an aggregate should be materialized or calculated on the fly?

Where to execute which part of an enterprise application?

Business logic can be developed as stored procedures in the database, server-side code in the application server or as client-side code in the representation layer. Previously, the application logic was mainly executed in application servers to reduce the workload on the central database and to realize a scalable architecture.

Research Question 1: Where should which part of an enterprise application be executed and how can this execution be simplified to support the application developer?

Research Question 2: What is the future role of the traditional application server?

Future development of enterprise applications

In a situation where application code can be executed on client-side, server-side or within the database, software development of enterprise applications has to change. One way to do so is to extend the use of domain specific languages (DSL) for enterprise applications.

Research Question 1: Can enterprise systems be described in controlled natural languages?

Research Question 2: (How) can enterprise application developers benefit from such a DSL?

Research Question 3: (How) can the DSL code be used to change the locality of execution within the enterprise system based on the systems current state and workload to optimize the overall performance?

References

- [P09] Hasso Plattner: A Common Database Approach for OLTP and OLAP Using an In-Memory Column Database, Proceedings of the 35th SIGMOD International Conference on Management of Data, 2009
- [S11a] Schaffner, J. and Eckart, B. and Schwarz C. and Brunnert, J. and Jacobs, D. and Zeier, A. and Hasso Plattner: Simulating Multi-Tenant OLAP Database Clusters, 14. GI-Fachtagung für Datenbanksysteme in Business, Technologie und Web, 2011.
- [S11b] Schwarz, C. and Borovski, V. and Zeier, A.: Optimizing operation scheduling for in-memory databases, Proceedings of the 2011 International Conference on Modeling, Simulation and Visualization Methods, 2011.
- [P11] Hasso Plattner: SanssouciDB: An In-Memory Database for Processing Enterprise Workloads, BTW, 2011
- [PZ12] Hasso Plattner and Zeier, A.: In-Memory Data Management – Technology and Applications, Springer, 2012
- [B12]: Bog, A.: Benchmarking Composite Transaction and Analytical Processing Systems and Evaluation of Logical Database Schemas, Dissertation, 2012 (to appear)
- [G12]: Grund, M.: HYRISE – an In Memory Database Engine, Dissertation, 2012 (to appear)

Contact

Enterprise Platform and Integration Concepts Group, HPI, Potsdam

- Prof. Hasso Plattner, office-epic@hpi.uni-potsdam.de
- Jürgen Müller, juergen.mueller@hpi.uni-potsdam.de, +49 (331) 5509 – 1319, SNB-E.5