

Accelerating Key In-memory Database Functionality with FPGA Technology

Jordan Leggett, Andrew Hartwell, Akihiro Mizusawa, Robert Mushkot, Vadim Pelyushenko

Abstract

In-memory databases supporting online transactional and analytical workloads rely on *differential updates* to manage writes while maintaining the majority of data in an optimized read-only structure. This necessitates the merge of a write optimized delta partition with the read-only main partition. The purpose of this project is to develop an FPGA using OpenCL that can reduce this bottleneck on the database.

Approach

The diagram to the right represents the structure of the HANA database, with the delta merge operation split into 4 OpenCL kernels each running simultaneously :

Delta Store: Keeps track of database writes and uses dictionary compression (D_D) to store data in the delta column (C_D). This structure is optimized for transactional processing.

Main Store: A read-only partition that uses a sorted dictionary (M_D) for data compression in the main data column (C_M). This structure is optimized for analytical processing.

Merge: The FPGA recodes the main store column Value IDs to map to new values based on a provided dictionary while utilizing pipeline parallelism.

Acknowledgments

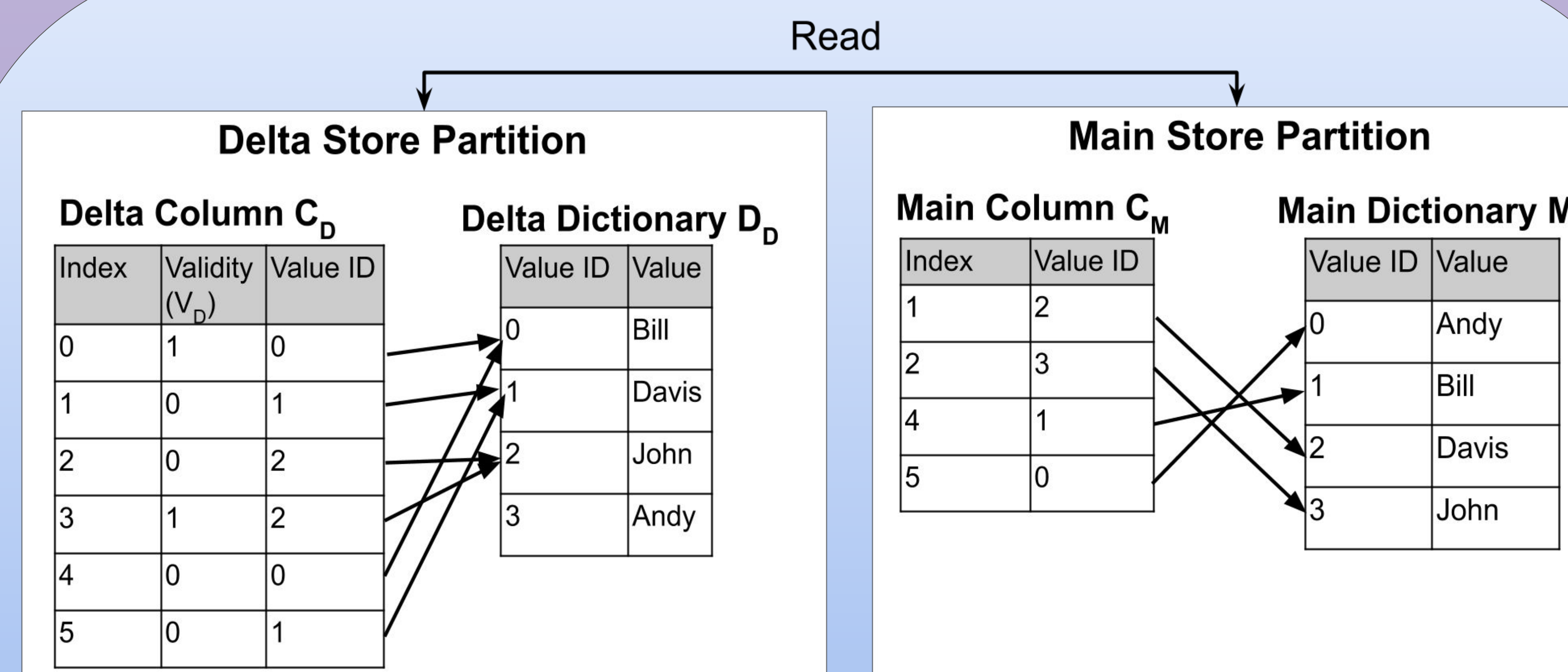
We would like to thank the following for all of their help and guidance throughout this project: John McGlone (SAP), Jody Glider (SAP), Suleyman Demirsoy (Intel), Christian Faerber (Intel), Ryan Tekerlek (Intel), Dan Nguyen (Intel), Akila de Silva (UCSC), and Professor Richard Jullig (UCSC).

Overview

SAP uses their High-Performance Analytic Application (HANA) In-Memory Database to quickly provide data to over 400,000 customers using their numerous cloud services. HANA requires the use of a *delta merge* operation to merge recently written data from a delta storage partition into a main storage partition. The delta merge operation requires extra memory usage and longer data access times, which is expensive for both SAP and any clients using HANA. With OpenCL, we can program an FPGA to build custom hardware tailored around data and pipeline parallelism and optimal performance of the delta merge operation. SAP's previous FPGA research without OpenCL achieved a bandwidth of 5-10 GB/s, however that prototype is outdated and the technology used is no longer supported.

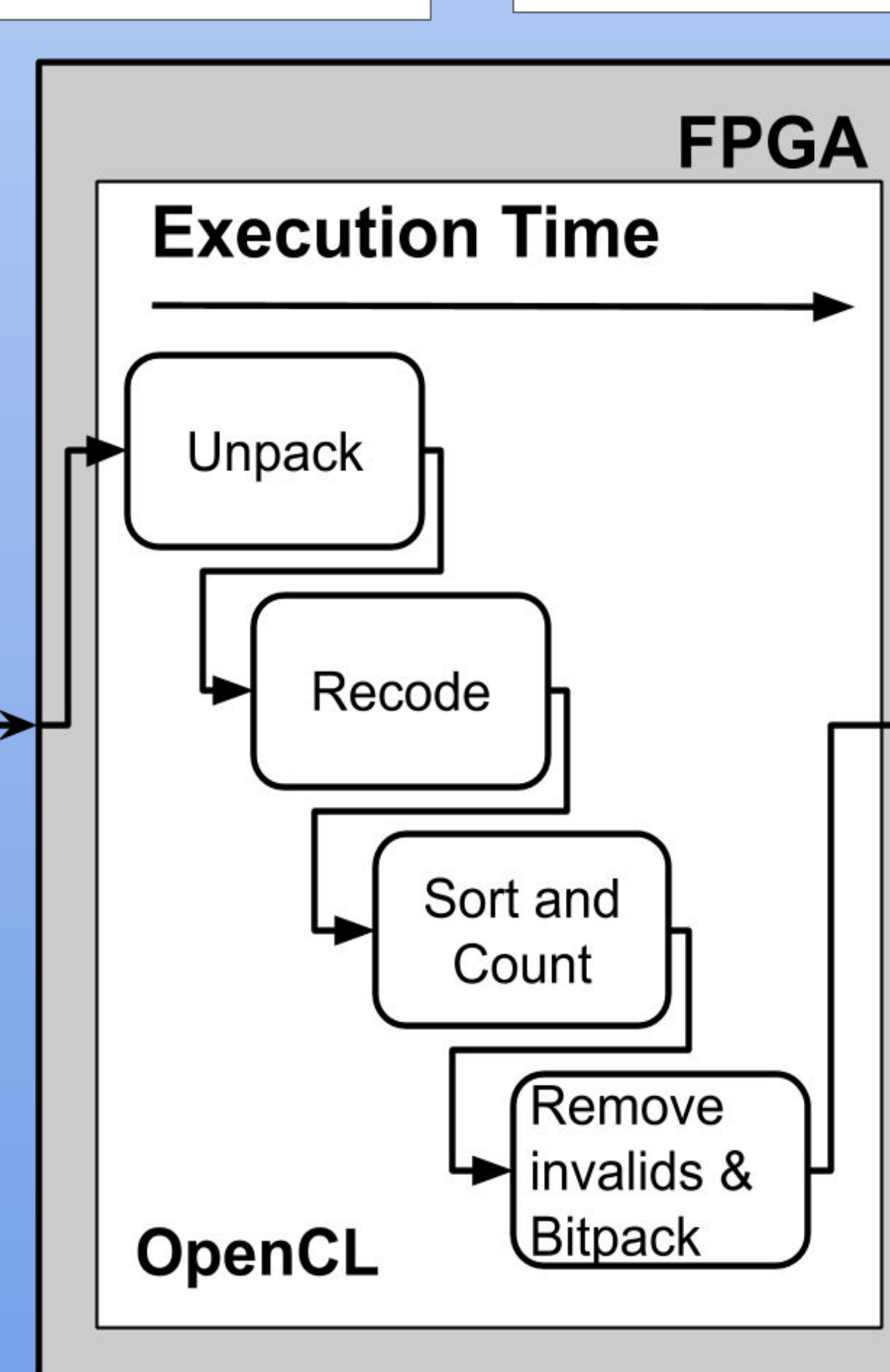
Technologies

- **FPGA:** A reprogrammable hardware device that allows flexible, reconfigurable computing and rapid prototyping of hardware designs. FPGAs are capable of data/pipeline/task parallelism.
- **OpenCL:** A programming language designed for parallel computing on hardware accelerators. OpenCL provides a framework for software developers who want to work with hardware.



- **Unpack:** Decompresses bitpacked delta column elements (C_D).

- **Sort and Count:** Moves all entries flagged for deletion by the validity vector (V_D) to one end.



- **Recode:** Assigns new value IDs to values based on externally provided mapping.

- **BitPack / Remove Invalids:** Removes the deleted entries, compresses the column elements, and returns the final column (C_M).

Results

- Delta merge operation in C++.
- Initial OpenCL FPGA design with bandwidth of 48 MB/s.
- Used loop optimization techniques and utilized channels for data pipelining and fast kernel communication.
- Several FPGA designs with OpenCL and achieved bandwidth of up to approximately 2.9 GB/s.

Conclusion

The delta merge is a core function of SAP's HANA database, therefore efficiency is critical. We have identified several sections with optimization potential which could increase our throughput in hopes of surpassing the 5-10 GB/s achieved by SAP's previous research. Our project will serve as a prototype for a future version implemented into SAP's HANA.