Enterprise Platform and Integration Concepts: Research Group of Prof. Dr. Hasso Plattner

Master Project Summer Term 2016

Dynamic Data Structures for Non-Volatile RAM

Motivation

Non-volatile memory is an upcoming technology that fills the gap between fast, but expensive DRAM, and slower, but bigger SSD storage. It will come in two forms: a version that is attached to the PCIe bus where it is accessed similar to an SSD, and a version (NVRAM) that is directly attached to the CPU's memory controller where it behaves like DRAM. For in-memory databases, this second type is of bigger interest as it could be used to build bigger systems at lower hardware costs.

For the foreseeable future, NVRAM will have a higher latency and lower bandwidth than DRAM, both for reads but even more so for writes. When we imagine a system in which both types of memory are used side-by-side, this means that we want frequently used data structures on fast DRAM and rarely used data on NVRAM. The access frequency is, however, not the only criteria that we want to optimize by. Additionally, access patterns (random vs. sequential), read/write ratio, and others come into play.

In this project, our goal is to write a library that can transparently manage data structures on a hybrid DRAM/NVRAM system and optimize the placement of the data structures so that the limited DRAM space is used efficiently.

In a previous project, we looked at exploiting the non-volatility of NVRAM to allow for faster database restarts. This project will focus on the higher capacity of NVRAM and will leave the question of persistence aside.

Project Goals

1. Instrument existing data structures (vectors, trees, hash maps, etc.) to gather access statistics.
2. Find efficient placement strategies for these data structures when DRAM space is limited.
3. Move data structures between memory types as needed.
4. Modify our research database HYRISE to make use of this library and benchmark the benefits.

Technology & Skills

Our goal is to develop a library that serves as a drop-in replacement for existing data structures in C++. As such, basic knowledge of C++ is required. Additional knowledge of generic data structures and templates is welcomed. Prior understanding of the fundamentals of in-memory databases is helpful, but not required.

Contact

You are welcome to visit us in the “Villa” or reach out to one of the contacts listed below. For further information, we also invite you to an upfront meeting at room V2.16 on Feb 11, 2016 at 3.15 PM.

Dr. Matthias Uflacker (matthias.uflacker@hpi.uni-potsdam.de)
Markus Dreseler (markus.dreseler@hpi.uni-potsdam.de)
Martin Boissier (martin.boissier@hpi.uni-potsdam.de)
Stefan Klauck (stefan.klauck@hpi.uni-potsdam.de)

Getting started

If you cannot wait to hear more about non-volatile data structures or access monitoring, we recommend the following readings:

- “Let’s Talk About Storage & Recovery Methods for Non-Volatile Memory Database Systems“, Arulraj et al., SIGMOD 2015 [http://www.cs.cmu.edu/~pavlo/papers/p707-arulraj.pdf]
- “Compacting Transactional Data in Hybrid OLTP&OLAP Databases”, Funke et al., VLDB 2012 [http://vldb.org/pvldb/vol5/p1424_florianfunke_vldb2012.pdf]