DXGraph: large-scale graph processing based on a distributed in-memory key value store

Stefan Nothaas, Kevin Beineke, Michael Schöttner
Heinrich-Heine University Duesseldorf

Online graph analytics and large-scale interactive applications such as social media networks require low-latency data access to billions of small data objects. In order to provide short response times for interactive queries all data need to be stored in memory. Unfortunately, access patterns in these application domains are mostly irregular which makes caching inefficient and requires very large caches like Facebook's memcached infrastructure storing around 75% of all data in caches.

DXRAM is a distributed key-value store addressing these challenges by keeping all data always in memory. The system is designed to run in a single data center and is optimized for efficiently managing small data objects. A super-peer overlay network provides scalability where super peers are responsible for data lookups, peer recovery coordination, monitoring and load balancing. Peers provide and manage their memory as storage and their SSDs for remote backups.

The memory management is optimized for efficiently supporting small data objects and highly concurrent access. Furthermore, it is carefully designed to also support efficient distributed meta-data management on super peers.

A novel remote logging scheme is optimized for high throughput and small data objects for SSDs. By scattering the state of one peer to many backup nodes, DXRAM provides a fast parallel recovery. This is complemented by a concurrent log reorganization which keeps logs compact and reduces the overhead for detecting outdated entries during recovery time.

DXGraph is a graph framework built on top of DXRAM providing graph processing on DXRAM storage nodes. This includes support for loading, generation and processing of very-large graphs with either lightweight jobs or master-slave coordinated tasks. For a natural graph representation, each vertex is stored as an object. For a preliminary evaluation, we have implemented the breadth-first search (BFS) algorithm, as specified by the Graph500 benchmark. This stress test is designed for comparing shared-memory machines as well as cluster systems for graph processing performance. Our evaluations show that DXGraph's BFS implementation is up to five times faster than Grappa's and GraphLab's as well as providing the lowest overhead. Furthermore, DXGraph's implementation is scaling very well with a peak throughput of 323 million traversed edges per second on 104 physical nodes.

Currently, we are adopting DXGraph to an Infiniband environment in order to further reduce data access latencies.