Mathematisch-Naturwissenschaftliche Fakultät
Der Dekan

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Neue Promotionsordnung

An alle
hauptamtlichen Professoren/innen
und Privatdozenten/innen
des Faches Informatik der
Mathematisch-Naturwissenschaftlichen Fakultät

Promotionsangelegenheiten
Universitätsstr. 1
40225 Düsseldorf
Telefon: (0211) 81-15092
Telefax: (0211) 81-15090
E-Mail: promotion@mnf.uni-duesseldorf.de

16.11.2018

Promotionsverfahren von **Herrn M.Sc. Kevin Beineke**
Auslage der Dissertation und Gutachten sowie Termin der mündlichen Prüfung
Anlage: Einseitige Zusammenfassung der Dissertation

Sehr geehrte Damen und Herren,

in dem oben genannten Promotionsverfahren wird die Annahme der Dissertation

Schnelle parallele Fehlererholung in verteilten In-Memory Key-Value Systemen

von den Berichterstattern Prof. Dr. M. Schöttner, Prof. Dr. M. Mauve und Prof. Dr. L. Bougé beantragt. Sie kann zusammen mit den Gutachten in der Zeit

vom 01.12.2018 bis 12.12.2018

im Promotionsbüro (Gebäude 25.32, Ebene 00, Raum 36) zu den Sprechzeiten eingesehen werden.

Einsprüche gegen diese Dissertation können nur zwei Tage nach der vorgenannten Frist geltend gemacht werden. Erfolgt kein Einspruch, so gilt die Dissertation als angenommen (§ 7 Ziffer (5) PO).

Sofern die Dissertation angenommen wird, findet die mündliche Prüfung am

17.12.2018 um 10:30 Uhr

im **Raum 25.12.01.51** statt. Als Prüfer sind vorgesehen:
Juniorprof. Dr. K. Graffi, Prof. Dr. M. Lercher und PD Dr. W. Linder.

Zuhörer sind bei der Befragung zugelassen.

Mit freundlichen Grüßen
im Auftrag

A. Stefanidou
Athina Stefanidou

Schnelle parallele Fehlererholung in verteilten In-Memory Key-Value Systemen

Abstract of the inaugural-dissertation by Kevin Beineke, Juli 2018

Big data analytics and large-scale interactive graph applications require low-latency data access and high throughput for billions to trillions of mostly small data objects. Distributed in-memory systems address these challenges by storing all data objects in RAM and aggregating hundreds to thousands of servers, each providing 128 GB to 1024 GB RAM, in commodity clusters or in the cloud. This thesis addresses two main research challenges of large-scale distributed in-memory systems: (1) fast recovery of failed servers and (2) highly concurrent sending/receiving of network messages (small and large messages) with high throughput and low latency.

Masking server failures requires data replication. We decided to replicate data on remote disks and not in remote memory because RAM is too expensive and volatile, resulting in data losses in case of a data center power outage. For interactive applications, it is essential that server recovery is very fast, i.e., the objects' availability is restored within one or two seconds. This is challenging for servers storing hundreds of millions or even billions of small data objects. Additionally, the recovery performance depends on many factors like disk, memory and network bandwidth as well as processing power for reloading the storage. This thesis proposes a novel backup and recovery concept based on replicating the data of one server to many backup servers which store replicas in logs on their local disks. This allows a fast-parallel recovery of a crashed server by aggregating resources of backup servers, each recovering a fraction of the failed server's objects. The global replica distribution is optimized to enable a fast-parallel recovery of a crashed server as well as providing additional options for tuning data loss probability in case of multiple simultaneous server failures. We also propose a new two-level logging approach and efficient epoch-based version management both designed for storing replicas of large amounts of small data objects with a low memory footprint. Server failure detection, as well as recovery coordination, is based on a super-peer overlay network complemented by a fast, parallel local recovery utilizing multiple cores and mitigating I/O limitations. All proposed concepts have been implemented and integrated into the Java-based in-memory system DXRAM. The evaluation shows that the proposed concept outperforms state-of-the-art distributed in-memory key-value stores. Large-scale experiments in the Microsoft Azure cloud show that servers storing hundreds of millions of small objects can be recovered in less than 2 seconds, even under heavy load.

The proposed crash-recovery architecture and the key-value store itself require a fast and highly concurrent network subsystem enabling many threads per server to synchronously and asynchronously send/receive small data objects, concurrently serialized into messages and aggregated transparently into large network packets. To the best of our knowledge, none of the available network systems in the Java world provide all these features. This thesis proposes a network subsystem providing concurrent object serialization, synchronous and asynchronous messaging and automatic connection management. The modular design is able to support different transport implementations, currently implemented for Ethernet and InfiniBand. We combine several well-known and novel techniques, like lock-free programming, zero-copy sending/receiving, parallel de-/serialization and implicit thread scheduling, to allow low-latency message passing while also providing high throughput. The evaluation of the developed network subsystem shows good scalability with constant latencies and full saturation of the underlying interconnect, even in a worst-case scenario with an all-to-all communication pattern, tested with up to 64 servers in the cloud. The network subsystem achieves latencies of sub 10 µs (round-trip) including object de-/serialization and duplex throughputs of more than 10 GB/s with FDR InfiniBand and good performance with up to hundreds of threads sending/receiving in parallel, even with small messages ($<$ 100 bytes).