Technische Universität Braunschweig

Institute of Operating Systems and Computer Networks

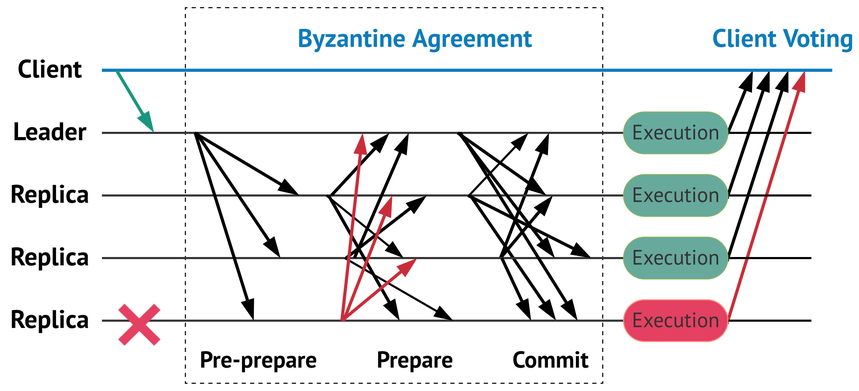
Low-Latency Network-Scalable Byzantine Fault-Tolerant Replication

Ines Messadi, TU Braunschweig, Germany. Advisor: Rüdiger Kapitza
messadi@ibr.cs.tu-bs.de, rrkapitz@ibr.cs.tu-bs.de

Byzantine Fault Tolerance (BFT)

- **Traditional BFT protocols**
  - Tolerating arbitrary (Byzantine) faults
  - $3f + 1$ nodes to tolerate f faults
  - **TCP/IP-based communication**

↪ **High reliability** and consistency



Problem Statement

- **Drawbacks of BFT protocols**
  - Multiple rounds of communication ⇒ **high latency**
  - Costly message complexity
  - Limited throughput & scalability
  - TCP incurs **high latency**

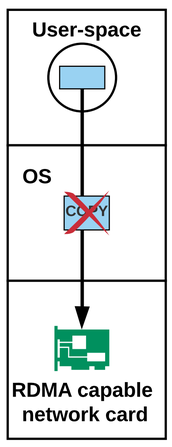
↪ Availability of modern hardware technology such as the **low-latency Remote Direct Memory Access (RDMA) networking**

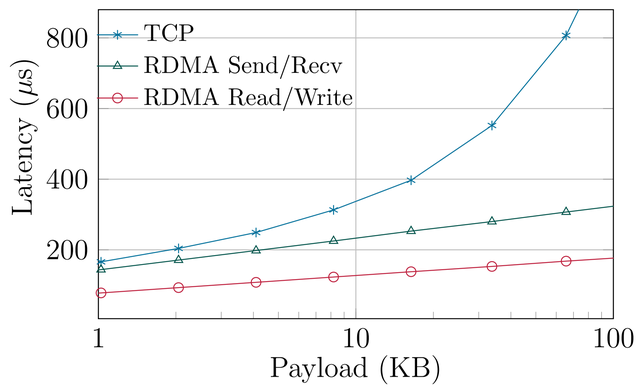
- **Challenge**
  - RDMA interfaces require an explicit design of applications

⇒ Necessity of the redesign of **existing BFT protocols for RDMA**

Remote Direct Memory Access

- Direct access to memory on remote systems with **without CPU involvement**
- **Zero-copy** data transfer
- Communication resources
  - Queue based-communication
  - Memory registration & buffer management
- Richer data transfer primitives
  - One-sided Read/Write
  - Two-sided Send/Receive





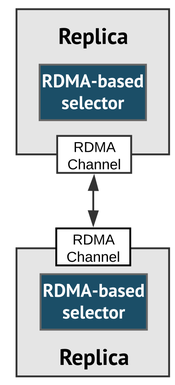
Towards building RDMA-based BFT

How can we build a secure scalable RDMA-based BFT ?

- Effcient use of RDMA one-sided and two-sided communication
  - Clients use Send primitive to not saturate the leader
  - Dynamically Connected Transport (DCT) for better scalability
  - Replicas perform a direct RDMA Write into remote memory
- **Problem:**
  - Malicious replicas have access to remote nodes' memory
  - Memory RDMA keys are not secure

  → Implement counter-measures
- **Basis BFT protocol**: Hybster [Behl et al., EuroSys'17]
  - Use of two-sided to avoid security issues

  ↪ **Preliminary approach**
  Design an RDMA-based selector replacing the **Java NIO selector**



Related Work

- **DARE** is an RDMA-based **Raft protocol** optimized using RDMA features[a]
- **APUS** is scalable RDMA-based **Paxos protocol** performing replication using RDMA Write[b]

[a] Marius Poke and Torsten Hoefler. DARE: high-performance state machine replication on RDMA networks. In ACM Symposium on High-Performance Parallel and Distributed Computing (HPDC), 2015

[b] Cheng Wang et al. APUS: Fast and scalable Paxos on RDMA. In: Proceedings of the 2017 Symposium on Cloud Computing . ACM. 2017, pp. 94107

Example Applications: Blockchain and Coordination Services

- Building a coordination service with a similar interface as ZooKeeper
  - Strong consistency and availability
- Implementing BFT-ordering service

↪ Benefit from improved performance of BFT