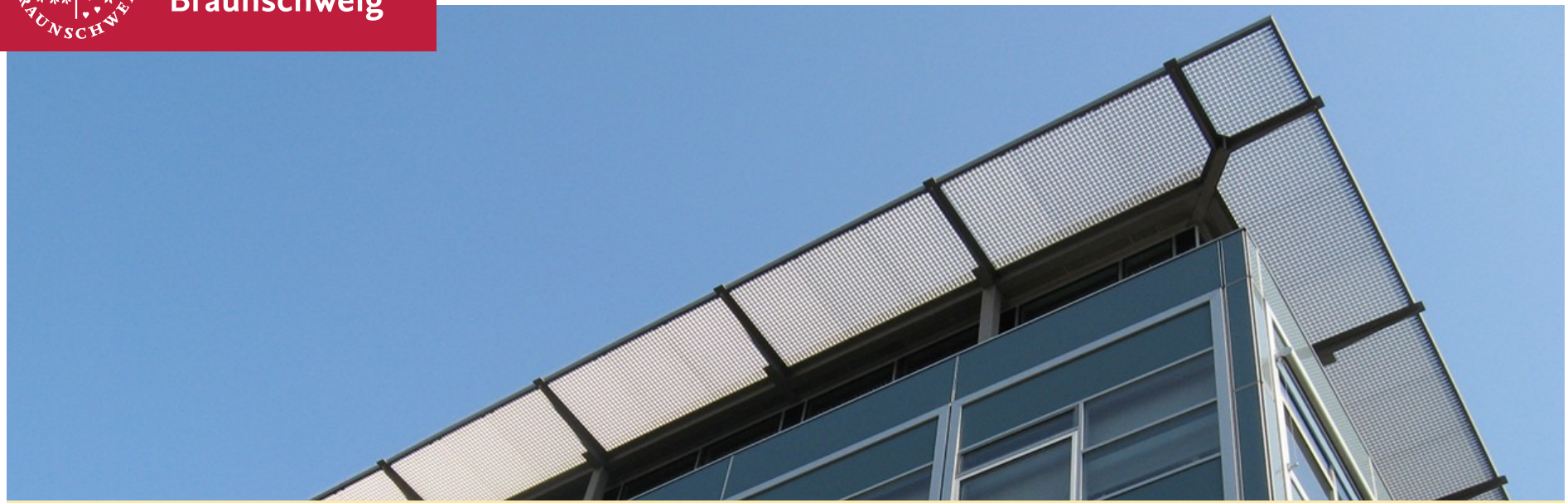
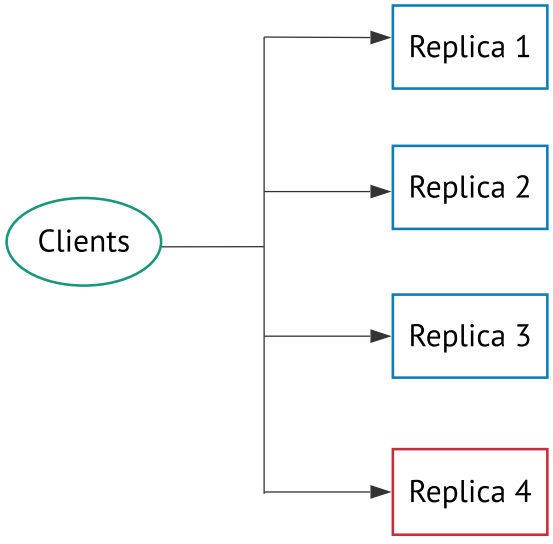


Low-Latency Network-Scalable Byzantine Fault-Tolerant Replication

12th EuroSys Doctoral Workshop (EuroDW 2018)

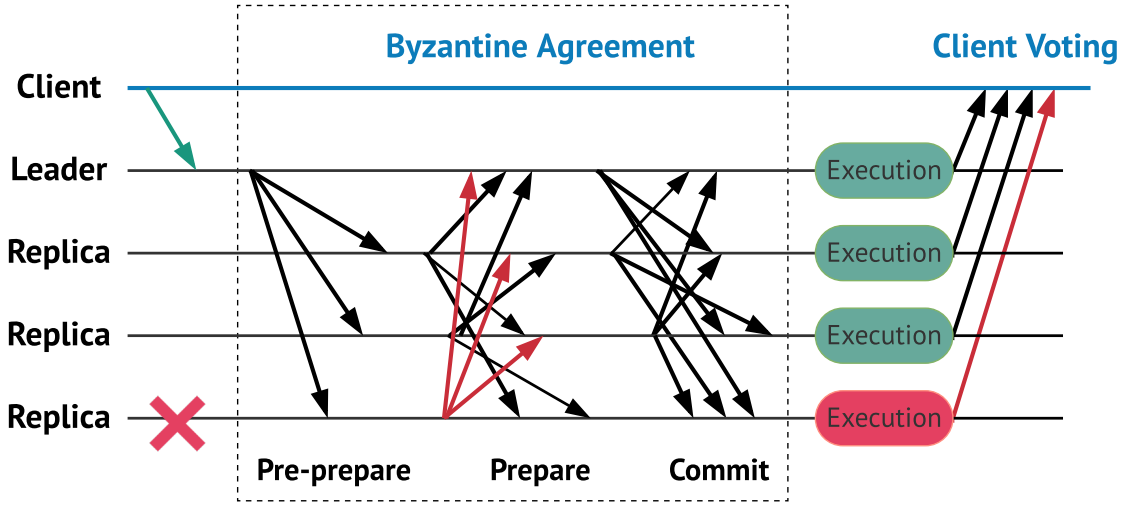
Ines Messadi, TU Braunschweig, Germany, 2018-04-23
New PhD student (Second month) in the distributed systems group
Research area: Resiliency of distributed systems, Byzantine Fault Tolerance
Advisor: Rüdiger Kapitza

Overview

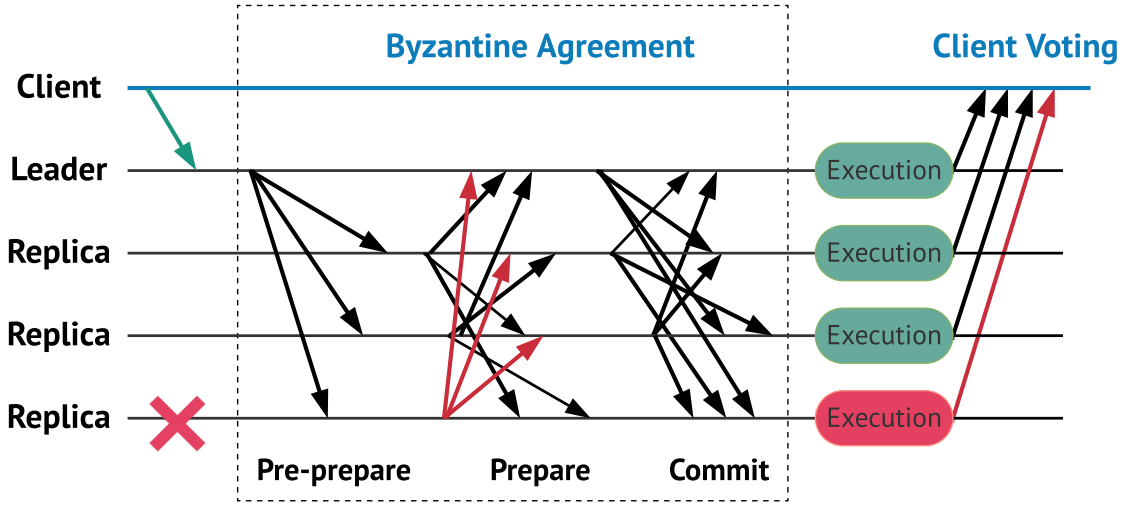


$3f + 1$ nodes to tolerate f faults

Overview

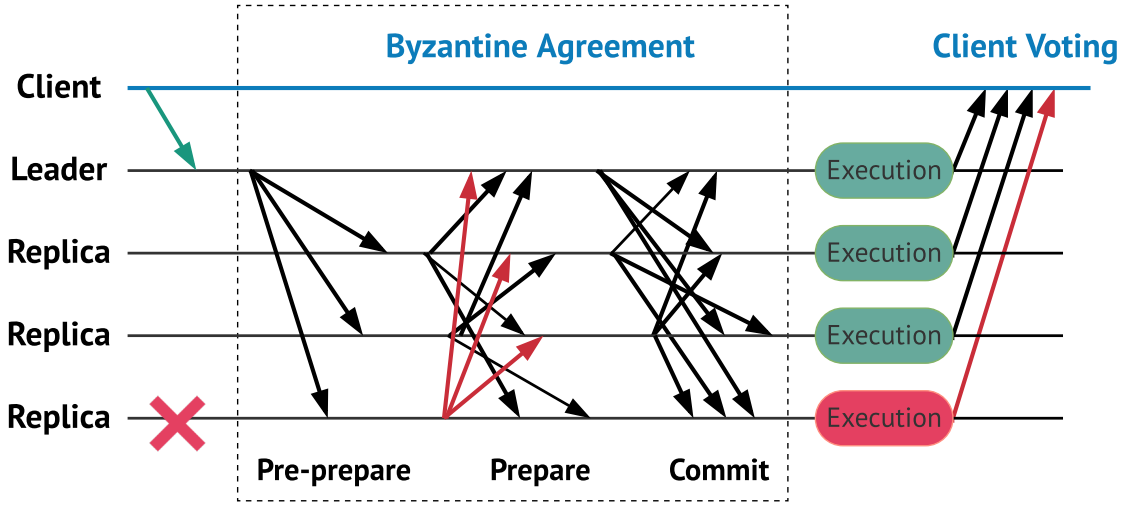

Byzantine Agreement
Client Voting
Client
Leader
Replica
Replica
Replica
Execution
Execution
Execution
Execution
Pre-prepare
Prepare
Commit

Overview



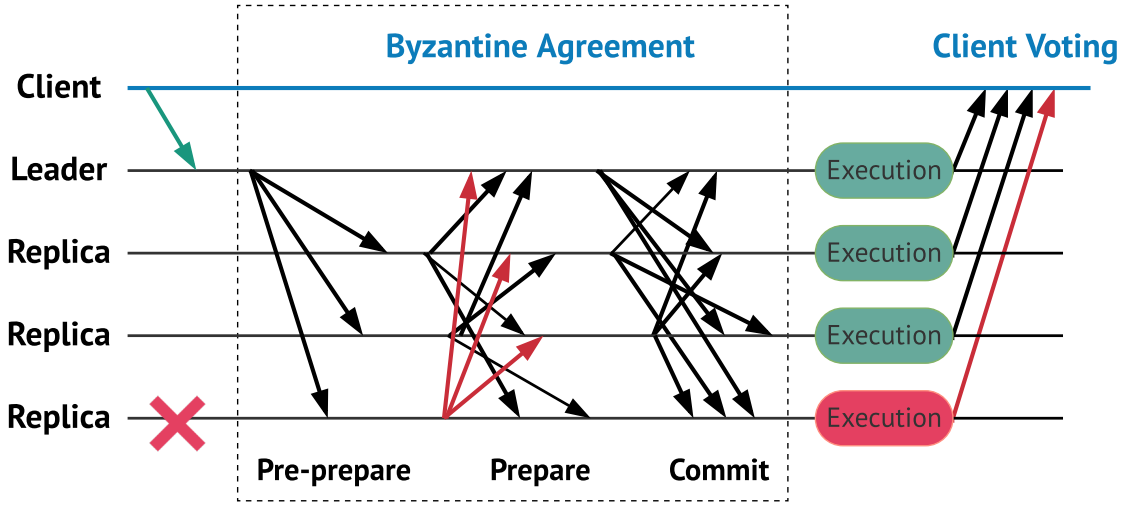
- **Problem:** Agreement latency overhead & message complexity in BFT

Overview



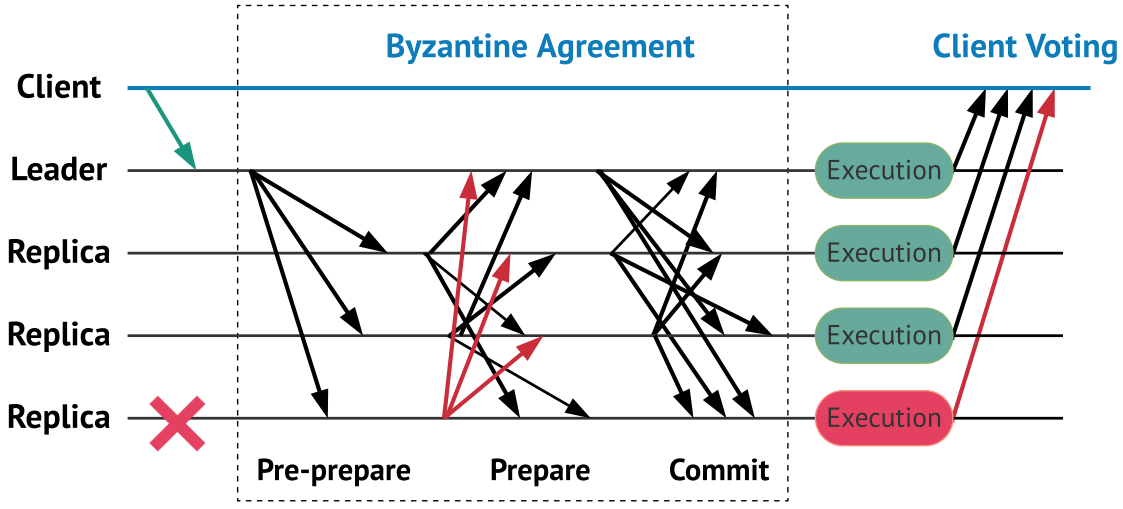
- **Problem:** Agreement latency overhead & message complexity in BFT
- **Reason:** Multiple communication rounds & slow TCP networking

Overview



- **Problem:** Agreement latency overhead & message complexity in BFT
- **Reason:** Multiple communication rounds & slow TCP networking
- **New trend:** Availability of modern hardware technology such as Remote Direct Memory Access (RDMA)

Overview



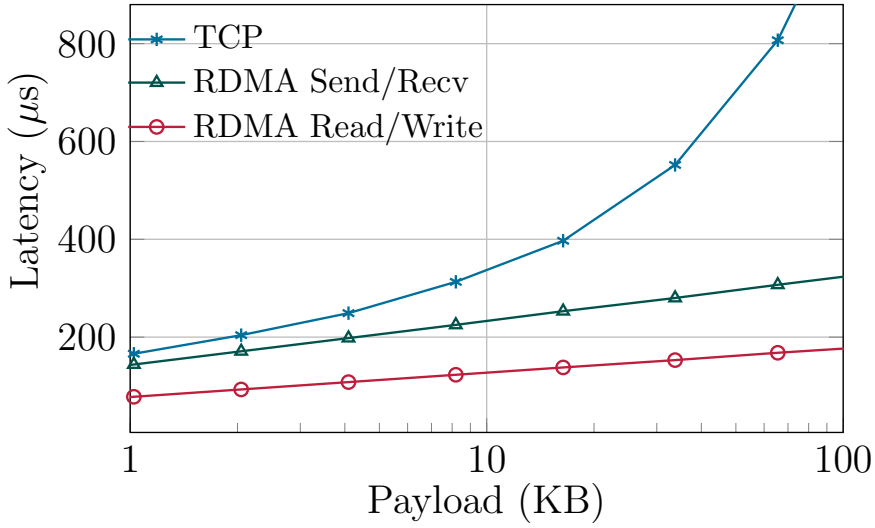
- **Problem:** Agreement latency overhead & message complexity in BFT
- **Reason:** Multiple communication rounds & slow TCP networking
- **New trend:** Availability of modern hardware technology such as Remote Direct Memory Access (RDMA)
- **Consequence:** A need to redesign current BFT systems

↪ **How can we build a secure fast and scalable RDMA-based BFT?**

Remote Direct Memory Access (RDMA)

- Why RDMA ?
  - Zero-copy data transfer
  - Reduce communication CPU usage

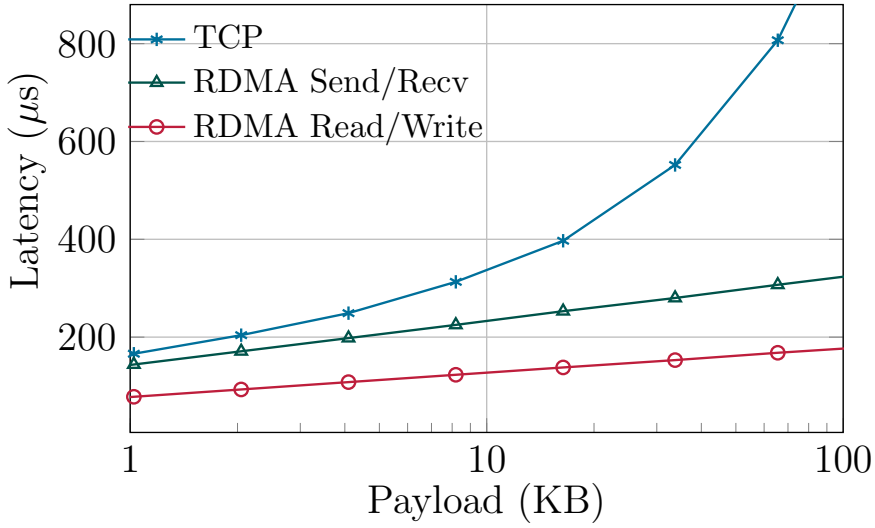
↪ Low latency and CPU efficiency



Remote Direct Memory Access (RDMA)

- Why RDMA ?
  - Zero-copy data transfer
  - Reduce communication CPU usage

↪ Low latency and CPU efficiency



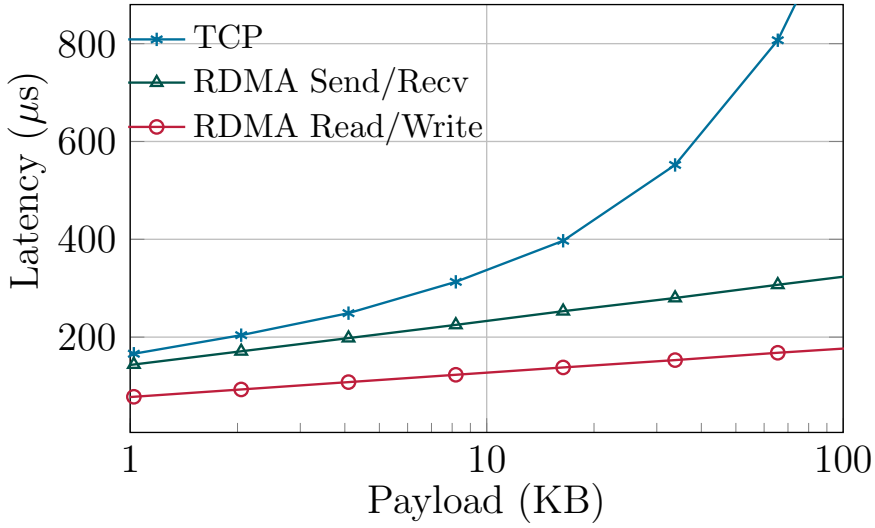
- Challenges
  - Different communication mechanisms
  - Inappropriate design ⇒ **unexpected bad performance**
  - Security issues

↪ Require an explicit design of applications

Remote Direct Memory Access (RDMA)

- Why RDMA ?
  - Zero-copy data transfer
  - Reduce communication CPU usage

↪ Low latency and CPU efficiency



- Challenges
  - Different communication mechanisms
  - Inappropriate design ⇒ **unexpected bad performance**
  - Security issues

↪ Require an explicit design of applications

Observation

Necessity to redesign the existing BFT protocols for RDMA

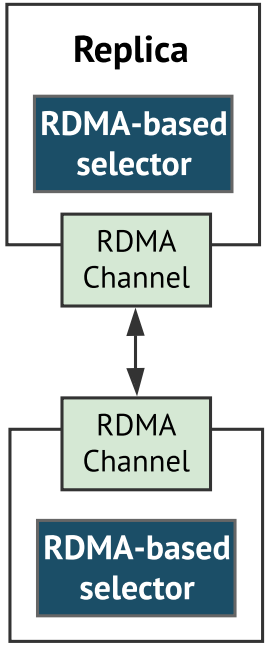
Towards building RDMA-based BFT

- Basis BFT protocol: **Hybster** [Behl et al., EuroSys'17]
  - Building an RDMA-tailored BFT protocol
  - Investigating RDMA communication tradeoffs
  - Counter-measures for the resilient use of RDMA

Towards building RDMA-based BFT

- Basis BFT protocol: **Hybster** [Behl et al., EuroSys'17]
  - Building an RDMA-tailored BFT protocol
  - Investigating RDMA communication tradeoffs
  - Counter-measures for the resilient use of RDMA
- Preliminary approach
  - Build similar interfaces to TCP programming using RDMA

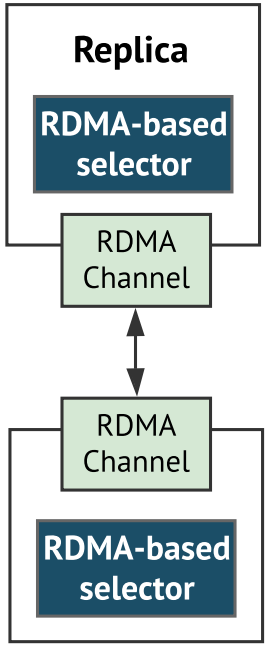
  ⇒ Aiming to take fully advantage of RDMA



Towards building RDMA-based BFT

- Basis BFT protocol: **Hybster** [Behl et al., EuroSys'17]
  - Building an RDMA-tailored BFT protocol
  - Investigating RDMA communication tradeoffs
  - Counter-measures for the resilient use of RDMA
- Preliminary approach
  - Build similar interfaces to TCP programming using RDMA

  $\Rightarrow$ Aiming to take fully advantage of RDMA



- Example applications: Blockchain & coordination services