EPFL

HovercRaft: Achieving Scalability and Fault-tolerance for microsecond-scale Datacenter Services

Marios Kogias Edouard Bugnion

Eurosys 2020

École
polytechnique
fédérale
de Lausanne

Datacenter Services

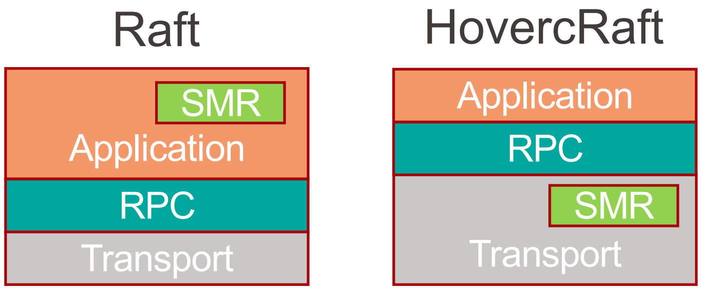
- microsecond-scale computing
- fast networking
  - 10/40/100 Gbps links
  - few µs RTTs
  - kernel bypass
  - in-network programmability
- in-memory services
- tight latency SLOs

- Failures are the common

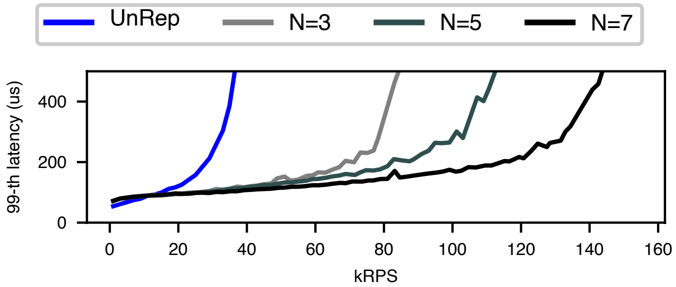
Need for microsecond-scale fault-tolerant systems

Contribution

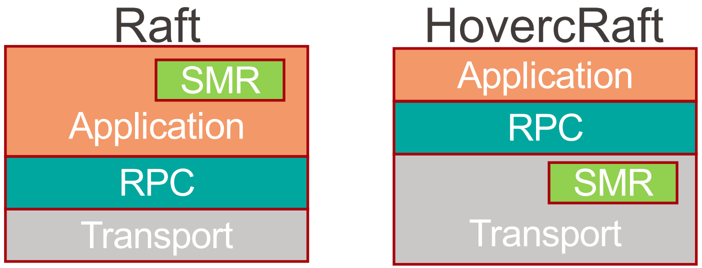
- How to implement **application-agnostic** fault-tolerance by integrating SMR in the transport protocol?



- How to achieve both **fault-tolerance** and **scalability** in SRM?



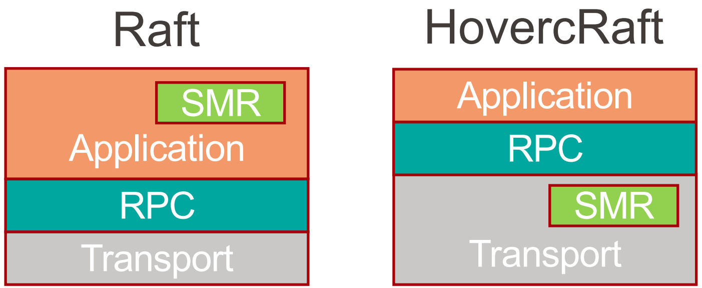
HovercRaft



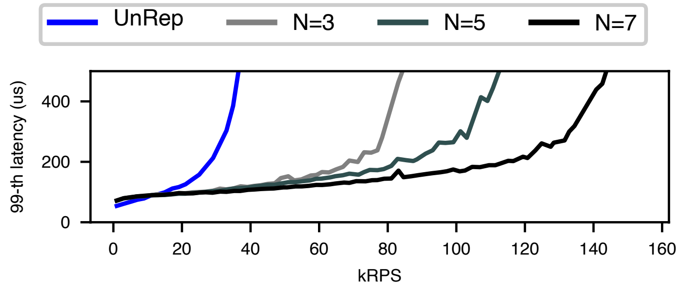
- SMR in the Transport layer
  - Fault-tolerance at the RPC boundaries
- Forward RPC only when committed
- HovercRaft on R2P2 (**R**equest-**R**espose-**P**air-**P**rotocol)
  - Transport protocol for datacenter RPCs
  - Request-Response abstraction at the end-points and the network
  - Designed for in-network RPC policy enforcement
- Fault-tolerance as an RPC policy
- Allows further optimisations
  - e.g IP multicasting, RPC load balancing etc

Contribution

- ~~How to implement **application-agnostic** fault-tolerance by integrating SMR in the transport protocol?~~



- How to achieve both **fault-tolerance** and **scalability** in SRM?



HovercRaft Design Summary

Marios Kogias

Technique	Benefit
▪ Separate request **data** and **metadata** • IP multicast for request replication	☞ Avoid leader IO Tx bottleneck due to replication
▪ Load balance client **replies**	☞ Avoid leader IO Tx bottleneck
▪ Load balance **read-only** execution	☞ Avoid leader CPU bottleneck
▪ Offload **fan-out/fan-in** management to programmable switches	☞ Decouple SMR cost from #followers

Evaluation

- DPDK-based server
- Microbenchmarks
  - Synthetic service time
  - Synthetic request-reply size
- Redis with YCSB-E workload
- Metrics
  - Latency vs throughput
  - Max throughput under latency SLO

- TLDR Results
  - 1M RPS under 500 μs 99-th Latency
  - Fixed SMR cost with different #followers
  - Scalability with #followers for:
    - IO-bottlenecked workloads (client replies)
    - CPU-bottlenecked read-only workloads

Conclusion

- HovercRaft
  - Fault-tolerance at the RPC boundaries
  - Embed SMR (Raft) in R2P2
- Use redundancy for **fault-tolerance** & **scalability**
  - Data and metadata separation and IP multicast
  - Careful reply and read-only load balancing
  - In-network SRM acceleration with P4 switches

https://github.com/epfl-dcsl/hovercraft

Thank you!