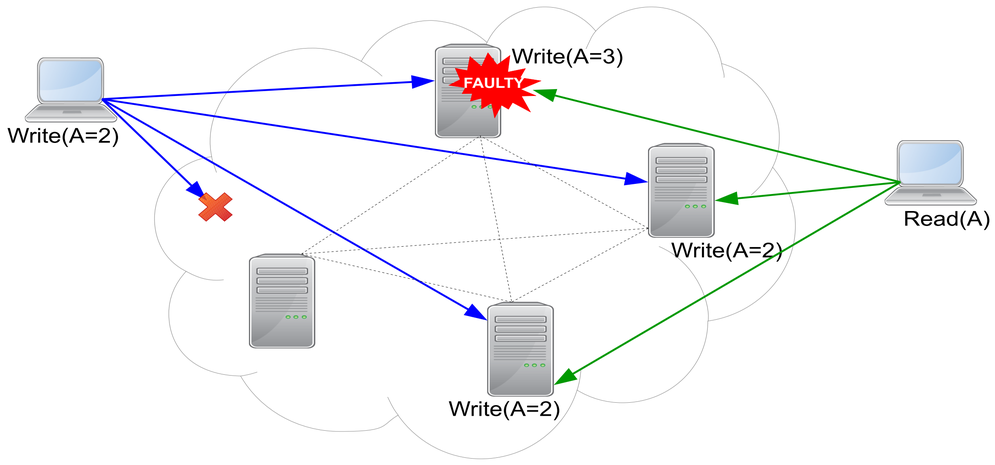
BFT-Bench: Framework to Evaluate Robustness and effectiveness of BFT Protocols in Practice

Divya Gupta, Lucas Perronne
University of Grenoble, LIG, France
{Divya.Gupta, Lucas.Perronne}@imag.fr

Sara Bouchenak
INSA Lyon, LIRIS, France
Sara.Bouchenak@insa-lyon.fr

Background



- Byzantine Fault Tolerance (BFT) protocols aim to tolerate arbitrary failures using replication techniques while maintaining consistency across replicas.
- BFT protocols aim to improve system's dependability & performance while ensuring its correctness.

Contributions

- BFT-Bench, the first framework for benchmarking and comparing BFT protocols in practice
- Dynamic injection of faultloads and workloads
- Integration of prototypes of BFT protocols
- Mechanisms for automatic deployment of experiments in cluster & cloud environments
- Performance monitoring & reporting
  - High-level statistics: Throughput, Latency
  - Low-level statistics: Network bandwidth usage, CPU utilization, Total number of re-transmissions of each request, etc.

BFT-Bench Design Principles

- *Byzantine faults in consideration*
  - Replica Crash
  - Message Delay
  - Network Flooding
  - System Overloading
- *Faultload for injecting faults*
  - Fault Trigger Time
  - Fault Type
  - Fault Parameters
    (fault location, delay time, message type, request message size, #clients)
- *BFT protocols in consideration*
  - *PBFT* - A practical BFT protocol
  - *Chain* - Performance enhancement in fault free conditions
  - *RBFT* - Performance enhancement in presence of faults

Motivation

- Absence of a testbed to evaluate implementations of BFT protocols under:
  - fault-free conditions, and
  - different byzantine failures
- No framework to identify and evaluate impacts of byzantine behaviors on BFT systems:
  - To incorporate *corrective measures* during the designing of BFT protocols
  - To test the *robustness & effectiveness* of prototypes of BFT protocols in faulty settings
  - For *easy adoption* of these protocols in real world sytems

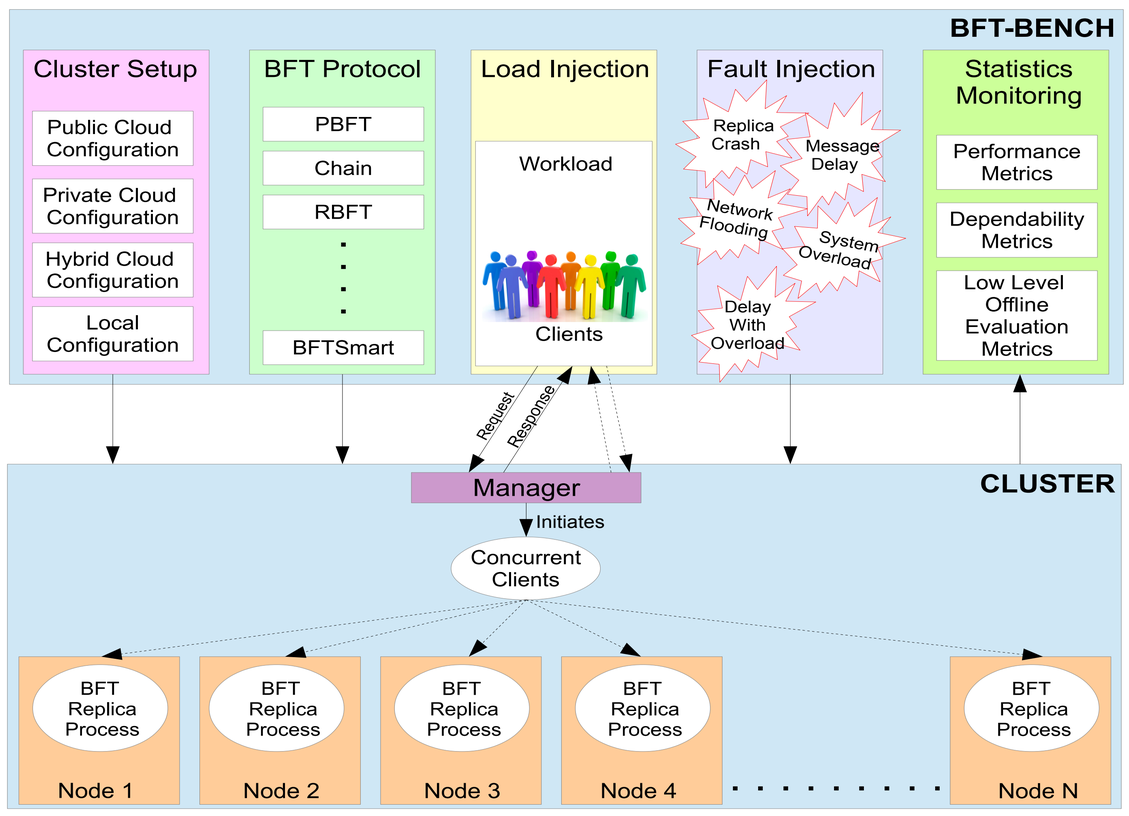
Related Work

PROGRESS OF BFT PROTOCOLS	
BFT from Theory to Practice	
First theoretical byzantine general problem	by Lamport
Practical BFT (PBFT)	First practical protocol
Performance enhancement in fault free scenarios	
Speculation based	Zyzzyva, Zeno, ZZ
Quorum based	HQ, Q/U, Scrooge, Quorum
Trusted Component based	BFT-TO, MinBFT, MinZyzzyva
Switching based	Chain, Aliph, CheapBFT
Trusted Client based	OBFT
Performance enhancement in presence of faults	
Robust	Aardvark, Spinning, Prime, RBFT

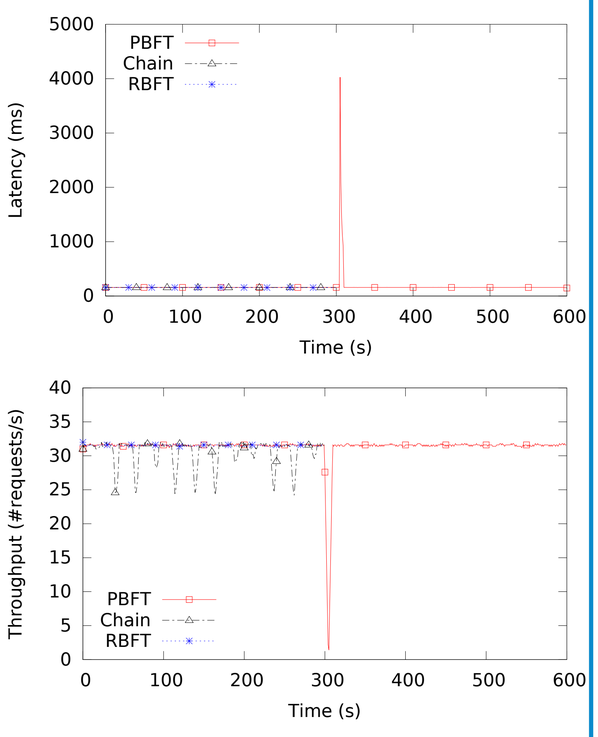
Vs

PROGRESS OF BFT BENCHMARKING TOOLS	
Achilles tool	Evaluates and detects trojan messages in PBFT
Hermes framework	Evaluates BFTSmart protocol in presence of certain byzantine attacks

BFT-Bench Architecture

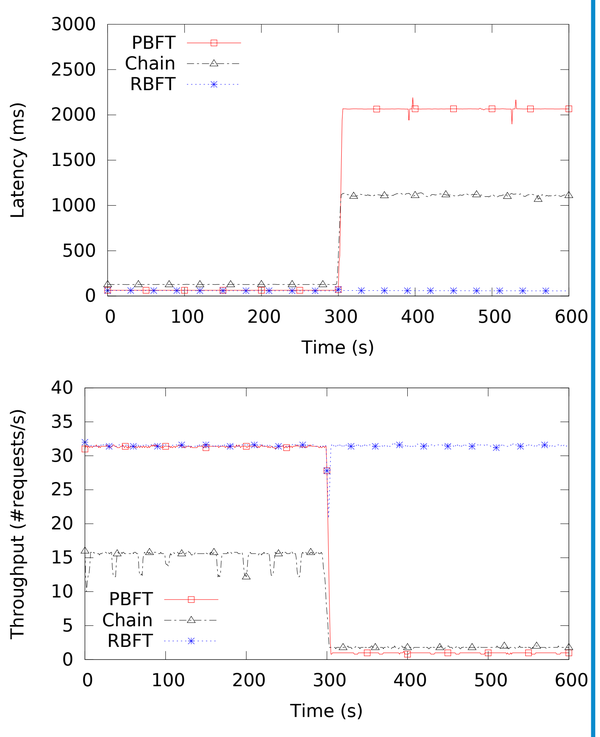


Performance Evaluation: Replica Crash



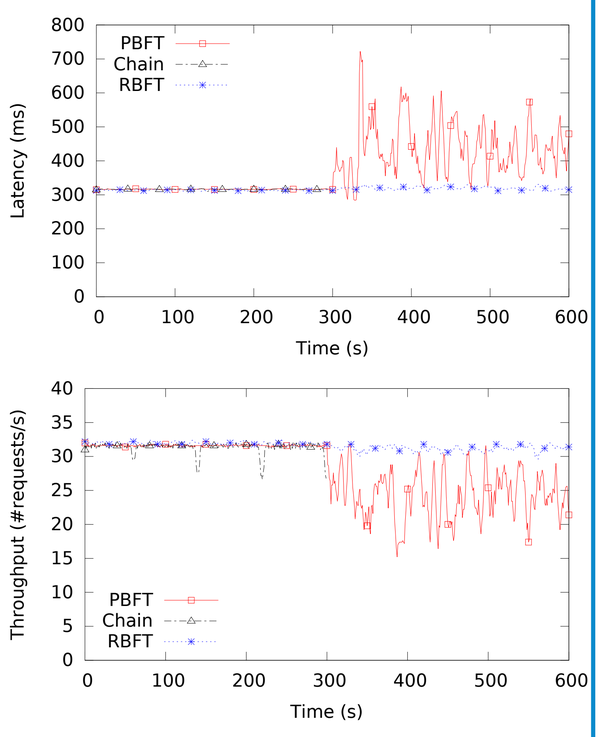
- Chain and RBFT stop to progress; Availability drops to zero.
- PBFT continues after a view change.

Performance Evaluation: Message Delay



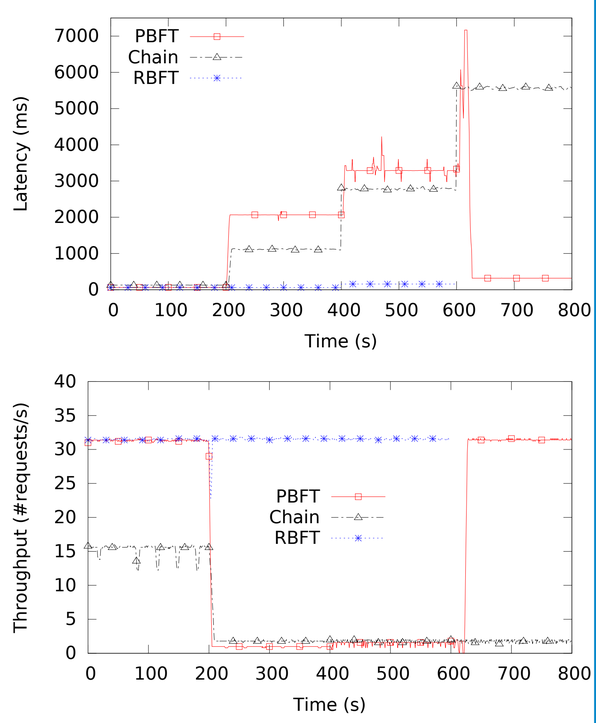
- RBFT maintains its performance while PBFT and Chain degrade.

Performance Evaluation: Network Flooding



- Chain crashes upon fault injection.
- RBFT exhibits constant performance while PBFT becomes chaotic.

Performance Evaluation: Delay with Overloading



- RBFT stops; cannot handle 10 clients.
- Chain degrades in latency.
- PBFT undergoes a view change.

Conclusion & Perspective

- Proposed BFT-Bench framework that aims to help researchers and practitioners to better analyze and evaluate the effectiveness and robustness of BFT systems.
- Experimental analysis demonstrates that BFT-Bench successfully compares various BFT protocols in face of many faulty behaviors & also exhibits the incapabilities of considered BFT prototypes.
- We aim to extend the work to integrate and analyse other prototypes of BFT protocols with BFT-Bench.