



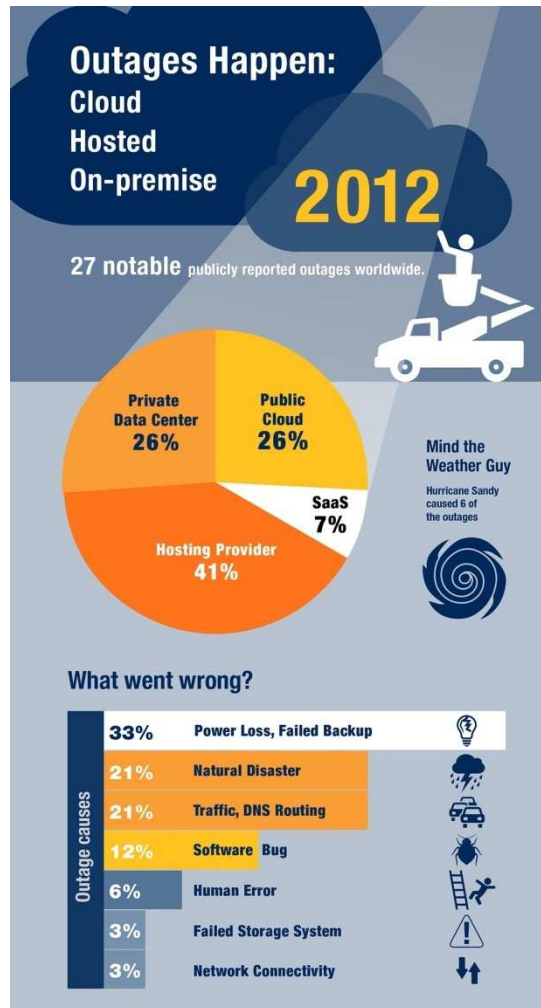
Divya Gupta<sup>\*</sup>, Lucas Perronne<sup>\*</sup>, Sara Bouchenak<sup>+</sup>

<sup>\*</sup>Univ. Grenoble Alpes, LIG Grenoble, France, <sup>+</sup>Univ. Lyon, INSA Lyon, LIRIS, Lyon, France  
divya.gupta@imag.fr, lucas.perronne@imag.fr, sara.bouchenak@insa-lyon.fr

**BFT-Bench: A Framework to Evaluate BFT Protocols**

Lucas Perronne

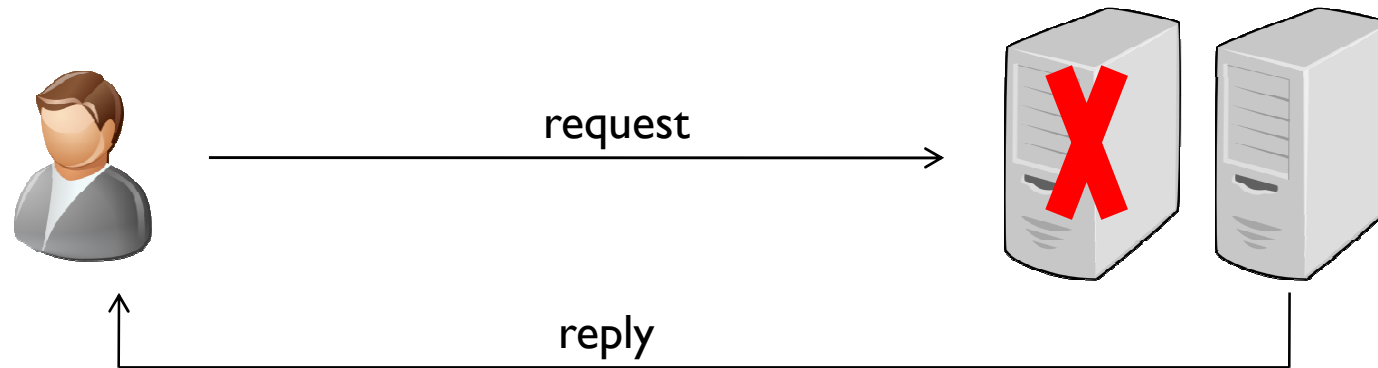
# Cloud Computing Outages



1	<b>Microsoft's Windows Azure</b> October 30 >20hrs	⚠️ A sub-component of the system failed worldwide
2	<b>Google</b> August 16 <5mins	⚠️ Services went down, causing global Internet traffic volume to plunge by about 40%
3	<b>Amazon Web Services</b> September 13 <3hrs	⚠️ Connectivity issues affected a single availability zone, disrupting a notable portion of Internet activity
4	<b>NASDAQ</b> August 22 3hrs	⚠️ Software bug, and inadequate built-in redundancy capabilities, triggered a massive trading halt in the U.S.
5	<b>OTC Markets Group Inc</b> November 7 >5hrs	⚠️ Network failure prompted a shutdown in over-the-counter stock trading in the U.S.
6	<b>HealthCare.gov</b> October 27-28 >16hrs	⚠️ Downtime caused by a service outage at Verizon Terremark data center
7	<b>Amazon.com</b> January 31 49mins	⚠️ 1hr of interrupted service may have translated to \$5M in lost revenue
8	<b>Microsoft /Hotmail/Outlook.com</b> March 13 <16hrs	⚠️ Firmware update caused servers to overheat. Hotmail and Outlook.com suffered a service loss
9	<b>Google Drive</b> March 18-20 17hrs	⚠️ Slow download times caused by a network control software glitch, resulted in latency and recovery problems
10	<b>Google's Gmail</b> September 23 12hrs	⚠️ Slow download times triggered by dual network failure affected 29% of users

# Fault Tolerance through State Machine Replication

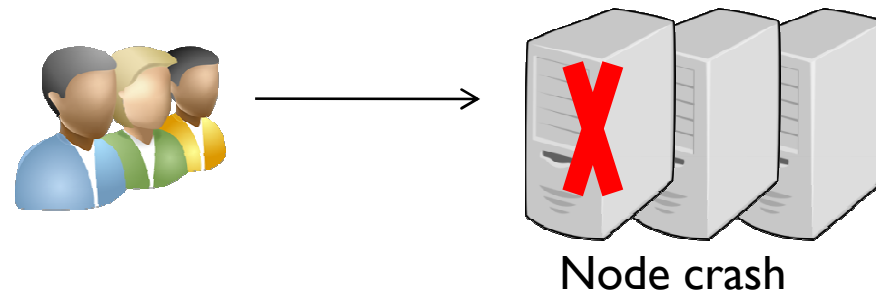
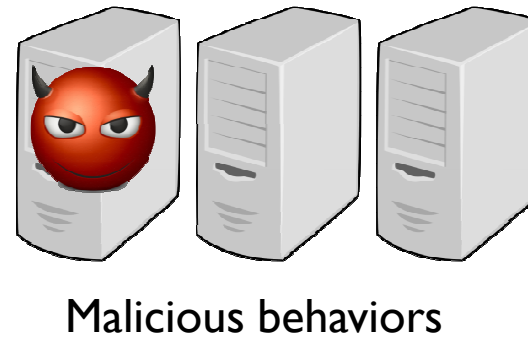
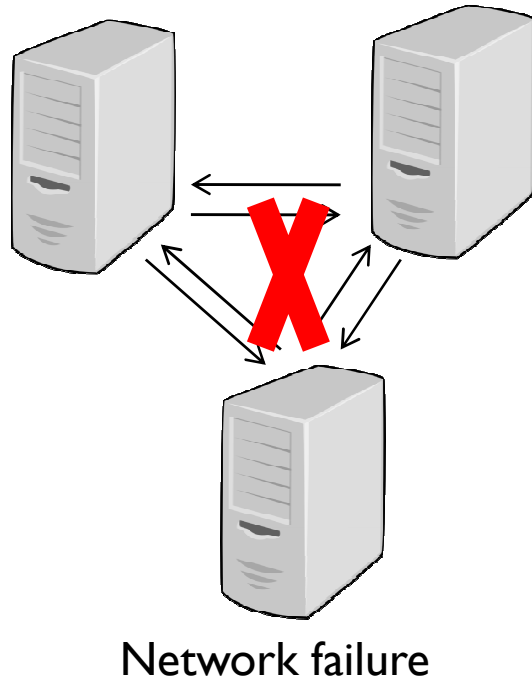
---



- ▶ The same state must be shared by correct replicas
  - ▶ *Safety*
- ▶ Correct requests must be eventually executed
  - ▶ *Liveness*

# Byzantine Fault Tolerance

---



- ▶ Providing Byzantine Fault Tolerance is the ability to ensure *Liveness* and *Safety* in presence of any **byzantine** components.

# Problem Statement

---

## ▶ **Byzantine Fault Tolerance - Related works**

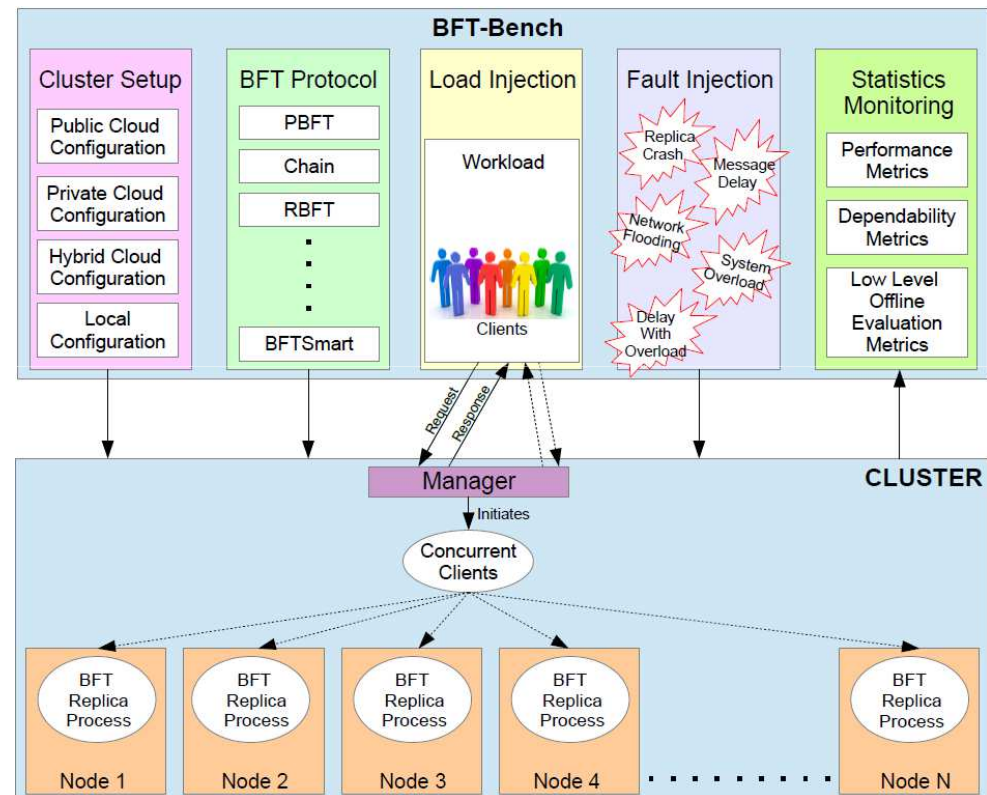
- ▶ BFT From Theory to Practice
  - PBFT
- ▶ Performance Improvement in Fault-Free Conditions
  - Zyzzyva, Aliph, ...
- ▶ Performance Improvement in Presence of Faults
  - Aardvark, RBFT, ...

## ▶ **Evaluating BFT protocols**

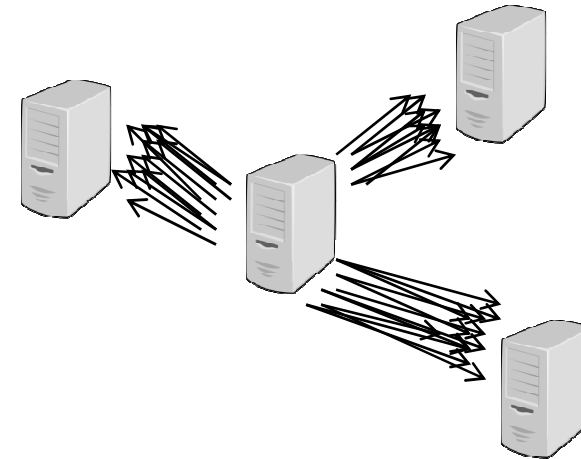
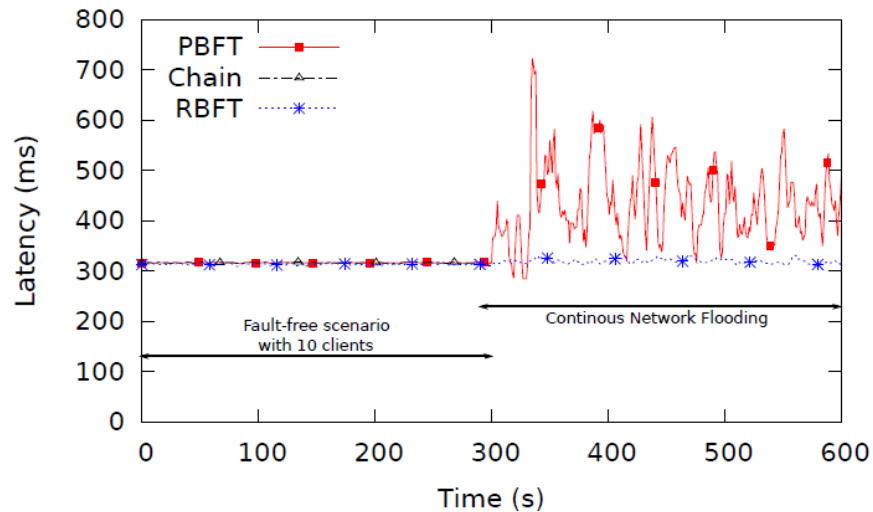
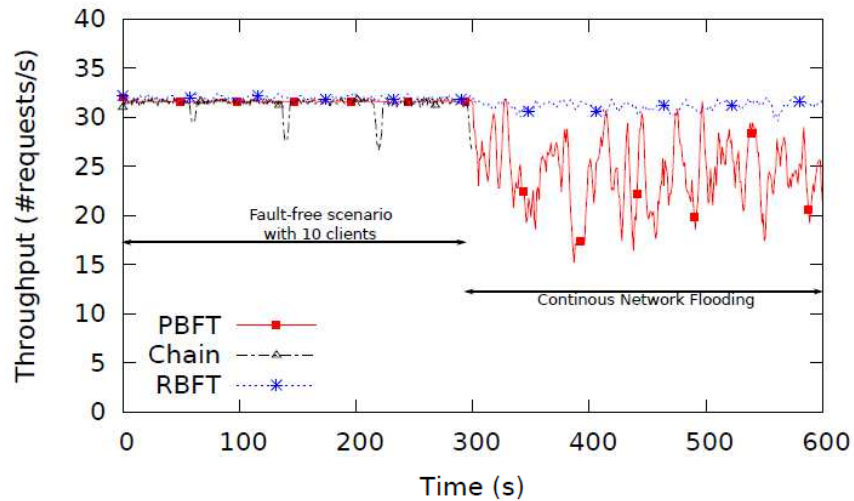
- ▶ Usage of micro-benchmarks
- ▶ Ad-hoc settings for evaluations
- ▶ No benchmarking environment dedicated to BFT

# BFT-Bench

1. Several protocols implementations
2. Various fault scenarios
3. Configurable workload
4. Statistics monitoring



# Network flooding



## Workload

Number of clients: 10

Request size: 4Kb

Computational time: 30 ms

## Faultload

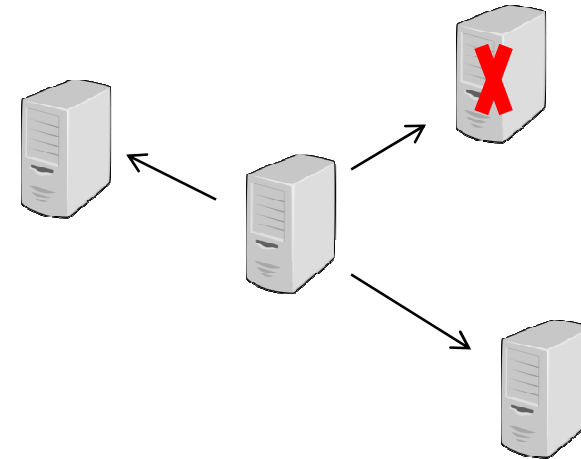
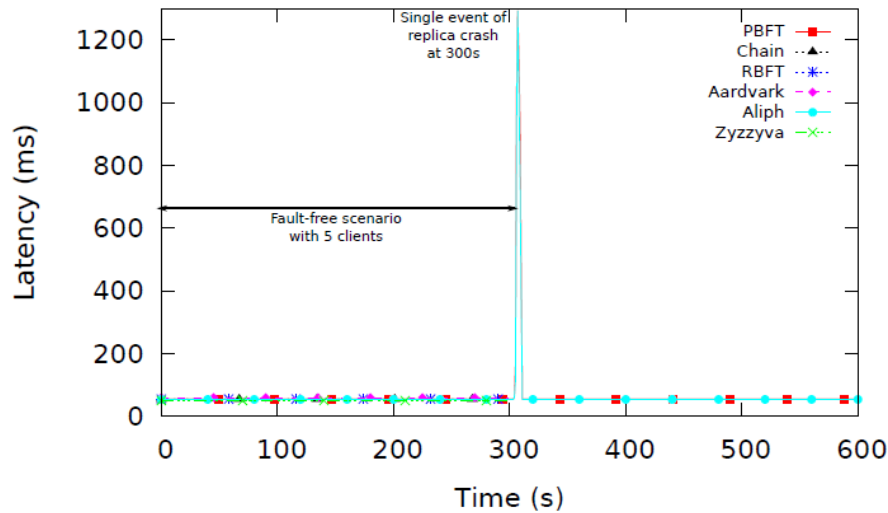
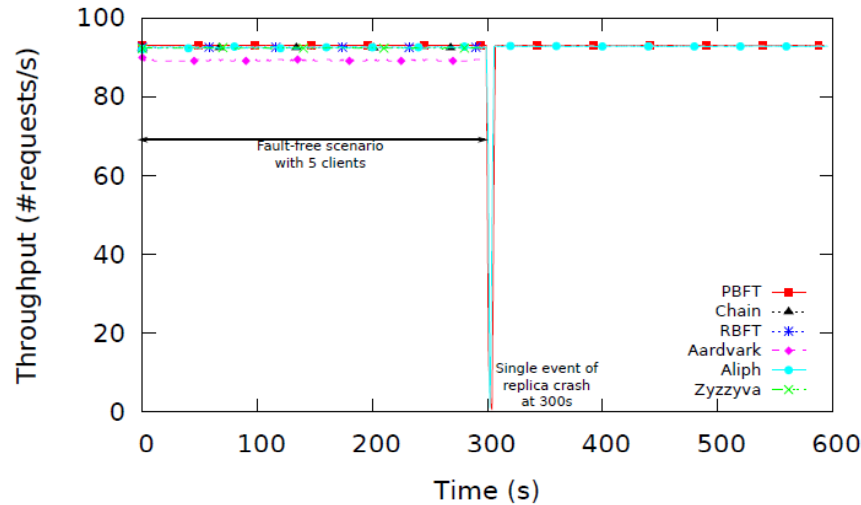
Type of fault: **Network Flooding**

Fault trigger time: 300s

Faulty node: Non-primary replica

Message size: 4Kb

# Replica Crash



## Workload

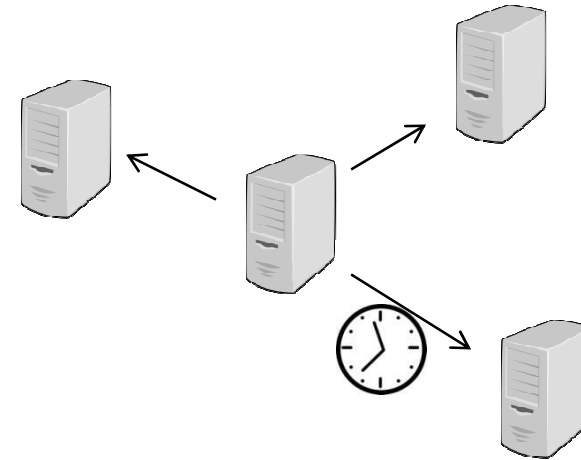
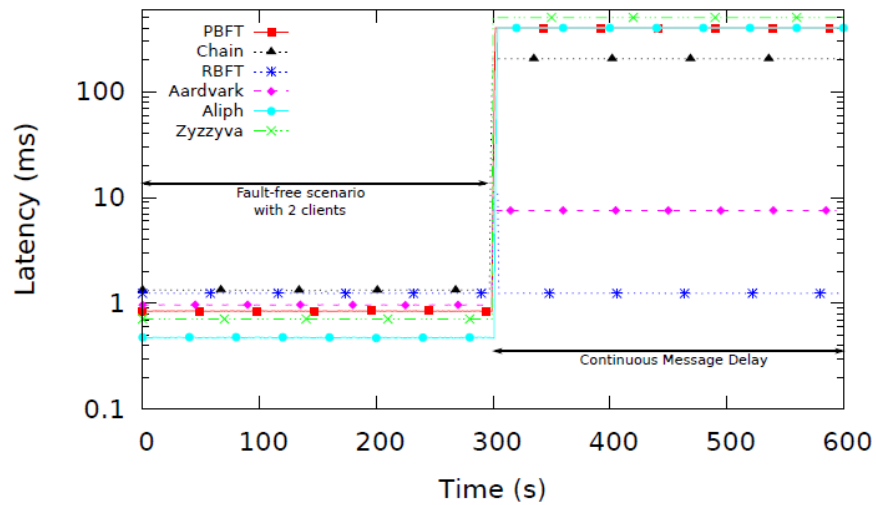
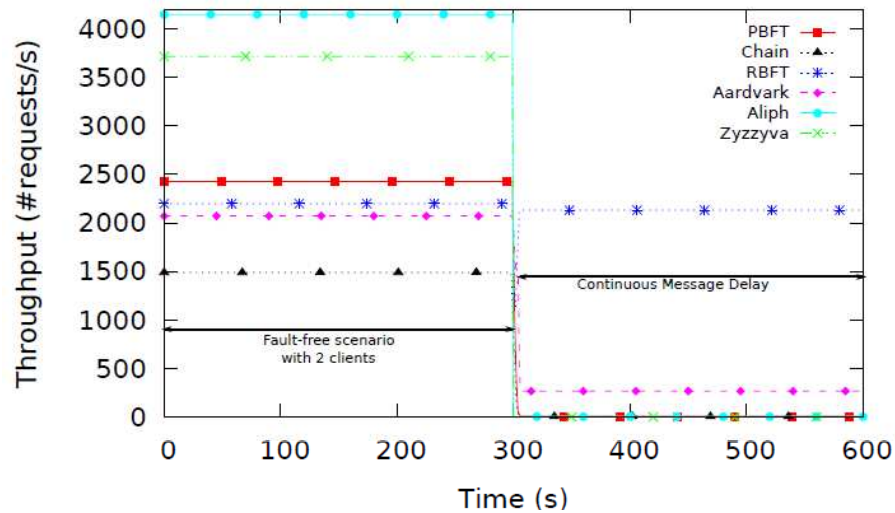
Number of clients: 5  
 Request size: 4Kb  
 Computational time: 10 ms

## Faultload

Type of fault: **Replica Crash**  
 Fault trigger time: 300s  
 Faulty node: Primary replica



# Message delay



## Workload

Number of clients: 2  
 Request size: 4Kb  
 Computational time: 100  $\mu$ s

## Faultload

Type of fault: *Message delay*  
 Fault trigger time: 300s  
 Faulty node: Primary replica  
 Delay time: 100 ms

# Conclusion & Perspectives

---

- BFT-bench includes several state-of-the-art BFT protocols
- It implements many faulty behaviors on different workloads

## Perspectives

- Additional Protocols
  - More faulty scenarios
- 
- <https://bftbench.gforge.liris.cnrs.fr/>

