

Byzantine Consensus in the Jungle

Geographically-Scalable BFT with Adaptive Weighted Replication

Alysson Bessani



Joint work with João Sousa, Christian Berger, and Hans P. Reiser



Byzantine Fault Tolerance Protocols

- **Performance**

- The racehorses: PBFT, Zyzzyva, Alyph, ...

- **Robustness**

- Slow but steady: Prime, Aardvark, RBFT, ...

- **Resource efficiency**

- Strong assumptions: MinBFT, CheapBFT, XFT, ...

- **Scalability**

- Blockchainers: HoneyBadger, FastBFT, SBFT, ...

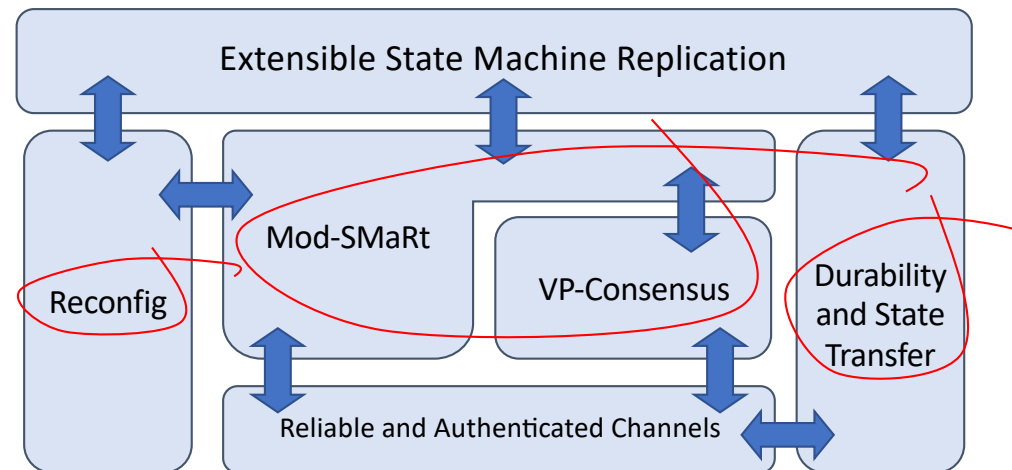
BFT-SMaRt

Sousa, Bessani. From Byzantine Consensus to BFT State Machine Replication: A Latency-optimal Transformation. EDCC'12.

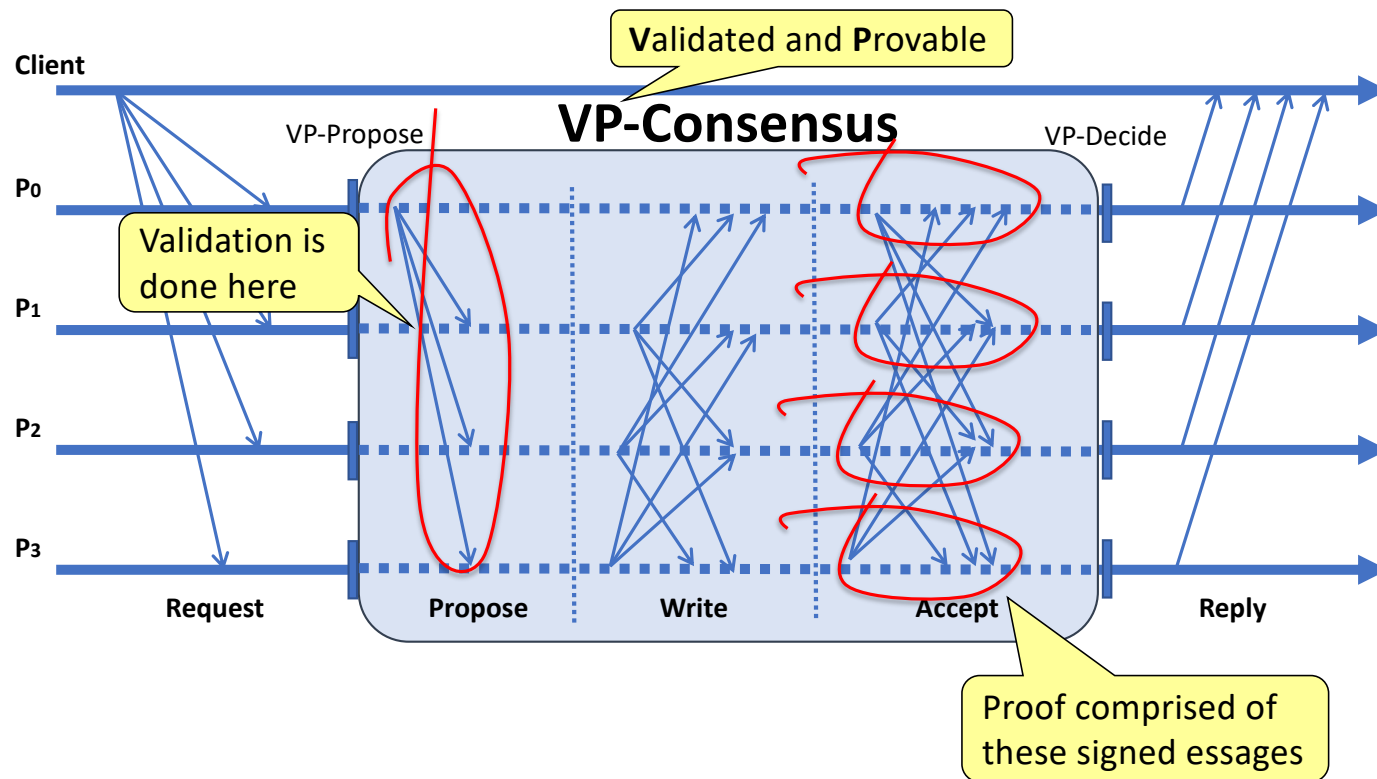
Bessani, Sousa, Alchieri. State Machine Replication for the Masses with BFT-SMaRt. IEEE/IFIP DSN'14.

BFT-SMaRt

- Byzantine/Crash fault tolerant state machine replication library
 - Written in Java, maintained and evolved during more than 10 years
 - Available under Apache license: <http://bft-smart.github.io/library/>
- Key features: modularity, reconfigurations, robustness, performance

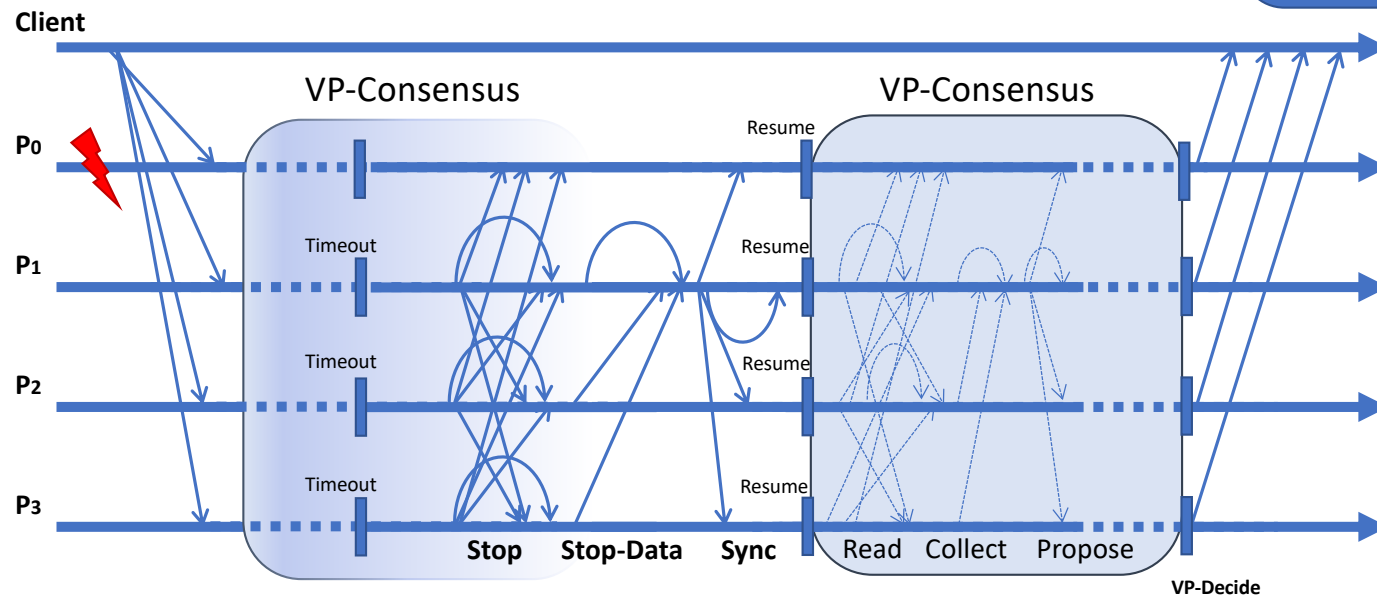


Mod-SMaRt: Normal Phase



Mod-SMaRt: Synchronization Phase

IMPORTANT:
It looks like
PBFT, but it is
not PBFT 😊



Some Facts about BFT Consensus in WANs

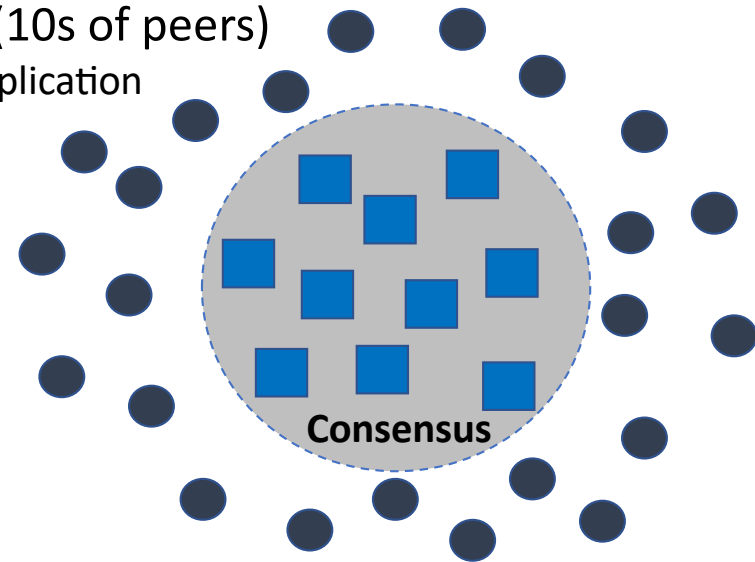
- There's not much experience with BFT consensus in production on the internet
 - Permissionless blockchains solve eventual consensus
 - (as far as I know) There's no BFT consensus in production on the Internet
 - Stellar and Ripple is the closest we have...
 - Even CFT systems (Paxos, RAFT) are rarely used in this context
- Decentralization and fault independence requires BFT consensus peers to be deployed on different sites
 - Otherwise, it is difficult to justify the use of BFT?

Some Facts about BFT Consensus

- Node-scalability is not always required for BFT
 - Current consortiums typically are small (10s of peers)
 - Libra implements classical state machine replication
 - It aims to 100 validators at launch



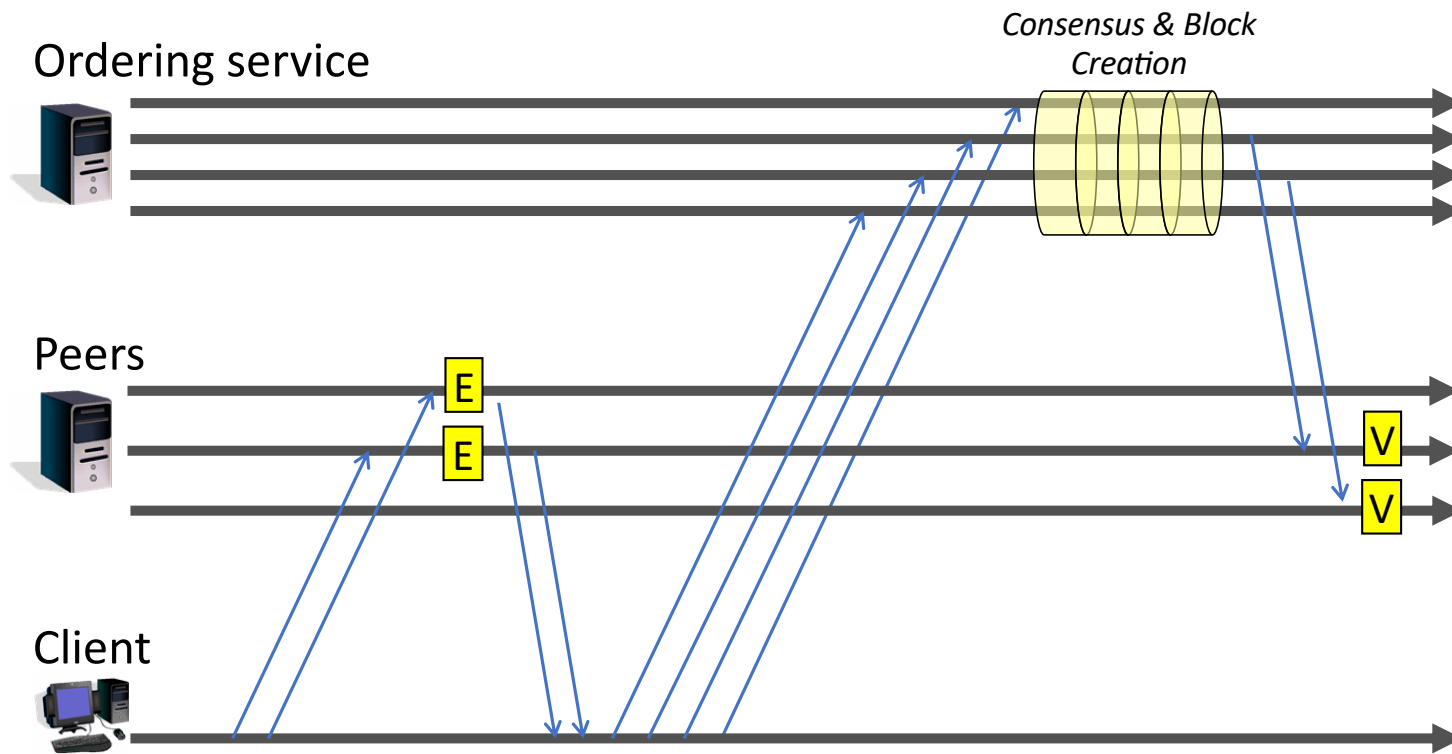
■ Validator
● Client



- Most permissioned systems tend to isolate consensus in a subset of peers

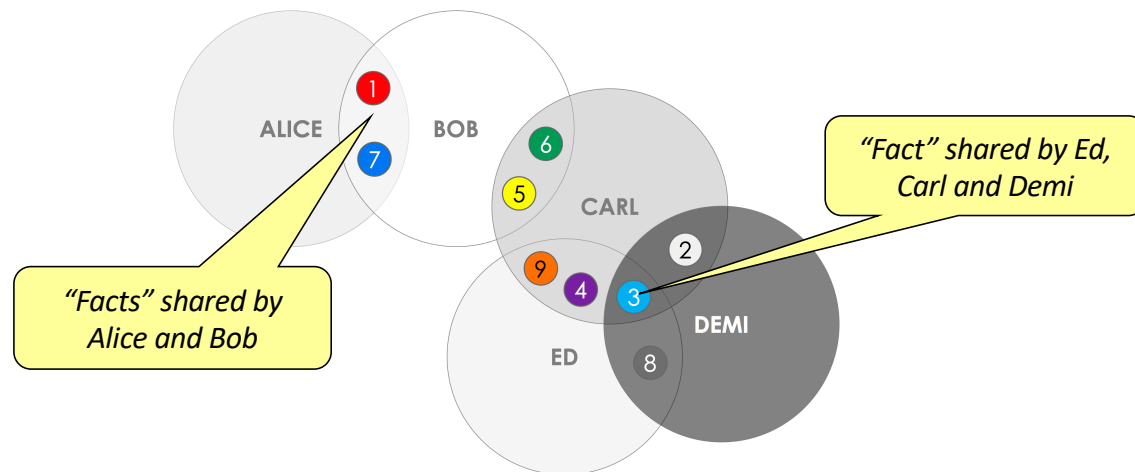


HYPERLEDGER FABRIC



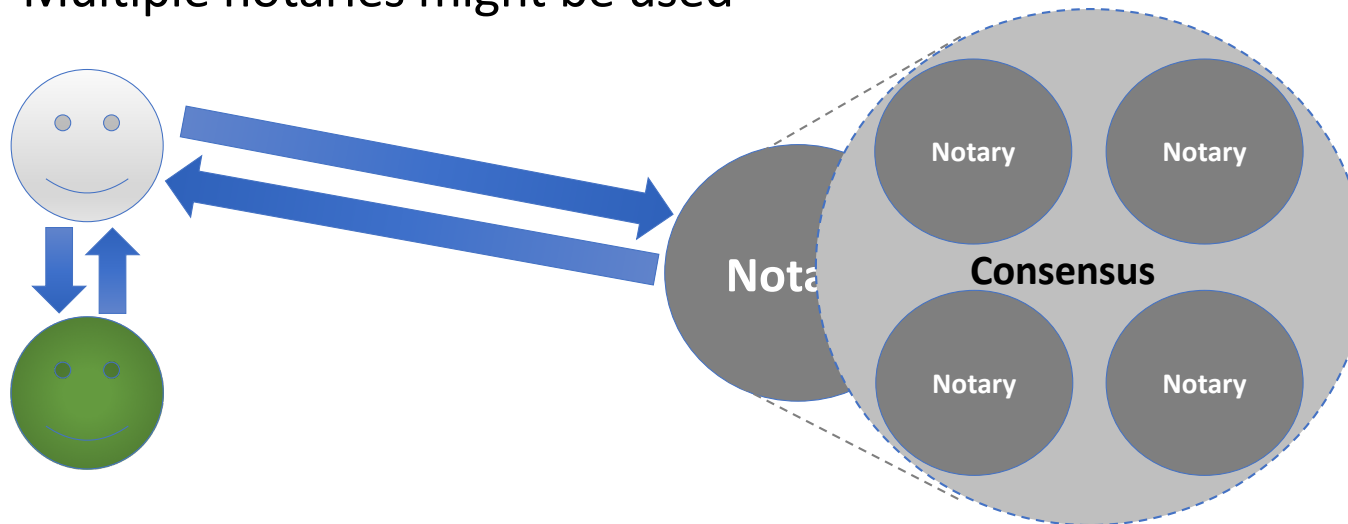
corda

- Open-source blockchain project targeting (at least initially) the financial market
- Key idea: **there is no shared global ledger**
 - Instead, **there are many distributed ledgers**



corda

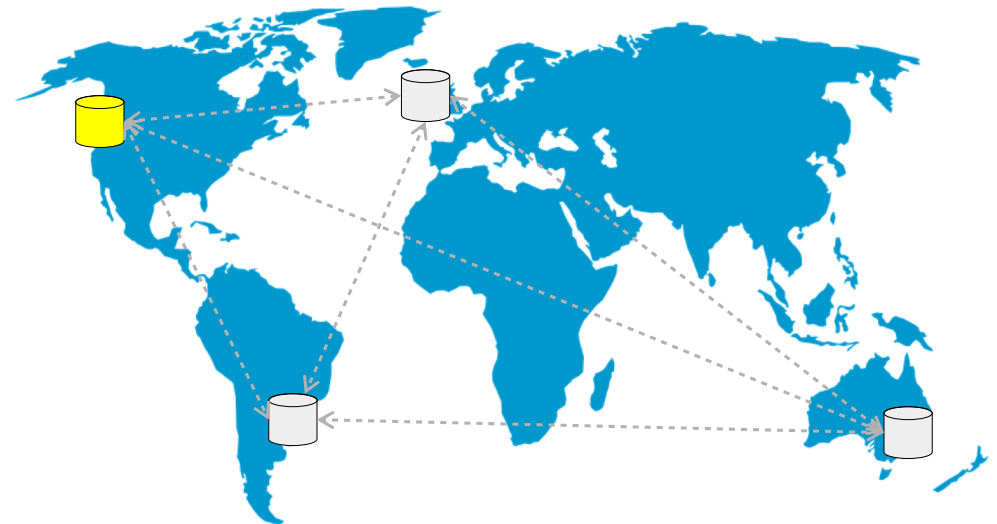
- Notary implements a key-value store that register all state “consumptions”
- Some specific transaction validation might be executed
- Multiple notaries might be used



Geographically-Scalable BFT

Issues with Geo-Replication

- Different administrative domains
- Performance diversity
 - Across replicas
 - Across time
- Throughput can be improved with better networks
- Latency requires protocol **optimizations**
 - Speed of light is the network limit
 - Latency proportional to the roundtrip to a fast quorum

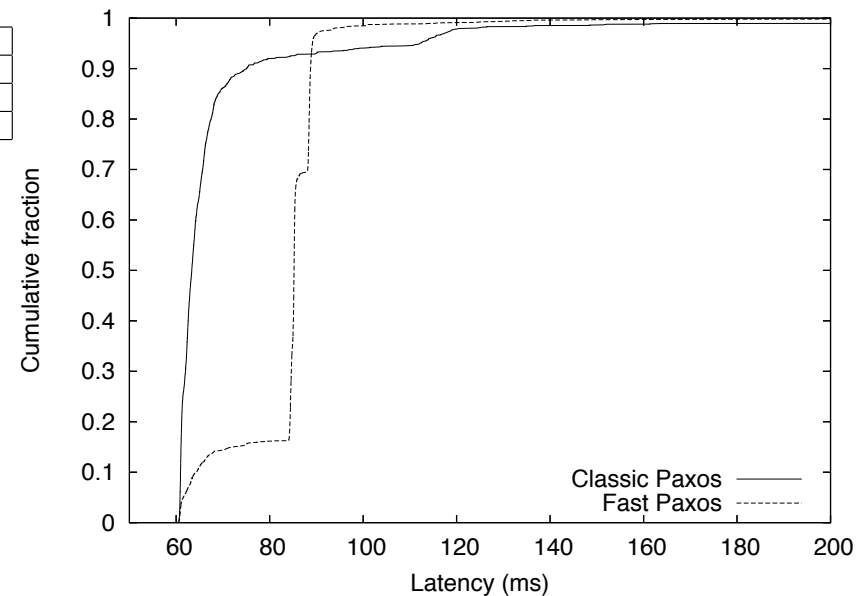


Classic vs Fast Paxos

	Classic Paxos	Fast Paxos
Comm. steps	3	2
Number of replicas	$2t + 1$	$3t + 1$
Quorum size	$t + 1$	$2t + 1$

Comparison is made through trace-driven simulations using latencies from 2002 obtained from the internet weather service.

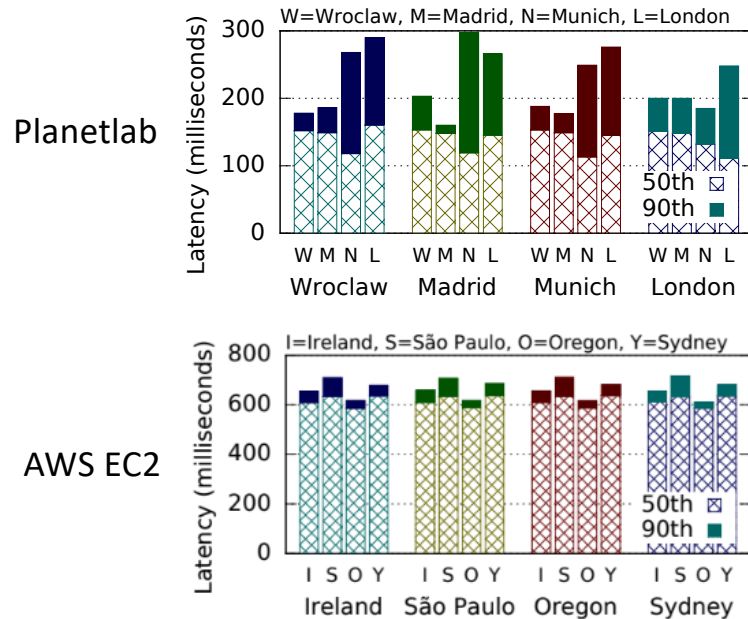
Classic Paxos is faster than Fast Paxos 60% of times



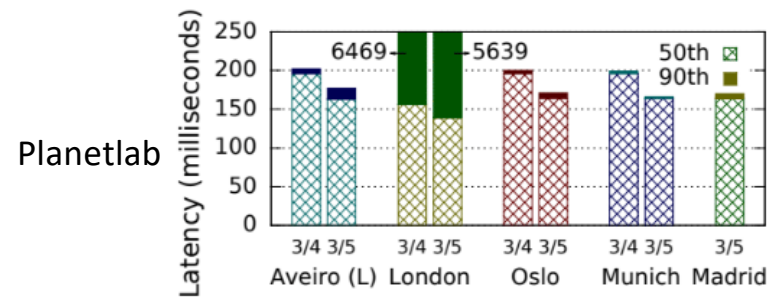
Flavio Junqueira, Yanhua Mao, and Keith Marzullo. Classic Paxos vs. fast Paxos: caveat emptor. Proc. of the 3rd workshop on on Hot Topics in System Dependability (HotDep'07). 2007.

Experimental study conducted with BFT-SMaRt on Planetlab and Amazon EC2

Leader location



Quorum size



Summary:

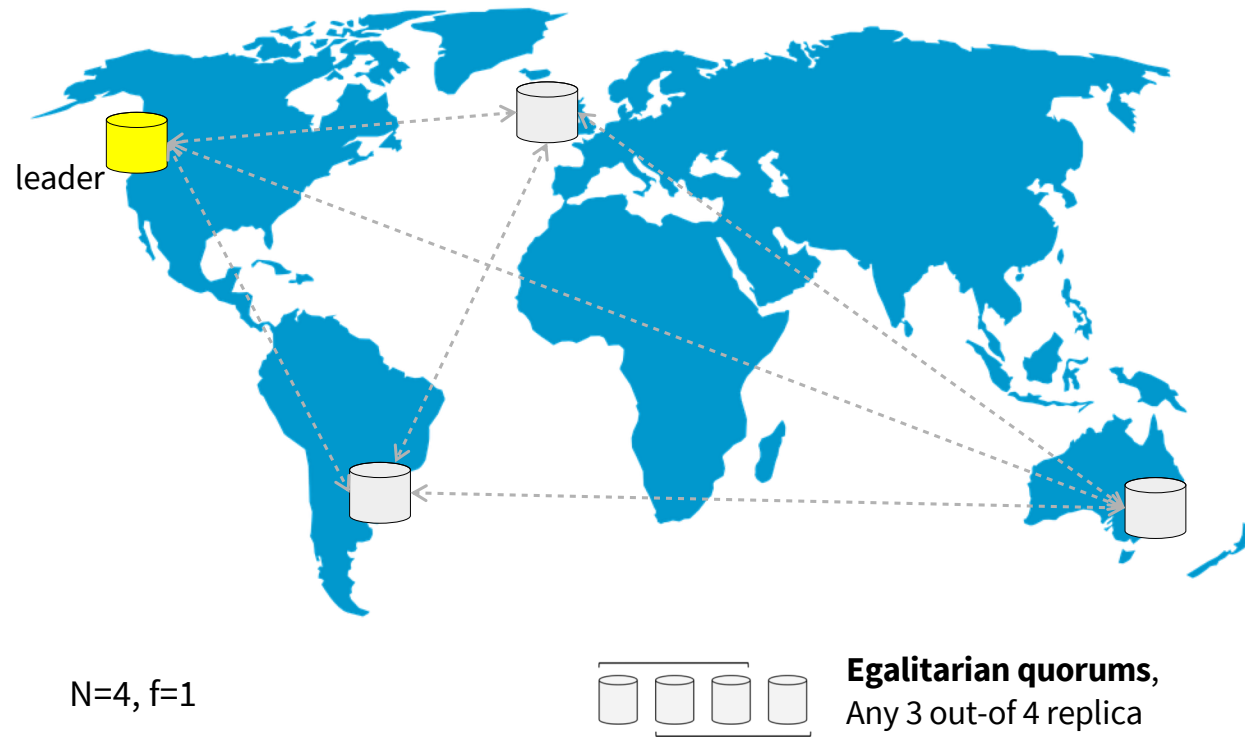
- Leader in the best-connected site yields better results than employing a rotating or multiple leader(s) strategy
- Smaller quorums create opportunities for improving latency

Our solution: WHEAT + AWARE

Sousa, Bessani. Separating the WHEAT from the Chaff: An Empirical Design for Geo-replicated State Machines. IEEE SRDS'15.

Berger, Reiser, Sousa, Bessani. Resilient Wide-area Byzantine Consensus using Adaptive Weighted Replication. IEEE SRDS'19.

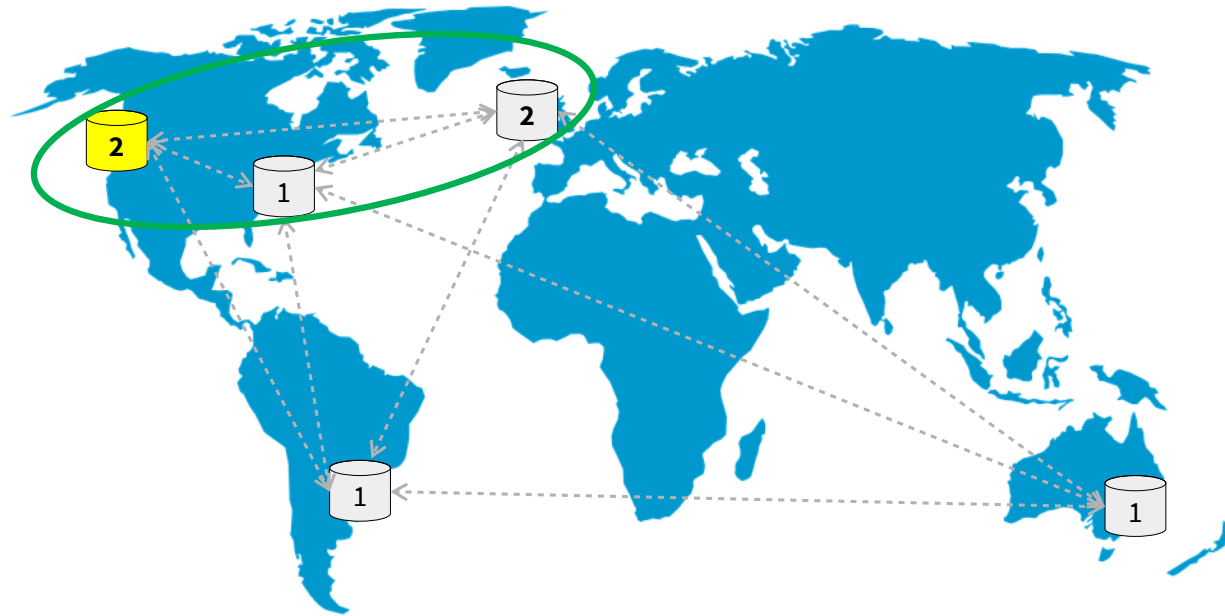
Classical BFT Replication



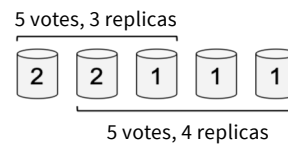
WHEAT: WeigHt-Enabled Active replicaTion

- Use optimizations that lead to significant latency reduction:
 - Single leader in the best-connected site
 - Tentative executions (from PBFT)
 - Employs smaller quorums (weighted replication)
- Weighted replication: safe voting assignment scheme for SMR
 - Uses **Δ extra replica(s)** for quorum formation
 - Improves latency by enabling more choice upon quorum formation
 - Needs a to preserve quorum intersection and tolerance to f faulty replicas

Weighted BFT Replication



$N=4$, $f=1$, $\Delta=1$ (extra)



Weighted quorums,
One set of 3 out-of-5
and any set 4 out-of-5

Weighted BFT Replication

- **Consistency:** All quorums that hold Q votes intersect by at least one correct replica
- **Availability:** There is always a quorum available in the system that holds Q votes
- **Safe minimality:** There exists at least one minimal quorum in the system

Weighted BFT Replication

Define the number of replicas u that hold $V_{max} > 1$ votes, without violating f

CFT mode

$$n = 2f + 1 + \Delta$$

$$N_v = \sum V_i = 2F_v + 1$$

$$Q_v = F_v + 1$$

$$u = f$$

Input:
 f and Δ

BFT mode

$$n = 3f + 1 + \Delta$$

$$N_v = \sum V_i = 3F_v + 1$$

$$Q_v = 2F_v + 1$$

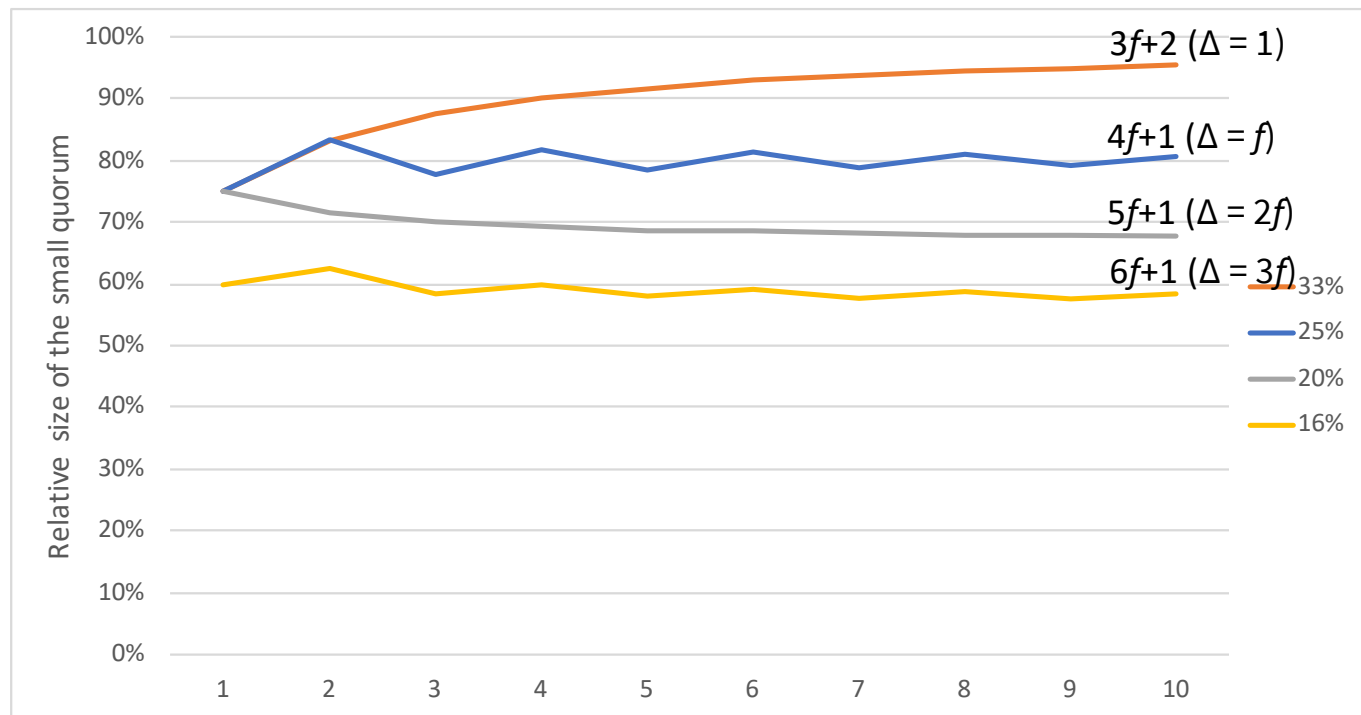
$$u = 2f$$

$$F_v = \Delta + f$$

$$V_{max} = \frac{\Delta + f}{f} = 1 + \frac{\Delta}{f}$$

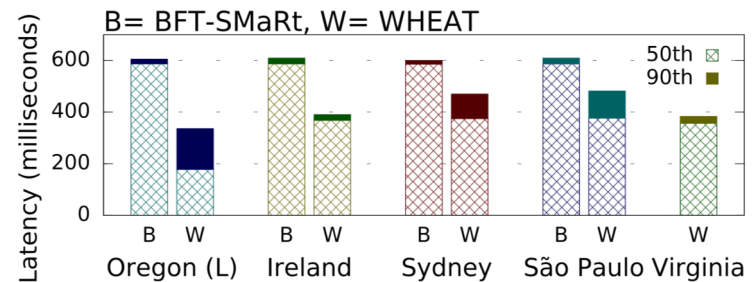
Output:
 u and V_{max}

Size of fast quorums with different f and Δ



AWARE: Adaptive Wide-Area REplication

The benefit of weighted replication depends on **choosing an optimal weight configuration**



- The environment of the system (i.e., network characteristics) may **change at runtime** (e.g., due to a DDoS attack)



AWARE enables a geo-replicated consensus-based system to **adapt to its environment!**

AWARE Approach



- **Self-Monitoring**

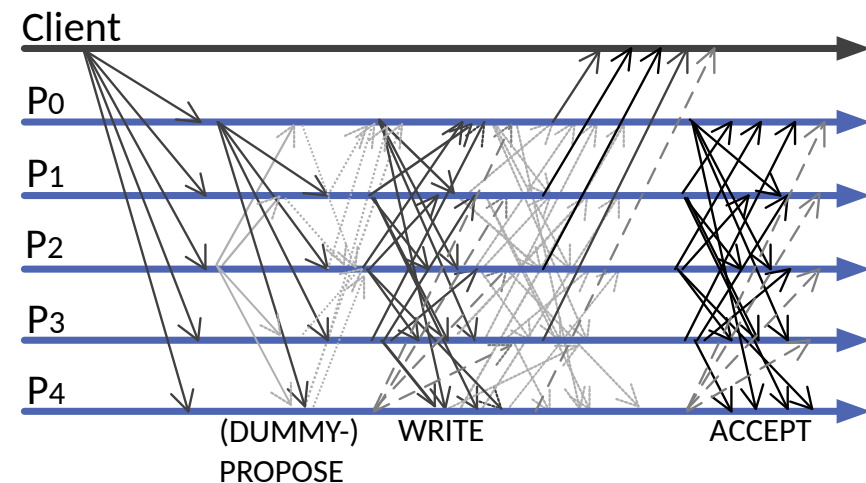
- AWARE uses reliable self-monitoring as decision-making basis for adapting replicas' voting weights and leader position at runtime

- **Self-Optimization**

- AWARE continuously strives for consensus latency gains at runtime
- Changes weights and leader location to minimize consensus latency

Self-Monitoring: Measuring Latency

- Each replica measures its point-to-point latency to other replicas for consensus protocol messages
- **Non-Leader's Propose**
 - Periodically an alternately selected dummy leader broadcasts a dummy proposal
- **Write-Response**
 - Replicas immediately respond by sending acknowledgments



Self-Monitoring: Consolidating Measurements

- Replicas periodically disseminate their measurements to others with **total order** until they have the same latency matrices
- AWARE maintains **synchronized matrices** for both PROPOSE and WRITE latencies \hat{M}^P and \hat{M}^W used for decisions later

	Oregon	Ireland	Sydney	Sao Paulo	Virginia
Oregon	0	65	69	92	40
Ireland	65	0	132	93	38
Sydney	69	132	0	158	105
Sao Paulo	92	93	158	0	61
Virginia	40	38	105	61	0

Self-Optimization

- With the same matrices \hat{M}^P and \hat{M}^W the replicas can **solve deterministically** the following optimization problem:

$$\langle \hat{l}, \hat{W} \rangle = \arg \min_{W \in \mathfrak{W}, l \in \mathfrak{L}} \textit{PredictLatency}(l, W, \hat{M}^P, \hat{M}^W)$$

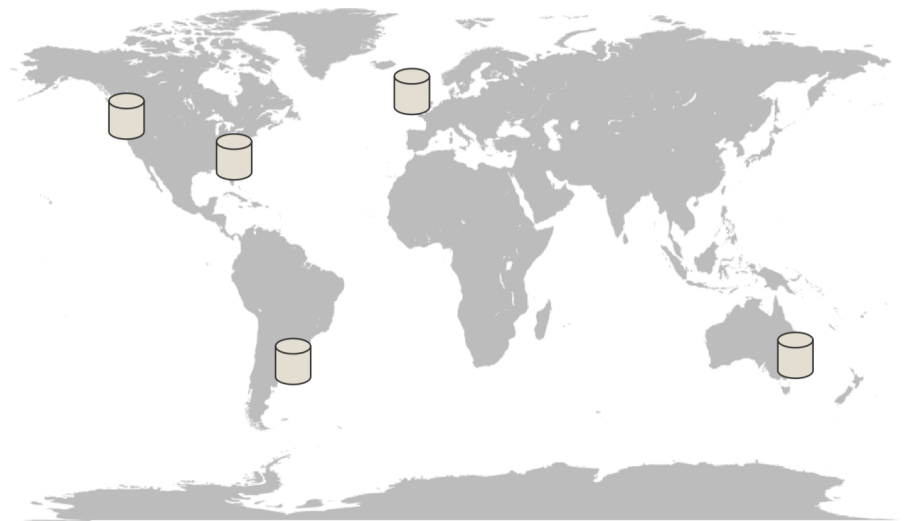
- All correct replicas reach the same, optimal weight distribution and invoke a reconfiguration in the system



Evaluation of WHEAT and AWARE

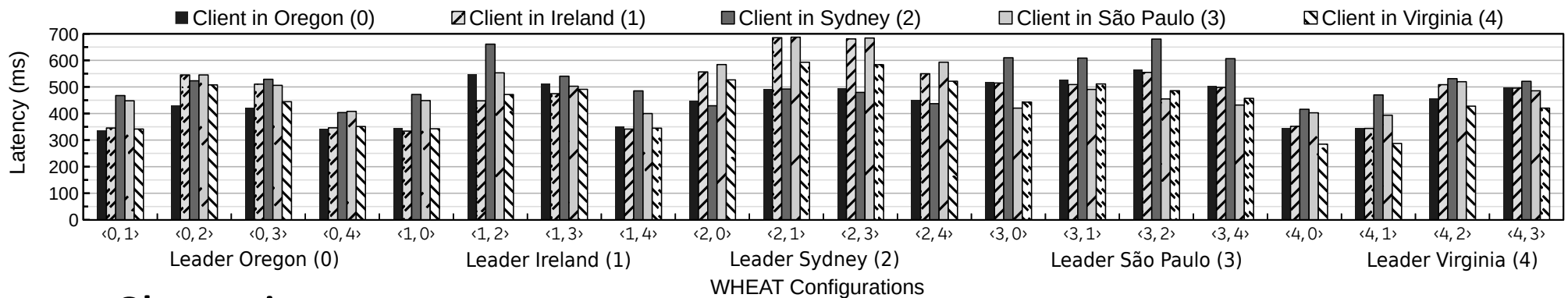
Setup

- **AWARE** is implemented on top of WHEAT, which is based on BFT-SMaRt
- For evaluation, we use the **Amazon AWS cloud**, using EC2 instances of t2.micro type with 1 vCPU, 1 GB RAM and 8 GB SSD volume
- We select regions **Oregon, Ireland, Sydney, São Paulo** and **Virginia** for instances (1 client and 1 replica on each instance)
- Clients simultaneously send 1kB-requests across all sites



Clients' Request Latency

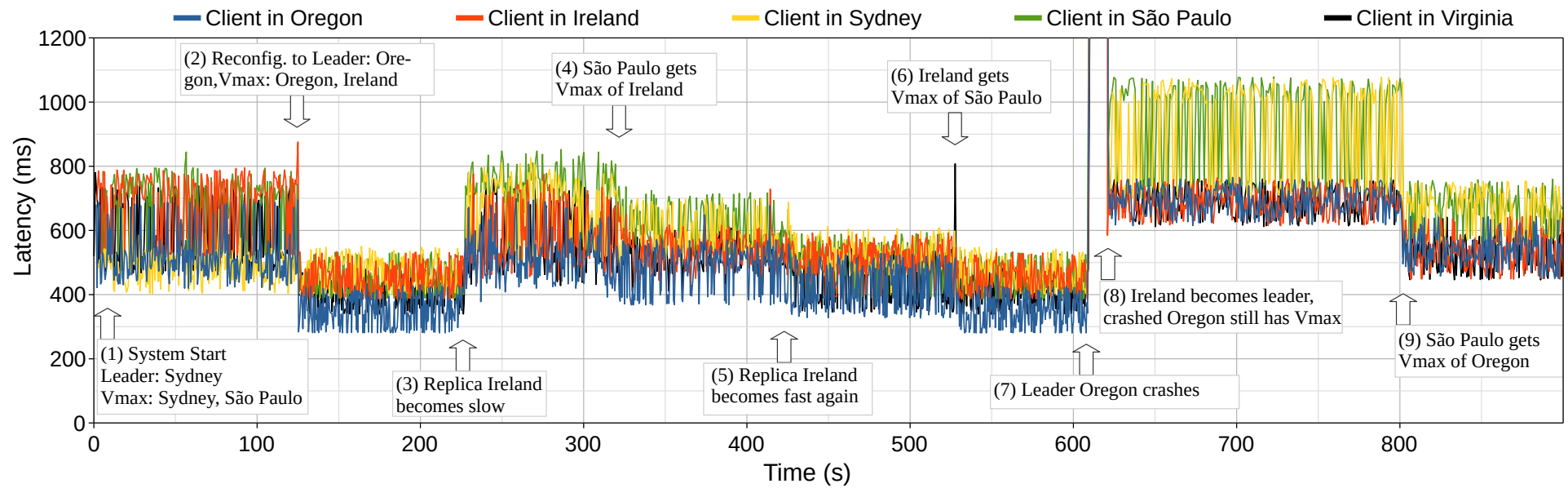
Average latencies of all clients and 20 configurations



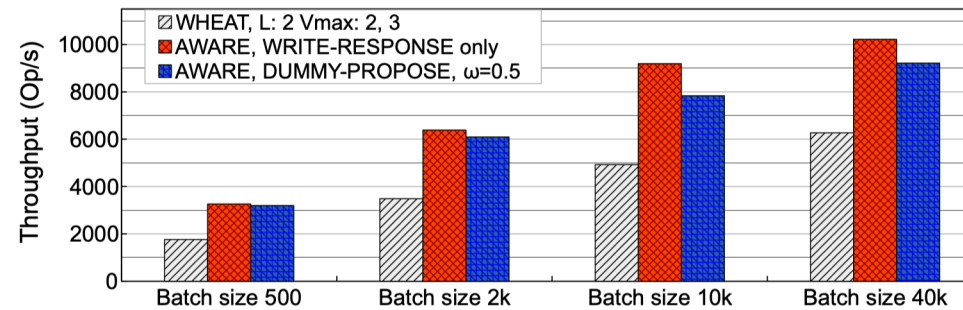
Observations

- The best configuration (<4,0>) performs about 39% faster than the median (<3,4>), 64% faster than the worst (<2,1>)
- Tuning voting weights can reduce latency (see configs. with the same leader)
- Leader relocation may be necessary for achieving optimal consensus latency

Runtime Behavior of AWARE



AWARE Throughput



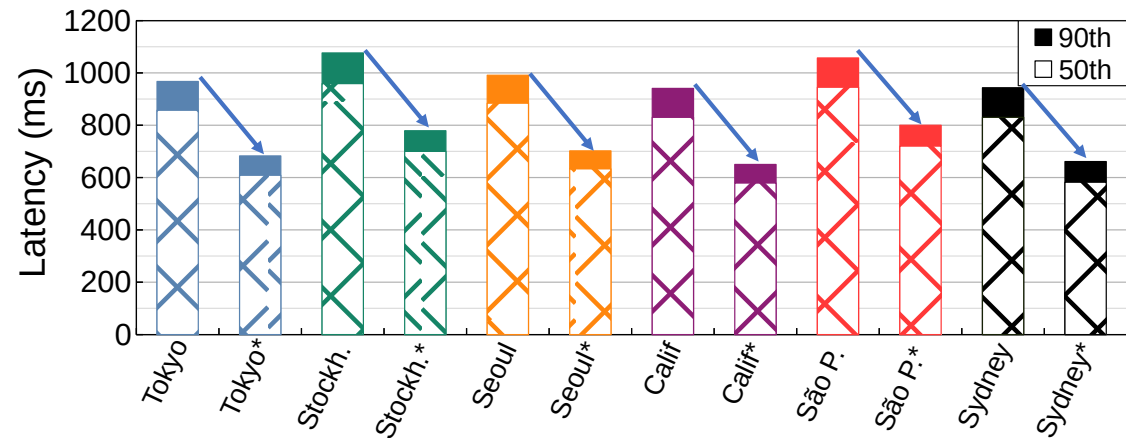
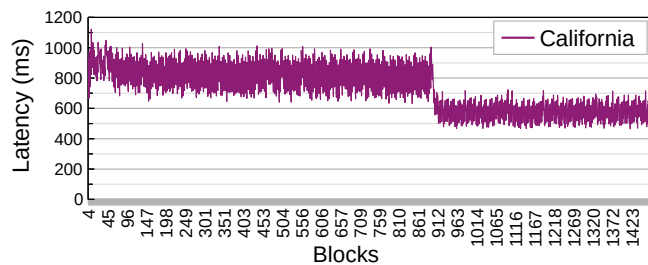
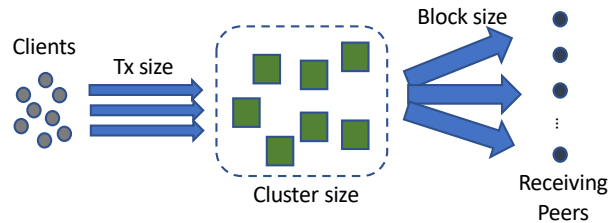
Observations

- Low consensus latency indeed has positive effects on throughput for different batch sizes
- The monitoring overhead induced by the Dummy-Propose is noticeable, but still passable, given that AWARE's main ambition is latency optimization



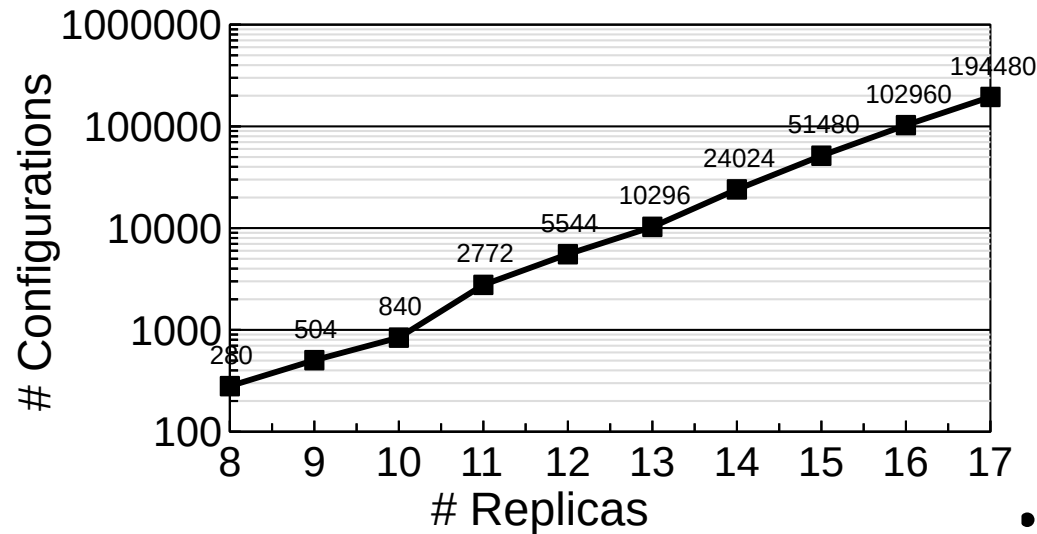
HYPERLEDGER
FABRIC

BFT Ordering with AWARE



Latency across clients before and after optimization*

AWARE with More Nodes: the challenge



- Number of configurations explodes

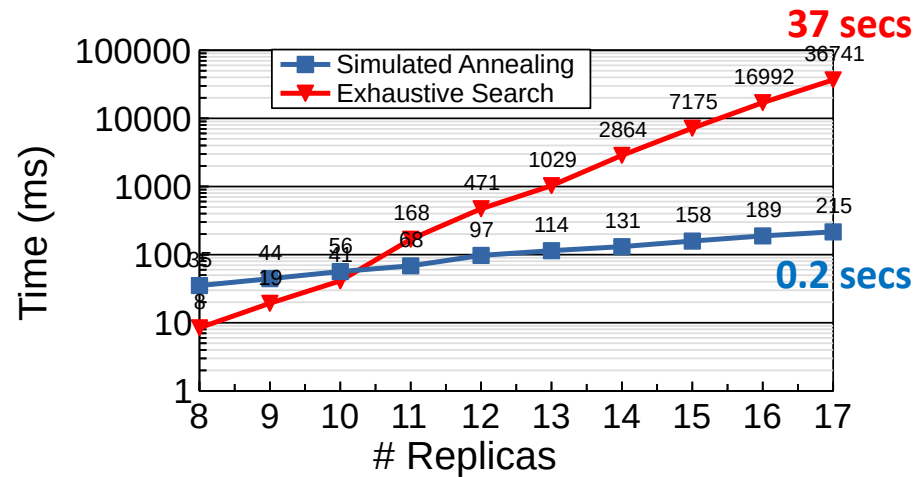
Number of weight distribution possibilities

$$\binom{3f + 1 + \Delta}{2f} \cdot 2f = \frac{\prod_{i=2f}^{3f+1+\Delta} i}{(f + 1 + \Delta)!}$$

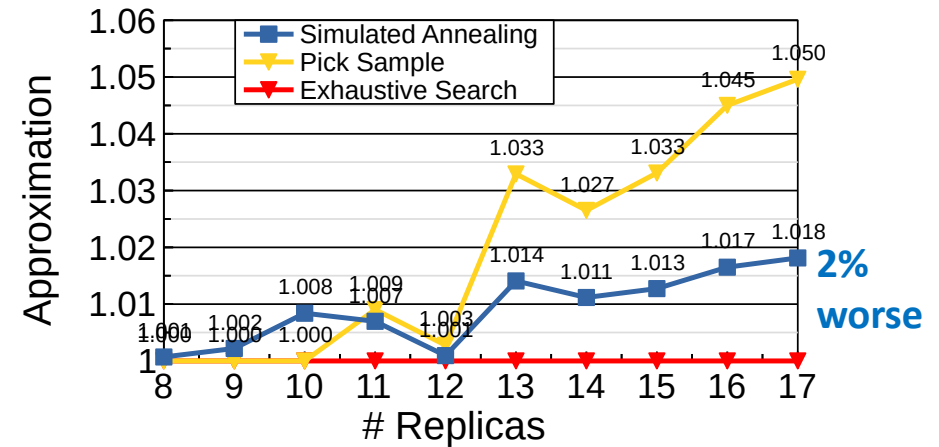
Possible leader location

- Finding the best configuration becomes a huge challenge

AWARE w/ More Nodes: simulated annealing

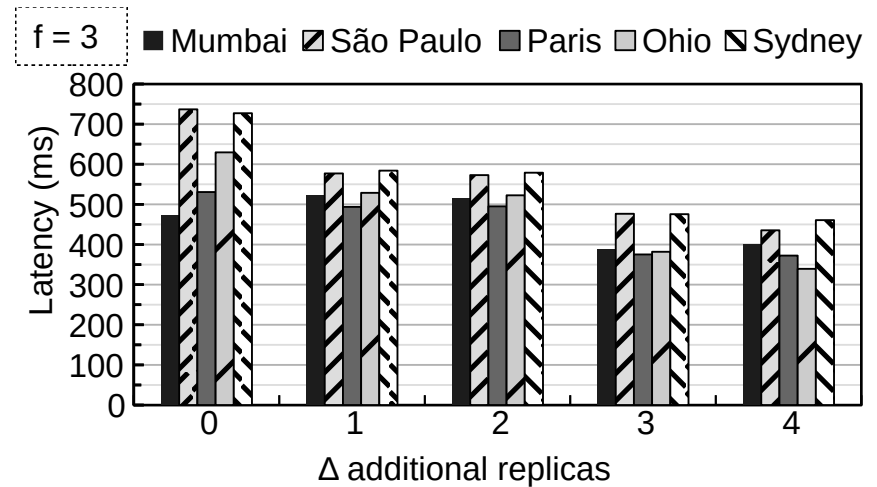
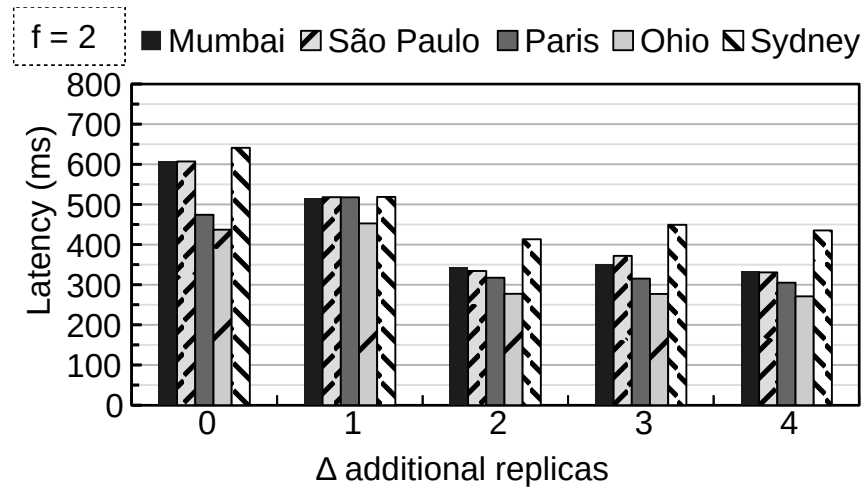


(b) Computation time.



(c) Approximation quality.

AWARE with More Nodes



Summary: WHEAT + AWARE

- **Ease of deployment**

- AWARE provides the needed automation for finding an optimal configuration by tuning voting weights and/or relocating the leader

- **Adjust to varying conditions**

- AWARE dynamically adjusts to changing conditions by shifting high voting power to replicas that are the fastest in a recent time frame

- **Compensate for faults**

- AWARE detects (non-malicious) high-weight replicas failures and restores the availability of up to $f(V_{max} - V_{min})$ voting power by redistributing high weights

- **Ultimately, it is a way to deal with heterogeneity**

Questions?

- Alysson Bessani
 - anbessani@fc.ul.pt
 - www.di.fc.ul.pt/~bessani



- To know more:
 - BFT-SMaRt & BFT Fabric Orderer: <https://github.com/bft-smart/>
 - Sousa, Bessani. *From Byzantine Consensus to BFT State Machine Replication: A Latency-optimal Transformation*. EDCC'12.
 - Bessani, Sousa, Alchieri. *State Machine Replication for the Masses with BFT-SMaRt*. IEEE/IFIP DSN'14.
 - Sousa, Bessani. *Separating the WHEAT from the Chaff: An Empirical Design for Geo-replicated State Machines*. IEEE SRDS'15.
 - Berger, Reiser, Sousa, Bessani. *Resilient Wide-area Byzantine Consensus using Adaptive Weighted Replication*. IEEE SRDS'19.

