# BFT State Machine Replication with 2f+1 Replicas

*What good are hybrid models and what hybrid models are good*

Miguel Correia

joint work with Paulo Veríssimo, Nuno Neves, Alysson Bessani, Giuliana Veronese, Lau C. Lung

TÉCNICO LISBOA

inesc id lisboa

## Outline

- 2002: Wormholes, TTCB, BRM
- 2004: BFT-TO and TOW
- 2007: A2M-PBFT-EA
- 2008-…: MIN-BFT, EBAWA, USIG
- 2010: 2f+1 Consensus

2

# 2002: WORMHOLES, TTCB, BRM

M. Correia, P. Veríssimo, Nuno F. Neves. **The Design of a COTS Real-Time Distributed Security Kernel**. *In Proceedings of the Fourth European Dependable Computing Conference.* Toulouse, France, pages 234--252, October 2002.

M. Correia and L. C. Lung and N. F. Neves and P. Veríssimo. **Efficient Byzantine-Resilient Reliable Multicast on a Hybrid Failure Model**. *In Proceedings of the 21th IEEE Symposium on Reliable Distributed Systems*. Suita, Japan, pages 2--11, October 2002.

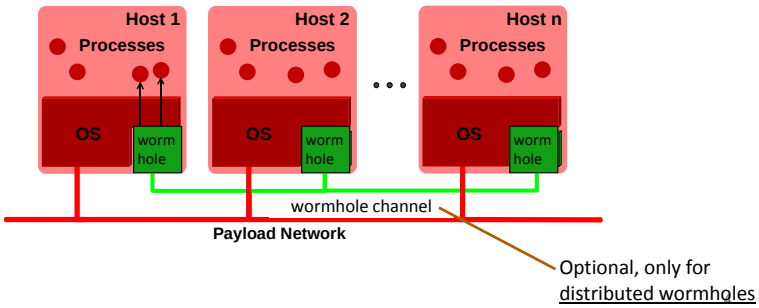
P. Verissimo. **Uncertainty and predictability: Can they be reconciled?** In Future Directions in Distributed Computing, volume 2584 of Lecture Notes in Computer Science, pages 108–113. Springer-Verlag, 2003

P. Verissimo. **Travelling through Wormholes: a new look at Distributed Systems Models**. ACM SIGACT News, vol. 37, no. 1, pages 66-81, 2006.

3

# Wormhole model / hybrid fault model

- Most of the system has weak guarantees
  - e.g., asynchronous, Byzantine faults
- Wormhole: a subsystem built to provide stronger properties (aka trusted component), e.g., partial synchronous, crash faults



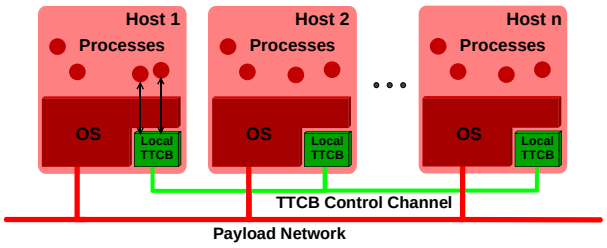
# Why hybrid system models?

- Expressive models w.r.t. reality
- Sound theoretical basis for proofs of correctness
- Naturally supported by hybrid architectures (like the wormholes architecture)
- Enablers of concepts for building totally new algorithms

5

# TTCB

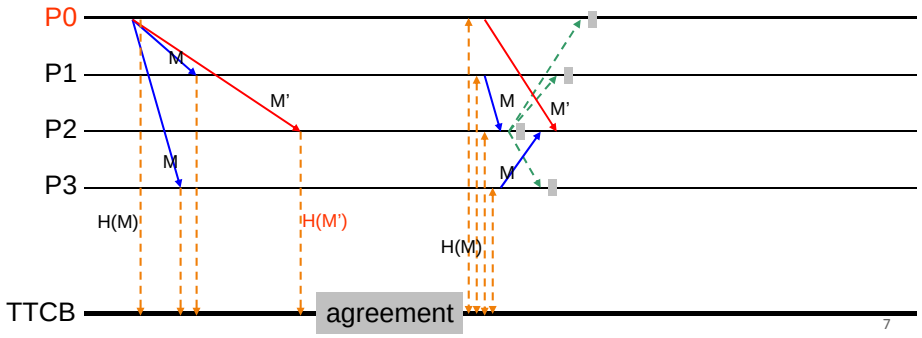
- TTCB – a wormhole to support the execution of *intrusion-tolerant algorithms/applications*
  - They run mostly in the payload system that can be attacked
  - They use the TTCB to execute some critical steps securely




6

# BRM – *2f+1 BFT reliable multicast*

- BRM = Byzantine-resilient Reliable Multicast
  - Based on the TTCB agreement service that runs inside the TTCB (crash faults, better synch)
  - The service tells which one is the correct hash




7

# 2004: BFT-TO AND TOW

M. Correia and N. F. Neves and P. Veríssimo. **How to Tolerate Half Less One Byzantine Nodes in Practical Distributed Systems**. *In Proceedings of the 23rd IEEE Symposium on Reliable Distributed Systems*. Florianopolis, Brasil, pages 174-183, October 2004.

M. Correia, N. F. Neves, P. Verissimo. **BFT-TO: Intrusion Tolerance with Less Replicas**. Computer Journal, Accepted for publication. (extended version of the previous paper)

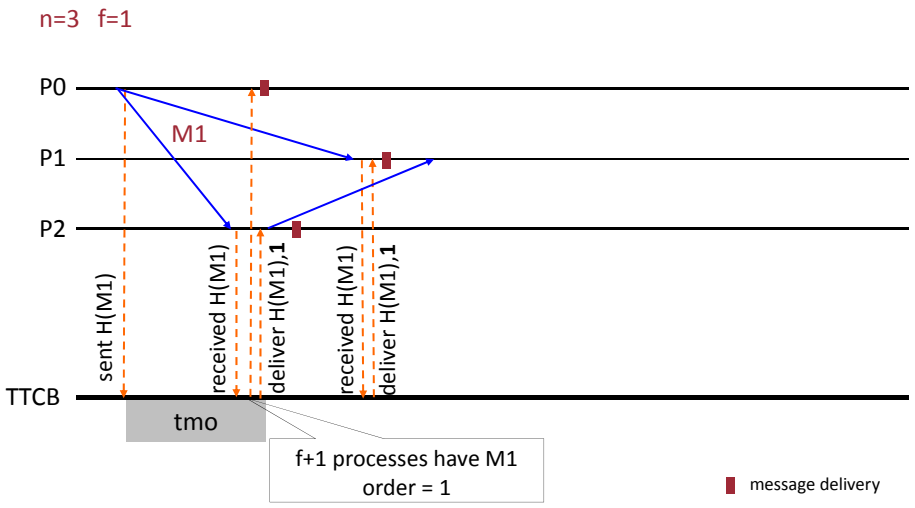
8

# BFT-TO – *2f+1 BFT SMR*

- Wormhole = TOW (Trusted Ordering Wormhole)
  - distributed like the TTCB, only in the servers (not clients)
- Basic algorithm:
  - Client sends request to one server, which sends to the rest
  - When getting the request, serves tell the TOW about it
  - TOW runs internally an agreement and tells servers the order in which they must run it
  - When a server processes the request, sends reply to client
  - Client picks the reply most voted

9

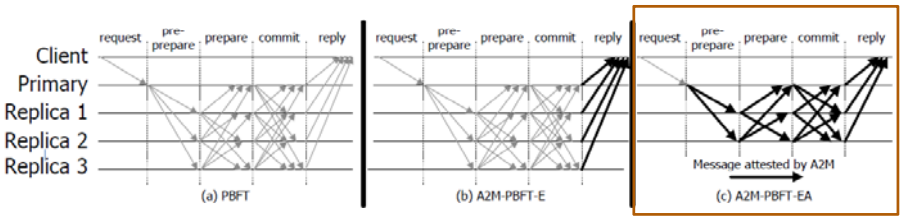
# BFT-TO execution




10

# A2M-PBFT-EA – *2f+1 BFT SMR*

- Chun et al. 2007
- Wormhole: A2M (Attested Append-only Memory)
  - equips a host with set of trusted, undeniable, ordered logs
  - interface with several ops: append, lookup, end, truncate, advance
  - local, not distributed (unlike the TTCB)
- A2M-PBFT-EA: first 2f+1 BFT SMR with a local wormhole




11

# 2008-…: MIN-BFT, EBAWA, USIG

Giuliana Santos Veronese, Miguel Correia, Alysson Neves Bessani, Lau Cheuk Lung. **Highly-Resilient Services for Critical Infrastructures**. *In Proceedings of the Workshop on Embedded Systems and Communications Security (ESCS)*. September 2009

G. S. Veronese, M. Correia, A. N. Bessani, L. C. Lung. **EBAWA: Efficient Byzantine Agreement for Wide-Area Networks**. In Proceedings of the 12th IEEE International High Assurance Systems Engineering Symposium (HASE). November 2010

G. S. Veronese, M. Correia, A. N. Bessani, L. C. Lung, P. Verissimo. **Efficient Byzantine Fault Tolerance**. IEEE Transactions on Computers, vol. 62, n. 1, pp. 16-30, Jan. 2013

12

# Simpler wormhole: USIG

- TOW is complex (distributed, agreement); A2M has complex API, memory grows
- USIG: local wormhole, one service, one call, simple
  - Single call: *createUI* (*m*) – assigns a unique ID to a message *m*
  - Includes only (monotonic) counter + signature function
- How does it help?
  - Faulty server can't send two messages with the same ID
  - Faulty server can't "go back" and use/reuse "old" IDs
  - …because the service won't return such IDs signed

13

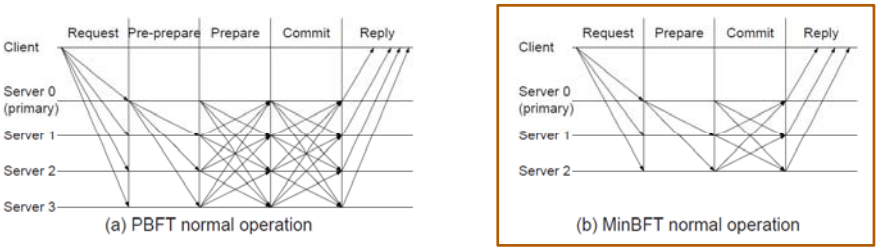
# USIG

- Optionally: counter + MAC function
  - faster
  - but verification must also be part of the wormhole (a 2nd call)
- Local service means it can be some hardware chip in server
  - We've implemented it on top of the Trusted Platform Module (TPM), "a commercial wormhole"
- Very similar to Trinc, developed in parallel (1st pub. 2009)

14

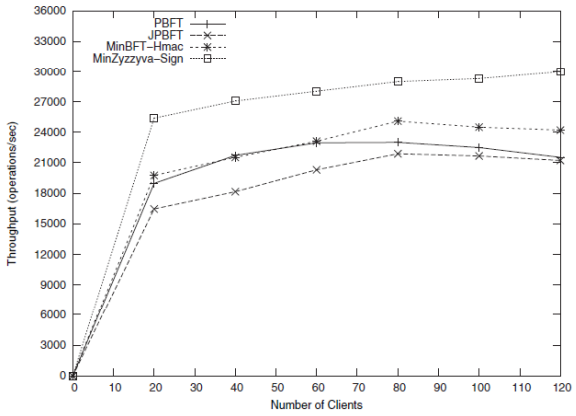
# MinBFT – *2f+1 BFT SMR*

- Wormhole: USIG
- Message pattern similar to Castro&Liskov's PBFT…
- …but less *f* replicas, 1 communication step less:




15

# MinBFT throughput (~2009)




- MinZyzzyva: a similar algorithm but based on Zyzzyva (speculative)

16

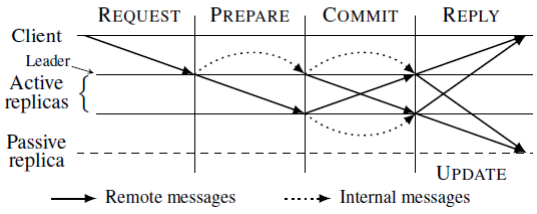
# EBAWA – *2f+1 BFT SMR* for WANs

- Wormhole: USIG
- Rotating primary: the primary only orders a batch of reqs
  - performance attacks / load balancing (we did it before in the Spinning alg.)
  - Merge operation provides liveness when the primary is faulty
- Asynchronous views:
  - a server starts an agreement as soon as it receives a client request by sending a prepare message
- Servers without pending client requests skip their turn
  - by sending a special message
- Measurements in LAN / PlanetLab / emulated WAN …
  - competitive in LANs, outperforms all in several WAN settings

17

# CheapBFT – *f+1 BFT SMR*

- Kapitza et al., 2012
- Wormhole: USIG
  - Implemented USIG in hardware (FPGA)
- CheapBFT
  - Runs CheapTiny with f+1 replicas in the normal case
  - Falls back to MinBFT




18

# 2010: 2F+1 CONSENSUS

Miguel Correia, Giuliana Santos Veronese, Lau Cheuk Lung, **Asynchronous Byzantine Consensus with 2f+1 Processes**, In Proceedings of the 25th Annual ACM Symposium on Applied Computing, March 2010.

19

## Byzantine Consensus with *2f+1* Processes

- Question: how to do BFT consensus with 2f+1 replicas? Who's the culprit behind 3f+1?
- Reliable multicast needs *3f+1* but if we use USIG (or TTCB or TOW or A2M), then *f+1* are enough
- We have shown that (f+1) reliable multicast is enough to solve *2f+1* consensus (with a few tricks more)…
- …by giving a methodology to transform CFT consensus algorithms into BFT consensus algorithms

20

# Transforming CFT->BFT consensus

Four steps:

1. reliable channels → *authenticated* reliable channels
2. broadcast → *reliable* broadcast
3. message reception → message reception *+validation*
4. Wait for messages from N-f processes →
   same thing + wait for either messages or *suspicions* of the other f processes (using special muteness failure detector)

21

# Transforming Mostefaoui/Raynal's CFT consensus algorithm

1. estimate ← proposal
2. **loop**
3. coordinator = round *mod* N
4. // ---------- phase 1 ----------
5. **if** coordinator **then reliable broadcast** message (phase1, estimate, round)
6. **wait until valid** phase1 message is received from the coordinator or the coordinator is suspected
7. **if** message received **then** estimate = estimate in message
8. // ---------- phase 2 ----------
9. **reliable broadcast** message (phase2, estimate, round)
10. **wait until valid** phase2 messages received from at least N-f processes and the rest (if any) are suspected
11. **if** same estimate in N-f messages **then broadcast** decision message and **decide**
12. **if** same estimate in N-2f messages **then** set estimate to that one
13. **endloop**

14. **upon valid** decision message received, **broadcast** decision msg. and **decide**

22

# Summary

- 2f+1 BFT SMR, 10+ years of research
- Based on a well-defined hybrid fault model
- Distributed vs local wormholes
- USIG: as simple as it can be?
- MinBFT: as simple/efficient as CFT SMR?

23