Intrusion Tolerance: The "killer app" for BFT Protocols (?)

Alysson Neves Bessani* Miguel Correia Paulo Verissimo
University of Lisbon, Faculty of Sciences – Portugal

An *intrusion-tolerant* (IT) system is one that maintains its security properties (i.e., *confidentiality*, *integrity* and *availability*) despite some of its components being compromised by an adversary [4]. The term was coined by Fraga and Powell in 1985 and was almost forgotten for 15 years due to the prohibitive performance costs of the mechanisms required to implement IT systems. Byzantine fault-tolerant (BFT) replication is perhaps the most notorious of such mechanisms. It is used to protect the integrity and availability of a system whose nodes are subject to being attacked and intruded by an intelligent adversary.

In 1999, Castro and Liskov showed that it is possible to build and deploy BFT replicated systems without too much performance overhead when compared with a non-replicated system [3]. After this, many others improved the performance of BFT protocols for specific scenarios, turning BFT into a hot-topic on the systems and dependability communities. Nonetheless, this apparent enthusiasm is not reflected in practice. BFT protocols are not being used in real systems.

Recently, some researchers have been exploring the applicability of BFT protocols to cloud-based services in order to make these resilient to certain non-fail-silent failures that can cause major disruptions (e.g., [5]). Clearly, BFT protocols can tolerate these component failures. But is it clear that BFT replication is the simpler way to achieve this goal? Knowing that BFT protocols are very complex and difficult to implement, is it reasonable to use them to tolerate some exceptional events that perhaps could be handled with simpler techniques such as error detection codes and majority voting?

One of the claims of this position paper is that, despite of the popularization of cloud-based systems, we do not see them as the *primary application* for BFT protocols. Instead, we argue that these protocols should first be used to improve the *security* of critical systems, i.e., to make them *intrusion-tolerant*. In particular, critical infrastructures (e.g., power plant distributed control networks and other SCADA/PCS systems), network infrastructure systems (e.g., name services, authentication systems, firewalls) and coordination services for open and dynamic systems (e.g., wireless ad-hoc and P2P networks). Our main point here is that cloud environments are believed to be secure, and there is no evidence that they need to pay for the complexity of BFT. On the opposite side, critical systems are deployed on extremely harsh environments and are a high-priority target for certain systematic malicious activities such cyber-terrorism and cyber-war.

Another claim of this paper is that BFT replication is only *one of the components* of an intrusion-tolerant system. *Proactive- and reactive-recovery* (i.e., to clean compromised replicas before or after they exhibit malicious behavior) [6], *confidentiality schemes* [1], *intrusion detection mechanisms*, *firewalls* and *diversity management tools* are all crucial components that can make an IT system resilient to a *malicious* and *intelligent* adversary.

The last claim of this paper is that to make BFT protocols hit the mainstream we should address three important concerns. First of all, we have to show that an IT system is more secure and dependable than a non-IT system. There are several works that show through dependability evaluation techniques that a crash-tolerant replicated system is more resilient than a non-replicated system. The question is, would it be possible to have some quantitative metric showing that an IT system (implementing BFT replication, for instance) is more secure and dependable than a "crash" fault-tolerant replicated system? Notice that this

*Contact author. Email: *bessani* AT *di.fc.ul.pt*.

problem is related with the problem of measuring the security of a system, which is an open problem and believed to be very difficult to address.

The second concern regards the complexity of the BFT protocols and prototypes. Today, 10 years after the PBFT paper of Castro and Liskov [3], we still don't have a *"stable implementation"* of a BFT protocol. Most of the complexity of the implementations is due to certain optimizations to make these protocols perform well in particular micro-benchmarks. Maybe, what we need is less-optimized protocols that provide good performance and that could be correctly implemented in a reasonable way.

Finally, we should consider what kinds of abstractions and system models are important for our target environments. Most BFT papers propose state machine replication (SMR) algorithms that can be easily used to model any deterministic service. However, the experience on crash-tolerant systems suggests that this paradigm is limited when one consider real systems [7, 2]. There are complex interactions in distributed systems that does not fit well in the SMR model. One option would be to start looking again to quorum protocols and build more flexible low-level abstractions. Another option would be to think about what kind of coordination services (like Chubby, Sinfonia, DepSpace and Zookeper) could be used in Byzantine environments [1].

The same criticism that we applied to the abstractions being currently addressed can also be applied to the system models usually considered. Are the ubiquitous f-out-of-n servers connected through fair links in a partial synchronous distributed system representative for most target environments for IT/BFT protocols? It is clear that this model is not suitable for dynamic systems, but is it OK for data center and wireless environments? Is a fixed and static f-out-of-n assumption reasonable against a malicious adversary? What about the well-known problem of fault independence?

These are the points we want to address in our talk. We plan to back our claims with insights gained from the almost 10-year involvement of the Navigators research group[1] with intrusion tolerance in EU-IST projects such as MAFTIA (Malicious- and Acidental-Fault Tolerance for Internet Applications) and CRUTIAL (CRitical UTility InfrastructurAL resilience). Our ultimate goal is to elicit questions, proposals, and foster general discussion about what are the relevant problems that should be addressed in the next years to make IT/BFT systems viable in practice.

References

[1] Alysson Neves Bessani, Eduardo Alchieri, Miguel Correia, and Joni Silva Fraga. DepSpace: a Byzantine fault-tolerant coordination service. In *Proceedings of the 3rd ACM European Systems Conference - EuroSys'08*, 2008.

[2] Mike Burrows. The chubby lock service. In *Proceedings of 7th Symposium on Operating Systems Design and Implementations - OSDI 2006*, November 2006.

[3] Miguel Castro and Barbara Liskov. Practical Byzantine fault tolerance. In *Proceedings of the 3rd Symposium on Operating Systems Design and Implementation*, pages 173–186. USENIX Association, February 1999.

[4] Joni Fraga and David Powell. A fault- and intrusion-tolerant file system. In *Proceedings of the 3rd International Conference on Computer Security*, pages 203–218, 1985.

[5] Atul Singh, Pedro Fonseca, Petr Kuznetsov, Rodrigo Rodrigues, and Petros Maniatis. Zeno: Eventually consistent Byzantine fault tolerance. In *Proc. of 6th Symposium on Networked Systems Design and Implementation - NSDI 2009*, April 2009.

[6] Paulo Sousa, Alysson Neves Bessani, Miguel Correia, Nuno Ferreira Neves, and Paulo Verissimo. Highly available intrusion-tolerant services with proactive-reactive recovery. *IEEE Transactions on Parallel and Distributed Systems*, 2009. to appear.

[7] Werner Vogels. Life is not a state-machine. Invited talk on ACM PODC'06. More info on `http://www.allthingsdistributed.com/2006/08/life_is_not_a_statemachine.html`, 2006.

[1] `http://www.navigators.di.fc.ul.pt`