DAHLIA MALKHI

Lead Researcher at Calibra

http://dahliamalkhi.wordpress.com
@ **dmalkhi@fb.com**

SHORT BIO

An applied and foundational researcher in broad aspects of distributed systems technology. Presently, research lead at Calibra, advancing the Libra technology [1] and co-inventor of HotStuff [2]. Co-founder and technical lead of VMware blockchain [3]. Co-inventor of Flexible Paxos [4], the technology behind Log Device [5]. Creator and tech lead of CorfuDB [6], a database-less database driving VMware's NSX-T [7] distributed control plane. Co-inventor of the FairPlay project [8].

Joined Calibra in June 2019 as a research lead. In 2014, after the closing of the Microsoft Research Silicon Valley lab, co-founded VMware Research and became a Principal Researcher at VMware until Jun 2019. From 2004-2014, a principal researcher at Microsoft Research, Silicon Valley. From 1999-2007, a tenured associate professor at the Hebrew University of Jerusalem. From 1995-1999, a senior researcher at AT&T Labs, NJ.

TECHNOLOGY IMPACT

For over two decades, my work has been straddling by choice between foundational and applied research. I published over 150 papers; recent ones are listed on my homepage, DBLP keeps track of the rest.

I was fortunate to bring several scientific results into fruition within leading industrial platforms. Below, I tell the stories of four technologies I participated in creating.

CorfuDB, Initiator and Technical Lead

2012- Microsoft,VMware

In 2012, Phil Bernstein [9] approached me at Microsoft Research with the following observation. RAM has grown cheap/large enough to hold a complete database index in memory. Therefore, one can build a fully replicated transaction processing engine by storing a database index completely in-memory, persisting index modifications to a shared commit-log. His team prototyped an in-memory index called Hyder. The key enabler for this vision would be a reliable, high throughput distributed log, which Phil wanted to stripe across an array of SSDs. Unfortunately (yet fotunate for me), the initial design of his distributed commit-log was flawed. While fixing the design, I extracted a foundational insight that motivated me to establish and lead the CorfuDB project [10].

CorfuDB [11] is a database-less database built around a global, reliable, high-throughput distributed commit-log. The CorfuDB log serves as the source of ground truth around which one

builds distributed control-planes for large clusters. The key paradigm underlying CorfuDB is the reliable log that operates at high throughput. This was the foundational insight I have taken from Hyder. I built the first CorfuDB PoC at Microsoft with OS license, and later drove it at VMware to production. At VMware, CorfuDB serves as the a distributed control-plane for NSX-T [12], a leading SDN product that has market volume of over $1B. At Facebook, CorfuDB was re-engineered in Delos [13], a control plane underlying a dynamic cluster storage backend system.

You might wonder what happened to Phil's in-memory fully replicated DB. Several years later, it became the backbone of the SQL Azure cloud database.

Flexible Paxos, Co-Inventor

2016- VMware

In the summer of 2016, I hosted a research intern named Heidi Howard from Cambridge, UK. I told her about the CorfuDB protocol and encouraged her to think about the performance benefit of separating the sequencer role from the rest of the system. The result has been a stunning revelation we named Flexible Paxos [14]:

Each of the phases of Paxos may use non-intersecting quorums. Only quorums from different phases are required to intersect. Majority quorums are not necessary as intersection is required only across phases.

Everyone in the field of distributed systems knows that quorums in Paxos must intersect, so what gives? What Heidi observed is that Paxos, which lies at the foundation of many production systems, is conservative. Within each of the phases of Paxos, it is safe to use disjoint quorums and majority quorums are not necessary. Since the second phase of Paxos (replication) is far more common than the first phase (leader election), we can use Flexible Paxos to reduce the size of commonly used second phase quorums. By no longer requiring replication quorums to intersect, we have removed an important limit on scalability. Through smart quorum construction and pragmatic system design, we enabled a new breed of scalable, resilient and performant consensus algorithms. The algorithmic core of a production scale-out messaging bus at Facebook called LogDevice [15] is based on it, as is the more flexible paxos [16] of YouTube's distributed MySQL backbone.

HotStuff, Co-Inventor and Technical Lead

2017- VMware,Calibra

Renewed interest in the Blockchain world on scaling and robustifying the long standing problem of asynchronous Byzantine Fault Tolerant (BFT) Consensus.

In 2016 when designing the blockchain infrastructure at VMwares blockchain project, we observed that all BFT solutions contain quadratic voting steps. Why is this so bad? When Byzantine consensus protocols were originally conceived, a typical target system size was n=4 or n=7, tolerating one or two faults. But scaling BFT consensus to n=2000 means that even on a "good day" when communication is timely and a handful of failures occurs, quadratic steps require 4,000,000 messages. A cascade of failures might bring the communication complexity to whopping 8,000,000,000 transmissions for a single consensus decision. No matter how good the engineering and how we tweak and batch the system, these theoretical measures are a roadblock for scalability.

Around that time, tremendous innovation was occurring outside academic circles by blockchain startups. Two of these caught our attention, Tendermint and Casper. These protocols dramatically simplified the view change mechanism by introducing a synchronous delay when a leader starts. I observed that by adding one more phase to Tendermint, we can maintain the advantage of simplicity while avoiding the delay it introduced. The result is HotStuff [17], named after a cartoon character in the same family of Casper, the first responsive BFT solution with a linear view-change.

Beyond improving communication complexity, HotStuff embodies a minimalist algorithmic framework that bridges between classical BFT solutions and the blockchain world; the entire protocol is captured in less than half a page of pseudo-code. HotStuff became popular in the blockchain developer community not only due to linearity, but (and perhaps mostly) due to its simplicity and developer-friendly design. Calibra adopted it to drive the blockchain infrastructure of Libra, as did (that we know of) Thunder, Celo, and Cypherium.

Fairplay, Co-Inventor

2004 Hebrew University of Jerusalem

In 2004, Noam Nisan and I asked ourselves whether cryptographic primitives which were considered completely impractical are actually becoming practical. With my PhD student Yaron Sella, we implemented the MPC protocol, while Noam supervised his grad-students to implement a language that compiles into a binary circuit. The first fully implemented Fairplay MPC platform [18] was alive shortly after. By 2008, the the millionaires problem, mini auctions, and other problems, could be solved over an interconnect in seconds. Since then, the Fairplay source code has been downloaded by hundreds of academic groups, and has sparked in the past decade a wave of crypto-engineering projects which bring crypto theory into practice, including heavy crypto methods like oblivious RAM, ZK proofs and PCP.

Notes

1. https://libra.org/en-US/.
2. https://dahliamalkhi.wordpress.com/2018/10/24/hotstuff-three-chain-rules/.
3. https://research.vmware.com/projects/vmware-enterprise-blockchain.
4. https://dahliamalkhi.wordpress.com/2016/08/26/flexible-paxos.
5. https://logdevice.io/docs/Consensus.html.
6. https://github.com/CorfuDB/CorfuDB.
7. https://shuttletitan.com/nsx-t/nsx-t-management-cluster-benefits-roles-ccp-sharding-and-failure-handling/.
8. https://www.cs.huji.ac.il/project/Fairplay/.
9. https://en.wikipedia.org/wiki/Phil_Bernstein.
10. https://github.com/CorfuDB/CorfuDB.
11. "CORFU: A Distributed Shared Log". Mahesh Balakrishnan, Dahlia Malkhi, John Davis, Vijay Prabhakaran, Michael Wei, Teb Wobber. ACM Transactions on Computer Systems 2013.

12. https://shuttletitan.com/nsx-t/nsx-t-management-cluster-benefits-roles-ccp-sharding-and-failure-handling/.

13. https://engineering.fb.com/data-center-engineering/delos/.

14. "Flexible Paxos: Quorum Intersection Revisited." Heidi Howard, Dahlia Malkhi, Alexander Spiegelman. Conference On Principles Of Distributed Systems (OPODIS) 2016.

15. https://logdevice.io/docs/Consensus.html.

16. http://ssougou.blogspot.com/2016/08/a-more-flexible-paxos.html.

17. "HotStuff: BFT Consensus with Linearity and Responsiveness". Maofan Yin, Dahlia Malkhi, Michael K. Reiter, Guy Golan Gueta, Ittai Abraham. ACM Principles of Distributed Computing (PODC) 2019.

18. "Fairplay–A Secure Two-party Computation System". Dahlia Malkhi, Noam Nisan, Benny Pinkas and Yaron Sella. In proceedings of the 13th Conference on USENIX Security Symposium, 2004.