Research Statement

Background. Computing systems evolve fast to satisfy increasingly complicated tasks and create massive fortunes. In the meantime, providing safety, resiliency, and efficiency in the face of malicious attackers draws more and more attention. Distributed systems are the fundamental building blocks for modern computing platforms, where data and services are replicated on multiple servers to provide accessibility and scalability. However, they are naturally more complex and harder to reason manually compared to standalone systems. Formal methods use rigorous mathematical models to help design, verify, and implement correct-by-construction computing systems. By concisely defining the specifications and mathematic models, formal methods enable provable correctness and automated verification for systems. I summarize my projects:

Distributed system synthesis. Distributed systems and replicated objects are pervasively used for fault tolerance, availability, responsiveness, and scalability. However, maintaining the correctness of the whole system while providing efficiency is challenging with the enlarged collection of consistency notions and coordination protocols. On the one hand, strong consistency provides desired safety properties that ensure replicated object states converge, stay correct, and are up to date. However, keeping a total order for all the transactions lacks availability under the network partition. On the other hand, weaker notions of consistency exhibit different extents of responsiveness and scalability at the cost of potential integrity loss. Further, figuring out the subtle differences in safety guarantees among the large collections of weaker consistency notions is unintuitive to system designers.

In [1], I revive the three pillars of consistency: integrity, convergence, and recency, while minimizing coordination. Recency is a useful yet neglected property in the literature which ensures the return value of methods maintains a bounded difference with the latest value. In fact, lots of distributed services can benefit from fairly recent data such as ticketing, distributed sensors, and network accounting. By offering a relational language and capturing object specifications with denotational semantics, I formally model object methods with a complete set of relational operators and specify user's integrity and recency requirements. I defined a set of sufficient conditions that are checked both statically and dynamically in the SMT solver and a novel operational semantics of replicated objects that provably guarantees all three properties. Interestingly, recency-awareness not only satisfy user's staleness requirements but also reduces the coordinator needed for integrity by buffering calls. Finally, I develop the first synthesis tool that automatically infers the optimal recency bounds and instantiates the run-time systems with our novel coordination protocols.

Resilient and safe partitioning and replication. The system above only considers crash failures. However, distributed systems in practice are facing Byzantine failures, which allow a limited number of servers to perform arbitrary operations. With the rise of inter-organization corporations in healthcare, finance, and military, multiple parties with different trust assumptions need to coordinate for a common goal. The heterogeneity of trust assumptions (confidentiality policies) leads to the distribution of data and computation across administrative boundaries, which prohibit simple full replication of the whole system. Moreover, the various integrity and availability policies from different parties result in different levels of replication, which makes system designs error-prone and hard to enforce system-wide policies. Research about end-to-end security and Byzantine fault

tolerance systems have been two separate paths: Information flow control can guarantee that sensitive data does not leak and computed results are correct. But there is no practical abstraction for availability; Byzantine quorum replication protocols can enforce availability in the face of Byzantine failures. However, they are monolithic systems with uniform trust assumptions. Therefore, I address the research question of how to enjoy the guarantees of all three end-to-end trustworthiness in a resource-restraint environment. I propose a security-typed object-based language to formally model users' objects and security policies; a partitioning transformation to split methods; operational semantics to model the distributed execution in run time; and an information flow type inference system for partitioned and replicated classes. [2] provably guarantees the non-interference of confidentiality, integrity, and availability of well-typed objects. Given a class and security policies from users, our synthesis tool automatically infers the placement of object fields and methods on Byzantine quorum systems with as few machines as possible.

Heterogeneity in blockchains. The recent advancement in blockchain technology attracted tremendous interest from both industry and academia. The nature of blockchains is consensus protocols, which can be summarized into two general categories: proof-of-resource-based and Byzantine-quorum-systems-based protocols. Bitcoin and other resource-based blockchains are naturally open and decentralized. However, they suffer from high energy consumption, low throughput, and probabilistic liveness guarantees. Quorum-based protocols have moderate energy consumption and high throughput. However, their quorums are uniform and closed, which prevents individuals from expressing their own trust assumptions freely and open membership.

To unleash the potential of quorum-based consensus protocols, I present a general model of heterogeneous quorum systems (HQS), where each participant can declare its own quorums and capture their properties. It has been proved that two properties: quorum intersection and availability, are necessary conditions for consensus abstraction. Interestingly, I prove that they are not sufficient in the heterogeneous setting. I also define the notion of quorum inclusion, and show that the three conditions together are sufficient: I present reliable broadcast and consensus protocols with correctness proof.

In order to support open membership, I present reconfiguration protocols for HQS and prove their correctness: joining and leaving a process, and adding and removing a quorum. I present trade-offs for the properties that reconfigurations can preserve. Further, I present a graph characterization of heterogeneous quorum systems and its application for reconfiguration optimization.

Border impact. Cloud computing is one of the platforms that can benefit from the same rigorous analysis, where data and services may be replicated and updated inconsistently without careful design. When facing more than ever complex systems, it is not feasible for engineers to understand all the details of the different technologies. Program synthesis provides a way to bridge the high-level specifications and low-level implementation and guarantees. Further, formal methods help build correct-by-construction systems that complement software testing.

Another essential aspect of my research is exploring new challenges with the inevitable trends of heterogeneity and diversity in both society and technology. With ever-growing computation power and resources at hand, it is possible to discover systems where individuals' needs are respected while we collaborate efficiently for a common goal. Potentially all people with specific computation needs can benefit from this fundamental research topic.

[1] Li, Xiao, Farzin Houshmand, and Mohsen Lesani. "Hampa: Solver-aided recency-aware replication." *Computer Aided Verification: 32nd International Conference, CAV 2020, Los Angeles, CA, USA, July 21–24, 2020, Proceedings, Part I*. Cham: Springer International Publishing, 2020.
[2] Li, Xiao, Farzin Houshmand, and Mohsen Lesani. "HAMRAZ: Resilient Partitioning and Replication." *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022.