Research Statement

Chenyuan Wu

For the past decade, data management systems have been governed by central authorities, typically assuming trusted environments and tolerating only crash failures. However, events like the recent collapse of multinational banks [1], the unexpected shutting down of user accounts [2], and the glitch of large stock exchange [3] have deteriorated trust in such monopolized systems. This trend has facilitated the rapid rise of decentralized systems where multiple untrusting parties collaborate on data management, making untrusted environments prevalent and shifting the focus of fault tolerance to tolerating malicious failures.

My research centers around untrusted environments, with the long-term goal of **building fully adaptive and flexible distributed systems that can be deployed in any untrusted contexts**. In pursuit of this goal, my dissertation research explores the intersection of distributed systems and machine learning. Specifically, I focus on blockchains as the most notable instance of untrustworthy distributed systems, and have pioneered AI-driven adaptive blockchains. As summarized in Figure 1, my dissertation research adopts a top-down layered approach across the blockchain systems stack, from transaction management [4], Byzantine fault-tolerant consensus protocols [5, 6, 7, 8, 9] to their underlying infrastructure [10] and cross-layer adaptation [11]. Overall, I have published papers in top database [4, 8, 10, 11, 12], distributed systems [5, 6, 7], and networking venues [13], winning Best Paper Awards at NSDI 2024 and EDBT 2020.

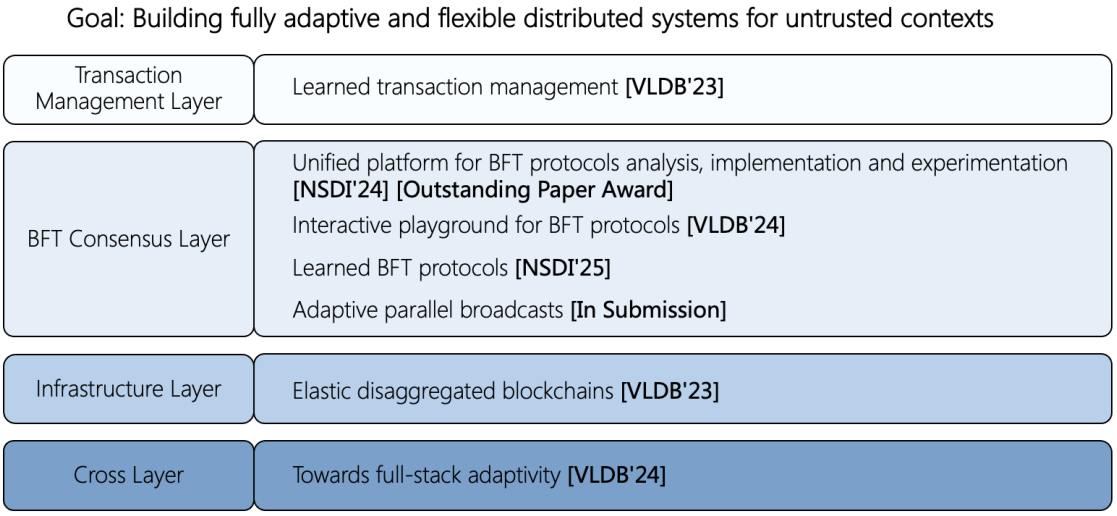

Figure 1: Summary of my dissertation research.

Achieving adaptivity and flexibility is crucial yet challenging. Recently, a broad range of decentralized applications has flourished on top of blockchains, encompassing real world asset tokenization, gaming, supply chain assurance, crowdworking, central bank digital currencies, and decentralized AI inference. At the same time, a proliferation of blockchain systems have been proposed. This rapid growth presents challenges at every layer of the system stack:

Challenge 1: No one-size-fits-all transaction management paradigm. Blockchain systems present different transaction management paradigms, including the sequence in which ordering, execution, and validation are performed, the number of transactions in a block, stream processing, and the use of reordering and early aborts, etc. Our measurement study [4] has shown that different transaction management paradigms are optimal for different workloads. Unfortunately, current users have to choose a fixed paradigm prior to application development. Choosing the right configuration is not easy even for experienced users, due to the vast search space of paradigms and their internal parameters. Further, client requests fluctuate with different patterns throughout the day, and workloads may permanently shift as business evolves. These dynamic factors are especially pronounced in decentralized applications, resulting in poor performance of the chosen system, since no single configuration provides dominant performance.

In an effort to address this challenge, I built AdaChain [4], the first learned system that adaptively manages blockchain transactions under dynamic workloads.

Challenge 2: No one-size-fits-all consensus protocol. Each transaction management paradigm can be further customized with a broad spectrum of Byzantine fault-tolerant (BFT) consensus protocols. Over the past decade, the BFT protocol landscape has surged from a handful to over a hundred protocols, each embodying different design choices, including commitment strategy, number of communication phases, and leader replacement mechanism, etc. Selecting the right protocol and properly configuring its parameters from this vast search space is critical for end-to-end system performance. However, users are continually challenged by dynamic client workloads, failures, and network conditions, as no single protocol consistently performs well in all scenarios. To make matters worse, migrating between these siloed protocols midway — each with its own code base and distinct interfaces — is difficult, if not impossible.

I have been pioneering the research on a unified consensus engine. My work on Bedrock [7], a unified platform for BFT protocols analysis, implementation, and experimentation was awarded Best Paper at NSDI 2024. Building on this foundation, I developed BFTBrain [5], the first reinforcement learning based adaptive consensus engine, as well as BFTGym, an interactive playground for BFT protocols [8]. Additionally, I also worked with industry to propose a novel adaptive parallel broadcast regime [9].

Challenge 3: No one-size-fits-all resource provisioning. With machine learning increasingly integrated on-chain, decentralized applications might perform memory-intensive operations, compute-intensive operations, or a mix of both at different times and execution stages. Although Blockchains-as-a-Service (BaaS) offers a convenient way for users to manage their resources, the current rigid BaaS infrastructure poorly supports such heterogeneous resource demands. First, to ensure high-throughput services over various workloads, servers are usually over-provisioned, leading to low utilization of certain or all types of resources. Second, as workloads evolve, it is hard to scale up/down resources independently and seamlessly, e.g., expanding/shrinking memory resources independent of CPUs without pausing transactions. Such limitations require a radical rethinking of how resources are arranged for BaaS infrastructure.

To tackle this challenge, I built FlexChain [10], the first disaggregated blockchain infrastructure with efficient resource utilization and elasticity.

Summary. Overall, I believe machine learning and resource disaggregation techniques provide solutions to these challenges, enabling us to build the next generation of untrustworthy distributed systems — fully adaptive to changes in workload, hardware, or attacks, flexible in switching protocol implementations and resource provisioning, and capable of operating without human intervention. In my experience, developing such systems requires a deep understanding of both machine learning and the systems stack.

Transaction Management Layer

The main stages of processing transactions in fault-tolerant systems are sequencing and execution. Building on top of a sequencer abstraction, the transaction management layer decides how sequenced transactions are executed, and updates the world state and ledger upon commitment. My work has pioneered adaptive transaction management.

Learned transaction management. AdaChain [4] is a reinforcement learning-based blockchain framework that chooses the best transaction management paradigm and sets appropriate parameters in order to maximize effective throughput for dynamic transaction workloads.

AdaChain relies on two key innovations. First, it models the selection of a paradigm as a contextual multi-armed bandit problem. This formulation allows AdaChain to apply classical algorithms, such as Thompson sampling, to select paradigms in a way that minimizes *regret* (the difference between the performance of the chosen paradigm and the optimal paradigm). AdaChain will strategically test different architectures to learn which ones are well-suited to the current workload.

Second, AdaChain introduces protocols to switch from one paradigm to another in a live system, while maintaining strong serializability properties. This switching protocol is not only required for AdaChain to function (multi-armed bandits generally require making multiple decisions before the optimal is reached), but also enables a new class of blockchains that can seamlessly transition between different transaction management paradigms to support the shifting workloads in the real-world. Intuitively, it works by splitting switching decisions between two paths. In the normal path, all nodes agree to switch to the same new paradigm after a certain number of blocks have been committed, while in the slow path, all nodes switch to the same paradigm after failing to make progress on processing transactions for a certain amount of time.

Experimentally, we show that AdaChain is not only able to select optimal or near-optimal configurations for a wide variety of dynamic workloads, but its reinforcement learning approach also allows it to quickly adapt to new hardware, new storage subsystems, and new unanticipated workload changes on the fly.

BFT Consensus Layer

The BFT consensus layer handles the sequencing, i.e., assigning a sequence to each transaction and ensuring such sequence is correctly replicated on all replicas in the presence of malicious failures. My research team and I have been pioneering the work on a unified and learned consensus engine.

Unified platform for BFT protocols analysis, implementation, and experimentation. BFT protocols are different along several dimensions, including the number of replicas, processing strategy (i.e., optimistic, pessimistic, or robust), supporting load balancing, etc. While dependencies and trade-offs among these dimensions lead to several design choices, there is currently no unifying tool that provides the foundations for studying and analyzing BFT protocols' design dimensions and their trade-offs. We envision that such a unifying foundation will provide an in-depth understanding of existing BFT protocols, highlight the trade-offs among dimensions, and facilitate protocol designers to invent new protocols that fit their needs. This requires us to extract essential elements of the agreement and define atoms that connect these elements.

To address this challenge, we developed Bedrock [7] that can be used to analyze and navigate the evergrowing BFT landscape. Bedrock defines a set of design choices, where each design choice expresses a *one-to-one function* to map plausible input points (i.e., a BFT protocol) to plausible output points (i.e., another BFT protocol) in the design space. In addition to protocol analysis and discovery, Bedrock provides a domain specific language for rapidly prototyping BFT protocols, as well as a unified implementation environment for different BFT protocols proposed in diverse settings and contexts, laying the foundation for protocol switching. Designing such a platform is the first step towards developing an adaptive BFT multi-protocol engine.

Interactive playground for BFT protocols. Building upon Bedrock, we developed BFTGym [8], a platform designed to further simplify and enhance the *experimentation* of BFT protocols. While Bedrock provides the analytical and implementation backbone, BFTGym focuses on minimizing the experimental burden and maximizing interactivity and user engagement. BFTGym features an intuitive and *interactive* user interface, which accepts users' specifications for client workloads, system settings and fault scenarios. At the same time, the interface provides users with instant feedback and real-time performance visualizations.

Learned BFT protocols. Building upon Bedrock, we also developed BFTBrain [5], a reinforcement learning (RL) based BFT consensus engine that provides significant operational benefits: a plug-and-play system suitable for a broad set of hardware and network configurations, and adjusts effectively in real-time to changing fault scenarios and workloads. BFTBrain adapts to system conditions and application needs by switching between a set of leader-based BFT protocols in real-time.

To leverage RL, BFTBrain measures in real-time performance metrics obtained locally by replicas. Beyond standard metrics like commit throughput, BFTBrain employs fine-grained metrics that offer deeper performance insights, such as the ratio of requests that are committed in the fast path of dual-path protocols, the number of received messages per slot, and the interval between consecutive leader proposals. These new metrics are measured in a distributed manner and serve as features for BFTBrain's RL engine. By modeling the selection of a BFT protocol as a contextual multi-armed bandit problem, the RL engine strategically tests different protocols at run-time to learn which ones fit the current system conditions. BFTBrain coordinates RL in a decentralized manner, where nodes share local features/rewards by consensus and reach the same learning output, achieving resilience to adversarial data pollution.

Our extensive evaluation shows, in addition to providing significant operational benefits, BFTBrain improves throughput over fixed protocols by 18% to 119% under dynamic conditions and outperforms state-of-the-art adaptive systems by 44% to 154%.

Adaptive parallel broadcasts. BFTBrain focuses on *leader-based* protocols, but recent advances in *leader-less* protocols that allow parallel broadcasts have greatly improved protocol throughput. We studied leader-less protocols and introduced the concept of "ticketing", through which atomic broadcasts are orchestrated by distributed nodes. Our study showed that managed ticketing outperforms unmanaged ticketing in heterogeneous, dynamic systems, as it prevents slow nodes from hampering overall progress. We then designed and implemented a hybrid scheme which combines both managed and unmanaged ticketing, striking a balance between adaptivity and resilience [9].

Infrastructure Layer

Optimizing software layers alone without consideration of the underlying hardware infrastructure leads to a performance ceiling and resource under-utilization. The final component of my dissertation thus focuses on the infrastructure of untrustworthy distributed systems.

Elastic disaggregated blockchains. FlexChain [10] is a novel disaggregated permissioned blockchain service that

leverages recent innovations in disaggregated data center (DDCs). FlexChain's design is based on a few key insights. We observe that the Execute-Order-Validate (XOV) architecture used in Hyperledger Fabric and several other systems lends itself naturally to disaggregation. First, the execution and most of the validation stages are inherently parallelizable, leading to a design where we can auto-scale these stages to arbitrary numbers of CPU cores in the compute pool. Second, most of the blockchain state resides in key-value stores, which can be hosted in the memory pool. This design reduces the need of storing and retrieving data to and from the disk because there is sufficient memory for buffering the working sets of transactions.

Hence, FlexChain designs a tiered key-value store and leverages auto-scaling to achieve both memory and compute elasticity. Remote memory connected by a high-speed network is used as a cache for data stored on disk. Read/write and buffer eviction protocols all need careful designs to ensure correctness. As smart contract executions are amenable to parallelization in a DDC, sequential conflict check and world state update in the validation stage of XOV would become the bottleneck. We further adopted a concurrency control mechanism to parallelize the validation stage.

We experimentally evaluated both the operational and performance benefits. Our comparison between FlexChain and traditional distributed architectures demonstrated that disaggregating complex distributed systems like blockchains is not only feasible (achieving comparable or even better performance), but also preferable (achieving independent elasticity for hosting diverse workloads and better resource utilization).

Towards full-stack adaptivity. We present FAB [11], a learning-based adaptive framework for untrustworthy distributed databases, emphasizing full-stack adaptivity. FAB outlines the remaining research challenges for each layer in the stack, motivates the need for cross-layer adaptation, and presents our initial solutions.

Other Work

In addition to the systems that constitute my dissertation, I have also worked on several exciting projects that focus on other aspects of distributed systems and data management. I collaborated on NetSpec [13], **a tool that synthesizes network specifications from input-output examples**. NetSpec aims to accelerate the adoption of formal verification in networking practice, by reducing the effort and expertise required to specify network models or properties. I also contributed to P3 [12], **a novel provenance model and system for analyzing probabilistic logic programming programs**. P3 enables four types of provenance queries: traditional explanation queries, queries for finding the set of most important derivations within an approximate error, top-K most influential queries, and modification queries that enable us to modify tuple probabilities with fewest modifications to program or input data. **P3 received the Best Paper Award at EDBT 2020.**

Future Directions

My future research directions center around my long-term goal — building fully adaptive and flexible distributed systems that can be deployed in any untrusted contexts.

AI-driven distributed systems. My immediate next step focuses on building autonomous distributed systems through machine learning, with two main areas of interests. First, current autonomous consensus protocols have large room for improvements through finer-grained control and learning. For instance, while BFTBrain [5] focuses on adaptation at the protocol level, dissecting the performance impact of and learning protocol-specific parameters (e.g., various timers, block size, selection of leader or commit aggregator) would be highly valuable. This would require designing more efficient reinforcement learning algorithms given the significantly larger action space. Additionally, incorporating workload forecasting into the learning agent is crucial, as it could reduce protocol switching overhead by adapting epoch length and improve resilience to Byzantine workloads. Moreover, different reward functions could be explored. Optimizing fairness via machine learning is an interesting topic for large scale decentralized systems.

Second, I plan to extend the AI-driven learning approach to a broader class of distributed protocols. As today's distributed protocols demonstrate increasingly complex design and engineering, manually controlling such protocols through hand-crafted heuristics is becoming infeasible. It is appealing to build "learned" data dissemination protocols (e.g., mempool or gossip), distributed hash tables, sharding protocols, and parallel broadcast regimes.

Adversarial machine learning. The use of machine learning introduces new attack vectors for distributed systems, such as the manipulation of locally measured features and reward data. In the medium term, I plan to study the impact of adversarial machine learning on learned distributed systems, especially Byzantine fault-tolerant systems. For example, if certain Byzantine nodes launch projected gradient descent (PGD) attacks on their local features, but the protocol applies a median filter after aggregating these features, how would that PGD attack affect throughput of

the system? I will also explore the trade-off between defense against such attacks and maintaining systems adaptivity. As potential ways of defense against adversarial attacks, techniques such as randomized smoothing could be explored. This is also a topic that I am excited to collaborate on with other machine learning experts.

Revisiting system models. In the medium term, I will also revisit different system models. Due to the boom of blockchains, over the past five years, systems under the Byzantine fault-tolerant (BFT) model have seen more significant advancements than those under the benign crash fault-tolerant (CFT) model. Unfortunately, the recent CFT and BFT worlds are quite isolated, with no performance comparisons between state-of-the-art protocols in both domains. Now is the time to look back and bridge this gap. It is promising to study how the recent BFT engineering and algorithmic innovations such as routine leader rotation, proposal pipelining, and directed acyclic graph-based mempools could inform the design and implementation of modern CFT protocols.

Additionally, it is meaningful to reconsider the system model adopted by blockchain itself. Although solving trust issues and providing incentives, blockchains encode each single step of the computation as a transaction and interpret all transactions on the chain. Such model is struggling to even enable traditional applications from cloud computing, far from fulfilling the futuristic picture of "Web 3". I am interested in designing a more foundational and *expressive* computation model, where many decentralized micro-services could be loosely organized without consulting the ledger, and the ledger (a.k.a. consensus) itself could be a useful micro-service in the model.

New infrastructures. New digital infrastructures are emerging rapidly, such as 6G networks, disaggregated data centers, and large language model (LLM) systems. It is crucial to research how distributed systems techniques can *leverage* these new infrastructures (e.g., designing resource-efficient consensus protocols on top of RDMA and disaggregated data centers), as well as how they can *contribute* to these new infrastructures (e.g., decentralizing and democratizing LLM inference via geo-distributed GPUs). Moreover, the speed at which people invent distributed protocols should keep pace with the rapid emergence of these infrastructures. In the long term, I plan to design a generic fault tolerance layer that can be easily customized and reused across different infrastructures. This layer would present a declarative interface, where users only need to specify the properties of the underlying infrastructure, the adversarial factors they are concerned about, and the guarantees they seek from the layer. From there, a suitable protocol could be automatically invented and its implementation compiled.

In general, I am interested in distributed data management, machine learning for systems, and blockchains. In the course of my Ph.D. study, I am fortunate to have closely collaborated with five professors, five researchers in industry, and four students across multiple institutes on a variety of topics. I am looking forward to embarking on exciting adventures with new colleagues and students.

References

[1] C. Thorbecke, "Silicon valley bank collapse sends tech startups scrambling." [Online]. Available: https://www.cnn.com/2023/03/10/tech/silicon-valley-bank-tech-panic/index.html

[2] T. S. Bernard and R. Lieber, "Banks are closing customer accounts, with little explanation." [Online]. Available: https://www.nytimes.com/2023/04/08/your-money/bank-account-suspicious-activity.html

[3] N. Randewich and A. K. Basil, "Nyse glitch sparks volatility in dozens of stocks." [Online]. Available: https://www.reuters.com/markets/us/nyse-equities-investigating-reported-technical-issue-2024-06-03/

[4] C. Wu, B. Mehta, M. J. Amiri, R. Marcus, and B. T. Loo, "Adachain: A learned adaptive blockchain," in *VLDB 2023*.

[5] C. Wu, H. Qin, M. J. Amiri, B. T. Loo, D. Malkhi, and R. Marcus, "Bftbrain: Adaptive bft consensus with reinforcement learning," in *NSDI 2025*.

[6] C. Wu, H. Qin, M. Javad Amiri, B. Thau Loo, D. Malkhi, and R. Marcus, "Towards truly adaptive byzantine fault-tolerant consensus," in *SIGOPS OSR 2024*.

[7] M. J. Amiri, C. Wu, D. Agrawal, A. El Abbadi, B. T. Loo, and M. Sadoghi, "The bedrock of byzantine fault tolerance: A unified platform for bft protocols analysis, implementation, and experimentation," in *NSDI 2024*.

[8] H. Qin, C. Wu, M. J. Amiri, R. Marcus, and B. T. Loo, "Bftgym: An interactive playground for bft protocols," in *VLDB 2024*.

[9] P. Sheng, C. Wu, D. Malkhi, M. K. Reiter, C. Stathakopoulou, M. Wei, and M. Yin, "On orchestrating parallel broadcasts for distributed ledgers," in *preparation*.

[10] C. Wu, M. J. Amiri, J. Asch, H. Nagda, Q. Zhang, and B. T. Loo, "Flexchain: an elastic disaggregated blockchain," in *VLDB 2023*.

[11] C. Wu, M. J. Amiri, H. Qin, B. Mehta, R. Marcus, and B. T. Loo, "Towards full stack adaptivity in permissioned blockchains," in *VLDB 2024*.

[12] S. Wang, H. Lyu, J. Zhang, C. Wu, X. Chen, W. Zhou, B. T. Loo, S. B. Davidson, and C. Chen, "Provenance for probabilistic logic programs," in *EDBT 2020*.

[13] H. Chen, C. Wu, A. Zhao, M. Raghothaman, M. Naik, and B. T. Loo, "Synthesizing formal network specifications from input-output examples," in *ToN 2022*.