RESEARCH STATEMENT

Weihai Shen

My research advances high-performance distributed systems and transactional databases by challenging a fundamental principle: *that performance and consistency are inherently at odds*. Modern distributed systems—from financial markets to social networks—face an escalating challenge: processing massive transaction volumes while maintaining strong consistency guarantees. But the accepted trade-off between these goals stems not from physical laws, but from conservative architectural design. My work demonstrates that by decoupling network replication from the critical path, we can achieve both objectives simultaneously.

I design systems where servers proceed speculatively rather than waiting for replication to complete. The central insight driving my work: replication is not inherently expensive; waiting for replication is. Traditional systems place network replication protocols like Paxos on the critical path of transaction execution, prolonging their execution time unnecessarily. My approach fundamentally restructures this relationship: execution proceeds immediately while replication happens asynchronously in parallel. By embracing speculation with principled rollback mechanisms, my systems achieve substantial performance improvements without sacrificing correctness guarantees. Through systematic application of speculation principles, I have built systems that achieve high throughput while maintaining strict serializability—demonstrating that many perceived performance-consistency trade-offs reflect architectural limitations rather than fundamental constraints.

My research contributions span foundational techniques with practical impact, culminating in systems that challenge long-standing assumptions about distributed system design. Rolis [1] (EuroSys 2022) represents a breakthrough in replicated transaction processing, demonstrating that replicated systems can achieve performance close to non-replicated databases while providing full fault tolerance—a result that has influenced how the community thinks about replication overhead. Building on this success, Mako [2] (OSDI 2025) addresses the grand challenge of geo-distributed systems, extending speculation principles across continental distances and achieving significant improvements over established systems like Calvin [3] and Spanner [4] while maintaining strong consistency guarantees.

Throughout my Ph.D., I have focused on creating impactful systems. I have published my research at premier systems conferences and released open-source implementations to expand the impact of these ideas beyond academia, enabling both researchers and practitioners to build upon these foundations.

1 Research Journey: From Replication Bottlenecks to Planetary Scale

1.1 Breaking the Replication Performance Barrier

Traditional architectures create an artificial tight coupling between these operations—systems like Spanner apply Paxos to replicate every step in transaction execution, forcing CPU cores to remain idle. This conservative approach prioritizes safety over performance, achieving only several thousand transactions per second per node despite powerful multi-core hardware. But rather than accepting this as inevitable, I questioned a fundamental assumption: why must transaction execution wait for replication?

To break through this fundamental bottleneck, I developed Rolis, a replicated transaction system built on optimism rather than pessimism, speculation rather than synchronization. The key architectural insight: transaction execution and replication are orthogonal concerns that have been artificially coupled by conservative thinking rather than technical necessity. Rolis introduces a novel execute-replicate-replay architecture that cleanly separates these concerns. Transactions execute speculatively on the primary replica using standard multi-core concurrency control, while independent replication streams asynchronously propagate transaction results to backup replicas, which then deterministically replay the identical sequence of operations.

The technical breakthrough lies in this architectural separation of concerns combined with novel multi-

threaded replication. The primary replica focuses exclusively on maximizing transaction throughput using established multi-core concurrency control algorithms, while completely independent per-thread Paxos streams handle replication without blocking transaction execution. Each execution thread generates its own replication stream, preserving the precise serialization order required for deterministic replay across all backup replicas while completely eliminating expensive cross-thread coordination overhead that bottlenecks traditional systems.

This architectural innovation enables breakthrough performance results: Rolis achieves throughput nearly matching non-replicated single-node databases while providing identical fault tolerance guarantees. By revealing this "fundamental trade-off" as merely an artifact of conservative design choices, Rolis opened the door to rethinking other "fundamental" limitations in distributed systems.

1.2 Scaling Speculation to Planetary Distribution

Building on Rolis's breakthrough results, I next tackled an even more formidable challenge: scaling speculation principles to geo-distributed, multi-shard systems where hundreds of thousands of shards span multiple continents and data centers. This environment presents qualitatively different challenges: network latencies increase by orders of magnitude (from microseconds to hundreds of milliseconds), and failure recovery complexity grows exponentially with the geographic scale and shard count. Traditional approaches that work reasonably well in local area networks completely break down under these conditions.

The fundamental challenge revealed a deeper problem: existing geo-distributed systems place network replication on the critical path even at planetary scale. Wide-area consensus protocols introduce hundreds of milliseconds of latency for replication. These approaches reflect a mindset that requires certainty from replication before proceeding. But what if we could proceed speculatively while replication happens in parallel?

To prove that speculation principles can indeed scale to planetary distribution, I developed Mako, which removes wide-area replication from the critical path of distributed transactions. The key innovation is a vector watermark protocol that enables speculative execution to proceed while replication messages traverse continental distances—maintaining full strict serializability without waiting for cross-continent Paxos to complete. Mako's vector watermarks allow each individual shard to make completely independent progress based on local information, preventing cascading failures and eliminating the system-wide blocking that plagues existing geo-distributed architectures.

Mako's technical breakthrough lies in its coarse dependency tracking mechanism that uses vector timestamps to simultaneously capture logical transaction ordering within individual shards and physical message propagation delays across geographic regions. Each shard maintains vector clocks that track causal dependencies with respect to all other shards in the system, enabling efficient conflict detection without requiring any synchronization or dependency booking. When failures inevitably occur, Mako performs efficient rollbacks of transactions affected by failed shards, while all unrelated transactions continue to commit—avoiding the cascading rollbacks that devastate and block the entire system.

Our comprehensive evaluation demonstrates that Mako achieves significant throughput improvements over state-of-the-art systems like Calvin and Spanner while maintaining robust, predictable performance even under adverse failure conditions. Under high-contention workloads where traditional systems experience complete performance collapse due to replication overhead, Mako maintains stable, predictable performance by effectively isolating conflict effects to minimize system-wide impact. Most significantly, individual shard failures do not block the entire system execution, providing concrete proof that speculation techniques can simultaneously deliver both dramatic performance enhancements and improved fault tolerance properties at planetary scale.

Mako's success validates a broader architectural principle: speculation techniques that eliminate idle waiting by pipelining replication in single-machine systems can be systematically extended to eliminate replication overhead across planetary-scale distribution. This breakthrough suggests that speculation may

provide a general architectural solution to replication bottlenecks across all scales of distributed computing, from multi-core systems to global infrastructures. The implications extend far beyond databases—these principles could transform how we architect distributed consensus protocols, serverless platforms, and distributed machine learning systems.

I have also contributed to systems programming accessibility through collaborative work on DepFast [5] (USENIX ATC 2022), a framework that enables developers to write distributed systems using synchronous programming constructs while maintaining asynchronous performance. This work complements my core research by addressing the practical challenges of building the complex systems that speculation techniques enable. Traditional asynchronous programming models require developers to manually manage complex state machines, callback chains, and coordination protocols, making distributed system development accessible only to experts. DepFast bridges this gap by providing a programming model that allows developers to write natural, sequential code while automatically generating efficient asynchronous implementations underneath. This approach democratizes high-performance distributed programming by hiding the complexity of asynchronous coordination behind familiar synchronous abstractions, enabling a broader community of developers to build scalable distributed systems without requiring deep expertise in distributed systems theory or asynchronous programming patterns.

2 Future Research Directions

My work demonstrates that speculative execution can effectively address traditional performance-consistency trade-offs, pointing toward a future where speculation becomes a foundational design principle for distributed systems. Rather than treating speculation as a specialized optimization, I believe speculative execution can become as fundamental to distributed system design as pipelining is to processor architecture.

Speculation as a First-Class Systems Abstraction : My immediate goal is distilling speculation patterns from Rolis and Mako into a unified design framework applicable across diverse distributed domains. This framework must capture essential elements—dependency tracking, conflict detection, efficient rollbacks—while abstracting implementation complexity that currently limits speculation to experts.

This unified framework would enable speculation to benefit system domains far beyond transactional databases. Distributed consensus protocols could pipeline multiple agreement phases speculatively rather than waiting for each phase to complete sequentially, potentially achieving significant performance improvements. Serverless computing platforms could speculatively pre-execute functions based on predicted request patterns, rolling back only when predictions prove incorrect while reducing cold start latencies. The impact extends to emerging domains as well: edge computing systems could speculatively cache and pre-compute results based on mobility patterns and usage predictions, while blockchain systems could achieve higher throughput through speculative transaction execution that maintains correctness through conflict resolution mechanisms.

References

[1] **Shen, Weihai**, Ansh Khanna, Sebastian Angel, Siddhartha Sen, and Shuai Mu. "Rolis: A software approach to efficiently replicating multi-core transactions". *Proceedings of the Seventeenth European Conference on Computer Systems (Eurosys)*. 2022.

[2] **Shen, Weihai**, Yang Cui, Siddhartha Sen, Sebastian Angel, and Shuai Mu. "Mako: Speculative Distributed Transactions with Geo-Replication". *Proceedings of USENIX Symposium on Operating Systems Design and Implementation (OSDI)*. 2025.

[3] Alexander Thomson, Thaddeus Diamond, Shu-Chun Weng, Kun Ren, Philip Shao, and Daniel J Abadi. "Calvin: fast distributed transactions for partitioned database systems". *Proceedings of the ACM SIGMOD international conference on management of data (SIGMOD)*. 2012.

[4] James C Corbett, Jeffrey Dean, Michael Epstein, Andrew Fikes, Christopher Frost, Jeffrey John Furman, Sanjay Ghemawat, Andrey Gubarev, Christopher Heiser, Peter Hochschild, et al. "Spanner: Google's globally distributed database". *ACM Transactions on Computer Systems (TOCS)* (2013).

[5] Xuhao Luo, **Shen, Weihai**, Shuai Mu, and Tianyin Xu. "DepFast: Orchestrating Code of Quorum Systems". *USENIX Annual Technical Conference (USENIX ATC)*. 2022.