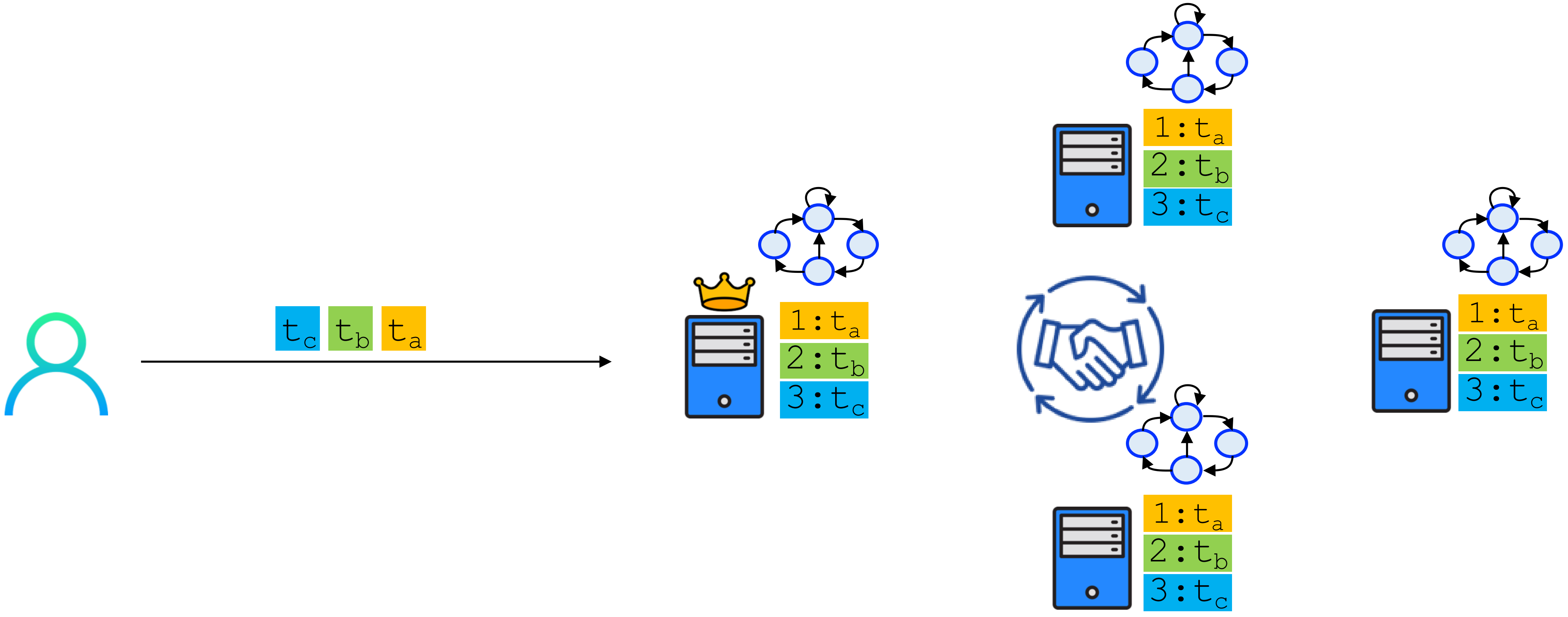
Towards Full-Stack Adaptivity in Transaction Management Systems deployed in Untrusted Environments

Chenyuan Wu   Mohammad Javad Amiri   Haoyun Qin   Bhavana Mehta   Ryan Marcus   Boon Thau Loo

University of Pennsylvania   Stony Brook University

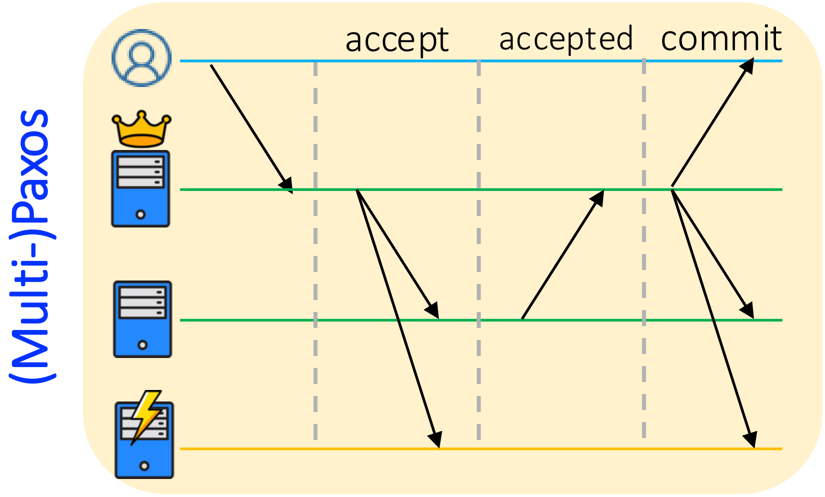
Distributed transaction processing



State Machine Replication: a replicated service whose state is mirrored across different deterministic replicas

- Assign each client request an order in the global service history and execute it in that order

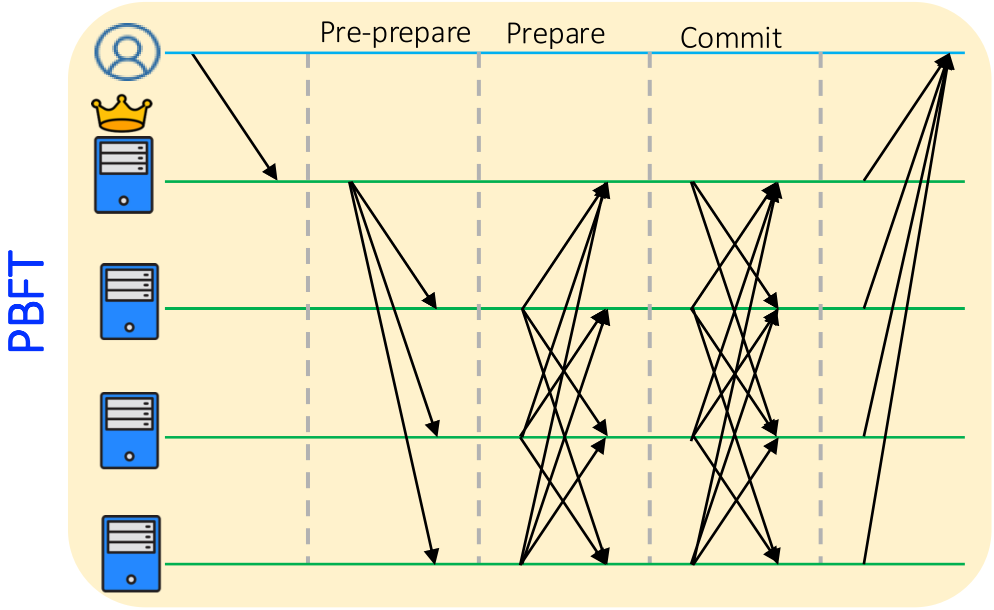
Crash fault-tolerant protocol: (Multi-)Paxos



- Requires 2f+1 nodes to be able to tolerate f parallel crash failures
- How to deal with Byzantine failure?
    - nodes exhibit arbitrary, potentially malicious, behavior
    - Potential causes: software bugs, hardware failures, malicious attacks

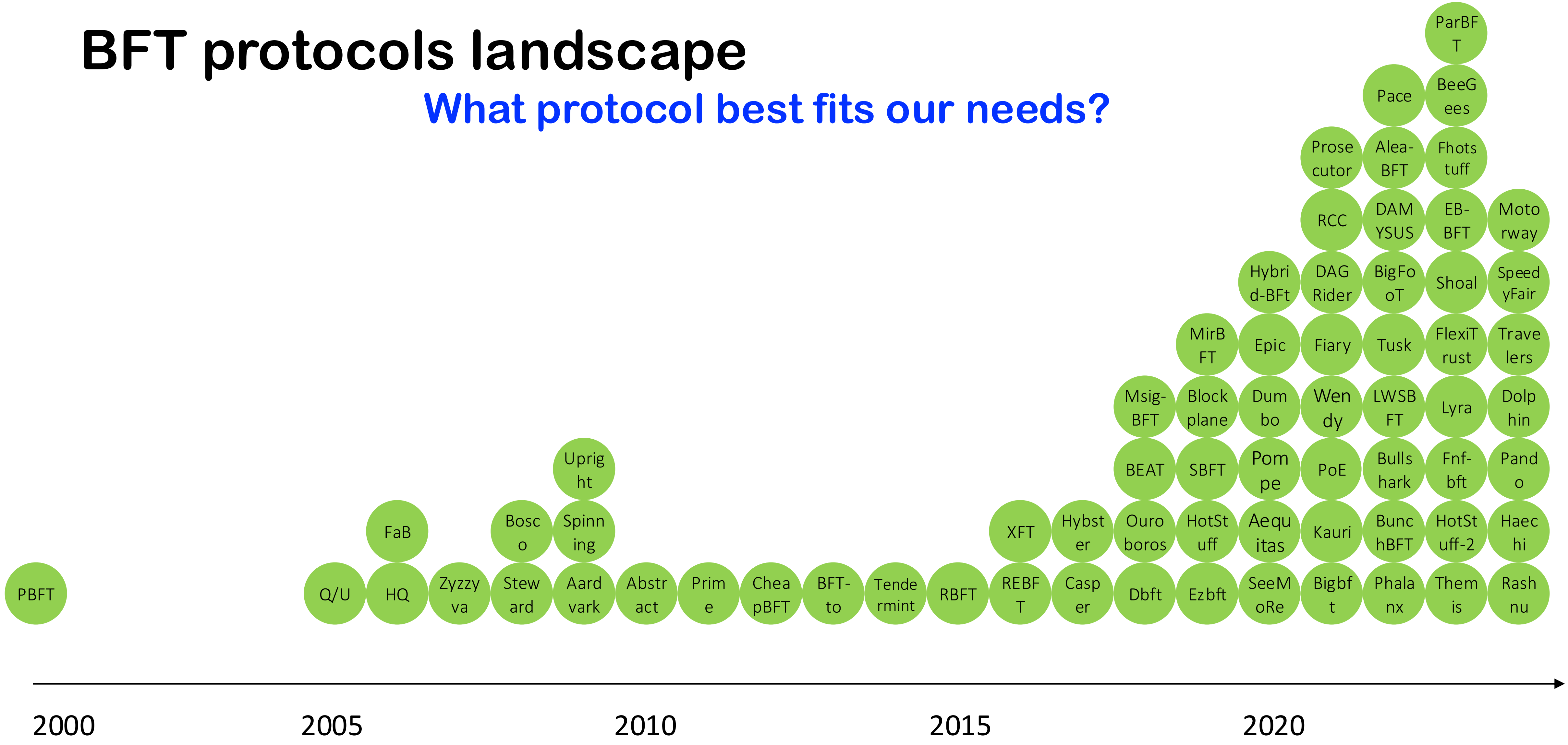
Byzantine fault-tolerant protocol: PBFT

- Nodes can fail arbitrarily, including deviating from the protocol
- Require 3f+1 nodes to tolerate f concurrent failures
- E.g., PBFT



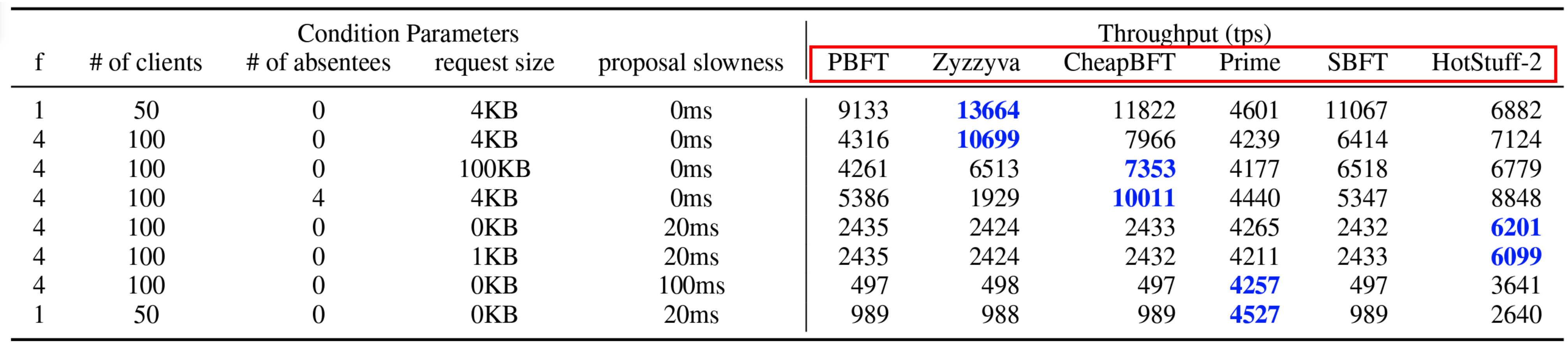
BFT protocols landscape

What protocol best fits our needs?



Challenge 1: consensus protocol

- No one-size-fits-all BFT consensus protocol

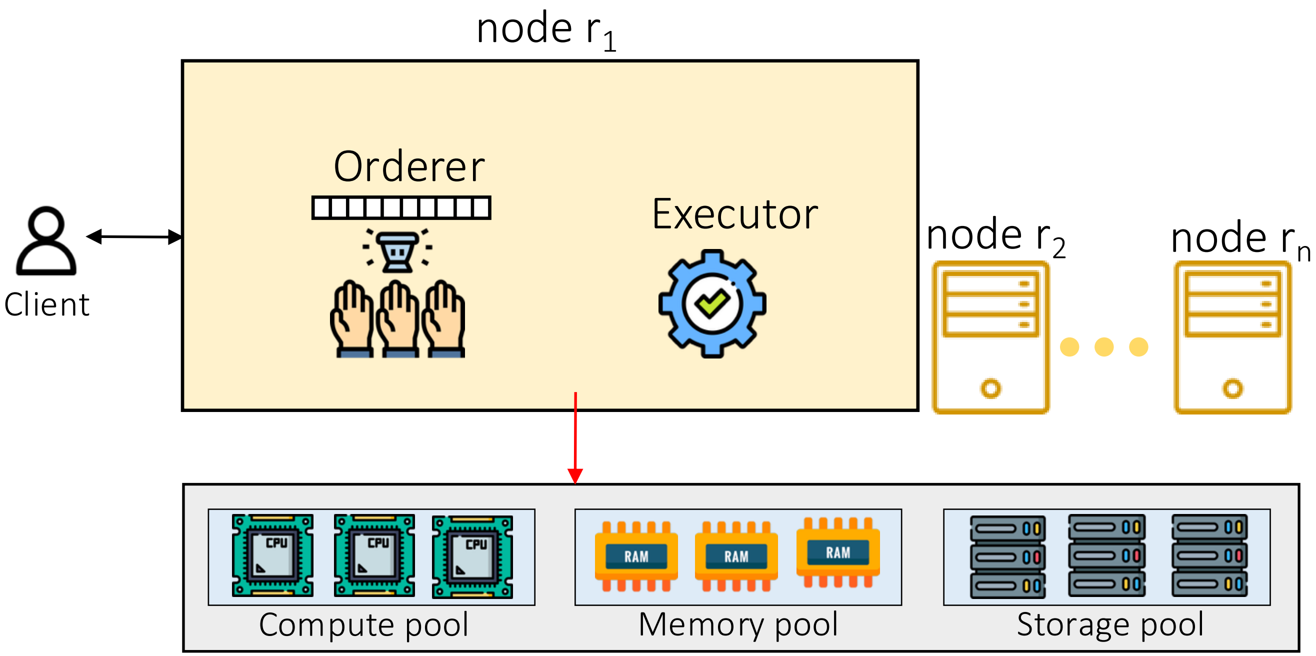


		Condition Parameters					Throughput (tps)			
f	# of clients	# of absentees	request size	proposal slowness	PBFT	Zyzzyva	CheapBFT	Prime	SBFT	HotStuff-2
1	50	0	4KB	0ms	9133	**13664**	11822	4601	11067	6882
4	100	0	4KB	0ms	4316	**10699**	7966	4239	6414	7124
4	100	0	100KB	0ms	4261	6513	**7353**	4177	6518	6779
4	100	4	4KB	0ms	5386	1929	**10011**	4440	5347	8848
4	100	0	0KB	20ms	2435	2424	2433	4265	2432	**6201**
4	100	0	1KB	20ms	2435	2424	2432	4211	2433	**6099**
4	100	0	0KB	100ms	497	498	497	**4257**	497	3641
1	50	0	0KB	20ms	989	988	989	**4527**	989	2640

Various BFT protocols

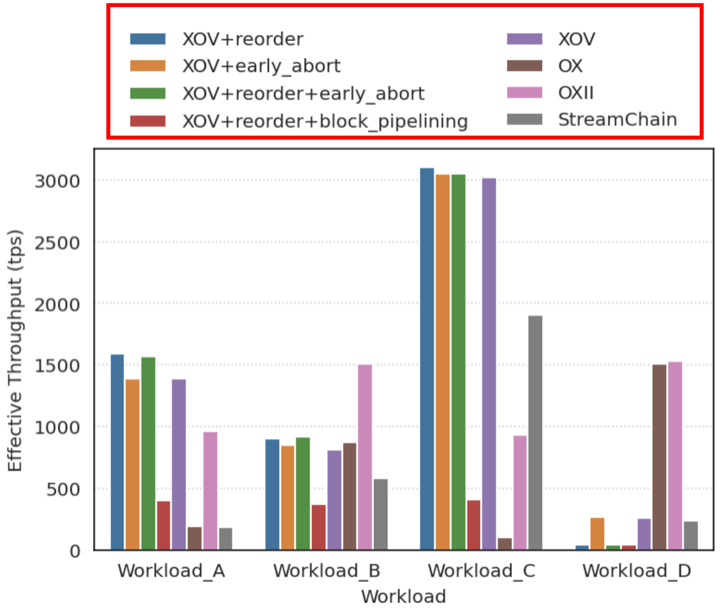
Beyond consensus

- The system performance is affected by other layers as well…
  - Transaction management paradigm
  - Infrastructure (hardware resources)



Challenge 2: transaction management

- **No one-size-fits-all transaction management paradigm**
- Varying design principles
  - The sequence in which ordering/execution/validation are performed, # of transactions in a block, the use of reordering and early aborts
- Transaction workloads fluctuate, nodes join and leave, intermittent faults and attacks

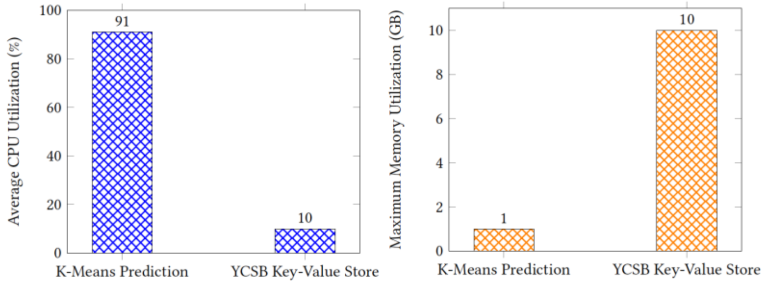


Various transaction management paradigms

Workload	Write Ratio	Contention Level	Load	Compute Intensity
A	low	high	high	high
B	moderate	high	moderate	low
C	moderate	low	high	Very high
D	high	very high	moderate	Very low

Challenge 3: hardware

- **No one-size-fits-all resource provisioning**
- Diverse new applications are emerging every day, each with distinct and heterogeneous resource demands
- A smart contract can interchangably be compute- and memory-intensive at different times and execution stages



Filecoin is making the web more secure and efficient with a decentralized data storage marketplace.
Medicalchain uses blockchain technology to securely manage health data for a collaborative, smart approach to healthcare.
Aggregata is scaling decentralized value of AI Data, powered by DePIN-driven aggregation.

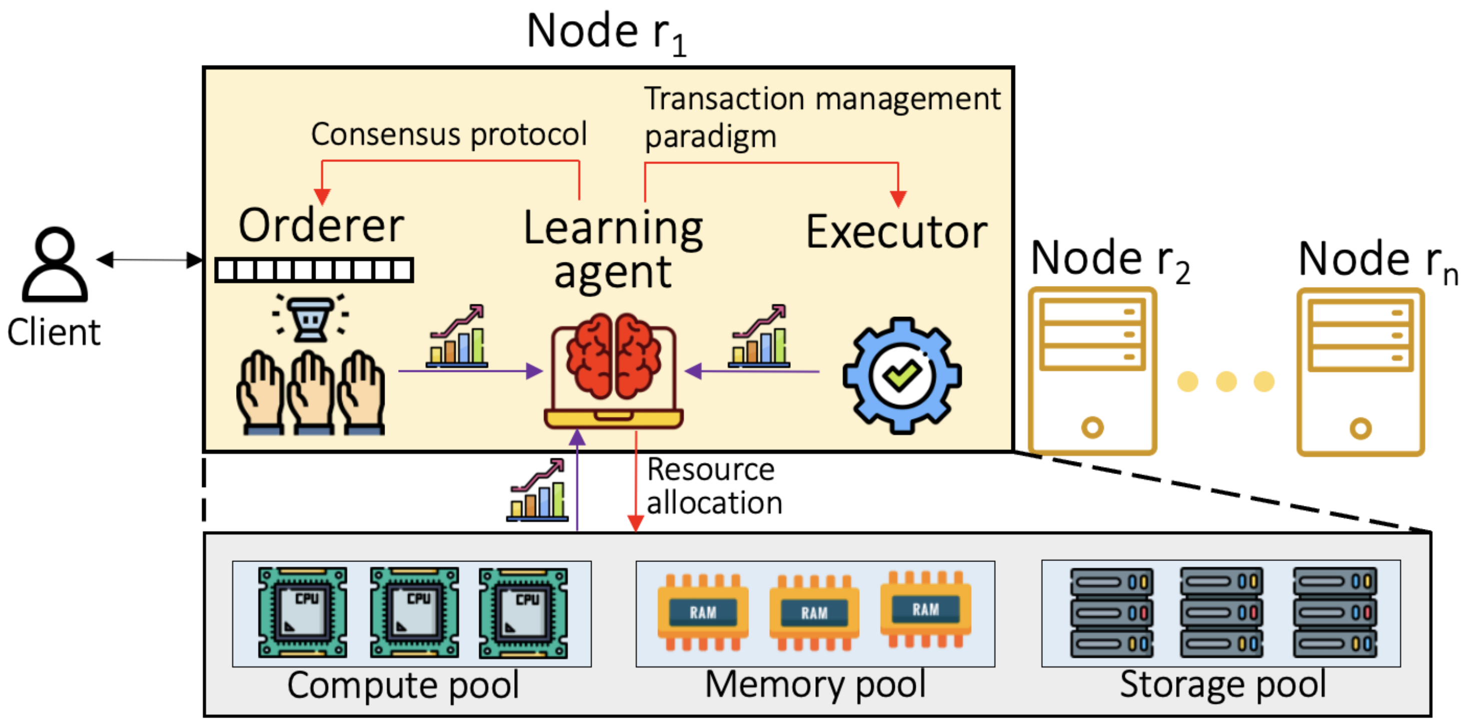
Memory and storage intensive

Nesa is the lightweight Layer-1 executing critical AI inference on queries that require a high degree of privacy, security, and trust using ZKML on-chain.
Holoworld is a decentralized AI character marketplace and social platform where anyone can create powerful, intelligent AI bots with a few clicks.
Story Chain is an innovative multi-level AI-based dApp that fosters collaborative storytelling.

Computation intensive

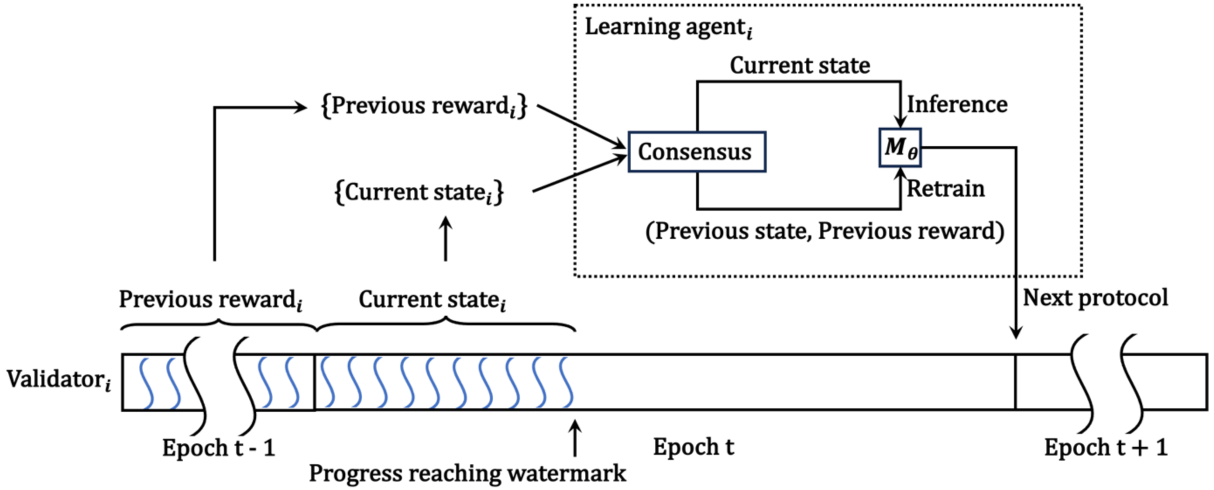
Vision: a fully adaptive system

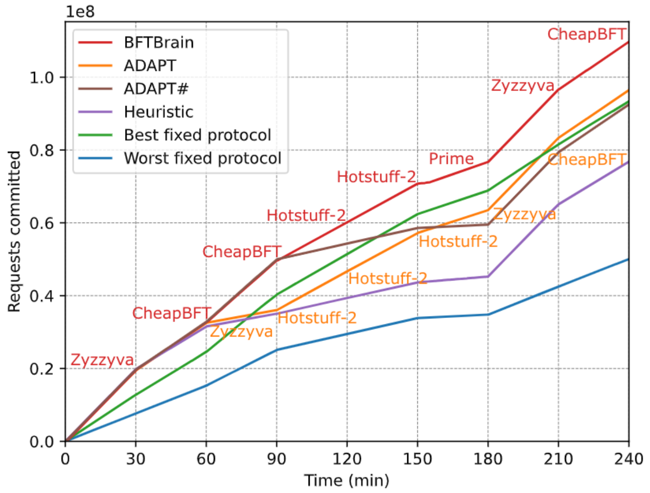
- Leverages machine learning techniques and resource disaggregation to address abovementioned challenges
- Designed with full-stack adaptivity in mind



Adaptive BFT protocol: our first step

- BFTBrain [NSDI’25] uses reinforcement learning techniques to learn the next BFT protocol to switch to *on-the-fly*
- Key innovations
  - Employs novel fine-grained features that offer deeper performance insights, e.g., # of received messages per slot, interval between consecutive leader proposals
  - Decentralized coordinates real-time feature collection and RL engine, resillient to data pollution





Wu, C., Qin, H., Amiri, M. J., Loo, B. T. , Malkhi, D., Marcus, R., BFTBrain: Adaptive BFT Consensus with Reinforcement Learning. NSDI’25

Adaptive BFT protocol: future

- Adversarial machine learning
  - Decision attacks: target the inference phase, an adversary reports false value to disturb the global feature
  - Poisoning attacks: target the training phase, an adversary reports carefully selected feature values and labels to cause the next trained model to be inaccurate
  - How would they affect the system and how to defend against it?

- Protocol specific parameters tuning
  - Continuously tune protocol behavior including internal parameterization, e.g., timer values, batch size

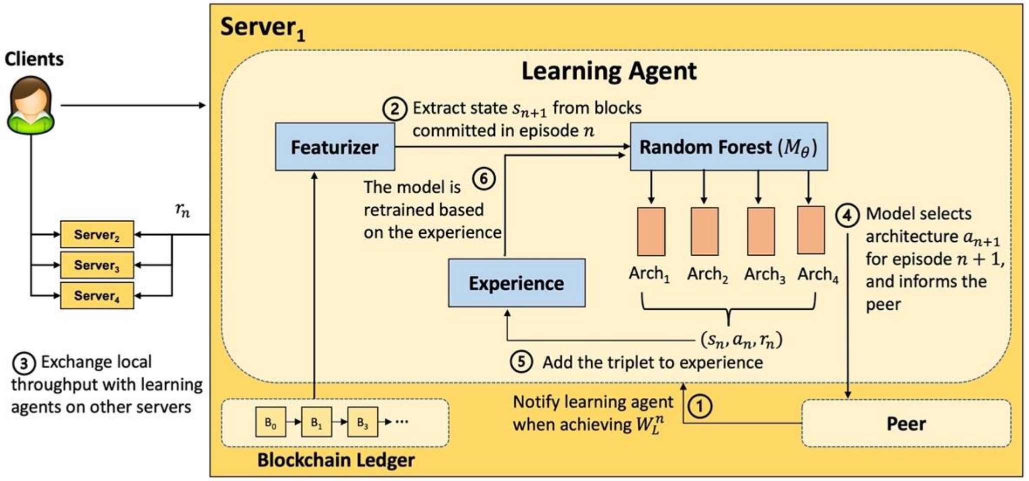
- Reducing overhead
  - Although exploration is inherent, its overhead can be made lower by launching experiments in some "shadow mode" such that mainline performance is not affected

- Discovery of novel protocols
  - Discover new BFT protocols that fit new environments or meet new application requirements through the learning framework, by changing protocol design attributes outlined by Bedrock [NSDI'24]
  - How to systematically ensure and prove the correctness of the newly discovered protocols?

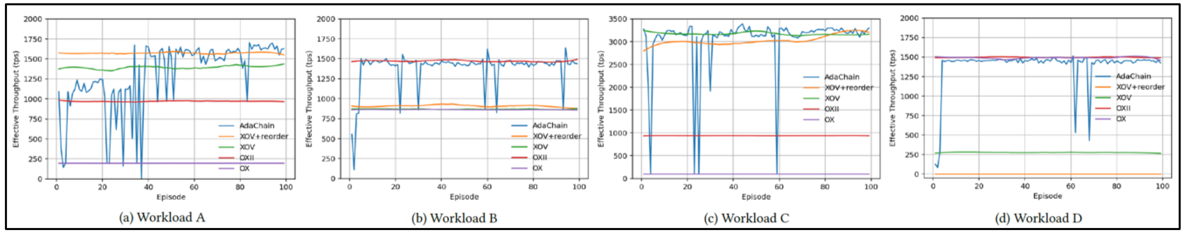
Amiri, M. J., Wu, C., Agrawal, D., El Abbadi, A., Loo, B. T., & Sadoghi, M. The Bedrock of Byzantine Fault Tolerance: A Unified Platform for BFT Protocol Analysis, Implementation and Experimentation, NSDI'24 **[Outstanding Paper Award]**

Adaptive transaction management: our first step

- AdaChain [VLDB'23] uses reinforcement learning to learn the mapping of (workload characteristics -> optimal transaction management paradigm) *on-the-fly*
- Key innovations
  - Models the selection of a paradigm as a contextual multi-armed bandit problem
  - Introduces protocols to switch from one paradigm to another in a live system while respecting correctness and security concerns



Clients
Server$_1$
Learning Agent
② Extract state s_{n+1} from blocks committed in episode n
Featurizer
Random Forest (M_θ)
The model is ⑥ retrained based on the experience
Experience
Arch$_1$
Arch$_2$
Arch$_3$
Arch$_4$
④ Model selects architecture a_{n+1} for episode $n+1$, and informs the peer
(s_n, a_n, r_n)
⑤ Add the triplet to experience
r_n
Server$_2$
Server$_3$
Server$_4$
③ Exchange local throughput with learning agents on other servers
B$_0$
B$_1$
B$_3$
…
Blockchain Ledger
Notify learning agent when achieving W_L^n ①
Peer



(a) Workload A

(b) Workload B

(c) Workload C

(d) Workload D

Wu, C., Mehta, B., Amiri, M. J., Marcus, R., & Loo, B. T. AdaChain: A Learned Adaptive Blockchain, VLDB'23

Adaptive transaction management: future

- Learning strategies
  - Different problem formulation: CMAB (assumes epoch independency) vs. full RL (current action affects the future state)
  - Value-based model vs. policy-based model
  - How well do they perform?

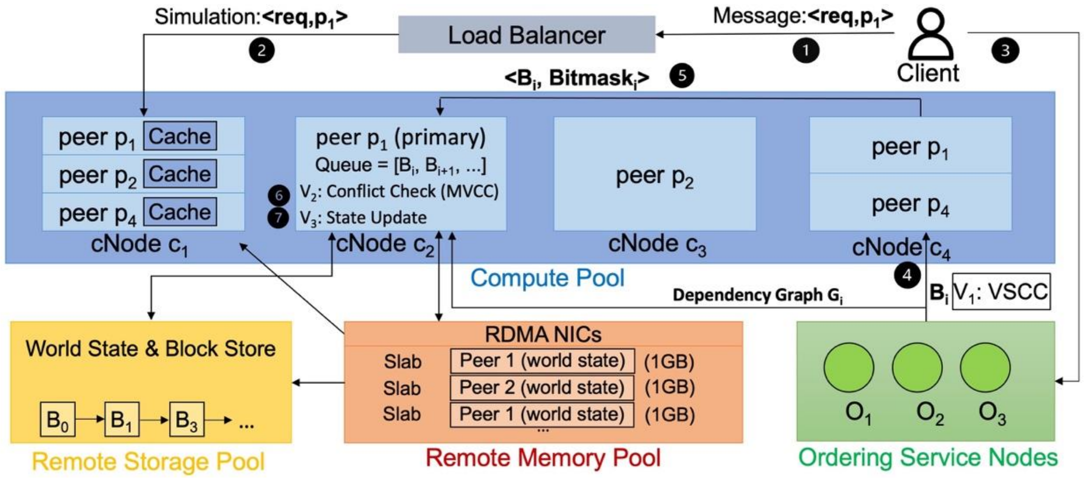
- Featurization
  - Automatic state extraction from ledger (log): deserialize to conflict graph, where each edge is annotated with the submission and commit timestamp, then use GCN to extract the states automatically

- Uncovering new transaction processing paradigms
  - Can learning framework mix and match design attributes? E.g., OXII + reordering + early aborts
  - Consider three transactions: T1(A), T2(A, B), T3(B)

- Finer-grained adaptation
  - Can we directly learn the best final order of transactions, instead of choosing between enabling/disabling the reordering algorithms used by Fabric++?
  - Can we adapt on a per-transaction basis, instead of on a per-epoch basis (a constant number of blocks)?

Adaptive infrastructure: our first step

- FlexChain [VLDB’23]: a disaggregated infrastructure
  - Demonstrating efficient resource utilization and elastic scaling, while incurring at most 12.8% overhead in using remote memory
- Adopts Execute-Order-Validate (XOV) architecture
  - The execution phase is fully in parallel
  - The first part of the validation phase (endorsement policy evaluation) is inherently parallelizable
  - Most of the states resides in key-value stores



Wu, C., Amiri, M. J., Asch, J., Nagda, H., Zhang, Q., & Loo, B. T. FlexChain: an elastic disaggregated blockchain, VLDB’23

Cross-layer adaptivity: future

- Identifying performance bottleneck in an end-to-end system
  - Transaction processing? BFT consensus? Under-provisioned resource?
  - Avoid unnecessary configuration switching or resource over-provisioning

- Disaggregation or not?
  - How do other transaction processing paradigms (other than XOV) perform on DDCs?
  - Given the current workload and remote memory overhead, is it worthwhile to disaggregate?
  - View the blockchain ledger as multi-channel time series data, and forecast the workload changes that should lead to a transition in infrastructure

Questions?