

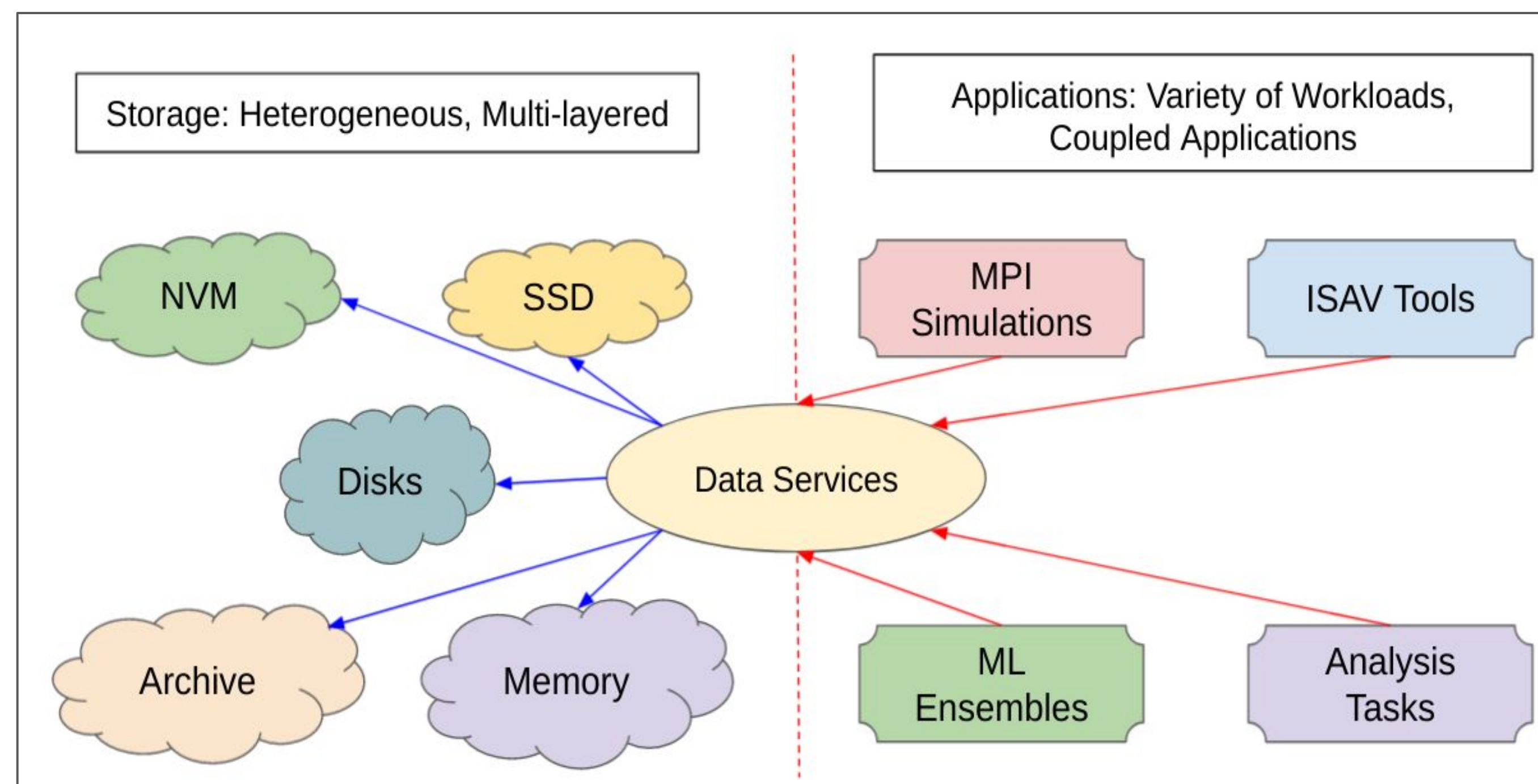
SYMBIOMON: A High Performance, Composable Monitoring Service for Online Application Introspection and Adaptation

S. Ramesh, A.D. Malony, and K. Huck
University of Oregon
<http://tau.uoregon.edu>

R. Ross, M. Dorian, and P. Carns
Argonne National Laboratory
<https://www.mcs.anl.gov/research/projects/mochi>

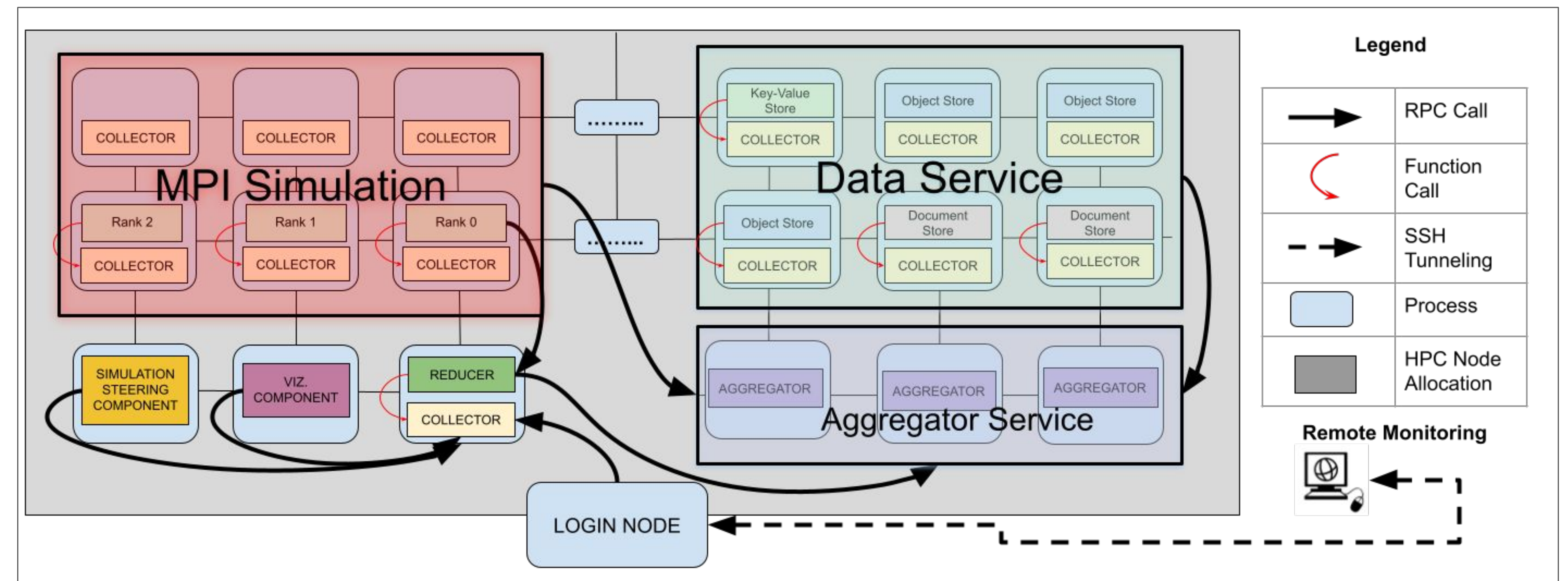
Research Challenge

Creating a high performance, composable monitoring service capable of simultaneously collecting, aggregating, and exporting performance data from multiple distributed components while operating with low overhead and offering an interface that enables seamless integration with HPC applications and distributed services alike.



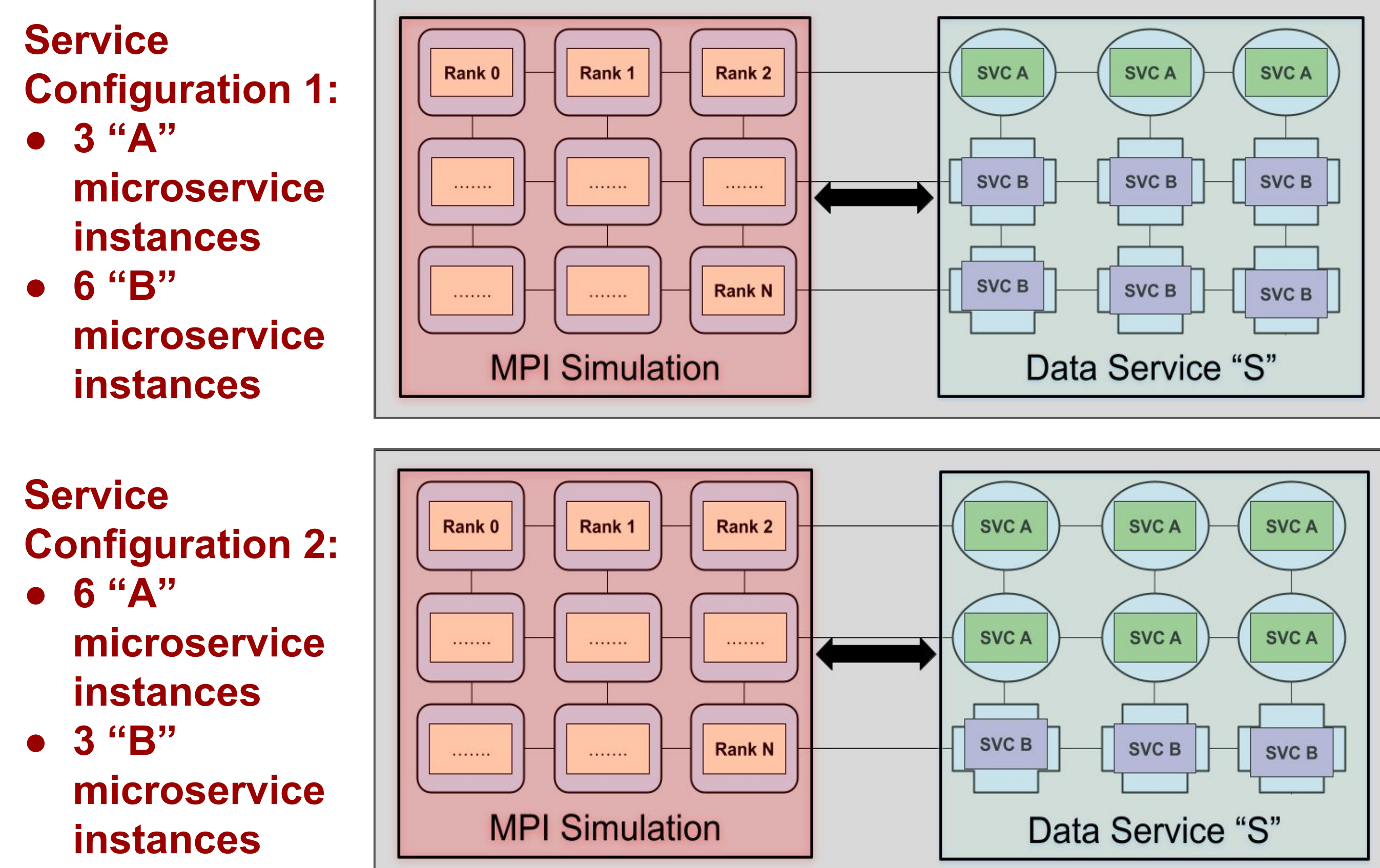
Proposed Approach

- Time-series monitoring service composed of three core components: COLLECTOR, AGGREGATOR, and REDUCER
- Composable design is chosen for maintainability, flexibility in resource allocation, and the ability to selectively toggle analysis capabilities
- Components are Mochi microservices composed using high-performance Mercury RPC

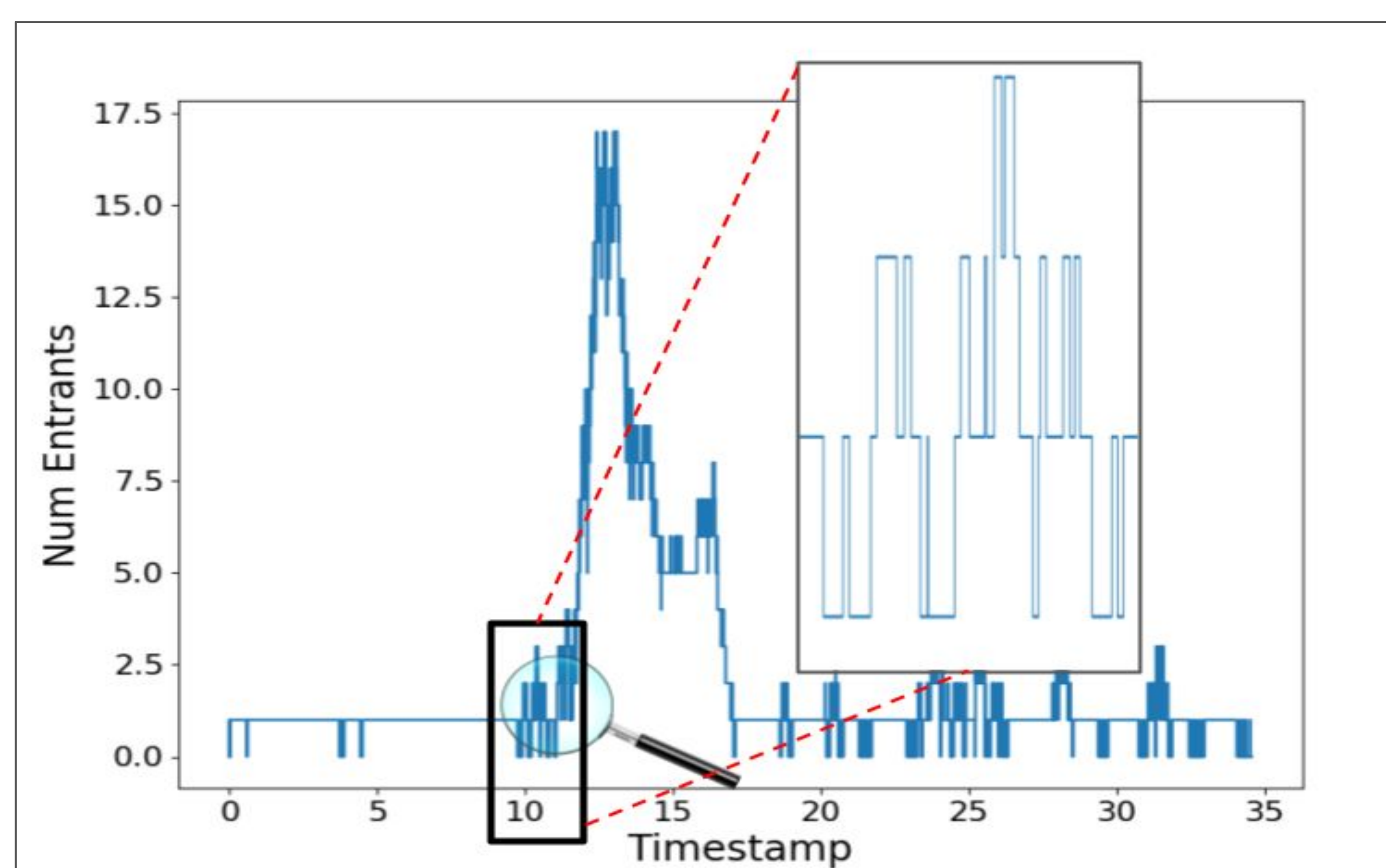


Case Study: Data Service Configuration

Data service configurations can be functionally equivalent but have vastly different performance characteristics



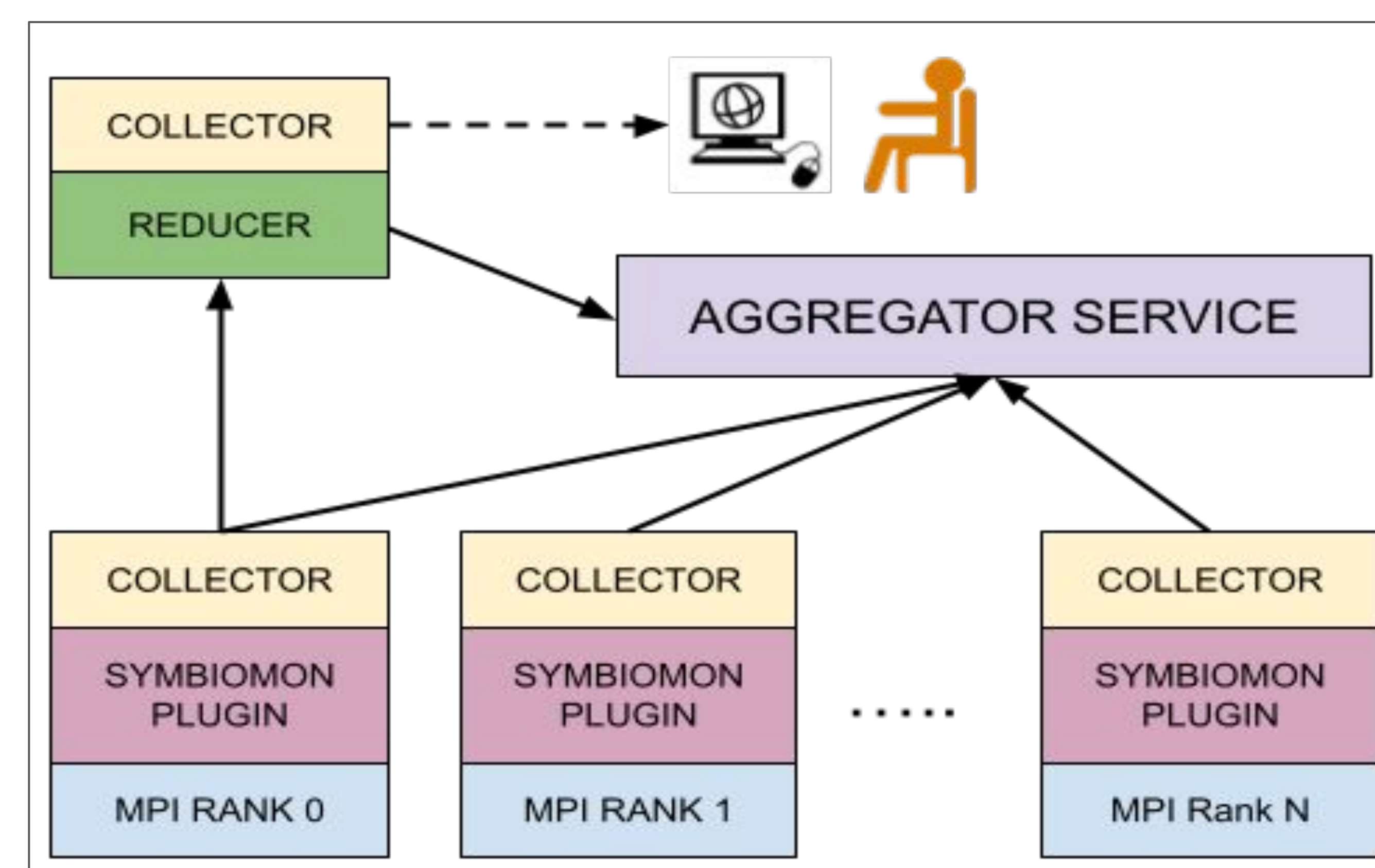
Using SYMBIOMON to generate a better data service configuration by observing the impact of concurrency on database performance



- Extract and monitor the "num entrants" SYMBIOMON metric to observe the degree of concurrent database write access for the Mochi SDS key-value microservice (SDSKV)
- Analyze the "concurrency regions" to reveal the degree of performance loss for concurrent access as compared to baseline serialized execution
- Use the analysis to iteratively approach optimal database configuration

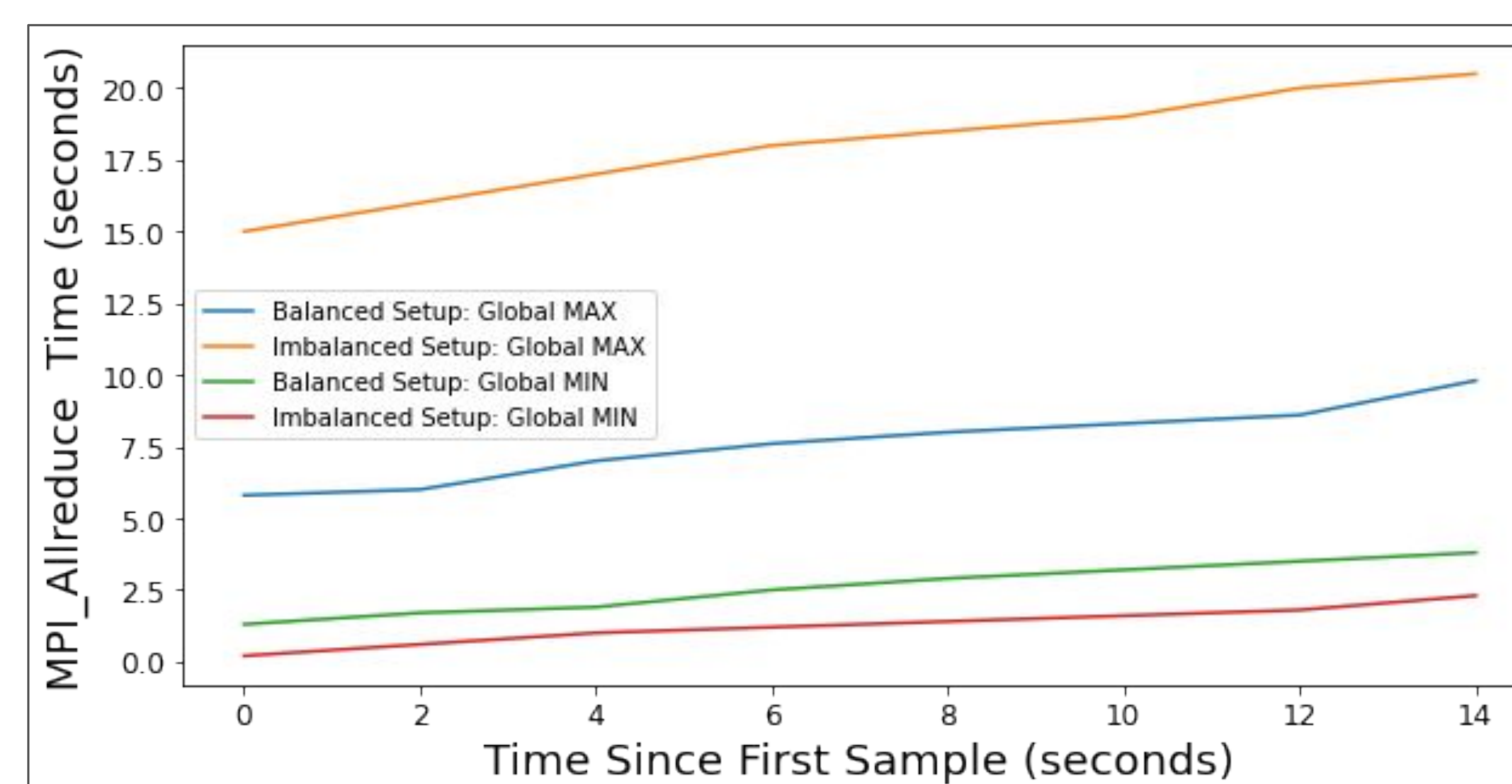
Case Study: Application Monitoring

Use of the TAU plugin infrastructure to monitor, aggregate, and analyze application performance data online



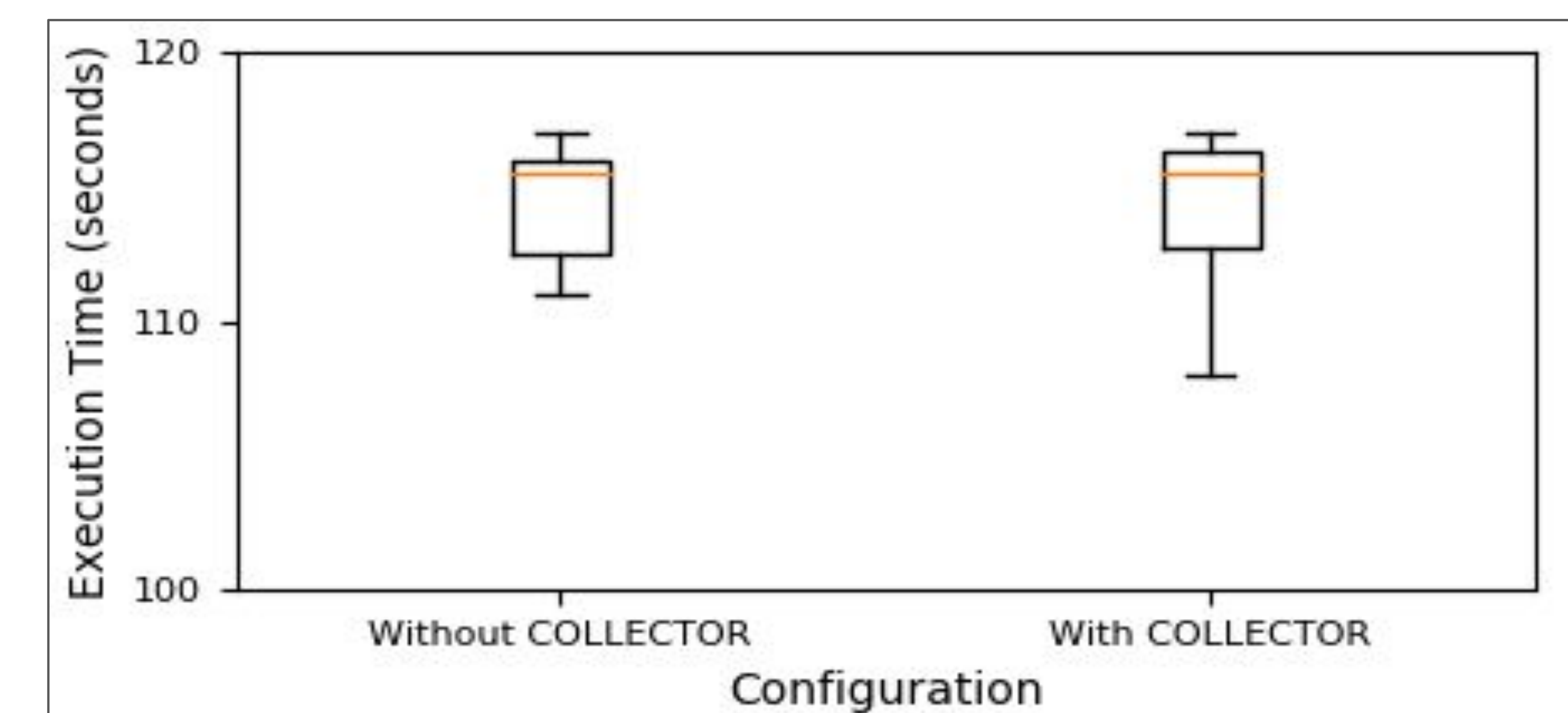
- Plugin converts TAU performance data and events to corresponding SYMBIOMON metrics
- Process-local COLLECTOR instance connects with the AGGREGATOR service to store locally-reduced time-series metrics
- Rank 0 contacts the REDUCER instance to enable global reduction of performance data
- Reduction operators currently supported: MIN, MAX, AVG, SUM, ANOMALY
- REDUCER exposes globally reduced value for online monitoring

LULESH: Monitoring load imbalance by measuring the range of time spent inside MPI_Allreduce among the MPI processes

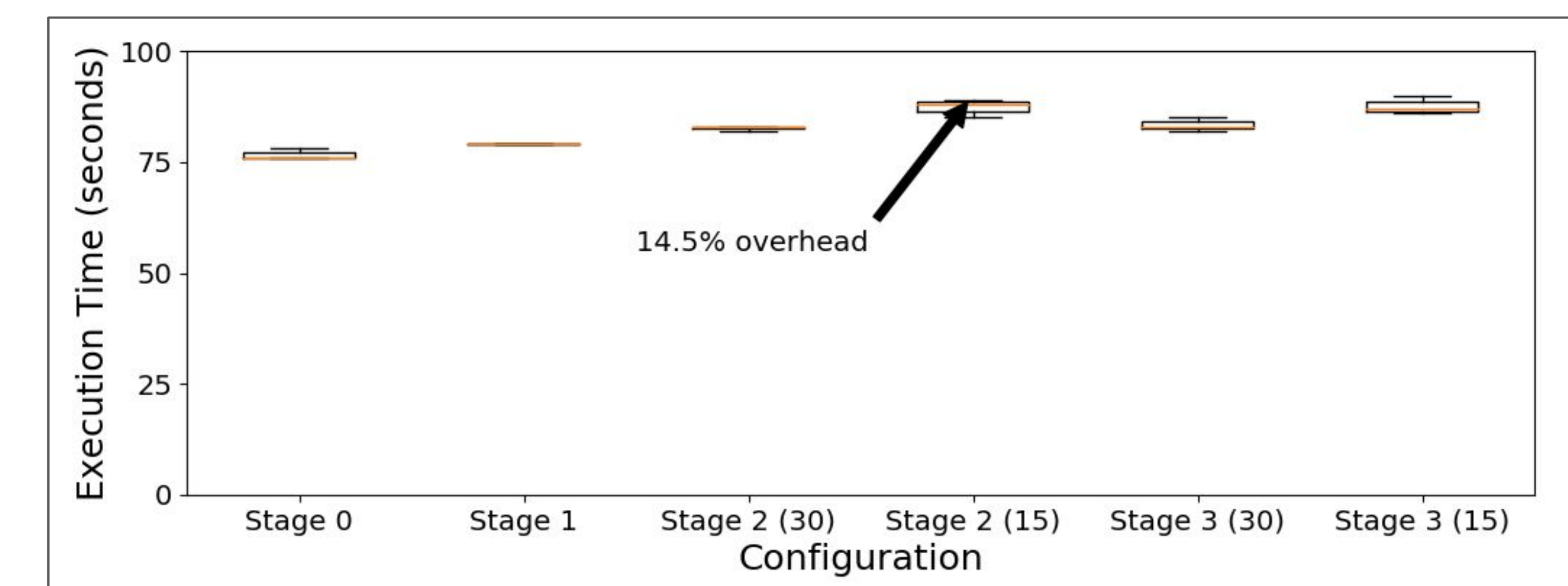


Performance Overhead

Impact of enabling the COLLECTOR for the SDSKV microservice



Impact of enabling the COLLECTOR, AGGREGATOR, and REDUCER for LULESH performance monitoring



- Stage 0: Uninstrumented binary; Stage 1: Enabling COLLECTOR, AGGREGATOR; Stage 3: Enabling all three components
- Results: COLLECTOR introduces no overhead; Local aggregation introduces between 6-14% overheads for reducing 45 metrics depending on the frequency of reduction; Global aggregation reduces no additional overhead

Future Work & Research Dissemination

- Use of SYMBIOMON to assist in the automatic online re-configuration of HPC data services
- Employing SYMBIOMON as a complementary aid to more general, ML-based optimization frameworks
- Extending SYMBIOMON for dynamic control of other distributed components