Automated Configuration Parameter Tuning in Distributed Messaging Systems

Emmanuel Etti, Md. Monzurul Amin Ifath and Israat Haque

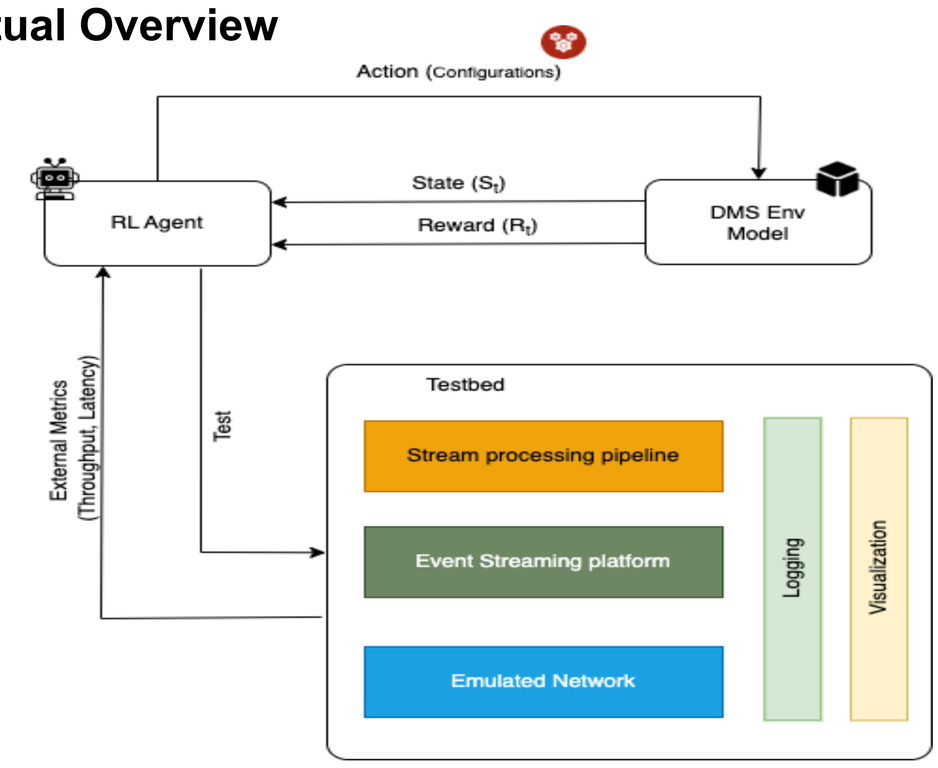
1. Problem Statement

- Distributed Messaging Systems (DMS) are essential for real-time data analytics (e.g., fraud detection, video analytics, IoT).
- Popular DMS (e.g., Apache Kafka) require manual configuration, which is time-consuming and error-prone.
- How can we automate DMS configuration tuning to improve performance?

2. Approach

- We propose S2Gconfig, a system that uses deep reinforcement learning to automate configuration parameter tuning for DMS.

Conceptual Overview



Inputs

- External metrics (throughput, latency).
- Desired network topology in GraphML file.

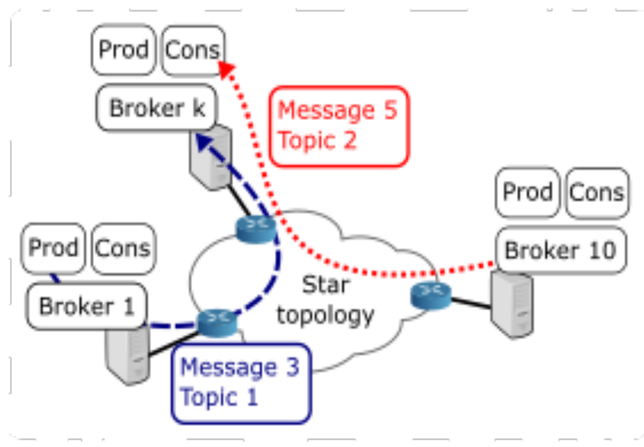
Workflow

- Collect training data(configuration parameters and metrics) from testbed.
- Train DMS environment prediction model.
- Provide RL agent with metrics that should be optimized
- Recommend the best configuration parameters to achieve optimal performance within certain latency bound.
- Starts DMS environment with recommended config.

3. Setup Details

- Network topology emulated in Mininet 2.3.0 [1].
- Apache Kafka 2.8.0 [2] for streaming events.
- OpenAI Gym [3] for RL Environment.
- Network configuration and logs as OpenFlow 1.3 rules and statistics, respectively.
- Data visualization using Matplotlib 3.3.4.

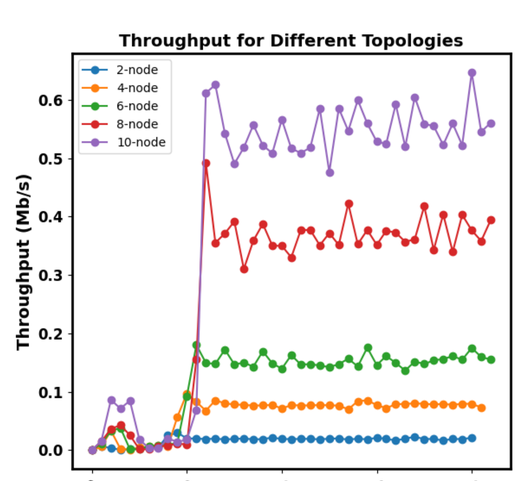
4. Example Use Case

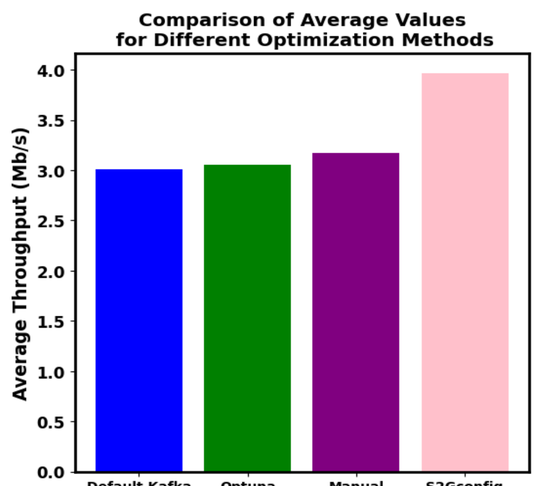


- Dissecting DMS's performance.
- Ten brokers placed in geo-distributed locations.
- Each location hosts one producer and one consumer.
- Star topology, 1 Gbps links.
- Each producer publishes data uniformly over 10 topics at a total 30KBps rate.

5. Preliminary Results

- Our system is capable of automatic parameter configurations using Deep Deterministic Policy Gradient (DDPG).
- Facilitate 2,4,6,8,10 node topologies.
- S2Gconfig provides parameters to achieve 1.3x better throughout than default kafka within the latency boundary.





6.Next Steps

- Develop a realtime system that recommends and applies recommended configurations.
- Compare different RL methods such as PPO, SAC, A2C.
- Expand capability to generate test data for different topology sizes.

7. References

[1] Mininet. 2021. http://mininet.org/

[2] Apache. 2021. https://kafka.apache.org/

[3] OpenAI Gym. 2016. https://gymnasium.farama.org/