Configuration Tuning for Distributed IoT Message Systems Using Deep Reinforcement Learning

Zhuangwei Kang, Yogesh D. Barve, Shunxing Bao, Abhishek Dubey, Aniruddha Gokhale
Vanderbilt University, Nashville, USA

Goals and Overview: Distributed messaging systems (DMSs) provide users with a set of continuous and discrete configurable parameters that have different data types and value ranges, which together result in a **hybrid, multidimensional configuration space**. By fine-tuning DMS configurations, we aim to optimize the publisher-side throughput of DMS applications while meeting latency constraints, such as:

$$\max_{C \in CP} TP(W, T, R, C)$$
$$s.t.\ latency \leq L_c$$

, where TP denotes publisher-side throughput, W is input workload, T is system topology, R is system resource profile, C is a specific configuration vector, and L_c is the restriction imposed on system latency.

Challenges

- The search space grows exponentially as the number of tunable parameters increases.
- Requires significant domain knowledge and in-depth understanding of the impact of each parameter on application performance and their unseen interactions.
- The default configurations are usually suboptimal.
- Naive exhaustive search methods are laborious, error-prone and suboptimal.

Methodology

We propose a Deep Reinforcement Learning (DRL)-based configuration recommendation system, called **DMSConfig**. It is built using container-based emulation techniques, conventional machine learning, and the DDPG[1] DRL-based algorithm, which is utilized in three stages data collection, DMS simulator training, and configuration tuning.

Data Collection

- Sample multi-dimensional config space using Latin Hypercube Sampling.
- Emulate DMS workloads using container techniques and traffic control.
- Collect DMS internal and external state metrics using Collectd.

DMS Simulator Training

DMSConfig adopts the random forest[2] (RF) algorithm to train a DMS simulator that takes a number of performance-relevant parameters as input and forecasts several software internal state metrics, publisher-side throughput, and latency.

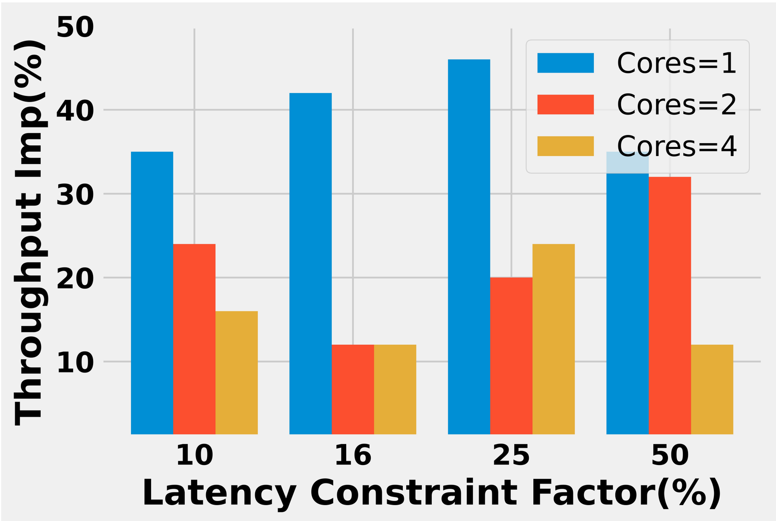
DRL-based Configuration Tuning

- Convert the latency-constrained configuration tuning problem to a Markov Decision Process and solve it using the DDPG algorithm.
- The auto-tuner (RL Agent) gradually enhances the likelihood of selecting high-quality configurations (RL Action) through trial and error.
- The derived optimal searching strategy (RL Policy) can navigate the auto-tuner to obtain the maximum cumulative return, and the action taken to reach the terminal state is the best configuration.

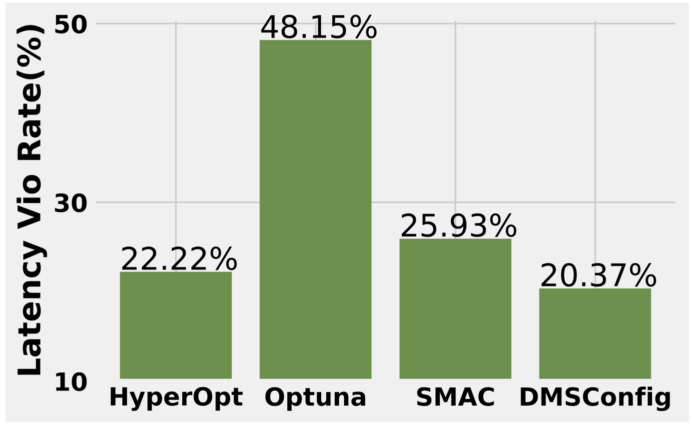
Why DRL?

- The sequential decision-making process in RL coincides with the essence of iterative parameter adjustment.
- DDPG has been proven a robust approach for settling continuous control problems(continuous configuration in our context).
- The reward function in RL guides the tuning process by applying revenue or penalty to the agent, which satisfies our demand for throughput and latency simultaneously.
- Driven by the model-based DMS simulator and RL reward mechanism, DMSConfig can rapidly adapt tuning requests that have different latency constraints.

Initial Experimental Results



Our initial experimental results, conducted on a single-broker **Kafka** cluster, reveal that the configurations identified by DMSConfig significantly outperform the default configuration provided by Kafka vendor under several levels of lcf. DMSConfig is also able to guarantee application performance under resource-constrained(CPU, bandwidth) environments by making effective configuration recommendations.



DMSConfig earns analogous throughput performance compared with the three baselines but delivers the most reliable latency guarantees.

Future Work

1. Optimize the DDPG reward function and neural network design to enhance throughput and reduce latency violation occurrence rate;
2. Extend the single-broker DMS configuration problem to multi-broker scenarios.

[1] Lillicrap, Timothy P., Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. "Continuous control with deep reinforcement learning." *arXiv preprint arXiv:1509.02971* (2015).
[2] Breiman, Leo. "Random forests." *Machine learning* 45, no. 1 (2001): 5-32.