1 Paper Overview

"Ray: A Distributed Framework for Emerging AI Applications", written by Robert Nishihara, Stephanie Wang, et al. is a system paper published in OSDI'18. Identifying the trend of continuous interaction with environment for future AI application, this paper propose a distributed machine learning system to address new requirement in terms of performance and flexibility [3]. This system enables simulation, training and serving for RL applications, providing a unified interface to support task-parallel and stateful computation. In general, the contributions of this paper is summarized as follows:

1. This is the first distributed framework that unifies training, simulation and serving.
2. Unify actor and task-parallel abstractions on top of a dynamic task execution engine.
3. A system design principle to help scalability and fault tolerance, which separate lineage recording from other system components, is proposed.
4. A bottom-up distributed scheduling strategy with concerns about data locality and local node loads is illustrated.

1.1 Problem Summary

There is no general-purpose system today can efficiently support the tight loop of training, serving, and simulation, which express the core building blocks and meet the demands of emerging AI applications [3].

1.2 Related Works

- Clipper: A low-latency online prediction serving system [2].
- Tensorflow: A system for large-scale machine learning [1].

2 Paper Strengths

This paper is well written and the illustration is easy to follow. The developing experience for the API extension and the reflection about the fault tolerance are insightful to read. The design of GCS is one of the unique ideas proposed in this system for its internal state record and check, demonstrating better debuggability than other production machine learning system. The effort to construct a better usage system is worthy acknowledging, including system-wise design and extension awareness. Also, the evaluation manner, which compares different stages with other systems, is illustrative and easy to follow.

1. The design concerns for GCS and two-level task scheduler possibly illustrates a future way for distributed machine learning system.
2. The contribution of this paper is illustrated by comparison with other system and the block-by-block evaluation is good for system comparison.

3 Paper Weaknesses

Although this system provides an all-in-one solution for machine learning task, especially for reinforcement learning, it lacks uniqueness for addressing problems in machine learning life cycle. There are multiple replacements designed for specific stage challenges and this system is unable to overrun other systems. The usage of local data locality is one of the mentioned contribution of this system. It seems a huge improvement in the data simulation evaluation. However, in the comparison with other systems, there is no bottleneck found in terms of data locality. As for training, the performance of different systems are in the same level. The evaluation of SGD performance is not persuasive, considering the possible optimization for TensorFlow ecosystem, and the difference for multi-core performance is increasing, leading skepticism for scalability of this proposed system. Also, the influence of this system seems to be much smaller than the other machine learning system, such as PyTorch and TensorFlow. This system is built with actor design pattern, which could suffer from dead locks. And this possible concern is not illustrated in the paper.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

[2] Daniel Crankshaw, Xin Wang, Guilio Zhou, Michael J Franklin, Joseph E Gonzalez, and Ion Stoica. Clipper: A low-latency online prediction serving system. In *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)*, pages 613–627, 2017.

[3] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I Jordan, et al. Ray: A distributed framework for emerging {AI} applications. In *13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18)*, pages 561–577, 2018.