Universität Paderborn — Fakultät EIM
Fachgebiet Rechnernetze
http://www.upb.de/cs/cn/

Master Thesis

Learning Coflows Scheduling

Background

Data-parallel applications run on clusters of machines, which are connected through a datacenter network with multiple links. An applications' resource demands on these links are often *correlated*, e.g., their communication usually has data patterns like MapReduce, partition-aggregate etc. In literature, one way to abstract such communication patterns is a *coflow*. A coflow is only considered complete when all of its constituent flows have finished their data transmission.

Current coflow scheduling heuristics or optimization algorithms improve network performance for different objectives like meeting coflows deadlines. They typically start with the assumption of improving network performance for a specific type of workload and consider different factors for designing a coflow scheduler, e.g., what is the data structure (triggered by data-programming models like MapReduce), complexity of network model, or constraints like routing data together with network schedulers.

The best coflow scheduler for a particular workload often depends on workload characteristics and network (or system) state. Therefore, the question is: *can machine learning help in developing a single coflow scheduler for different workloads (or data-parallel applications)?*

Fig.1 shows an initial design for such a coflow scheduler. It considers reinforcement learning on a graph neural network for learning workload characteristics. The design has three main modules:

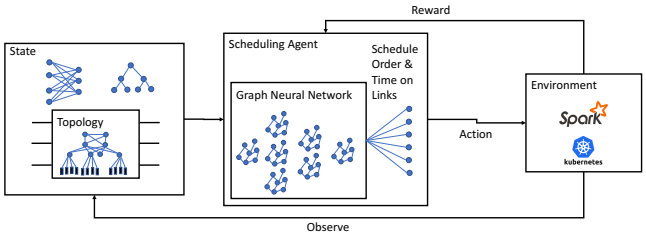

Figure 1: Design to learn scheduling policies through reinforcement learning and neural network

Reinforcement learning (RL) observed state:

- Given network resources (i.e., links) in cluster (i.e., environment in Fig.1)
- Learned structure of data communication (e.g., MapReduce or partition-aggregate) and

Scheduling agent:

- Derive features from above RL state information through graph neural network (GNN). A flat vector with all RL state information is not useful (both computation- and storage-wise) so GNN must *reduce the features vector*.
- The features vector should consider *reward*, which is based on an objective (e.g., meet coflow deadlines) and then produce the scheduling actions (i.e., order and time for coflows) for environment.

Environment:

- A data-processing system like Spark, e.g., running on bare metal or in a Kubernetes cluster.

Thesis Goals

The goal of this thesis is to develop a RL-based scheduling algorithm that can learn workload characteristic and derive scheduling decisions. The training of the algorithm can use production traces (e.g., from Facebook or Alibaba) or potentially use an actual testbed (to be discussed).

Milestones

- Familiarizing with reinforcement learning. [MT]
- Refine design for implementing coflow scheduler. [MT]
- Evaluate results through comparison with an existing heuristic for a specific objective, for instance, maximize coflows admission while meeting deadlines. [MT]

Required knowledge (or willing to learn)

- Understanding of networking concepts
- Good software development skills in Python
- Optionally, experience with a data-processing system e.g., Spark or Flink

Contact: Asif Hasnain, asif.hasnain@upb.de, Phone: 05251 / 60 1757, Room: O3.164