Formulating Big Data Throughput Optimization as a RL Problem

{mdhasibu, elvisdav, jacobgol} @buffalo.edu

April 4, 2022

1 Problem Formulation and Topic

Given a source endpoint e_s and destination endpoint e_d with a network link bandwidth b and network packet round trip time rtt; a dataset with a total size d_{all}, average file size f_{avg}, buffer size $buff_{size}$ and number of files n; the load from contending transfers (or background traffic- traffic that is using the same shared network) l_{ctd}; and set of application protocol parameters θ $= cc, p, pp$, the achieved throughput T for sharing data from source endpoint e_s and destination endpoint e_d can be defined as a function f:

$$T = f\left(b, rtt, f_{avg}, buff_{size}, n, cc, p, pp, l_{ctd}\right) \tag{1}$$

where Concurrency (cc) controls the number of server processes/threads, which can transfer various files concurrently. As a result, concurrency can accelerate the transfer throughput when many files need to be moved. Parallelism (p) is the number of data connections that each server process can open to transfer the different portions of the same file in parallel. It is a good option for large or medium file transfers to maximize transfer throughput. Pipelining (pp) is useful for small file transfers as it eliminates the delay imposed by the acknowledgment of the previous file before starting the following file transfer.

Due to the dynamic nature of several variables of equation (1) (i.e., available bandwidth b, rtt and l_{ctd}) and the objective is to maximize cumulative T over time of data transmission session, the target of this study is to formulate equation (1) as a RL problem. The RL agents task is to to maximize cumulative throughput by function approximation of equation 1 and finding optimal application protocol parameters $\theta = cc, p, pp$ for the duration of the transfer.

Table 1: Historical log example with application parameter and corresponding achieved throughput

entry No	Filesize	NumberOfFiles	RTT	BufferSize	Bandwidth	ThroughPut	CC	P	PP
1	100	250	10	200	10	5	1	2	2
2	100	200	8	150	15	3	1	1	1
3	50	150	15	250	20	4	1	2	1
4	40	150	20	225	5	1	1	2	2
5	150	225	15	150	8	5	2	3	3
6	100	250	10	200	10	8	2	3	3
7	100	200	8	150	15	10	3	4	4
8	50	150	15	250	20	8	3	1	4
9	40	150	20	225	5	4	3	2	3
10	150	225	15	150	8	4	2	1	3

Here in the example experimental historical set log which is collected as data point while transferring actual data over a shared network. The purpose of the collection is to solve equation 1. We have 5 different groups of key parameters and their corresponding throughput with different application parameter setting. For each group there are two logs; for example group key (100,250,10,200,10) as fields $f_{avg}, n, rtt, buf_{size}, b$ matches two log entries-log entry 1 &6 and group name is $G1$. Log entry 1 &6 achieves different throughput due to different $\theta = cc, p, pp$. **Here I am interested in using the historical logs as the environment and train an RL agent whose task is to maximize cumulative Throughput (so reward function should definitely relate to achieved throughput). As the transfer process will require some time to complete, the transfer time could be divided into multiple time steps and each of these transfer time step could be regarded as episode steps in RL problem. Below I will describe the state space, action space and reward function.**

1.1 Defining State Space

The state space S is a set of all the states that the agent can transition to and consist of group identification parameters $f_{avg}, n, rtt, buf_{size}, b$ along with $\theta = cc, p, pp$. The initial state (at the beginning of each episode) has $\theta = cc, p, pp$ =(0,0,0) along with other group identification parameters

$f_{avg}, n, rtt, buf_{size}, b$. As $f_{avg}, n, buf_{size}, b$ are constant in nature (b is link bandwidth which is usually constant during the transfer period), during the episode the agent will traverse state space by changing $\theta = cc, p, pp$ parameters and rtt. $\theta = cc, p, pp$ are control variables and rtt is an independent stochastic variable.

1.2 Defining Action Space

Action space is discrete and consist of the allowable combination of $\theta = cc, p, pp$ parameters. **One key takeaway is that the action space is also inclusive to state space**.

1.3 Reward function

After each step during the episode, agent will receive a reward function based on following:

$$R(t) = \begin{cases} -b & \text{if } T(t) < T(t-1) \\ T(t)/T_{Gmax} & \text{if } T(t) \geq T(t-1) \end{cases} \tag{2}$$

where b is a constant, $T(t)$ is throughput achieved after taking the action (i.e., changing $\theta = cc, p, pp$), $T(t-1)$ is throughput achieved in previous action step and T_{Gmax} is maximum throughput that is achievable for the state keys (i.e.,$f_{avg}, n, rtt, buf_{size}, b$) from the historical logs.

Defining Terminal Condition of a Episode The initial state (at the beginning of each episode) has $\theta = cc, p, pp$ =(0,0,0) along with other group identification parameters for an incoming transfer request $f_{avg}, n, rtt, buf_{size}, b$. The terminal condition of an episode could be defined as a finite number of steps in a episode (i.e., max step) or once the agent reaches the $T(t) = T_{Gmax}$ condition.

2 Objective

Our objective is to train, test and deploy an RL agent that optimizes data-transfer throughput given historical logs, data and network parameters. The key challenge being we need to maximize throughput while ensuring we do not break fairness in the end user's experience.

3 Related Work

The following references we have listed are the initial explorations into tackling the problem of application layer throughput optimization using RL. Reference [Sha21] is an overall exploration of different algorithms that can handle the problem of optimizing on such a stochastic environment. It did well in showing strong initial results using Dueling-DQN which is what the team plans to do. Reference [HYZ+17] is an exploration of how RL can be used to set the cache parameters, SDN policies and many other low level networking features on a simulated network in a big city. The goal being to solve the resource allocation problem, which happens when we try to construct networks for millions of people while ensuring everyone has stable and consistent access. This article uses Deep Q Learning but it seems to fall short in that it is running on pure simulation data. It would be more interesting if we could run this in real time and would show real benefit in minimizing energy consumption.

4 Technical Outline

This is covered above in 1.1 to 1.3

References

[HYZ+17] Ying He, F. Richard Yu, Nan Zhao, Victor C. M. Leung, and Hongxi Yin. Software-defined networks with mobile edge computing and caching for smart cities: A big data deep reinforcement learning approach. *Comm. Mag.*, 55(12):31–37, dec 2017.

[Sha21] Abhimanyu D. Shah. *Application Layer Optimization of Big Data Transfer Throughput using RL*. PhD thesis, 2021. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2021-08-06.