MINA: Auto-scale In-network Aggregation for Machine Learning Service

Shichen Dong[1], Zhixiong Niu[2], Mingchao Zhang[1], Zhiying Xu[1], Chuntao Hu[1], Wei Wang[1], Pengzhi Zhu[3], Qingchun Song[3], Lei Qu[2], Peng Cheng[2], Yongqiang Xiong[2], Chen Tian[1], Camtu Nguyen[1], Xiaoliang Wang[1]

[1]Nanjing University [2]Microsoft Research [3]NVIDIA

ACM Reference Format:
Shichen Dong[1], Zhixiong Niu[2], Mingchao Zhang[1], Zhiying Xu[1], Chuntao Hu[1], Wei Wang[1], Pengzhi Zhu[3], Qingchun Song[3], Lei Qu[2], Peng Cheng[2], Yongqiang Xiong[2], Chen Tian[1], Camtu Nguyen[1], Xiaoliang Wang[1] . 2023. MINA: Auto-scale In-network Aggregation for Machine Learning Service. In *7th Asia-Pacific Workshop on Networking (APNET 2023), June 29–30, 2023, Hong Kong, China.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3600061.3603276

1 INTRODUCTION

Distributed training on multi-rack clusters is used to support large machine learning models. Providing machine learning as a service (MLaaS) is becoming a new trend [4]. However, for the large-scale distributed ML platform, the communication is expensive and is where we are spending most of the time [2]. The technology of in-network aggregation (INA) has thus been applied to speed up the ML training in MLaaS clusters [3, 5]. INA offloads the aggregation to programmable switches in the network, effectively reduces the latency and bandwidth consumption.

Scalable Hierarchical Aggregation and Reduction Protocol (SHArP) has been used to speed up high performance computing (HPC) through commodity switches over InfiniBand (IB) networks [1] and obtains large performance enhancements. However, a 200Gbps HDR switch with SHArP enabled only supports a single ML job of a specified user at the same time due to the limited hardware memory. Cloud providers lack the estimation and understanding for multi-job and multi-tenant scenarios which is common in datacenters. To effectively support multiple jobs while avoiding conflict at switch, the allocation of computation resources of switches on multiple layer topology needs to be carefully determined.

We study, for the first time, the usage of INA capability of SHArP for supporting multiple ML jobs in multi-layer datacenter networks. Based on the distinct temporal and spatial characteristics of ML jobs, we design an efficient job allocation algorithm for coordinating the use of INA resources on switches. We estimate the capacity of the algorithm through simulation based on realistic traffic. The average number of accepted jobs increases by 52.7% and the job completion time (JCT) decreases by 4.09%.

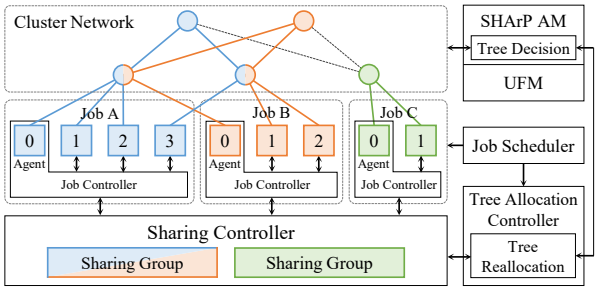

Figure 1: MINA architecture

2 SYSTEM DESIGN AND INA RESOURCE ALLOCATION

We introduce MINA to exploit the usage of SHArP in multi-job ML service in datacenter. Figure 1 illustrates the architecture of MINA, which consists of three primary components: the Tree-Allocation Controller, responsible for optimizing the aggregation tree in the space dimension; the Sharing Controller, responsible for managing resource sharing in the time dimension; and the Job Controller, responsible for controlling job start/stop and resource utilization of SHArP. These components operate on both the control and data planes to globally assign resources of SHArP and manage job execution on nodes to achieve best overall performance.

To carefully allocate of limited SHArP INA resources, our key idea is first allocating the switches of the cluster in coarse granularity among jobs to avoid resource conflicts at the spatial level, then determining the SHArP aggregation resources

APNET 2023, June 29–30, 2023, Hong Kong, China

ACM ISBN 979-8-4007-0782-7/23/06.
https://doi.org/10.1145/3600061.3603276

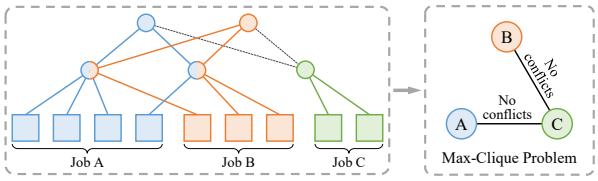

Figure 2: Aggregation trees and conflict graph

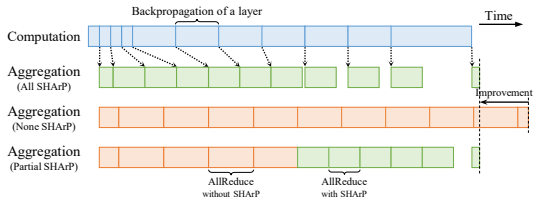

Figure 3: SHArP acceleration effectiveness

sharing with the fine granularity of AllReduce operations with the minimum communication overhead.

Space dimension allocation. Given the available SHArP-enabled switches, we first determine how to allocate jobs in the space dimension. The allocation aims to find an aggregation tree of switches for each job while minimizing the conflict of trees belonging to different jobs.

Time dimension sharing. After the space dimension allocation, there are still jobs that have to share the same SHArP switches. To maximize the benefit of SHArP when shared among multiple jobs, we carefully schedule when each job uses the INA resources in the time dimension.

2.1 Space Dimension Allocation Algorithm

The input of the algorithm includes the datacenter topology, ML jobs, and the corresponding computation nodes for each job. Note that if a switch has been allocated to other jobs, it can still forward packets to other switches.

To achieve an effective space scheduling, we need to solve the following two problems. The first problem is to generate aggregation trees for each ML job. Note that given the computation nodes of a ML job, we can find multiple aggregation trees in the topology for a single job. Due to the hierarchical nature of the topology, the high-level switches are shared by multiple jobs. Therefore, an intuitive solution is to generate switch trees that occupy fewer nodes and more lower-level nodes to improve the resource utilization.

The second problem is to find a maximum set of trees such that each pair of the trees does not conflict with each other. As shown in Figure 2, we build a conflicting graph by abstracting the aggregation trees into vertices, and draw edges between two vertices with no conflict. Therefore, the problem is transformed into finding *the maximum clique* in the graph. Since the maximum clique problem is NP-hard, we use an approximation algorithm by first obtaining multiple approximate-maximum cliques in the graph, then choosing a clique with the maximum earning. The earning of a clique is the sum of the JCT decrease of ML jobs.

2.2 Time Dimension Sharing Algorithm

The network pattern of training steps exhibits a great opportunity to share SHArP resources among multiple jobs. Figure 3 illustrates the computation and aggregation process of a training step. First, we can see that if we use SHArP for all AllReduce operations, we can reduce JCT in comparison with the aggregation without using SHArP, as shown in the second and third rows in Figure 3. One interesting observation is that *we can achieve the same amount of improvement by using SHArP for only parts of the AllReduce operations,* as shown in the second and fourth rows in Figure 3. It happens because JCT is determined by the last AllReduce operation. By accelerating the former AllReduce operations can not truly speed up the training step.

Based on the observation, we propose the fine-grained time sharing algorithm to effectively reduce the occupation of INA resources of one job (i.e. enlarge the chance of sharing among jobs) and achieve the best acceleration effect. Specifically, we partition the AllReduce operations within a training step into two portions, where the latter portion uses SHArP acceleration while the former one does not. We greedily make the latter portion as small as possible on the premise of keeping the best acceleration effect.

3 EVALUATION & FUTURE WORK

We first measure the overhead of our implementation of MINA on a testbed consisting of 5 SHArP switches and 4 hosts with an A100 on each interconnected with 100Gbps links. In the space dimension, releasing switch resources and reapplying them takes 125ms and 294ms on average respectively, allowing a seamless migration required in MLaaS. In the time dimension, switching between using and not using SHArP takes 33.1μs on average, which is negligible compared to the interval of the switching (~100ms).

We collect 14 traces of ML job requests in real scenarios, and use them as inputs of our scheduling algorithm to perform the large-scale simulation. Compared with the naive greedy allocation, our scheduling in the space dimension can increase the number of INA jobs by 35% on average (up to 67.7%) and decrease the total JCT by 3.52% on average (up to 6.9%). With the time dimension sharing enabled, the average number of INA jobs will increase by 52.7% (up to 85.8%), and the average JCT will decrease by 4.09% (up to 7.6%).

We conducted this study on 200Gbps HDR switches and are continuing our research on the 400Gbps NDR ones. It is notable that with the release of new devices which support multiple INA jobs, our study can easily adapt to the new

feature by allowing n jobs instead of two sharing the switch. Another potential optimization is integrating MINA into the job scheduler to achieve INA-aware job allocation.

REFERENCES

[1] Richard L. Graham, Devendar Bureddy, et al. 2016. Scalable Hierarchical Aggregation Protocol (SHArP): A Hardware Architecture for Efficient Data Reduction. In *2016 COMHPC*.

[2] Ronny Krashinsky, Olivier Giroux, et al. 2020. NVIDIA ampere architecture in-depth. *NVIDIA blog: https://devblogs. nvidia. com/nvidia-ampere-architecture-in-depth* (2020).

[3] ChonLam Lao, Yanfang Le, et al. 2021. ATP: In-network Aggregation for Multi-tenant Learning. In *18th USENIX NSDI*.

[4] Mauro Ribeiro, Katarina Grolinger, and Miriam AM Capretz. 2015. Mlaas: Machine learning as a service. In *IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE.

[5] Amedeo Sapio, Marco Canini, et al. 2021. Scaling distributed machine learning with in-network aggregation. In *18th USENIX NSDI*.