Accelerating Distributed Machine Learning in Disaggregated Architectures with Flexible Optically Interconnected Computing Resources

Shijia Yan, Ziyi Zhu, Madeleine S. Glick, Zhenguo Wu, and Keren Bergman

Department of Electrical Engineering, Columbia University, 500 W 120th St., New York, New York 10027

sy2629@columbia.edu

Abstract: We introduce an optically interconnected disaggregated architecture for GPU resources and demonstrate a 3× increase in GPU utilization and up to 73.2% acceleration of application runtime for distributed machine learning workloads. © 2022 The Author(s)

1. Introduction

Distributed machine learning applications have become a significant part of today's high performance computing and data center workloads[1]. When training a distributed machine learning model across multiple nodes due to system fragmentation in the conventional server-centric architecture, inter-node communication becomes more bottlenecked than intra-node communication due to both higher latency and lower link bandwidth. The problem becomes even worse when scaling to a larger number of nodes as the inter-node links will be further congested due to heavy inter-node communication. This results in significant resource under-utilization in conventional server-centric architectures[2]. Resource disaggregation provides a solid solution to this problem by providing a flexible allocation of fine-grained hardware into virtual nodes[3].Optical circuit switches (OCSs) provide high bandwidth, low cost solution to de-fragment and pool disaggregated resources with minimal latency overhead. In previous work, OCSs have been proposed to achieve resource disaggregation[2][4][5], however, the prior work was limited to a generic concept without system-level demonstration.

In this work we demonstrate rack-scale GPU disaggregation with OCSs and emulated aggregator switches, and show the capability to reconfigure resources to reduce communication overhead. Our proposed architecture leverages the flexibility of optical switches to increase hardware utilization and reduce application runtimes by alleviating system fragmentation. We build up our testbed using 4 commercial rack servers with 6 GPUs, 2 emulated aggregator switches and a MEMS switch to demonstrate resource disaggregation and defragamentation using application containers. Our testbed experimental results show 73.2% and 62.0% improved workload completion time with 3× increased GPU utilization in two test cases.

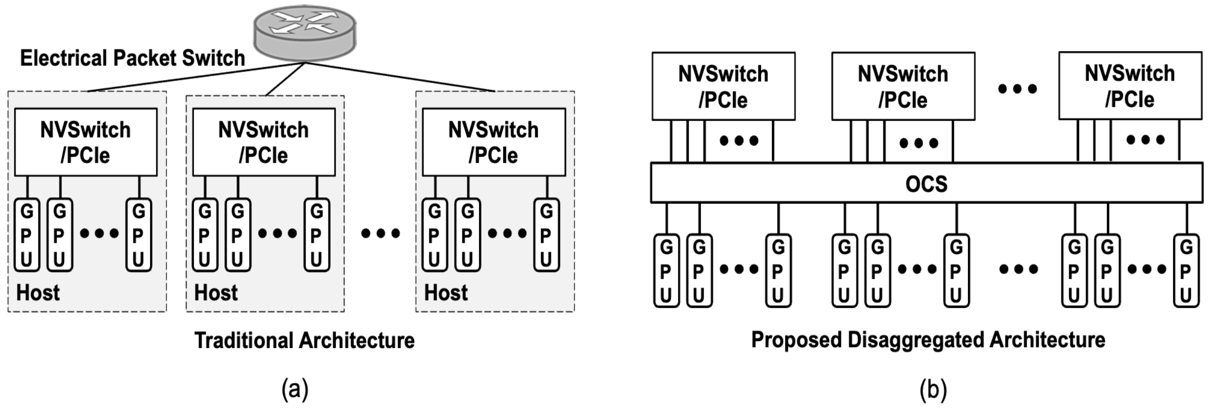


Fig. 1. (a) Traditional architecture for GPU clusters; (b) proposed system architecture for disaggregated GPU resources.

2. System Architecture

Figure 1 shows our proposed system architecture with Aggregator Switches (AS), Optical Circuit Switches (OCS) and a pool of disaggregated GPU resources. The aggregator switches are essentially packet switches that connect a group of heterogeneous resources. For example, an equivalent device in traditional architecture would be NVSwitch or PCIe Switch. Traditionally in distributed AI/ML applications, nonstop synchronization of large gradients implies an urgent need to allocating its GPU instances in a single machine[6]. Top-of-rack switches can act as aggregators to hold recomposed resources and to connect multiple types of devices. By reconfiguring the optical links between GPUs and our aggregator switch, disaggregated resources can be rearranged to reduce system fragmentation and increase job locality. As a result, the disaggregated system performance and efficiency is improved by the increased utilization and maximized traffic locality.

Accommodating OCSs in the network of disaggregated resources enables flexible connectivity of the network topology as needed by the application. For GPUs, different AI/ML applications have different requirements on the number of GPUs needed, varying from one to hundreds. Furthermore, this architecture can be adapted to provide flexibility to re-assemble heterogeneous computing resources for specific application needs, such as CPU, memory. In this case, the issue of low port counts of OCSs can be tackled by using multiple low-radix switches at the same time combined with other methods like locality-driven job scheduling.

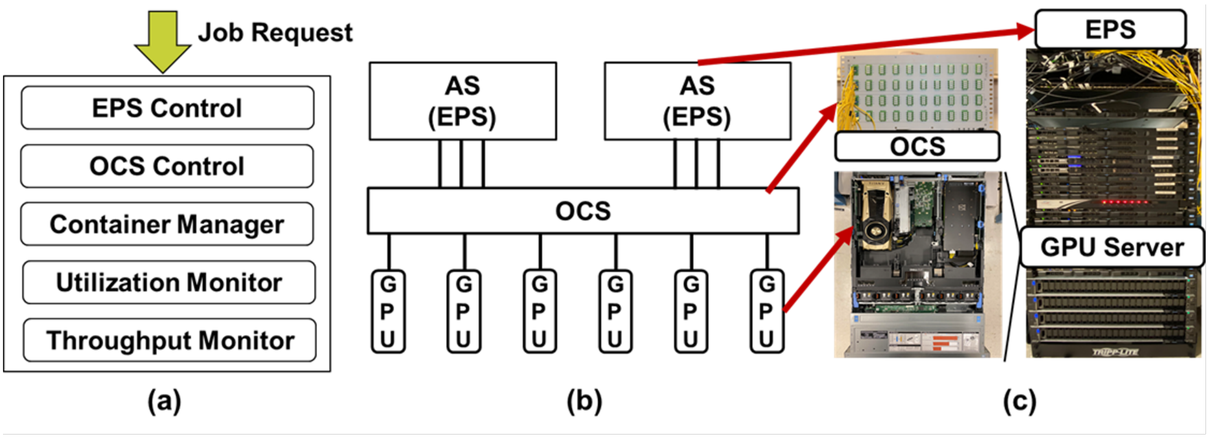


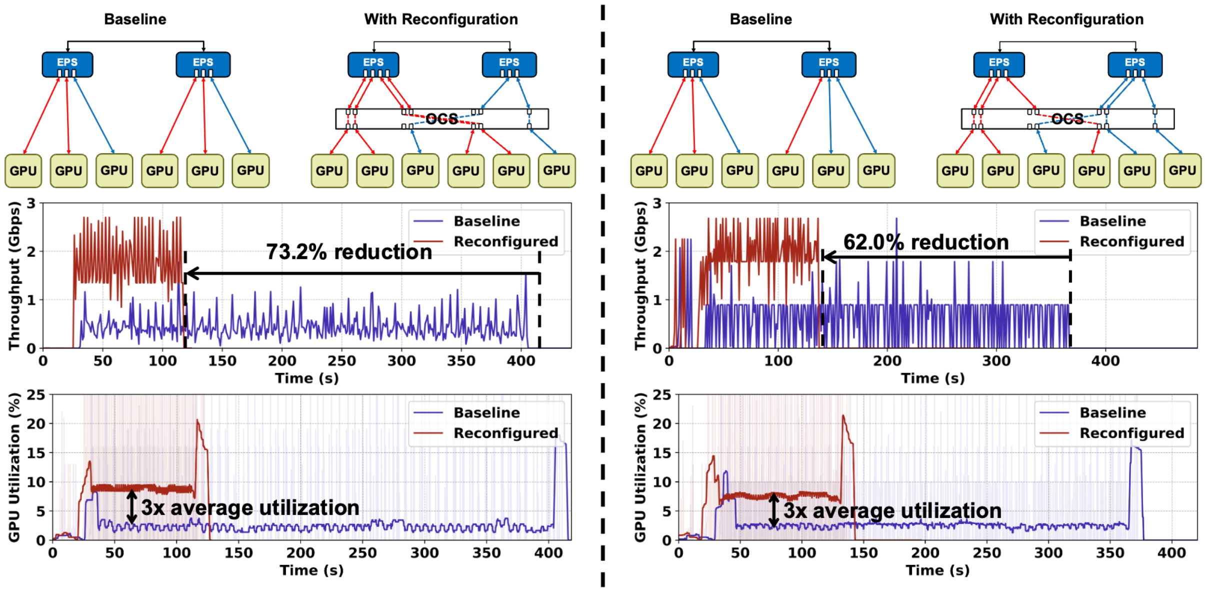
Fig. 2. (a) Network control plane, (b) testbed architecture and (c) hardware implementations using Calient MEMS switch, server rack with ToR switch and rack server.

3. Experimental Setup and Results

We build up our testbed with 6 NVIDIA GPUs installed into 4 commercial rack servers to demonstrate the capabilities of emulating deep disaggregation of GPUs and resource reconfiguration with OCSs. Each server is installed with a Tesla M40 GPU. In addition, two Titan V GPUs are inserted into the servers. All of them are connected with Mellanox ConnectX-4 NICs that enables RDMA over Converged Ethernet v2 (RoCEv2) for improved performance. In Fig.2 we show the network control plane for our testbed. We use a SDN controller to host management applications, including EPS flow table control, optical switch control, and container management through Ethernet.

When running an ML workload we reserve a certain number of GPUs and create one Docker container for each of them on their corresponding servers. On the servers, each container has one and only one GPU visible to it. The container's network I/O is then mapped to a specific NIC port on the physical machine with a unique IP address. By doing this we're able to (1) send gradient updates directly from one GPU to another without the participation of other resources; (2) assign specified IP address for each GPU resource in the network. This allows us to emulate and study these GPUs as they are deeply disaggregated. We use Tensorflow to build distributed machine learning applications to synchronously train a modified VGG16 neural network across multiple workers. Two different cases of ML workloads were studied on this topology. One in Fig. 3a shows two different jobs (red/blue) requiring 4 and 2 GPU resources for synchronous training, using ring all-reduce algorithm. The other case, as shown in Fig. 3b, shows two workloads with 3 and 3 GPUs required for synchronous training, using the same ring all-reduce algorithm. We obtained network I/O throughput for each Docker container using Ryu SDN monitoring program, and the GPU performance data is queried by nvidia-smi.

In case (a) the red workload spent 407s to complete when there is no reconfiguration, compared to 109s after regrouping all 4 GPUs under the same ToR switch. This yields a 73.2% reduction in completion time and a 3×



(a)

(b)

Fig. 3. **Top**: testbed topology for (a) Two synchronous ML workload using 4 and 2 GPUs for training, respectively (b) Two synchronous ML workload using 3 and 3 GPUs, respectively. Two jobs are ran simultaneously, represented by red/blue connections to its related GPU. Experimental results from the red workload is presented here as **middle**: network throughput and **bottom**: average GPU utilization over time. Finer-grained utilization data are shown as in background underlay.

utilization increase during training phase. In case (b) our workload of concern spent 368s before reconfiguration and 140s afterwards. Here we obtained 62.0% reduction in training time, while no significant improvement in resource utilization is observed. However, there is an expected increase in network throughput that indicates a higher effective utilization of the GPU after reconfiguration. Without the reconfiguration, resource fragmentation undermines the application performance of the workload running on top and interferes with other application flows. We find that OCSes can significantly help increase performance and hardware utilization in disaggregated systems running AI/ML workloads.

4. Conclusion

We demonstrated an optical disaggregated architecture with flexible resource connectivity for distributed machine learning applications. Our proposed control plane separates the GPUs within a single server and enables disaggregation using RoCEv2. Our results show 73.2% and 62.0% improved training time and 3× increase in GPU utilization for the machine learning workloads.

Acknowledgments. This work was supported by ARPA-E ENLITENED Program (project award DE-AR00000843) and the National Security Agency (NSA) Laboratory for Physical Sciences (LPS) Research Initiative (R3/NSA) (Contract FA8075-14-D-0002-0007, TAT 15-1158).

References

[1] Myeongjae Jeon et al. “Analysis of Large-Scale Multi-Tenant GPU Clusters for DNN Training Workloads”. In: *CoRR* abs/1901.05758 (2019).

[2] Madeleine Glick et al. “PINE: Photonic Integrated Networked Energy efficient datacenters (ENLITENED Program) [Invited]”. In: *Journal of Optical Communications and Networking* (2020).

[3] Keren Bergman et al. “PINE: An Energy Efficient Flexibly Interconnected Photonic Data Center Architecture for Extreme Scalability”. In: *2018 IEEE Optical Interconnects Conference (OI)*. 2018.

[4] John Shalf et al. “Photonic Memory Disaggregation in Datacenters”. In: *Photonics in Switching and Computing*. 2020.

[5] Peter X. Gao et al. “Network Requirements for Resource Disaggregation”. In: *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. 2016.

[6] Qizhen Weng et al. “MLaaS in the Wild: Workload Analysis and Scheduling in Large-Scale Heterogeneous GPU Clusters”. In: 2021.