On the benefits of Optical Disaggregated Datacentre Infrastructures for Machine Learning Applications

Albert Pagès[(1),*], Fernando Agraz[(1)], Salvatore Spadaro[(1)]

[(1)] Optical Communications Group (GCO), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
*albert.pages-cruz@upc.edu

Abstract *ML applications present time varying requirements that should be accounted for an optimized deployment. We evaluate the benefits of disaggregated DC infrastructures for supporting them and propose a dynamic re-orchestration strategy for improved resource usage. ©2023 The Authors*

Introduction

The popularity of Machine Learning (ML)/Artificial Intelligence (AI) applications to boost performance at all technological levels (service, control, management, etc.), has given rise to specialized DCs to host AI/ML functions [1]. Such applications have very intense requirements in terms of managed data volumes, storage space and CPU and GPU utilization [1, 2]. These requirements have a clear time varying behaviour as, for instance, transition from the learning phase, with high storage, CPU, GPU and memory requirements, to the inference phase, with lesser resource usage. Moreover, different types of ML algorithms have intrinsic requirements regarding resource usage [3]. All this heterogeneity makes the hosting of ML applications in DCs a dauting task.

Traditional DCs, based on integrated servers, fall short on accommodating these requirements, due to the fragmentation of the computational resources. To overcome these limitations, the disaggregated DCs (DDCs) paradigm has been proposed [4, 5]. It consists on separating the computational resources into independent modular blades, so physical computational infrastructures, tailored to the needs of the applications, can be composed. High capacity and low latency networks are a must for blade-to-blade communications, for which optical network technologies are envisioned. We analyse the benefits of DDCs to support ML applications, and present a resource re-orchestration strategy, exploiting the time behaviour of applications and the modularity of DDCs.

DDCs in support of ML applications

ML applications have significant and heterogeneous resource requirements. Broadly speaking, an ML application/pipeline can be divided into four different sub-applications [6]. First, a storage sub-application that stores the historical data and data inputs for the training of the ML as well as the real-time data for the inference of the model, which require large storage spaces. Then, an acceleration sub-application, which executes the training of the model and processes the data inputs. This requires high computational power, preferably in the form of GPUs. Third, a file system (FS) sub-application is responsible for storing the outputs of the ML model. Finally, a main sub-application is responsible for the coordination of all sub-processes as well as the data treatment from the data storage to the acceleration and from the acceleration to the FS. These sub-applications then may be allocated to different computational resources, providing the necessary network capacity across them.

In a traditional DC (Integrated), each of the sub-applications is deployed in a virtual machine (VM). To minimize the DC network (DCN) utilization, these VMs should be deployed into a single server. However, due to resource fragmentation, sub-applications may be forced to be spread over different servers/racks. This requires to plan extra physical computational resources. In scenarios with limited server units, this increases the chances of application blocking due to insufficient resources. The network usage may also increase, thus requiring more resources to be planned, or a higher connection blocking.

In a DDC, due to the modularity of blades, an ML application may benefit from being deployed as a single VM over a unique composed server, exploiting resources available over multiple blades [7]. Then, the computational resource saving is two-fold: first, resource fragmentation is minimized, so ML applications are consuming the strictly requested resources, lowering the hardware that needs to be planned; second, since all the functionalities of the application are deployed as a single VM, there are some computational resources that can be shared across sub-applications, such as CPU cores, further reducing the resource footprint. However, resource disaggregation requires blade-to-blade interconnection, namely between employed CPU blades and the rest. This may increase the required network capacity at the DCN.

We consider here a DDC in which resources

are split in racks, with each rack having a set of blades of each type of computational resource, namely, CPUs, GPUs, memory (MEM), hard disk drive (HDD) and solid-state drive (SSD) (Fig. 1). We assume two types of storage since different types of ML applications have heterogeneous needs on data volume and access rate. All the blades have interfaces that allow them to connect to an optoelectronic top-of-the-rack (ToR) switch that performs traffic aggregation. Due to the different capacity requirements posed by the communication with HDD blades, which require a less stringent performance, ToRs interconnect with each other via either an optically or electrically switched DCN; blades have interfaces towards the optical one (CPU, GPU, MEM, SDD) or the electrical one (CPU, HDD).

We employ the following strategy for initial application provisioning. The joint resource requirement of all sub-applications is calculated. Then, for each type of required resource, blades with free capacity are found. A blade consolidation strategy is used to minimize the number of blades, which impacts on the number of computational resources to be planned and the network connections between them. For blade interconnection, the connection between CPU and GPU, MEM and SSD exploits the optical DCN. For the interconnection of CPU and HDD blades, the electronic DCN fabric is used.

In addition, to further reduce the resource footprint of ML applications, the modularity of DDCs is exploited, which allows to down-scale the allocated resources when applications transition to inference. In this phase, applications are reallocated following the same stated principles as before, considering the total amount of resources required for the inference of the trained models, which pose much lower computational and networking requirements.

Performance evaluation

We analyse here the benefits of the modularity of DDCs in terms of infrastructure resources to be planned. The integrated DCs case is used for benchmarking. We also analyse the benefits of dynamic resource re-orchestration. A DDC with constrained capacity of computational resources

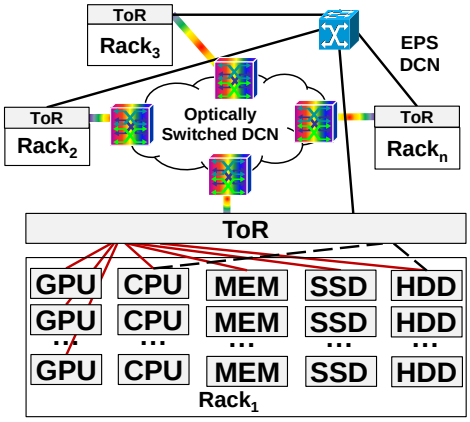

Fig. 1: DDC infrastructure with multi-technology DCN.

and unbounded networking ones is assumed. This allows to analyse the amount of network resources needed to exploit at maximum the computational ones.

We consider a DDC cluster of 8 racks, each hosting 10 GPU blades at 10 GPUs, 20 CPU blades at 20 cores, 20 MEM blades at 128 GB, 10 HDD blades at 12 TB and 20 SSD blades at 6 TB. For the DCN, we consider the afore-mentioned multi-technology DCN, unbounded in terms of blade interface capacity and number of optical ports at ToRs, although a 400 Gb/s capacity per port is considered. For the integrated DC case, the equivalent number of computational resources is hosted per rack in the form of CPU and GPU servers, namely, 10 GPU servers with (CPU, GPU, MEM, SSD) = (10, 20, 128 GB, 6 TB) and 10 CPU servers with (CPU, MEM, HDD, SSD) = (10, 128 GB, 12 TB, 6 TB). All servers are connected to the ToR, which are then interconnected by an optical DCN. Optical ports at ToRs work here at 40 Gb/s to account for the difference in network requirements compared to blade-to-blade communications.

We consider an exponentially distributed arrival rate (λ) of ML applications. Their duration follows a uniform distribution for the two phases, training and inference, with HT denoting the average application duration. We also consider three types of ML applications in equal share, reinforcement learning (RL), unsupervised (Un.) and semi-supervised (Semi.), which impose different resource requirements [3, 4]. Tab. 1 depicts the resource requirements for the training

Tab. 1: Resource profile of the considered ML applications

Type	Resource profile (CPU cores, GPUs, MEM (GB), Storage (GB))							
	Storage sub-app		Main sub-app		Acc. sub-app		FS sub-app	
	Training	Inference	Training	Inference	Training	Inference	Training	Inference
RL	1-2, 0, 2-4, 500-1000	1-2, 0, 2-4, 5-10	2-4, 0, 2-4, 20-40	1-2, 0, 2-4, 20-40	1-2, 1-3, 2-8, 20-40	0, 0, 0, 0	1-2, 0, 1-2, 100-500	1-2, 0, 1-2, 5-10
Un.	1-2, 0, 4-8, 2000-4000	1-2, 0, 2-4, 20-40	2-4, 0, 2-4, 20-40	1-2, 0, 2-4, 20-40	2-4, 3-5, 2-8, 20-40	0, 0, 0, 0	1-2, 0, 2-4, 1000-2000	1-2, 0, 1-2, 10-20
Semi.	1-2, 0, 8-16, 3000-6000	1-2, 0, 2-4, 30-60	2-4, 0, 2-4, 20-40	1-2, 0, 2-4, 20-40	4-6, 5-7, 2-8, 20-40	0, 0, 0, 0	1-2, 0, 2-4, 1000-2000	1-2, 0, 1-2, 10-20

and inference phases. We assume that the storage requirements of RL applications can be fulfilled with HDDs, while SSD is used for the rest. We consider that sub-applications need to communicate at 1 Gb/s and 10 Mb/s for training and inference phases, respectively, in regards to storage transfers. For the inter-blade communication 100 Gb/s are required between CPU and GPU/MEM.

We start by depicting the percentage of computational resources required to support the applications with respect to the total considered computational capacity. As a representative resource, we focus on the GPUs. Fig. 2 depicts the obtained results (columns), considering 10^5 application arrivals per data point for increasing values of λ*HT. First, it can be seen how disaggregated infrastructures allow for significant reductions on the resources that need to be planned (up to halve) when compared to integrated ones. Then, thanks to leveraging on the modularity of DDCs, the proposed re-orchestration strategy is also evaluated (wR). It allows to even further reduce the resources required, thanks to re-provisioning the resources mapped to ML applications to match the current resource requirements during inference. We also depict the acceptance rate of applications (lines), considering that provisioning rejections only happen due to lack of computational resources. It can be appreciated how the acceptance rate in integrated infrastructures is lower, as it rapidly reaches the usage of the full capacity of computational resources. This is due to resource fragmentation, which does not allow to fully exploit all the planned capacity. On the other hand, DDCs increase the acceptance rate for almost all considered working points, special in the wR case, as it significantly reduces the number of wasted computational resources.

Having demonstrated the benefits of DDCs, to complement the study, we also analyse the network resources required to achieve the aforementioned performance. In Fig. 3 (columns), we measure for all types of blades the average required interface capacity for two representative loads, with CPU-E and -O referring to the interfaces of CPU blades connected to the electrical or optical DCN. It can be seen how a significant blade interface capacity is required, being the CPU, GPU and MEM the ones that require most. Considering standardized interface capacities, up to 400 Gb/s are required for lower loads while these rise up to 1 Tb/s for higher loads. These requirements are lowered by around a 50% factor if re-orchestration is employed, thanks to the lower number of blades communicating with each other. Fig. 3 (lines) also

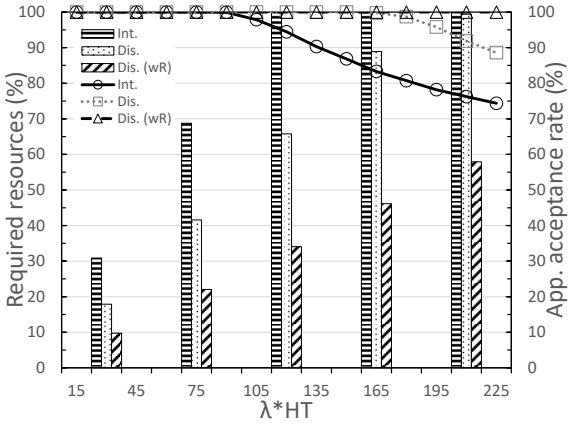

Fig. 2: Required resources and application acceptance rate.

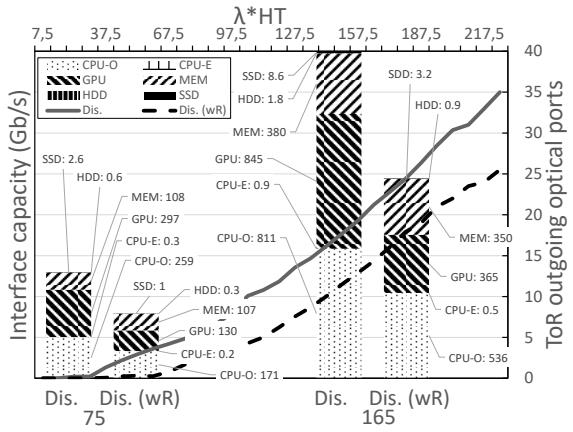

Fig. 3: Required interface capacity and optical ports.

depicts for both DDC scenarios the number of required outgoing optical ports from ToRs to the optical DCN. Same conclusions hold here, the wR case allows to save resources thanks to the dynamic re-composition of ML applications. For completeness, let us mention that for the integrated case, server interface capacities of up to 6 Gb/s are required. This highlights that, although DDCs are beneficial in terms of computational resource usage, they impose a significant burden in the network, which is mitigated exploiting the time characteristics of applications and re-orchestration operations.

Conclusions

ML applications impose a significant burden on DC resources, both computational and networking. We have shown how their allocation can benefit from the modularity of the disaggregation paradigm and associated re-orchestration techniques, exploiting the applications particular time behaviour, dramatically reducing the computational resource capacity to be planned at the infrastructure at the expenses of a potential higher network burden.

Acknowledgements

This publication is supported by the project TRAINER-B (PID2020-118011GB-C22) funded by MCIN/AEI/10.13039/501100011033.

References

[1] K. Hazelwood, S. Bird, D. Brooks, S. Chintala, U. Diril, D. Dzhulgakov, M. Fawzy, B. Jia, Y. Jia, A. Kalro, J. Law, K. Lee, J. Lu, P. Noordhuis, M. Smelyanskiy, L. Xiong, X. Wang, "Applied Machine Learning at Facebook: A Datacenter Infrastructure Perspective", *Proceedings of 2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Vienna (Austria), February 2018, DOI: 10.1109/HPCA.2018.00059

[2] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, J.-R. Wen, "A Survey of Large Language Models", *arXiv*, March 2023, DOI: arXiv:2303.18223

[3] S. V. Mahadevkar, B. Khemani, S. Patil, K. Kotecha, D. R. Vora, A. Abraham, L. A. Gabralla, "A Review on Machine Learning Styles in Computer Vision - Techniques and Future Directions", *IEEE Access*, vol. 10, pp. 107293-107329, September 2022, DOI: 10.1109/ACCESS.2022.3209825

[4] X. Guo, X. Xue, F. Yan, B. Pan, G. Exarchakos, N. Calabretta, "DACON: a reconfigurable application-centric optical network for disaggregated data center infrastructures [Invited]", *Journal of Optical Communications and Networking*, vol. 14, no. 1, pp. A69-A80, January 2022, DOI: 10.1364/JOCN.438950

[5] O. O. Ajibola, T. E. H. El-Gorashi, J. M. H. Elmirghani, "Network Topologies for Composable Data Centers", *IEEE Access*, vol. 9, pp. 120955-120984, August 2021, DOI: 10.1109/ACCESS.2021.3106375

[6] A. Barrak, F. Petrillo, F. Jaafar, "Serverless on Machine Learning: A Systematic Mapping Study", *IEEE Access*, vol. 10, pp. 99337-99352, September 2022, DOI: 10.1109/ACCESS.2022.3206366

[7] A. Pagès, F. Agraz, S. Spadaro, "On the complexity of configuration and orchestration for enabling disaggregated server provisioning in optical composable data centers", *Journal of Optical Communications and Networking*, vol. 14, no. 12, pp. 998-1009, December 2022, DOI: 10.1364/JOCN.471937