The Future of AI/ML Innovation is Row-Scale Disaggregation

Matthew Williams
Chief Technology Office (CTO)
Cerio
Ottawa, Ontario, Canada
matt@cerio.io

Ryan E. Grant
Electrical and Computer Engineering Department
Queen's University
Kingston, Ontario, Canada
ryan.grant@queensu.ca

***Abstract*—Traditional system architectures were never designed for the mix of application tasks that are now demanded of them. Composable Disaggregated Infrastructure (CDI) introduces a new distributed system architecture for the dynamic compute systems that will train future AI/ML models and provide large-scale inference for organizations worldwide. By utilizing a high-performance, multi-path network fabric design, we can extend PCIe beyond the compute node – and the compute rack – to provide configurable, efficient row-scale computing solutions for the data centers of tomorrow.**

I. Introduction

While Artificial Intelligence (AI) and Machine Learning (ML) have made great strides in computing outcomes transforming every industry, AI/ML adoption has introduced new challenges in system efficiency. AI/ML has brought large scale "capability" computing to the masses, an area traditionally confined to scientific applications with highly specialized users. Capability computing concentrates on solving limited numbers of very large problems. This contrasts with "capacity" computing that focuses on solving large numbers of simple problems. The difference in training a Large Language Model (LLM) and completing millions of web searches outlines the challenges of capability computing over capacity computing.

System models have been designed for traditional capacity computing, assuming that many jobs can be assigned to a single compute "node/server" in a system. Capability computing requires rethinking architectures as multiple nodes are instead assigned to a single problem. Using multiple nodes to solve a problem creates new challenges. Reserving compute nodes for large problems dedicates those resources to the task. CPUs are reserved alongside GPUs for the entire job execution. This results in inefficiencies in the overall system as resources are underutilized during job execution. The fact that the vast majority of codes do not use CPUs and GPUs simultaneously for computing further extends this inefficiency.

The current state-of-the-art in capability computing systems rely on "homogeneous-heterogeneous" systems. The node architecture that is homogeneous in the large system is CPU-based, with accelerators (typically GPUs) providing the heterogeneous nature of the system. As the economics of specialized accelerator hardware are flipping to favor specialization [1], this places stress on this traditional system model. Adding specialized accelerators does not need to be universal to all of the compute nodes, but instead many kinds of accelerators are now needed.

Therefore, the current node architecture of our large capability-class systems is producing inefficiencies that create idle hardware, hardware trapped by reservations that are unused, and inefficiencies in the data center in powering and cooling the underutilized resources. Many of these inefficiencies are difficult to remove in the current node architecture model as hardware becomes physically "trapped" in nodes without being fully utilized, but we cannot assign other tasks to that hardware without slowing down the main critical task, such as training an LLM.

II. A New Row-Scale System Architecture

Existing approaches can help address the problem of inefficient hardware allocation and "trapped" hardware. Disaggregated system solutions such as PCIe extension technologies allow us to build rack-level hardware sharing. Unfortunately, PCIe does not scale well past the rack level. The non-lossy nature of PCIe communication combined with PCIe root complex architecture and backwards compatibility mean that it is not designed for large scale hardware communication. Growing a system too large with PCIe leads to significant performance costs (e.g. when recovering from losing PCIe data on the wire).

While PCIe can help solve the efficiency problem within the rack, AI/ML, large LLMs and the growing use of massive engineering simulation models require solutions that solve the problem at a larger scale. Composable Disaggregated Infrastructure (CDI), scaling to an entire row or compute racks (a few dozen racks per row), can maximize efficiency and utilization while enabling us to build systems that would otherwise be impossible with traditional server hardware.

III. Flexibility

CDI takes the approach that the PCIe bus can be extended over a high-performance network fabric that provides end-to-end reliability and excellent latency and bandwidth. With these capabilities, systems can be composed at job runtime with large numbers of accelerators that would be impossible to install in a typical system with physical limits to the number of PCIe slots that can be engineered into them. In addition, CDI allows us to reconfigure accelerators on the fly, releasing or adding hardware based on the workload. There are challenges to this allocation of hardware to ensure that resources are available when needed, but CDI makes it possible and practical

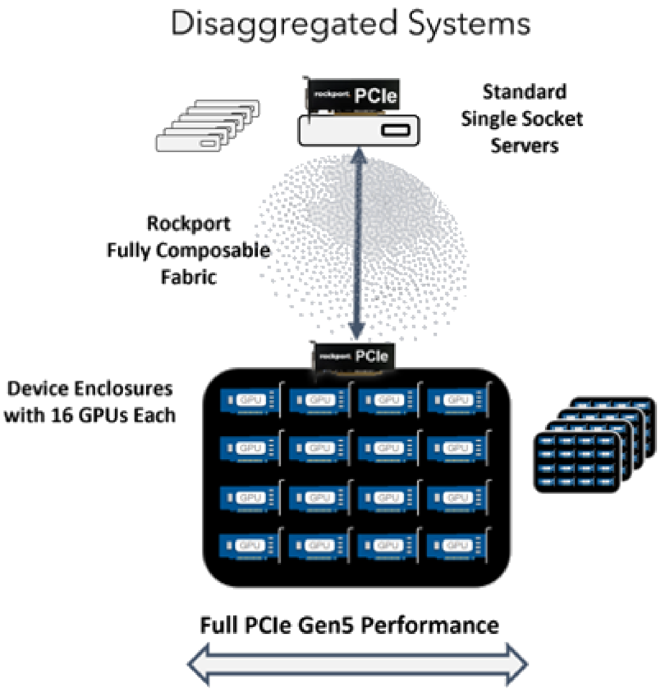

Fig. 1: A disaggregated row-scale fabric and chassis example

to design a scheduling system that makes dynamic use of system hardware at row-scale.

CDI utilizes a high-performance network fabric to provide the linkage between a compute server with a CPU and the accelerators/GPUs that it wishes to use. (See Figure 1.) A single CPU server can have dozens of GPUs attached to it at a time. These GPUs are not located in another compute server, but instead are located in PCIe chassis placed throughout the system. These chassis hold many GPUs at a time and can be connected to servers on an on-demand basis. Once attached through the network with the compute server, GPUs appear as if they are local to the server. However, they share a remote PCIe root complex solution that allows them to communicate with each other without having to flow messages back through the attached compute server.

IV. Physical Proximity Yields Performance

The benefits to row-scale disaggregation extend beyond system flexibility and efficient utilization of hardware. There is a performance advantage to placing accelerator hardware physically close to each other. By designing and placing PCIe devices in chassis, communications between large numbers of GPUs are now local. This is important as up to 70% of the time of AI/ML tasks can be consumed in GPU communication. The type of communication used is typically “collective” communication, where all of the software processes in a job must communicate with each other. These normally take the form of large reductions of data and distribution of the result to all job participants. These types of collectives are called “allreduce” collectives, although many other types are popular and in common use. They all share the same characteristics of all processes needing to participate in a data exchange.

By relocating GPUs into a single chassis, the time to communicate is significantly reduced. In current systems, these communications are limited to 4 up to a maximum of 8 GPUs locally in a system and then need to use the system network to communicate between compute nodes and with other GPUs. The PCIe chassis increases this number of local GPUs by 4X-8X, increasing overall communication efficiency.

Having multiple paths through the high-performance network fabric gives us the flexibility to connect multiple chassis and servers without worrying about interfering cross-traffic on the network. We can use dedicated paths between the chassis and compute nodes that can lead to performance predictability as network congestion from other workloads or other chassis in the same workloads can be avoided.

Of course, CDI is not free. There is a performance cost to moving accelerator hardware out of the compute node into a chassis, but it’s relatively minimal. Studies show the cost of moving commands and data over the network to a GPU is typically <1% [2]. This is easily mitigated by tighter integration of the GPUs themselves into large locally connected groups. Additionally, CDI allows jobs to start faster, as they are no longer waiting for specialized servers. The economic advantages of CDI also make a larger number of GPUs available within the same budget, allowing more GPUs to be applied to each job, further accelerating job completion time.

V. Achieving Economies of Scale

CDI has ramifications outside of system efficiency and data center operating costs. The economics of the traditional homogeneous-heterogeneous system require buying servers to contain all of the PCIe slots for the accelerators needed. To add new accelerators, you also need to purchase CPUs, memory and (potentially) local storage to support that accelerator purchase. This significantly increases the cost of adding accelerator technologies to a data center. With a CDI approach, adding new accelerators means adding comparatively inexpensive PCIe chassis and managing server/chassis balance in system design. It also makes custom accelerators more attractive to include in a system. As the purchase cost of a new accelerator technology is mostly confined to the accelerator itself, data centers can add capacity as needed and not worry about keeping systems powered and cooled for accelerators that are not 100% utilized. This changes the economics of the specialized accelerator market, further pushing down adoption costs and allowing easy adoption of small numbers of specialized hardware units.

References

[1] N. C. Thompson and S. Spanuth, “The decline of computers as a general purpose technology,” *Communications of the ACM*, vol. 64, no. 3, pp. 64–72, 2021.

[2] T. Groves, C. Daley, R. Gayatri, H. A. Nam, N. Ding, L. Oliker, N. J. Wright, and S. Williams, “A methodology for evaluating tightly-integrated and disaggregated accelerated architectures,” in *2022 IEEE/ACM International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*. IEEE, 2022, pp. 71–81.