GIGAIO Technical Paper

The GigaIO™ FabreX™ Memory Fabric

FabreX™ Composability

CONTENTS

FIGURES

This document is for informational purposes only and may contain typographical errors and technical inaccuracies. The content is provided as is, without express or implied warranties of any kind.

June 2019 | Rev 1.0

General

In today’s Hyper scale computing environment, it is becoming increasingly imperative capital spending and power consumption has to be reined in from the perspective of the entities involved in its deployment. These are issues of paramount importance that need to be tackled to bolster the bottom line as well as project an honest image of being environmentally friendly.

Composability in Rack scale architecture is fast becoming a requirement for the Chief Information Officers (CIO) to consider on architecting their datacenters.

Introduction

Some aspects of composability in the case of certain infrastructure has existed for some time in computing environments. However, it has assumed a whole new dimension with the advent of Software as a Service (SaaS) phenomenon.

This entails the infrastructure composed of resources within a Rack be able to be reconfigured to address a particular application or set of applications to be run on servers comprising a Rack. This implies the hardware resources required to run a particular application be configured to execute in an optimal manner. Upon the completion of running a particular job comes the necessity to reconfigure the infrastructure for running the next application in an optimal manner. IT professionals view this to be a nightmarish scenario because it demands painstakingly meticulous care for reconnecting plethora of cables on the port side of hardware resources in the Rack. This has users scrambling for solutions to the Composability issue.

Current Approach

Figure 1 depicts an example of a Rack scale system made up of a variety of hardware resources. The Compute Servers are not restricted to x86 based platforms and alternately can be FPGA, ASICs with embedded SoC, and a variety of other Computing elements.

The cluster of GPUs in a traditional PCIe Expansion chassis is called JBoG.

Similarly, a cluster of NVMe drives in a traditional PCI Express (PCIe) Expansion is called JBoF.

Figure 1 shows three sets of servers with each connected to a JBoG and JBoF respectively. It also shows GPU card and NVMe Drive resident in every server installed in its backplane. These servers are also connected by a NIC card for Ethernet or HBA for InfiniBand serving as interconnect within the Rack. For discussion purposes these are generically termed Communication Controller.

PCIe being the prevalent and ubiquitous I/O signaling protocol used in servers constituting its backplane is used to connect the JBoG and JBoF appliances.

A point to note is these servers in the Rack act as autonomous islands of compute ecosystems communicating with each other via the communication link.

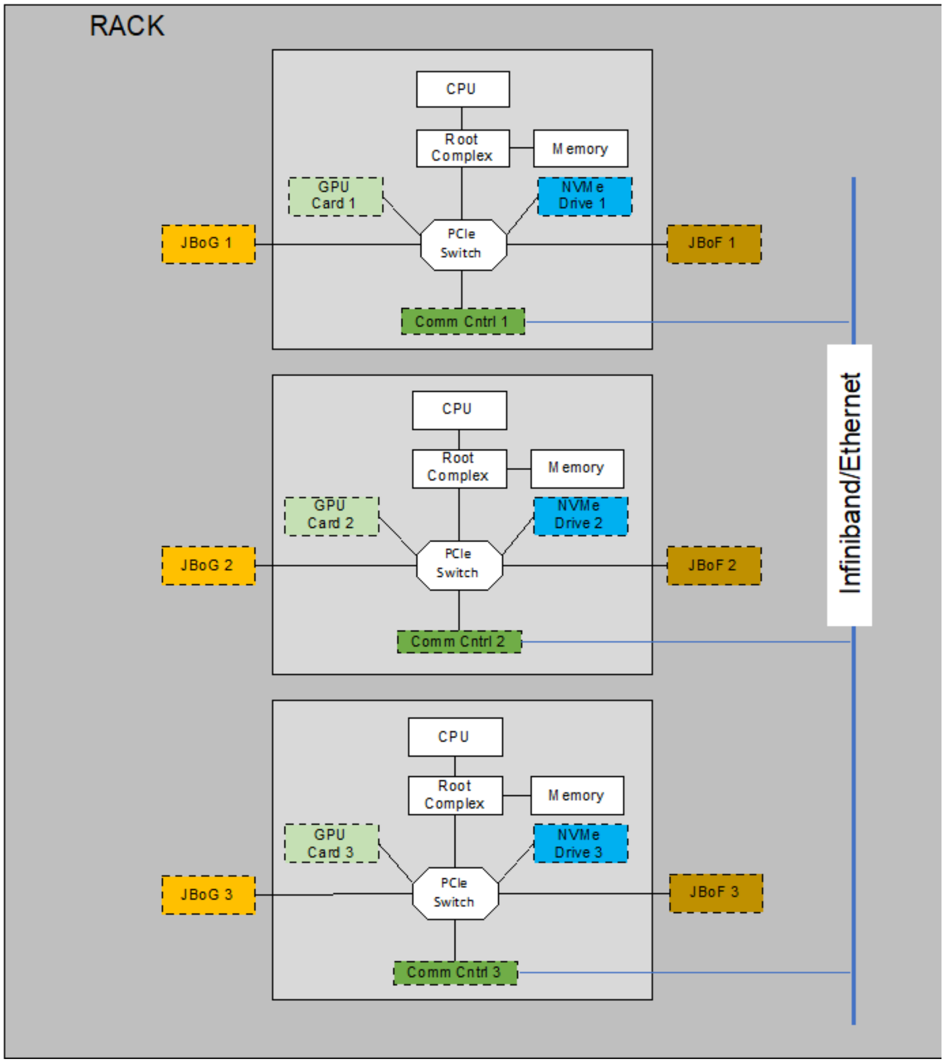

FIGURE 1 – EXAMPLE RACK SCALE SERVER

Temporary Solution

Figure 2 shows how a Top of Rack (ToR) Switch installed in the Rack is used to connect to some of the peripheral devices exclusively through PCIe Switching embedded in the ToR.

With this configuration the JBoGs and JBoFs associated with the servers as shown in Figure 1 can be under the domain of any one of the three servers shown. Figure 2 is an example of how these resources are assigned to a server pretty much at will. This configuration can be performed under software control with the help of some level of intelligence residing in the ToR.

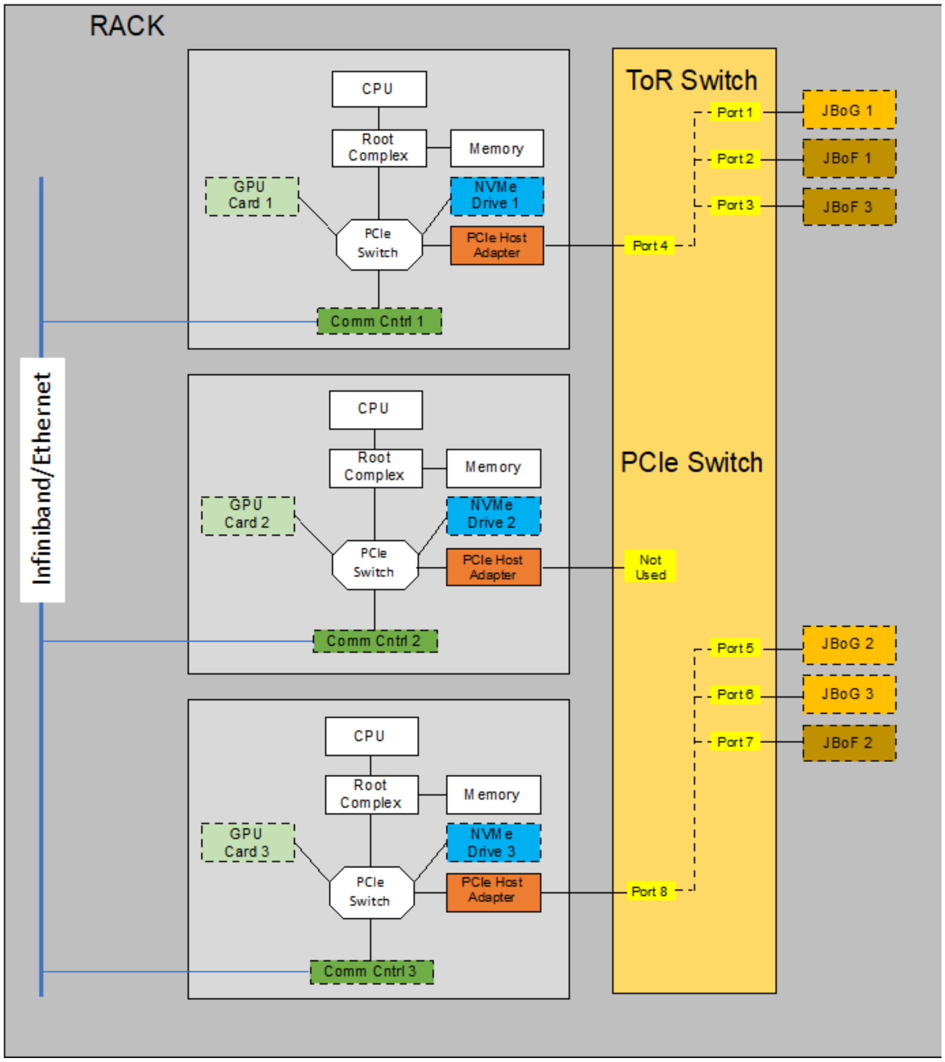

FIGURE 2 – TOP OF RACK SOLUTION

The first drawback with this approach is that the GPU and NVMe PCIe I/O cards within the servers cannot be configured to be under the domain of any other server besides the one they already reside in.

The second drawback is that the JBoG and the JBoF will act as one resource I/O box. This implies all of the individual GPU cards and NVMe Drives comprising these resource boxes have to be all assigned to one server thereby strictly restricting the resolution of composability factor.

The third drawback is the servers still need to communicate with each other over the Ethernet and/or InfiniBand communication links as shown and not be able to use the ToR for this purpose.

FabreX in its minimum configuration, like other similar solutions in the marketplace, support all of the features just described.

Ideal Solution

The ideal solution for composability will be to transform the JBoGs and JBoFs into servers. This entails having Root Complex Processor (RCP) in the I/O resource boxes. The primary function of the RCP is to enumerate the PCIe I/O devices in their respective boxes and carve out unique Memory Mapped IO (MMIO) windows for the individual PCIe I/O devices and publish this to the FabreX Switch controller. This is primarily where the role of RCP ends for all practical purposes.

This is shown in **Figure 3**.

FabreX assigns a unique Virtual Memory window in its 64-bit Virtual Global Address Space and enables communication of all servers amongst each other and with all of the PCIe I/O devices including PCIe I/O device resident in individual servers.

This configuration also allows for communication between PCIe I/O devices as for instance in the case of GPUs communicating with each other using GDR. GDR and NVMe-oF communication paths imply DMA operations.

The physical connections shown are the attachment to FabreX switch via a cable with PCIe signaling. The virtual connections are logical attachments to FabreX via the respective physical connection as shown.

This configuration also allows for server to server communications.

Figure 4 shows the concept of mapping all of the resources attached to FabreX within its 64-bit Global Virtual Address Space. It shows the path of the server CPUs directly communicating with all of the PCIe I/O devices.

It also shows DMA paths between and amongst all GPUs and also the NVMe DMA paths to the server’s system memories. Data transfer take place at PCIe native speeds.

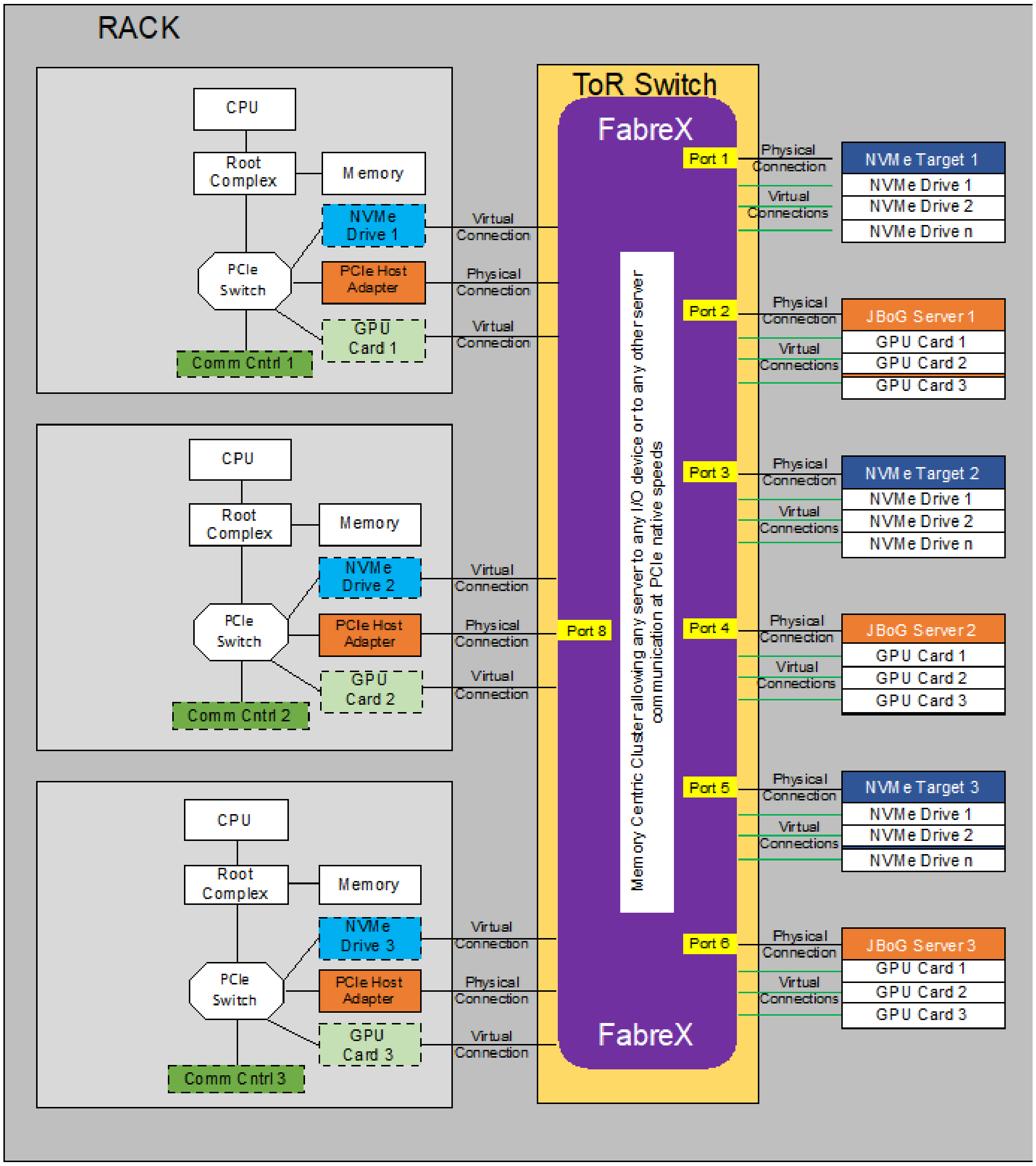

Figure 3 – FabreX Virtual Mapping Windows

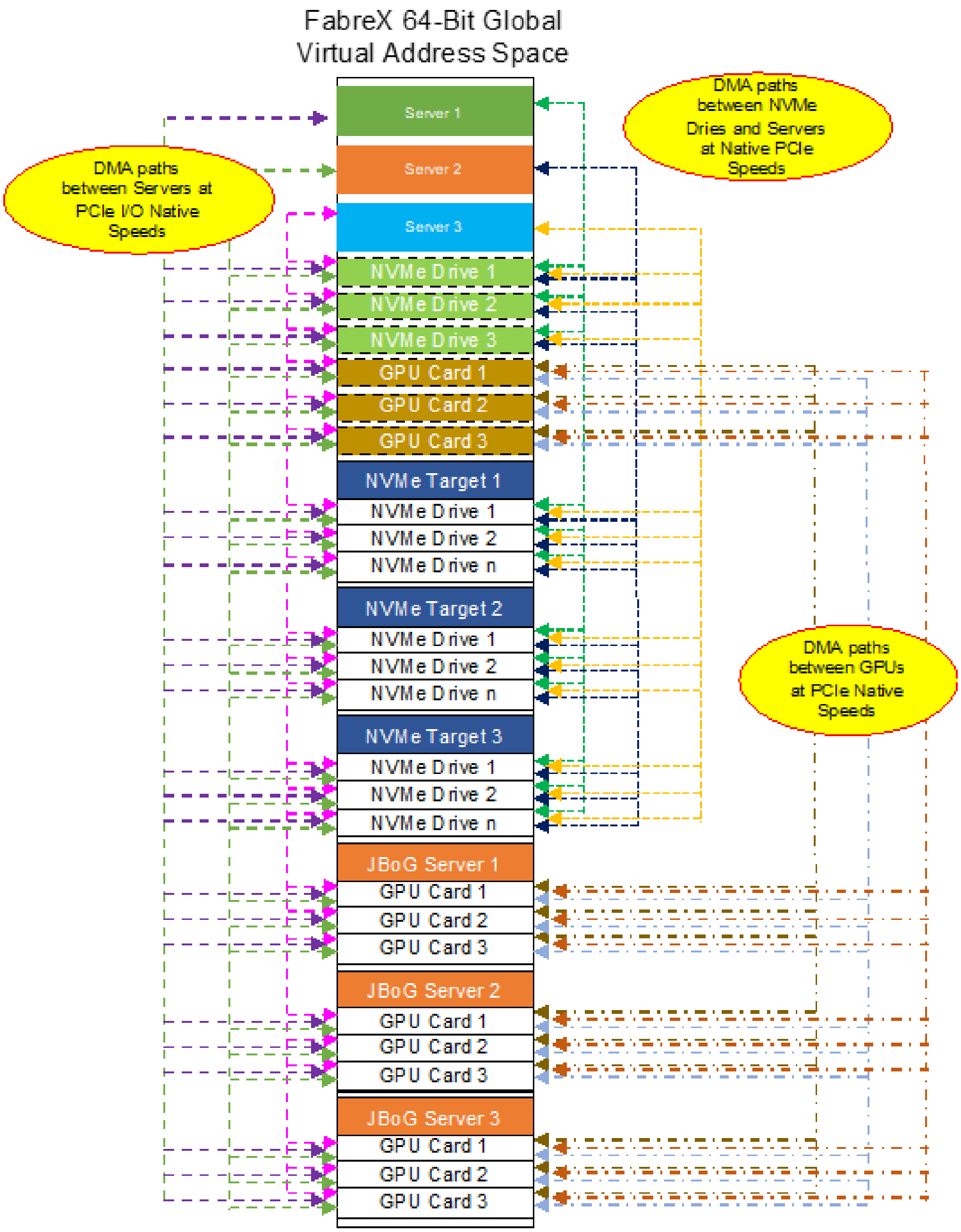

FIGURE 4 – MAPPING ALL RESOURCES TO 64-BIT GLOBAL ADDRESS SPACE

About GigaIO

Headquartered in Carlsbad, California, GigaIO democratizes AI and HPC architectures by delivering the elasticity of the cloud at a fraction of the TCO (Total Cost of Ownership). With its universal dynamic infrastructure fabric, FabreX™, and its innovative open architecture using industry-standard PCI Express/soon CXL technology, GigaIO breaks the constraints of the server box, liberating resources to shorten time to results. Data centers can scale up or scale out the performance of their systems, enabling their existing investment to flex as workloads and business change over time. For more information, contact info@gigaio.com or visit www.gigaio.com. Follow GigaIO on Twitter and LinkedIn.