bitfusion | Elastic GPUs Shared Pools

Challenge

Neural networks created a new compute infrastructure landscape. As CPUs are no longer sufficient, purpose-built machine learning hardware based on GPUs, FPGAs and ASICs are being deployed in growing volume to provide accelerated compute. However, this new hardware deployment comes with the opex and capex price of uncoordinated, disparate, non-virtualized and non-compose-able infrastructure. GPU servers are deployed as isolated entities, independently administered, with fixed assignment of GPUs.

Bitfusion changes that. With Bitfusion, GPUs can be deployed as one shared and dynamic pool. Workloads can remotely attach to the pool and get assigned a fraction of a GPU, a single GPU or a cluster of GPUs, for only the duration of the AI run-time workload. In addition, Bitfusion alleviates the fixed attachment of workloads to physical GPUs by allowing any physical GPU to be partitioned to any number of virtual GPUs, which can be assigned remotely to an AI workload. Bitfusion provides the missing piece for AI at scale: the AI Attached Network.

Bitfusion FlexDirect Offers Elastic GPU Pools

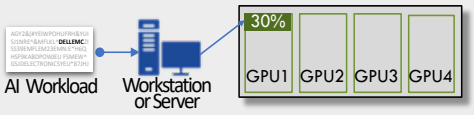
Bitfusion elastically connects GPUs to AI frameworks in a way similar to storage clusters connecting on-demand over Ethernet to client machines. Bitfusion FlexDirect software is installed on the clients (either on bare metal, VM or container), and also at the GPU server. Whenever the user runs AI frameworks that require GPU resources, Bitfusion FlexDirect will make the on-demand assignment to the remote GPU cluster, without any need for the user to be involved. In fact, users can operate as if the GPUs are connected locally to their client server's PCIe bus. Upon completion of the AI framework execution, Bitfusion FlexDirect will release the GPU resources back to the pool, again without the user needing to get involved in the task of assigning and releasing compute resources. Without any hardware changes, a shared virtualized pool can be created.

Sharing and Maximizing Utilization

GPU servers are a valuable resource. It is hard to predict the number of users, training and testing time, and how to share and allocate this resource pool across developers, researchers, testers and data scientists. GPU upgrade cycles can also be frequent, adding either capacity or newer generations of GPU servers. Bitfusion offers a radical new concept: share the AI resources across all users and client machines, in the most elastic and real-time way possible. Assume, for example, a university campus with a cluster of eight GPU servers, and total of 64 physical GPUs. That's it. With Bitfusion these are some of the ways the cluster can be used (real-time assignments and not *hard* provisioning):

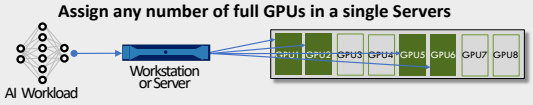
1. **64 remote users**, working with their client machines and environment (each can be a VM, container or bare metal, with completely different software packages); each of the users is allocated with a **single GPU**; or
2. **256 remote users**, each allocated **25% of a physical GPU:** optimized for development, debugging and getting familiar with new frameworks; or
3. **8 heavy** users with four GPUs each, **20 users** with single GPU, and **24 lite users** each with 50% of a physical GPU; or
4. Any possible combination of assigning a user to 1/20 of a GPU all the way to 64 GPUs (the practical limit of CUDA)

Assign any partial of a GPU in a Server



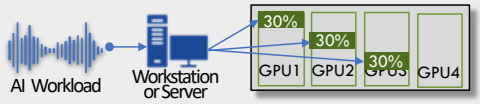
A machine learning workload running on a (GPU-less) workstation or CPU server, executes a run-time code that requires a fraction of a GPU (e.g. 30% of V100). Bitfusion connects over Ethernet the fractional GPU to the CPU server for only the duration of the run-time code, and then releases the fractional GPU back to the shared pool.

Assign any number of full GPUs in a single Servers



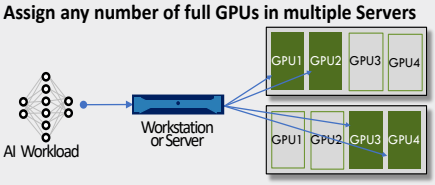
A machine learning workload running on a (GPU-less) workstation or CPU server executes a run-time code that requires multiple GPUs (e.g. 4x V100). Over Ethernet, Bitfusion connects the 4x GPUs to the CPU server for only the duration of the run-time code, and then releases the 4x GPUs back to the shared pool.

Assign any number of partial GPUs in a Server



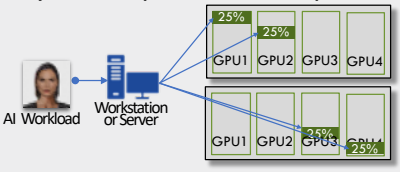
A machine learning workload running on a (GPU-less) workstation or CPU server executes a run-time code that requires multiple fractional GPUs (e.g. 3x 30% of V100). Over Ethernet, Bitfusion connects the 3 fractional GPUs to the CPU server for only the duration of the run-time code, and then releases the fractional GPUs back to the shared pool.

Assign any number of full GPUs in multiple Servers



A machine learning workload running on a (GPU-less) workstation or CPU server executes a run-time code that requires multiple GPUs (e.g. 4x V100). Over Ethernet, Bitfusion connects 2x GPUs from each GPU server to the CPU server for only the duration of the run-time code, and then releases the 4x GPUs back to the shared pool.

Assign any number of partial GPUs in multiple Servers



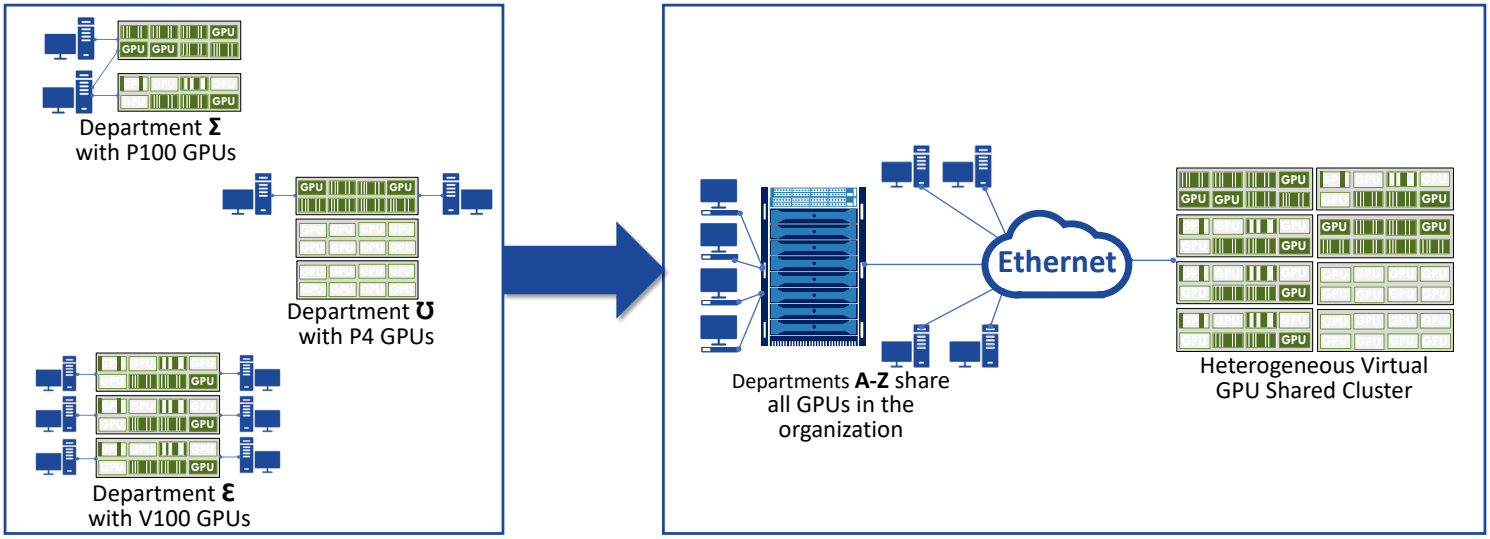
GPUs are at peak utilization. A machine learning workload requires one or more GPUs, but there are no GPUs available, just fractions of GPUs across multiple servers. Bitfusion can assign multiple partial GPUs (e.g. 4x 25%) to the workload over Ethernet.

In any of these scenarios, users can be anywhere on the campus, as long as there is an Ethernet connection. Users do not need to regenerate or recreate their favorite container or VM environment and port it to the GPU server.

The same CPU machines (without GPUs) used for general purpose development will run the deep learning framework with CUDA kernels. When there is a call for CUDA, Bitfusion will attach the remote GPUs on demand, and execution will continue un-interrupted. It is important to note that Bitfusion requires no modification to the OS, hypervisor, container or the AI framework. Bitfusion will run transparently with any GPU, any GPU server and with any network interface card (NIC).

Flexibility, Agility and Cost Savings

With the emergence of AI, we see developers, testers and researchers across organizations rushing to purchase GPU servers for their labs and projects. This creates silos and fragmented deployment across the enterprise. Within the same organization it is very common to have multiple GPU servers (potentially clusters) in different locations, rooms, departments – all locally serving users. When profiling these scattered GPU servers, a pattern emerges: GPU servers are unused or underutilized for long periods of time (or different teams are using different generations of GPUs). With Bitfusion installed, all GPU servers (unrelated to their physical location) can serve all of the organization's AI demand as a unified cluster, elastically and with maximum utility. The only requirement is to have the GPU servers connected to the internal Ethernet network. Now there is complete freedom for any user to connect from any VM/container to the virtual cluster and have a consumption-based GPU model. IT administrators can create policies assigning the top-of-the line GPUs to advance researchers, reserving the mid-range GPUs for other developers. The virtual cluster can be dressed with multiple personalities, such as development with short consumption cycles during the day, and at night the cluster is assigned for longer training cycles.



*From scattered, underutilized, non-optimized and uncoordinated GPUs deployment to **Unified**, **Virtualized** and **Elastic** GPU cluster*

As AI technology continues to mature and proliferate across industries, there will be a growing need for deployment of more and more GPU resources. There will be a significant churn of workloads, utilization, execution time, down time and up time. AI and GPUs will follow the same evolution we saw in storage – from locally attached, to remote attached and to elastic network attached storage. For larger scale deployments, it is not possible to bundle the compute and GPU to a single local resource that restricts the developer and also restricts mass deployment. Many industry experts believe that AI frameworks and machine learning will have to be developed in CPU machines with the proper CPU, memory and storage resources while the GPUs are attached and detached based on consumption of the CUDA kernels.

Bitfusion delivers that now.