THE DATA CENTER ARCHITECTURE FOR GRAPHCORE COMPUTING

DESIGNED FOR SCALABILITY OF PARALLEL WORKLOADS

INTRODUCTION

This research examines the Graphcore data center architecture that enables highly scalable parallel processing for Artificial Intelligence (AI) and High-Performance Computing (HPC). This architecture encompasses efficient low-latency communications between Intelligence Processing Units (IPUs) within a node, within a rack, and across a data center with hundreds or even thousands of accelerators to handle exponentially increasing AI model complexity. The IPU fabric dynamically connects IPU accelerators with disaggregated servers and storage. Critically, this agile platform for parallel applications supports a comprehensive software stack to develop and optimize these workloads using open-source frameworks and Graphcore-developed libraries and development tools. Shortly, we look forward to seeing high-scale benchmarks to validate this highly scalable platform's potential.

THE IPU-MACHINE BUILDING BLOCK

We start with the foundational building block for Graphcore environments, the IPU-Machine. With the second-generation Intelligence Processing Unit, Graphcore introduced a one-rack-unit (1U) building block, the IPU-M2000, with four IPU accelerators. As we will see, this platform design enables a high level of scaling, a large amount of memory for ever-increasing model sizes, and a flexible and agile configuration of CPU servers, storage, and networking.

THE IPU-MACHINE LAYOUT

The IPU-Machine consists of four IPUs, a gateway chip, up to 450GB of DRAM, and network connections, along with dual power supplies and an innovative, self-contained liquid cooling system. The idea is simple: interconnect a series of IPU-Machines and dynamically connect them to servers in whatever server/IPU ratio makes sense. Users can adjust that ratio dynamically to optimize total compute efficiency and cost.

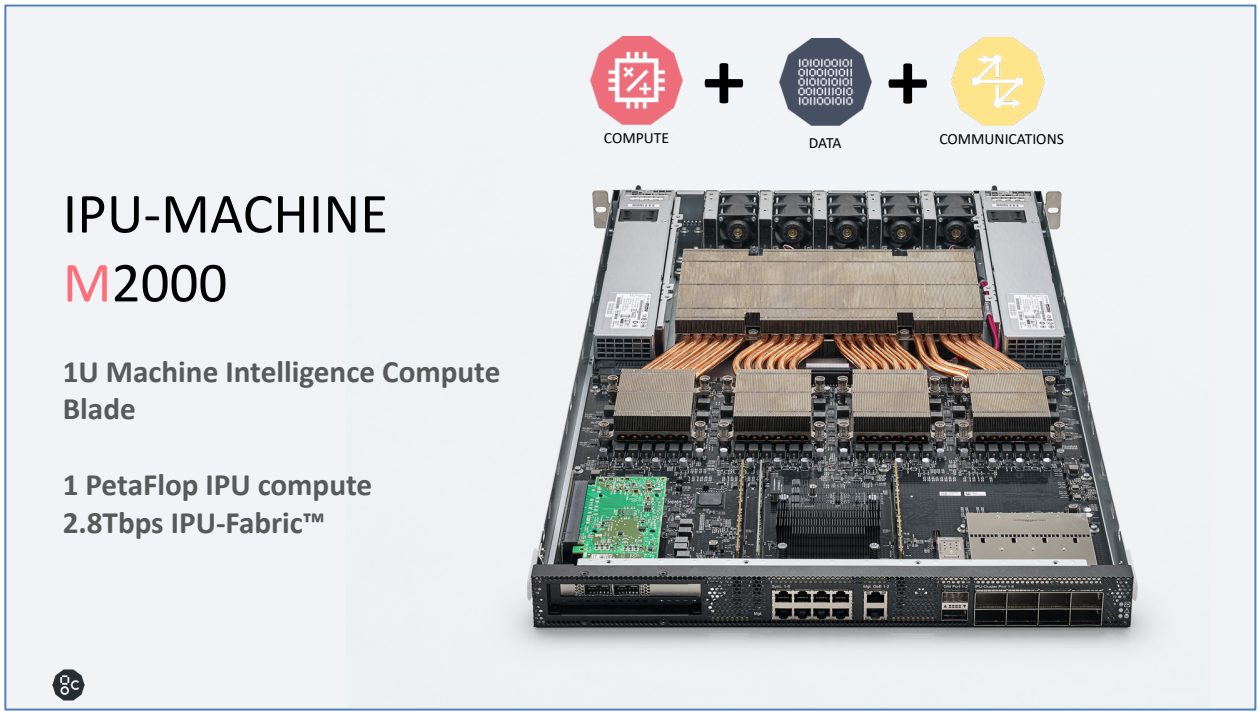

Figure 1: The IPU-Machine M2000 forms the foundation of the scalable Graphcore architecture. Source: Graphcore

NETWORKING THE IPU-MACHINE

The IPU-M2000 enables communications networking for a scalable fabric of IPU-Machines, storage, and servers. The networking is comprised of two 100Gb Ethernet links to a host server(s), four 512Gb/s IPU-Fabric links that interconnect up to 64 IPUs (an IPU-POD$_{64}$) two 100Gb Ethernet for inter-POD networking, and a "Sync-Link" to initiate the bulk synchronous parallel (BSP) model.

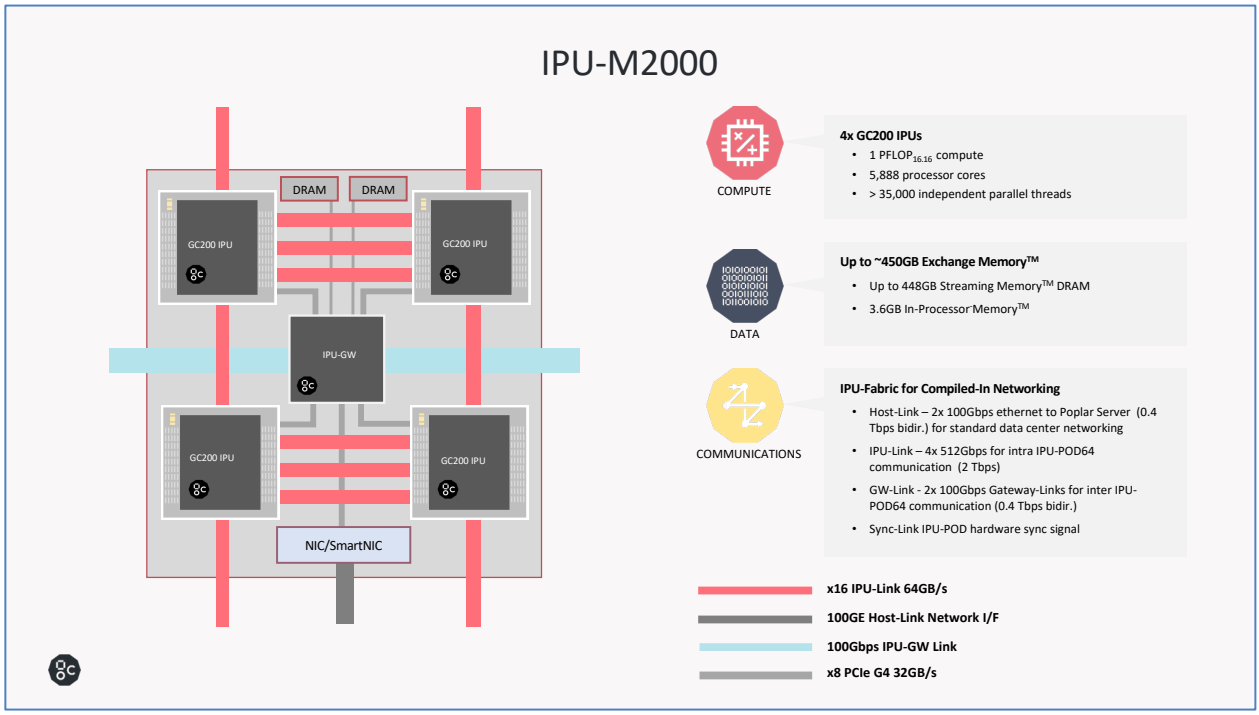

Figure 2: The IPU-M2000 creates a computing fabric, directly connecting IPU-Machines into PODs, and interconnecting these PODs over Ethernet cabling, along with Ethernet Host-link connectivity to servers and PCIe to storage. Source: Graphcore

The fabric is designed from the ground-up for AI, using an efficient, low-level point-to-point protocol that is "compiled in," eliminating message passing overhead. The two active-active inter-POD links (planes) per IPU-M2000 run this same protocol across racks over Ethernet physical layers. There is a total of 32 planes per POD_{64}. The fabric enables collectives and all-reduce operations that are managed and pre-determined at compile time. This approach provides a near-constant communication latency independently of the number of IPUs and PODs.

This approach's advantage is rooted in the simplicity and flexibility of Lego-like modules from which one can build and extend the compute fabric over a near-limitless scale. As we shall see, Graphcore has defined reference architecture PODs to enable adopters to install a scalable datacenter designed for parallel computing.

IPU Memory Architecture

As AI models continue to double in size every few months, many accelerator architectures become memory-bound. Accelerators are typically constrained to a few hundred megabytes of on-die memory or access High Bandwidth or GDDR memory. So, chip designers deal with a tradeoff between smaller SRAM caches and larger but slower off-chip memory. The Graphcore design addresses this dilemma by providing both a large (900MB) and fast on-chip SRAM with additional shared DDR "streaming"

memory across the four IPUs. The Poplar compiler determines where to put weights, activations, and other graph information. Note that HBM is quite expensive, so Graphcore delivers a cost-benefit with this approach as well.

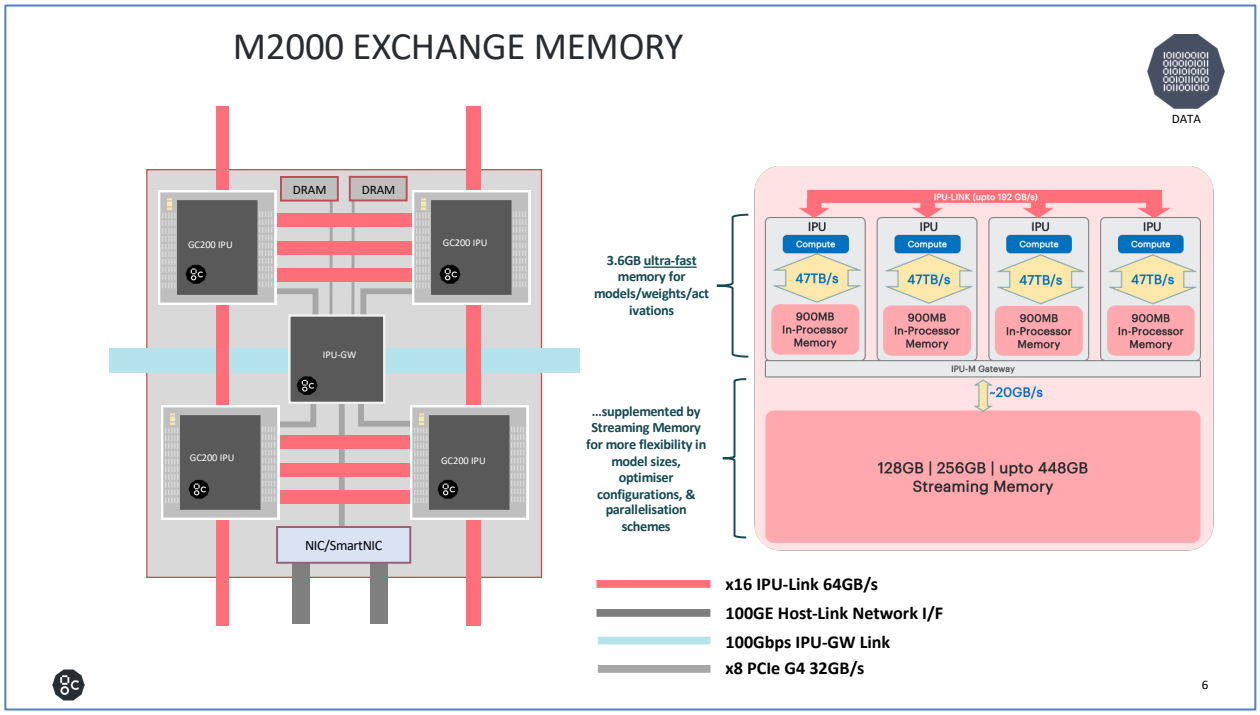

Figure 3: Each M200 has its on-die SRAM, complemented by a large DDR4 MRAM shared across four IPUs. Source: Graphcore

So, let's make a comparison. The M2000 memory approach is unique in that 448GB of DDR4 "Streaming Memory" supplements each chip's 900MB of on-die SRAM (or 3.5GB per IPU-Machine). The on-chip memory bandwidth and memory is an order of magnitude larger (900MB vs. 40MB), enabling a much larger working set on-chip. So, while a smaller HBM2 memory is faster (1.6TB/s vs. 20GB/s to DRAM) and more expensive, the far larger bulk memory capacity, on-chip memory, and memory bandwidth of the M2000 design may likely reduce the frequency of accesses to server memory. However, it is also slower by an order-of-magnitude once a model runs reach beyond the on-die memory. Unfortunately, these tradeoffs' likely impact on a particular model is not simple or obvious and can likely be discerned only through extensive testing and evaluations.

Scaling Out With IPU-PODs

Graphcore has simplified and streamlined data center implementation decisions by defining a set of reference designs. Let's look at how the IPU-Machine building block concept extends to racks, rows, and entire data centers.

IPU-POD Reference Designs

Graphcore realizes that selecting, configuring, and testing a customer data center architecture for the IPU could cost customers precious time and money. Consequently, the team has pre-configured reference designs that can easily be acquired and installed with the knowledge and comfort that Graphcore has thoroughly vetted the complete system. The IPU-POD building blocks start small at 4 IPUs (one IPU-Machine), which then simply scales to 8, 16, 32, and 64 IPU clusters with pre-configured, direct-attached networking and a single server. Once one grows to or beyond an IPU-POD$_{64}$ design, the flexible number of servers and storage reside in a separate server and storage rack.

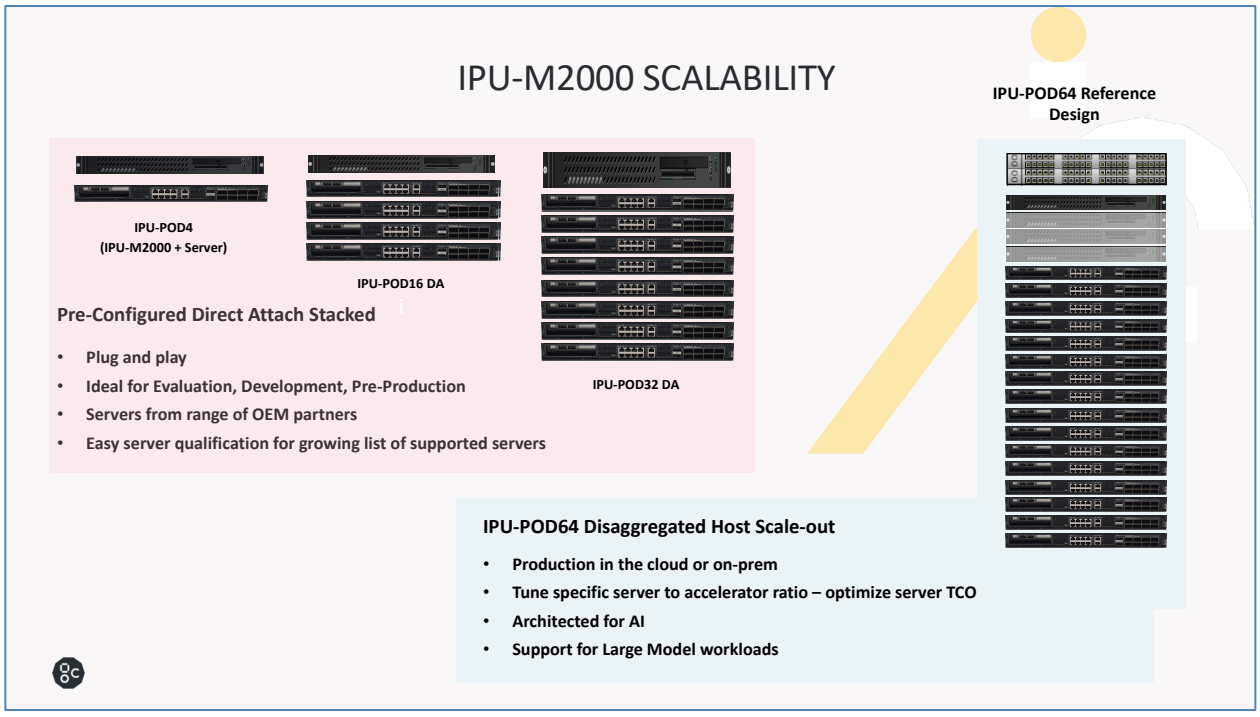

Figure 4: The IPU scalable reference design. Source: Graphcore

Let's look more deeply at the network configuration of such a rack. The host-link connects each IPU-Machine to one or more servers running the Poplar run-time software through a Top of Rack (ToR) switch. The in-rack IPU-Fabric links form a 2D torus optimized for machine learning peer-to-peer and collectives that eliminate the need for additional inter IPU switching, reducing latencies and costs. Compiled-in networking with the Graphcore Communications Library creates a 16 Peta-Flop compute pool without any internal switches or network management. We find this approach to be simple, elegant, and likely very efficient at scale. We hope to see benchmarks soon that would validate the architectural advantages.

Graphcore also has an "IPU-POD$_{16}$ Direct Attach" appliance, a pre-configured platform directly connected to a Poplar server to support Graphcore evaluation projects for HPC and AI. This POD is available through Graphcore partners and Graphcloud, a service from Cirrascale.

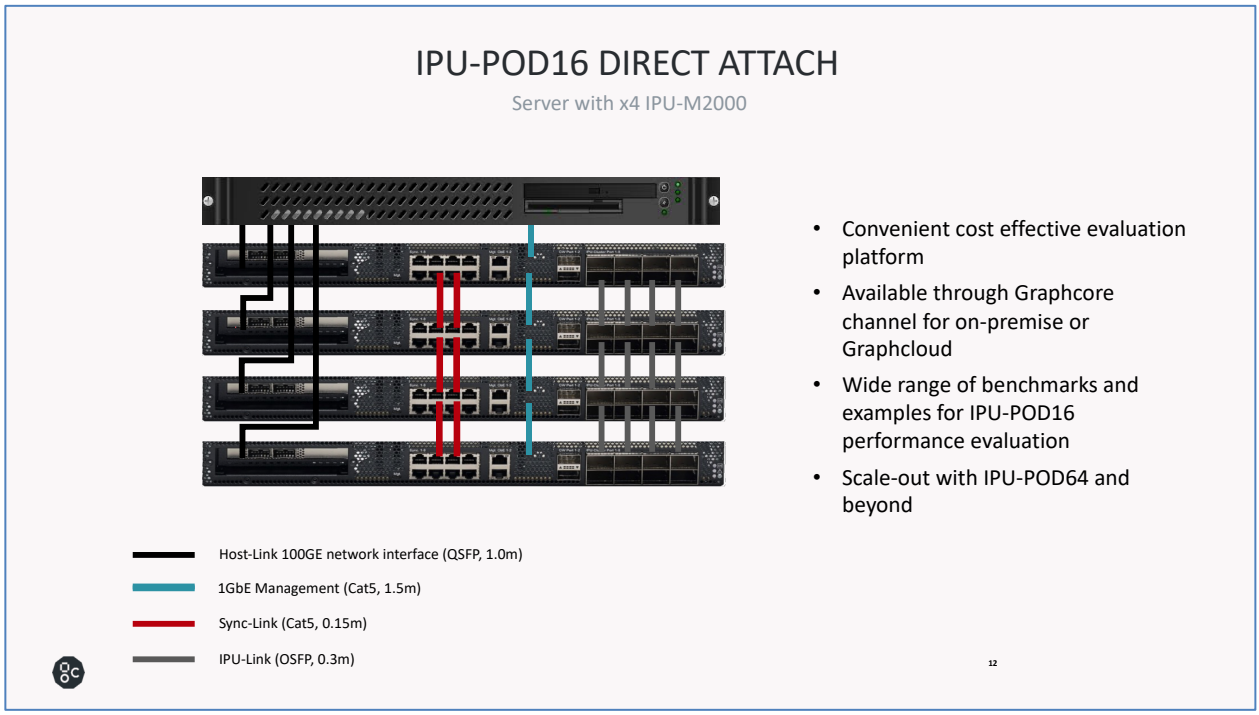

Figure 5: The IPU-POD$_{16}$ Direct Attach configuration provides a starting point for Graphcore evaluations. Source: Graphcore

As models become more extensive, many users will consider the IPU-POD$_{64}$ rack as the data center building block, so let's look at this design. (Graphcore has complete documentation available here). Graphcore selected a specific ToR 100GbE switch and a management switch from Arista, plus a qualified Dell or Supermicro Poplar server. The POD$_{64}$ networks these to the 16 IPU-Machines in the rack. Additional servers, when required, and storage connect through the ToR Switch.

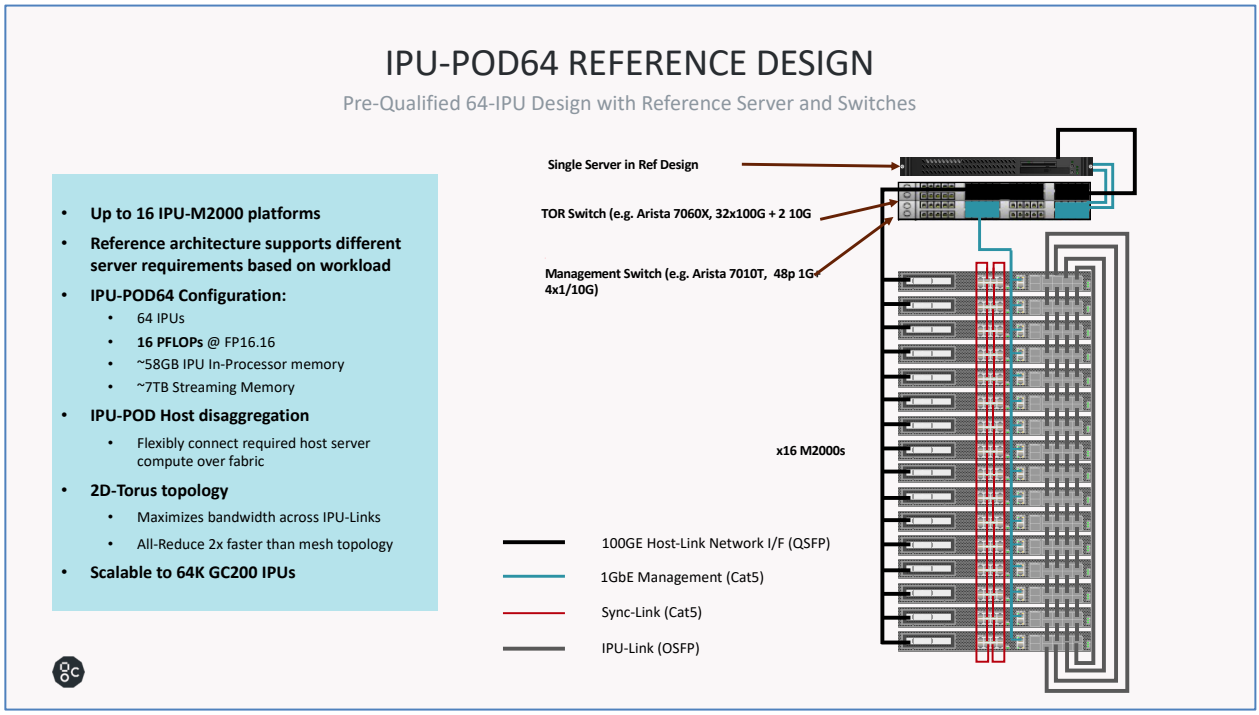

Figure 6: The POD64 forms a new data center building block for AI and other parallel workloads. Source: Graphcore

Let's examine how this modular approach scales out to larger, or indeed a vast, IPU-POD complexes. It seems evident to us that Graphcore's potential clients are evaluating the system for applications that demand significant scaling.

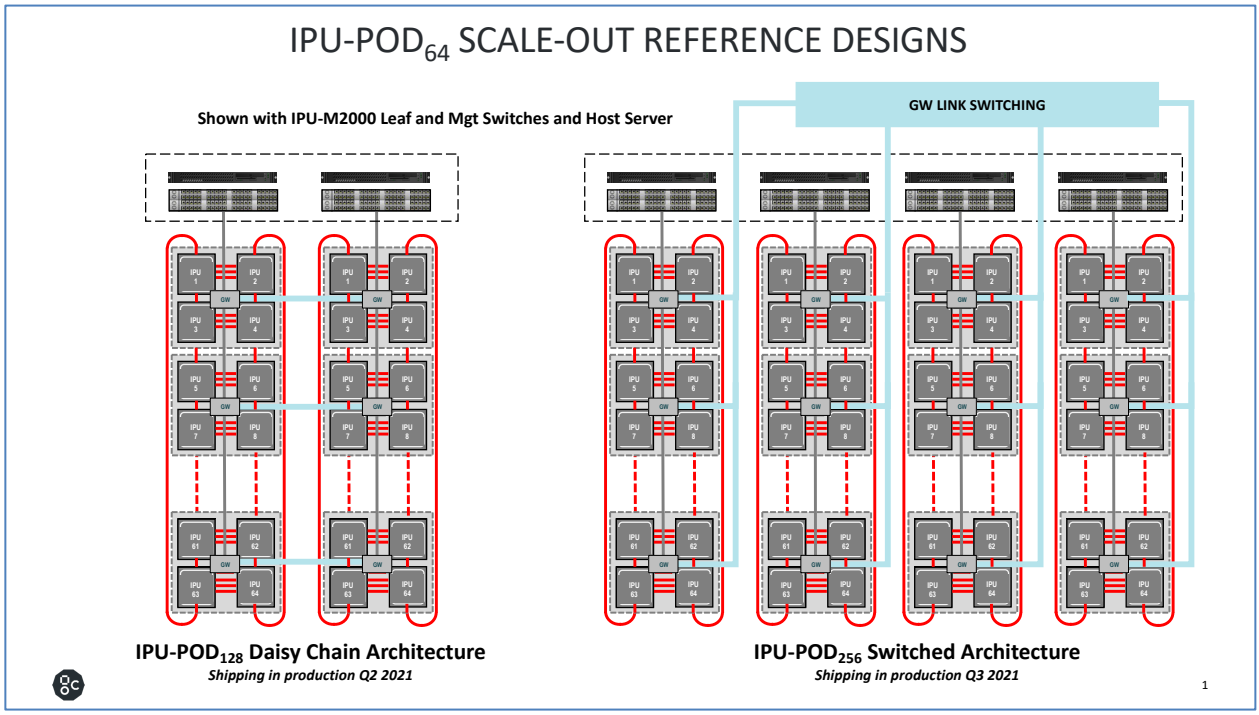

Figure 7: The IPU-POD64 building block scales to larger PODs in an elegant topology to support extremely large-scale AI workloads. Source: Graphcore

The IPU-POD$_{256}$ which is slated for production later this year, consists of 4x IPU-POD racks, wherein the individual racks connect to their dedicated Host Server, a ToR Switch, and a Management Switch. A switched fabric for the GW-Links extends to neighboring racks. This approach enables each IPU-M2000 in a rack to address any other IPU-M2000 in the IPU-POD system as a neighbor, offering seamless scaling when performing large group collective functions (All-gather, All-reduce, Scatter). In the future, up to an IPU-POD$_{16k}$ can be connected through a single switch tier consisting of 32 planes.

POPLAR SOFTWARE GCDS

While the Poplar software stack is beyond the scope of this research, which we covered in a previous research paper, we note here that the Poplar compiler can flexibly compile AI models across IPUs in "Graph Compile Domains" or GCDs. The largest GCD is currently 64 IPUs; therefore, the IPU-POD$_{64}$ is the physical platform for the Poplar SDK's maximum GCD support. Developers also have access to smaller GCDs ranging from single IPUs, 2 IPUs, 4 IPUs, and up to 64 IPUs. Note that the Poplar SDK can use the total available Exchange Memory of the IPU-POD$_{64}$ for 7 TB of available memory.

THE BENEFITS OF HOST DISAGGREGATION

AI models and other parallel applications may require vastly different accelerator/server ratios, depending on the scalar processing and management required. Consequently, instead of designing an IPU server on a PCIe card inserted into a 1- or 2-socket server, as is the case for almost all AI accelerators, the IPU-Machines are disaggregated from the host server(s).

In a typical volume deployment, the disaggregated host servers will sit behind a switched 100GE data center network. Servers are assigned to specific groups of IPU-M2000 as part of provisioning services running on the cluster, establishing a VLAN with an IP subnet. Then tenants would get specific virtual IPUs as part of an ML job queuing application like Kubernetes. Therefore, the server IPU-M2000 relationship is quite dynamic with the IPUs provisioned in "vPODs" and "vIPUs" in a way that can be set up and torn down as required. Each vPOD may consist of 1 to many servers, each with 1 to many IPUs.

The key benefits of this approach are:

1. Optimized performance for models that require more servers than, say, two CPUs for 4 or 8 accelerators,
2. Lower costs for models that require fewer CPUs by avoiding over-provisioning,
3. A flexible data center infrastructure that can handle both,
4. Servers can reside in utility racks for optimal rack power utilization and serviceability.

STORAGE CONFIGURATIONS

As is the case for most data center workloads, current storage configurations come down to direct-attached and network-attached storage. In the direct-attached case, storage is connected to the host servers, as the name implies.

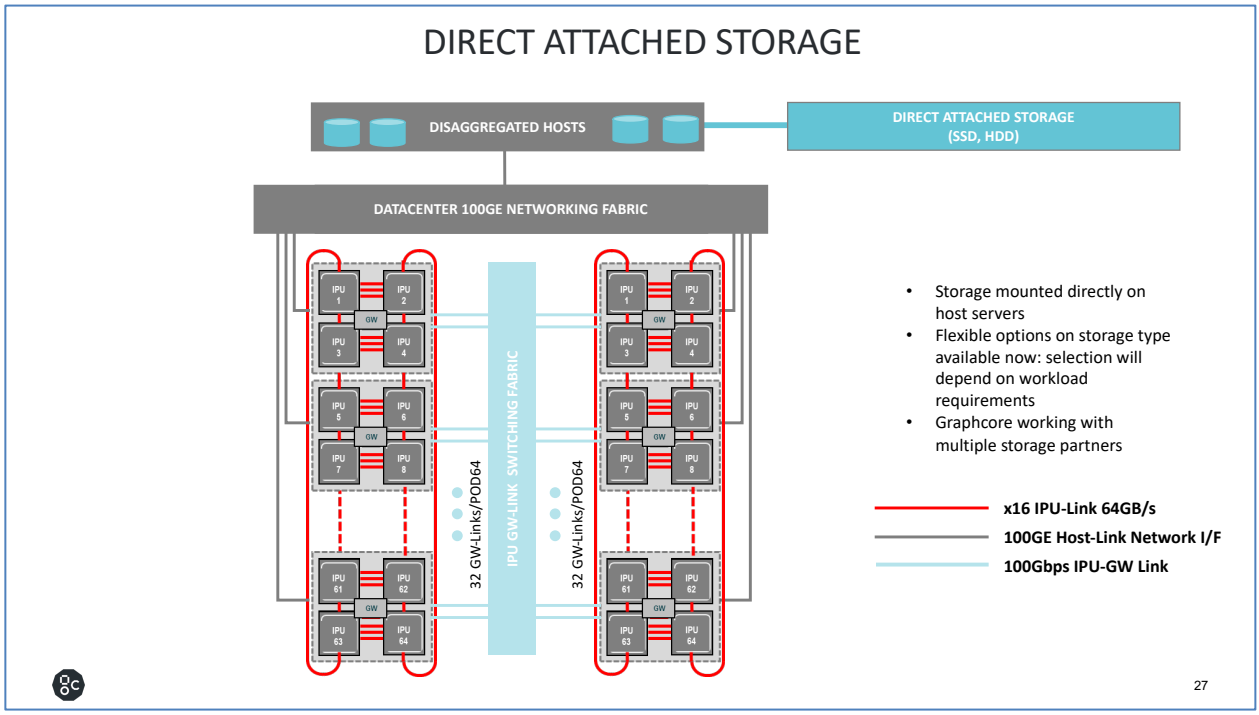

Figure 8: A typical storage topology where the data resides in storage directly attached to the host server(s). Source: Graphcore

If other servers access the data store using network-attached storage, these connect over the LAN via the ToR Switch(es).

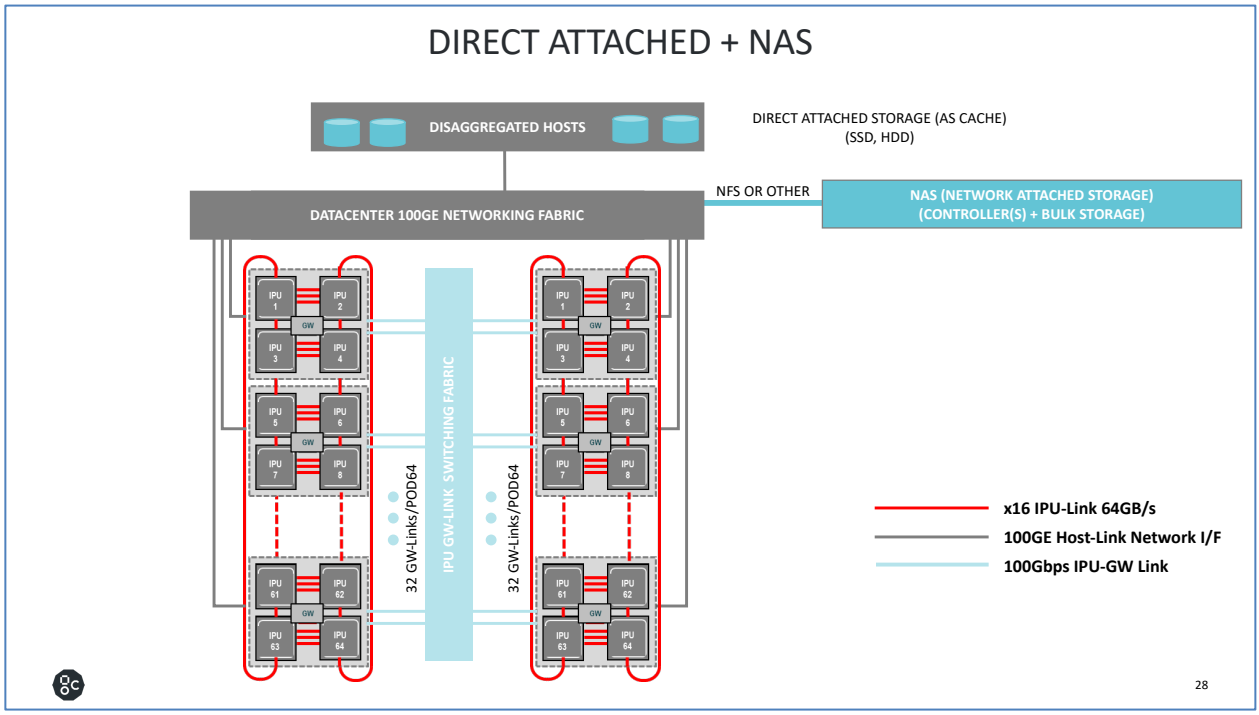

Figure 9: Storage using network-attached devices. Source: Graphcore

BRINGING IT ALL TOGETHER AT DATA CENTER SCALE

Figure 10 outlines the IPU, server, and storage deployment across a larger data-center-scale compute complex. On the left side are the NAS servers, on the right side are the host servers, and in the middle, you see the cluster of IPU_{64}, each with one server and ToR Switch. This approach's elegance is straightforward: storage and servers are co-located in a utility rack, and the POD_{64} are all identical.

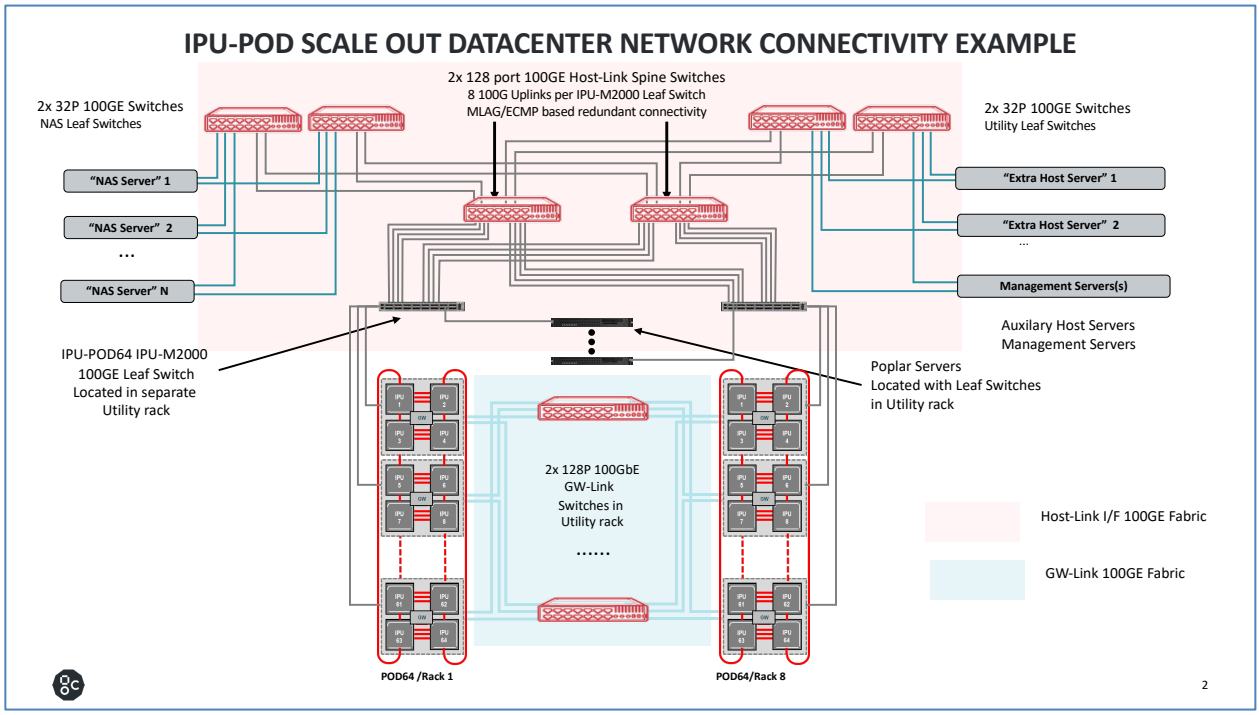

Figure 10: A compute complex comprised of accelerators, servers, networking, and storage. Source: Graphcore

CONCLUSIONS AND RECOMMENDATIONS

As we explore the data center architecture afforded by the Graphcore IPU-Machine and the Poplar software, the building-block approach's simplicity and elegance are impressive. One can start with a single IPU-Machine, then build up to 16, 32, and 64 PODs as needed. Then one can step and repeat to virtually any size fleet as business and research needs dictate. We do not yet know, but we hope to learn how well this architecture supports models running at scale. The realized performance will depend on the Poplar Software stack, especially the compiler, and should become more apparent as Graphcore completes benchmarking efforts using the MLPerf benchmarks from MLCommons.

We recommend that any organization that demands AI or other appropriate parallel computing capacities at a significant scale consider and evaluate this exciting approach, perhaps starting at the Cirrascale Graphcloud.

IMPORTANT INFORMATION ABOUT THIS PAPER

AUTHOR: Karl Freund, Founder and Principal Analyst at Cambrian-AI Research

INQUIRIES:

Contact us if you would like to discuss this report, and Cambrian-AI Research will respond promptly.

CITATIONS

This paper can be cited by accredited press and analysts but must be mentioned in the context, displaying the author's name, author's title, and "Cambrian-AI Research." Non-press and non-analysts must receive prior written permission from Cambrian-AI Research for any citations.

LICENSING

DISCLOSURES

Graphcore, Inc commissioned this paper.

DISCLAIMER

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. Cambrian-AI Research disclaims all warranties about the accuracy, completeness, or adequacy of such information and shall have no liability for errors, omissions, or inadequacies in such information. This document consists of the opinions of Cambrian-AI Research and should not be construed as statements of fact. The views expressed herein are subject to change without notice.

Cambrian-AI Research provides forecasts and forward-looking statements as directional indicators and not as precise predictions of future events. While our forecasts and forward-looking statements represent our current judgment on what the future holds, they are subject to risks and uncertainties that could cause actual results to differ materially. You are cautioned not to place undue reliance on these forecasts and forward-looking statements, which reflect our opinions only as of the date of publication for this document. Please keep in mind that we are not obligating ourselves to revise or publicly release the results of any revision to these forecasts and forward-looking statements in light of new information or future events.