White Paper

Data Center
AI GPU Virtualization

Lightweight Generative AI Inference with Intel® Data Center GPU Flex 170 on VMware ESXi*

Intel® Data Center GPU Flex 170 proves effective for lightweight AI workloads, enhancing CPU offload and AI scalability in virtualized environments.

Authors

Nathan Dragos Ciobanu

Cloud Software Development Engineer

Anirudh Krishna Lakshmanan

Cloud Software Development Engineer

Anitha Suresh

Director of Engineering

Table of Contents

Abstract

This paper outlines a series of experiments designed to evaluate the practicality of deploying the Intel® Data Center GPU Flex 170 for lightweight generative AI tasks. The central question is to determine if these GPUs can effectively offload CPUs from the computational demands of inference processes. The findings are particularly relevant for scenarios where Intel® Data Center GPU Flex 170s are used with Linux Virtual Machines (VM) in Single Root I/O Virtualization (SR-IOV) mode as specialized inference VMs in the same environments that are used for the Virtual Desktop Infrastructure (VDI) solutions. The selection of specific workloads and data types for testing was influenced by the capacity and speed of the Intel® Data Center GPU Flex 170's onboard memory and Xe-core execution units. Two benchmarks were chosen for these tests: IPEX-LLM with Meta-Llama-3-8B model and Resnet50 for image detection. We analyzed performance data across different configurations. Data showed that these GPUs can run such workloads well especially if Error Correction Code (ECC) is disabled to maximize the available device memory. In addition, the SR-IOV technology can be leveraged to manage a set of these devices with similar performance compared to direct device passthrough assignment and great manageability with VMware* vCenter* server software. Furthermore, multiple Virtual Functions (VFs) from different GPUs can be assigned to the same VM for AI workload scalability, as shown with the Resnet50 workload.

Introduction

The purpose of the experiments described in this paper is to demonstrate the feasibility of using the Intel® Data Center GPU Flex 170 for lightweight, generative AI workloads and to answer the question of whether these GPUs can be used to offload the CPU computational resources for generative AI applications. The data can be informative for use cases such GPU backed Linux VMs in SR-IOV mode can serve either as backends for VDI VMs or as dedicated inference VMs for advanced large language model (LLM) or image detection inference.

The decision to utilize the following workloads and data types was made based on the size and bandwidth of the local device memory available on an Intel Data Center GPU Flex 170, which is a maximum of 16GB GDDR6 type memory. These two representative benchmarks were proven to work well:

- IPEX LLM Meta-Llama-3-8B first and next token latency, various data types.
- Resnet50, Inference, throughput and latency, INT8, BS1024

Baseline Experiment

The experiments have been conducted on an Ice Lake node using VMware ESXi* hypervisor as the host operating system and Intel's Data Center Graphics Driver for VMware ESXi* as the software that enables the ability to split a Physical Function (PF) GPU into one or more VF GPUs that can be passed through to 1 or several individual VMs. The baseline test was conducted with 1 VF allocated to one VM, 13542 MB of GPU memory, and memory ECC enabled by default.

IPEX LLM Meta-Llama-3-8B, 1024 input tokens, 128 output tokens

The workload results are measured in terms of 'first token cost' and 'next token cost' with the expectation that the first token will always be slower than the rest of the tokens. The first iteration is expected to be slower in order of magnitude than the following iterations, so it was discarded for purposes of data comparison.

	INT4	FP4	INT8
First token latency (ms)	596.3	600.9	613.7
Next token latency (ms)	16.3	18.7	28.9
Memory used 1st token (MB):	8816	8816	12750
Memory used next token (MB):	7554	7554	11488
Memory allocated (MB):	13542	13542	13542
GPU Utilization (%):	56.19	60.77	74.28
Compute Engine Utilization (%):	55.33	59.02	72.54
Copy Engine Utilization (%):	21.97	29.29	30.56

*Table 1 - ECC On, LLM Latency as a Function of Data Type**

It is noteworthy that the amount of GPU memory is significantly reduced compared to the maximum available (13GB vs. 16GB) in part due to the buffer required for ECC when it is enabled. Since this is the shipping configuration for Intel Data Center GPU Flex 170 cards, it is part of the baseline configuration.

Due to the reduced memory running the same workload with FP16 data type was impossible as the workload quickly reached the maximum allocated GPU memory and failed as a result. This is noted as an expected limitation.

Resnet50 Inference INT8 BS1024

The workload results are measured as throughput (images/second). This workload uses the INT8 data type, and there is enough graphics memory to support it even with ECC enabled.

Throughput (images/sec)	10002.18
Average Time per Batch (ms)	113
Memory Used (MB)	5831

*Table 2 - ECC On, Resnet50 Performance**

The maximum amount of GPU memory utilized during the benchmark's runtime was 5831 MB (43%) and enough memory is allocated to support two instances of this type of workload at the same time.

ECC Off Experiment

In the following experiment, the host configuration was kept the same as in the Baseline Experiment. The only change was disabling ECC, which resulted in the VM getting 15 GB of GPU device memory instead of 13 GB when ECC was enabled.

IPEX LLM Meta-Llama-3-8B, 1024 input tokens, 128 output tokens

Data Type	INT4	FP4	INT8
First token latency (ms)	545.2	548.6	557.5
Next token latency (ms)	16.6	18.4	26.5
Memory used 1st token (MB):	8816	8816	12750
Memory used next token (MB):	7554	7554	11488
Memory allocated (MB):	16000	16000	16000
GPU Utilization (%):	53	84.06	68.44
Compute Engine Utilization (%):	51.18	83.75	66.68
Copy Engine Utilization (%):	21.86	41.24	23.06

*Table 3 - ECC Off, LLM Latency as a Function of Data Type**

* See backup for workloads and configurations. Results may vary.

In this configuration, we notice a slight improvement in the first token latency, 9%, which is explainable by the reduced overhead required by ECC when that feature is enabled.

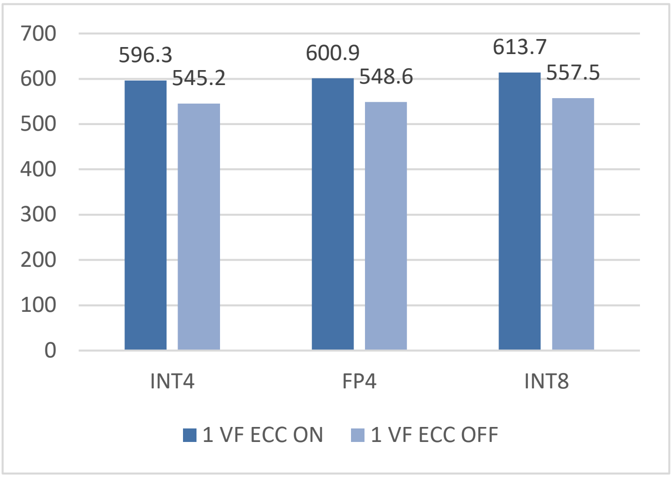

*Figure 1 - Llama3 ECC On vs Off First Token Latency (ms)**

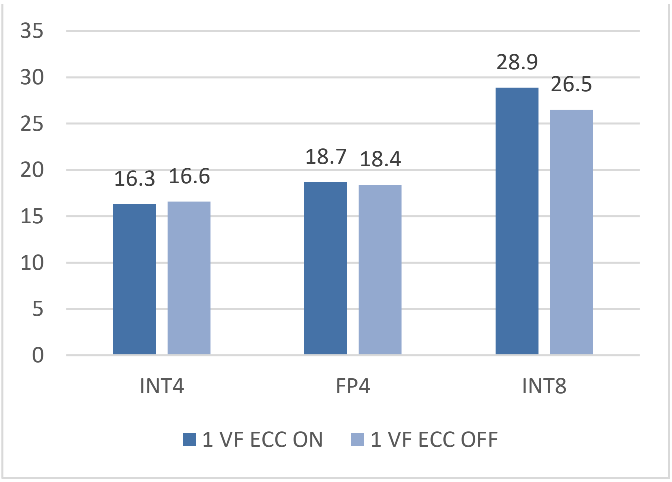

*Figure 2 - Llama3 ECC On vs Off Next Token Latency (ms)**

Resnet50 Inference INT8 BS1024, ECC Off

Throughput (images/sec)	11090.79
Average Time per Batch (ms)	94
Memory Used (MB)	5831

*Table 4 - ECC Off, Resnet50 Performance**

The maximum amount of GPU memory utilized during the runtime of the benchmark was 5831 MB, the same as the baseline, as expected. However, the throughput improved by 10% over the ECC ON configuration, and latency was reduced by 17%, which is a significant improvement.

* See backup for workloads and configurations. Results may vary.

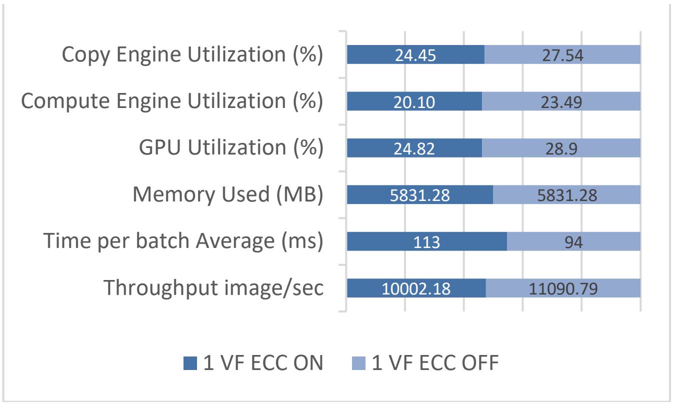

*Figure 3 - Resnet50 ECC On vs Off Performance and Utilization**

Users who want to maximize the amount of memory for AI inference use cases and get a slight performance boost have the option to disable ECC on specific GPUs. This can be done from the ESXi command line interface:

```
$ esxcli intdcgpu configuration boot ecc set -d <GPU SBDF> -s 0
```

This is also possible at-scale through the Intel® Device Manager for VMware* vCenter Server.

All other experiments were conducted with ECC disabled to maximize the amount of GPU memory and give a slight boost in performance.

IPEX LLM Meta-Llama-3-8B, data type resource scaling, input/output 32 tokens

In the following test, we changed the input and output tokens to 32 to accommodate the FP16 data type, which is required to fit the execution of the LLM benchmark into the available GPU memory.

	INT4	FP4	INT8	FP16
First token latency (ms)	84.7	92.1	86.6	82
Next token latency (ms)	14.1	17.3	25.3	49.8
Memory 1st token (MB)	6222	6222	10120	15526
Memory next token (MB)	6220	6224	10118	15526
Memory allocated (MB)	16000	16000	16000	16000

*Table 5 - Llama3 Latency, 32 IN/OUT Tokens, Memory Scaling as a Function of Data Type**

* See backup for workloads and configurations. Results may vary.

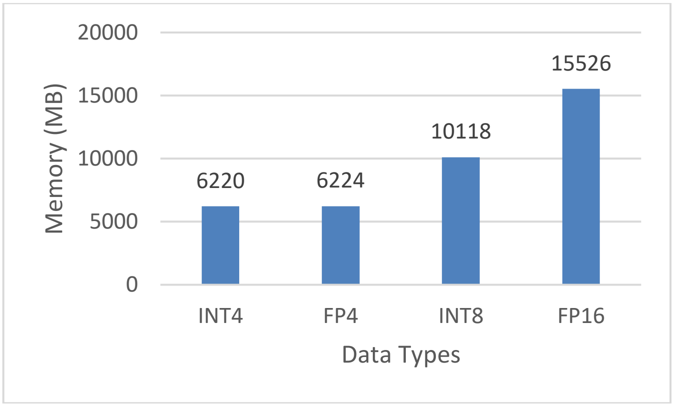

*Figure 4 - Llama3 ECC off, 32 IN/OUT Tokens, Next Token Memory Usage as a Function of Data Type**

Note that the amount of GPU memory necessary for 32 input tokens is almost maximized. Therefore, any workloads utilizing more than 32 tokens will not be able to execute on this configuration. The option is to utilize the int4/int8 and fp4/fp8 and similarly sized data types.

IPEX LLM Meta-Llama-3-8B, INT4, input scaling, output 128 tokens

In the following configuration, we wanted to show if the latency is affected by the size of the input tokens, keeping the data type fixed. This should help system administrators decide how best to configure the Intel Data Center GPU Flex 170 for large language inference tasks.

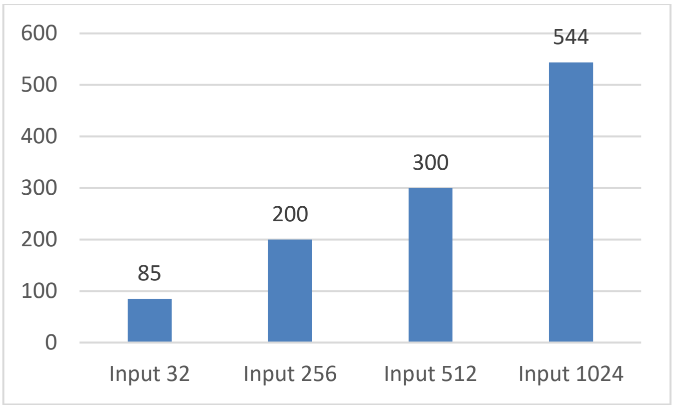

*Figure 5 - Llama 3 ECC Off, First Token Latency (ms) as a Function of Input Size with INT4 Data Type**

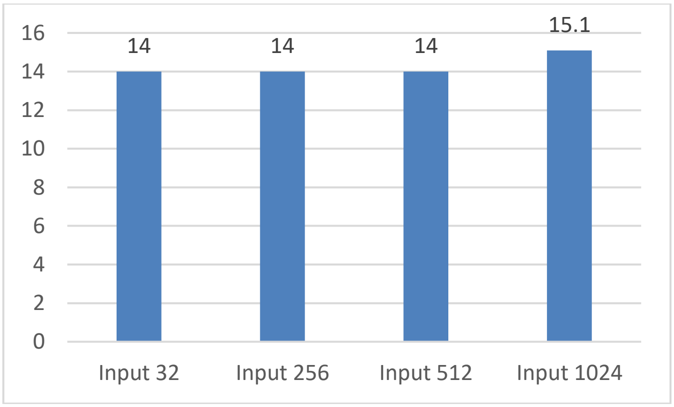

*Figure 6 - Llama 3 ECC Off, Next Token Latency (ms) as a Function of Input Size with INT4 Data Type**

* See backup for workloads and configurations. Results may vary.

As expected, we see an increase in the average token latency for the first token as we increase the input count; the same growth doesn't occur with the next token latency.

Device Passthrough Assignment

A slightly more performant configuration is to assign the entire GPU device to a single VM, thus taking advantage of full access to the hardware. The expectation is that performance would increase by 10%, but this needs to be put to the test.

	INT4	FP4	INT8
First token latency (ms)	545.8	548.4	555.4
Next token latency (ms)	15.3	18.4	26.3
Memory used 1st token (MB):	8816	8816	12750
Memory used next token (MB):	7554	7554	11488
Memory allocated (MB):	16228	16228	16228

*Table 6 - Llama 3 Latency Scaling as a Function of Data Type in Passthrough Mode with 128 IN / 1024 OUT Tokens**

In this configuration, the entire 16228 MB of GPU memory is available to the Virtual Machine, which is about a gigabit higher than in 1 VF SR-IOV mode with ECC Off. The increased memory size does not allow for the workload to be executed with the FP16 data type with 128 tokens. Performance-wise, we don't see an improvement compared to SR-IOV mode, either.

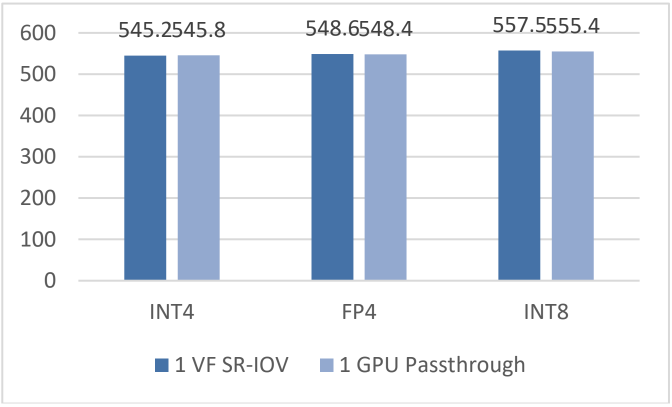

*Figure 7 - Llama 3 First Token Latency SR-IOV vs Passthrough as a Function of Data Type with 128 IN / 1024 Out Tokens, ECC Off**

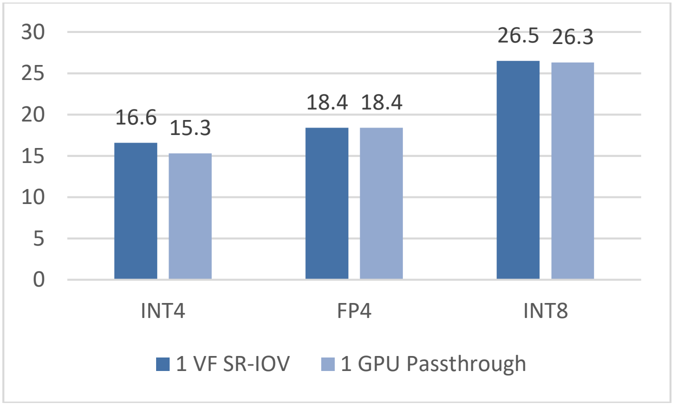

*Figure 8 - Llama 3 Next Token Latency SR-IOV vs Passthrough as a Function of Data Type with 128 IN / 1024 Out Tokens, ECC Off**

* See backup for workloads and configurations. Results may vary.

Checking image detection inference performance with Resnet50 reveals the same conclusion: no performance improvement.

Resnet50 Inference INT8 BS1024 Passthrough Mode

Throughput (images/sec)	11114.57
Average Time per Batch (ms)	95
Memory Used (MB)	5834

*Table 7 - Resnet50 Performance in Passthrough Mode**

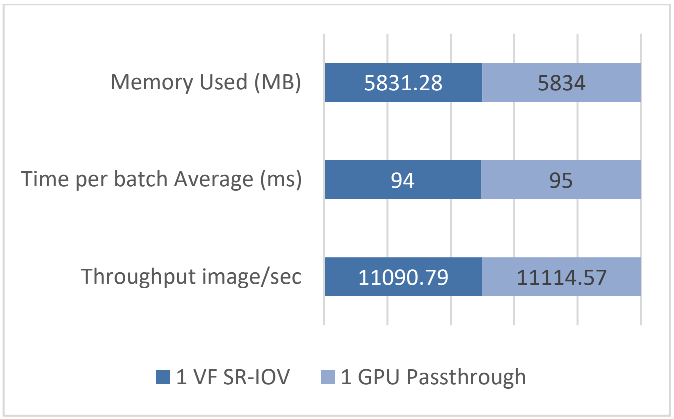

*Figure 9 - Resnet50 Performance SR-IOV vs Passthrough Mode**

Looking at the difference between passthrough and SR-IOV with regards to throughput and latency, the differences in this type of workload are not significant; therefore, given the loss of manageability features in full device passthrough and no empirical performance advantage, it is more advantageous for system administrators to configure the full virtualization solution with Intel hardware/software stack and the VMware SR-IOV virtualization solution.

VF Scale-Up

In this experiment, two VFs from two different GPUs on the same system, were assigned to a single VM to see if the workloads scale. The IPEX LLM workload is not capable of taking advantage of software scale-up, so data is based solely on the Resnet50 workload. This is done by assigning two VFs, one from each physical GPU, to a single VM in SR-IOV mode with ECC disabled.

The data shows a 2x improvement compared to 1 VF per VM experiment with ECC Off.

Throughput (images/sec)	22003.09
Average Time per Batch (ms)	95.5
Memory Used (MB)	5821.33
GPU Utilization (%)	27.985
Compute Engine Utilization (%)	22.27
Copy Engine Utilization (%)	26.635

*Table 8 - Resnet50 Scale-Up Performance, ECC Off**

Noteworthy here is that the workload distribution is done purely by software between the two GPUs. The same amount of memory was used on each GPU, and latency and utilization were basically the same on each GPU. The throughput doubled on 2 VFs vs. 1 VF, so the workload scales up as expected.

* See backup for workloads and configurations. Results may vary.

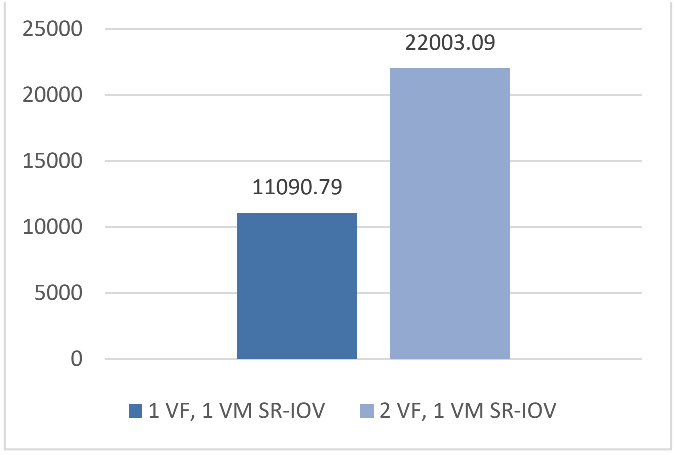

*Figure 10 - Resnet50 2 VF vs 1 VF Scale-Up Throughput (images/sec)**

Conclusion

These experiments were set out to prove that Intel® Data Center GPU Flex 170 can be used as a light AI graphics accelerator in a virtualization environment and that the performance would be acceptable to be considered as a low TCO in the data center; data presented proved that. In addition, the SR-IOV technology can be leveraged to manage a multiple of these devices with no performance delta compared to direct device passthrough.

The data also shows that with the adequate use of multi-GPU scaling software such as DeepSpeed, we can combine 2 GPUs in SR-IOV mode to double the performance of a single device, as shown in the VF Scale-Up experiment with the Resnet50 workload.

Finally, even though the Flex 170 has reduced memory size and number of Xe cores, administrators can pick and finetune the type of workloads they would run on these devices. As the name implies, these GPUs are flexible enough to be combined in VDI and AI low-cost backend accelerators and CPU off-loaders, taking advantage of the software management provided by VMware by Broadcom* and Intel®'s data center stacks.

* See backup for workloads and configurations. Results may vary.

Additional Resources

Host graphics driver: Intel® Data Center Graphics Driver for VMware ESXi*
Download and Documentation: https://www.intel.com/content/www/us/en/download/786751/intel-data-center-graphics-driver-for-vmware-esxi.html

Guest Graphics Driver: i915
Download and Documentation: https://dgpu-docs.intel.com/releases/stable_775_20_20231219.html

Intel® Device Manager for VMware* vCenter Server
Download and Documentation: https://www.intel.com/content/www/us/en/download/772505/intel-device-manager-for-vmware-vcenter-server.html?cache=1677785960

IPEX-LLM Documentation, Framework: https://github.com/intel-analytics/ipex-llm

Meta-Llama-3-8B Model: https://huggingface.co/meta-llama/Meta-Llama-3-8B

PyTorch.Resnet50:
https://pytorch.org/vision/main/models/generated/torchvision.models.resnet50.html

Backup: Test Configuration

Testing was performed 06/01/2024 – 06/03/2024 by Intel®

Time	Sat Jun 1 14:32:30 UTC 2024
System	Intel Corporation M50CYP
Baseboard	Intel Corporation M50CYP
Chassis	Rack Mount Chassis
CPU Model	Intel(R) Xeon(R) Platinum 8360Y
Microarchitecture	CYP
Sockets	2
Cores per Socket	36
Hyperthreading	Enabled
CPUs	144
Intel Turbo Boost	Enabled
Base Frequency	2.4GHz
All-core Maximum Frequency	2.7GHz
Maximum Frequency	3.5GHz
NUMA Nodes	2
Prefetchers	MLS, DCU, LLC
Installed Memory	128GB 8 x 16 GB Hynix 3200 MT/s DDR4 DIMMs
NIC	Intel Corporation Gigabit CT Desktop Adapter, 1000MB/s
Disk	INTEL SSDSC2KG96 894.25 GB
BIOS	SE5C6200.86B.0027.P15.2205121306
Microcode	0x0d0003a5
OS	VMware ESXi 8.0U2-22380479
Kernel	VMkernel 8.0.2 #1 SMP Release build-22380479 Sep 4 2023 15:00:49 x86_64 x86_64 x86_64 ESXi
TDP	250 watts
Power & Perf Policy	Balanced
Max C-State	C6

Table 9 - System Host Configuration

Time	Mon Jun 3 04:36:28 PM UTC 2024
System	VMware, Inc. VMware20,1
Baseboard	Intel Corporation 440BX Desktop Reference Platform
Chassis	No Enclosure Other
CPU Model	Intel(R) Xeon(R) Platinum 8360Y CPU @ 2.40GHz
Microarchitecture	ICX
Sockets	32
Cores per Socket	1
Hyperthreading	Disabled
CPUs	32
Intel Turbo Boost	Disabled
Base Frequency	2.3GHz
NUMA Nodes	1
Prefetchers	L2 HW: Enabled, L2 Adj.: Enabled, DCU HW: Enabled, DCU IP: Enabled
PPINs	0
Accelerators Available [used]	DLB 0 [0], DSA 0 [0], IAA 0 [0], QAT 0 [0]
Installed Memory	64GB (4x16GB DRAM Unknown [4800 MT/s])
Hugepagesize	2048 kB
Transparent Huge Pages	madvise
Automatic NUMA Balancing	Disabled
NIC	1x VMXNET3 Ethernet Controller
Disk	1x 300G Virtual disk, 1x 2G VMware Virtual SATA CDRW Drive
BIOS	VMW201.00V.21805430.B64.2305221830
Microcode	0xd0003a5
OS	Ubuntu 22.04.4 LTS
Kernel	5.15.0-107-generic
Power & Perf Policy	Performance (0)
Max C-State	9

Table 10 - System Guest VM Configuration

Graphics Firmware Version	DG02_1.3271
AMC Firmware Version	6.8.0

Table 11 - GPU Firmware

Software Stack	Intel® oneAPI Base Toolkit 2024.0
Framework	IPEX LLM 2.1.0b20240524 and IPEX_2.1.0+xpu RC3

Table 12 - Software Framework Configuration

VF GPU Driver	I915_24.1.5_PSB_240117.8
PF GPU Driver	1.4.0.16-1OEM.800.1.0.20613240
Host OS	VMware ESXi 8.0U2-22380479
Guest OS/kernel	Ubuntu Server 22.04 LTS, 5.15.0-107-generic
VF Profile	Flex170_16

Table 13 - Driver and OS Versions

Notices & Disclaimers