



BENCHMARKING OF REAL-WORLD AI INFERENCE WORKLOADS ON THE INTEL GAUDI 3 ACCELERATORS

GIANLUCA LONGONI, PH.D., P.E.

EXECUTIVE TECHNICAL DIRECTOR AI/HPC

FEDDATA TECHNOLOGY SOLUTIONS

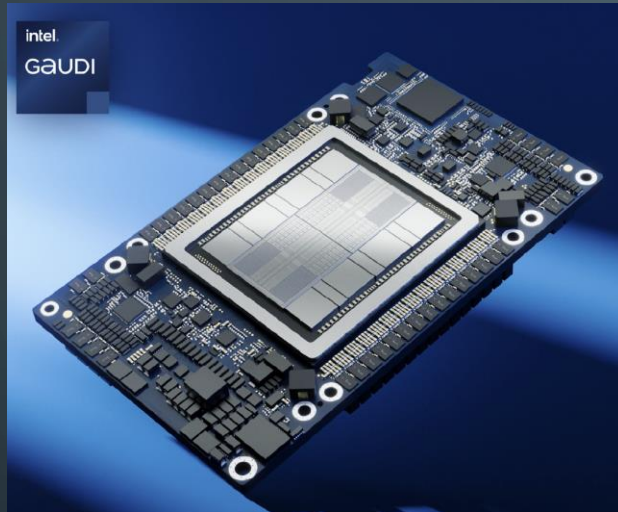
Presented to the HPC User Forum
Spring 2025, Santa Fe, NM

OVERVIEW

- Description of the benchmarking environment
 - Intel Gaudi 3 Habana Processing Unit (HPU)
 - vLLM AI serving engine
 - Benchmark description and metrics used in the analysis
- Discussion of the results
- Some considerations from a developer perspective on Gaudi 3...
- Conclusions
- Future Work
- Acknowledgements

The results presented herein do not constitute an endorsement to any of the technologies evaluated.

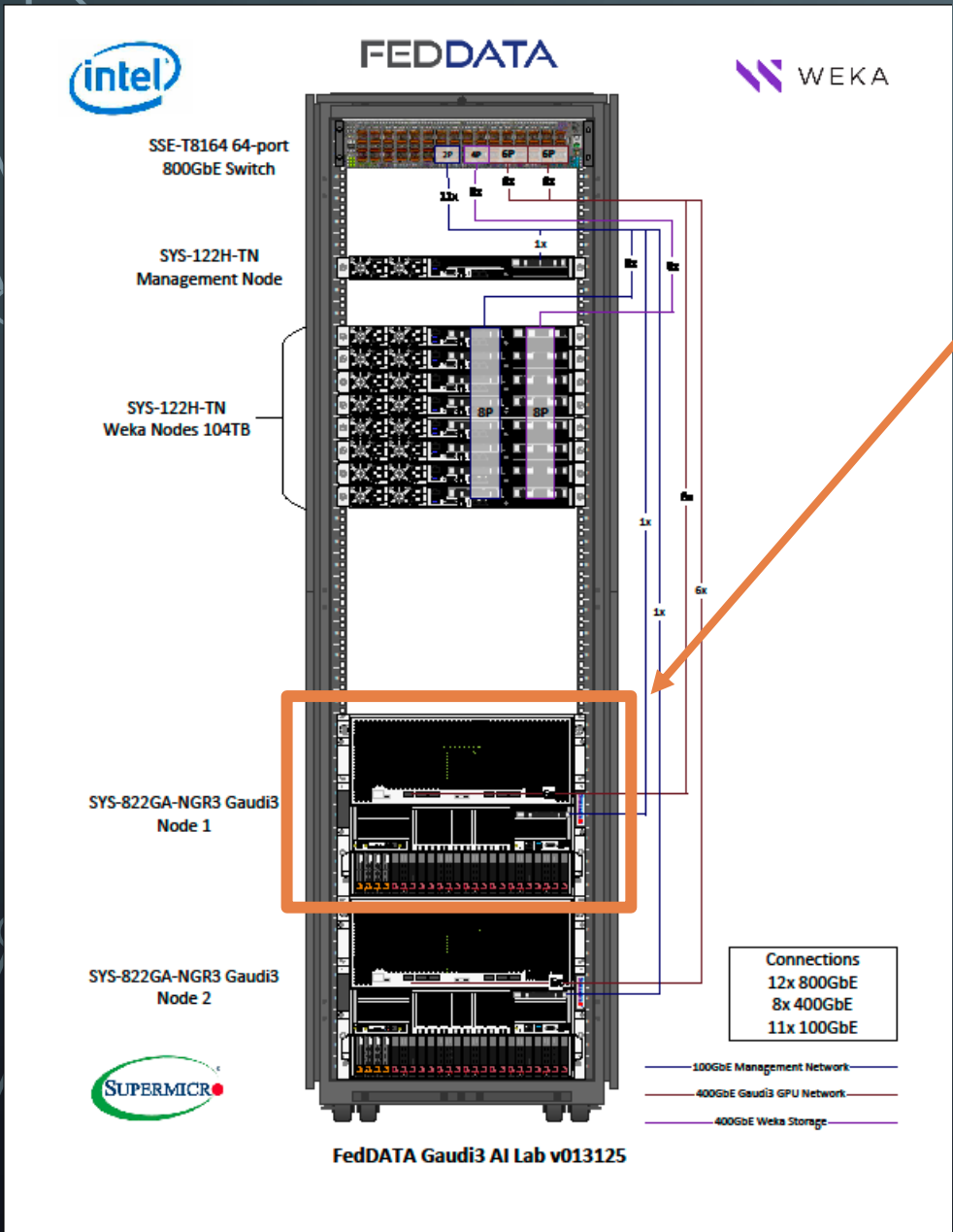
INTEL GAUDI 3 AI ACCELERATOR



Gaudi 3 presents significant improvements on the HW side that makes it a viable candidate for production-level AI workloads

Feature/Product	Intel® Gaudi® 2 AI Accelerator	Intel® Gaudi® 3 AI Accelerator
BF16 MME TFLOPS	432	1835
FP8 MME TFLOPS	865	1835
BF16 Vector TFLOPS	11	28,7
MME Units	2	8
TPC Units	24	64
HBM Capacity	96 GB	128 GB
HBM Bandwidth	2.46 TB/s	3.7 TB/s
On-die SRAM Capacity	48 MB	96 MB
On-die SRAM Bandwidth	6.4 TB/s	12.8 TB/s
Networking (bidirectional)	600 GB/s	1200 GB/s
Host Interface	PCIe Gen4 X16	PCIe Gen5 X16
Host Interface Peak BW	64 GB/s (32 GB/s per direction)	128 GB/s (64 GB/s per direction)
Media Decoders	8	14

- 2x**
FP8 GEMM FLOPs
- 4x**
BF16 GEMM FLOPs
- 1.5x**
Faster HBM Bandwidth
- 1.33x**
Larger HBM Capacity



Overall System Technical Specifications:

- 1 Node
- Intel(R) Xeon(R) 6980P – 256 cores - 2 threads/core 3.9 GHz, 2.3 TB RAM
- 8x HL-325L Intel Gaudi 3 accelerators, latest firmware patches Applied (hl-1.19.2-fw-57.2.4.0)

HL-SMI Version:		hl-1.19.2-fw-57.2.4.0					
Driver Version:		1.19.2-ff37fea					
AIP	Name	Persistence-M	Bus-Id	Disp.A	Volatile	Uncor-Events	
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	AIP-Util	Compute M.	
0	HL-325L	N/A	0000:17:00.0	N/A	0%	0	
N/A	36C	N/A	187W / 900W	672MiB / 131072MiB		N/A	
1	HL-325L	N/A	0000:2c:00.0	N/A	0%	0	
N/A	34C	N/A	186W / 900W	672MiB / 131072MiB		N/A	
2	HL-325L	N/A	0000:97:00.0	N/A	0%	0	
N/A	37C	N/A	190W / 900W	672MiB / 131072MiB		N/A	
3	HL-325L	N/A	0000:ba:00.0	N/A	0%	0	
N/A	32C	N/A	182W / 900W	672MiB / 131072MiB		N/A	
4	HL-325L	N/A	0000:4e:00.0	N/A	0%	0	
N/A	38C	N/A	190W / 900W	672MiB / 131072MiB		N/A	
5	HL-325L	N/A	0000:a9:00.0	N/A	0%	0	
N/A	33C	N/A	188W / 900W	672MiB / 131072MiB		N/A	
6	HL-325L	N/A	0000:3d:00.0	N/A	0%	0	
N/A	34C	N/A	187W / 900W	672MiB / 131072MiB		N/A	
7	HL-325L	N/A	0000:cb:00.0	N/A	0%	0	
N/A	36C	N/A	187W / 900W	672MiB / 131072MiB		N/A	
Compute Processes:				AIP Memory Usage			
AIP	PID	Type	Process name				
0	N/A	N/A	N/A	N/A			
1	N/A	N/A	N/A	N/A			
2	N/A	N/A	N/A	N/A			
3	N/A	N/A	N/A	N/A			
4	N/A	N/A	N/A	N/A			
5	N/A	N/A	N/A	N/A			
6	N/A	N/A	N/A	N/A			
7	N/A	N/A	N/A	N/A			

VLLM: LLM INFERENCE AND SERVING



Highlights:

- State-of-the-art serving throughput
- Seamless integration with popular HuggingFace models
- Tensor parallelism and pipeline parallelism support for distributed inference
- Support NVIDIA GPUs, AMD CPUs and GPUs, Intel CPUs, Gaudi® accelerators and GPUs, PowerPC CPUs, TPU, and AWS Trainium and Inferentia Accelerators.
- Extensive and up-to-date benchmarking:
- <https://simon-mo-workspace.observablehq.cloud/vllm-dashboard-v0/perf>

Benchmark Specifications:

- vLLM 0.6.3.dev2224+g3457022d1, pulled directly from GitHub vllm-fork at <https://github.com/HabanaAI/vllm-fork>
- vLLM Serving Benchmark (benchmark_serving.py)
 - `python3 benchmark_serving.py --num-prompts <#prompts> --dataset ./ShareGPT_V3_unfiltered_cleaned_split.json --port 9050 --model <#cached LLM>`
 - `#prompts = [50, 100, 200, 500, 1000]`
 - `#cached LLM = [Mistral-7B-Instruct-v0.3, Llama-3.1-8B, Llama-3.3-70B-Instruct]`
 - Dataset: ShareGPT_V3_unfiltered_cleaned_split, about 53K chat conversations

BENCHMARK DESCRIPTION AND METRICS

- The benchmark simulates AI inferencing based on a number of concurrent user queries
- Variable number of concurrent queries can be specified, 50, 100, 200...
- This is a stress-test that measures real-world serving of an LLM for production purposes, i.e., chatbot, RAG...

Dataset: ShareGPT_V3_unfiltered_cleaned_split

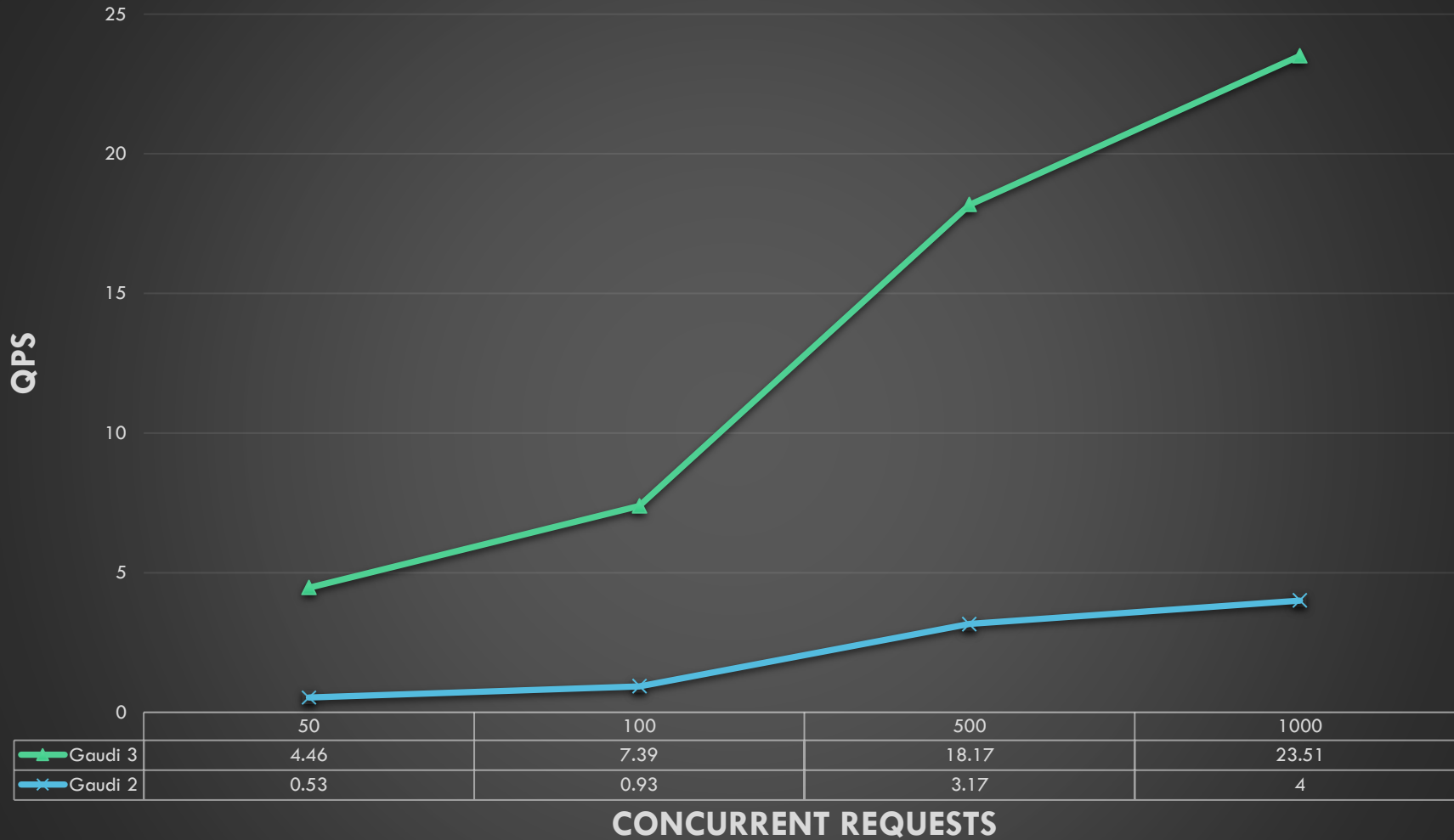
```
[
  {
    "id": "QWJhYvA_0",
    "conversations": [
      {
        "from": "human",
        "value": "Summarize the main ideas of Jeff Walker's Product Launch Formula into bullet points as it pertains to a growth marketing agency implementing the strategies and tactics for their clients..."
      },
      {
        "from": "gpt",
        "value": "Here are the main ideas of Jeff Walker's Product Launch Formula that can be applied by a growth marketing agency for their clients:\n\n1. Identify the target audience and their needs: Understand the ideal customer for the product or service, and create a messaging that resonates with them.\n2. Pre-launch: Build anticipation and excitement for the launch by creating buzz, gathering testimonials and case studies, and using social media to create awareness.\n3. Launch: Use a well-crafted launch sequence to maximize sales and conversions. This can include offering bonuses, creating scarcity, and using a deadline to create urgency.\n4. Post-launch: Follow up with customers, gather feedback, and continue to provide value to keep them engaged and loyal.\n5. Create a product suite: Once the initial product is successful, expand the offering by creating additional products or services that address other needs of the same audience.\n6. Continual optimization: Continually monitor and optimize the launch process and product suite to improve results.\n7. Build a community: Use the launch process to build a community of customers who are passionate about the product and can help spread the word.\n8. Use automation: Use technology and automation to streamline the launch process and improve efficiency."
      }
    ]
  },
]
```

BENCHMARK DESCRIPTION AND METRICS (CONT'D)

- **Queries Per Second (QPS):** The total number of queries being successfully served per second, somewhat related to the number of users being served per second.
- **Time To First Token (TTFT):** How quickly users start seeing the model's output after entering their query. Low waiting times for a response are essential in real-time interactions. This metric is driven by the time required to process the prompt and then generate the first output token.
- **Time Per Output Token (TPOT):** Time to generate an output token for each user that is querying our system. This metric corresponds with how each user will perceive the "speed" of the model. For example, a TPOT of 100 milliseconds/tok would be 10 tokens per second per user, or ~450 words per minute, which is faster than a typical person can read.
- **Total Throughput:** The number of input tokens (query) processed and output tokens per second an inference server can generate across all users and requests.

QPS - MISTRAL-7B V0.3 INSTRUCT LLM

Gaudi 3 vs. Gaudi 2 - Queries Per Second (QPS)

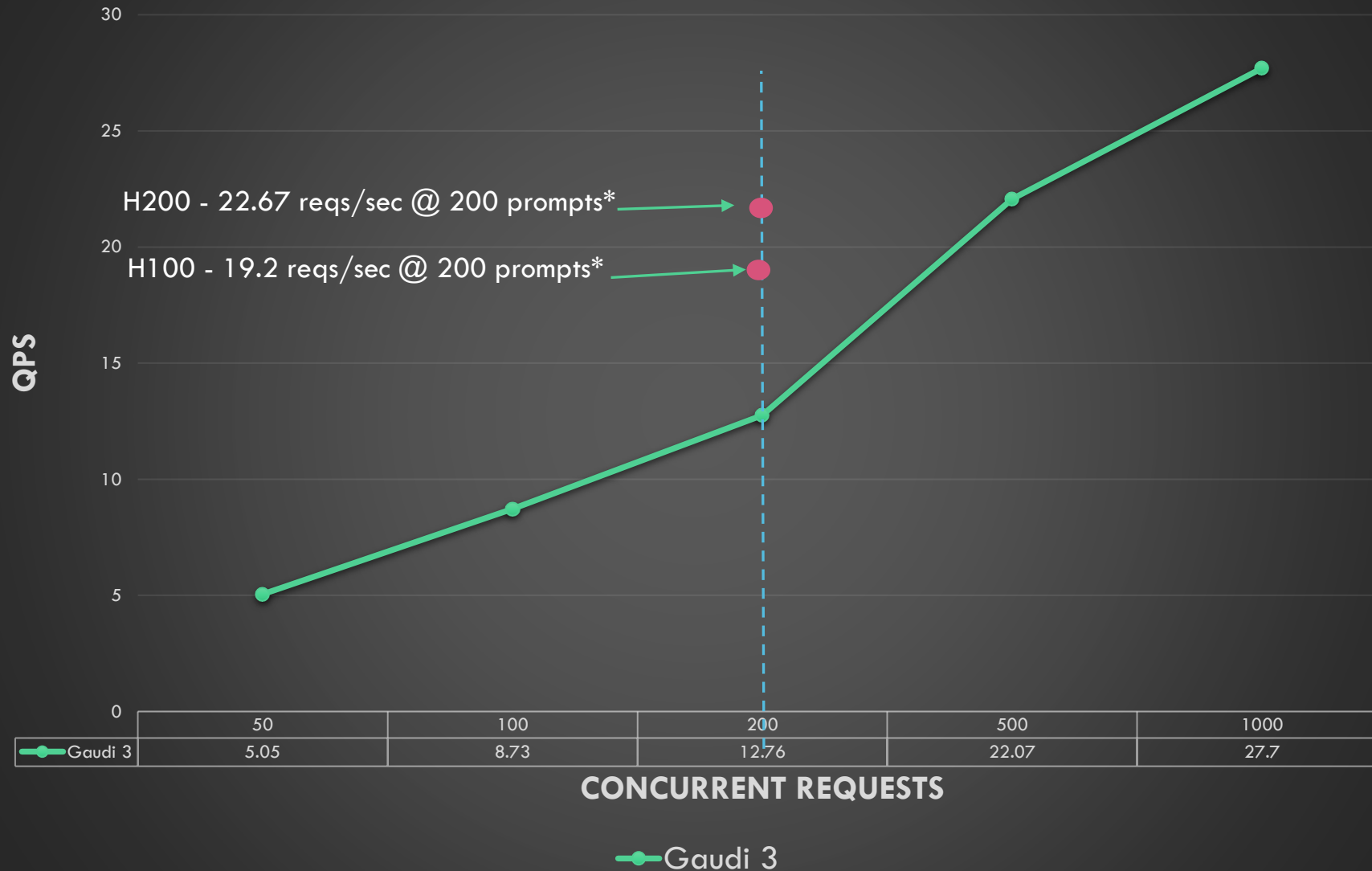


- Gaudi 3 yields a ~6X average speed-up as compared to Gaudi 2
- Higher is better

—▲— Gaudi 3 —x— Gaudi 2

QPS - LLAMA-8B LLM

Gaudi 3 vs. H200/H100 GPUs - Queries Per Second (QPS)

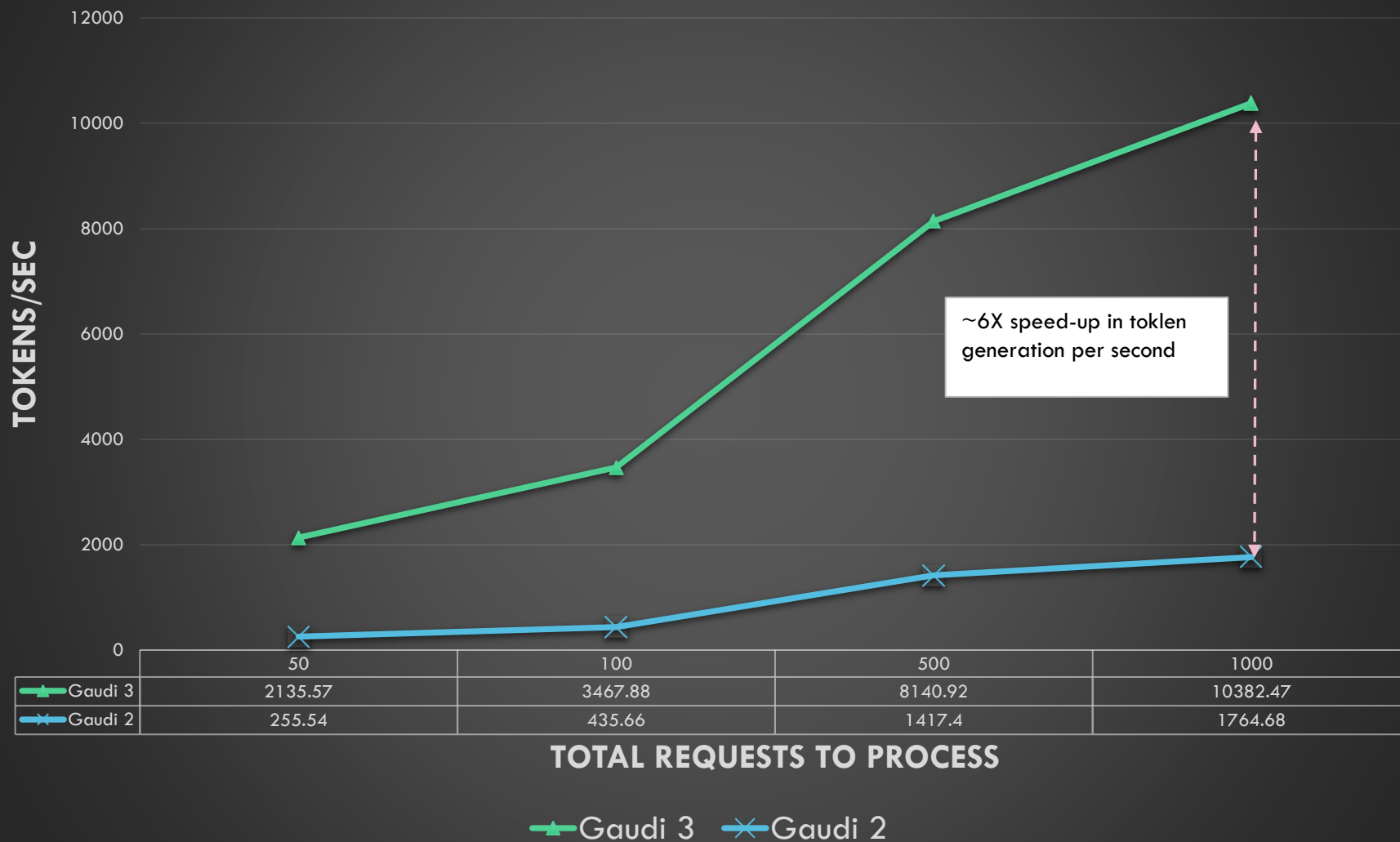


Benchmark Results from vLLM official benchmark website as of 3/2025:

<https://simon-mo-workspace.observablehq.cloud/vllm-dashboard-v0/perf>

- For this benchmark, Gaudi 3 outperforms H200 and H100 overall, with 27.7 requests/sec processed for 1000 input prompts
- Model running on 1 accelerator
- **Higher is better**

Gaudi 3 vs. Gaudi 2 - Total Throughput



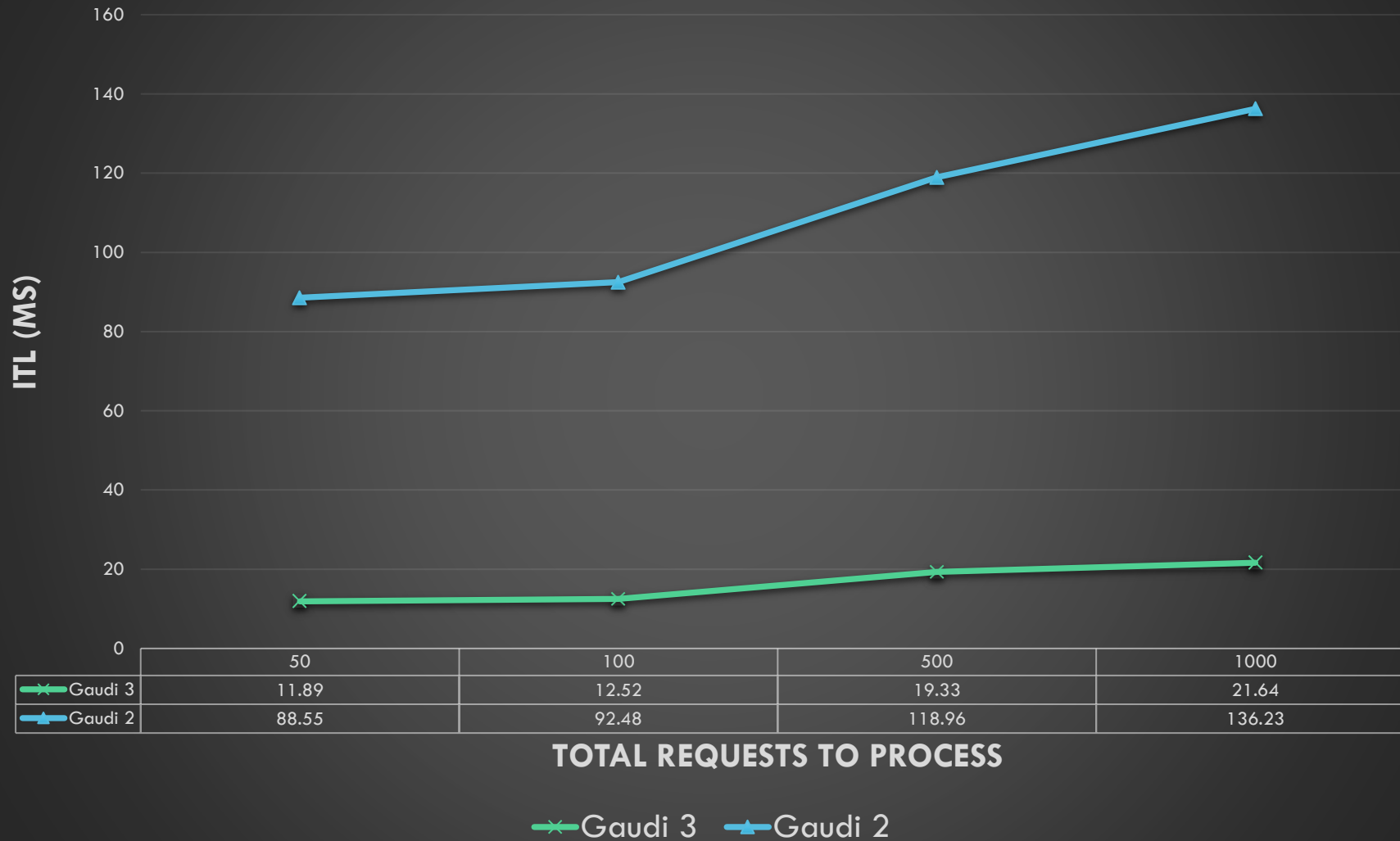
- LLM: Mistral 7B v0.3 Instruct
- Maximum throughput achieved is ~10,400 tokens/sec
- Gaudi 3 yields ~6X speed-up vs. Gaudi 2 in token generation/sec
- **Higher is better**

Gaudi 3 vs. Gaudi 2 - Time Per Output Token (TPOT)



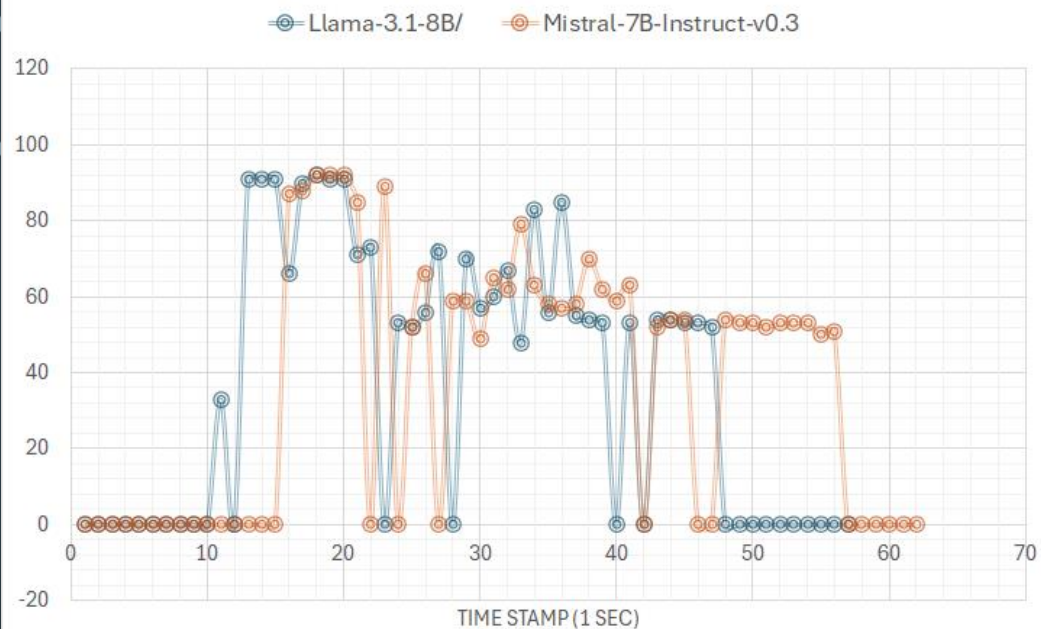
- LLM: Mistral 7B v0.3 Instruct
- Gaudi 3 has a TPOT below 50 ms even for 1000 concurrent requests
- Content is generated “faster than human can read”
- **Lower is better**

Gaudi 3 vs. Gaudi 2 - Inter-Token Latency (ITL)

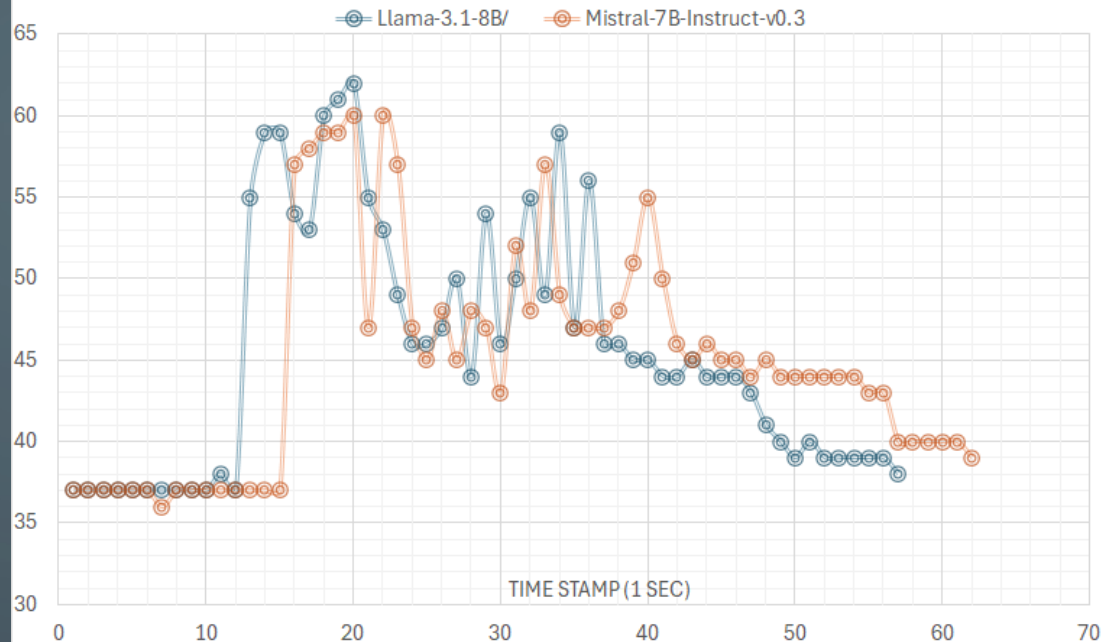


- LLM: Mistral 7B v0.3 Instruct
- Gaudi 3 yields ~6X speed-up vs. Gaudi 2
- Lower is better

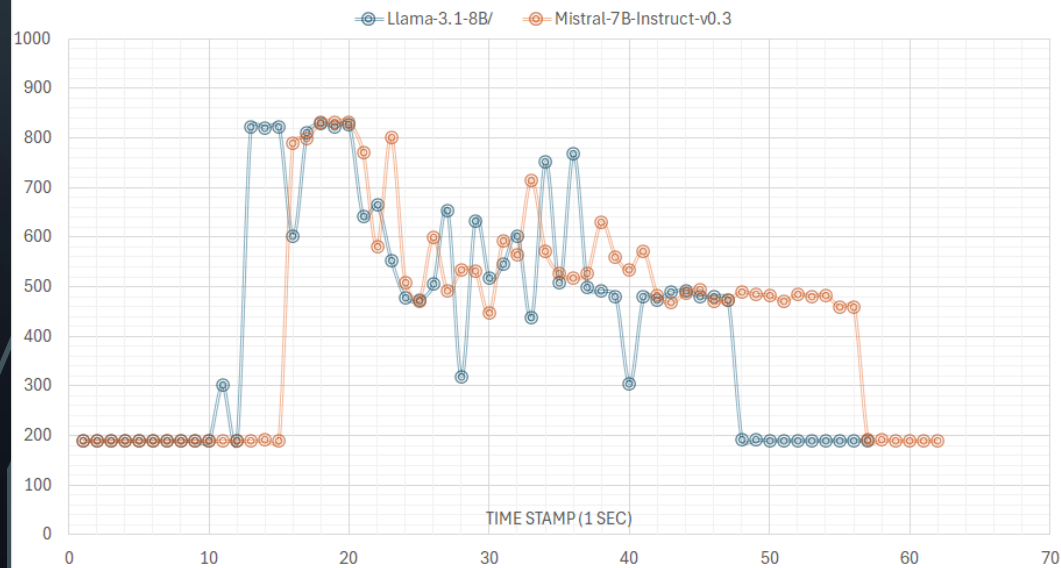
Utilization HPU %



Temperature (C)



Power Draw (W)



Mistral-7B-Instruct-v0.3 (1 HPU)

- Avg. Temperature: 45C
- Avg. HPU Utilization: 36%
- Avg. Power Draw: 440W or 0.05 Reqs/sec/W

Llama 3.1 8B (1 HPU)

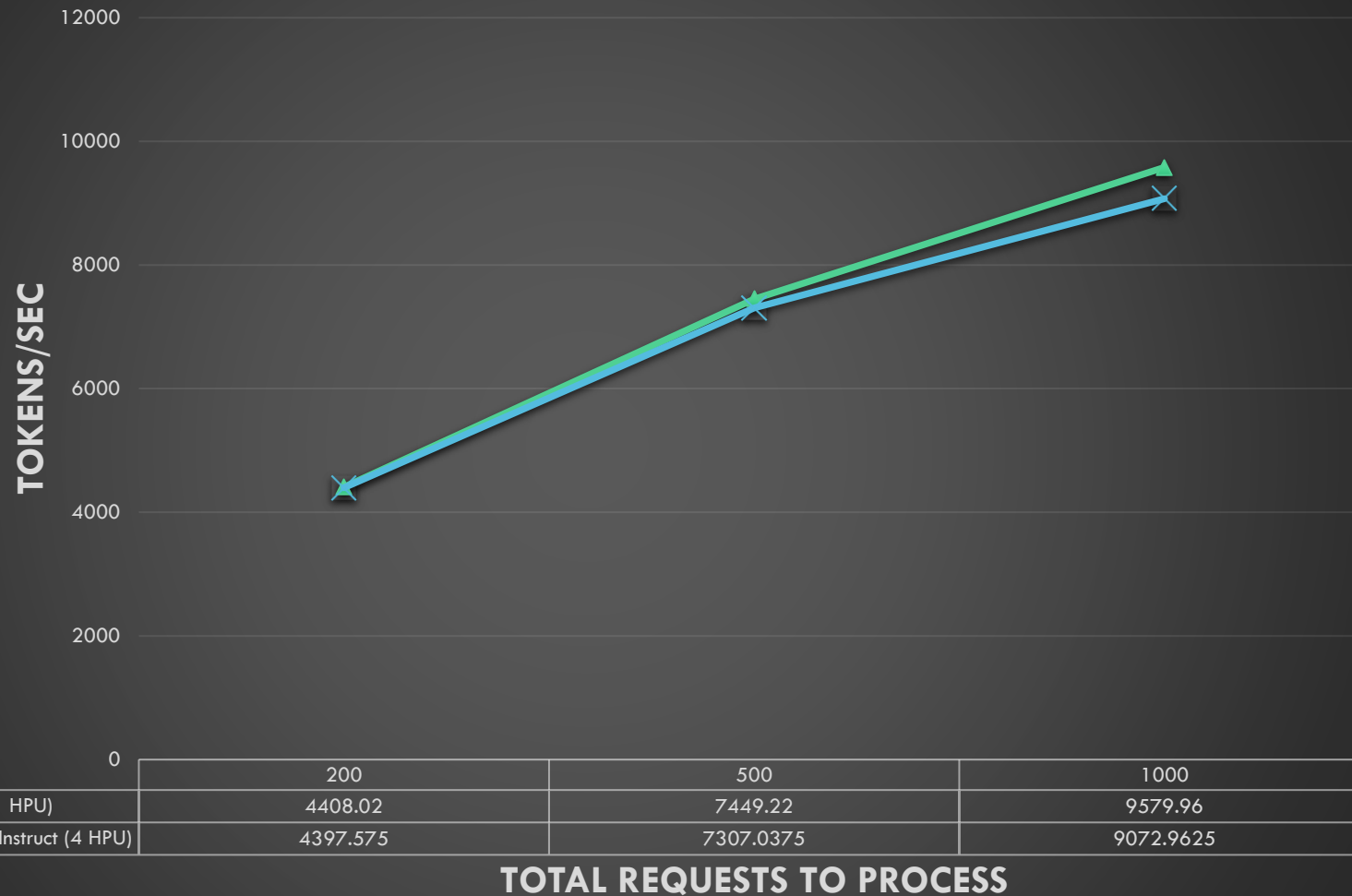
- Avg. Temperature: 46C
- Avg. HPU Utilization: 37%
- Avg. Power Draw: 436W or 0.06 Reqs/sec/W

EVALUATION

- Power draw measured for 1000 concurrent requests using Mistral 7B v0.3 Instruct
- Metric is the ratio of the number of requests served per second vs. average power draw (W)
 - Gaudi 3: 0.05 Reqs/sec/W
 - Gaudi 2: 0.025 Reqs/sec/W
- Result: Gaudi 3 HPUs are not just 6X more performant vs. Gaudi 2, but power efficiency is also increased by 2X, or put it differently, Gaudi 3 requires half the power of a Gaudi 2...
- Likely the explanation is the significant performance improvement for the same workload.

LLAMA-3.3-70B-INSTRUCT – PARALLELIZED ON MULTIPLE HPU(S)

Llama 8B vs. Llama 70B Total Throughput

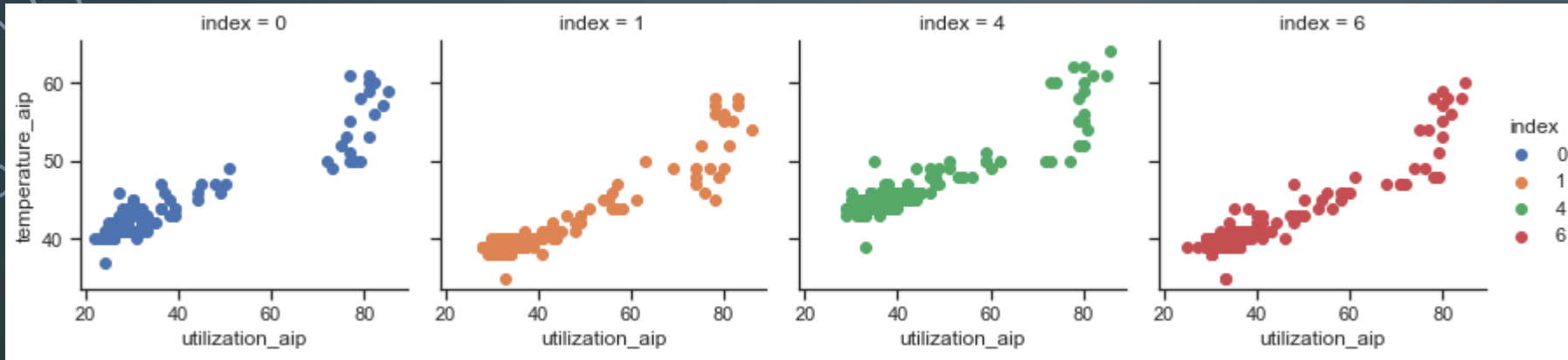


▲ Llama-3.1-8B (1 HPU) ✕ Llama-3.3-70B-Instruct (4 HPU)

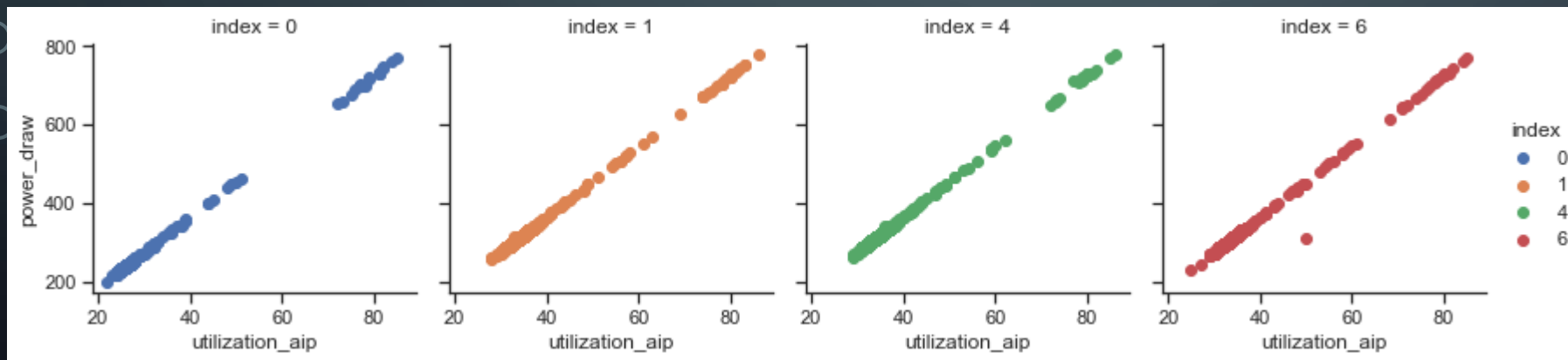
- Running a larger LLM, i.e., Llama-3.3 70B, sharded on 4 HPUs
- Objective is to evaluate scalability: compare total throughput on a “per model parameter basis” between Llama 3.1 8B and Llama 3.3 70B
- Llama 3.3 70B has roughly 8.75X more parameters than Llama 3.1 8B
- Throughput is normalized on a per-parameter basis for Llama 3.3 70B
- Good scalability also on larger models sharded on multiple HPUs

LLAMA-3.3-70B-INSTRUCT – PARALLELIZED ON MULTIPLE HPU(S)

Temperature (C) vs. HPU Utilization (%) for 4-HPU LLM Inferencing



Power Draw (W) vs. HPU Utilization (%) for 4-HPU LLM Inferencing Case



- Temperature profiles are very similar among HPUs indicating well balanced thermal management
- Temperature around 60-65C even for high utilization fractions
- Power draw is linear as a function of HPU utilization and consistent among all HPUs
- Linear power draw is efficient, and very similar across the board, indicating the system is balanced

SOME CONSIDERATIONS FROM A DEVELOPER PERSPECTIVE ON GAUDI 3...

- PyTorch and Tensorflow ports available from Habana toolkit
- GPU Migration toolkit simplifies migrating PyTorch models
- Habana Tools includes a TPC-C compiler, assembler, dis-assembler and all necessary headers

PyTorch Habana

```
import habana_frameworks.torch.core as htcore
model = torch.compile(model, backend="hpu_backend")
device = torch.device("hpu")
```

PyTorch CUDA

```
import torch
import torch_tensorrt
model = YourModel()
compiled_model = torch.compile(model, backend="tensorrt")
```

TensorFlow Habana

```
import tensorflow as tf
from TensorFlow.common.library_loader import load_habana_module
load_habana_module()

(x_train, y_train), (x_test, y_test) = tf.keras.datasets.mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0
model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(10),
])
loss = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
optimizer = tf.keras.optimizers.SGD(learning_rate=0.01)
model.compile(optimizer=optimizer, loss=loss, metrics=['accuracy'])
model.fit(x_train, y_train, epochs=5, batch_size=128)
model.evaluate(x_test, y_test)
```

CONCLUSIONS

- Performed an extensive set of benchmarks using real-world scenarios on Intel Gaudi 3
- Gaudi 3 is 6X more performant than Gaudi 2
- Gaudi 3 is 2X more power efficient than Gaudi 2
- On similar workloads Gaudi 3 (27.7 Reqs/s @1000 prompts) is equivalent (or better) to an Nvidia H200 (22.67 Reqs/s @200 prompts)
- Gaudi 3 has somewhat "high" idle power (almost 0% utilization)
 - Noticed that idle power seems to be a function of the memory footprint for the accelerator, i.e., larger memory footprint -> higher idle power
 - This situation may be not so out of the ordinary

FUTURE WORK

- Explore low-level capabilities of the Intel Gaudi 3 using the TPC-C compiler
- Attempt porting some relevant benchmarks from CUDA to Habana, i.e., C++

ACKNOWLEDGEMENTS

- Supermicro: Mr. Al Ford, Mr. Patrick Fallon
- Intel: Mr. Eric Eberline
- FDTS: Mr. Jerry Whitacre, Mr. Steven Pritchard