OPEN
AI + DATA
FORUM

@

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

HW, SW, **Performance and Costs** for Llama-2 70b and Mixtral 8x7b LLM Inference with Low Concurrency

Iván Baldo ibaldo@netlabs.com.uy

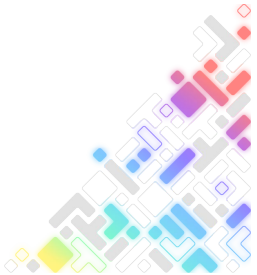
THE LINUX FOUNDATION #ossummit @handle

Introduction

- Iván Baldo - Research at Netlabs
- Been tasked recently to learn how to run Large Language Model inference and more
- I am still a beginner on this topic
- No data science background
- Was a programmer and later a systems administrator
- Lots of things to learn!
- LOTS OF THINGS TO LEARN!

Path to learning LLMs - Inference (Generation) part 1

- Several models (weights): Llama-2, Mixtral 8x7b 0.1, DBRX, Grok…
- Several sizes (parameters / weights) of those models: 7b, 13b, 70b, etc.
- Instructed (fine tuned for question / answering) and base (just completion of a given text)
- Quantization to reduce memory and speed up: Int8, FP8, NF4, QuIP#...
- Several runners: vLLM, Huggingface Transformers, ExLlama, MLC…
- + different ways and configurations for those
- Tensor (parts of layers) vs Pipeline (layers) vs Expert parallelism
- Speculative decoding: use a smaller model in parallel to speed up
- Continuous batching



Path to learning LLMs - Inference (Generation) part 2

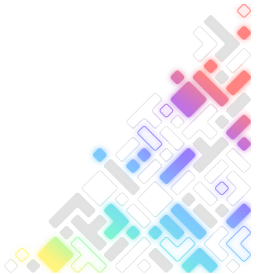
- Prompt engineering: ways to adapt the behavior and style of responses
- In-context learning: adding information before doing the actual query
- Understand the hardware needed to run them, batch and context sizes, etc.

Path to learning LLMs - RAG

- RAG - Retrieval Augmented Generation
- Chunking (splitting frases)
- Extracting data and storing embeddings + metadata in a Vector Database
- Embeddings are generated by another small LLM
- Search on the vector database
- Re-Ranking (another small LLM sorts again what was retrieved from the database)
- Frameworks like Langchain, LlamaIndex, etc.
- Custom functions for querying SQL databases and other sources

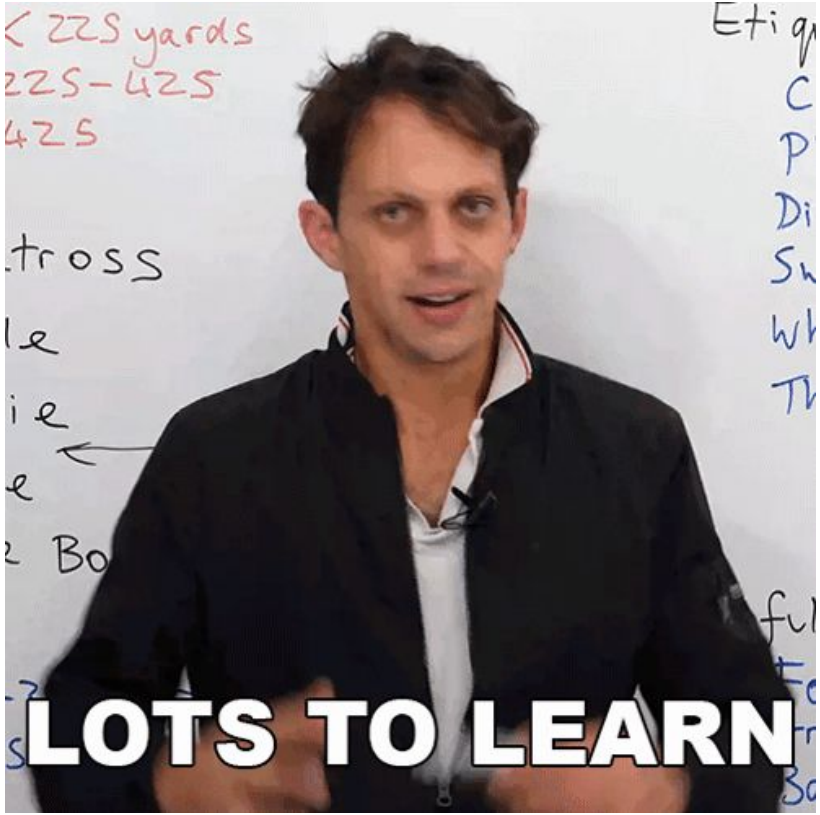
Path to learning LLMs - Fine-tuning

- Different techniques: regular fine tuning which alters the parameters directly, PEFT (Parameter Efficient Fine Tuning) implemented with LoRA (Low Rank Adaptation), etc.
- Data extraction and tokenization
- Evaluation of the resulting model: it may “forget” about it’s previous things or not learn successfully the new things
- Standard evaluation suites: MMLU, TriviaQA, GSM8K, HumanEval, AGIEval, HellaSWAG, ARC, Winogrande, TruthfulQA, etc.



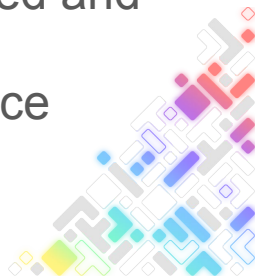
Path to learning LLMs - Multimodality

- Speech to text and text to speech
- Image understanding and generation
- Video
- Use a calculator or programs for math problems
- Physical body, sensors and actuators
- CONQUER THE WORLD!


LOTS TO LEARN

Define a mission and a scope

- Lots of reading and notes, but also need to actually do some stuff: benchmarking seemed good for this purpose
- Focus on current clients we have
- Low concurrency (at least at first they won't have too many users at the same time)
- Half of them prefer On Premises hardware because of data sovereignty, others prefer Public Cloud
- We recommend to use the Cloud as an utility provider: try to not use cloud vendor specific services and use more general and easily migratable ones
- Possibility to pay for required hardware to run Llama 2 70b unquantized and avoid any possible quality loss
- Mission: define how to run and on which hardware to run LLM inference



The Plan

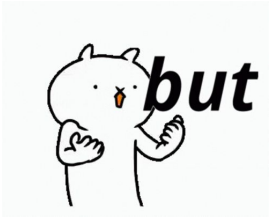
- Find benchmarks and characteristics of inference executors
- Discover what hardware is needed
- Found out https://github.com/hamelsmu/llama-inference by Hamel Husain through his blog https://hamel.dev/notes/llm/inference/03_inference.html
- Very similar to what I needed regarding low concurrency, but focused on small hardware like A6000 with 48G VRAM with a small Llama-2 7b model with quantization
- Decided to use his repo as base and contribute back: forked on https://github.com/ivanbaldo/llama-inference/
- He seems busy: 11 PRs from myself since December 2023 and not a single comment
- https://github.com/premAI-io/benchmarks found too late but more similar


ITS OKAY..... IM FINE
LOL

Start small (and cheap)

- Prepare tests and notes for running them for Llama-2 7b
- Requires about ~15G VRAM
- https://github.com/ray-project/llm-numbers says 2x the number of parameters at the native bfloat16 of Llama-2 + 1M per output token ~1G for 1024 tokens
- bfloat16 has more quality than float16 so CUDA 8.0 or later (Ampere or later)
- AWS has the g5.2xlarge with 32G RAM, 8vCPU, 1xA10G 24G VRAM at USD $1.2120 per hour in Virginia us-east-1 (April 2024)
- Didn't look at other clouds: cheap enough, already had AWS account with a good budget in it, etc.
- 220G storage required at the end: big container images (CUDA + PyTorch are big), model conversions to formats required for different executors…

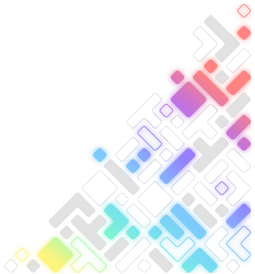
Why NVIDIA ?



- Not as much documentation, examples and support for AMD in December 2023
- Now in April 2024 there's much better documentation, framework support, more competitive hardware, etc.
- Although some reports of instabilities in drivers…
- Not so common on public clouds
- AMD is way more open than Nvidia: open firmware, drivers, etc.
- Hoping things get better at end of this year!!!
- There are other "exotic" hardware around: prefer more mainstream and supported things to recommend to our clients

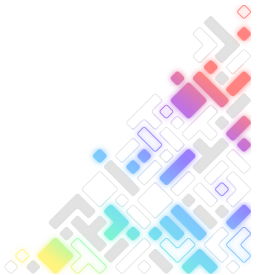
The tests

- Many use Conda but wanted something not Python specific + more flexible
- Need to use different packages versions and Linux distributions
- I LOVE CONTAINERS (you should too!)
- Decided to use Rocky Linux 9 with Podman using its CDI interface to access the HW
- Many clients use OpenShift and Kubernetes
- Did containers for most of Hamel's benchmarks
- Added new benchmarks too
- 9 questions, 1st one for warmup and not included in results
- Response token limit of 200



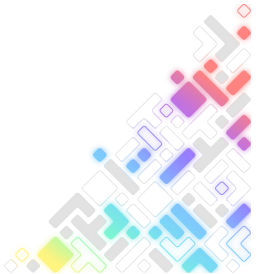
Server setup - basics

```
sudo dnf upgrade -y
sudo dnf install -y podman git epel-release bash-completion
sysstat pciutils
sudo dnf config-manager --add-repo
http://developer.download.nvidia.com/compute/cuda/repos/rhel9/$(
arch)/cuda-rhel9.repo
sudo dnf install -y nvidia-container-toolkit nvidia-driver
cuda-drivers-fabricmanager kmod-nvidia-latest-dkms
sudo dnf clean all
```

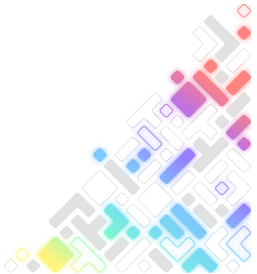
Server setup - considerations

- *cuda-drivers-fabricmanager* is when using multiple GPUs (found later about it after obscure errors…), optionally use *cuda-drivers* instead for single GPU
- Use DKMS: Nvidia kernel module not always up to date with the newest kernel, just compile it: also this is *dnf upgrade* friendly
- Nvidia-container-toolkit doesn't come with UDev rules or SystemD service to generate the CDI (Container Device Interface) for Podman, which is needed to access the GPUs
- There's lots of documentation in different places to do this, but at the end, it's simple
- Nvidia should improve these things…



Server setup - finishing touches 1

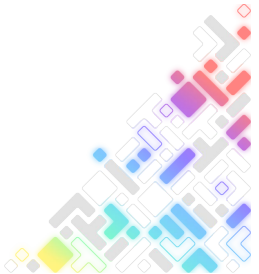
```
sudo sh -c 'cat >/etc/systemd/system/local-nvidia-ctk-cdi-generate.service'   <<EOT
[Unit]
Description=Regenerates /run/cdi/nvidia.yaml

[Service]
Type=oneshot
ExecStart=/usr/bin/nvidia-ctk cdi generate --output=/run/cdi/nvidia.yaml
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
EOT
```

Server setup - finishing touches 2

```
sudo systemctl enable --now nvidia-persistenced
nvidia-fabricmanager local-nvidia-ctk-cdi-generate
```

```
sudo reboot
```


4GIFs.com


LET'S
TASTE-TEST

Great success!

```
[rocky@ip-172-31-43-246 ~]$ podman run --rm --device nvidia.com/gpu=all --security-opt=label=disable nvcr.io/nvidia/cuda:12.3.2-runtime-rockylinux9 nvidia-smi

==========
== CUDA ==
==========

CUDA Version 12.3.2

Container image Copyright (c) 2016-2023, NVIDIA CORPORATION & AFFILIATES. All rights reserved.

This container image and its contents are governed by the NVIDIA Deep Learning Container License.
By pulling and using the container, you accept the terms and conditions of this license:
https://developer.nvidia.com/ngc/nvidia-deep-learning-container-license

A copy of this license is made available in this container at /NGC-DL-CONTAINER-LICENSE for your convenience.

Fri Apr 12 17:47:37 2024
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 550.54.15              Driver Version: 550.54.15      CUDA Version: 12.4     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA A10G                    On  |   00000000:00:1E.0 Off |                    0 |
|  0%   20C    P8              9W /  300W |       0MiB /  23028MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|  No running processes found                                                           |
+---------------------------------------------------------------------------------------+
```

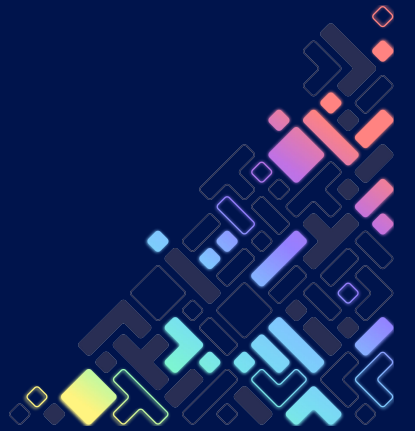
Defining a baseline: OpenAI API test

OpenAI API tokens / second from AWS N. Virginia…

title	mean	max	min	stddev	first%
OpenAI gpt-3	64.5	88.7	46.6	14.4	100.0
OpenAI gpt-4-turbo-preview	25.7	32.3	14.9	6.6	39.8

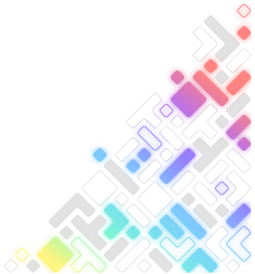
- 26 t/s seems accepted and reasonable to aim for

Llama-2 7b benchmarks on A10 (April 2024)



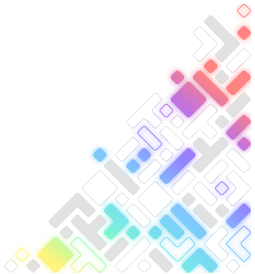
Huggingface Transformers 4.39.3

- with Accelerate 0.28.0, Optimum 1.18.0, Torch 2.2.2
- Apache 2.0 license
- Had to use torch_dtype=torch.bfloat16 in Ampere, otherwise it consumes twice as much VRAM (although bfloat16 is good for quality)



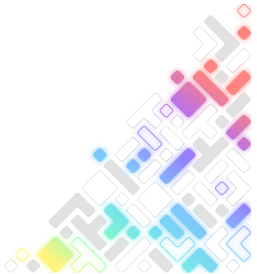
CTranslate2 4.1.1

- with CUDA 12.2, Torch 2.2.2
- MIT license
- Short-staffed but pretty good nonetheless



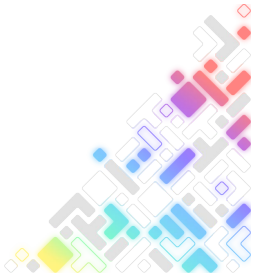
MLC AI 0.15.dev228 + MLC LLM 0.1.dev1068

- with CUDA 12.1, Torch 2.2.2
- Apache 2.0 license
- Seems to be near a stable release
- Currently disabling some features by default to make it more stable
- although slower than in December
- If those disabled features get re-enabled and considered stable, it’s one of the fastest



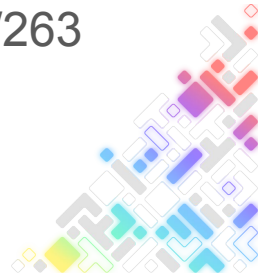
Huggingface Text Generation Inference (TGI) 1.4.5

- Recently reverted to Apache 2.0 license yay!
- Super easy to run from its official *huggingface/text-generation-inference* container!



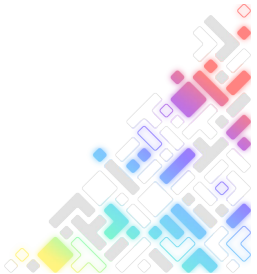
Nvidia Triton Inference Server 24.03 2.44.0 + TensorRT-LLM 0.8.0

- with CUDA 12.1, CUDNN 8.9, Torch 2.1.2
- BSD 3-clause and Apache 2.0 respectively
- Need to be careful with the versions used of both components: not all are synchronized to run together
- Even if using the latest versions of each, they may not be compatible
- The most difficult and time consuming to run of the tests
- https://github.com/triton-inference-server/tensorrtllm_backend/issues/246
- https://github.com/triton-inference-server/tensorrtllm_backend/issues/263
- https://github.com/triton-inference-server/server/issues/6807.



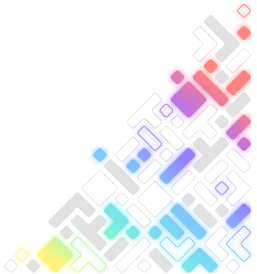
vLLM API Server 0.4.0

- Apache 2.0 license
- Super easy to run from its official *vllm/vllm-openai* container!
- Fast turnaround time for supporting newer models, techniques, etc.
- Popular choice in the ecosystem (subjectively)



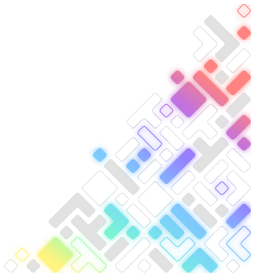
Text Generation WebUI (TGW)

- Using *atinoda/text-generation-webui* container snapshot-2024-03-31
- GNU Affero General Public License v3.0
- Backends tested: ExLlamav2, ExLlamav2_HF, Transformers, Transformers --bf16, Transformers --use_flash_attention_2
- Besides a good web, it provides an OpenAI compatible API

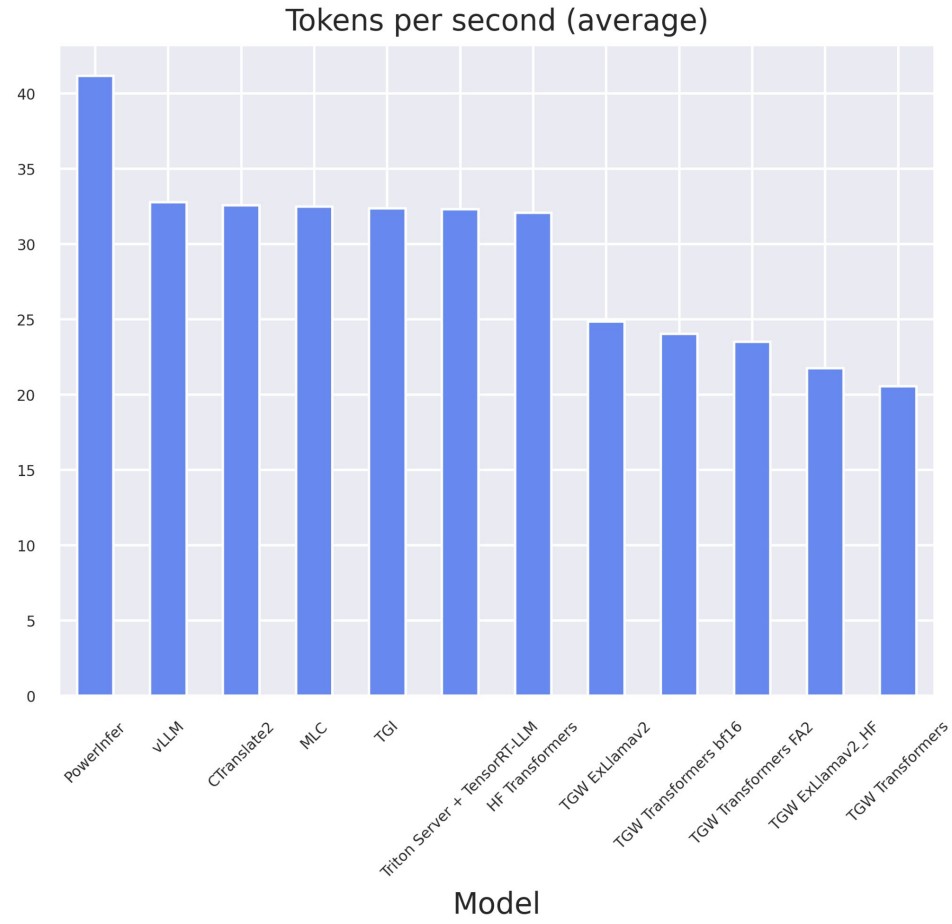


PowerInfer 0.0.1 build 1577 b478398

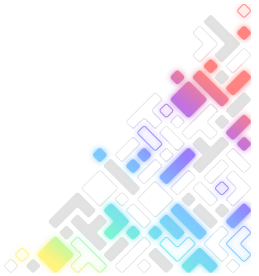
- with CUDA 12.1, CUDNN 8.9, Torch 2.2.2
- MIT license
- The models are retrained to a ReLU activation function, so not exactly the same to the original
- in the case of Llama-2 70b there's significant loss of quality (GPU time-money constraints I guess)
- although for Llama-2 7b not so much
- It's interesting but not so fair to compare to the others which retain full quality
- + also has some parts of the processing on the CPUs.



And the winner is…


Tokens per second (average)
40
35
30
25
20
15
10
5
0
PowerInfer
vLLM
CTranslate2
MLC
TGI
Triton Server + TensorRT-LLM
HF Transformers
TGW ExLlamav2
TGW Transformers bf16
TGW Transformers FA2
TGW ExLlamav2_HF
TGW Transformers
Model

title	mean	max	min	stddev	first%
PowerInfer	41.2	42.8	35.9	2.3	100.0
vLLM	32.8	32.8	32.7	0.0	79.6
CTranslate2	32.6	32.8	31.3	0.5	79.1
MLC	32.5	33.2	31.7	0.5	78.9
TGI	32.4	32.4	32.3	0.0	78.6
Triton Server + TensorRT-LLM	32.3	32.5	32.0	0.2	78.5
HF Transformers	32.1	40.8	28.5	3.9	77.9
TGW ExLlamav2	24.9	30.4	17.6	5.7	60.4
TGW Transformers bf16	24.1	26.0	15.3	3.6	58.4
TGW Transformers FA2	23.5	26.2	14.0	4.0	57.1
TGW ExLlamav2_HF	21.7	28.7	2.6	9.5	52.8
TGW Transformers	20.6	26.1	6.4	8.1	49.9



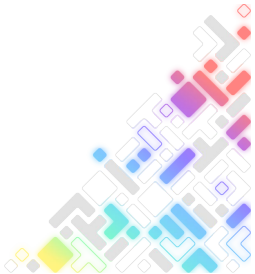
GPU usage columns

- InfVRAM: sum of all the GPUs memory used during inference
- MaxVRAM: maximum sum of all the GPUs memory used since the start of the test
- InfVRAMBW%: average percentage of all the GPUs time the RAM was busy during inference
- InfMaxSinglVRAMBW%: the maximum percentage of all the GPUs RAM that was busy during inference
- InfGPU%: average GPUs compute usage during inference
- InfMaxSinglGPU%: maximum of a single GPU usage during inference

title	InfVRAM	MaxVRAM	InfVRAMBW%	InfMaxSinglVRAMBW%	InfGPU%	InfMaxSinglGPU%
HF Transformers	13604	13604	94	94	97	97
TGW Transformers	13670	13368	94	94	96	96
TGW Transformers FA2	13690	13550	95	95	97	97
TGW Transformers bf16	13692	13592	94	94	96	96
CTranslate2	13712	13520	100	100	100	100
Triton Server + TensorRT-LLM	14838	14838	100	100	97	97
PowerInfer	14864	14856	85	85	95	95
TGW ExLlamav2	14994	14994	100	100	99	99
TGW ExLlamav2_HF	15022	15022	100	100	98	98
MLC	15884	15884	100	100	96	96
vLLM	20406	20386	100	100	97	97
TGI	21138	21138	100	100	97	97

CPU RAM usage columns

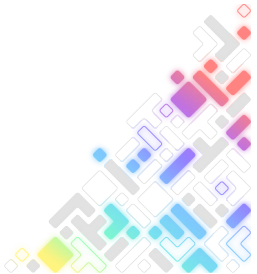
- MaxMem: maximum amount of CPU RAM used since the start (loading model, on the fly conversions, etc.) until the end
- InfMem: actual memory used during inference (near the end of generation)



title	MaxMem	InfMem
PowerInfer	1776	1774
MLC	1796	1743
TGI	2163	2072
TGW ExLlamav2	2277	1983
TGW ExLlamav2_HF	2352	1043
TGW Transformers	2524	1052
TGW Transformers FA2	2531	2225
vLLM	6945	5655
CTranslate2	11989	1435
TGW Transformers bf16	13615	2224
HF Transformers	14090	1986
Triton Server + TensorRT-LLM	14443	842

CPU usage columns

- MaxCPU: maximal percentage of a single CPU usage during inference
- InfCPU: average percentage of all CPUs usage during inference



title	InfCPU	MaxCPU
TGW ExLlamav2	5.8	100
TGW Transformers FA2	10.9	100
TGW Transformers bf16	12.1	100
TGW Transformers	13.0	100
HF Transformers	13.0	100
MLC	13.0	100
vLLM	13.0	100
TGI	13.0	100
CTranslate2	13.0	100
TGW ExLlamav2_HF	13.1	100
Triton Server + TensorRT-LLM	13.2	100
PowerInfer	50.7	100

Observations on Llama-2 7b benchmarks in the A10G

- A group of 32 tokens/s runtimes: vLLM, CTranslate2, MLC, TGI, Triton TensorRT, HF Transformers (GPT-4 is 26 so this is good)
- TGW for some reason is slow maxing out at 25 tokens/s, even when using HF Transformers as backend: didn't had time to investigate, maybe old versions or missing packages in the Atinoda container or bad configuration?
- MLC wins in low CPU memory usage at 1.8G
- vLLM and TGI reserving more VRAM for context and batching I think but probably they can run with 16G VRAM too.
- All limited by VRAM bandwidth at these single concurrency / single batch tests.

Benchmarks that didn't work

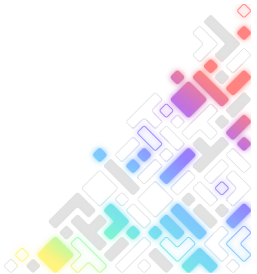
- Candle VLLM
- Uses Huggingface Candle Rust library + additions
- https://github.com/EricLBuehler/candle-vllm/issues/36 "Doesn't compile"
- MIT license

- Mistral.RS
- Uses Huggingface Candle Rust library + additions
- https://github.com/EricLBuehler/mistral.rs/issues/44 which is about using too much VRAM
- Seems to be the successor to Candle VLLM
- MIT license

Missing benchmarks / roadmap

Due to lack of time, we couldn’t make tests for everything:

- Burn
- DeepSpeed
- Lightning
- ONNX Runtime
- Tinygrad
- More suggestions? (FOSS ones only)



Choosing hardware for Llama-2 70b

Llama-2 70b hardware definition needs

- Goal of validating the HW first before buying: use what's available on public clouds and intersect with what's available from On Premise HW providers
- 70b x 2 + 4096 tokens of context at 1M per token (4G) x 4 concurrent users
- ~ 156G VRAM needed
- Most server HW providers don't publish prices, Dell has some though!

Llama-2 70b hardware definition strategy

- 8xL4 24G each / 192G total is the cheapest publicly at Dell at USD $4200 each / $33600 total, that is also available in Google Cloud as g2-standard-96 384 RAM 96 vCPU at $8.004/hr in Northern Virginia us-east4 (April 2024).
- No NVLink, each one only 300GB/s memory bandwidth, 121 TFLOPS BF16
- Probably not good enough speed
- … but let's test anyway!

The Dell 8xL4 server USD 65k

PowerEdge R760xa Rack Server Summary

Back to Customization

Dell Price $65,021.57

Add to Cart

View Delivery Dates

The Dell 8xL4 server 24C/48T 256G DDR5-4000

Components

Option	Selection	SKU / Product Code	Quantity
Base	PowerEdge R760XA Server	[210-BGRR] / GEVL8I7	1
Trusted Platform Module	Trusted Platform Module 2.0 V3	[461-AAIG] / GGX1VDO	1
Chassis	2.5" Chassis with up to 8 SAS/SATA Drives, Front PERC 11	[321-BJKF] / GSI316X	1
Processor	Intel® Xeon® Silver 4410Y 2G, 12C/24T, 16GT/s, 30M Cache, Turbo, HT (150W) DDR5-4000	[338-CHSG] / GBDDC9W	1
Additional Processor	Intel® Xeon® Silver 4410Y 2G, 12C/24T, 16GT/s, 30M Cache, Turbo, HT (150W) DDR5-4000	[338-CHSG][379-BDCO] / GZC0HYT	1
Processor Thermal Config	Heatsink for 2 CPU Configuration with OCP	[412-BBCP] / GZ54VFA	1
Memory Configuration	Performance Optimized	[370-AAIP] / GH9QBEI	1
Memory DIMM Type and Speed	4800MT/s RDIMMs	[370-AHCL] / GGBEML0	1
Memory	16GB RDIMM, 4800MT/s Single Rank	[370-AGZO] / GAD7JYE	16

The Dell 8xL4 server RAID 1 800G SSD 24Gbps

RAID	C3, RAID 1 for 2 HDDs or SSDs (Matching Type/Speed/Capacity)	[780-BCDN] / GOV1697	1
RAID/Internal Storage Controllers	Front PERC H355 Rear Load	[405-ABCQ][750-ADWP] / G90FD1E	1
Storage	800GB SSD SAS Mixed Use up to 24Gbps 512e 2.5in Hot-Plug, AG Drive	[345-BEPV] / GQCU8YK	2
Power Management BIOS Settings	Power Saving Dell Active Power Controller	[750-AABF] / G06TYXW	1
Advanced System Configurations	UEFI BIOS Boot Mode with GPT Partition	[800-BBDM] / GSFTG4Y	1
Advanced System Configurations	No Energy Star	[387-BBEY] / G9IONL3	1
Fans	Gen 2 Fan	[750-BBCC] / GUAQJ0D	1
Power Supply	Dual, Hot-Plug, Fully Redundant Power Supply (1+1), 2400W, Mixed Mode	[450-AJEV] / GCZ2T8R	1
Power Cords	PowerCord, 250V, 2FT, C19/C20, US	[450-AEIT] / GQKO80Y	2
PCIe Riser	Riser Config 0, 4x16 FH Slots (Gen5), 4x16 FH DW GPU Capable Slots (Gen5)	[330-BCFY] / GC92WG5	1

The Dell 8xL4 server 2x10GbE + 1GbE

Embedded Systems Management (Multi)	iDRAC9, Express 16G	[528-CTIJ] / G78FVSP	1
OCP 3.0 Network Adapters	Broadcom 57416 Dual Port 10GbE BASE-T Adapter, OCP NIC 3.0	[540-BCOD] / G91LG8U	1
Additional Network Cards	Broadcom 5720 Dual Port 1GbE LOM	[540-BDKD] / GY8XSA9	1
GPU Acceleration Cards	NVIDIA L4, PCIe, 72W, 24GB Passive, Single Wide Full Height GPU	[490-BJRP] / G1UIYQ5	8
Bezel	PowerEdge 2U LCD Bezel	[325-BFCF][350-BCML] / GLWQH7O	1
Boot Optimized Storage Cards	BOSS Blank	[329-BERC] / G74DI3A	1

Accessories

Option	Selection	SKU / Product Code	Quantity
Rack Rails	ReadyRails Sliding Rails With Strain Relief Bar	[770-BFCF][770-BFCG] / G7HVFWT	1

The Dell 8xL4 server 5 years warranty

Support Services

Option	Selection	SKU / Product Code	Quantity
Protect your purchase - View Support offers below	Basic Next Business Day 36 Months, 36 Month(s)	[709-BBFN] / GH1DGNF	1
Extended Services	Basic Next Business Day, 60 Month(s)	[865-BBNP] / G01749W	1


IT'S JUST LOVELY

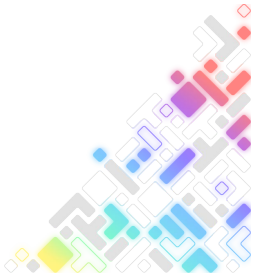
And the winner is…

We don’t know!

- Google Cloud declined our request to increase our account quota from 1 L4 to 8 L4 GPUs to be able to launch the g2-standard-96 server…
- We (Netlabs) even are partners with them…

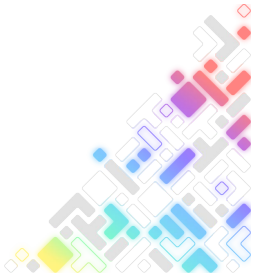
Finding the alternative…

- Looked at eBay for sellers with good reputation and new A100 GPUs
- They range from USD $9K to $10K for the 40G VRAM PCIe version and exactly twice that for the 80G version
- The 80G version is faster and we only need two
- No GPUs below $40K on Dell that are also on a public cloud
- We can test the 40G version EASILY (remember this…) in AWS
- We ignore Google, and we didn’t had too much time to try on Azure
- (been overly optimistic about the timings for preparing this talk)
- So the 4xA100-40G GPUs are the chosen ones



Finding the A100 40G price…

- Asked Dell for pricing for servers with A100's but they require an USA shipping address and landline phone, which we don't have, and they refused to send the info (and we didn't want to bother our USA clients)
- The previous Dell server without the 8xL4 but with the 4xA100-40G bought from eBay ends up being USD 71.4K
- Need to add some $$$ for the NVLink bridges though ~ USD 72K



The alternative is not bad!

- The A100-40G PCIe version has 1555 GB/s memory bandwidth and 312 BF16 TFLOPS each
- Compare that to the L4 300 GB/s and 121 TFLOPS!
- 5x memory bandwidth and 2.5x TFLOPS
- with a modest 10% total increase in price of the server
- And we get NVLink between the cards
- AWS has the p4d.24xlarge 1152 RAM 96vCPU 8xA100-40 server at USD $32.7726/hr in North Virginia us-east-1
- But we will only use 4 of those 8 of course (super easy with containers)!

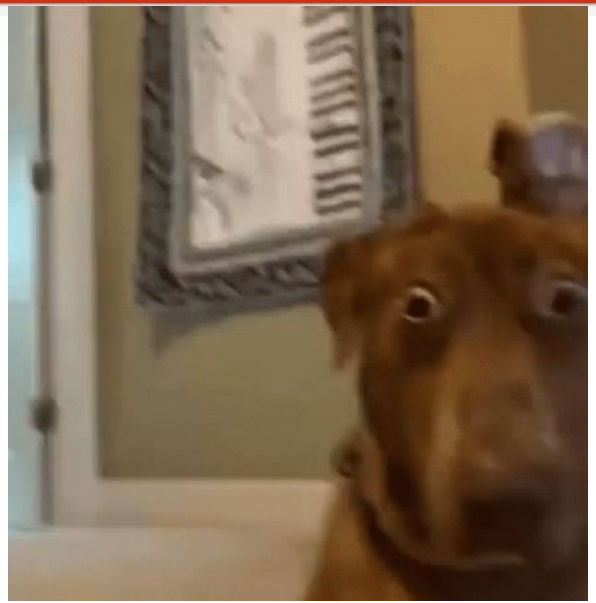

IT'S JUST LOVELY


LET'S
TASTE-TEST

And the winner is…

Not so fast!!!

Failed to start the instance i-0e994da3f4bc52e41
Insufficient capacity.



Even hyperscalers have issues securing AI HW!

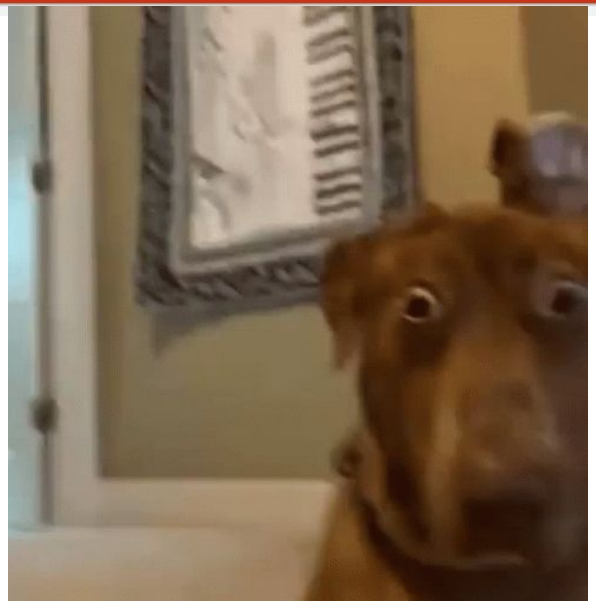
- Tried different parameters, times of day and days unsuccessfully
- Then tried different regions (and not all regions have the p4d.24xlarge)
- Testing from Uruguay  , South America, so from nearest to farthest:
- N. Virginia us-east-1
- Ohio us-east-2
- Oregon us-west-2
- Ireland eu-west-1
- Frankfurt eu-central-1
- Seoul ap-northeast-2
- Tokyo ap-northeast-1

Long live Seoul, South Korea 🇰🇷 !!!

- Price increases from $32.77 to $45.39/hr, but nevermind…
- Strange (and not so useful) errors from a couple of inferencers
- After some trying found out that we need to install cuda-drivers-fabricmanager and start the nvidia-fabricmanager service to setup the NVLink switches
- Downloading the Llama-2 70b model 138 gigabytes isn't that fast
- Did a new benchmark for Huggingface Transformers since it needs special handling for multiple GPUs with Accelerate but at the end didn't work, skipped…
- CTranslate2 worked after running it with mpirun: this was simple
- Let's continue the next day…

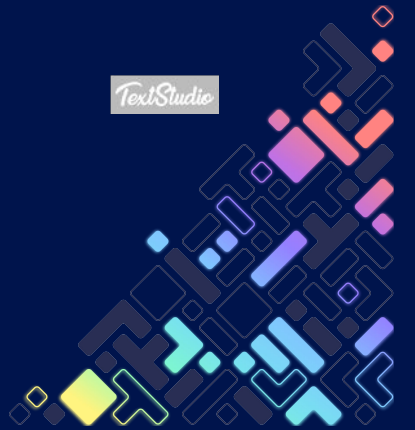
… the next day …

Failed to start the instance i-0e994da3f4bc52e41
Insufficient capacity.




IM SORRY
My Apolocheese..

And the winner is…PLEASE



Tokens per second

title	mean	max	min	stddev	first%
CTranslate2	22.9	23.3	20.4	1.0	100.0

GPU usage

	title	InfVRAM	MaxVRAM	InfVRAMBW%	InfMaxSinglVRAMBW%	InfGPU%	InfMaxSinglGPU%
CTranslate2	141548	141548	67	67	94	94

CPU RAM

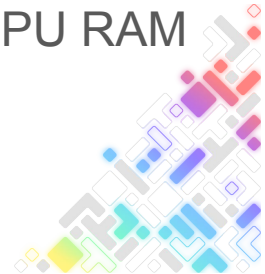
title	MaxMem	InfMem
CTranslate2	140923	10529

CPU compute

title	InfCPU	MaxCPU
CTranslate2	4.0	100

Observations

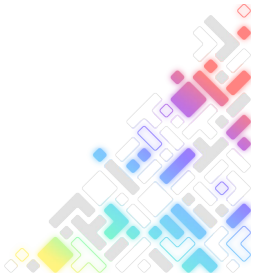
- 23 tokens per second is comparable to OpenAI's GPT-4 26 t/s, this is good
- CTranslate2 is being bottlenecked in the GPU compute, I wasn't expecting this, too bad we can't compare with the others to see if they too are bottlenecked the same way or achieve higher speed by avoiding it a bit
- Mixture of Experts (MoE) models are faster since for example for Mixtral 8x7b it runs 2 7b models (experts) simultaneously per each token and don't need to read the whole VRAM and do all the math: higher quality and faster
- So expect way higher than 23 tokens per second for MoE's
- CTranslate2 uses CPU RAM for the whole model (use servers with CPU RAM > GPU VRAM)



Some pricing considerations

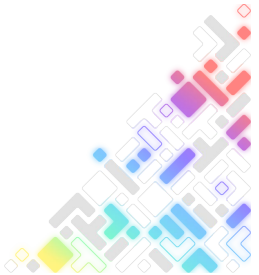
OpenAI vs On Premises HW inputs

```
# Per prompt:
itokensavg = 100
otokensavg = 400
hwcost = 72000 #Dell 4xA100-40G 2024-04-05
hwuseyears = 5
usedhoursperday = 8
# 2024-04-13 USD per 1k tokens https://openai.com/pricing:
gpt4turbokitokenprice = 10 / 1000
gpt4turbokotokenprice = 30 / 1000
gpt35turbokitokenprice = 0.5 / 1000
gpt35turbokotokenprice = 1.5 / 1000
```

OpenAI vs On Premises HW computations

```
usedhoursperweek = 5 * usedhoursperday
weeksperyear = 365.25 / 7
hwusetotalhours = usedhoursperweek * weeksperyear * hwuseyears
hwcostperhour = hwcost / hwusetotalhours

gpt4turbocostperprompt = gpt4turbokitokenprice / 1000 * itokensavg + gpt4turbokotokenprice / 1000
* otokensavg
gpt35turbocostperprompt = gpt35turbokitokenprice / 1000 * itokensavg + gpt35turbokotokenprice /
1000 * otokensavg
gpt4turbopromptsperhour = hwcostperhour / gpt4turbocostperprompt
gpt35turbopromptsperhour = hwcostperhour / gpt35turbocostperprompt
gpt4turbopromptsperminute = gpt4turbopromptsperhour / 60
gpt35turbopromptsperminute = gpt35turbopromptsperhour / 60

f"{hwcostperhour=} {gpt4turbopromptsperhour=} {gpt35turbopromptsperhour=}
{gpt4turbopromptsperminute=} {gpt35turbopromptsperminute=}"
```

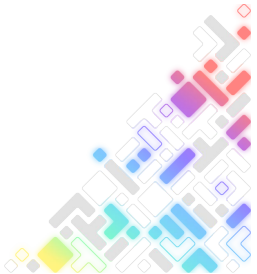



OpenAI vs On Premises HW cost analysis

- Each used working hour of on premises HW is equivalent to USD 6.9
- That corresponds to 531 GPT-4-Turbo prompts per hour, 8.8 per minute
- Or to 10614 GPT-3.5-Turbo prompts per hour, 177 per minute
- If you have more than 177 concurrent users then Llama-2 70b or Mixtral 8x7b are good alternatives to GPT-3.5-Turbo since they are at the same level of output quality
- But it’s unrealistic because that would require more VRAM
- 89G for 512 tokens context per batch/user
- So cost isn’t a reason to use on premises hardware

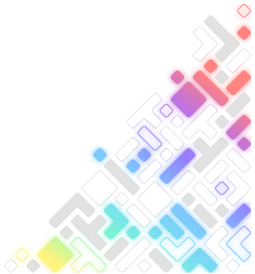
Reasons for using On Premises HW

- Data sovereignty: some businesses (both private and gubernamental) require some data to never leave their own data centers, but others also while not obliged prefer to have the data on-site
- Be free from external providers
- Have fixed costs
- RAG and other uses require several prompts which are also bigger so that steers the balance more favorably to the local HW
- Faster response since avoiding round trip times from LLM to Vector DB to LLM again etc.



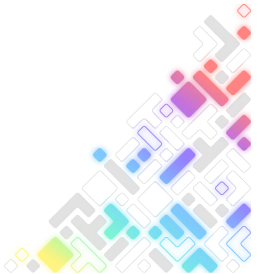
What we learned - 1

- Testing LLMs is time consuming and full of surprises, expect 10x your estimated time
- Documentation sometimes obsolete: Huggingface BetterTransformer obsolete class, made a new test with that and later found out it wasn't necessary any longer, nowadays it mentions this though (so it improved!)
- and sometimes with incomplete or unclear examples
- and sometimes spread out in different parts or even sites: Nvidia drivers install, CUDA install, Container Toolkit install, and then one has to bring the pieces together…



What we learned - 2

- Sometimes different code or running parameters for multi-GPU
- Compiling CUDA kernels TAKES AGES!
- Big container images (Pytorch is big, CUDA is big, etc.)
- You may run out of RAM and even crash your server or desktop while running tests
- but also when compiling them: Rust's Cargo launches processes in parallel and some Crate also launches processes in parallel without regard for RAM (disabled CPUs to workaround while trying to avoid losing too much concurrency and speed)

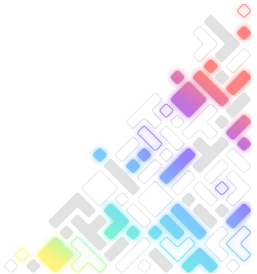


What we learned - 3

- Models need space, need to be downloaded, for some execution engines they need to be converted, it takes time…
- GPU scarcity is real, even for hyperscalers
- hope AMD’s future adoption helps alleviate this and appear on public clouds
- AWS allows to reserve usage slots (for a price of course!) but they are 3 days in advance on Virginia
- Others may not even allow you to use the GPUs you need!
- On Premises LLM inference with >GPT-3.5 quality at good speed is possible
- For low concurrency opt for high memory bandwidth on the GPU

What we learned - 4

- Containers are great to untangle the host server and guest containers, giving flexibility, easy of use and security (rootless and isolation), and all without loss of any performance
- One of the tests had trouble linking with Rocky Linux 9, 8 and Ubuntu 23.10, it worked with Ubuntu 22.04: doing this with containers is easier and doesn't need any changes on the host



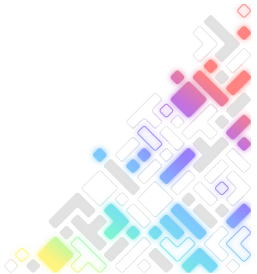
Disabling CPUs to avoid OOM while compiling

Disable the hyperthreads and not full cores:

```
for c in 1 3 5 7 9 11 12 13 14 15; do
    echo 0 >/sys/devices/system/cpu/cpu$c/online
done
```

Re-enable:

```
for c in /sys/devices/system/cpu/cpu*/online; do
    echo 1 >$c
done
```

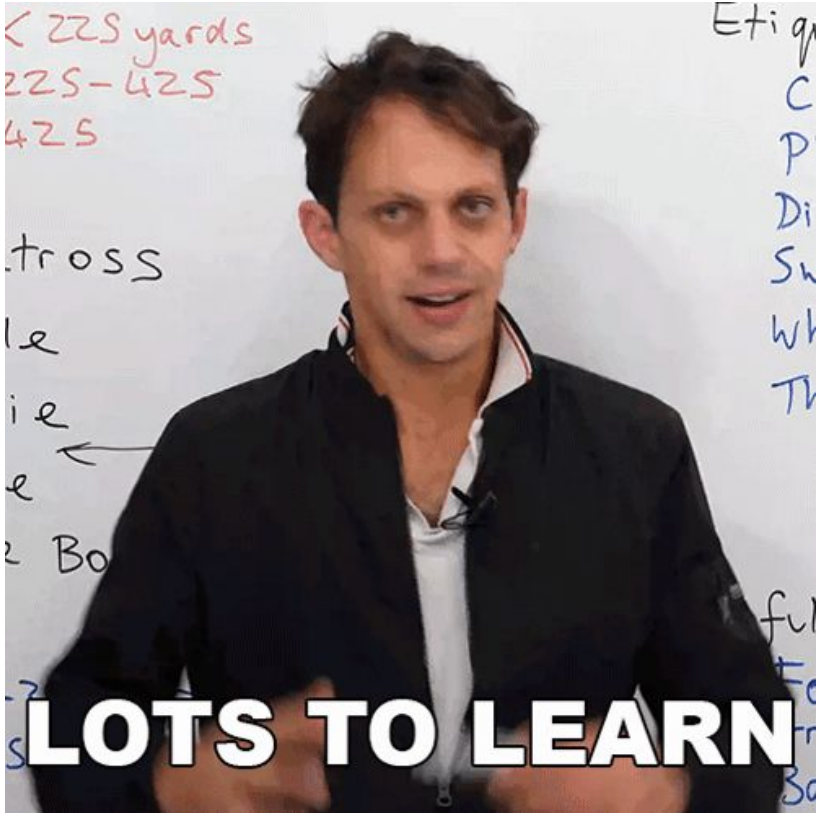



CTranslate2 benchmark example model conversion

```
podman build --build-arg USERID=$(id -u) -t local/ctranslate-bench ctranslate

PODMAN_DEVICES=nvidia.com/gpu=0,nvidia.com/gpu=1,nvidia.com/gpu=2,nvidia.com/gpu=3
HF_MODEL=meta-llama/Llama-2-70b-hf

podman run --rm -it -v$HOME/.cache/huggingface/:/home/user/.cache/huggingface/:Z\
   -v$PWD:/home/user/llama-inference:Z -e CT2_VERBOSE=1 local/ctranslate-bench \
   sh -c 'cd /home/user/llama-inference/ctranslate && ct2-transformers-converter \
   --model '$HF_MODEL' --output_dir llama-2-7b-ct2'
```

CTranslate2 benchmark example run

```
sed -i 's@from_pretrained(".*/.*")@from_pretrained("$HF_MODEL'")@'
ctranslate/bench.py
# Don't use mpirun for single device, and change 4 by the number of GPUs:
testStart "CTranslate2"; \
podman run --rm -it -v$HOME/.cache/huggingface/:/home/user/.cache/huggingface/:Z\
   -v$PWD:/home/user/llama-inference:Z --device $PODMAN_DEVICES
--security-opt=label=disable \
   -e CT2_VERBOSE=1 local/ctranslate-bench \
   sh -c 'cd /home/user/llama-inference/ctranslate && mpirun -np 4 python3 bench.py';
\
testEnd ctranslate/bench-ctranslate-int8.csv
```

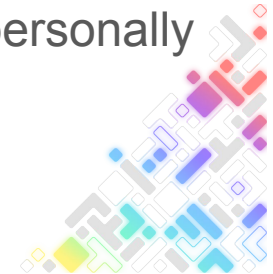

LOTS TO LEARN

This is obsolete already!

- Mixtral 8x22B base model released the 9th April 2024, requires 286G VRAM + context, 4xA100-80 should be enough
- So the Dell server would be USD 112K to run it
- Rumors of Llama-3 getting released this May… MoE (probably)? Size?
- In case of buying a server and then in the future if wanting to change to a newer model, quantizing to Int8 at first, then NF4, then CuIP# (2.3 bits per token avg) could help
- AI and LLM's are a fast moving space both in software, models and hardware

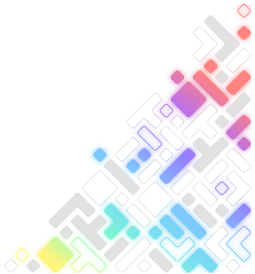
Next steps

- Upload some missing scripts and how to run all the tests to https://github.com/ivanbaldo/llama-inference/ in the *ibaldoall* branch
- Use instructed models always and not mix between the base and chat ones.
- Maybe run the other tests in the 4xA100-40G ($$$ though..)
- + also test Mixtral 8x7b (more $$$...)
- Add more LLM inference engines tests (time)
- Obtaining a client that would be willing to allow us to run tests outside office hours in his HW would enable us to provide quarterly updated speed benchmark results: it seems that would be good for the community, personally not happy with what I found currently



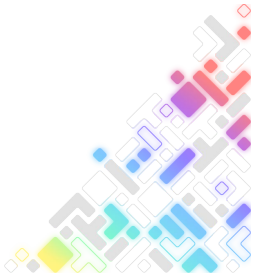
Next steps

- Consider using and contributing to https://github.com/premAI-io/benchmarks
- Or otherwise maybe start from scratch our own (last resort)

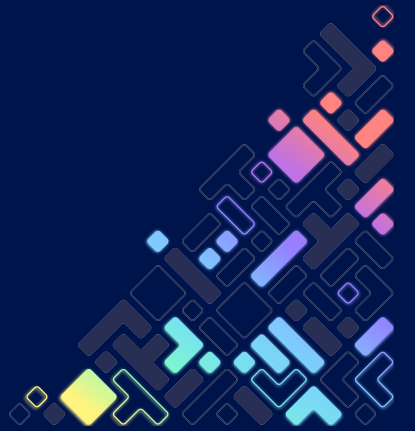


Thanks!

- Netlabs for giving the time, money, travel, etc. for me to prepare this talk and come here
- Pablo Casal (Netlabs CEO and owner) for trusting in me
- Federico Varela from Netlabs for parsing the results + HW usage information and obtaining the graphs
- Many others at Netlabs helping in different capacities
- The Open Source Summit for accepting this talk
- All of you for being patient with me, tolerating my bad english and accent, etc.!



Still awake? Questions?

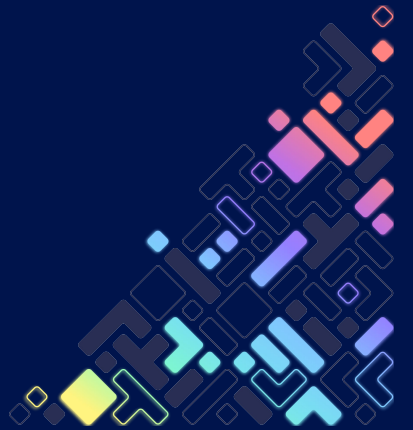


THANK YOU

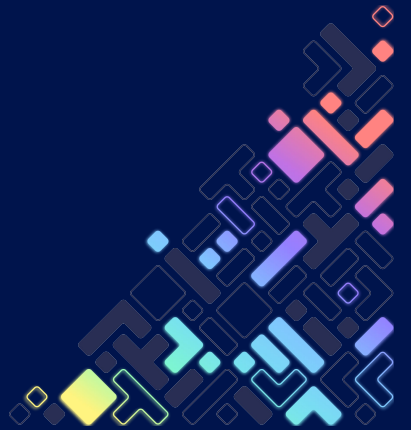
THE LINUX FOUNDATION

OPEN SOURCE SUMMIT

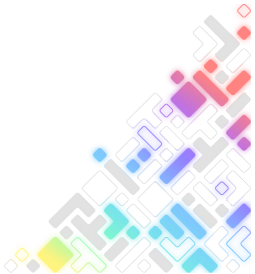
NORTH AMERICA



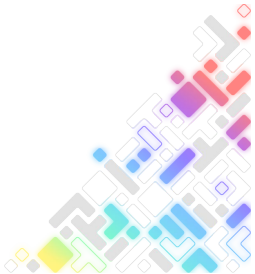

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA




THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA




THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA




THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA