Intel ＆ AMD iGPUでGPGPU最強コスパ。vulkanとipex-LLM生成AIとSimonW/LLM最強

1. 自己紹介、生成AIとは
2. Intel･AMD iGPUでGPGPU
3. SimonW/LLM簡単な使い方
4. iGPUベンチマーク
5. ipex-llmの未更新問題

発表者：Kapper

ガジェットハッキング
ユーザーグループ

This Presentation:
Slideshare & PDF files
publication of my HP
http://kapper1224.sakura.ne.jp
Gadget Hacking User Group
Speaker:Kapper

メインテーマ「ガジェット＆生成AI」

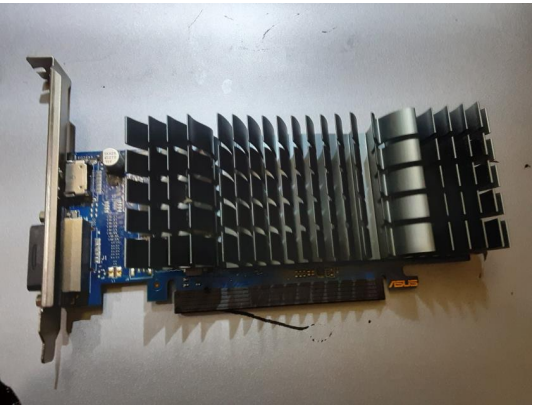
東海道らぐ2025年9月　2025年9月27日
Place:熱田生涯学習センター

自己紹介　Self Introduction

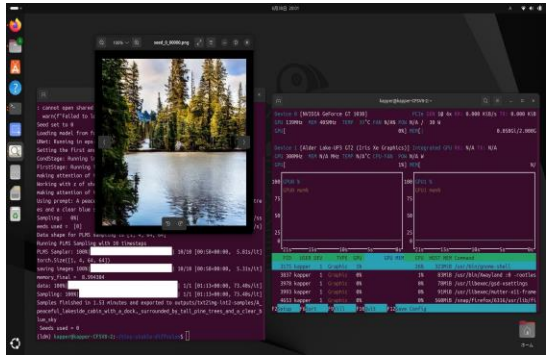
- My name: Kapper
- X(Twitter) account：@kapper1224
- HP：http://kapper1224.sakura.ne.jp
- Slideshare: http://www.slideshare.net/kapper1224
- Docsell: https://pawoo.net/@kapper1224/
- Mastodon: https://pawoo.net/@kapper1224/
- Facebook：https://www.facebook.com/kapper1224/
- My nobels：https://ncode.syosetu.com/n7491fi/
- My Posfie(Togetter)：https://posfie.com/@kapper1224
- My Youtube：http://kapper1224.sakura.ne.jp/Youtube.html
- My Hobby：Linux、*BSD、Generative AI and Mobile Devices
- My favorite words：The records are the more important than the experiment.　「記録は実験より勝る」
- Test Model：Netwalker、Nokia N900、DynabookAZ、RaspberryPi、Nexus7、Nexus5、Chromebook、GPD-WIN、GPD-Pocket、Macbook、NANOTE、SteamDeck、Windows Tablet、SailfishOS、UBPorts、postmarketOS、NetBSD and The others...
- Recent my Activity：
- Hacking Generative AI (Images and LLM) on a lot of devices.
- Hacking Linux on Windows1x Tablet (Intel Atom and Gaming tablet) and Android Smartphone.
- Hacking NetBSD and OpenBSD on UEFI and Windows Tablet.
- Hacking SimonW/LLM, RAG and Finetuning LLM.
- 後、最近小説家になろうで異世界で製造業と産業革命の小説書いていますなう。

Recently my activities

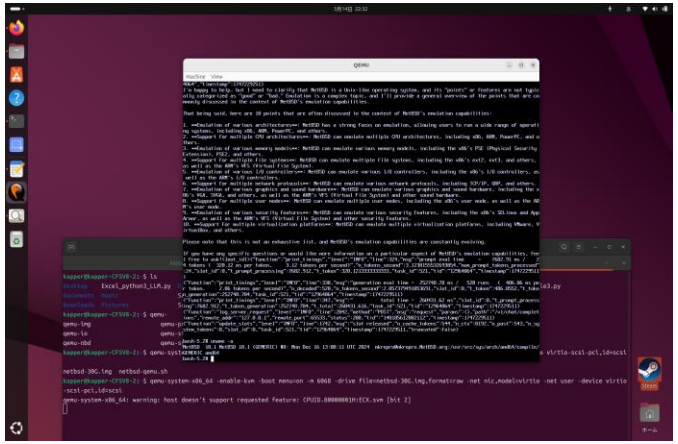
Generative AI
on Junk GPU Boards(2GB)



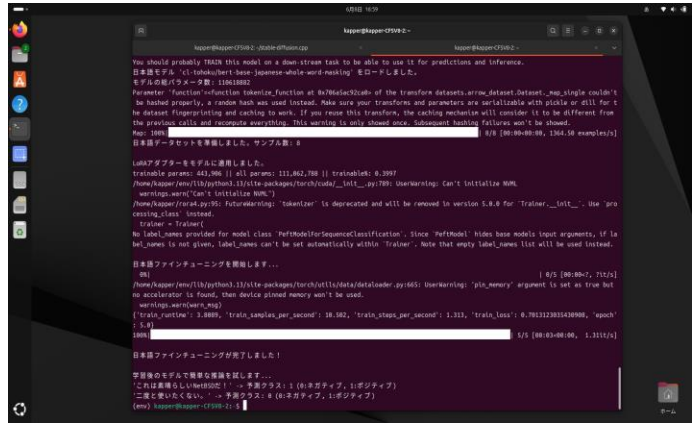
Tiny Stable-diffuson
(VRAM 1GB)



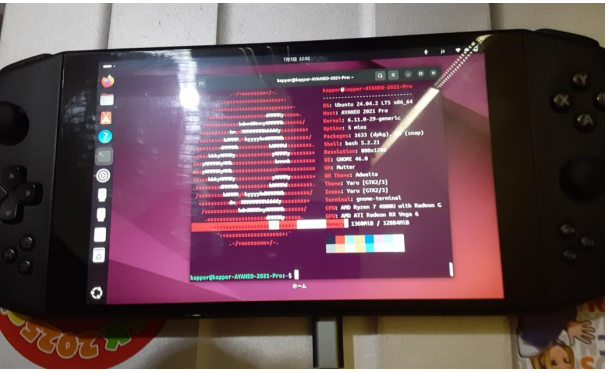
SimonW/LLM and Python
(CUDA,vulkan,CPU)



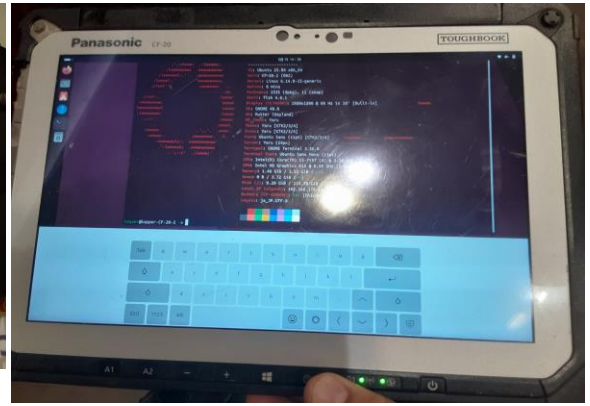
RAG,LLM Finetuning
on Python



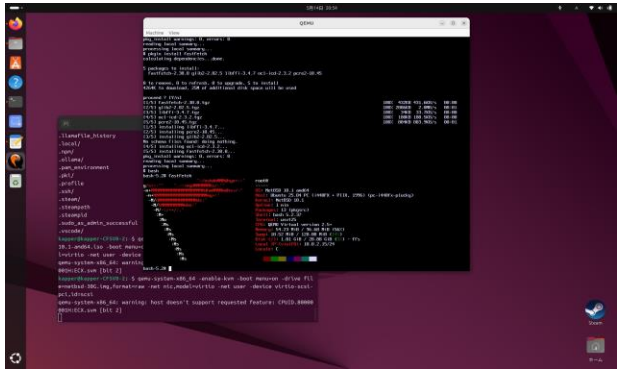
Linux on Gaming Tablet PC



Junk Windows Tablet



Generative AI on NetBSD



OSC,KOF参加

突然ですが月刊I/Oに投稿しました。2025年11月号を是非読んで頂けたら幸いです。

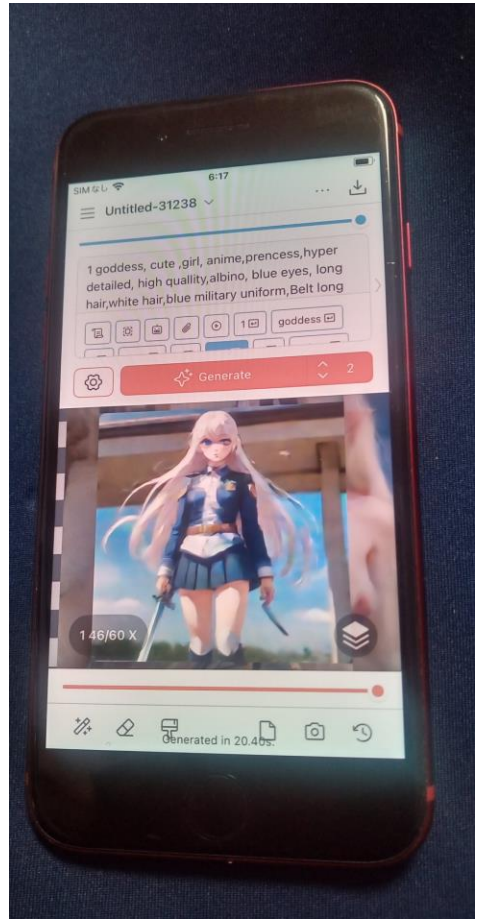
- 「古いPC・スマホの再利用術」特集でLineageOS22.2のインストール

生成AIとは?

- テキストなど 小さい情報から画像、動画、文章などを生成
- 短時間で複雑なデータを生成可能で生産性が良い
- 複雑な計算が必要なので高性能なCPU,GPU,NPUが必要
- 大規模言語モデル（Large Language models）が有名
- 最近はAndroidやiPhoneでもOK
- 他にも画像生成AIなど
  定番モデルpony diffusion、Animagine XL他
  参考としあき wiki https://wikiwiki.jp/sd_toshiaki/SDXLモデル

- AIについて初心者向けの説明をして欲しいと意見が出るかもしれませんが、
  1から説明し始めると最低1時間以上はかかりそうなので、
  初心者向けの細かい説明はOSC京都2025で発表したので割愛します。

「【初心者向け】生成AI SimonW/LLMとOllama・llamafile無料APIで
コマンドラインをAI革命するセミナー。NetBSDもOK」の資料をご参照下さい



最近Intel Meteor-Lake Core Ultraを買いました

- ゲームでAMDに負けるため不人気モデルのジャンク
  当初はAMD Z1Extremeを買う予定だったが安すぎる価格に負けました
- だってガチで安かったんだもん

詳細画像を見る ›

¥37,980 (税込)

paidy ペイディなら月々 ¥12,660 ⓘ

状態 中古 | ランク C | 付属品 一部付 | 保証 1ヶ月 | 1点限り

Linux上でのNPUの現在
まだNPUはLLMをLinuxでAI推論使えないそうです
(Windows Only) ぐぬぬ

Run IPEX-LLM on Intel NPU

This guide demonstrates:

- How to install IPEX-LLM for Intel NPU on Intel Core™ Ultra Processors
- Python and C++ APIs for running IPEX-LLM on Intel NPU

> Important
>
> IPEX-LLM currently only supports Windows on Intel NPU.

AMD's GAIA For GenAI Adds Linux Support: Using Vulkan For GPUs, No NPUs Yet

Written by Michael Larabel in AMD on 25 September 2025 at 08:26 PM EDT. 8 Comments

Back in March AMD announced the open-source GAIA software for GenAI but as noted in that former article, at launch it was limited to Windows-only support. AMD recently released a new version of GAIA with Linux support albeit in a rather interesting twist is limited to Vulkan acceleration.

AMD GAIA is described as an open-source solution from the company for running large language model (LLM) agents on Ryzen AI PCs "in minutes" by having both a GUI and command line interface and building off the likes of Lemonade and Llama.cpp.

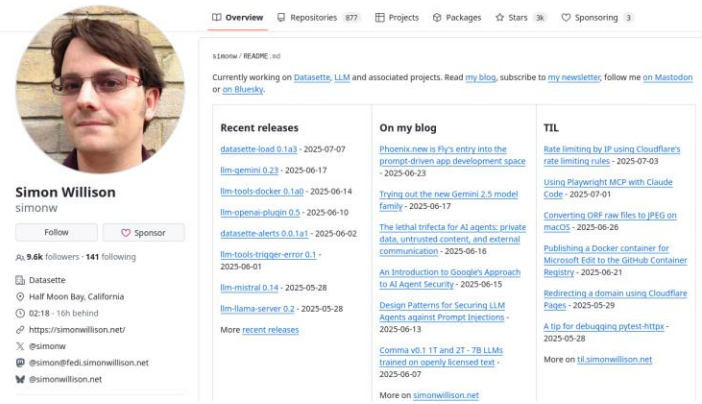
At launch it was limited to Windows-only support and thus quickly fell off my radar. But a Phoronix reader recently raised GAIA again and I was pleased to see there is now Linux support as of last month.

AMD GAIA 0.10 released back on 20 August and now comes with Linux support. But the Linux support is rather interesting in terms of hardware/driver targets:

> ***"Linux Support - Native CLI and UI (RAUX) support for Ubuntu with unified cross-platform installation (currently supports iGPU via llama.cpp/Vulkan backend using the Lemonade server)"***

While GAIA was developed in part for showcasing Ryzen AI NPU potential, the Linux support at this point doesn't mention any NPU coverage with the new AMDXDNA driver. It also doesn't mention ROCm. But rather is using Llama.cpp's Vulkan back-end with the Lemonade Server.

SimonW/LLMとはそもそもなんぞや？



- Simon Willisonさんが作ったコマンドラインで動くLLMツール。
- 星の数ほどありそうな、コマンドラインで動くLLMツールの一つ。世界中で広く使用
- 多数のプラグインとOS対応で広く使われている。無料APIに公式対応
- 挙動がシンプルで短いコマンドで使用可能なのが長所。Shell script,Awk,Pythonで拡張すると凄く応用性が高まる
- 基本的な使い方は

  ①LLMに情報をインプットして出力する方法。日本語で指示や翻訳など

  ②コマンドの出力をLLMにパイプで送り解析、日本語化など

  ③バッチファルやAWKなどに組み込んで自動化

  ④PythonやJavascriptに組み込んで多数の機能をAI化

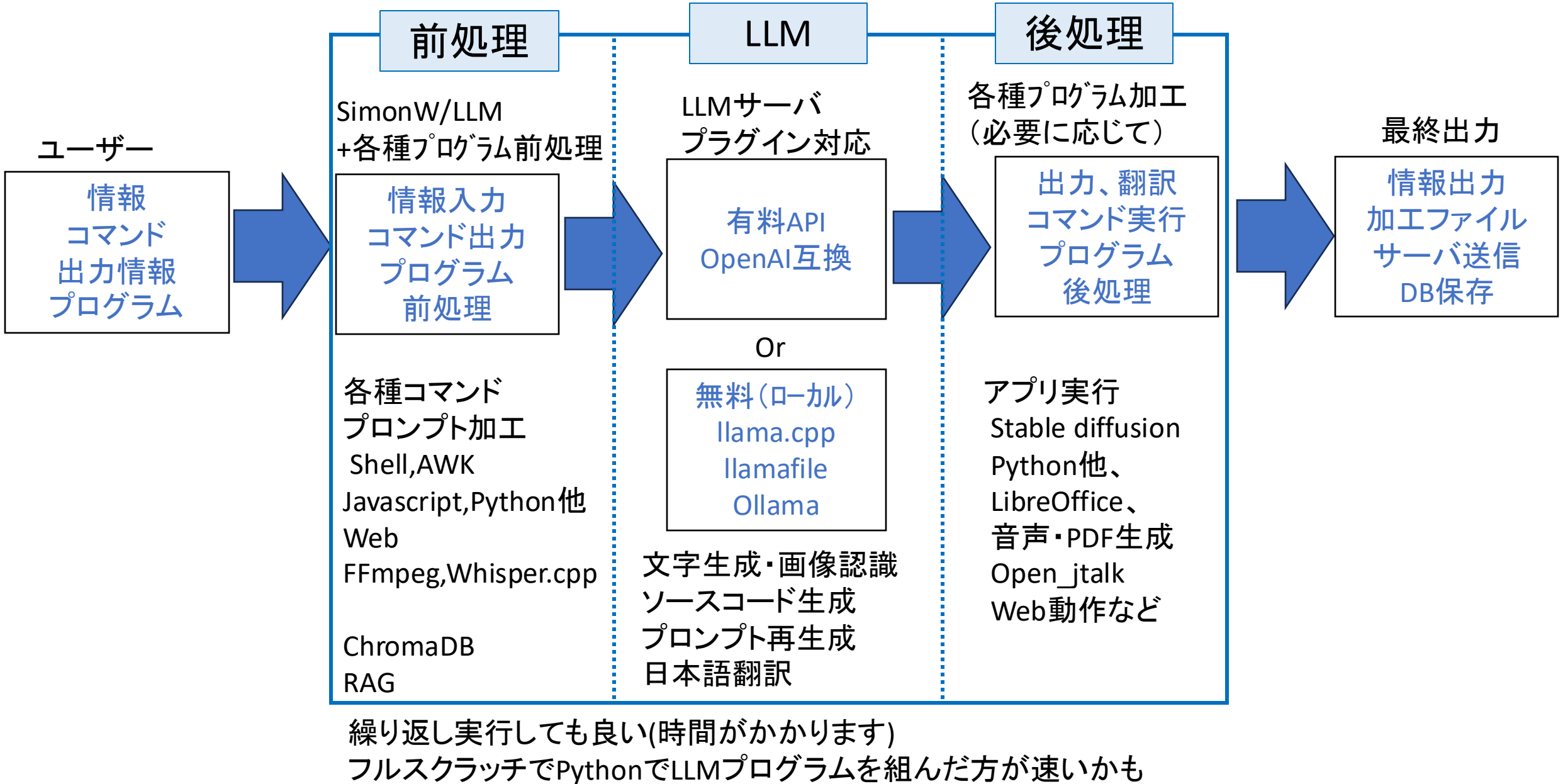
  ⑤Whisper.cppや音声合成AIと連携してAIボイス出力

・Pythonで書かれているが依存関係が少なくNetBSDでも動作確認済

・Python APIとしてimport llmと直接プログラムで叩いて使用も可能

・AIオートサジェスチョン機能としてはFish Shell-AIを使った方が良いかも

SimonW/LLMのイメージ図



簡単なSimonW/LLM使い方例

- llm "NetBSDの説明"　●通常のLLM(OpenAI API)
- llm "Ubuntuの長所" -m llama3.2:latest　●ローカルLLMモデル使用時
- llm –c "会話しましょう" -m llama3.2:latest　●チャットモード
- llm -m llama3.2:latest "Linuxでストレージを表示. コマンドのみ出力."　●モデル選択
- cat sample.py | llm –s "日本語でソースコードを解析"　●出力をパイプで処理＋システムモード
- echo "LinuxでSSHポート（22番）を開けるファイアウォール設定のコマンドとその説明を日本語で教えてください" | llm –m llama3.2:latest　●Linuxコマンド解説
- man curl | llm -s "以下を要約して、日本語で使用方法と代表的なオプションを簡潔に説明してください。" -m llama3.2:latest　●Man を日本語訳と要約
- cat script.sh | llm -s "このシェルスクリプトの問題点を修正して、修正後の完全なコードだけを出力してください" -m llama3.2:latest > fixed_script.sh　●＞で結果をファイルに出力

人気あるローカルLLMモデルデータ

- 主にHuggingfaceなどからGGUFファイルをダウンロードしてmodels以下の各フォルダに保存して使用

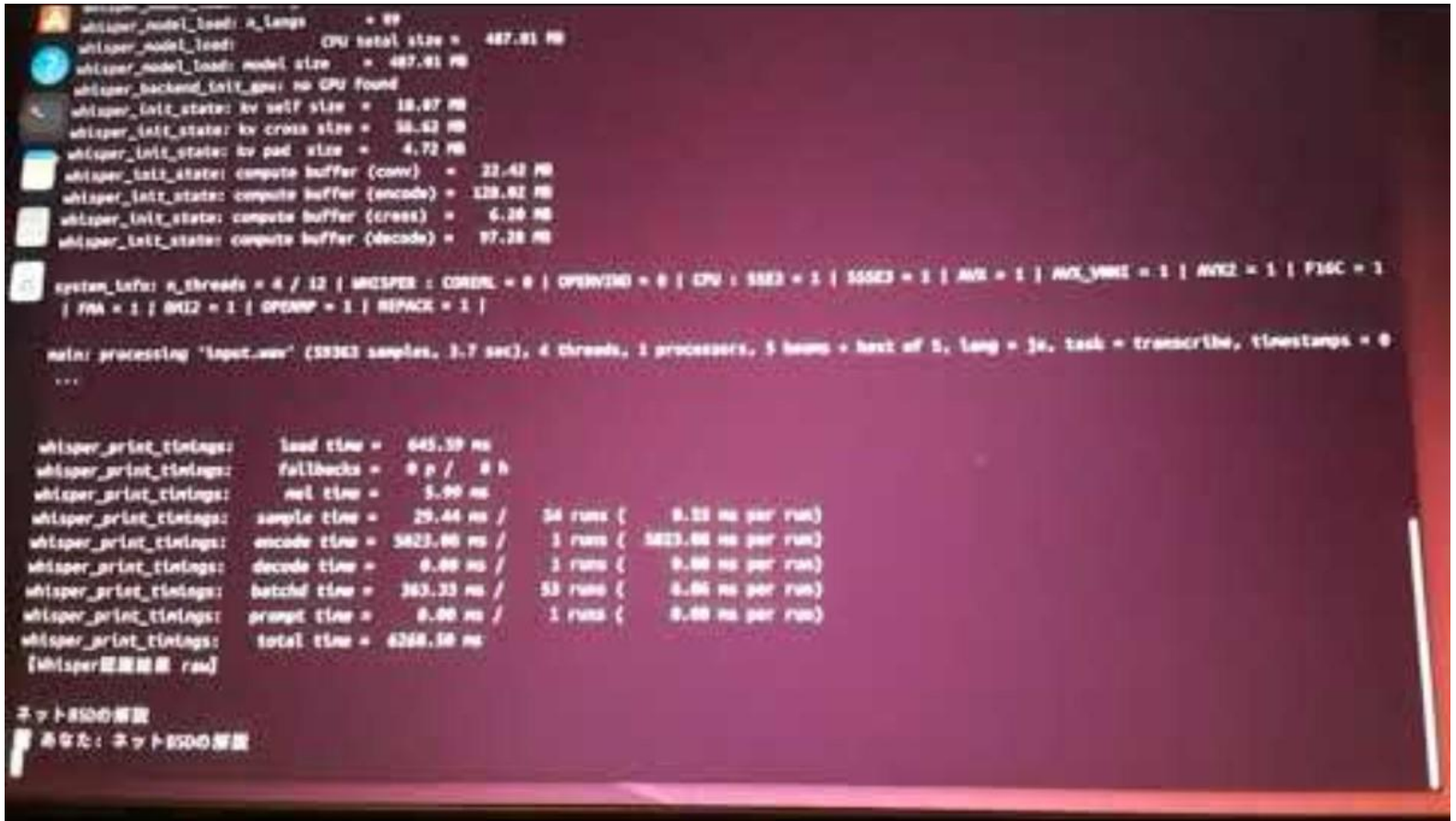
- 日本語対応：llama3.2 1B,3B(古いけどサイズが小さく1.5GB相当から)
- 高性能：Gemma3n n2b(5.6GB)、n4b(7.5GB)、Gemma3 4B(3.3GB)
- 中国系：Deepseek-r1(1.1~404GB)、Qwen3(1.2~142GB)　<think>で高性能化
- 最軽量：Qwen3 1.7B Q4_K_M(1.28GB) 、Gemma3 1B Q4_1(764MB)、Gemma3 270M Q4 (253MB)
  日本語可モデル。低スペックPC、低VRAMに特にオススメ

- マルチモーダル：Qwen2.5vl(3.2~49GB)、llava (4.7~20GB)
- コーダー系：Qwen2.5-coder(1.0~20GB)
- ベクトル埋め込みモデル：mxbai-embed-large

- ~0.2GB 殆ど会話出来ない。0.25~1.0GB 辛うじてギリギリ日本語で会話が成立、英語混じり
  1.0～2.0GB わずかに日本語可、英語混じり。2.0～3.0GB 日本語で多少会話出来る、
  3.0~4.0GB この位あると普通、5.0～6.0GB 犬猫レベルの簡単な擬人化、
  7.0～8.0GB 超最低限の人格の語尾を再現

さらに応用。生成AI会話アプリ作成

- aplayで音声をWav保存
  whisper.cppでwavをテキストに変換
  SimonW/LLMとllama.cppでテキストを生成AIで変換
  Open_jtalkで変換したテキストを読み上げ



ハードウェア目安

- 最低（動くだけ。速度は一切考慮しない）
  x86:SSE3搭載CPU、メモリ+Swap：1GB以上（モデル量子化の例外あり）
  Android メモリ＋Swap 1GB以上(モデル量子化の例外あり)
  RapsberryPi4,5 メモリ＋Swap 1GB以上
- 必要
  x86:第8世代Intel、AMD Ryzen以降、メモリ+Swap 3GB以上
  Android メモリ+Swap 3GB以上
  GPU:VRAMメモリ2GB以上 (Intel内蔵GPU iris Xe、UHD以上)
- 推奨（GPU前提）
  x86:第8世代Intel、AMD Ryzen以降、メモリ8GB以上
  GPU：NVIDIA GTX1060、AMD RX580以降 VRAMメモリ6GB以上(gpt-oss 20Bなどで12GB以上が望ましい)

- できれば10 tokens / sec 以上の速度を狙う(推奨15～25 tokens / sec)
- 速度を上げるには小さいモデル＋高速GPU＋オプション(-B 512 –C 2024 –t (コア数))など使用
- 低スペックPC、低VRAMならGemma3 1B Q4_1(764MB)かQwen3 1.7B Q4_K_M(1.28GB)のモデルを推奨
- 賢い擬人化させる場合には出来るだけGPU VRAMを増やしてデータサイズの大きいモデルを使用
- 邪道であるが、CPUメインメモリを増やして内蔵GPUかNPUで処理する方法もあり

GPGPU

GPU

- NVIDIA GPU
- AMD GPU
- Intel GPU

- AMD内蔵APU
- Intel内蔵GPU
  (Iris Xe,UHD)
- ARM内蔵GPU

OS

- Linux
- Windows
- Mac
- *BSD
- Android他

Driver

- メーカー製
  Driver
- OSS Driver
  (MESA他)
  →DRM

Framework

- NVIDIA CUDA
  Vulkan
  OpenCL(CLBLAS)
- AMD ROCm
  Vulkan,ZLUDA
  OpenCL(CLBLAS)
- Intel oneAPI
  OpenVINO
  SYCL、Vulkan
  OpenCL(CLBLAS)
- CPU
  OpenBLAS
  OpenVINO

APPLICATIONS

- Ollama
  llama.cpp
  LM Studio他

- Stable-Diffusion
  Stable diffusion.CPP
  SD.Next
  FastSD CPU他

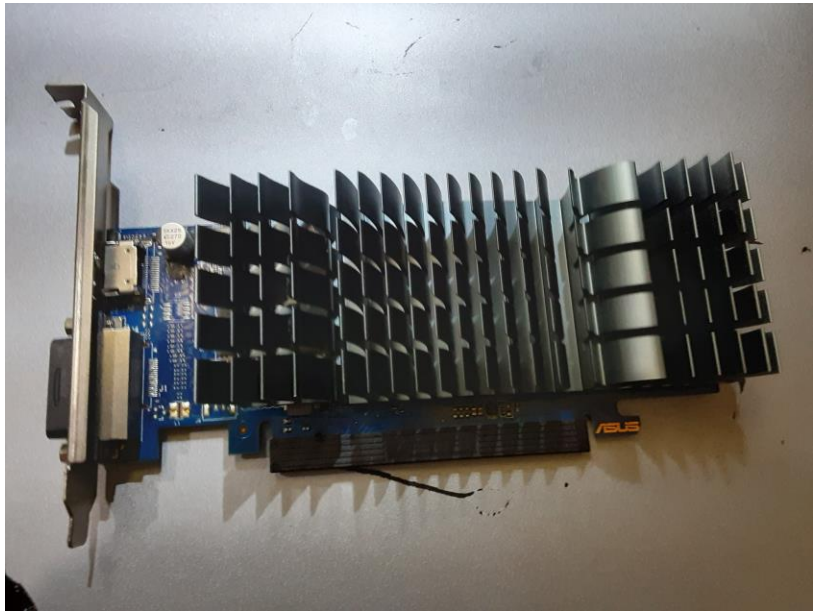
- Yolo
- Gimp
- Tensor-flow

簡単なGPGPUフレームワーク一覧表

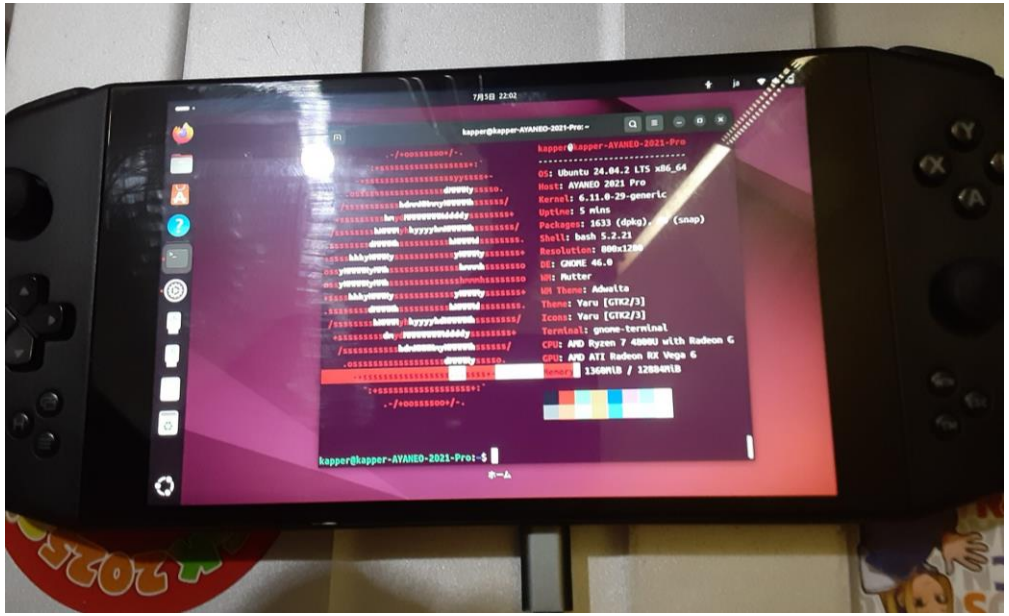
	CUDA	ROCm	vulkan	oneAPI +ipex-llm	oneAPI (SYCL)	CLBLAS (Open CL)	Metal	Open BLAS (CPU)	コメント
NVIDIA	とても速い	ー	やや遅い	ー	ー	とても遅い	ー	CPU	CUDA万能
AMD	ー	とても速い	かなり速い	ー	ー	とても遅い	ー	CPU	vulkan強い
Intel	ー	ー	とても遅い	とても速い	遅い	とても遅い？	ー	CPU	ipex-llm不安定
ARM (Adreno GPU)	ー	ー	？	ー	ー	？	ー	CPU	？
ARM (Mac)	ー	ー	？	ー	ー	？	とても速い	CPU	非常に高速

実は爆安ジャンクGPUボードLLMに使えます

- 日本語対応のLLAMA3.2 3Bの容量が1.5GBなのでVRAM2GBで動きます。
- AMDのAPU内蔵GPUでもOK。ゲーミングタブレットPCなど。
- 情報収集には物足りなくてもSimonW/LLMなら十分な性能
- できればGTX1050Tiクラスだとなお良い



GTX1030 VRAM 2GB



Ayaneo 2021 Pro VRAM 3GB

GPUベンチマーク

- llama.cpp公式GithubにVulkanベンチマーク結果。
  おおよそ他のベンチマークと類似(CUDA,ROCm,SYCLはvulkanより高速)
  https://github.com/ggml-org/llama.cpp/discussions/10879
- 代表例(llama 2 7B)
  NVIDIA RTX4090:170.95 tokens/sec
  NVIDIA RTX3090:136.81 tokens/sec
  AMD RX 7900XT:123.18 tokens/sec
  Apple M3 Ultra:115.54 tokens/sec
  AMD RX 9060XT:70.13 tokens/sec
  NVIDIA RX3060:64.76 tokens/sec
  Intel Arc A770:49.43 tokens/sec
  AMD Ryzen AI Max+395:46.75 tokens/sec
  NVIDIA GTX1070Ti:42.86 tokens/sec
  Intel Core Ultra 7 258V:21.86 tokens/sec
  AMD Ryzen AI 9 HX 370:21.23 tokens/sec
  AMD Ryzen Z1 Extreme:18.77 tokens/sec
  Intel Core i7 1100:7.28 tokens/sec
  Intel Core i5 8000:3.23 tokens/sec

Vulkan Scoreboard for Llama 2 7B, Q4_0 (no FA)

Chip	pp512 t/s	tg128 t/s	Commit	Comments
Nvidia RTX 4090	8534.56 ± 200.32	170.95 ± 0.32	bb4f7a9	coopmat2
AMD Radeon RX 7900 XTX	3489.67 ± 82.17	145.00 ± 0.89	d1aa0cc	
Nvidia RTX 3090	4543.96 ± 73.80	136.81 ± 3.63	bb4f7a9	coopmat2
Nvidia RTX 5070 Ti	6213.63 ± 27.72	135.63 ± 0.18	d13d0f6	coopmat2
AMD Radeon RX 9070 XT	3831.64 ± 1.82	130.57 ± 0.03	fd1234c	
AMD Radeon RX 7900 XT	2941.58 ± 17.17	123.18 ± 0.40	71e74a3	
Nvidia A100 (80GB)	3103.32 ± 4.21	121.83 ± 0.54	d394a9a	
Apple M3 Ultra Mac Studio	1116.83 ± 0.55	115.54 ± 0.78	2d451c8	MoltenVK
AMD Radeon RX 6900 XT	1257.98 ± 1.55	101.42 ± 0.02	44e18ef	
AMD Radeon RX 7800 XT	2145.60 + 23.14	96.89 + 0.22	baad948	
AMD Radeon RX 6800 XT	1533.60 ± 2.47	95.56 ± 0.72	N/A	
Nvidia RTX 4070	3179.37 ± 46.16	92.29 ± 0.28	9a48399	
AMD Radeon PRO W6800X	510.80 ± 0.13	86.47 ± 0.46	13b4548	MoltenVK

NVIDIAGTX1060 6GBでもまずまずの速度。十分実用

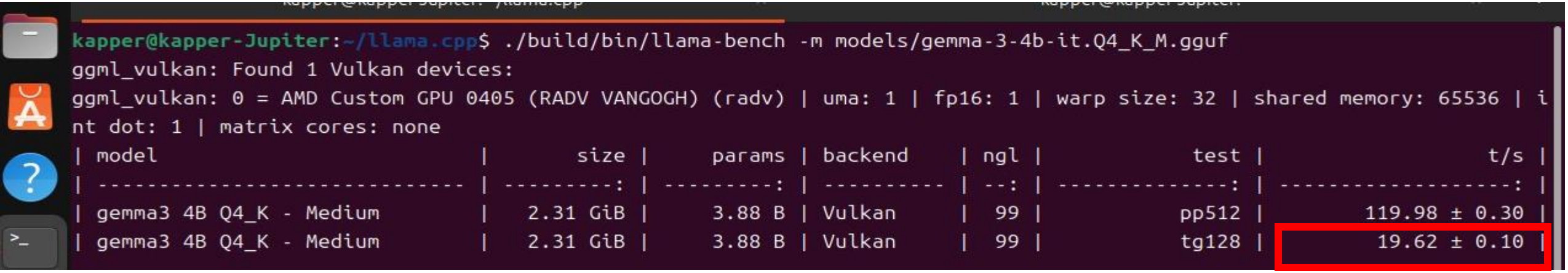
```
kapper@kapper-CFSV8-2:~/llama.cpp$ ./build/bin/llama-bench -m ../Llama-3.2-3B-Instruct-Q4_K_L.gguf -ngl 999 -b 512
ggml_cuda_init: GGML_CUDA_FORCE_MMQ:    no
ggml_cuda_init: GGML_CUDA_FORCE_CUBLAS: no
ggml_cuda_init: found 1 CUDA devices:
  Device 0: NVIDIA GeForce GTX 1060 6GB, compute capability 6.1, VMM: yes
| model                          |       size |     params | backend    | ngl | n_batch |            test |                  t/s |
| ------------------------------ | ---------: | ---------: | ---------- | --: | ------: | --------------: | -------------------: |
| llama 3B Q4_K - Medium         |   1.96 GiB |     3.21 B | CUDA       | 999 |     512 |           pp512 |        823.37 ± 1.94 |
| llama 3B Q4_K - Medium         |   1.96 GiB |     3.21 B | CUDA       | 999 |     512 |           tg128 |         42.12 ± 0.11 |
```

NVIDIA GTX1030 2GBでも頑張ればここまでは

```
kapper@kapper-CFSV8-2:~/llama.cpp$ ./build/bin/llama-bench -m ../Llama-3.2-3B-Instruct-Q4_K_L.gguf -ngl 20 -b 64
ggml_cuda_init: GGML_CUDA_FORCE_MMQ:    no
ggml_cuda_init: GGML_CUDA_FORCE_CUBLAS: no
ggml_cuda_init: found 1 CUDA devices:
  Device 0: NVIDIA GeForce GT 1030, compute capability 6.1, VMM: yes
| model                          |       size |     params | backend    | ngl | n_batch |            test |                  t/s |
| ------------------------------ | ---------: | ---------: | ---------- | --: | ------: | --------------: | -------------------: |
| llama 3B Q4_K - Medium         |   1.96 GiB |     3.21 B | CUDA       |  20 |      64 |           pp512 |        123.42 ± 0.84 |
| llama 3B Q4_K - Medium         |   1.96 GiB |     3.21 B | CUDA       |  20 |      64 |           tg128 |         10.26 ± 0.48 |
```

SteamDeck(AMD Zen2)でVulkan GPU

- AMD APU内蔵GPUの割に侮れない性能。GTX1060の半分程度？
  小さいLLMモデルなら快適に動く(問題はVRAM)
- SteamDeckはCPUのコア数を減らしてiGPUをフルコアにした仕様



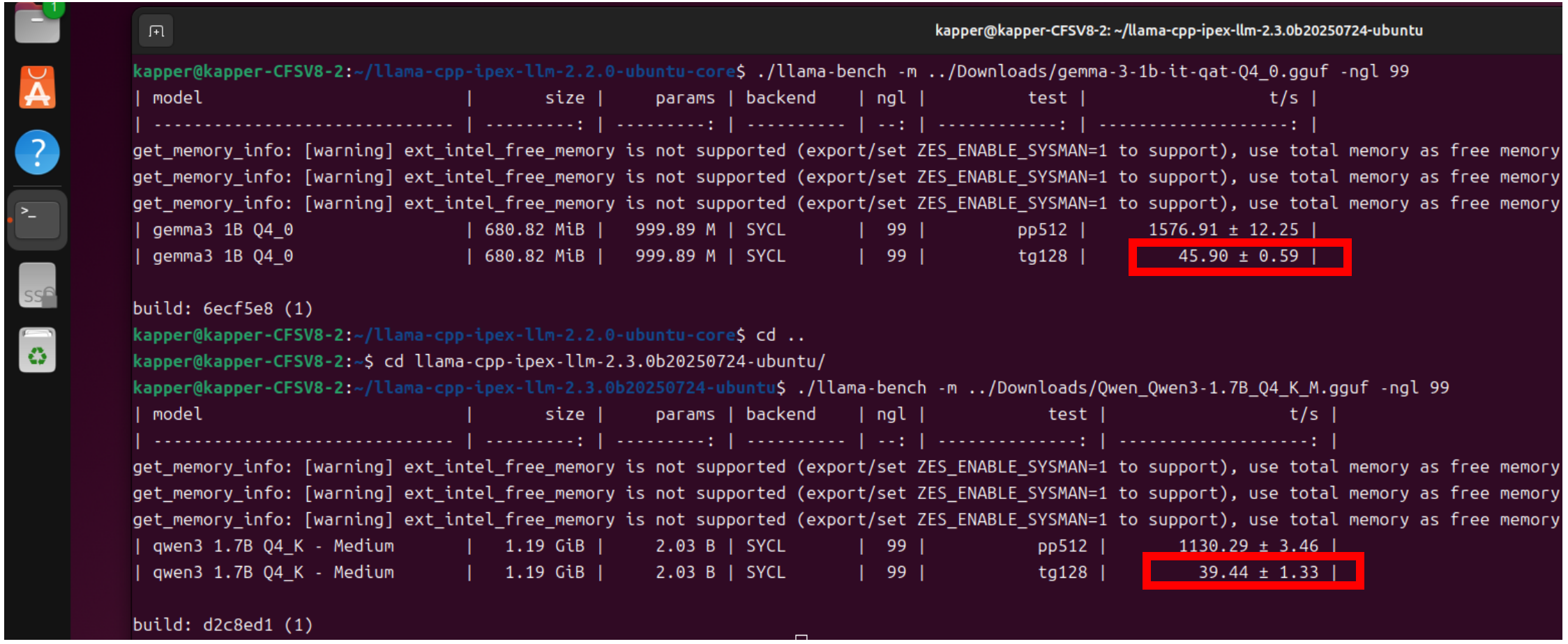
Intel oneAPI対応ハード

- 第11世代 以降CoreシリーズCPU、N100系 UHD、Iris Xe GPU
- Intel Arc GPU
- Intel XeonサーバCPU
- Intel Datacenter GPU
- NVIDIA,AMD の GPU (Codeplay の oneAPI プラグインを使用)

- 注) Core 第8～10世代のGPU(UHD)は未対応
  →とても遅いけどvulkan・CLBLASを使うしか無い
- LLMモデルサイズが同じでも計算速度に差が出るので注意

ipex-llm ベンチマーク Intel CoreUltra Arc(モバイル)

- Core Ultra Intel ArcでもoneAPI+SYCLでGPU動作しました
- CPUの2倍程度の速度で動作。小さいモデルならそこそこ動く。
  メインメモリとVRAM共有なので大きなモデルも動きます



Gemma3-270mの場合

- -ngl 0のCPUでもこの速度。古いCPUやARMでも良いかもしれない

```
(base) kapper@kapper-CFSV8-2:~/llama-cpp-ipex-llm-2.3.0b20250724-ubuntu$ ./llama-bench -m ../Downloads/gemma-3-270m-it-Q4_1.gguf -ngl 0
| model                          |       size |     params | backend    | ngl |            test |                  t/s |
| ------------------------------ | ---------: | ---------: | ---------- | --: | --------------: | -------------------: |
get_memory_info: [warning] ext_intel_free_memory is not supported (export/set ZES_ENABLE_SYSMAN=1 to support), use total memory as free m
get_memory_info: [warning] ext_intel_free_memory is not supported (export/set ZES_ENABLE_SYSMAN=1 to support), use total memory as free m
| gemma3 ?B Q4_1                 | 229.98 MiB |   268.10 M | SYCL       |   0 |           pp512 |        386.55 ± 1.53 |
| gemma3 ?B Q4_1                 | 229.98 MiB |   268.10 M | SYCL       |   0 |           tg128 |        101.70 ± 1.70 |

build: d2c8ed1 (1)
```

各機種比較用llama3.2ベンチマーク

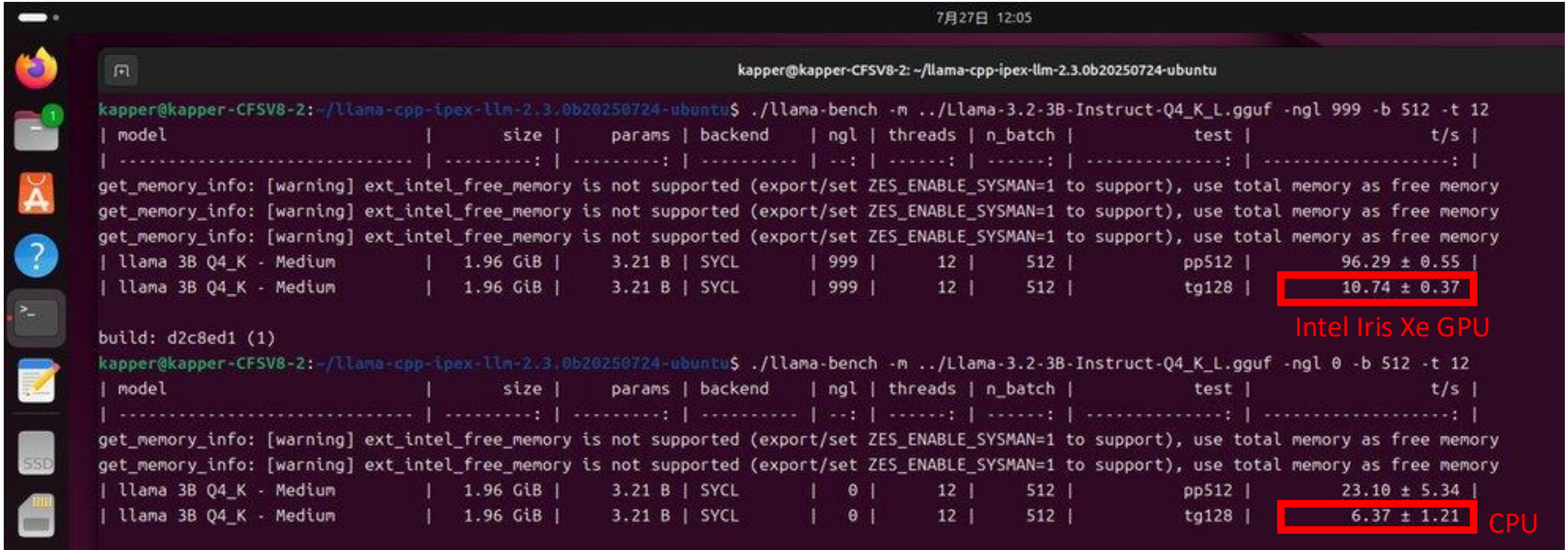
- Intel Arc(Mobile)の速度はCore 1245U iris Xeの約2倍弱
- Intel Arc(Mobile)の速度はCore 8365U UHD620の約6倍
- Intel Arc(Mobile)の速度はNVIDIA GTX1060の半分弱(GTX1050相当?)
  ipex-llmがまだCUDAに負けているせいかもしれない
- 公証では2023年10月同世代のRadeon780mをわずかに上回る速度
  まだまだ速度は出せるはず？

```
(base) kapper@kapper-CFSV8-2:~/llama-cpp-ipex-llm-2.3.0b20250724-ubuntu$ ./llama-bench -m ../Downloads/Llama-3.2-3B-Instruct-Q4_K_M.gguf -ngl 99
| model                          |       size |     params | backend    | ngl |            test |                  t/s |
| ------------------------------ | ---------: | ---------: | ---------- | --: | --------------: | -------------------: |
get_memory_info: [warning] ext_intel_free_memory is not supported (export/set ZES_ENABLE_SYSMAN=1 to support), use total memory as free memory
get_memory_info: [warning] ext_intel_free_memory is not supported (export/set ZES_ENABLE_SYSMAN=1 to support), use total memory as free memory
get_memory_info: [warning] ext_intel_free_memory is not supported (export/set ZES_ENABLE_SYSMAN=1 to support), use total memory as free memory
| llama 3B Q4_K - Medium         |   1.87 GiB |     3.21 B | SYCL       |  99 |           pp512 |        332.81 ± 6.83 |
| llama 3B Q4_K - Medium         |   1.87 GiB |     3.21 B | SYCL       |  99 |           tg128 |         18.24 ± 1.35 |

build: d2c8ed1 (1)
```

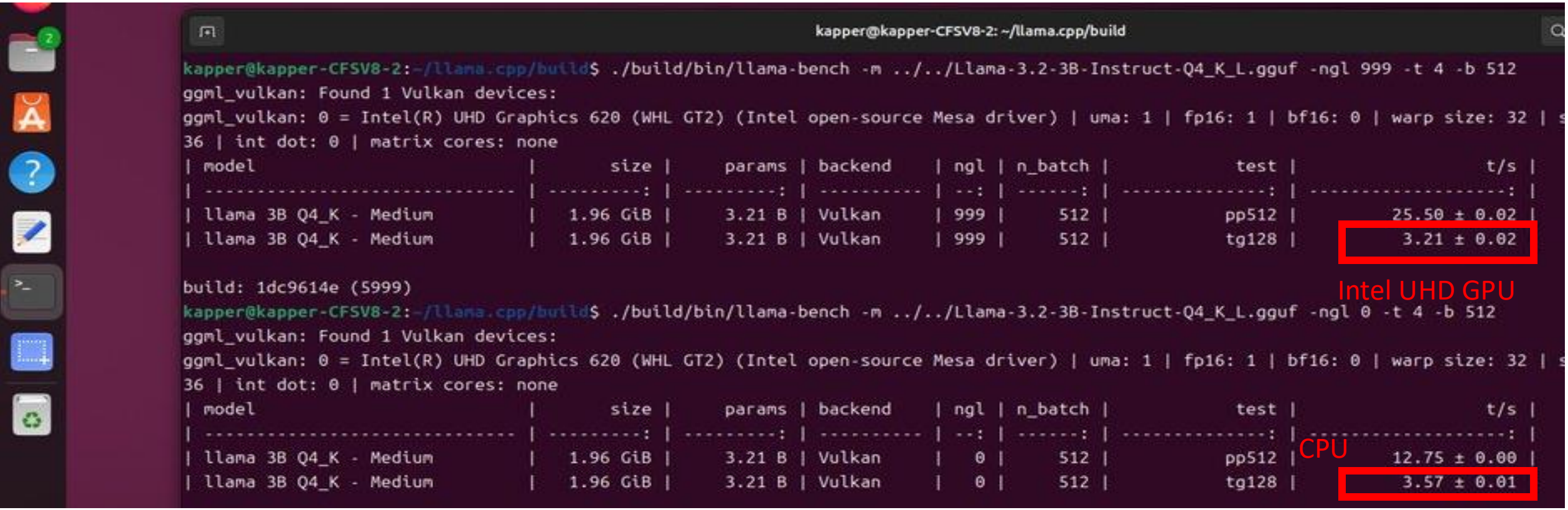
ipex-llm ベンチマーク Intel 12世代 1245U Iris Xe

- Core第12世代 Iris XeでもoneAPI+SYCLでGPU動作しました
- CPU12コアの2倍程度の速度で動作。小さいモデルならそこそこ動く。メインメモリとVRAM共有なので大きなモデルも動きます



Vulkan ベンチマーク Intel Core第8世代 UHD

- Core第8世代 UHDでもvulkanでGPU動作しましたが遅い
- CPUプロセスを少ししか食わないのでブラウザを使いながらllama.cppを使う程度には良いですが・・・



Vulkan ベンチマーク Intel Core第8世代 UHD

- Gemma3 1B Q4_1なら8.10 tokens/sec
  日本語怪しいですが Gemma3 270M Q4で17.75 tokens/sec
- これだけモデルサイズが小さければRaspberryPiでも動きそうです。

```
kapper@kapper-CFSV8-2:~/llama.cpp$ ./build/bin/llama-bench -m ../gemma-3-1b-it-Q4_1.gguf -ngl 99 -b 512 -mmp 1
ggml_vulkan: Found 1 Vulkan devices:
ggml_vulkan: 0 = Intel(R) UHD Graphics 620 (WHL GT2) (Intel open-source Mesa driver) | uma: 1 | fp16: 1 | bf16: 0 | warp size: 32 | shared memory: 65536 |
int dot: 0 | matrix cores: none
| model                          |       size |     params | backend    | ngl | n_batch |           test |                  t/s |
| ------------------------------ | ---------: | ---------: | ---------- | --: | ------: | -------------: | -------------------: |
| gemma3 1B Q4_1                 | 722.41 MiB |   999.89 M | Vulkan     |  99 |     512 |          pp512 |         60.87 ± 0.76 |
| gemma3 1B Q4_1                 | 722.41 MiB |   999.89 M | Vulkan     |  99 |     512 |          tg128 |          8.10 ± 0.01 |

build: 228f724d (6129)
kapper@kapper-CFSV8-2:~/llama.cpp$ ./build/bin/llama-bench -m ../gemma-3-270m-it-Q4_K_M.gguf -ngl 99 -b 512 -mmp 1
ggml_vulkan: Found 1 Vulkan devices:
ggml_vulkan: 0 = Intel(R) UHD Graphics 620 (WHL GT2) (Intel open-source Mesa driver) | uma: 1 | fp16: 1 | bf16: 0 | warp size: 32 | shared memory: 65536 |
int dot: 0 | matrix cores: none
| model                          |       size |     params | backend    | ngl | n_batch |           test |                  t/s |
| ------------------------------ | ---------: | ---------: | ---------- | --: | ------: | -------------: | -------------------: |
| gemma3 ?B Q4_K - Medium        | 235.16 MiB |   268.10 M | Vulkan     |  99 |     512 |          pp512 |        313.54 ± 0.30 |
| gemma3 ?B Q4_K - Medium        | 235.16 MiB |   268.10 M | Vulkan     |  99 |     512 |          tg128 |         17.75 ± 0.08 |

build: 228f724d (6129)
```

Intel UHD GPU

Intel UHD GPU

最軽量日本語対応モデル Qwen3 1.7B Q4 K_M

- 僅か1.28GBなのにギリギリ日本語が使える。とても軽量・高速
CPUやIntel UHDなど限られたリソースに最適。ipex_LLMで高速
おおよそllama3.2 3Bの1.5倍高速モデル。オススメ

https://huggingface.co/Alcoft/Qwen_Qwen3-1.7B-GGUF/tree/main

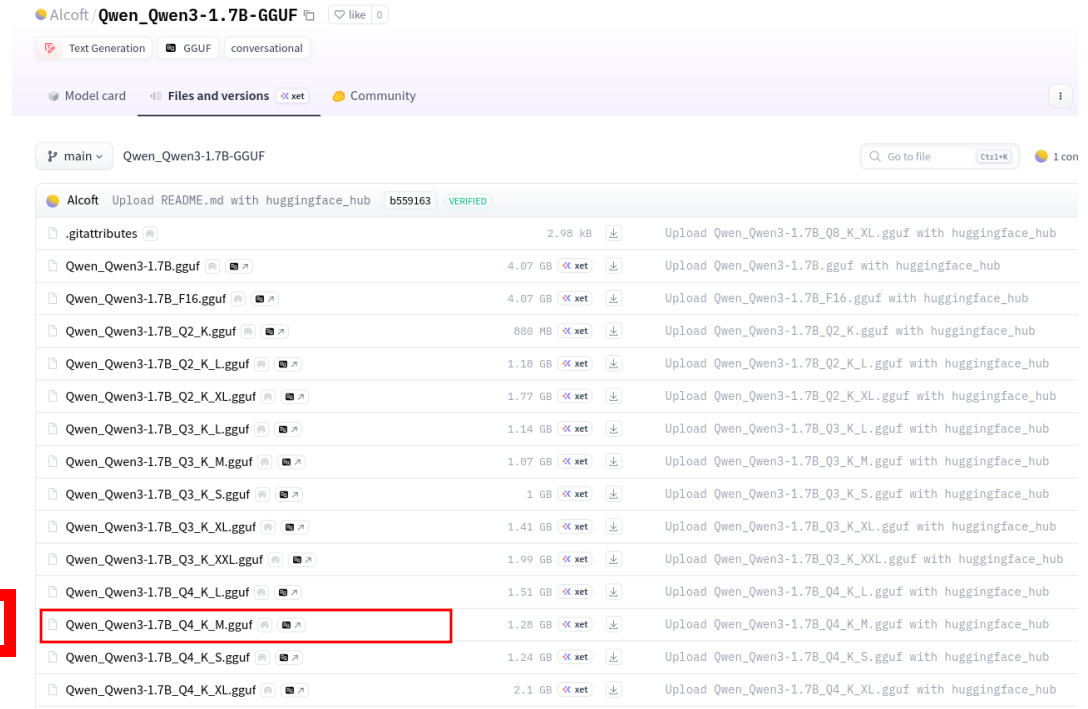
N100内蔵UHD+ipex-LLM(oneAPI+SYCL)： 7.68 tokens/sec

```
llama_perf_sampler_print:    sampling time =       8.97 ms /    49 runs   (    0.18 ms per token,  5459.61 tokens per second)
llama_perf_context_print:        load time =    8542.45 ms
llama_perf_context_print: prompt eval time =     716.42 ms /    16 tokens (   44.78 ms per token,    22.33 tokens per second)
llama_perf_context_print:        eval time =    4168.22 ms /    32 runs   (  130.26 ms per token,     7.68 tokens per second)
llama_perf_context_print:       total time =    4973.81 ms /    48 tokens
kapper@kapper-CFSV8-2:~/llama-cpp-ipex-llm-2.3.0b20250724-ubuntu$ echo "今日の天気。 no_think" | ./llama-cli -m ../Qwen_Qwen3-
```

Core 第8世代 内蔵UHD+Vulkan：5.66 tokens/sec

```
kapper@kapper-CFSV8-2:~/llama.cpp$ ./build/build/bin/llama-bench -m ../Qwen_Qwen3-1.7B_Q4_K_M.gguf -ngl 999
ggml_vulkan: Found 1 Vulkan devices:
ggml_vulkan: 0 = Intel(R) UHD Graphics 620 (WHL GT2) (Intel open-source Mesa driver) | uma: 1 | fp16: 1 | bf16: 0 | warp s
 int dot: 0 | matrix cores: none
| model                          |       size |     params | backend    | ngl |           test |                  t/s |
| ------------------------------ | ---------: | ---------: | ---------- | --: | -------------: | -------------------: |
| qwen3 1.7B Q4_K - Medium       |   1.19 GiB |     2.03 B | Vulkan     | 999 |          pp512 |                      |
| qwen3 1.7B Q4_K - Medium       |   1.19 GiB |     2.03 B | Vulkan     | 999 |          tg128 |          5.66 ± 0.17 |

build: 1dc9614e (5999)
```



最新版ipex-llmをcondaで使う

- llama.cppの場合はIntel GPUドライバとoneAPIを入れてから、
  conda create -n llm-cpp python=3.11
  conda activate llm-cpp
  pip install --pre --upgrade ipex-llm[cpp]
  mkdir llama-cpp
  cd llama-cpp
  init-llama-cpp
- Ollamaの場合は同様にIntel GPUドライバとoneAPIを入れてから、
  conda activate llm-cpp
  init-ollama
  export OLLAMA_NUM_GPU=999
  export no_proxy=localhost,127.0.0.1
  export ZES_ENABLE_SYSMAN=1

  source /opt/intel/oneapi/setvars.sh
  export SYCL_PI_LEVEL_ZERO_USE_IMMEDIATE_COMMANDLISTS=1
  export ONEAPI_DEVICE_SELECTOR=level_zero:0
  ./ollama serve &
  ./ollama run llama3.2:latest

ipex-llmは2025年5月以降ほぼ未更新 問題

- Intel GPU高速化の核となるipex-llmがほぼ未更新(特にPortableZIP版)
- 動かないモデル、遅いモデルが増えてきた
- 今後はoneAPI(SYCL)かvulkanを使うしか無いがとても遅い
- NPUはWindowsしかサポートされていない
- 結論として、NVIDIAやAMDのiGPUが相対的に使いやすくなる
  →オープンソースですが、ソース内部を読み解かないといけない

よくある質問・疑問点

- CPUではだめなんですか？ --- 遅いだけです。その分CPUメモリは安い。
- 古いPCあるんですが・・・---ジャンクGPUかノートパソコンならeGPUで高速化。
- AndroidスマホのGPUは動きますか？--- VRAMが機種により差があるがVulkan
  Termuxとllama.cppで標準でvulkanが動くそうですが問題もあり。
  Subtle Vulkan shader compilation bug when running on Adreno GPUs (Samsung Galaxy S23 Ultra) #5186
  https://github.com/ggml-org/llama.cpp/issues/5186
- NVIDIA以外のGPUは動くのですか？ --- ROCm,DirectML,Metal,oneAPI+SYCLでも動く
- なぜIntel Arcは生成AIであまり使われないの？ --- 動作は問題なし。ipex-llm推奨、ただし更新遅れ
- Intel NPUでも動きますか？--- WindowsならOpenVINOかoneAPI+SYCL推奨(ipex-llm)
- ノートパソコンの内蔵GPUは使えますか？ ---遅いけどIntel内蔵GPUは動く。Intel Iris XeやUHD動作確認

  AMDのデスクトップやゲーミングPCは内蔵VRAMをBIOSで変更できるかも。
- Ollamaのモデルを追加したけどSimonW/LLMで動かない？---llm install llm-ollamaをもう一度実行
- Ollamaは何故*BSDに対応していないの？---vulkanを強制指定してエラーが出るバグあり。どうも開発者が*BSDのサポートに興味が無いらしく修正されていません。FreeBSDコミュニティが過去に大騒ぎしていました。
  https://github.com/ollama/ollama/issues/1102

- そもそもPythonフルスクラッチの方が良くない？--- もちろんYES。SimonW/LLMは手軽さと汎用性重視

まとめ

- 簡単なSimonW/LLMの使い方と説明
- 各種GPUをLinuxで動作確認。ノートパソコン内蔵GPUもOK
- SimonW/LLM程度なら小さいLLMモデルでも十分使える
- 現時点最軽量日本語対応モデル
  Gemma3 1B Q4_1かQwen3 1.7B Q4_K_M を低スペックPCでも推奨