CS6886W - Systems Engineering for Deep Learning

Assignment - 2 Performance Analysis of GPT-2

Name : **Manish Kumar**
Roll No : **CS24M525**

Github repository link : https://github.com/cs24m525/cs6886w_asg2

This assignment has been done on :
System Details : Dell Inspiron 15 5510 , Windows 11 Home OS
CPU details : Intel(R) microarchitecture code named Tigerlake
11th Gen Intel(R) Core(TM) i5-11320H @ 3.20GHz
Base speed: 2.50 GHz (Base Speed for sustained all-core load TDP of 28 W)

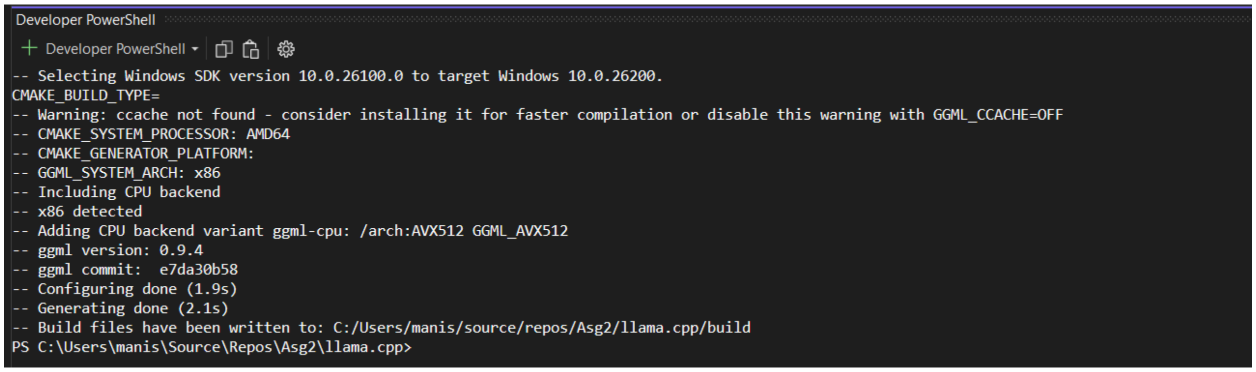
Task 1: Install llama.cpp from Source

Step 1: Prerequisites Installation
Git,CMake,C/C++ Compiler,MSYS2,MinGW-w64

Step 2: Source Code Acquisition
git clone https://github.com/ggml-org/llama.cpp

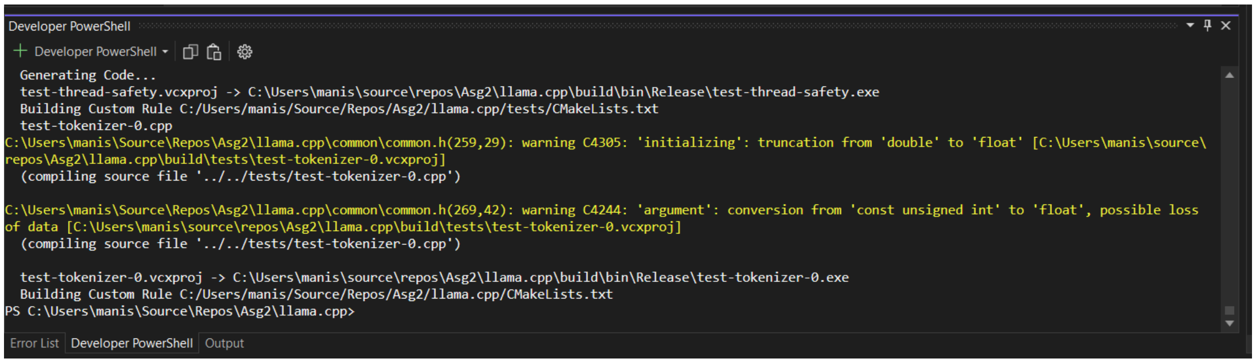
Change current directory
cd llama.cpp

Step 3: Configure the project
Build llama.cpp using CMake -DLLAMA_CURL=OFF
cmake -B build



```
-- Selecting Windows SDK version 10.0.26100.0 to target Windows 10.0.26200.
CMAKE_BUILD_TYPE=
-- Warning: ccache not found - consider installing it for faster compilation or disable this warning with GGML_CCACHE=OFF
-- CMAKE_SYSTEM_PROCESSOR: AMD64
-- CMAKE_GENERATOR_PLATFORM:
-- GGML_SYSTEM_ARCH: x86
-- Including CPU backend
-- x86 detected
-- Adding CPU backend variant ggml-cpu: /arch:AVX512 GGML_AVX512
-- ggml version: 0.9.4
-- ggml commit:  e7da30b58
-- Configuring done (1.9s)
-- Generating done (2.1s)
-- Build files have been written to: C:/Users/manis/source/repos/Asg2/llama.cpp/build
PS C:\Users\manis\Source\Repos\Asg2\llama.cpp>
```

Step 4: Build project
cmake --build build --config Release



```
  Generating Code...
  test-thread-safety.vcxproj -> C:\Users\manis\source\repos\Asg2\llama.cpp\build\bin\Release\test-thread-safety.exe
  Building Custom Rule C:/Users/manis/Source/Repos/Asg2/llama.cpp/tests/CMakeLists.txt
  test-tokenizer-0.cpp
C:\Users\manis\Source\Repos\Asg2\llama.cpp\common\common.h(259,29): warning C4305: 'initializing': truncation from 'double' to 'float' [C:\Users\manis\source\
repos\Asg2\llama.cpp\build\tests\test-tokenizer-0.vcxproj]
  (compiling source file '../../tests/test-tokenizer-0.cpp')

C:\Users\manis\Source\Repos\Asg2\llama.cpp\common\common.h(269,42): warning C4244: 'argument': conversion from 'const unsigned int' to 'float', possible loss
of data [C:\Users\manis\source\repos\Asg2\llama.cpp\build\tests\test-tokenizer-0.vcxproj]
  (compiling source file '../../tests/test-tokenizer-0.cpp')

  test-tokenizer-0.vcxproj -> C:\Users\manis\source\repos\Asg2\llama.cpp\build\bin\Release\test-tokenizer-0.exe
  Building Custom Rule C:/Users/manis/Source/Repos/Asg2/llama.cpp/CMakeLists.txt
PS C:\Users\manis\Source\Repos\Asg2\llama.cpp>
```

Step 5: Install necessary Python packages required to run llama.cpp
pip install -r llama.cpp/requirements.txt
pip install transformers

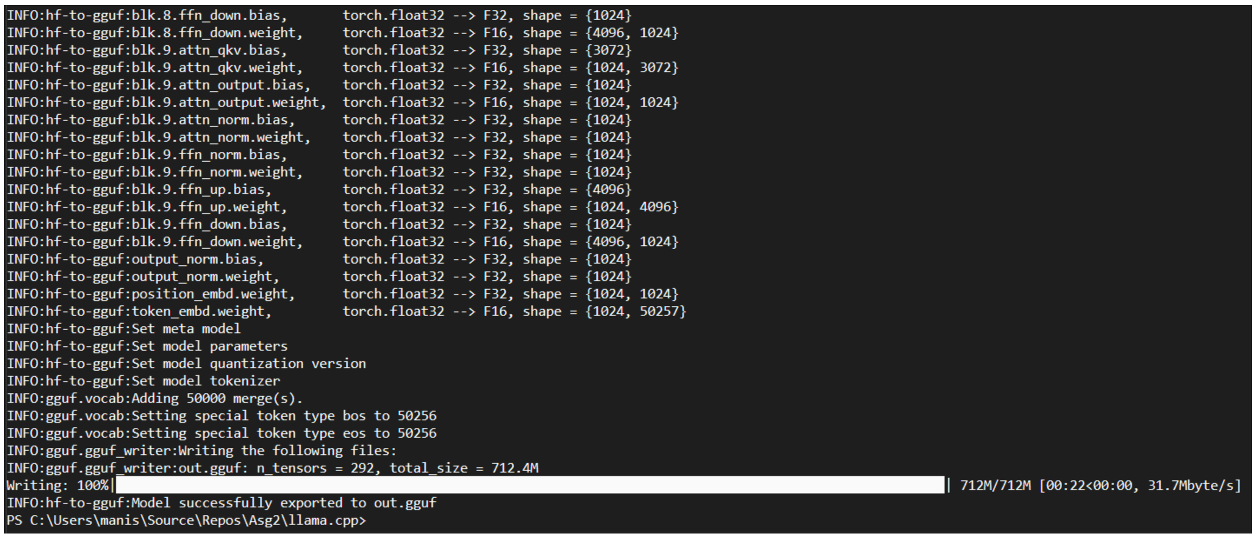
Task 2: Setting up GPT Model

A. Download GPT-2 Medium

git clone https://huggingface.co/openai-community/gpt2-medium

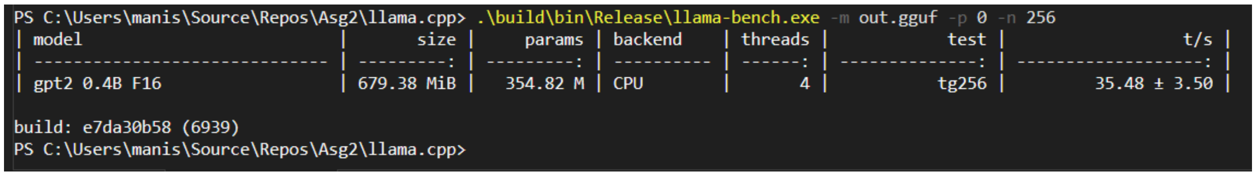
```
Cloning into 'C:\Users\manis\source\repos\Asg2\gpt2-medium'...
POST git-upload-pack (183 bytes)
Remote: Enumerating objects: 76, done.
Remote: Counting objects:  33% (1/3)
Remote: Counting objects:  66% (2/3)
Remote: Counting objects: 100% (3/3)
Remote: Counting objects: 100% (3/3), done.
Remote: Compressing objects:  50% (1/2)
Remote: Compressing objects: 100% (2/2)
Remote: Compressing objects: 100% (2/2), done.
Remote: Total 76 (delta 0), reused 0 (delta 0), pack-reused 73 (from 1)
Filtering content:  25% (2/8)
Filtering content:  25% (2/8), 2.64 GiB | 150.71 MiB/s
Filtering content:  37% (3/8), 2.64 GiB | 150.71 MiB/s
Filtering content:  37% (3/8), 4.15 GiB | 37.63 MiB/s
Filtering content:  50% (4/8), 4.15 GiB | 37.63 MiB/s
Filtering content:  50% (4/8), 5.67 GiB | 4.22 MiB/s
Filtering content:  62% (5/8), 5.67 GiB | 4.22 MiB/s
Filtering content:  62% (5/8), 7.09 GiB | 19.23 MiB/s
Filtering content:  75% (6/8), 7.09 GiB | 19.23 MiB/s
Filtering content:  75% (6/8), 8.50 GiB | 24.94 MiB/s
Filtering content:  87% (7/8), 8.50 GiB | 24.94 MiB/s
Filtering content:  87% (7/8), 10.11 GiB | 5.79 MiB/s
Filtering content: 100% (8/8), 10.11 GiB | 5.79 MiB/s
Filtering content: 100% (8/8), 11.63 GiB | 16.00 MiB/s
Filtering content: 100% (8/8), 11.63 GiB | 12.57 MiB/s, done.
Opening repositories:
C:\Users\manis\source\repos\Asg2\gpt2-medium
```

B. Convert to .gguf
python3 convert_hf_to_gguf.py C:\Users\manis\source\repos\Asg2\gpt2-medium --outfile out.gguf



```
INFO:hf-to-gguf:blk.8.ffn_down.bias,       torch.float32 --> F32, shape = {1024}
INFO:hf-to-gguf:blk.8.ffn_down.weight,     torch.float32 --> F16, shape = {4096, 1024}
INFO:hf-to-gguf:blk.9.attn_qkv.bias,       torch.float32 --> F32, shape = {3072}
INFO:hf-to-gguf:blk.9.attn_qkv.weight,     torch.float32 --> F16, shape = {1024, 3072}
INFO:hf-to-gguf:blk.9.attn_output.bias,    torch.float32 --> F32, shape = {1024}
INFO:hf-to-gguf:blk.9.attn_output.weight,  torch.float32 --> F16, shape = {1024, 1024}
INFO:hf-to-gguf:blk.9.attn_norm.bias,      torch.float32 --> F32, shape = {1024}
INFO:hf-to-gguf:blk.9.attn_norm.weight,    torch.float32 --> F32, shape = {1024}
INFO:hf-to-gguf:blk.9.ffn_norm.bias,       torch.float32 --> F32, shape = {1024}
INFO:hf-to-gguf:blk.9.ffn_norm.weight,     torch.float32 --> F32, shape = {1024}
INFO:hf-to-gguf:blk.9.ffn_up.bias,         torch.float32 --> F32, shape = {4096}
INFO:hf-to-gguf:blk.9.ffn_up.weight,       torch.float32 --> F16, shape = {1024, 4096}
INFO:hf-to-gguf:blk.9.ffn_down.bias,       torch.float32 --> F32, shape = {1024}
INFO:hf-to-gguf:blk.9.ffn_down.weight,     torch.float32 --> F16, shape = {4096, 1024}
INFO:hf-to-gguf:output_norm.bias,          torch.float32 --> F32, shape = {1024}
INFO:hf-to-gguf:output_norm.weight,        torch.float32 --> F32, shape = {1024}
INFO:hf-to-gguf:position_embd.weight,      torch.float32 --> F32, shape = {1024, 1024}
INFO:hf-to-gguf:token_embd.weight,         torch.float32 --> F16, shape = {1024, 50257}
INFO:hf-to-gguf:Set meta model
INFO:hf-to-gguf:Set model parameters
INFO:hf-to-gguf:Set model quantization version
INFO:hf-to-gguf:Set model tokenizer
INFO:gguf.vocab:Adding 50000 merge(s).
INFO:gguf.vocab:Setting special token type bos to 50256
INFO:gguf.vocab:Setting special token type eos to 50256
INFO:gguf.gguf_writer:Writing the following files:
INFO:gguf.gguf_writer:out.gguf: n_tensors = 292, total_size = 712.4M
Writing: 100%|██████████████████████████████████████████████████████████████| 712M/712M [00:22<00:00, 31.7Mbyte/s]
INFO:hf-to-gguf:Model successfully exported to out.gguf
PS C:\Users\manis\Source\Repos\Asg2\llama.cpp>
```

C. Run a Sanity Benchmark
.\build\bin\Release\llama-bench.exe -m gpt2-medium.gguf -p 0 -n 256



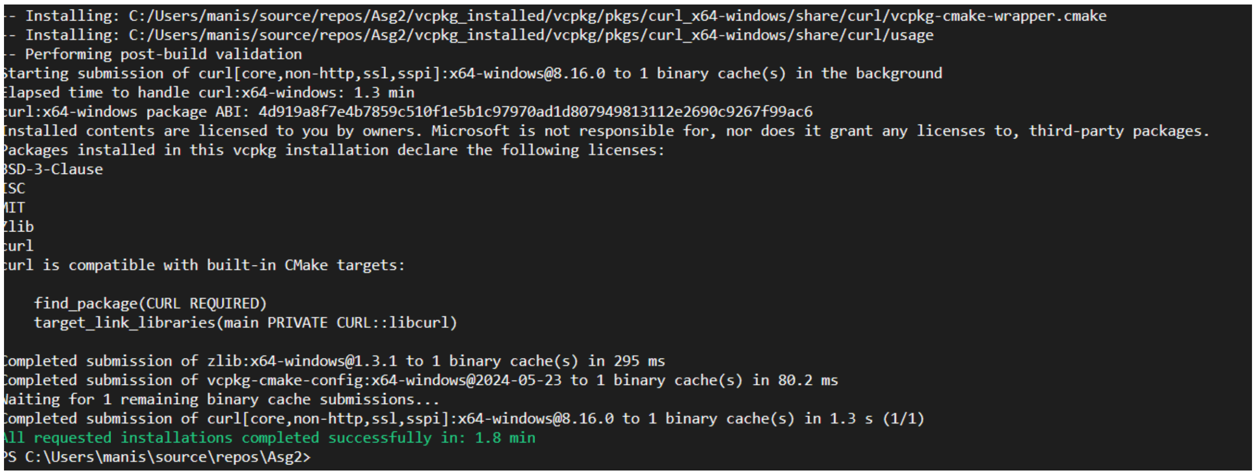
```
PS C:\Users\manis\Source\Repos\Asg2\llama.cpp> .\build\bin\Release\llama-bench.exe -m out.gguf -p 0 -n 256
| model                          |       size |     params | backend    | threads |            test |                  t/s |
| ------------------------------ | ---------: | ---------: | ---------- | ------: | --------------: | -------------------: |
| gpt2 0.4B F16                  | 679.38 MiB |   354.82 M | CPU        |       4 |           tg256 |         35.48 ± 3.50 |

build: e7da30b58 (6939)
PS C:\Users\manis\Source\Repos\Asg2\llama.cpp>
```

Task 3: Naive Execution (No Parallelism)

Step 1:
Install vcpkg:
git clone https://github.com/microsoft/vcpkg
cd vcpkg
./bootstrap-vcpkg.bat
./vcpkg integrate install

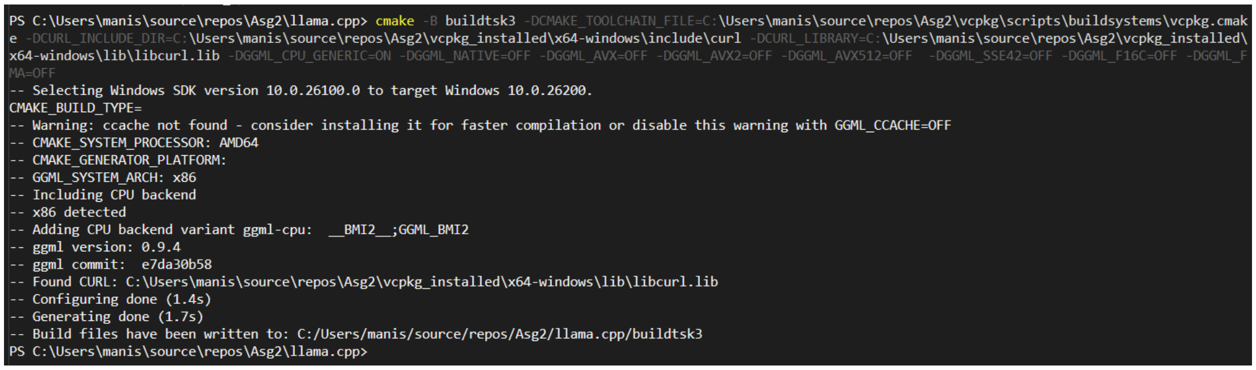
Install curl
vcpkg x-update-baseline --add-initial-baseline

vcpkg install curl:x64-windows



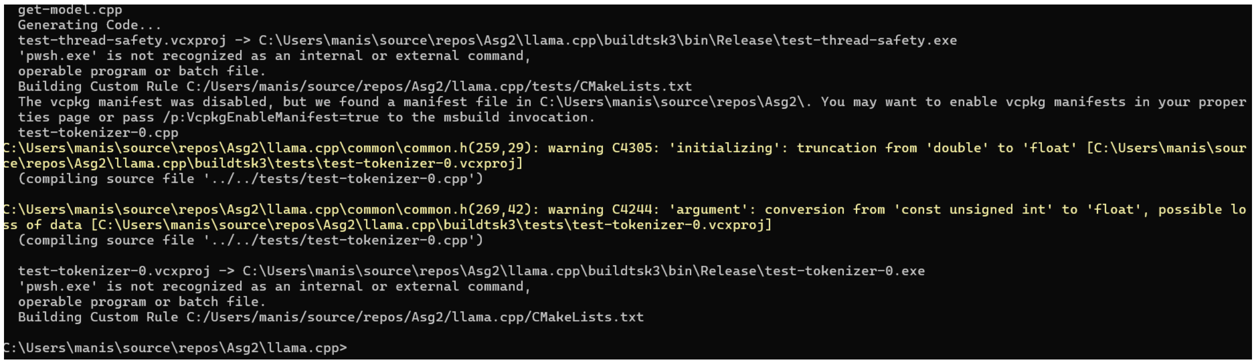
```
-- Installing: C:/Users/manis/source/repos/Asg2/vcpkg_installed/vcpkg/pkgs/curl_x64-windows/share/curl/vcpkg-cmake-wrapper.cmake
-- Installing: C:/Users/manis/source/repos/Asg2/vcpkg_installed/vcpkg/pkgs/curl_x64-windows/share/curl/usage
-- Performing post-build validation
Starting submission of curl[core,non-http,ssl,sspi]:x64-windows@8.16.0 to 1 binary cache(s) in the background
Elapsed time to handle curl:x64-windows: 1.3 min
curl:x64-windows package ABI: 4d919a8f7e4b7859c510f1e5b1c97970ad1d807949813112e2690c9267f99ac6
Installed contents are licensed to you by owners. Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.
Packages installed in this vcpkg installation declare the following licenses:
BSD-3-Clause
ISC
MIT
Zlib
curl
curl is compatible with built-in CMake targets:

    find_package(CURL REQUIRED)
    target_link_libraries(main PRIVATE CURL::libcurl)

Completed submission of zlib:x64-windows@1.3.1 to 1 binary cache(s) in 295 ms
Completed submission of vcpkg-cmake-config:x64-windows@2024-05-23 to 1 binary cache(s) in 80.2 ms
Waiting for 1 remaining binary cache submissions...
Completed submission of curl[core,non-http,ssl,sspi]:x64-windows@8.16.0 to 1 binary cache(s) in 1.3 s (1/1)
All requested installations completed successfully in: 1.8 min
PS C:\Users\manis\source\repos\Asg2>
```

Step 2: Configure the project
cmake -B buildtsk3
-DCMAKE_TOOLCHAIN_FILE=C:\Users\manis\source\repos\Asg2\vcpkg\scripts\buildsystems\vcpkg.cm
ake
-DCURL_INCLUDE_DIR=C:\Users\manis\source\repos\Asg2\vcpkg_installed\x64-windows\include\curl
-DCURL_LIBRARY=C:\Users\manis\source\repos\Asg2\vcpkg_installed\x64-windows\lib\libcurl.lib
-DGGML_CPU_GENERIC=ON -DGGML_NATIVE=OFF -DGGML_AVX=OFF -DGGML_AVX2=OFF
-DGGML_AVX512=OFF -DGGML_SSE42=OFF -DGGML_F16C=OFF -DGGML_FMA=OFF



```
PS C:\Users\manis\source\repos\Asg2\llama.cpp> cmake -B buildtsk3 -DCMAKE_TOOLCHAIN_FILE=C:\Users\manis\source\repos\Asg2\vcpkg\scripts\buildsystems\vcpkg.cmak
e -DCURL_INCLUDE_DIR=C:\Users\manis\source\repos\Asg2\vcpkg_installed\x64-windows\include\curl -DCURL_LIBRARY=C:\Users\manis\source\repos\Asg2\vcpkg_installed\
x64-windows\lib\libcurl.lib -DGGML_CPU_GENERIC=ON -DGGML_NATIVE=OFF -DGGML_AVX=OFF -DGGML_AVX2=OFF -DGGML_AVX512=OFF  -DGGML_SSE42=OFF -DGGML_F16C=OFF -DGGML_F
MA=OFF
-- Selecting Windows SDK version 10.0.26100.0 to target Windows 10.0.26200.
CMAKE_BUILD_TYPE=
-- Warning: ccache not found - consider installing it for faster compilation or disable this warning with GGML_CCACHE=OFF
-- CMAKE_SYSTEM_PROCESSOR: AMD64
-- CMAKE_GENERATOR_PLATFORM:
-- GGML_SYSTEM_ARCH: x86
-- Including CPU backend
-- x86 detected
-- Adding CPU backend variant ggml-cpu:  __BMI2__;GGML_BMI2
-- ggml version: 0.9.4
-- ggml commit:  e7da30b58
-- Found CURL: C:\Users\manis\source\repos\Asg2\vcpkg_installed\x64-windows\lib\libcurl.lib
-- Configuring done (1.4s)
-- Generating done (1.7s)
-- Build files have been written to: C:/Users/manis/source/repos/Asg2/llama.cpp/buildtsk3
PS C:\Users\manis\source\repos\Asg2\llama.cpp>
```

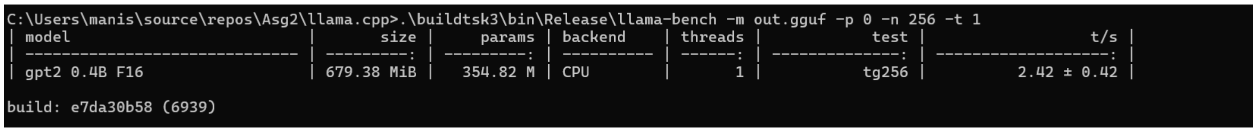
Step 3: Build project
cmake --build buildtsk3 --config Release



```
  get-model.cpp
  Generating Code...
  test-thread-safety.vcxproj -> C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk3\bin\Release\test-thread-safety.exe
  'pwsh.exe' is not recognized as an internal or external command,
  operable program or batch file.
  Building Custom Rule C:/Users/manis/source/repos/Asg2/llama.cpp/tests/CMakeLists.txt
  The vcpkg manifest was disabled, but we found a manifest file in C:\Users\manis\source\repos\Asg2\. You may want to enable vcpkg manifests in your proper
  ties page or pass /p:VcpkgEnableManifest=true to the msbuild invocation.
  test-tokenizer-0.cpp
C:\Users\manis\source\repos\Asg2\llama.cpp\common\common.h(259,29): warning C4305: 'initializing': truncation from 'double' to 'float' [C:\Users\manis\sour
ce\repos\Asg2\llama.cpp\buildtsk3\tests\test-tokenizer-0.vcxproj]
  (compiling source file '../../tests/test-tokenizer-0.cpp')

C:\Users\manis\source\repos\Asg2\llama.cpp\common\common.h(269,42): warning C4244: 'argument': conversion from 'const unsigned int' to 'float', possible lo
ss of data [C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk3\tests\test-tokenizer-0.vcxproj]
  (compiling source file '../../tests/test-tokenizer-0.cpp')

  test-tokenizer-0.vcxproj -> C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk3\bin\Release\test-tokenizer-0.exe
  'pwsh.exe' is not recognized as an internal or external command,
  operable program or batch file.
  Building Custom Rule C:/Users/manis/source/repos/Asg2/llama.cpp/CMakeLists.txt

C:\Users\manis\source\repos\Asg2\llama.cpp>
```

Step 4 : Run Benchmark

.\buildtsk3\bin\Release\llama-bench -m out.gguf -p 0 -n 256 -t 1

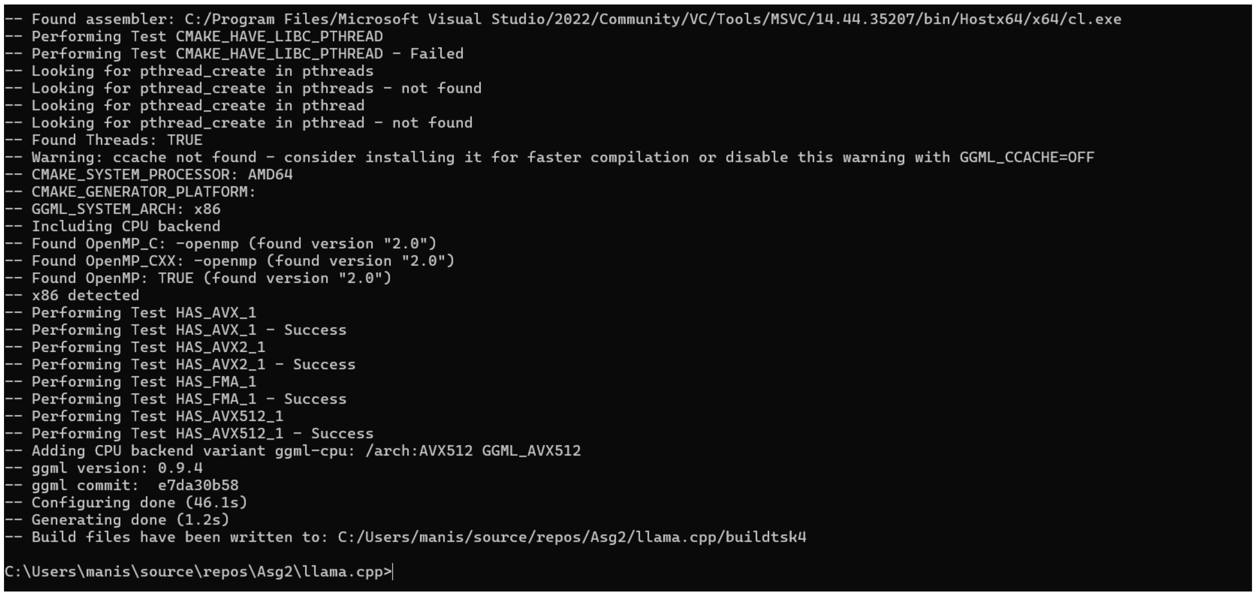


```
C:\Users\manis\source\repos\Asg2\llama.cpp>.\buildtsk3\bin\Release\llama-bench -m out.gguf -p 0 -n 256 -t 1
| model                          |       size |     params | backend    | threads |            test |                  t/s |
| ------------------------------ | ---------: | ---------: | ---------- | ------: | --------------: | -------------------: |
| gpt2 0.4B F16                  | 679.38 MiB |   354.82 M | CPU        |       1 |           tg256 |          2.42 ± 0.42 |

build: e7da30b58 (6939)
```

Task 4: Default Execution

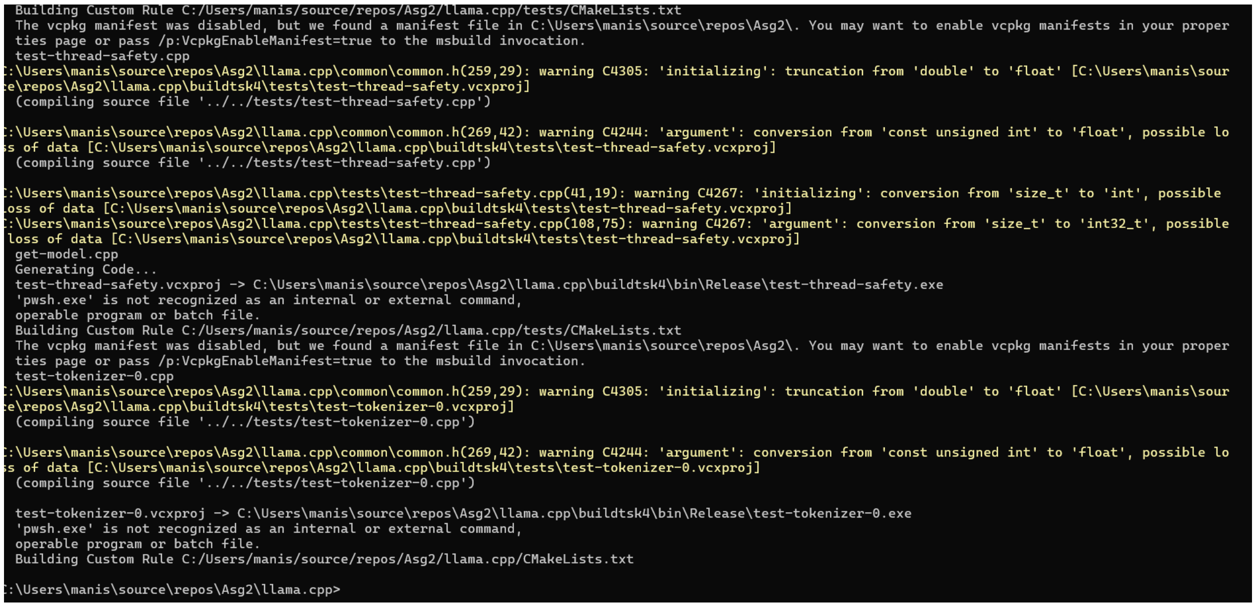
A. Build with default settings

Step 1: Configure project
cmake -B buildtsk4 -DLLAMA_CURL=OFF



```
-- Found assembler: C:/Program Files/Microsoft Visual Studio/2022/Community/VC/Tools/MSVC/14.44.35207/bin/Hostx64/x64/cl.exe
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD - Failed
-- Looking for pthread_create in pthreads
-- Looking for pthread_create in pthreads - not found
-- Looking for pthread_create in pthread
-- Looking for pthread_create in pthread - not found
-- Found Threads: TRUE
-- Warning: ccache not found - consider installing it for faster compilation or disable this warning with GGML_CCACHE=OFF
-- CMAKE_SYSTEM_PROCESSOR: AMD64
-- CMAKE_GENERATOR_PLATFORM:
-- GGML_SYSTEM_ARCH: x86
-- Including CPU backend
-- Found OpenMP_C: -openmp (found version "2.0")
-- Found OpenMP_CXX: -openmp (found version "2.0")
-- Found OpenMP: TRUE (found version "2.0")
-- x86 detected
-- Performing Test HAS_AVX_1
-- Performing Test HAS_AVX_1 - Success
-- Performing Test HAS_AVX2_1
-- Performing Test HAS_AVX2_1 - Success
-- Performing Test HAS_FMA_1
-- Performing Test HAS_FMA_1 - Success
-- Performing Test HAS_AVX512_1
-- Performing Test HAS_AVX512_1 - Success
-- Adding CPU backend variant ggml-cpu: /arch:AVX512 GGML_AVX512
-- ggml version: 0.9.4
-- ggml commit:  e7da30b58
-- Configuring done (46.1s)
-- Generating done (1.2s)
-- Build files have been written to: C:/Users/manis/source/repos/Asg2/llama.cpp/buildtsk4

C:\Users\manis\source\repos\Asg2\llama.cpp>
```

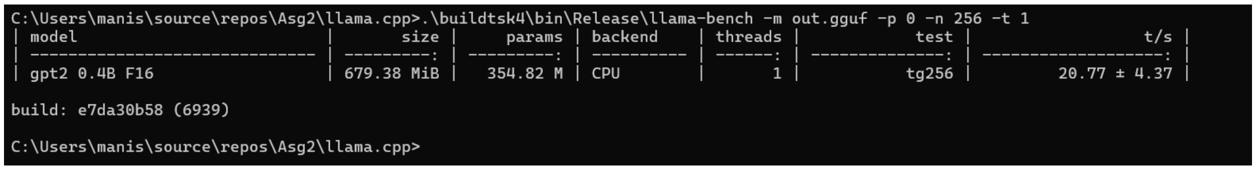
Step 2: Build project
cmake --build buildtsk4 --config Release



```
 Building Custom Rule C:/Users/manis/source/repos/Asg2/llama.cpp/tests/CMakeLists.txt
 The vcpkg manifest was disabled, but we found a manifest file in C:\Users\manis\source\repos\Asg2\. You may want to enable vcpkg manifests in your proper
 ties page or pass /p:VcpkgEnableManifest=true to the msbuild invocation.
 test-thread-safety.cpp
:\Users\manis\source\repos\Asg2\llama.cpp\common\common.h(259,29): warning C4305: 'initializing': truncation from 'double' to 'float' [C:\Users\manis\sour
e\repos\Asg2\llama.cpp\buildtsk4\tests\test-thread-safety.vcxproj]
 (compiling source file '../../tests/test-thread-safety.cpp')

:\Users\manis\source\repos\Asg2\llama.cpp\common\common.h(269,42): warning C4244: 'argument': conversion from 'const unsigned int' to 'float', possible lo
s of data [C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk4\tests\test-thread-safety.vcxproj]
 (compiling source file '../../tests/test-thread-safety.cpp')

:\Users\manis\source\repos\Asg2\llama.cpp\tests\test-thread-safety.cpp(41,19): warning C4267: 'initializing': conversion from 'size_t' to 'int', possible
oss of data [C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk4\tests\test-thread-safety.vcxproj]
:\Users\manis\source\repos\Asg2\llama.cpp\tests\test-thread-safety.cpp(108,75): warning C4267: 'argument': conversion from 'size_t' to 'int32_t', possible
 loss of data [C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk4\tests\test-thread-safety.vcxproj]
 get-model.cpp
 Generating Code...
 test-thread-safety.vcxproj -> C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk4\bin\Release\test-thread-safety.exe
 'pwsh.exe' is not recognized as an internal or external command,
 operable program or batch file.
 Building Custom Rule C:/Users/manis/source/repos/Asg2/llama.cpp/tests/CMakeLists.txt
 The vcpkg manifest was disabled, but we found a manifest file in C:\Users\manis\source\repos\Asg2\. You may want to enable vcpkg manifests in your proper
 ties page or pass /p:VcpkgEnableManifest=true to the msbuild invocation.
 test-tokenizer-0.cpp
:\Users\manis\source\repos\Asg2\llama.cpp\common\common.h(259,29): warning C4305: 'initializing': truncation from 'double' to 'float' [C:\Users\manis\sour
e\repos\Asg2\llama.cpp\buildtsk4\tests\test-tokenizer-0.vcxproj]
 (compiling source file '../../tests/test-tokenizer-0.cpp')

:\Users\manis\source\repos\Asg2\llama.cpp\common\common.h(269,42): warning C4244: 'argument': conversion from 'const unsigned int' to 'float', possible lo
s of data [C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk4\tests\test-tokenizer-0.vcxproj]
 (compiling source file '../../tests/test-tokenizer-0.cpp')

 test-tokenizer-0.vcxproj -> C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk4\bin\Release\test-tokenizer-0.exe
 'pwsh.exe' is not recognized as an internal or external command,
 operable program or batch file.
 Building Custom Rule C:/Users/manis/source/repos/Asg2/llama.cpp/CMakeLists.txt

:\Users\manis\source\repos\Asg2\llama.cpp>
```

B. Run Benchmark

.\buildtsk4\bin\Release\llama-bench -m out.gguf -p 0 -n 256 -t 1



```
C:\Users\manis\source\repos\Asg2\llama.cpp>.\buildtsk4\bin\Release\llama-bench -m out.gguf -p 0 -n 256 -t 1
| model                          |       size |     params | backend    | threads |            test |                  t/s |
| ------------------------------ | ---------: | ---------: | ---------- | ------: | --------------: | -------------------: |
| gpt2 0.4B F16                  | 679.38 MiB |   354.82 M | CPU        |       1 |           tg256 |         20.77 ± 4.37 |

build: e7da30b58 (6939)

C:\Users\manis\source\repos\Asg2\llama.cpp>
```

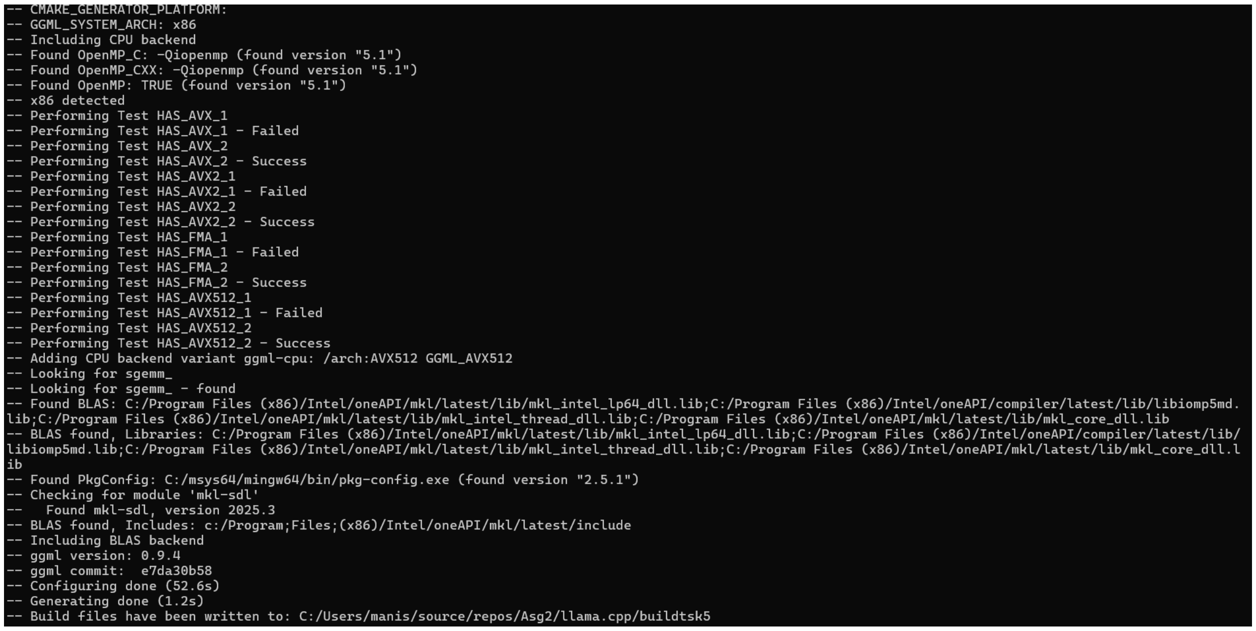
Task 5: Near-Optimal Execution with Intel MKL

A. Rebuild with Intel MKL

Step 1 : Install intel one api base toolkit and Execute setvars.bat from intel oneapi installed folder

Step 2: Configure project in intel oneapi terminal

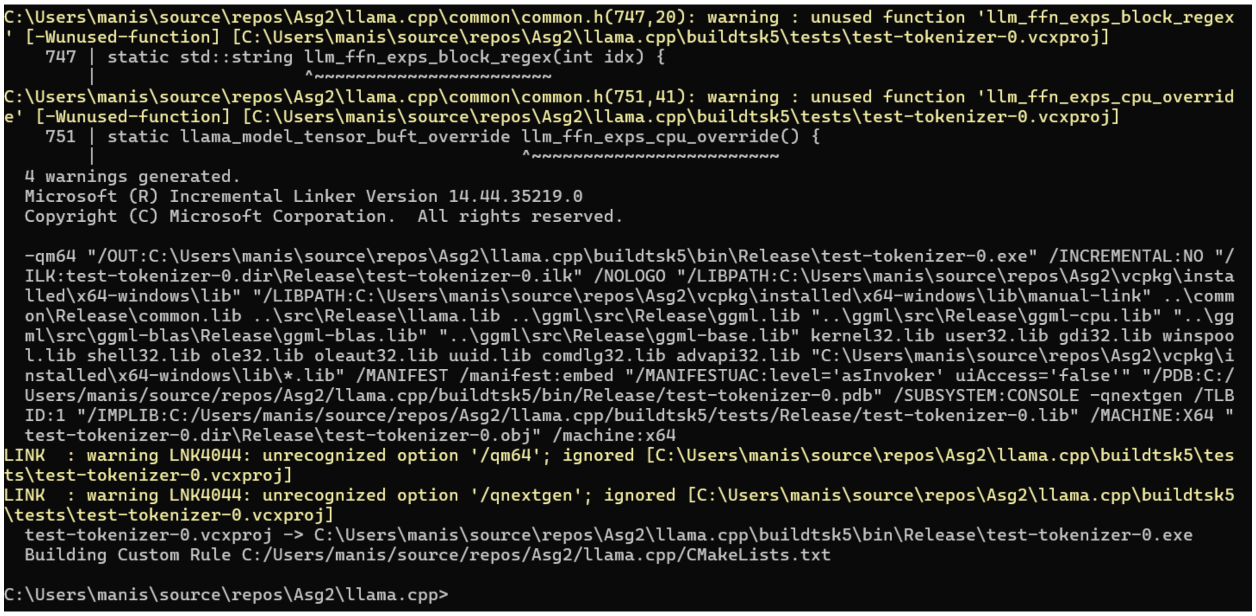
cmake -B buildtsk5 -G "Visual Studio 17 2022" -T "Intel C++ Compiler 2025" -DGGML_BLAS=ON -DGGML_BLAS_VENDOR=Intel10_64lp -DGGML_NATIVE=ON -DLLAMA_CURL=OFF



```
-- CMAKE_GENERATOR_PLATFORM:
-- GGML_SYSTEM_ARCH: x86
-- Including CPU backend
-- Found OpenMP_C: -Qiopenmp (found version "5.1")
-- Found OpenMP_CXX: -Qiopenmp (found version "5.1")
-- Found OpenMP: TRUE (found version "5.1")
-- x86 detected
-- Performing Test HAS_AVX_1
-- Performing Test HAS_AVX_1 - Failed
-- Performing Test HAS_AVX_2
-- Performing Test HAS_AVX_2 - Success
-- Performing Test HAS_AVX2_1
-- Performing Test HAS_AVX2_1 - Failed
-- Performing Test HAS_AVX2_2
-- Performing Test HAS_AVX2_2 - Success
-- Performing Test HAS_FMA_1
-- Performing Test HAS_FMA_1 - Failed
-- Performing Test HAS_FMA_2
-- Performing Test HAS_FMA_2 - Success
-- Performing Test HAS_AVX512_1
-- Performing Test HAS_AVX512_1 - Failed
-- Performing Test HAS_AVX512_2
-- Performing Test HAS_AVX512_2 - Success
-- Adding CPU backend variant ggml-cpu: /arch:AVX512 GGML_AVX512
-- Looking for sgemm_
-- Looking for sgemm_ - found
-- Found BLAS: C:/Program Files (x86)/Intel/oneAPI/mkl/latest/lib/mkl_intel_lp64_dll.lib;C:/Program Files (x86)/Intel/oneAPI/compiler/latest/lib/libiomp5md.
lib;C:/Program Files (x86)/Intel/oneAPI/mkl/latest/lib/mkl_intel_thread_dll.lib;C:/Program Files (x86)/Intel/oneAPI/mkl/latest/lib/mkl_core_dll.lib
-- BLAS found, Libraries: C:/Program Files (x86)/Intel/oneAPI/mkl/latest/lib/mkl_intel_lp64_dll.lib;C:/Program Files (x86)/Intel/oneAPI/compiler/latest/lib/
libiomp5md.lib;C:/Program Files (x86)/Intel/oneAPI/mkl/latest/lib/mkl_intel_thread_dll.lib;C:/Program Files (x86)/Intel/oneAPI/mkl/latest/lib/mkl_core_dll.l
ib
-- Found PkgConfig: C:/msys64/mingw64/bin/pkg-config.exe (found version "2.5.1")
-- Checking for module 'mkl-sdl'
--   Found mkl-sdl, version 2025.3
-- BLAS found, Includes: c:/Program;Files;(x86)/Intel/oneAPI/mkl/latest/include
-- Including BLAS backend
-- ggml version: 0.9.4
-- ggml commit:  e7da30b58
-- Configuring done (52.6s)
-- Generating done (1.2s)
-- Build files have been written to: C:/Users/manis/source/repos/Asg2/llama.cpp/buildtsk5
```

Step 3: Build project

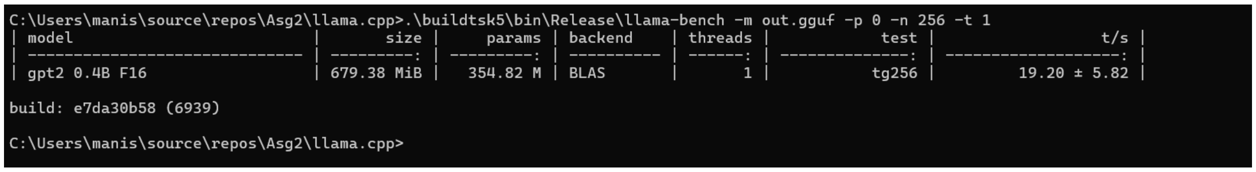
cmake --build buildtsk5 --config Release



```
C:\Users\manis\source\repos\Asg2\llama.cpp\common\common.h(747,20): warning : unused function 'llm_ffn_exps_block_regex
' [-Wunused-function] [C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk5\tests\test-tokenizer-0.vcxproj]
    747 | static std::string llm_ffn_exps_block_regex(int idx) {
        |                    ^~~~~~~~~~~~~~~~~~~~~~~~
C:\Users\manis\source\repos\Asg2\llama.cpp\common\common.h(751,41): warning : unused function 'llm_ffn_exps_cpu_overrid
e' [-Wunused-function] [C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk5\tests\test-tokenizer-0.vcxproj]
    751 | static llama_model_tensor_buft_override llm_ffn_exps_cpu_override() {
        |                                         ^~~~~~~~~~~~~~~~~~~~~~~~~
  4 warnings generated.
  Microsoft (R) Incremental Linker Version 14.44.35219.0
  Copyright (C) Microsoft Corporation.  All rights reserved.

  -qm64 "/OUT:C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk5\bin\Release\test-tokenizer-0.exe" /INCREMENTAL:NO "/
  ILK:test-tokenizer-0.dir\Release\test-tokenizer-0.ilk" /NOLOGO "/LIBPATH:C:\Users\manis\source\repos\Asg2\vcpkg\insta
  lled\x64-windows\lib" "/LIBPATH:C:\Users\manis\source\repos\Asg2\vcpkg\installed\x64-windows\lib\manual-link" ..\comm
  on\Release\common.lib ..\src\Release\llama.lib ..\ggml\src\Release\ggml.lib "..\ggml\src\Release\ggml-cpu.lib" "..\gg
  ml\src\ggml-blas\Release\ggml-blas.lib" "..\ggml\src\Release\ggml-base.lib" kernel32.lib user32.lib gdi32.lib winspoo
  l.lib shell32.lib ole32.lib oleaut32.lib uuid.lib comdlg32.lib advapi32.lib "C:\Users\manis\source\repos\Asg2\vcpkg\i
  nstalled\x64-windows\lib\*.lib" /MANIFEST /manifest:embed "/MANIFESTUAC:level='asInvoker' uiAccess='false'" "/PDB:C:/
  Users/manis/source/repos/Asg2/llama.cpp/buildtsk5/bin/Release/test-tokenizer-0.pdb" /SUBSYSTEM:CONSOLE -qnextgen /TLB
  ID:1 "/IMPLIB:C:/Users/manis/source/repos/Asg2/llama.cpp/buildtsk5/tests/Release/test-tokenizer-0.lib" /MACHINE:X64 "
  test-tokenizer-0.dir\Release\test-tokenizer-0.obj" /machine:x64
LINK  : warning LNK4044: unrecognized option '/qm64'; ignored [C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk5\tes
ts\test-tokenizer-0.vcxproj]
LINK  : warning LNK4044: unrecognized option '/qnextgen'; ignored [C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk5
\tests\test-tokenizer-0.vcxproj]
  test-tokenizer-0.vcxproj -> C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk5\bin\Release\test-tokenizer-0.exe
  Building Custom Rule C:/Users/manis/source/repos/Asg2/llama.cpp/CMakeLists.txt

C:\Users\manis\source\repos\Asg2\llama.cpp>
```

B. Run Benchmark

.\buildtsk5\bin\Release\llama-bench -m out.gguf -p 0 -n 256 -t 1



```
C:\Users\manis\source\repos\Asg2\llama.cpp>.\buildtsk5\bin\Release\llama-bench -m out.gguf -p 0 -n 256 -t 1
| model                          |       size |     params | backend    | threads |            test |                  t/s |
| ------------------------------ | ---------: | ---------: | ---------- | ------: | --------------: | -------------------: |
| gpt2 0.4B F16                  | 679.38 MiB |   354.82 M | BLAS       |       1 |           tg256 |         19.20 ± 5.82 |

build: e7da30b58 (6939)

C:\Users\manis\source\repos\Asg2\llama.cpp>
```

Task 6 : Report Floating-Point Performance Counters

System Details : Dell Inspiron 15 5510 , Windows 11 Home
CPU details : Intel(R) microarchitecture code named Tigerlake
11th Gen Intel(R) Core(TM) i5-11320H @ 3.20GHz
Base speed: 2.50 GHz (Base Speed for sustained all-core load
Thermal Design Power (TDP)of 28 W)
Cores: 4

Intel VTune profiler has been used to get the floating point counter and memory read/write counter.

Floating Point Counters (Using Intel VTune profiler):

FP_ARITH_INST_RETIRED.SCALAR_SINGLE	**Single** (32-bit), 1 FLOP Single Precision (32-bit), Total retired instructions that performed **one single-precision** floating-point operation
FP_ARITH_INST_RETIRED.SCALAR_DOUBLE	**Double** (64-bit), 1 FLOP Double Precision (64-bit), Total retired instructions that performed **one double-precision** floating-point operation
FP_ARITH_INST_RETIRED.128B_PACKED_SINGLE	128-bit (SSE),4 FLOPs Single Precision (32-bit), Total retired instructions that performed **4 single-precision** floating-point operation
FP_ARITH_INST_RETIRED.128B_PACKED_DOUBLE	128-bit (SSE), 2 FLOPs Double Precision (64-bit), Total retired instructions that performed **2 double-precision** floating-point operation
FP_ARITH_INST_RETIRED.256B_PACKED_SINGLE	256-bit (AVX2), 8 FLOPs Single Precision (32-bit), Total retired instructions that performed 8 **single-precision** floating-point operation
FP_ARITH_INST_RETIRED.256B_PACKED_DOUBLE	256-bit (AVX2), 4 FLOPs Double Precision (64-bit), Total retired instructions that performed 4 **double-precision** floating-point operation
FP_ARITH_INST_RETIRED.512B_PACKED_SINGLE	512-bit (AVX-512), 16 FLOPs Single Precision (32-bit), Total retired instructions that performed 16 **single-precision** floating-point operation
FP_ARITH_INST_RETIRED.512B_PACKED_DOUBLE	512-bit (AVX-512), 8 FLOPs Double Precision (64-bit), Total retired instructions that performed 8 **double-precision** floating-point operation

Memory Counter (Using Intel VTune profiler): The system has Dual Channel Memory and here I am considering both read and write operations.

UNC_MC0_RDCAS_COUNT_FREERUN	The total number of DRAM Read CAS (Column Address Strobe) commands issued by the memory controller 0 to retrieve data

	from main memory.Read Traffic (Data moved from DRAM to the CPU's Last-Level Cache)
UNC_MC1_RDCAS_COUNT_FREERUN	The total number of DRAM Read CAS (Column Address Strobe) commands issued by the memory controller 1 to retrieve data from main memory.Read Traffic (Data moved from DRAM to the CPU's Last-Level Cache)
UNC_MC0_WRCAS_COUNT_FREERUN	The total number of DRAM Write CAS commands issued by the memory controller 0 to write data to main memory.Write Traffic (Data moved from the CPU's Last-Level Cache to DRAM)
UNC_MC1_WRCAS_COUNT_FREERUN	The total number of DRAM Write CAS commands issued by the memory controller 0 to write data to main memory.Write Traffic (Data moved from the CPU's Last-Level Cache to DRAM)

Task 7: Performance Counters and Roofline Analysis

System Details : Dell Inspiron 15 5510 , Windows 11 Home
CPU details : Intel(R) microarchitecture code named Tigerlake
11th Gen Intel(R) Core(TM) i5-11320H @ 3.20GHz
Base speed: 2.50 GHz (Base Speed for sustained all-core load
Thermal Design Power (TDP)of 28 W)
Cores: 4

Theoretical Peak Memory Bandwidth (GB/s) :
Data Rate (MT/s): 3200 * 10^6 transfers/sec
Channels: 2
Bus Width: 64 bit
Peak Memory Bandwidth (GB/s) = Data Rate (MT/s) * Bus Width (bits)* Channels * 1 \ Bits per Byte

Peak bandwidth = (3200 * 10^6) * 64 * 2 * (1/8)
= 51200000000 bytes/sec
= 51.2 GB/S

Peak bandwidth = 51.2 GB/S

Theoretical Peak Compute Capacity (GFLOPS) :

Cores: 4 Physical Cores
Frequency: 2.5×10^9 Hz {Base Speed for sustained all-core load Thermal Design
Power (TDP)of 28 W}
Vector Width (VW) : 512 Bits (Supports AVX-512)
FLOPs per Vector Instruction (FPI) : 16 For Single Precision (FP32){512/32=16}
FMA Units per Core (U): 2 (The CPU can execute 2 Fused Multiply-Add (FMA)
instructions per clock cycle)
Operations per FMA (OP):2 {A Fused Multiply-Add instruction performs 2 FLOPs

(1 Multiplication + 1 Addition)}

Total FLOPs per Core per Cycle = FMA Units per Core * Operations per FMA * FLOPs per Vector Instruction
= 2 * 2 * 16
= 64 Flops/cycle

Peak GFLOPS = Frequency (GHz) * Cores * Total FLOPs per Core per Cycle
= 2.5×10^9 * 4 * 64
= 640000000000 flops/s
= 640 GFLOPS

Peak GFLOPS = 640 GFLOPS

Floating Point Counters (Using Intel VTune profiler): The following table contains the counters for all three tasks 3,4 and 5.

Counter	**Task 3 (A)**	**Task 4 (B)**	**Task 5 (C)**	**Flops (D)**	**Total FLOP Task 3 (A*D)**	**Total FLOP Task 4 (B*D)**	**Total FLOP Task 5 (C*D)**
FP_ARITH_INST_RETIRED.SCALAR_SINGLE	474,560,711,840	1,440,002,160	1,120,001,680	1	474,560,711,840	1,440,002,160	1,120,001,680
FP_ARITH_INST_RETIRED.SCALAR_DOUBLE	472,864,709,296	0	0	1	472,864,709,296	0	0
FP_ARITH_INST_RETIRED.128B_PACKED_SINGLE	0	896,001,344	928,001,392	4	0	3,584,005,376	3,712,005,568
FP_ARITH_INST_RETIRED.128B_PACKED_DOUBLE	0	0	0	2	0	0	0
FP_ARITH_INST_RETIRED.256B_PACKED_SINGLE	0	384,000,576	0	8	0	3,072,004,608	0
FP_ARITH_INST_RETIRED.256B_PACKED_DOUBLE	0	0	0	4	0	0	0
FP_ARITH_INST_RETIRED.512B_PACKED_SINGLE	0	60,416,090,624	60,640,090,960	16	0	966,657,449,984	970,241,455,360
FP_ARITH_INST_RETIRED.512B_PACKED_DOUBLE	0	0	0	8	0	0	0
Total					947,425,421,136	974,753,462,128	975,073,462,608

Memory Counter (Using Intel VTune profiler):

Counter	**Task 3 (A)**	**Task 4 (B)**	**Task 5 (C)**	**bit size (D)**	**Total Traffic Byte Task 3 E=(A*D)**	**Total Traffic Byte Task 4 F=(B*D)**	**Total Traffic Byte Task 5 G=(C*D)**
UNC_MC0_RDCAS_COUNT_FREERUN	13,515,763,139	8,198,751,657	7,734,786,603	64	865,008,840,896	524,720,106,048	495,026,342,592
UNC_MC1_RDCAS_COUNT_FREERUN	13,443,103,412	8,190,405,045	7,727,524,845	64	860,358,618,368	524,185,922,880	494,561,590,080
UNC_MC0_WRCAS_COUNT_FREERUN	1,302,855,359	177,546,919	81,355,265	64	83,382,742,976	11,363,002,816	5,206,736,960
UNC_MC1_WRCAS_COUNT_FREERUN	1,304,047,936	177,886,520	81,406,765	64	83,459,067,904	11,384,737,280	5,210,032,960
Total					1,892,209,270,144	1,071,653,769,024	1,000,004,702,592

Operation Intensity = Total Floating-Point Operations (FLOPs) / Total Memory Data Transfer (Bytes)
Achievable Performance = Operation Intensity * Peak Bandwidth

Peak bandwidth = 51.2 GB/S

Task 3 :

Total FLOPS = 947,425,421,136 FLOPs
Total Memory Data Transfer = 1,892,209,270,144 Bytes
Operation Intensity = 947,425,421,136 / 1,892,209,270,144
= 0.5006980127 FLOPs/Byte
Achievable Performance = 0.5006980127 * 51.2
= 25.63573825 GFLOPs/s

Task 4 :

Total FLOPS = 974,753,462,128 FLOPs
Total Memory Data Transfer = 1,071,653,769,024 Bytes
Operation Intensity = 974,753,462,128 / 1,071,653,769,024
= 0.9095787187 FLOPs/Byte
Achievable Performance = 0.9095787187 * 51.2
= 46.5704304 GFLOPs/s

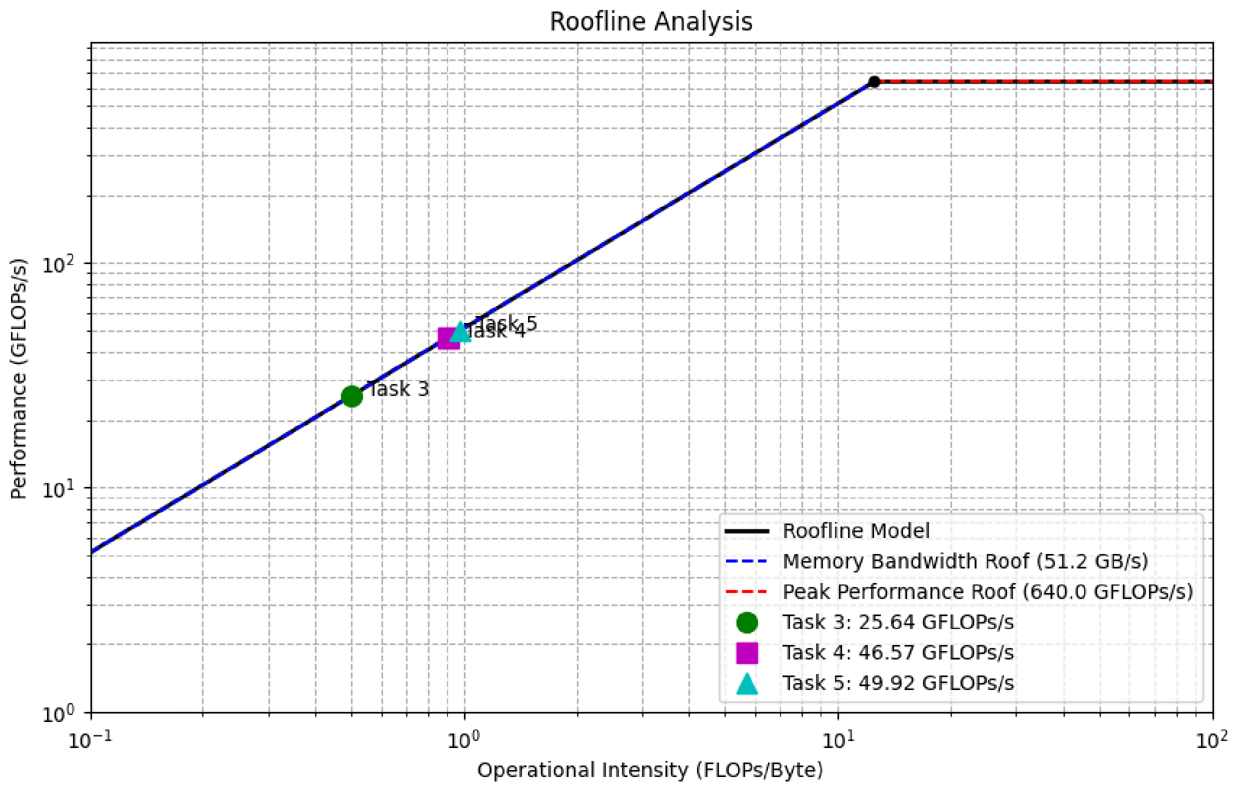
Task 5 :

Total FLOPS = 975,073,462,608 FLOPs
Total Memory Data Transfer = 1,000,004,702,592 Bytes
Operation Intensity = 975,073,462,608 / 1,000,004,702,592
= 0.9750688773 FLOPs/Byte
Achievable Performance = 0.9750688773 * 51.2
= 49.92352652 GFLOPs/s

Summary:

Tasks	Total FLOPS (A)	Total Memory bytes (B)	Operation Intensity (FLOPS/Byte) C=A/B	Theoretical Max Performance D=C*Peak Bandwidth (GFLOPS)
Task 3	947,425,421,136	1,892,209,270,144	0.5006980127	25.63573825
Task 4	974,753,462,128	1,071,653,769,024	0.9095787187	46.5704304
Task 5	975,073,462,608	1,000,004,702,592	0.9750688773	49.92352652

Roofline Analysis plot :



Tasks 3,4 and 5 have a very low operational intensity placing them in the Memory Bound region of the plot. This plot shows that the memory access is the primary bottleneck causing the whole bandwidth utilisation with very low operational intensity.

❖ **Task 3 (Naive Execution):**

The vast majority of floating-point operations are scalar (non-vectorized) and roughly split between single and double precision. This is highly inefficient where SIMD (Single Instruction, Multiple Data) execution is available.

Low Intensity (0.50): The high memory traffic (approx $1.89*10^{12}$ Bytes) relative to the FLOPS results in a low operational intensity.

Bound: **Memory Bound**. The task performance lies below the memory bandwidth roof in the plot hence its a memory bound task. The performance is far below the peak performance of 640 GFLOPS but sits directly on the memory bandwidth ceiling of 51.2 GB/S for its intensity.

- **Task 4 (Default Execution):**

  The task successfully enables vectorization (PACKED instructions). The 512B instructions dominate the count, which is good, but the mix of 128B 256B suggests the vectorization isn't perfect.This task achieves high computational efficiency through vectorization and work reduction but still suffers from substantial memory traffic, making it strongly Memory-Bound.
  Higher Intensity (0.909): Compared to Task 3, the operational intensity nearly doubled. This is achieved by significantly reducing the memory traffic (from approx $1.89*10^{12}$ Bytes to approx $1.07*10^{12}$ Bytes) while keeping the FLOPS count similar. This is likely due to better data locality, better cache use, or a more compact data layout.

  Bound: **Memory Bound**. The task performance lies below the memory bandwidth roof in the plot hence its a memory bound task. The performance of 46.57GFLOPS is still constrained by memory bandwidth, but the performance is almost doubled due to the increase in operational intensity.

- **Task 5 (Near-Optimal Execution with Intel MKL):**

  Optimal vectorization is achieved. The task is almost entirely dominated by FP_ARITH_INST_RETIRED.512B_PACKED_SINGLE, with the 256B instructions from Task 4 eliminated. This shows MKL is perfectly utilizing theAVX-512 capabilities of the Tiger Lake CPU.

  Highest Intensity (0.975): The Intel MKL library, being highly optimized, provides a marginal improvement in intensity over the default execution. MKL likely performs the same core computation (FLOPS) but achieves a slightly more efficient use of memory, minimizing data movement.

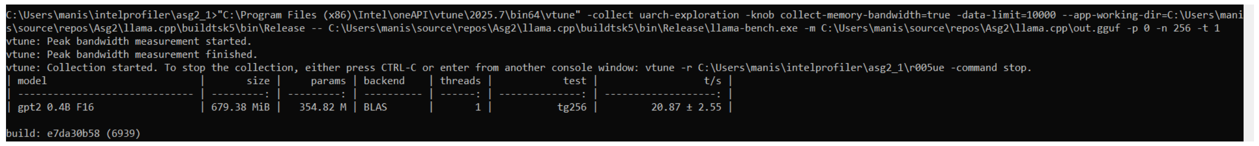
  Bound: **Memory Bound**. The task performance lies below the memory bandwidth roof in the plot hence its a memory bound task. At 49.92GFLOPS, this task is the best performer, pushing very close to the theoretical memory bandwidth limit for the given task.

The roofline plot clearly illustrates that for the given tasks, performance is overwhelmingly limited by memory bandwidth (51.2GB/S), not the CPU's maximum compute capability (640GFLOPS).

The performance improvement from Task 3 to Task 5 is a direct result of optimization efforts that increased the operational intensity (mostly by reducing the required memory traffic), which moved the data point up the memory-bound slope toward the peak performance.

Task 8.Fully Optimal Execution with Thread Scaling

1. Thread Count 1:



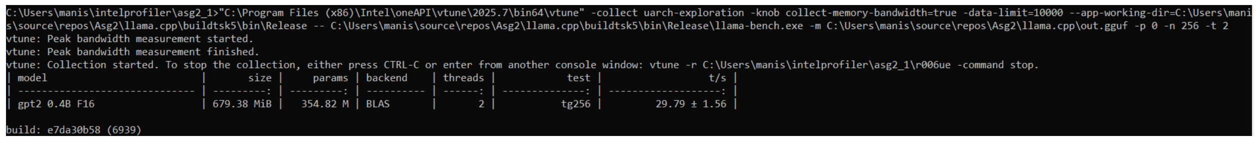
```
C:\Users\manis\intelprofiler\asg2_1>"C:\Program Files (x86)\Intel\oneAPI\vtune\2025.7\bin64\vtune" -collect uarch-exploration -knob collect-memory-bandwidth=true -data-limit=10000 --app-working-dir=C:\Users\mani
s\source\repos\Asg2\llama.cpp\buildtsk5\bin\Release -- C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk5\bin\Release\llama-bench.exe -m C:\Users\manis\source\repos\Asg2\llama.cpp\out.gguf -p 0 -n 256 -t 1
vtune: Peak bandwidth measurement started.
vtune: Peak bandwidth measurement finished.
vtune: Collection started. To stop the collection, either press CTRL-C or enter from another console window: vtune -r C:\Users\manis\intelprofiler\asg2_1\r005ue -command stop.
| model                          |       size |     params | backend    | threads |            test |                  t/s |
| ------------------------------ | ---------: | ---------: | ---------- | ------: | --------------: | -------------------: |
| gpt2 0.4B F16                  | 679.38 MiB |   354.82 M | BLAS       |       1 |           tg256 |         20.87 ± 2.55 |

build: e7da30b58 (6939)
```

Counter	Flops (A)	Counter Value (B)	Total Flop (C = A * B)
FP_ARITH_INST_RETIRED.SCALAR_SINGLE	1	1,120,001,680	1,120,001,680
FP_ARITH_INST_RETIRED.SCALAR_DOUBLE	1	0	0
FP_ARITH_INST_RETIRED.128B_PACKED_SINGLE	4	928,001,392	3,712,005,568
FP_ARITH_INST_RETIRED.128B_PACKED_DOUBLE	2	0	0
FP_ARITH_INST_RETIRED.256B_PACKED_SINGLE	8	0	0
FP_ARITH_INST_RETIRED.256B_PACKED_DOUBLE	4	0	0
FP_ARITH_INST_RETIRED.512B_PACKED_SINGLE	16	60,640,090,960	970,241,455,360
FP_ARITH_INST_RETIRED.512B_PACKED_DOUBLE	8	0	0
Total			975,073,462,608

Counter	Bit Size (A)	Counter Value (B)	Total Traffic Byte Task 3 (C = A * B)
UNC_MC0_RDCAS_COUNT_FREERUN	64	7,734,786,603	495,026,342,592
UNC_MC1_RDCAS_COUNT_FREERUN	64	7,727,524,845	494,561,590,080
UNC_MC0_WRCAS_COUNT_FREERUN	64	81,355,265	5,206,736,960
UNC_MC1_WRCAS_COUNT_FREERUN	64	81,406,765	5,210,032,960
Total			1,000,004,702,592

Total FLOPS = 975,073,462,608 FLOPs
Total Memory Data Transfer = 1,000,004,702,592 Bytes
Operation Intensity = 975,073,462,608 / 1,000,004,702,592
= 0.9750688773 FLOPs/Byte
Achievable Performance = 0.9750688773 * 51.2
= 49.92352652 GFLOPs/s

2. Thread Count 2:



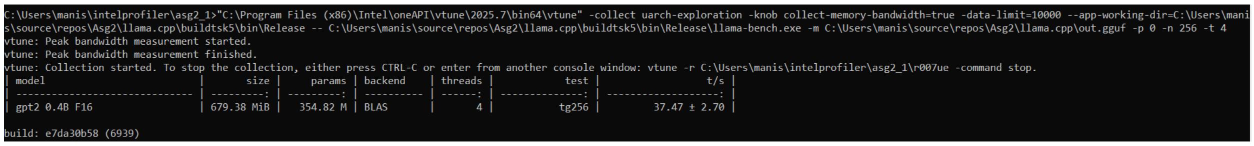
```
C:\Users\manis\intelprofiler\asg2_1>"C:\Program Files (x86)\Intel\oneAPI\vtune\2025.7\bin64\vtune" -collect uarch-exploration -knob collect-memory-bandwidth=true -data-limit=10000 --app-working-dir=C:\Users\mani
s\source\repos\Asg2\llama.cpp\buildtsk5\bin\Release -- C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk5\bin\Release\llama-bench.exe -m C:\Users\manis\source\repos\Asg2\llama.cpp\out.gguf -p 0 -n 256 -t 2
vtune: Peak bandwidth measurement started.
vtune: Peak bandwidth measurement finished.
vtune: Collection started. To stop the collection, either press CTRL-C or enter from another console window: vtune -r C:\Users\manis\intelprofiler\asg2_1\r006ue -command stop.
| model                          |       size |     params | backend    | threads |            test |                  t/s |
| ------------------------------ | ---------: | ---------: | ---------- | ------: | --------------: | -------------------: |
| gpt2 0.4B F16                  | 679.38 MiB |   354.82 M | BLAS       |       2 |           tg256 |         29.79 ± 1.56 |

build: e7da30b58 (6939)
```

Counter	Flops (A)	Counter Value (B)	Total Flop (C = A * B)
FP_ARITH_INST_RETIRED.SCALAR_SINGLE	1	1,088,001,632	1,088,001,632
FP_ARITH_INST_RETIRED.SCALAR_DOUBLE	1	0	0
FP_ARITH_INST_RETIRED.128B_PACKED_SINGLE	4	928,001,392	3,712,005,568
FP_ARITH_INST_RETIRED.128B_PACKED_DOUBLE	2	0	0
FP_ARITH_INST_RETIRED.256B_PACKED_SINGLE	8	0	0
FP_ARITH_INST_RETIRED.256B_PACKED_DOUBLE	4	0	0
FP_ARITH_INST_RETIRED.512B_PACKED_SINGLE	16	60,768,091,152	972,289,458,432
FP_ARITH_INST_RETIRED.512B_PACKED_DOUBLE	8	0	0
Total			977,089,465,632

Counter	Bit Size (A)	Counter Value (B)	Total Traffic Byte Task 3 (C = A * B)
UNC_MC0_RDCAS_COUNT_FREERUN	64	8,018,026,888	513,153,720,832
UNC_MC1_RDCAS_COUNT_FREERUN	64	8,012,201,950	512,780,924,800
UNC_MC0_WRCAS_COUNT_FREERUN	64	142,782,380	9,138,072,320
UNC_MC1_WRCAS_COUNT_FREERUN	64	143,127,094	9,160,134,016
Total			1,044,232,851,968

Total FLOPS = 977,089,465,632 FLOPs
Total Memory Data Transfer = 1,044,232,851,968 Bytes
Operation Intensity = 977,089,465,632 / 1,044,232,851,968
= 0.9357007527 FLOPs/Byte
Achievable Performance = 0.9357007527 * 51.2
= 47.90787854 GFLOPs/s

3. Thread Count 4:



```
C:\Users\manis\intelprofiler\asg2_1>"C:\Program Files (x86)\Intel\oneAPI\vtune\2025.7\bin64\vtune" -collect uarch-exploration -knob collect-memory-bandwidth=true -data-limit=10000 --app-working-dir=C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk5\bin\Release -- C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk5\bin\Release\llama-bench.exe -m C:\Users\manis\source\repos\Asg2\llama.cpp\out.gguf -p 0 -n 256 -t 4
vtune: Peak bandwidth measurement started.
vtune: Peak bandwidth measurement finished.
vtune: Collection started. To stop the collection, either press CTRL-C or enter from another console window: vtune -r C:\Users\manis\intelprofiler\asg2_1\r007ue -command stop.
| model                          |       size |     params | backend    | threads |            test |                  t/s |
| ------------------------------ | ---------: | ---------: | ---------- | ------: | --------------: | -------------------: |
| gpt2 0.4B F16                  | 679.38 MiB |   354.82 M | BLAS       |       4 |           tg256 |         37.47 ± 2.70 |

build: e7da30b58 (6939)
```

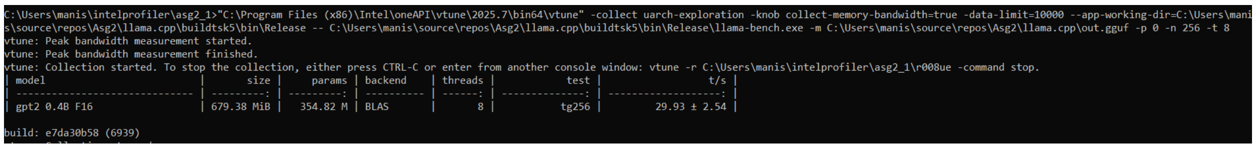
Counter	Flops (A)	Counter Value (B)	Total Flop (C = A * B)
FP_ARITH_INST_RETIRED.SCALAR_SINGLE	1	1,120,001,680	1,120,001,680
FP_ARITH_INST_RETIRED.SCALAR_DOUBLE	1	0	0
FP_ARITH_INST_RETIRED.128B_PACKED_SINGLE	4	928,001,392	3,712,005,568
FP_ARITH_INST_RETIRED.128B_PACKED_DOUBLE	2	0	0
FP_ARITH_INST_RETIRED.256B_PACKED_SINGLE	8	0	0
FP_ARITH_INST_RETIRED.256B_PACKED_DOUBLE	4	0	0
FP_ARITH_INST_RETIRED.512B_PACKED_SINGLE	16	60,480,090,720	967,681,451,520

FP_ARITH_INST_RETIRED.512B_PACKED_DOUBLE	8	0	0
Total			972,513,458,768

Counter	**Bit Size (A)**	**Counter Value (B)**	**Total Traffic Byte Task 3 (C = A * B)**
UNC_MC0_RDCAS_COUNT_FREERUN	64	7,882,935,380	504,507,864,320
UNC_MC1_RDCAS_COUNT_FREERUN	64	7,878,315,345	504,212,182,080
UNC_MC0_WRCAS_COUNT_FREERUN	64	126,200,115	8,076,807,360
UNC_MC1_WRCAS_COUNT_FREERUN	64	126,302,524	8,083,361,536
Total			1,024,880,215,296

Total FLOPS = 972,513,458,768 FLOPs
Total Memory Data Transfer = 1,024,880,215,296 Bytes
Operation Intensity = 972,513,458,768 / 1,024,880,215,296
= 0.9489045103 FLOPs/Byte
Achievable Performance = 0.9489045103 * 51.2
= 48.58391093 GFLOPs/s

4. Thread Count 8:



```
C:\Users\manis\intelprofiler\asg2_1>"C:\Program Files (x86)\Intel\oneAPI\vtune\2025.7\bin64\vtune" -collect uarch-exploration -knob collect-memory-bandwidth=true -data-limit=10000 --app-working-dir=C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk5\bin\Release -- C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk5\bin\Release\llama-bench.exe -m C:\Users\manis\source\repos\Asg2\llama.cpp\out.gguf -p 0 -n 256 -t 8
vtune: Peak bandwidth measurement started.
vtune: Peak bandwidth measurement finished.
vtune: Collection started. To stop the collection, either press CTRL-C or enter from another console window: vtune -r C:\Users\manis\intelprofiler\asg2_1\r008ue -command stop.
| model                          |       size |     params | backend    | threads |            test |                  t/s |
| ------------------------------ | ---------: | ---------: | ---------- | ------: | --------------: | -------------------: |
| gpt2 0.4B F16                  | 679.38 MiB |   354.82 M | BLAS       |       8 |           tg256 |         29.93 ± 2.54 |

build: e7da30b58 (6939)
```

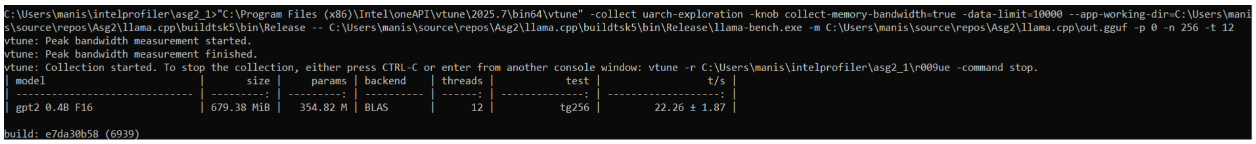
Counter	**Flops (A)**	**Counter Value (B)**	**Total Flop (C = A * B)**
FP_ARITH_INST_RETIRED.SCALAR_SINGLE	1	1,152,001,728	1,152,001,728
FP_ARITH_INST_RETIRED.SCALAR_DOUBLE	1	0	0
FP_ARITH_INST_RETIRED.128B_PACKED_SINGLE	4	960,001,440	3,840,005,760
FP_ARITH_INST_RETIRED.128B_PACKED_DOUBLE	2	0	0
FP_ARITH_INST_RETIRED.256B_PACKED_SINGLE	8	0	0
FP_ARITH_INST_RETIRED.256B_PACKED_DOUBLE	4	0	0
FP_ARITH_INST_RETIRED.512B_PACKED_SINGLE	16	60,800,091,200	972,801,459,200
FP_ARITH_INST_RETIRED.512B_PACKED_DOUBLE	8	0	0
Total			977,793,466,688

Counter	**Bit Size (A)**	**Counter Value (B)**	**Total Traffic Byte Task 3 (C = A * B)**
UNC_MC0_RDCAS_COUNT_FREERUN	64	7,876,023,270	504,065,489,280
UNC_MC1_RDCAS_COUNT_FREERUN	64	7,870,890,031	503,736,961,984

UNC_MC0_WRCAS_COUNT_FREERUN	64	111,370,568	7,127,716,352
UNC_MC1_WRCAS_COUNT_FREERUN	64	111,512,012	7,136,768,768
Total			1,022,066,936,384

Total FLOPS = 977,793,466,688 FLOPs
Total Memory Data Transfer = 1,022,066,936,384 Bytes
Operation Intensity = 977,793,466,688 / 1,022,066,936,384
= 0.9566824167 FLOPs/Byte
Achievable Performance = 0.9566824167 * 51.2
= 48.98213973 GFLOPs/s

5. Thread Count 12:



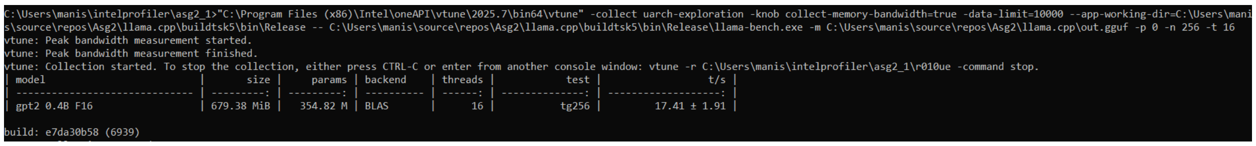
```
C:\Users\manis\intelprofiler\asg2_1>"C:\Program Files (x86)\Intel\oneAPI\vtune\2025.7\bin64\vtune" -collect uarch-exploration -knob collect-memory-bandwidth=true -data-limit=10000 --app-working-dir=C:\Users\mani
s\source\repos\Asg2\llama.cpp\buildtsk5\bin\Release -- C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk5\bin\Release\llama-bench.exe -m C:\Users\manis\source\repos\Asg2\llama.cpp\out.gguf -p 0 -n 256 -t 12
vtune: Peak bandwidth measurement started.
vtune: Peak bandwidth measurement finished.
vtune: Collection started. To stop the collection, either press CTRL-C or enter from another console window: vtune -r C:\Users\manis\intelprofiler\asg2_1\r009ue -command stop.
| model                          |       size |     params | backend    | threads |            test |                  t/s |
| ------------------------------ | ---------: | ---------: | ---------- | ------: | --------------: | -------------------: |
| gpt2 0.4B F16                  | 679.38 MiB |   354.82 M | BLAS       |      12 |           tg256 |         22.26 ± 1.87 |

build: e7da30b58 (6939)
```

Counter	Flops (A)	Counter Value (B)	Total Flop (C = A * B)
FP_ARITH_INST_RETIRED.SCALAR_SINGLE	1	1,120,001,680	1,120,001,680
FP_ARITH_INST_RETIRED.SCALAR_DOUBLE	1	0	0
FP_ARITH_INST_RETIRED.128B_PACKED_SINGLE	4	896,001,344	3,584,005,376
FP_ARITH_INST_RETIRED.128B_PACKED_DOUBLE	2	0	0
FP_ARITH_INST_RETIRED.256B_PACKED_SINGLE	8	0	0
FP_ARITH_INST_RETIRED.256B_PACKED_DOUBLE	4	0	0
FP_ARITH_INST_RETIRED.512B_PACKED_SINGLE	16	61,248,091,872	979,969,469,952
FP_ARITH_INST_RETIRED.512B_PACKED_DOUBLE	8	0	0
Total			984,673,477,008

Counter	Bit Size (A)	Counter Value (B)	Total Traffic Byte Task 3 (C = A * B)
UNC_MC0_RDCAS_COUNT_FREERUN	64	7,727,617,151	494,567,497,664
UNC_MC1_RDCAS_COUNT_FREERUN	64	7,721,007,649	494,144,489,536
UNC_MC0_WRCAS_COUNT_FREERUN	64	56,369,291	3,607,634,624
UNC_MC1_WRCAS_COUNT_FREERUN	64	56,240,129	3,599,368,256
Total			995,918,990,080

Total FLOPS = 984,673,477,008 FLOPs
Total Memory Data Transfer = 995,918,990,080 Bytes
Operation Intensity = 984,673,477,008 / 995,918,990,080
= 0.9887084058 FLOPs/Byte
Achievable Performance = 0.9887084058 * 51.2
= 50.62187038 GFLOPs/s

6. Thread Count 16:



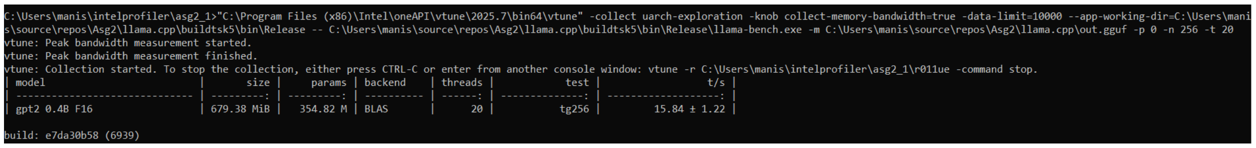
```
C:\Users\manis\intelprofiler\asg2_1>"C:\Program Files (x86)\Intel\oneAPI\vtune\2025.7\bin64\vtune" -collect uarch-exploration -knob collect-memory-bandwidth=true -data-limit=10000 --app-working-dir=C:\Users\mani
s\source\repos\Asg2\llama.cpp\buildtsk5\bin\Release -- C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk5\bin\Release\llama-bench.exe -m C:\Users\manis\source\repos\Asg2\llama.cpp\out.gguf -p 0 -n 256 -t 16
vtune: Peak bandwidth measurement started.
vtune: Peak bandwidth measurement finished.
vtune: Collection started. To stop the collection, either press CTRL-C or enter from another console window: vtune -r C:\Users\manis\intelprofiler\asg2_1\r010ue -command stop.
| model                          |       size |     params | backend    | threads |            test |                  t/s |
| ------------------------------ | ---------: | ---------: | ---------- | ------: | --------------: | -------------------: |
| gpt2 0.4B F16                  | 679.38 MiB |   354.82 M | BLAS       |      16 |           tg256 |         17.41 ± 1.91 |

build: e7da30b58 (6939)
```

Counter	Flops (A)	Counter Value (B)	Total Flop (C = A * B)
FP_ARITH_INST_RETIRED.SCALAR_SINGLE	1	1,120,001,680	1,120,001,680
FP_ARITH_INST_RETIRED.SCALAR_DOUBLE	1	96,000,144	96,000,144
FP_ARITH_INST_RETIRED.128B_PACKED_SINGLE	4	896,001,344	3,584,005,376
FP_ARITH_INST_RETIRED.128B_PACKED_DOUBLE	2	0	0
FP_ARITH_INST_RETIRED.256B_PACKED_SINGLE	8	0	0
FP_ARITH_INST_RETIRED.256B_PACKED_DOUBLE	4	0	0
FP_ARITH_INST_RETIRED.512B_PACKED_SINGLE	16	61,056,091,584	976,897,465,344
FP_ARITH_INST_RETIRED.512B_PACKED_DOUBLE	8	0	0
Total			981,697,472,544

Counter	Bit Size (A)	Counter Value (B)	Total Traffic Byte Task 3 (C = A * B)
UNC_MC0_RDCAS_COUNT_FREERUN	64	7,932,919,460	507,706,845,440
UNC_MC1_RDCAS_COUNT_FREERUN	64	7,925,010,592	507,200,677,888
UNC_MC0_WRCAS_COUNT_FREERUN	64	88,690,655	5,676,201,920
UNC_MC1_WRCAS_COUNT_FREERUN	64	88,933,442	5,691,740,288
Total			1,026,275,465,536

Total FLOPS = 981,697,472,544 FLOPs
Total Memory Data Transfer = 1,026,275,465,536 Bytes
Operation Intensity = 981,697,472,544 / 1,026,275,465,536
= 0.9565633258 FLOPs/Byte
Achievable Performance = 0.9565633258 * 51.2
= 48.97604228 GFLOPs/s

7. Thread Count 20:



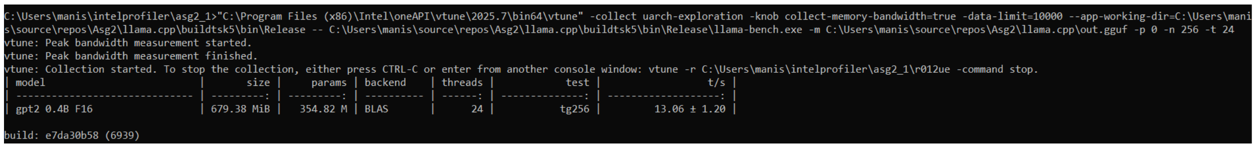
```
C:\Users\manis\intelprofiler\asg2_1>"C:\Program Files (x86)\Intel\oneAPI\vtune\2025.7\bin64\vtune" -collect uarch-exploration -knob collect-memory-bandwidth=true -data-limit=10000 --app-working-dir=C:\Users\mani
s\source\repos\Asg2\llama.cpp\buildtsk5\bin\Release -- C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk5\bin\Release\llama-bench.exe -m C:\Users\manis\source\repos\Asg2\llama.cpp\out.gguf -p 0 -n 256 -t 20
vtune: Peak bandwidth measurement started.
vtune: Peak bandwidth measurement finished.
vtune: Collection started. To stop the collection, either press CTRL-C or enter from another console window: vtune -r C:\Users\manis\intelprofiler\asg2_1\r011ue -command stop.
| model                          |       size |     params | backend    | threads |            test |                  t/s |
| ------------------------------ | ---------: | ---------: | ---------- | ------: | --------------: | -------------------: |
| gpt2 0.4B F16                  | 679.38 MiB |   354.82 M | BLAS       |      20 |           tg256 |         15.84 ± 1.22 |

build: e7da30b58 (6939)
```

Counter	Flops (A)	Counter Value (B)	Total Flop (C = A * B)
FP_ARITH_INST_RETIRED.SCALAR_SINGLE	1	1,120,001,680	1,120,001,680
FP_ARITH_INST_RETIRED.SCALAR_DOUBLE	1	224,000,336	224,000,336
FP_ARITH_INST_RETIRED.128B_PACKED_SINGLE	4	896,001,344	3,584,005,376
FP_ARITH_INST_RETIRED.128B_PACKED_DOUBLE	2	0	0
FP_ARITH_INST_RETIRED.256B_PACKED_SINGLE	8	0	0
FP_ARITH_INST_RETIRED.256B_PACKED_DOUBLE	4	0	0
FP_ARITH_INST_RETIRED.512B_PACKED_SINGLE	16	60,832,091,248	973,313,459,968
FP_ARITH_INST_RETIRED.512B_PACKED_DOUBLE	8	0	0
Total			978,241,467,360

Counter	Bit Size (A)	Counter Value (B)	Total Traffic Byte Task 3 (C = A * B)
UNC_MC0_RDCAS_COUNT_FREERUN	64	7,969,379,347	510,040,278,208
UNC_MC1_RDCAS_COUNT_FREERUN	64	7,961,174,502	509,515,168,128
UNC_MC0_WRCAS_COUNT_FREERUN	64	92,596,759	5,926,192,576
UNC_MC1_WRCAS_COUNT_FREERUN	64	92,498,515	5,919,904,960
Total			1,031,401,543,872

Total FLOPS = 978,241,467,360 FLOPs
Total Memory Data Transfer = 1,031,401,543,872 Bytes
Operation Intensity = 978,241,467,360 / 1,031,401,543,872
= 0.948458409 FLOPs/Byte
Achievable Performance = 0.948458409 * 51.2
= 48.56107054 GFLOPs/s

8. Thread Count 24:



```
C:\Users\manis\intelprofiler\asg2_1>"C:\Program Files (x86)\Intel\oneAPI\vtune\2025.7\bin64\vtune" -collect uarch-exploration -knob collect-memory-bandwidth=true -data-limit=10000 --app-working-dir=C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk5\bin\Release -- C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk5\bin\Release\llama-bench.exe -m C:\Users\manis\source\repos\Asg2\llama.cpp\out.gguf -p 0 -n 256 -t 24
vtune: Peak bandwidth measurement started.
vtune: Peak bandwidth measurement finished.
vtune: Collection started. To stop the collection, either press CTRL-C or enter from another console window: vtune -r C:\Users\manis\intelprofiler\asg2_1\r012ue -command stop.
| model                          |       size |     params | backend    | threads |            test |                  t/s |
| ------------------------------ | ---------: | ---------: | ---------- | ------: | --------------: | -------------------: |
| gpt2 0.4B F16                  | 679.38 MiB |   354.82 M | BLAS       |      24 |           tg256 |         13.06 ± 1.20 |

build: e7da30b58 (6939)
```

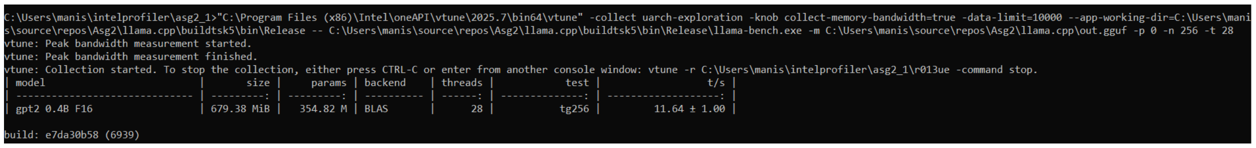
Counter	Flops (A)	Counter Value (B)	Total Flop (C = A * B)
FP_ARITH_INST_RETIRED.SCALAR_SINGLE	1	1,152,001,728	1,152,001,728
FP_ARITH_INST_RETIRED.SCALAR_DOUBLE	1	256,000,384	256,000,384
FP_ARITH_INST_RETIRED.128B_PACKED_SINGLE	4	896,001,344	3,584,005,376
FP_ARITH_INST_RETIRED.128B_PACKED_DOUBLE	2	0	0
FP_ARITH_INST_RETIRED.256B_PACKED_SINGLE	8	0	0
FP_ARITH_INST_RETIRED.256B_PACKED_DOUBLE	4	0	0
FP_ARITH_INST_RETIRED.512B_PACKED_SINGLE	16	61,632,092,448	986,113,479,168

FP_ARITH_INST_RETIRED.512B_PACKED_DOUBLE	8	0	0
Total			991,105,486,656

Counter	Bit Size (A)	Counter Value (B)	Total Traffic Byte Task 3 (C = A * B)
UNC_MC0_RDCAS_COUNT_FREERUN	64	7,961,799,566	509,555,172,224
UNC_MC1_RDCAS_COUNT_FREERUN	64	7,950,498,394	508,831,897,216
UNC_MC0_WRCAS_COUNT_FREERUN	64	83,325,424	5,332,827,136
UNC_MC1_WRCAS_COUNT_FREERUN	64	83,377,104	5,336,134,656
Total			1,029,056,031,232

Total FLOPS = 991,105,486,656 FLOPs
Total Memory Data Transfer = 1,029,056,031,232 Bytes
Operation Intensity = 991,105,486,656 / 1,029,056,031,232
= 0.9631210124 FLOPs/Byte
Achievable Performance = 0.9631210124 * 51.2
= 49.31179584 GFLOPs/s

9. Thread Count 28:



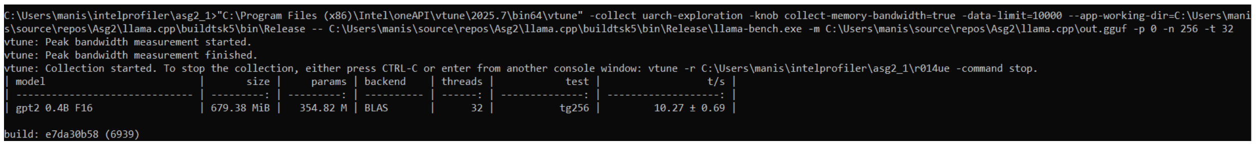
```
C:\Users\manis\intelprofiler\asg2_1>"C:\Program Files (x86)\Intel\oneAPI\vtune\2025.7\bin64\vtune" -collect uarch-exploration -knob collect-memory-bandwidth=true -data-limit=10000 --app-working-dir=C:\Users\mani
s\source\repos\Asg2\llama.cpp\buildtsk5\bin\Release -- C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk5\bin\Release\llama-bench.exe -m C:\Users\manis\source\repos\Asg2\llama.cpp\out.gguf -p 0 -n 256 -t 28
vtune: Peak bandwidth measurement started.
vtune: Peak bandwidth measurement finished.
vtune: Collection started. To stop the collection, either press CTRL-C or enter from another console window: vtune -r C:\Users\manis\intelprofiler\asg2_1\r013ue -command stop.
| model                          |       size |     params | backend    | threads |            test |                  t/s |
| ------------------------------ | ---------: | ---------: | ---------- | ------: | --------------: | -------------------: |
| gpt2 0.4B F16                  | 679.38 MiB |   354.82 M | BLAS       |      28 |           tg256 |         11.64 ± 1.00 |

build: e7da30b58 (6939)
```

Counter	Flops (A)	Counter Value (B)	Total Flop (C = A * B)
FP_ARITH_INST_RETIRED.SCALAR_SINGLE	1	1,152,001,728	1,152,001,728
FP_ARITH_INST_RETIRED.SCALAR_DOUBLE	1	256,000,384	256,000,384
FP_ARITH_INST_RETIRED.128B_PACKED_SINGLE	4	896,001,344	3,584,005,376
FP_ARITH_INST_RETIRED.128B_PACKED_DOUBLE	2	0	0
FP_ARITH_INST_RETIRED.256B_PACKED_SINGLE	8	0	0
FP_ARITH_INST_RETIRED.256B_PACKED_DOUBLE	4	0	0
FP_ARITH_INST_RETIRED.512B_PACKED_SINGLE	16	60,928,091,392	974,849,462,272
FP_ARITH_INST_RETIRED.512B_PACKED_DOUBLE	8	0	0
Total			979,841,469,760

Counter	Bit Size (A)	Counter Value (B)	Total Traffic Byte Task 3 (C = A * B)
UNC_MC0_RDCAS_COUNT_FREERUN	64	8,012,554,013	512,803,456,832
UNC_MC1_RDCAS_COUNT_FREERUN	64	8,000,187,080	512,011,973,120
UNC_MC0_WRCAS_COUNT_FREERUN	64	83,245,864	5,327,735,296
UNC_MC1_WRCAS_COUNT_FREERUN	64	83,298,561	5,331,107,904
Total			1,035,474,273,152

Total FLOPS = 979,841,469,760 FLOPs
Total Memory Data Transfer = 1,035,474,273,152 Bytes
Operation Intensity = 979,841,469,760 / 1,035,474,273,152
= 0.9462731187 FLOPs/Byte
Achievable Performance = 0.9462731187 * 51.2
= 48.44918368 GFLOPs/s

10. Thread Count 32:



```
C:\Users\manis\intelprofiler\asg2_1>"C:\Program Files (x86)\Intel\oneAPI\vtune\2025.7\bin64\vtune" -collect uarch-exploration -knob collect-memory-bandwidth=true -data-limit=10000 --app-working-dir=C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk5\bin\Release -- C:\Users\manis\source\repos\Asg2\llama.cpp\buildtsk5\bin\Release\llama-bench.exe -m C:\Users\manis\source\repos\Asg2\llama.cpp\out.gguf -p 0 -n 256 -t 32
vtune: Peak bandwidth measurement started.
vtune: Peak bandwidth measurement finished.
vtune: Collection started. To stop the collection, either press CTRL-C or enter from another console window: vtune -r C:\Users\manis\intelprofiler\asg2_1\r014ue -command stop.
| model                          |       size |     params | backend    | threads |            test |                  t/s |
| ------------------------------ | ---------: | ---------: | ---------- | ------: | --------------: | -------------------: |
| gpt2 0.4B F16                  | 679.38 MiB |   354.82 M | BLAS       |      32 |           tg256 |         10.27 ± 0.69 |

build: e7da30b58 (6939)
```

Counter	Flops (A)	Counter Value (B)	Total Flop (C = A * B)
FP_ARITH_INST_RETIRED.SCALAR_SINGLE	1	1,088,001,632	1,088,001,632
FP_ARITH_INST_RETIRED.SCALAR_DOUBLE	1	256,000,384	256,000,384
FP_ARITH_INST_RETIRED.128B_PACKED_SINGLE	4	896,001,344	3,584,005,376
FP_ARITH_INST_RETIRED.128B_PACKED_DOUBLE	2	0	0
FP_ARITH_INST_RETIRED.256B_PACKED_SINGLE	8	0	0
FP_ARITH_INST_RETIRED.256B_PACKED_DOUBLE	4	0	0
FP_ARITH_INST_RETIRED.512B_PACKED_SINGLE	16	60,960,091,440	975,361,463,040
FP_ARITH_INST_RETIRED.512B_PACKED_DOUBLE	8	0	0
Total			980,289,470,432

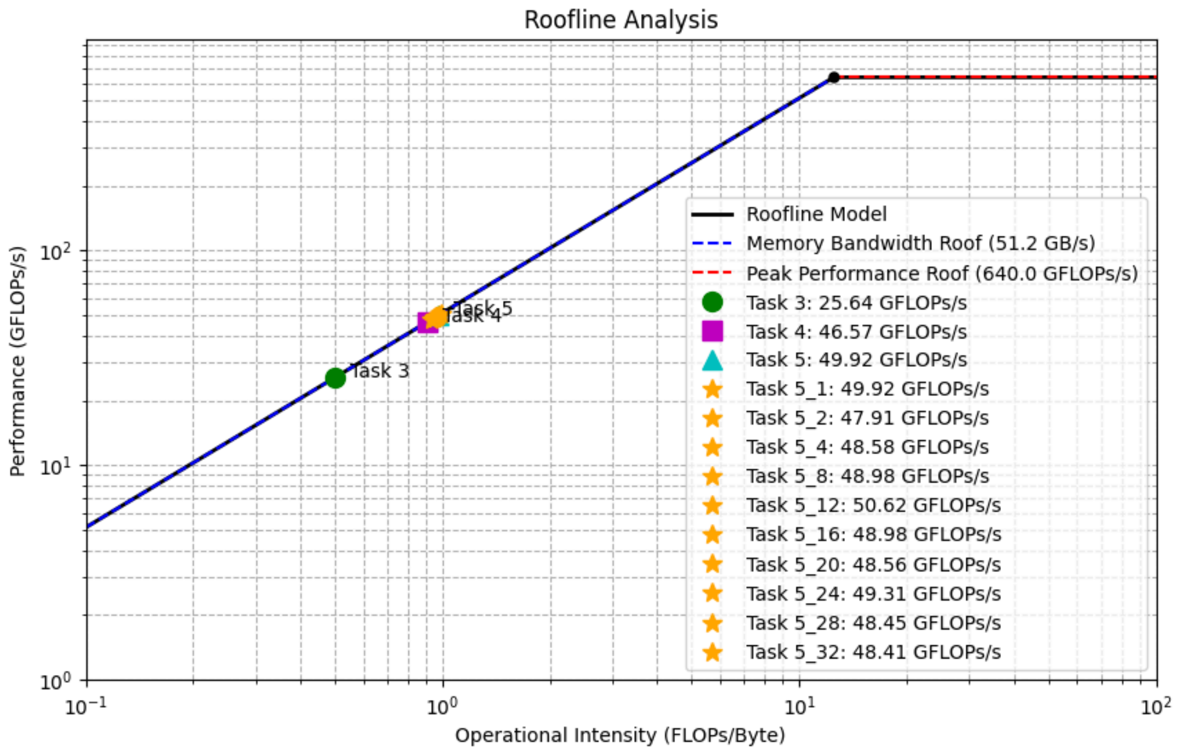
Counter	Bit Size (A)	Counter Value (B)	Total Traffic Byte Task 3 (C = A * B)
UNC_MC0_RDCAS_COUNT_FREERUN	64	8,028,136,911	513,800,762,304
UNC_MC1_RDCAS_COUNT_FREERUN	64	8,014,211,383	512,909,528,512
UNC_MC0_WRCAS_COUNT_FREERUN	64	78,000,492	4,992,031,488
UNC_MC1_WRCAS_COUNT_FREERUN	64	77,885,937	4,984,699,968
Total			1,036,687,022,272

Total FLOPS = 980,289,470,432 FLOPs
Total Memory Data Transfer = 1,036,687,022,272 Bytes
Operation Intensity = 980,289,470,432 / 1,036,687,022,272
= 0.9455982851 FLOPs/Byte
Achievable Performance = 0.9455982851 * 51.2
= 48.4146322 GFLOPs/s

Varying Thread Benchmark Summary :

Threads	Total FLOPS (A)	Total Memory bytes (B)	Operation Intensity (FLOPS/Byte) C=A/B	Theoretical Max Performance D=C*Peak Bandwidth (GFLOPS)
1	975,073,462,608	1,000,004,702,592	0.9750688773	49.92352652
2	977,089,465,632	1,044,232,851,968	0.9357007527	47.90787854
4	972,513,458,768	1,024,880,215,296	0.9489045103	48.58391093
8	977,793,466,688	1,022,066,936,384	0.9566824167	48.98213973
12	984,673,477,008	995,918,990,080	0.9887084058	50.62187038
16	981,697,472,544	1,026,275,465,536	0.9565633258	48.97604228
20	978,241,467,360	1,031,401,543,872	0.948458409	48.56107054
24	991,105,486,656	1,029,056,031,232	0.9631210124	49.31179584
28	979,841,469,760	1,035,474,273,152	0.9462731187	48.44918368
32	980,289,470,432	1,036,687,022,272	0.9455982851	48.4146322

Roofline Analysis Plot :



All varying thread count benchmarks run are Memory bound just like Task 3, Task 4 and Task 5. Their performance is limited by the rate at which the CPU can access data from RAM (the 51.2GB/S Memory bandwidth roof), not the CPU's maximum floating-point calculation rate (640 GFLOPS peak performance roof)

Bound: All varying thread benchmark points are **Memory Bound**. Their performance is limited by the 51.2GB/s memory bandwidth. The calculated performance values (OI*51.2) are near to the line of memory bandwidth roof which shows that system memory bandwidth is fully utilized.

The benchmark with 12 threads achieves the highest performance (50.62 GFLOPs/s), utilizing the memory bandwidth most effectively by having the highest operational intensity.This OI value is achieved by having the lowest Total Memory bytes for the workload, indicating the most successful data reuse and the most efficient communication with the memory hierarchy

Threading Effect: The tight clustering of the points shows that the Intel MKL library successfully manages to maintain a near-optimal performance level across a wide range of thread counts. At higher thread counts (e.g., 28 or 32), performance drops due to increased thread management overhead (like context switching and cache coherence).

- **Threads 1 to 8:** The intensity drops slightly, reflecting the small increase in memory overhead.
- **Threads 12:** This thread count achieves the highest Operation Intensity. This is the most memory-efficient point, corresponding to the lowest memory byte count.
- **Threads 16 to 32:** The intensity decreases for high thread counts, especially at 32 threads, where the intensity is near its lowest point. This confirms that the parallel overhead outweighs the benefits when heavily oversubscribing the CPU, leading to less efficient use of memory bandwidth.