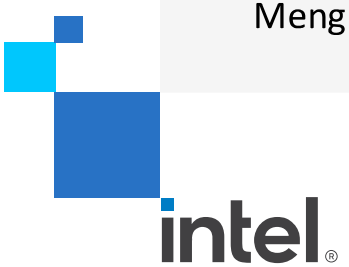
SYCL Backend for Llama.cpp

Meng Hengyu



Outlines

- **llama.cpp brief introduction**

Ecosystem

GEMM Operators: Innovation WOQ

Other GenAI Operators

Static graph and memory management

New backend registration

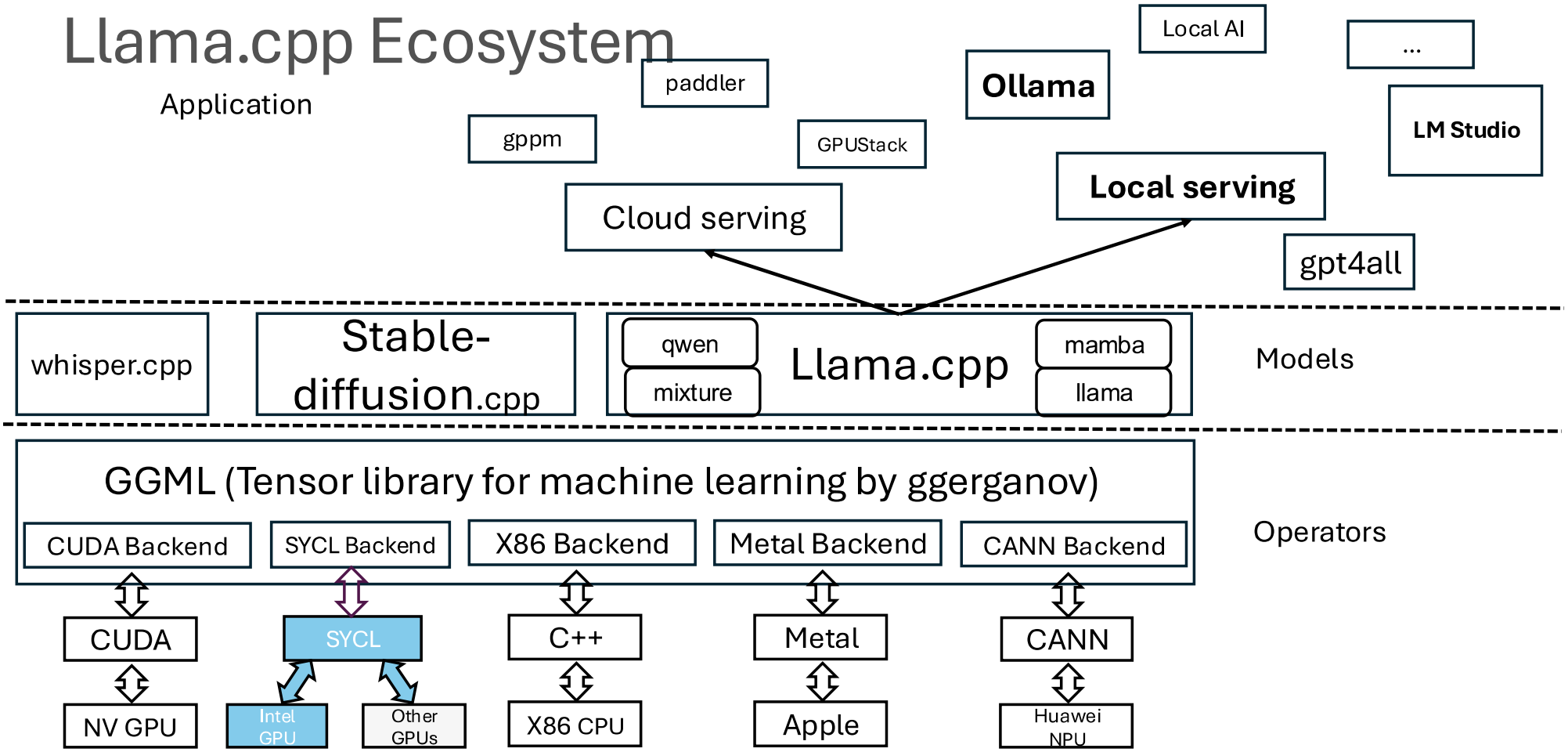
- **Status**

GEMM Operators dispatch

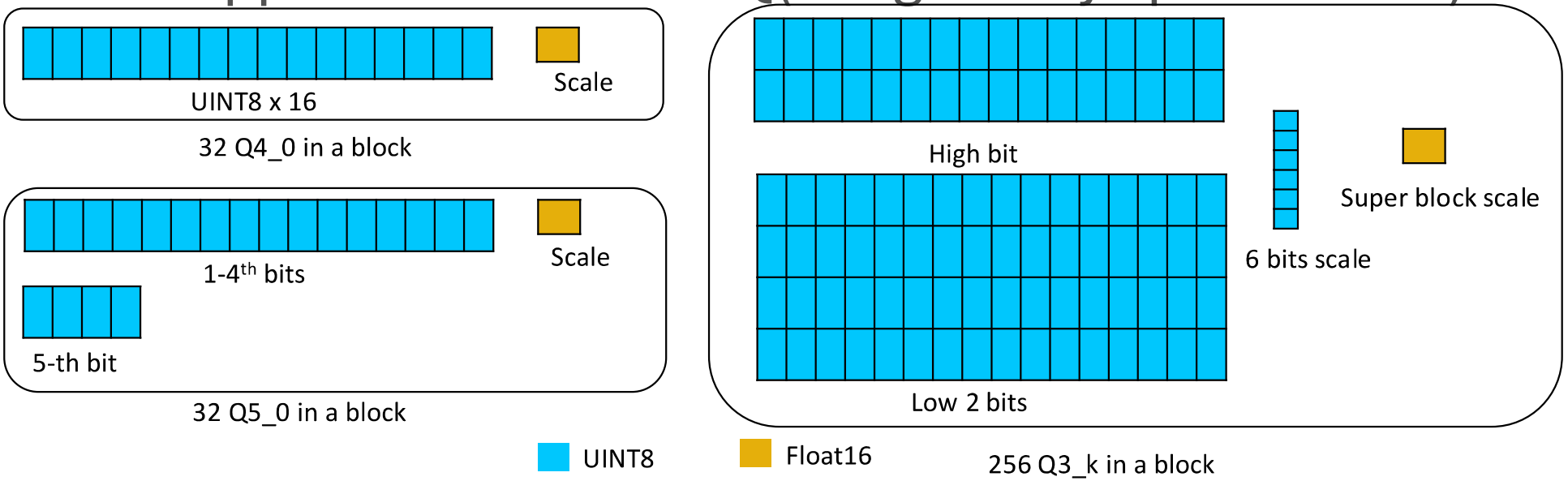
Cross-vendor(NVIDIA/AMD) support by codeplay

- **Future work**

Performance – Flash attention / Lower-level API


Llama.cpp Ecosystem
Application
paddler
gppm
GPUStack
Ollama
Local AI
...
LM Studio
Local serving
Cloud serving
gpt4all
whisper.cpp
Stable-diffusion.cpp
qwen
mixture
Llama.cpp
mamba
llama
Models
GGML (Tensor library for machine learning by ggerganov)
CUDA Backend
SYCL Backend
X86 Backend
Metal Backend
CANN Backend
Operators
CUDA
SYCL
C++
Metal
CANN
NV GPU
Intel GPU
Other GPUs
X86 CPU
Apple
Huawei NPU

Llama.cpp – Innovation WOQ(Weight-only-quantization)

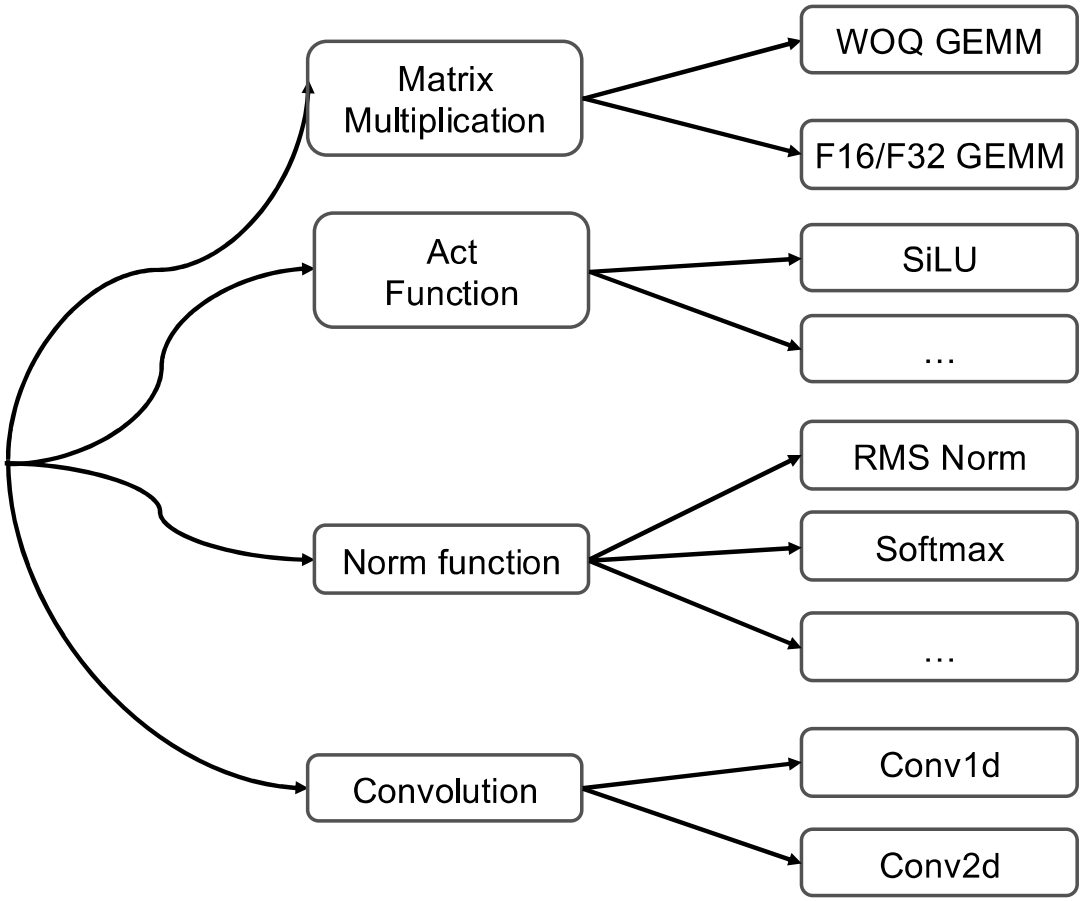


Ultra compression Innovation

1.5-bit, 2-bit, 3-bit, 4-bit, 5-bit, 6-bit, and 8-bit WOQ GEMM for **faster inference and reduced memory usage to fit into consumer GPUs.** For example, it would be quite helpful if llama3-70B with 10K context could be fit into 24G customer GPU.

- QX_0, block_size=32, no zero points: Q4_0, Q5_0, Q8_0
- QX_1, block_size=32, with zero points: Q4_1, Q5_1, Q8_1
- QX_K, block_size=256, quantized scales: Q2_K, Q3_K, Q4_K, Q5_K, Q6_K
- IQX_XXS, block_size=256, with important matrix, super block scales: IQ2_XXS, IQ3_XXS
- IQX_XS, block_size=256, with import matrix: IQ2_XS
- Others: IQ2_S, IQ4_NL, IQ1_M

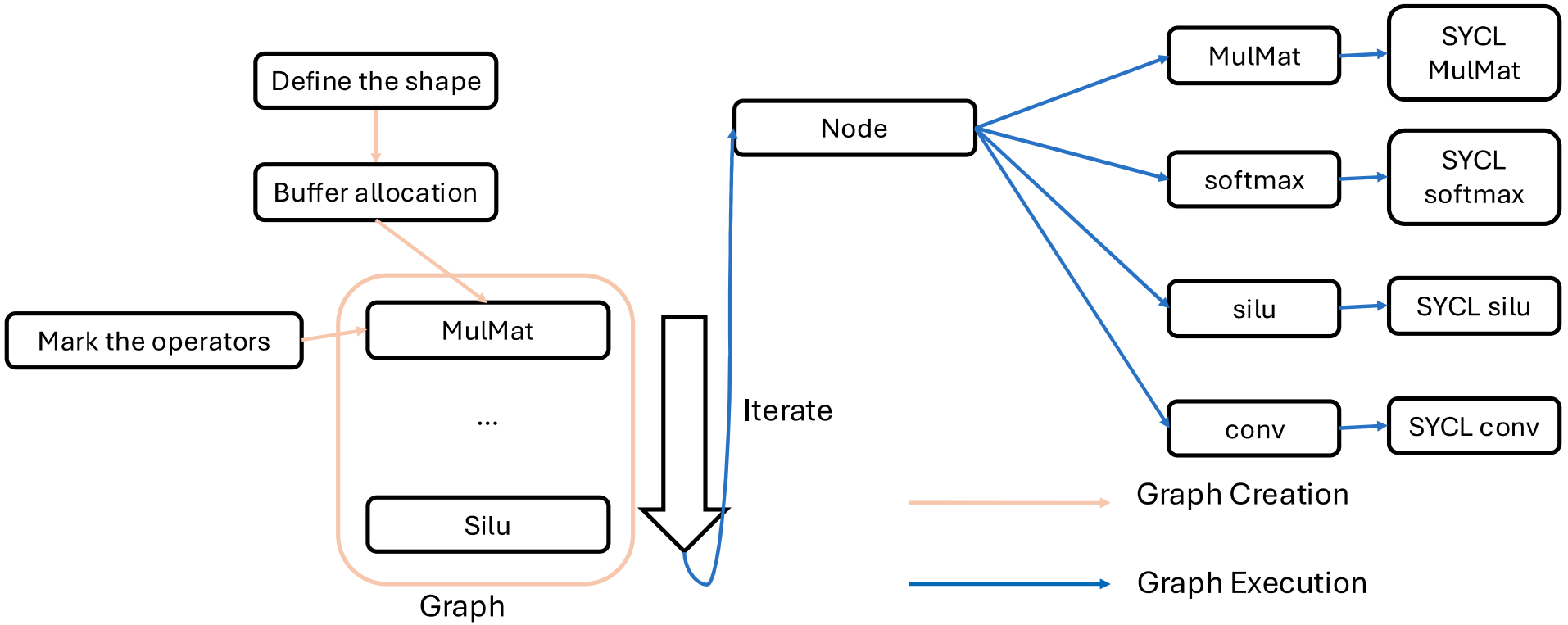
Llama.cpp – Operators Focus on GenAI



	Operators number
PyTorch	>2000
Llama.cpp	107

Llama.cpp/GGML focuses on 107 operators, enough for LLM and stable-diffusion

Llama.cpp –Static Graph



Llama.cpp - New backend registration

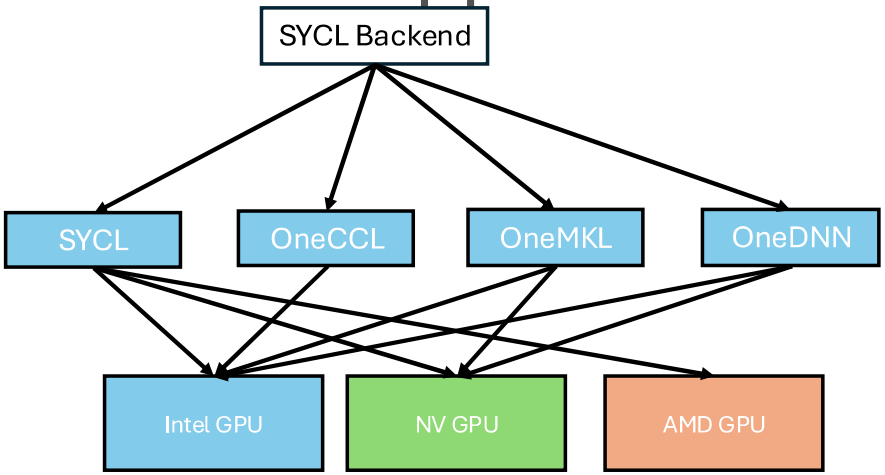
GGML has abstracted the heterogeneous computing as the basic functions, you can easily add a new backend via implementing the following functions:

- memcpy
- alloc/free
- Synchronize
- Compute
- Op support query (for fallback to host)
- Events

1st Step Enabling effort: SYCL backend in 1 month

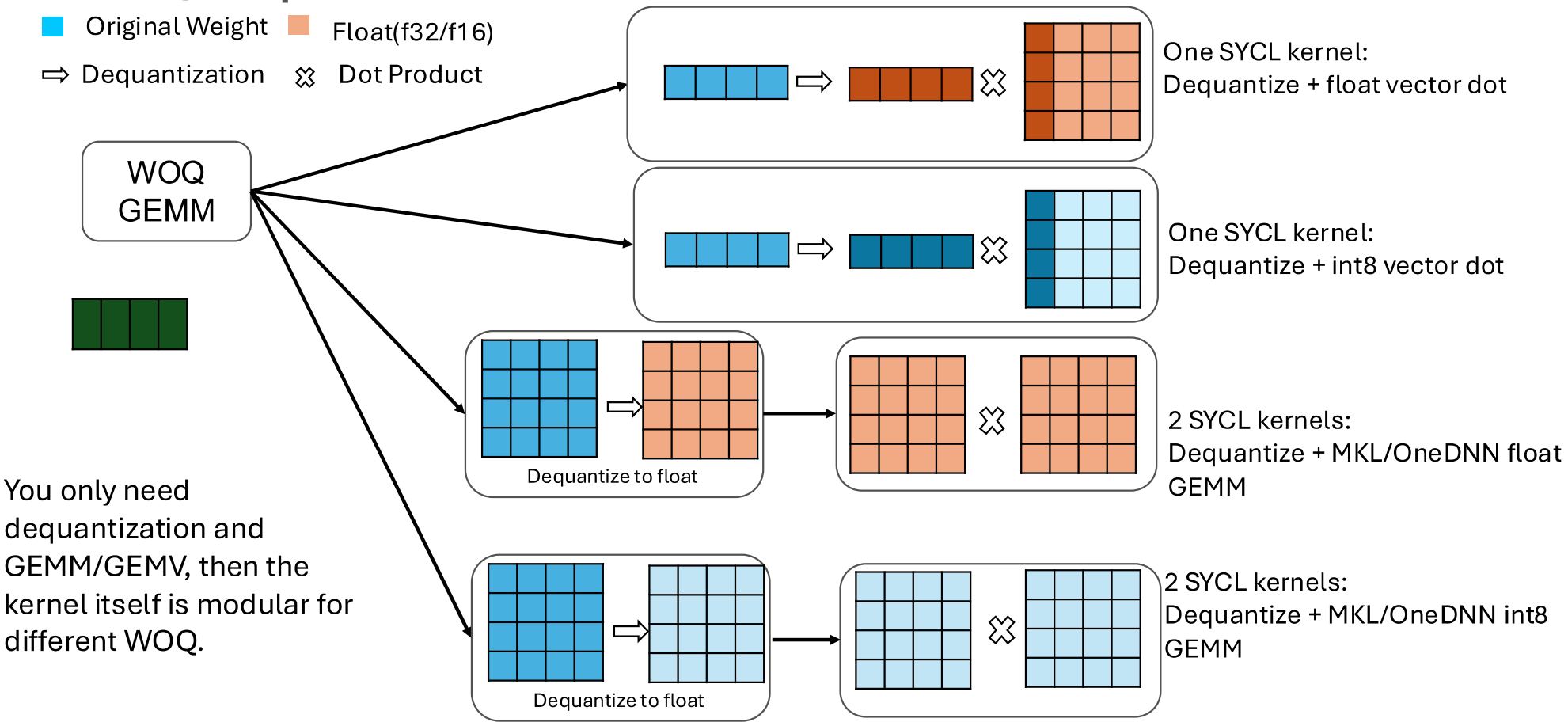
```
static ggml_backend_i ggml_backend_sycl_interface = {
    /* .get_name                = */ ggml_backend_sycl_name,
    /* .free                    = */ ggml_backend_sycl_free,
    /* .get_default_buffer_type = */
ggml_backend_sycl_get_default_buffer_type,
    /* .set_tensor_async        = */ ggml_backend_sycl_set_tensor_async,
    /* .get_tensor_async        = */ ggml_backend_sycl_get_tensor_async,
    /* .cpy_tensor_async        = */ NULL
    /* .synchronize             = */ ggml_backend_sycl_synchronize,
    /* .graph_plan_create       = */ NULL,
    /* .graph_plan_free         = */ NULL,
    /* .graph_plan_update       = */ NULL,
    /* .graph_plan_compute      = */ NULL,
    /* .graph_compute           = */ ggml_backend_sycl_graph_compute,
    /* .supports_op             = */ ggml_backend_sycl_supports_op,
    /* .supports_buft           = */ ggml_backend_sycl_supports_buft,
    /* .offload_op              = */ ggml_backend_sycl_offload_op,
    /* .event_new               = */ NULL,
    /* .event_free              = */ NULL,
    /* .event_record            = */ NULL,
    /* .event_wait              = */ NULL,
    /* .event_synchronize       = */ NULL,
};
```

Status – Cross-vendor support



Vendor	Intel	Nvidia	AMD
Basic functionality	Yes	Yes	Yes
HW List	Gen11+	Ampere+	TBD
Workloads	llama/qwen/deepseek+	llama/qwen/deepseek+	TBD
Ollama	Yes	Yes	TBD
Stable-diffusion.cpp	Yes	TBD	TBD

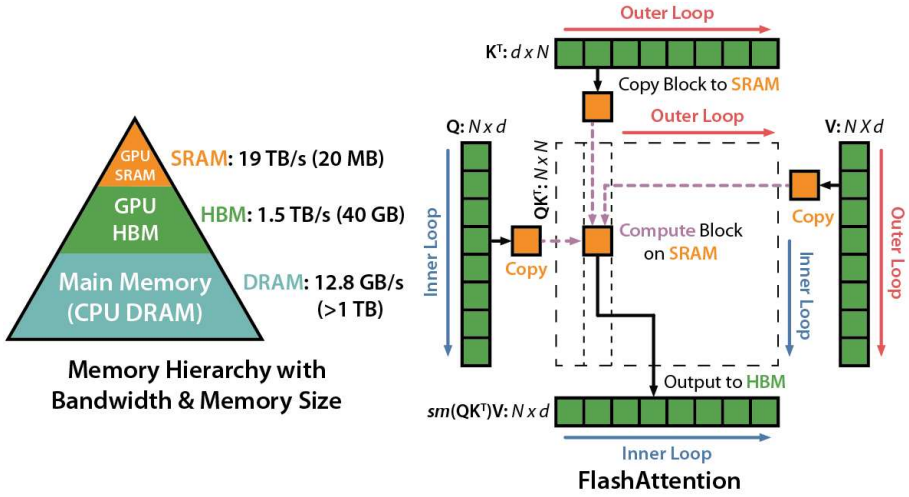
WOQ Implementation based on SYCL and Libraries



You only need dequantization and GEMM/GEMV, then the kernel itself is modular for different WOQ.

Future work – Performance optimization

Flash Attention to be implemented



GGML CUDA Flash attention is implemented in WMMA.

SYCL may provide a similar low-level API:

- There are several alternatives for SYCL: joint-matrix/assembly

Or OneDNN library may provide support for advanced LLM computation offload.

Summary

- llama.cpp is a bare metal, pure c/c++ inference framework and its main innovation: different kinds of WOQ GEMM
- we need programmable HWs and languages for these innovations, and SYCL can provide such capabilities.
- we think there are some opens of SYCL, for example, we need a low-level API for performance-wise kernels like flash attention.