

1
oneAPI
<HACK>ATHON

BUILD SOLUTIONS TO
UNLOCK THE POTENTIAL OF
HETEROGENEOUS COMPUTING

Medical Image Processing || oneAPI, SYCL

Team Name : Delta Force

Team Members : Jonnalagadda Dinesh, Patnala Rajya
Lakshmi, DR.G. Lalitha Kumari, Uppala Pavan Kumar

Problem Statement

SYCL-Accelerated Brain Tumor Segmentation and Classification from MRI Images

Architecture – Impact of oneAPI/SYCL (How oneAPI /SYCL helped you?)

The oneAPI/SYCL framework has had a significant impact on the architecture of the prototype and has greatly assisted in the development process. Here are some key ways in which oneAPI/SYCL has helped:

1. **Heterogeneous Computing:** oneAPI/SYCL allows for the efficient utilization of heterogeneous computing resources, such as CPUs and GPUs, through a single programming model. This has enabled the prototype to leverage the full power of different hardware accelerators for enhanced performance.
2. **Portable Code:** With oneAPI/SYCL, the code can be written once and run across various hardware platforms without the need for significant modifications. This portability has made it easier to deploy the prototype on different systems and take advantage of the available hardware resources.
3. **Performance Optimization:** oneAPI/SYCL provides a high-level programming abstraction that allows for the optimization of performance-critical sections of code. By utilizing parallelism and offloading computations to accelerators, the prototype can achieve better performance compared to traditional sequential approaches.
4. **Integration with Existing Libraries:** The oneAPI ecosystem includes various libraries, such as Eigen and ArrayFire, that are compatible with SYCL. These libraries provide optimized implementations of common operations, such as linear algebra and scientific computing, with built-in support for hardware acceleration. Integrating these libraries into the prototype has facilitated efficient and reliable implementation of complex algorithms.
5. **Ease of Development:** The SYCL programming model, with its C++ API, has made it easier to develop and maintain the codebase. The familiar syntax and extensive support for C++ features have simplified the implementation of complex algorithms and data processing tasks.

Overall, the oneAPI/SYCL framework has been instrumental in improving the architecture of the prototype by enabling efficient utilization of heterogeneous hardware resources, facilitating code portability, optimizing performance, integrating with existing libraries, and providing a developer-friendly programming model.

Core components of oneAPI/SYCL used in the project

The project utilized the following core components of oneAPI/SYCL:

- SYCL Programming Model: SYCL provides a high-level abstraction for heterogeneous computing with C++ code.
- SYCL Libraries: Libraries like Eigen and ArrayFire were used for linear algebra and GPU acceleration.
- oneAPI/DPC++ Compiler: The compiler translates SYCL code into optimized executables for different hardware platforms.
- SYCL Buffers and Accessors: Buffers and accessors manage data movement between the host and device.
- SYCL Queue: The queue manages the execution of SYCL kernels on the target device.

These components enabled efficient heterogeneous computing, optimized performance, and simplified development.

Demo Video/Live Demo

Please elaborate oneAPI/SYCL usage

<https://drive.google.com/file/d/1hNPmwFQ8mVPAppW1Dh7eYrKol5eaqGbQ/view?usp=sharing>

GitHub Link (Codes should be public and available after
hackathon also)

<https://github.com/Dineshjnlld/intel-oneAPI>

Results Summary (focus on unique aspects of oneAPI/SYCL that you have used)

The utilization of oneAPI/SYCL in the project provided several unique aspects and benefits:

- Heterogeneous Computing: oneAPI/SYCL allowed for seamless programming of heterogeneous systems, leveraging the power of CPUs, GPUs, and other accelerators in a unified manner.
- Portable and Cross-platform: The use of oneAPI/SYCL ensured portability across different hardware platforms, enabling the code to be executed on a wide range of devices.
- Performance Optimization: With SYCL, we could take advantage of hardware-specific optimizations, such as GPU acceleration through libraries like Eigen and ArrayFire, to enhance the performance of our application.
- Simplified Development: SYCL's C++ programming model and libraries provided a high-level abstraction, making it easier to express parallelism and exploit the capabilities of the underlying hardware without diving into low-level details.
- Memory Management: SYCL's buffers and accessors facilitated efficient data movement between the host and device, ensuring optimal memory management and minimizing data transfer overhead.

Overall, oneAPI/SYCL offered a unique combination of portability, performance, and ease of development, enabling us to harness the power of heterogeneous systems effectively in our project.

1
oneAPI
<HACK>ATHON

THANK YOU