intel ai summit
英特爾AI科技論壇

Bringing AI Everywhere

Intel® Developer Cloud 擁抱oneAPI及Intel AI開發工具：輕鬆升級你的AI代碼

Joel Lin

March 27th, 2024

intel Ai summit

Intel® Developer Cloud

Portfolio for customers

Register at http://cloud.intel.com

HW services:

- New platform evaluation & software bring up and porting (e.g. accelerators on 4th and 5th Gen Intel® Xeon® processors: AMX, DSA, IAA, QAT etc)
- Enterprise AI benchmark evaluation on Intel® Gaudi® 2 AI accelerators
- HPC benchmark testing on Intel® Data Center GPUs Max Series
- *Ecosystem partners (e.g., OEMs, CSPs, ISVs) enabling*
- *Large enterprise customers (data center)*

AI infrastructure services:

- LLM model training and optimization
- AI model training and deployment for inferencing
- AI model deployment via CLI/SSH automation
- AI container deployment via k8s APIs
- Hosting platform for deploying AIaaS
- *AI disruptors (startups)*
- *Established AI-savvy enterprises*

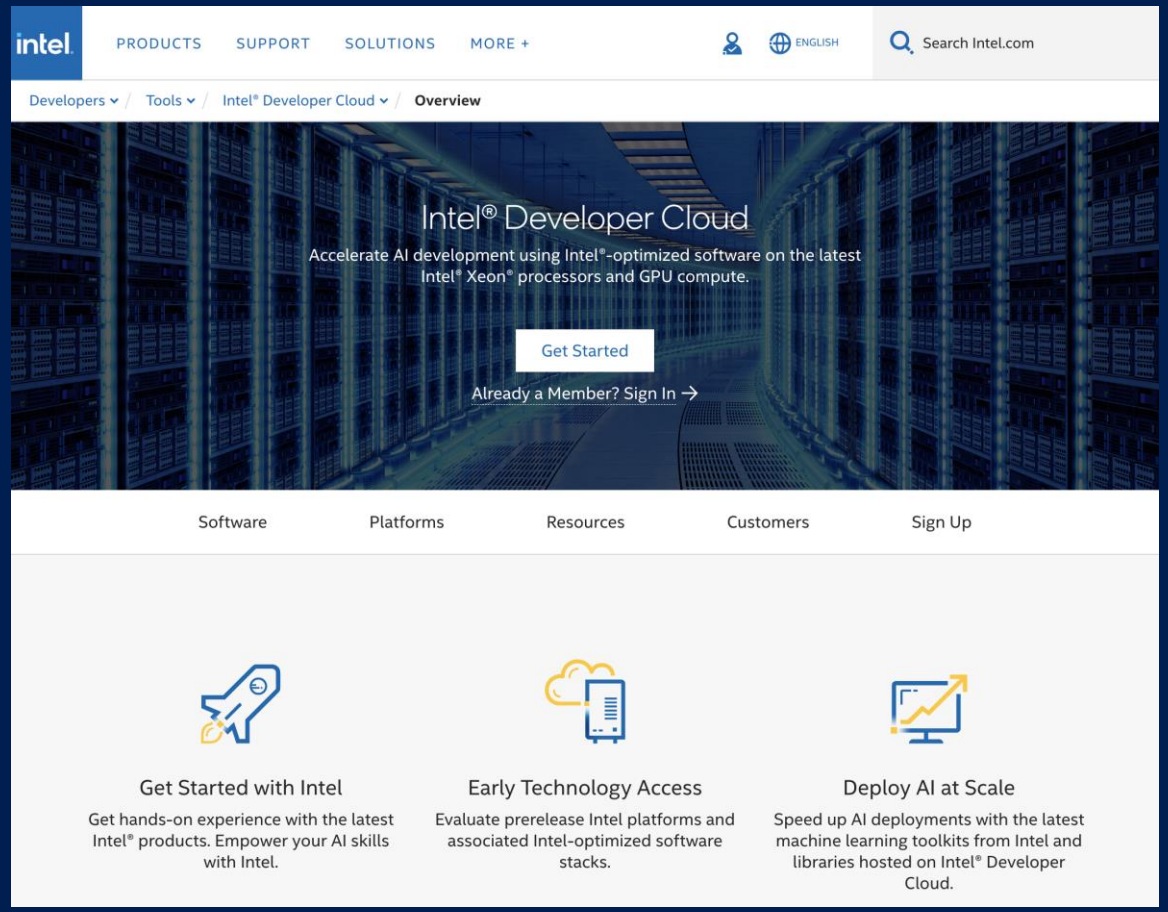
intel Ai summit

Intel® Developer Cloud

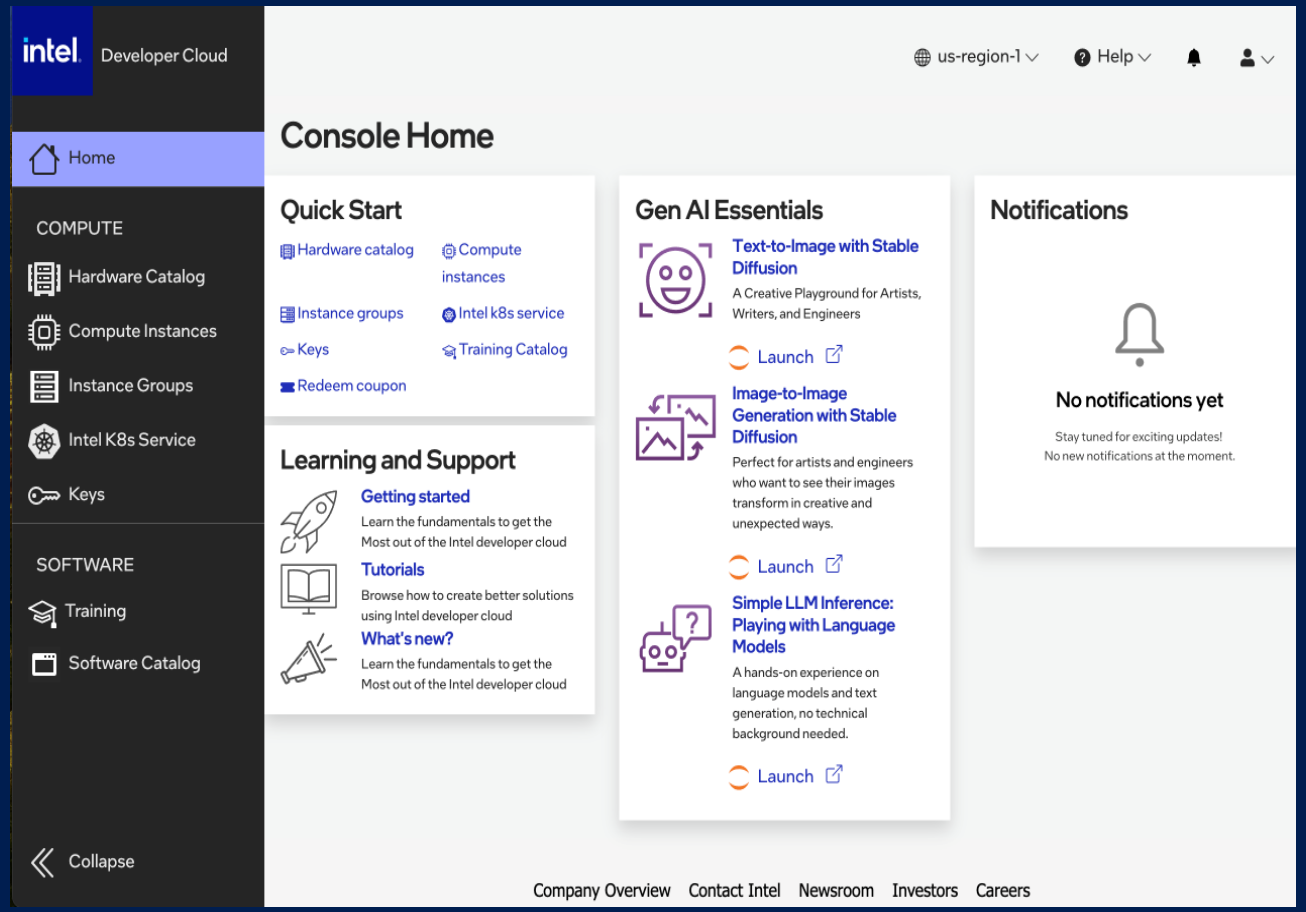
How it works - access

Video: Get Started with Intel® Developer Cloud | Intel Software

Register your account at http://cloud.intel.com



Access resources at http://console.cloud.intel.com



intel Ai summit

Intel Software Developer Tools

Flexible, Comprehensive, Open Software Stack – Powered by oneAPI

oneAPI POWERED

intel HPC TOOLKIT 1 oneAPI

- Modeling & Simulation
- Scientific Computing
- Analytics

intel AI TOOLS 1 oneAPI

Intel-Optimized AI Software Tools and Frameworks

- Data Analytics at Scale: MODIN, pandas, NumPy, SciPy
- DL Inference and Training: TensorFlow, PyTorch, OpenVINO, Intel® Neural Compressor
- Classical ML: scikit-learn, dmlc XGBoost, python

intel RENDERING TOOLKIT 1 oneAPI

- High-fidelity Graphics
- Visual Compute
- Ray Tracing

intel 1 oneAPI BASE TOOLKIT

- **Tools:** Intel® DPC++/C++ Compatibility Tool; Intel® VTune™ Profiler; Intel® Advisor; Intel® Distribution for GDB; Intel® Distribution for Python
- **Performance Libraries:** oneMKL; oneDNN; oneDAL; oneCCL; oneTBB; oneDPL; Intel® MPI
- **Direct Programming:** C++ with SYCL; C++; Fortran; OpenMP
- **Compilers:** Intel Fortran Compiler; Intel C++/DPC++ Compiler
- **Hardware Interface – oneAPI Level Zero**
- CPU GPU FPGA

Download at intel.com/oneAPI or run tools on the Intel® Developer Cloud at cloud.intel.com

intel AI summit

*Other names and brands may be claimed as the property of others. SYCL is a trademark of the Khronos Group Inc.

Data Parallel C++:

oneAPI's implementation of SYCL

https://github.com/intel/llvm/tree/sycl/sycl

DPC++ = ISO C++ and Khronos SYCL and community extensions

Freedom of Choice: Future-Ready Programming Model

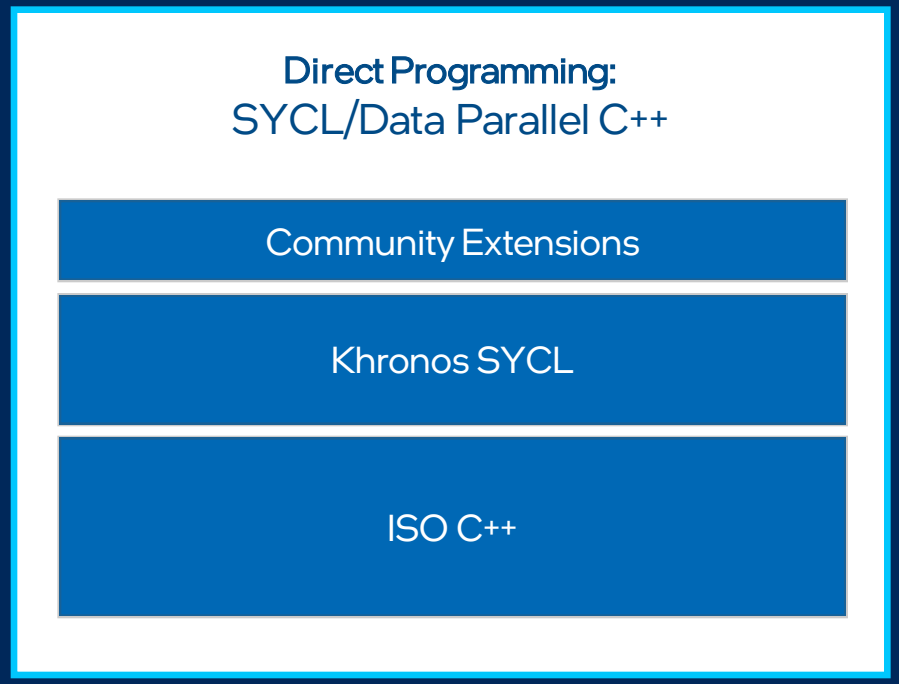
- Allows code reuse across hardware targets
- Permits custom tuning for a specific accelerator
- Open, cross-industry alternative to proprietary language

DPC++ = ISO C++ and Khronos SYCL and community extensions

- Designed for data parallel programming productivity
- Provides full native high-level language performance on par with standard C++ and broad compatibility
- Adds SYCL from the Khronos Group for data parallelism and heterogeneous programming

Community Project Drives Language Enhancements

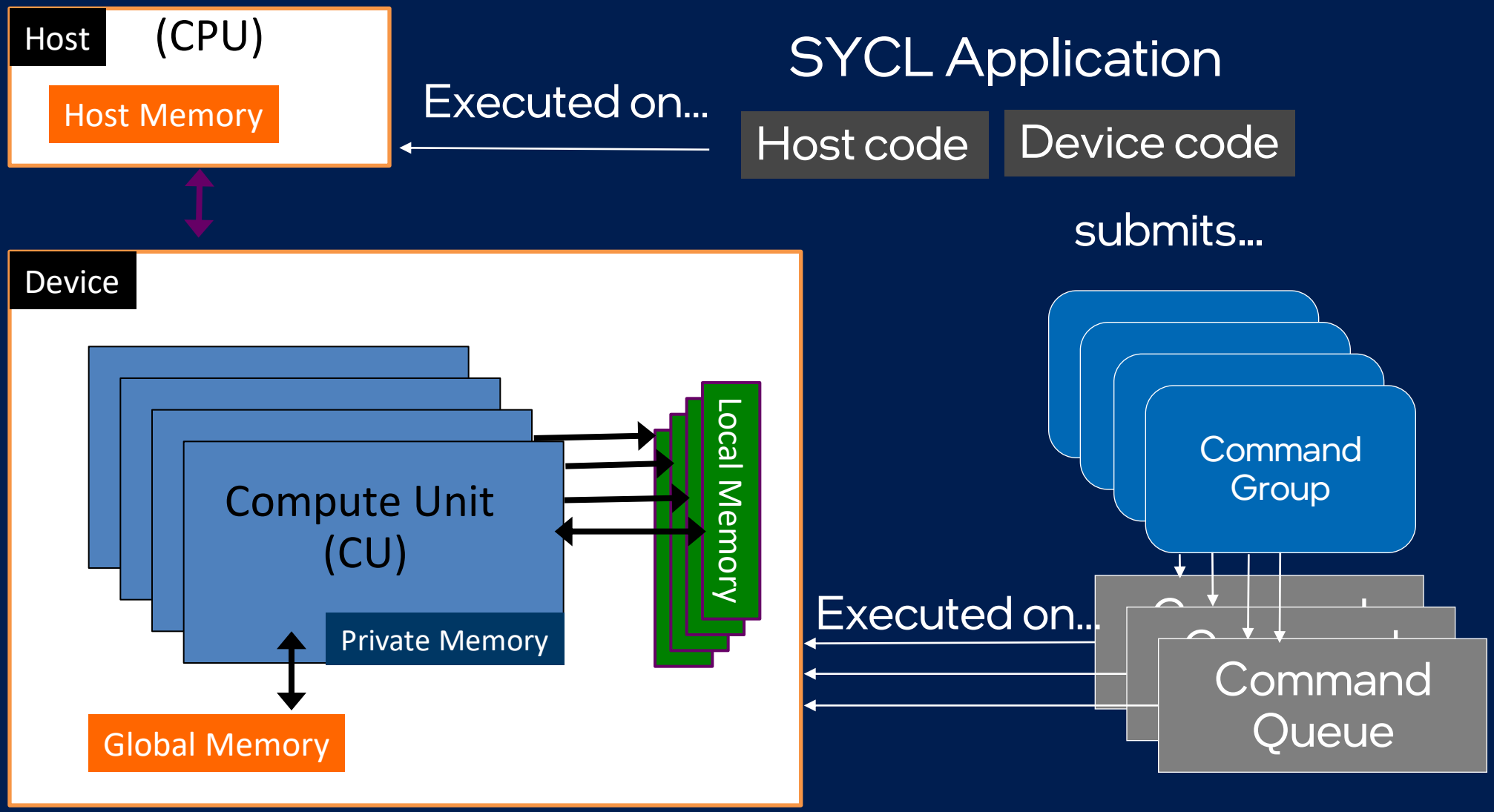
- Provides extensions to simplify data parallel programming
- Continues evolution through open and cooperative development
  - Ask questions in SYCL Forums https://community.khronos.org/c/sycl
  - Open issues for SYCL Specification in https://github.com/KhronosGroup/SYCL-Docs



Check the link to understand: SYCL™ 2020 Specification (revision 8) (khronos.org)
PDF version: Khronos SYCL Registry - The Khronos Group Inc

The open source and Intel DPC++/C++ compiler supports Intel CPUs, GPUs, and FPGAs.
Codeplay announced a DPC++ compiler that targets Nvidia GPUs.

SYCL Basics: Simplifying Heterogeneous Programming



SYCL Code in Action - A Glimpse Under the Hood

```
#include <sycl/sycl.hpp>
using namespace sycl;

int main() {
std::vector<float> A(1024), B(1024), C(1024);
// some data initialization
    {
      buffer bufA {A}, bufB {B}, bufC {C};
      queue q;
      q.submit([&](handler &h) {
          auto A = bufA.get_access(h, read_only);
          auto B = bufB.get_access(h, read_only);
          auto C = bufC.get_access(h, write_only);
          h.parallel_for(1024, [=](auto i){
              C[i] = A[i] + B[i];
          });
      });
    }
for (int i = 0; i < 1024; i++)
       std::cout << "C[" << i << "] = " << C[i] << std::endl;

}
```

Host code

Accelerator device code

Host code

Check Intel Developer Cloud – essential SYCL tutorials to know more

oneAPI

Specification and Open Source

Open industry initiative driving a vendor-neutral software ecosystem for multiarchitecture accelerated computing.

Now governed by the Linux Foundation.

UXL Unified Acceleration Foundation

Founding Members: ARM, Fujitsu, Google Cloud, Imagination Tech, Intel, Qualcomm, Samsung, VMware

Freedom to Make Your Best Choice

- An open alternative to single-vendor/proprietary lock-in enables easy architecture retargeting
- Open, standards-based programming (C++ with SYCL) so software investments continue to add value in future hardware generations

Performance – Realize All the Hardware Value

- Expose and exploit all the cutting-edge features and maximize performance across CPUs, GPUs, FPGAs, and other accelerators.
- Powerful libraries for acceleration of domain-specific functions

Productivity – Develop Performant Code Quickly

- One programming model for all – easy integration with existing code including migration of CUDA code to SYCL
- Based on familiar C++ – no need to learn a new language
- Interoperable with existing HPC standards including Fortran, C/C++, OpenMP, and MPI, as well as Python with a rich set of optimized Python libraries

Visit oneapi.io or https://uxlfoundation.org/ for more details

Middleware and Frameworks

PyTorch · TensorFlow · scikit-learn · OpenVINO · NumPy · XGBoost

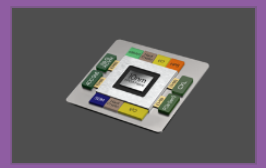
oneAPI Industry Specification

Direct Programming	API-Based Programming		
SYCL (C++)	Math oneMKL	Threading oneTBB	Parallel STL oneDPL
	Analytics/ ML oneDAL	DNN oneDNN	ML Comm oneCCL

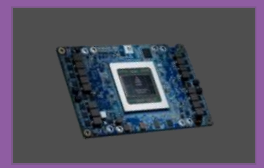
Low-Level Hardware Interface (oneAPI Level Zero)

CPU

GPU



FPGA



Other Accelerators

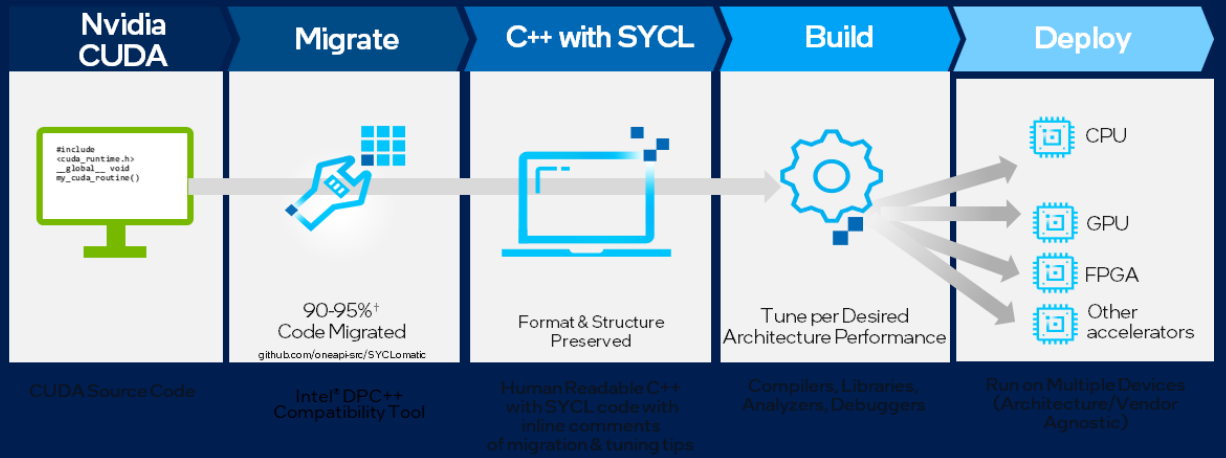
intel AI summit

Migrate from CUDA* to C++ with SYCL*

Stop writing and maintaining different codebases for different architectures

CUDA to SYCL Migration Portal

- Choose your accelerated computing hardware and reuse code with performance portability
- Single C++ with SYCL codebase can run on accelerators with multiple architectures from multiple vendors
- Intel ® DPC++ Compatibility Tool & Open Source SYCLomatic automatically migrates ~90-95%* of a typical CUDA app to SYCL
- Generates helpful comments to guide you to finish migration and tune performance
- Visit the CUDA to SYCL Migration Portal for tutorials, best practices, code samples, apps catalog, and community support



DPC++ Compatibility tool repository - https://github.com/oneapi-src/SYCLomatic

Migration Success Examples

Contrel

Using Intel® DPC++ Compatibility Tool, we successfully migrated our automatic inspection solution to SYCL*, which helps us to remove code barriers with a single, open, standards-based programming model for heterogeneous computing…

More in Ecosystem Support for Intel® oneAPI

intel AI summit

[1]Intel estimates as of March 2023. Based on measurements on a set of 85 HPC benchmarks and samples, with examples like Rodinia, SHOC, PENNANT. Results may vary.

oneAPI Plug-ins for Nvidia* & AMD*

Codeplay Support for Nvidia & AMD GPUs to Intel® oneAPI Base Toolkit

oneAPI for NVIDIA & AMD GPUs

- Free download of binary plugins to Intel® oneAPI DPC++/C++ Compiler:
- Nvidia GPU
- AMD beta GPU
- No need to build from source!
- Plug-ins updated quarterly in-sync with SYCL 2020 conformance & performance

Priority Support

- Available through Intel, Codeplay & our channel
- Requires Intel Priority Support for Intel oneAPI DPC++/C++ Compiler
- Intel takes first call, Codeplay delivers backend support
- Codeplay provides access to older plug-in versions

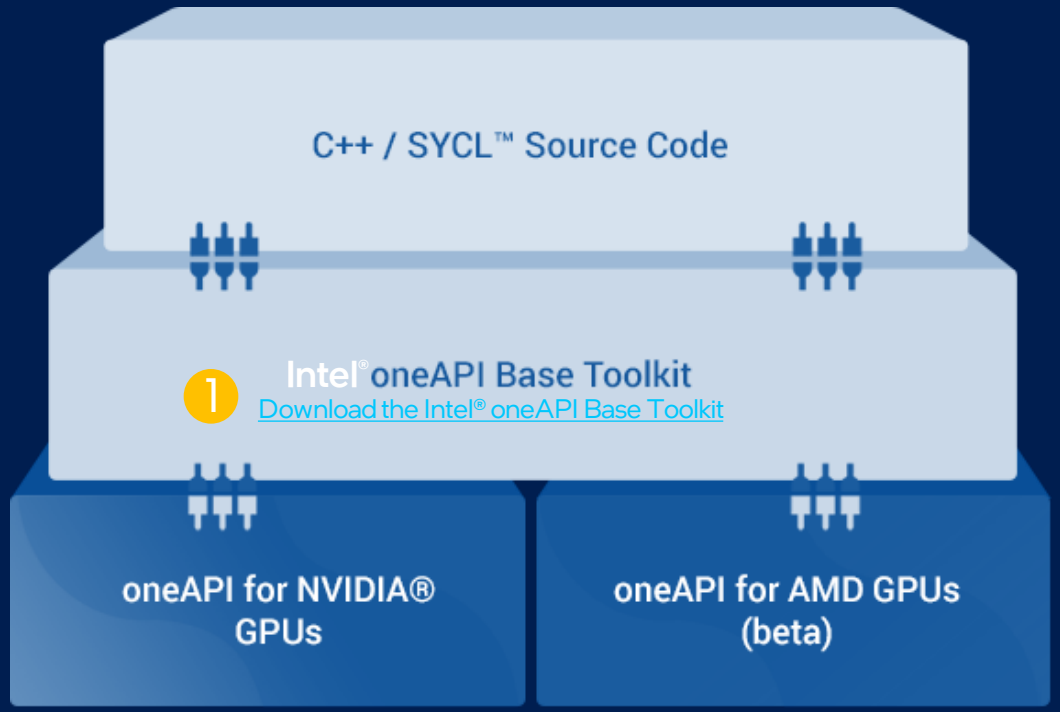


Image courtesy of Codeplay Software Ltd.

2 Download Nvidia GPU plug-in

2 Download AMD GPU plug-in

Codeplay blog

Codeplay press release

3 compiler command: icpx –fsycl –fsycl-targets={backend} sample.dp.cpp

{backend}: use different backend options to specify either Intel CPU/Intel GPU/Nvidia/AMD hardware targets

Check the backend target options from oneAPI DPC++ Compiler online users manual

intel Ai summit

SYCL migration (demo)

Single file migration example

NVCC build: "nvcc <prjfolder>/cuda/sample.cu –I<path>/include -DBUILD_CUDA"

DPCT usage: dpct --out-root=/path/to/output *sample.cu* --extra-arg="-I./include" --extra-arg="-DBUILD_CUDA"

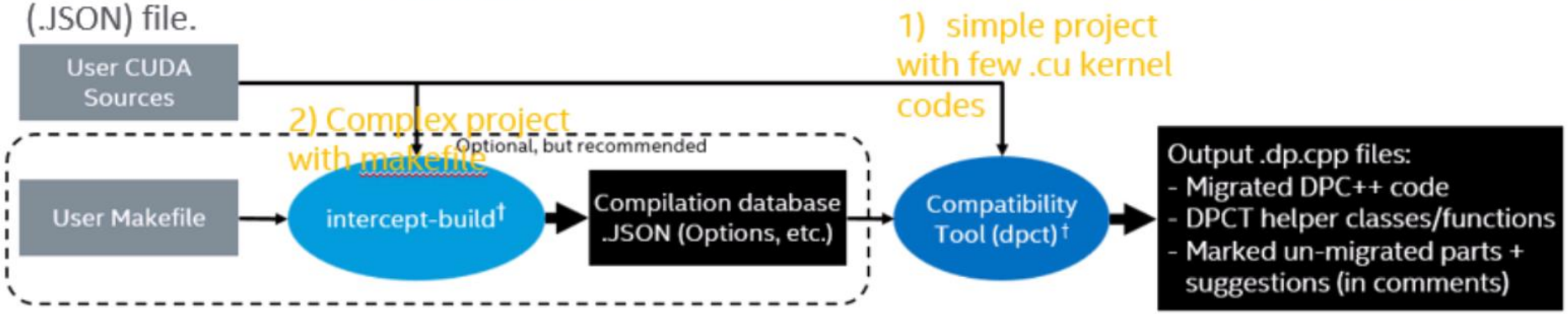
* Compile SYCL codes with Intel DPC++/C++ compiler: icpx –fsycl sample.dp.cpp

Multiple files migration example (ex. makefile project)

- DPCT usages:
  - cd cuda_project folder
  - make clean
  - **intercept-build** make // this will generate compile_commands.json file
  - dpct –p compile_commands.json --gen-build-script
    - generate a Makefile // need to manually check and modify toolchains names/parameters

Migration Workflow overview

- Preparation: make sure your CUDA project can be built with nvcc.
  - DPCT will use CUDA header files. Support multiple CUDA SDK versions.
- DPC++ Compatibility Tool (DPCT)
  - Take .cu source files as input and generate the migrated .cpp files. Simple project for example , migrate a single kernel code.
- Intercept-build
  - For complex projects use makefile, use intercept-build command to create a compilation database (.JSON) file.



† Certain CUDA language header files may need to be accessible to the Intel® DPC++ Compatibility Tool

User Guide Migration Rule

- Commands: dpct sample.cu --rule-file=rule_file1.YAML --rule-file=rule_file2.YAML
- YAML Rule files

```
Rule: rule_cudaMalloc
Kind: API
Priority: Takeover
In: cudaMalloc
Out: $type_name_of($2)"aaa = foo($deref($1), ($deref_type($1), $queue, $context, $device)
Includes: ["ccc.h"]
```

- CUDA source

```
// CUDA
Int* ptr;
cudaMalloc(&ptr, 50);
```

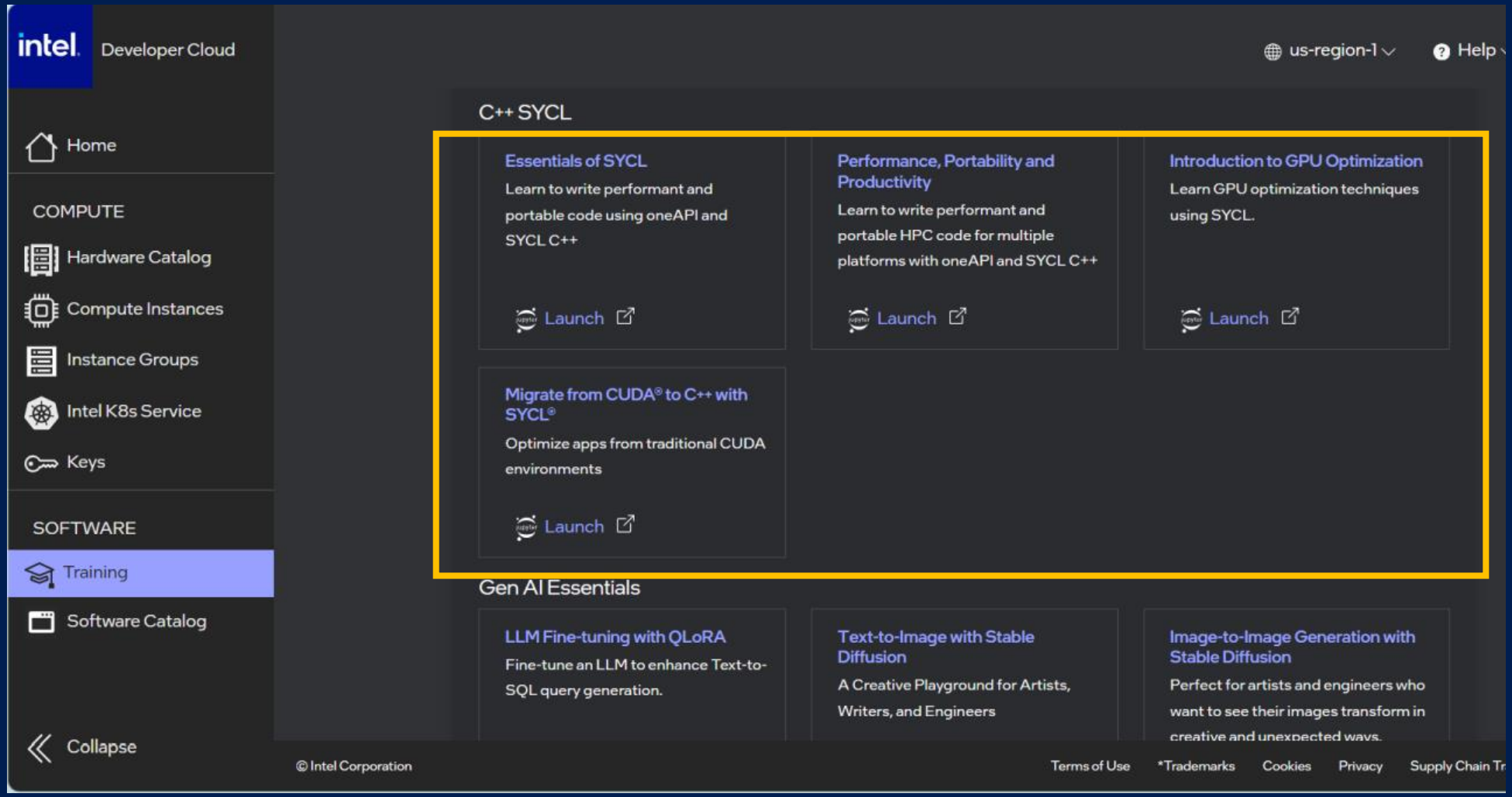
- Migrated SYCL

```
// Migrated SYCL
#include"ccc.h"
size_t *aaa = foo(ptr, (int *)&(&ptr), dpct:get_out_of_order_queue(),
                                        dpct::get_default_context(), dpct::get_current_device());
```

intel Ai summit

https://oneapi-src.github.io/SYCLomatic/dev_guide/migrate-a-project/user-defined-migration-rules.html

C++ SYCL Tutorials on Intel Developer Cloud



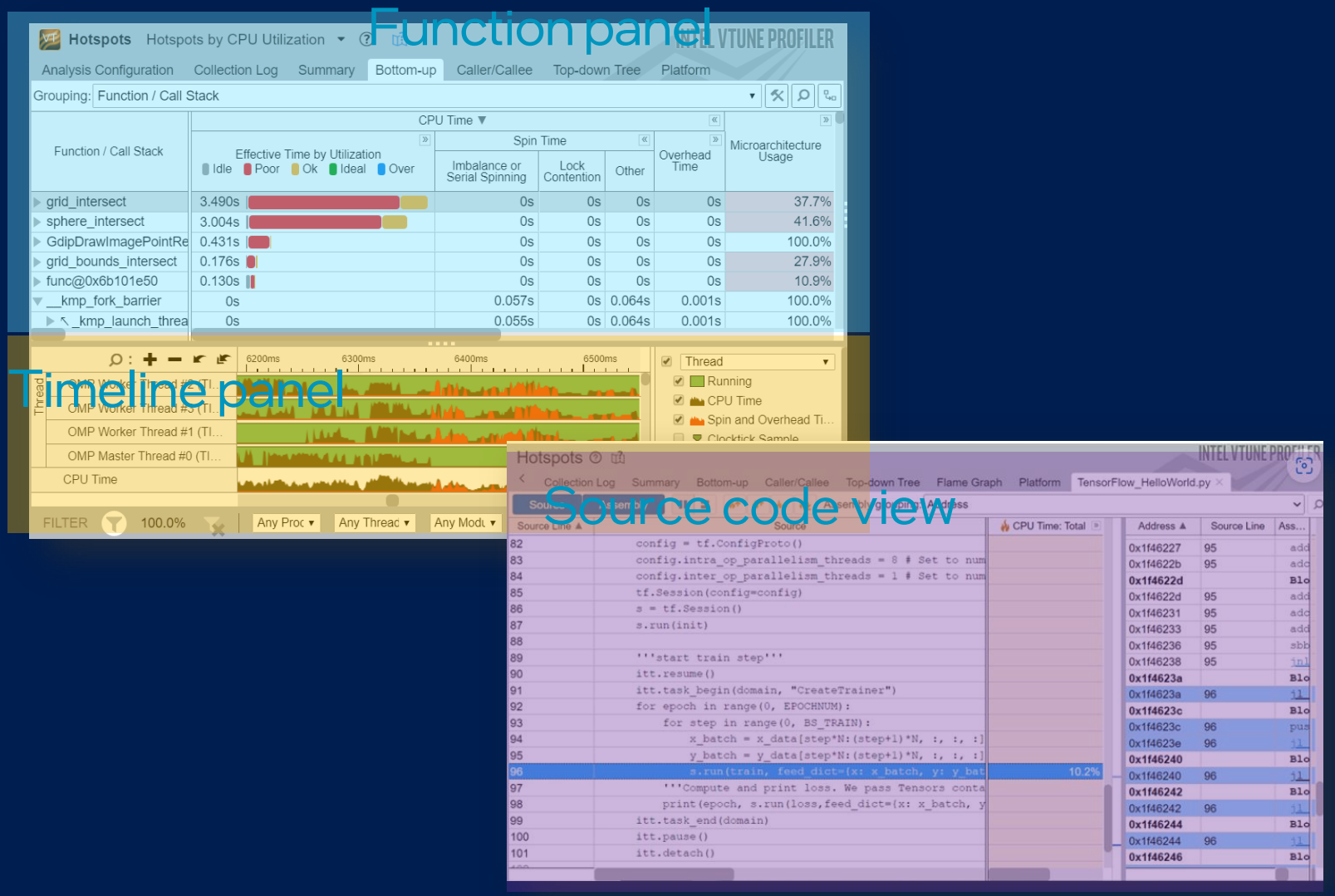
intel Ai summit

[Check the link to find more details :Intel® oneAPI DPC++/C++ Compiler Developer Guide and Reference]

Unveiling performance bottleneck - Intel® VTune™ Profiler

system/application performance profiler

- Data Collection
  - multiple hardware performance metrics
  - Hardware PMU, perf, ftrace, custom data collector.
- Data in groupings options
  - By functions, processes, module, threads, cores
- Data in Timeline
  - performance metrics distribution
- Flexible workflow –
  - GUI or command line
  - Remote collection
- Multi-purpose focus analysis types
  - Python, GPU, memory bandwidth, IO and etc.



intel AI summit

VTune Command Line: Power and Flexibility (Demo)

- vtune -collect gpu-offload -no-summary -r ./result_gpu-offload – {app}

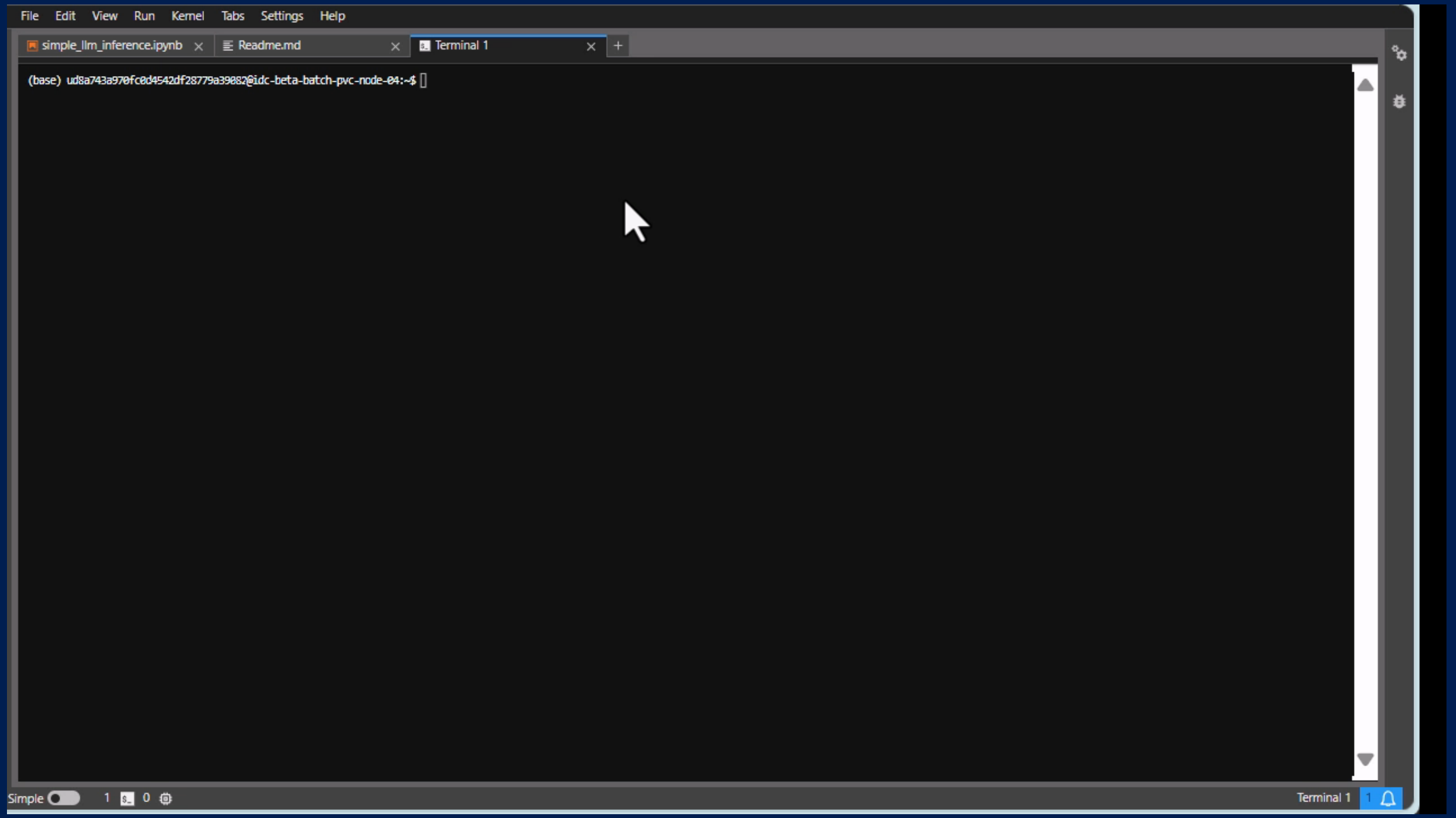
```
root@nntpat99-39:/home/vtune/matrix_multiply_vtune# vtune -report summary -r ./result_gpu-offload
vtune: Using result path `/home/vtune/matrix_multiply_vtune/result_gpu-offload'
vtune: Executing actions 75 % Generating a report                          Elapsed Time: 11.235s
    GPU Utilization: 10.1%
      | GPU utilization is low. Consider offloading more work to the GPU to
      | increase overall application performance.
      |

        GPU Utilization
        GPU Engine        Packet Type  GPU Time     GPU Utilization(%)
        ----------------  -----------  ---------  --------------------
        Render and GPGPU  Unknown        1.134s                10.1%

Hottest GPU Computing Tasks
Computing Task             Total Time  Execution  % of Total Time(%)  Instance Count
-------------------------  ----------  ---------  ------------------  --------------
Matrix1_1<float>              1.146s     1.130s               98.6%               1
zeCommandListAppendBarrier    0.000s         0s                0.0%               0
Collection and Platform Info
    Application Command Line: ./matrix.dpcpp
    Operating System: 5.4.30 DISTRIB_ID=Ubuntu DISTRIB_RELEASE=20.04 DISTRIB_CODENAME=focal DISTRIB_DESCRIPTION="Ubuntu 20.04 LTS"
    Computer Name: nntpat99-39
    Result Size: 98,0 MB
    Collection start time: 12:44:24 20/02/2021 UTC
    Collection stop time: 12:44:36 20/02/2021 UTC
    Collector Type: Event-based sampling driver,Driverless Perf system-wide sampling,User-mode sampling and tracing
    CPU
        Name: Intel(R) Processor code named Skylake
        Frequency: 2.592 GHz
        Logical CPU Count: 8
        Max DRAM Single-Package Bandwidth: 19.000 GB/s
    GPU
        Name: Iris Pro Graphics 580
        Vendor: Intel Corporation
        EU Count: 72
        Max EU Thread Count: 7
        Max Core Frequency: 950.000 MHz
        GPU OpenCL Info
            Version
            Max Compute Units: 72
            Max Work Group Size: 256
            Local Memory: 65,5 KB
            SVM Capabilities

Recommendations:
    GPU Utilization: 10.1%
      | GPU utilization is low. Switch to the for in-depth analysis of host
      | activity. Poor GPU utilization can prevent the application from
      | offloading effectively.
    EU Array Stalled/Idle: 69.8% of Elapsed time with GPU busy
      | GPU metrics detect some kernel issues. Use GPU Compute/Media Hotspots
      | (preview) to understand how well your application runs on the specified
      | hardware.
```

- vtune -report hotspots -group-by=source-computing-task -column="Total Time,Average Time,Instance Count" -sort-desc="Total Time" -r ./result_gpu-programming-api/ -q

```
mpetrova@dtc-nuc-031:/localdisk/mpetrova/matrix_multiply_vtune$ vtune -report hotspots -group-by=source-computing-task -column=
"Total Time,Average Time,Instance Count" -sort-desc="Total Time" -r ./result_gpu-programming-api/ -q
Column filter is ON.
Source Computing Task (GPU)          Computing Task:Total Time  Computing Task:Average Time  Computing Task:Instance Count
-----------------------------------  -------------------------  ---------------------------  -----------------------------
Matrix1<float>                                          0.117s                       0.117s                              1
zeCommandListAppendMemoryCopyRegion                     0.001s                       0.001s                              1
zeCommandListAppendBarrier                              0.000s                       0.000s                              1
```

intel Ai summit

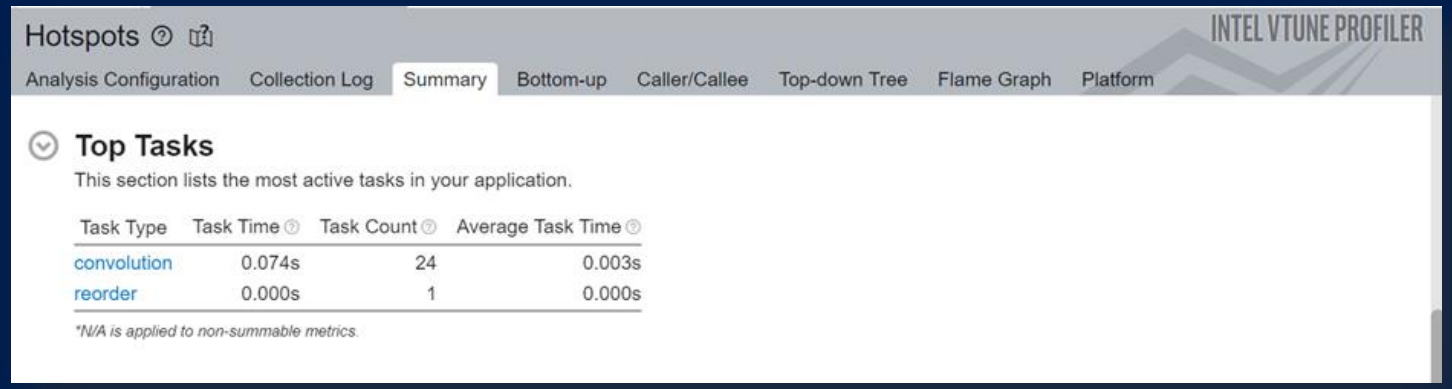
* Using the Command-Line Interface to Analyze the Performance of a... (intel.com)



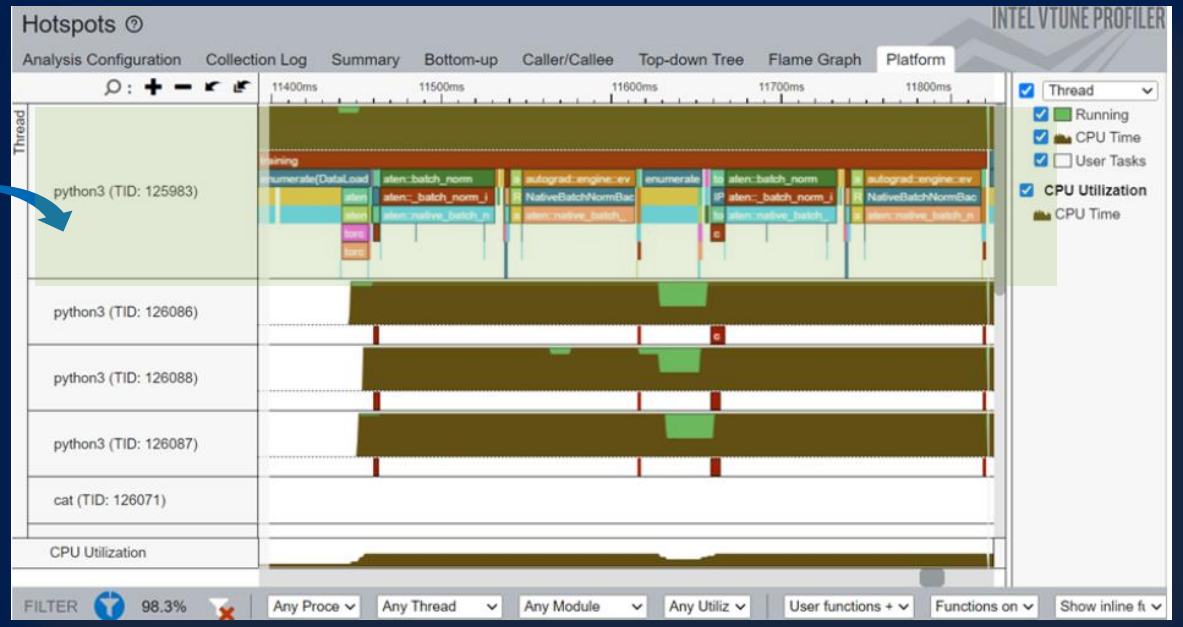
intel AI summit

Profiling Machine Learning Applications (Demo)

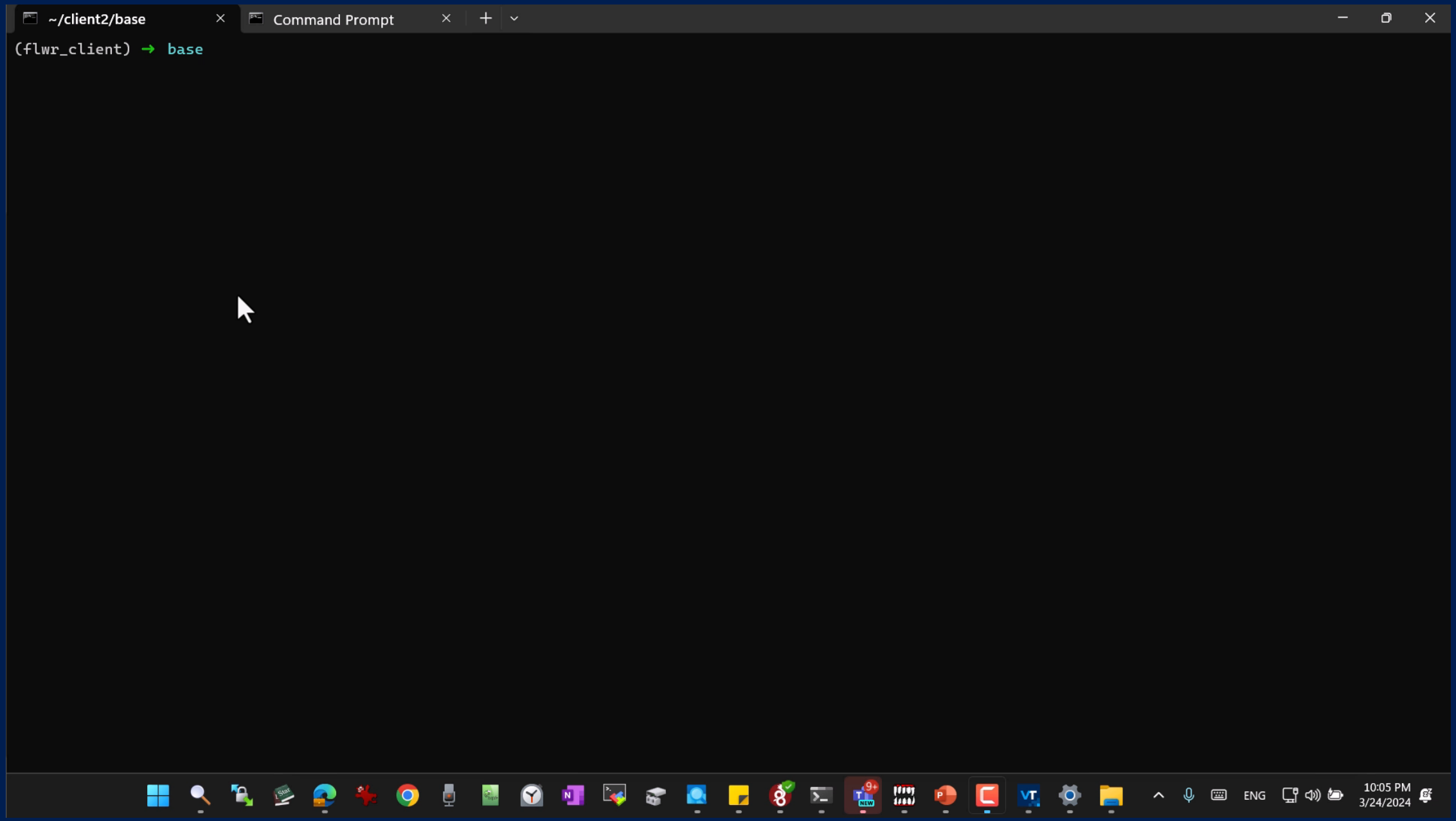
- Enabled ITT instrument APIs in up-streamed oneDNN(dnnl) library, Pytorch.
- export ONEDNN_ENABLE_JIT_PROFILING=ON
- Use python ITT APIs inside Pytorch framework




```
with torch.autograd.profiler.emit_itt():
        torch.profiler.itt.range_push('training')
        model.train()
        for batch_index, (data, y_ans) in enumerate(trainLoader):
            data = data.to(memory_format=torch.channels_last)
            optim.zero_grad()
            y = model(data)
            loss = crite(y, y_ans)
            loss.backward()
            optim.step()
```



Check the online documentation for details: Profiling Machine Learning Applications (NEW) (intel.com)



intel Ai summit

Resources for VTune Profiler

Documentation

- Installation Guide (All Operating Systems)
- User Guide
- Processor Tuning Guides
- Release Notes
- System Requirements

Training

Basics

Boost CPU Performance [2:00]

Seven Steps to GPU Application Performance

Analyze Common Performance Bottlenecks: Linux* | Windows*

Profile Heterogeneous Computing Performance [25:33]

Code Samples

Get Started with Profiling

Matrix Multiply for Heterogeneous Applications
Learn how to profile a code that's compliant with SYCL for CPU and GPU using Intel VTune Profiler. The sample contains three implementations of matrix multiplication using different SYCL features.

Matrix Multiply for C Code Running on a CPU

Learn how to use Intel VTune Profiler to profile C code running on a GPU. Six different implementations with various levels of CPU optimizations are included.

Configuration

Profile without Drivers

Profile Docker* Containers

Use Intel VTune Profiler Server with Microsoft Visual Studio* Code and Intel® Developer Cloud

GPU Profiling Tutorials

Profile an OpenMP Offload Application That Runs on a GPU

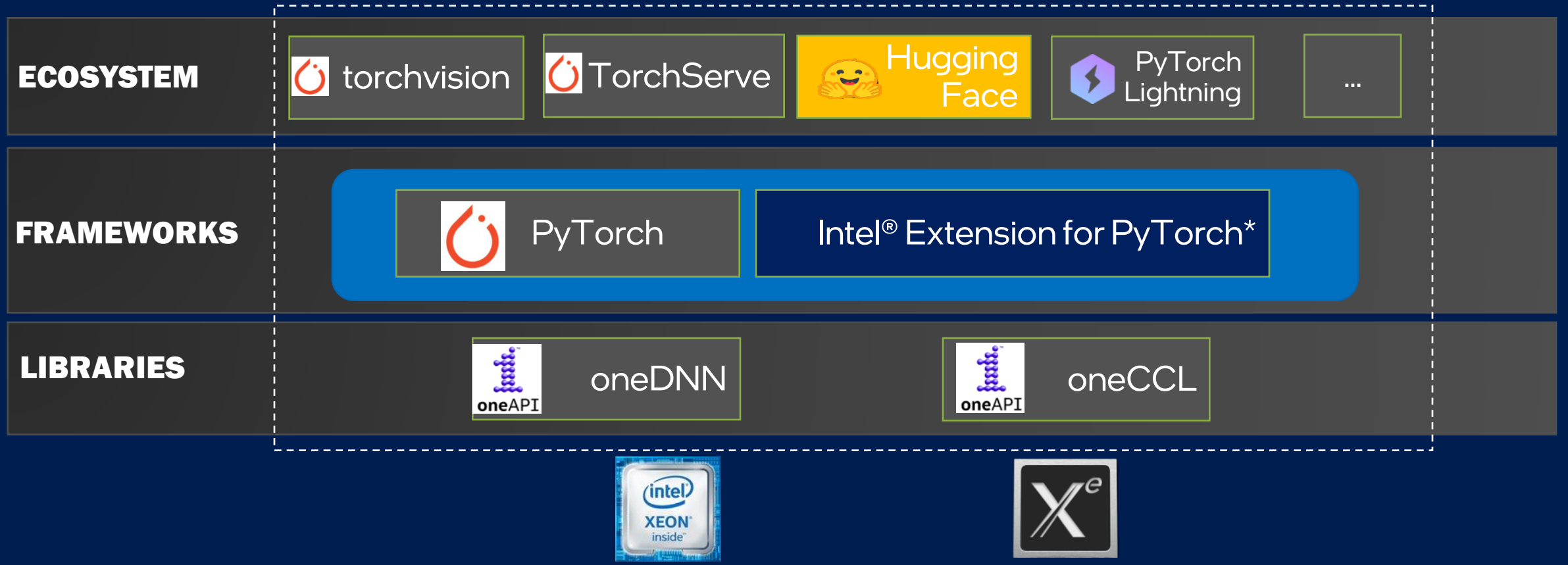
Build and compile an OpenMP application offloaded onto an Intel GPU. Use Intel VTune Profiler to run analyses with GPU capabilities (HPC performance characterization, GPU offload, and GPU compute and media hot spots) on the OpenMP application, and then examine the results.

Profile a SYCL* Application Running on a GPU

Learn how to use Intel VTune Profiler to run a GPU analysis on the SYCL application and examine the results.

Powering AI Innovation: Intel® Extension for PyTorch

Intel Optimized AI Software tools



intel AI summit

Check this official documentation link for further details - Welcome to Intel® Extension for PyTorch* Documentation!
Installation, code examples, optimization details

Get Started: oneAPI-Powered AI Reference Kits

Focusing on tackling deployment challenges with most popular AI use cases

SCANME

Finance & Insurance

- Claim Document Automation
- Fraud detection in credit card transactions
- Default Risk Prediction
- Disaster appraisal process

Process Automation

- Visual Process Discovery
- Intelligent Document Indexing
- Invoice-to-Cash Automation
- Historical Assets Document Processing

Manufacturing & Utilities

- Drone Navigation Segmentation
- Power Line Fault Detection
- Predictive Asset Analytics
- Visual Quality Inspection
- Digital Twin
- Engineering Design
- Traffic Camera object detection
- Aerodynamics / Fluid Flow Profiling

Health & Life Sciences

- Medical Imaging Diagnosis
- Disease Prediction
- AI Transcribe for Therapists

Customer Care

- Purchase Prediction
- Customer Segmentation
- Customer Churn Prediction
- Customer Care Chatbot
- Demand Forecasting
- Product Recommendations
- Order to delivery Forecasting

Tech & Security

- Vertical Search Engine
- Network Intrusion Detection
- Data Protection
- IoT (Data Streaming Anomality Detection)

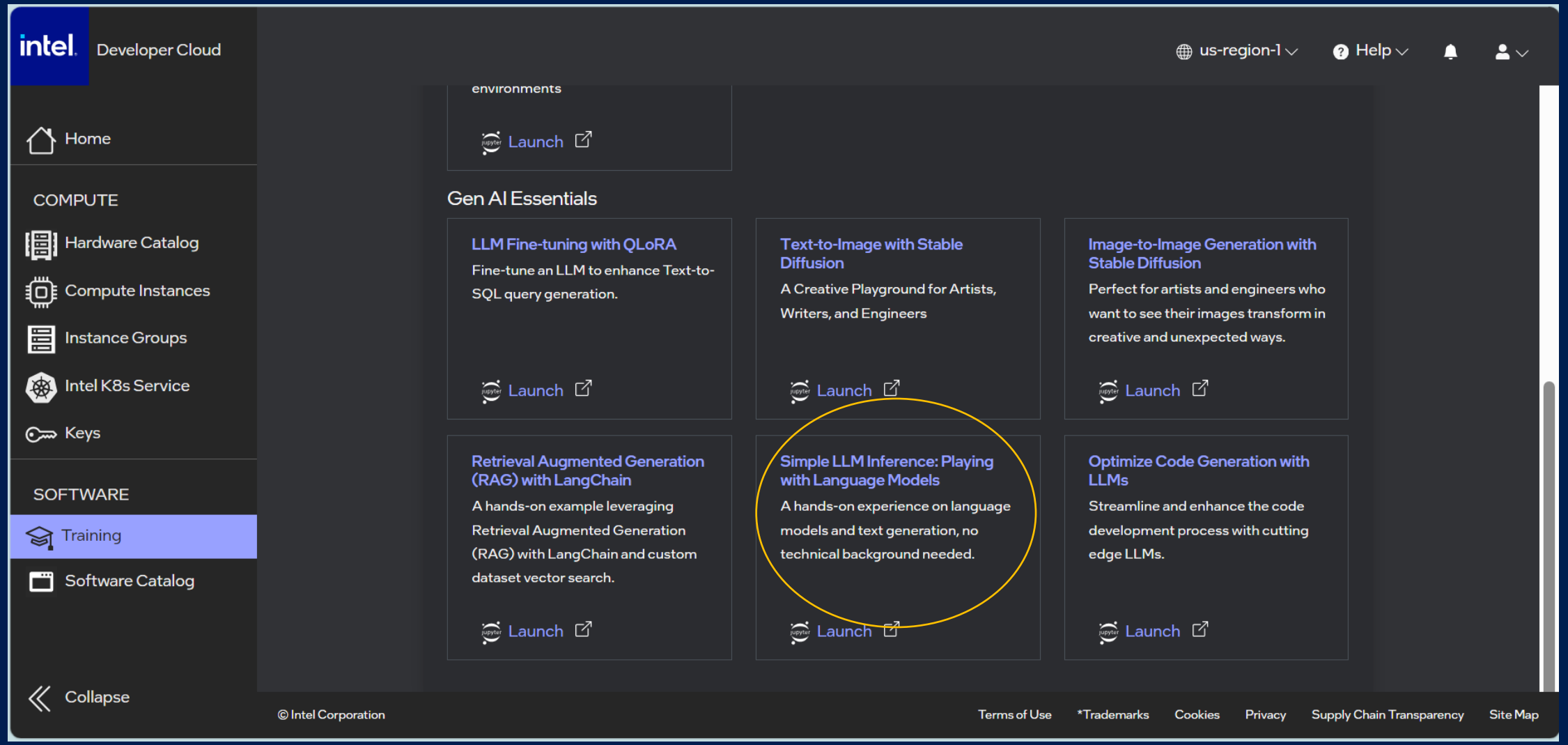
Synthetic Data

- AI Synthetic Data (Structured)
- AI Synthetic Data (Unstructured – Text)
- AI Synthetic Data (Unstructured – Image)
- AI Synthetic Data (Unstructured – Voice)

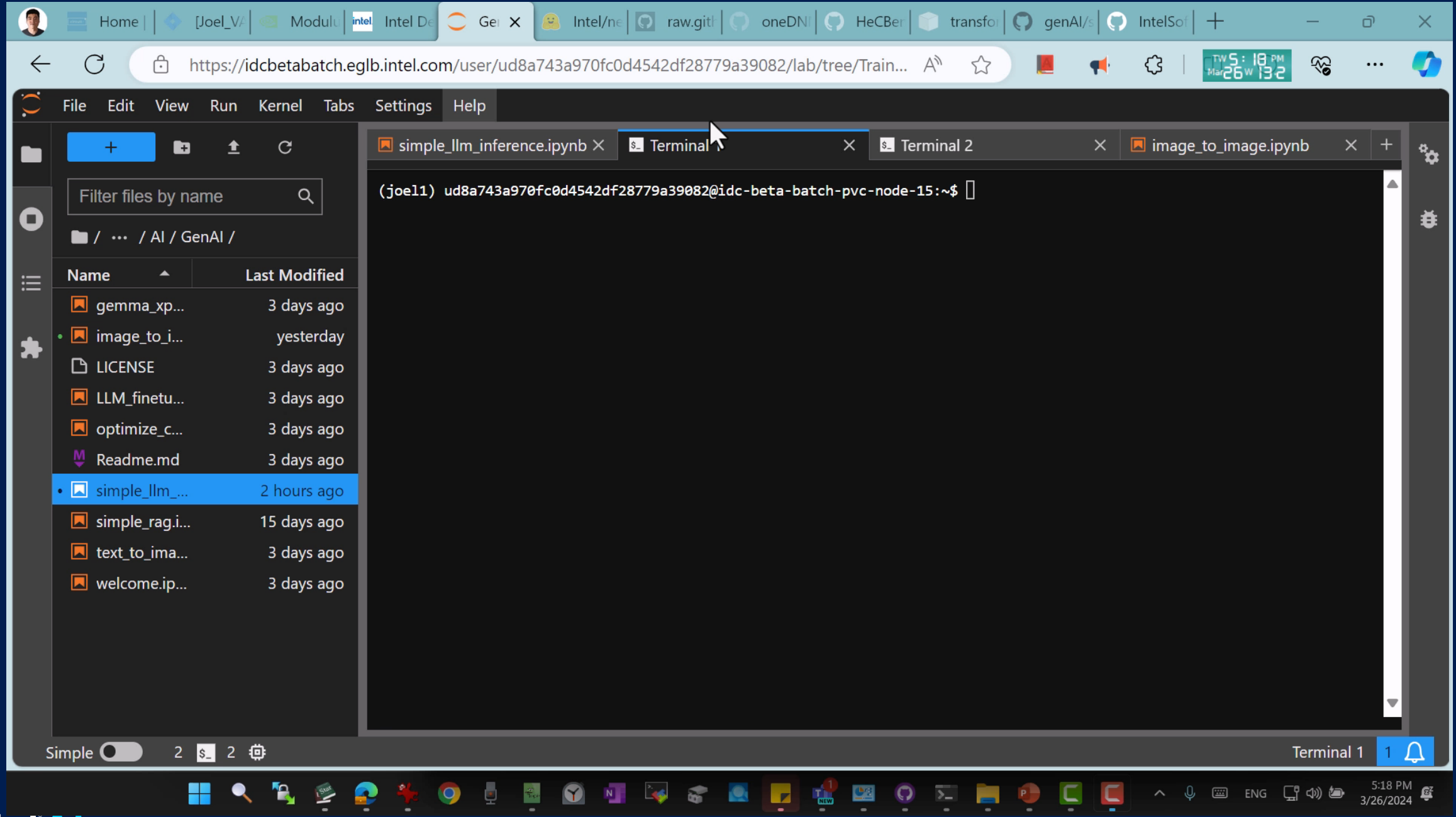
- More info at https://www.intel.com/aireferencekit
- Downloads available from GitHub at https://github.com/oneapi-src

intel AI summit

Simple LLM Inference: Playing with Language Models(Demo)



https://github.com/Joel-lin/IntelSoftwareWorkshop/tree/master/2024March27_AISummit



intel AI summit

Fine-tuning Llama 2 models on Intel® Data Center GPUs using BigDL LLM

Fine-tuning Llama 2 models on Intel® Data Center GPUs using BigDL LLM

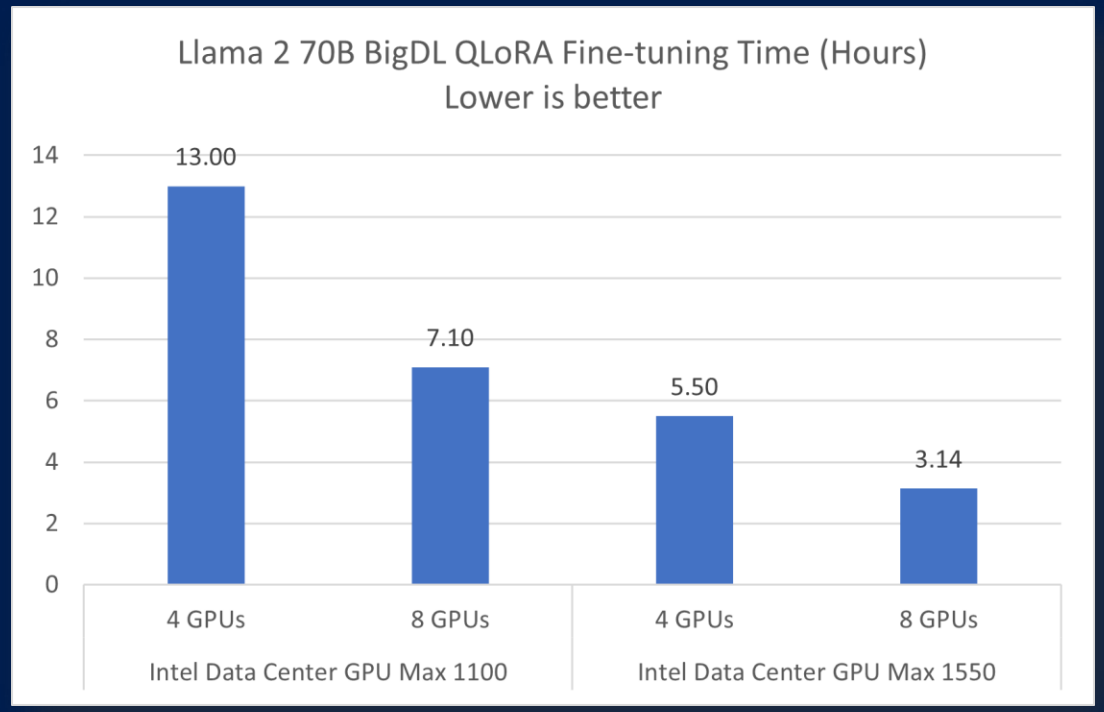


Figure 2. Llama 2 70B Fine-Tuning Performance on Intel® Data Center GPU

Refer to Configurations and Disclaimers for configurations

Fine-tuning larger LLMs, such as the Llama 2 70B, demands increased computational power, VRAM, and time. In our assessments with configurations of 4 and 8 Intel® Data Center GPU Max Series cards on a single server, we observed notable efficiency gains. Specifically, a single server equipped with 8 Intel® Data Center GPU Max Series GPUs significantly expedites the process, completing the fine-tuning of the Llama 2 70B model in roughly 200 minutes, or 3.14 hours. This setup emerged as the most efficient among those we tested.

Inference Performance data on Intel® Data Center GPUs

Accelerating LLM Inference on Intel Data Center GPUs using BigDL LLM

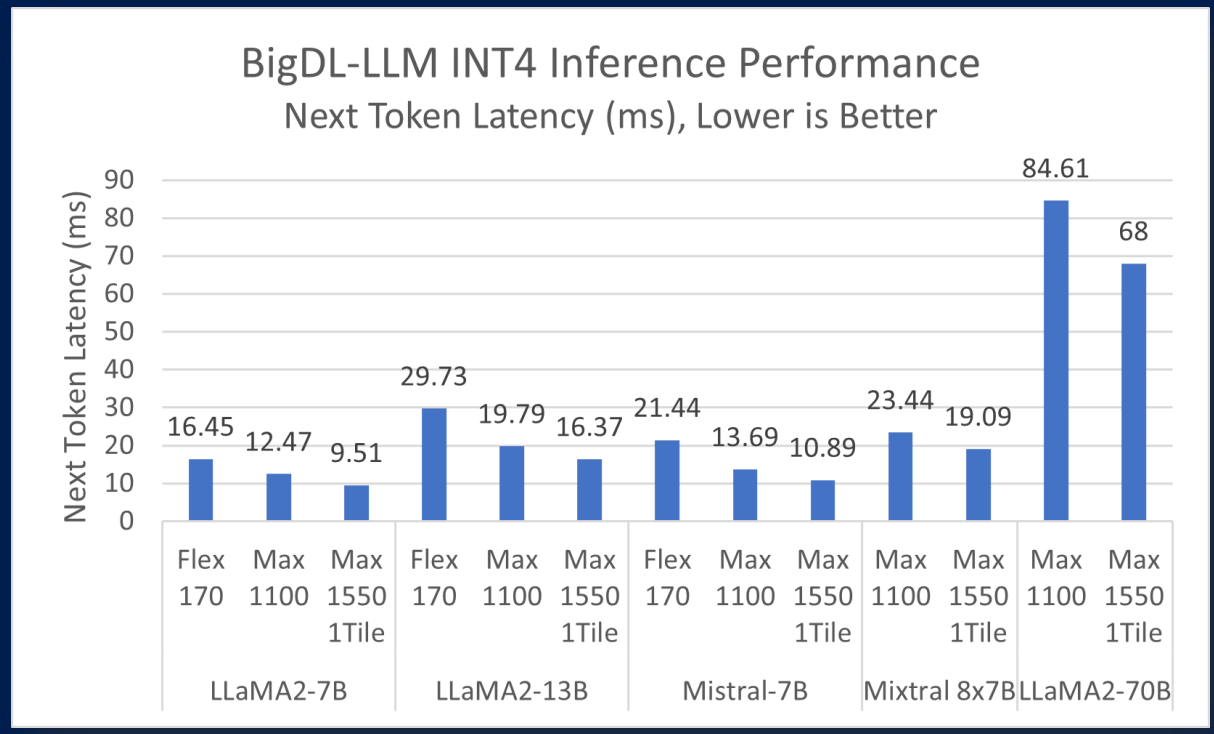


Figure 2. INT4 Inference Performance on Intel® Data Center GPUs

Refer to Configurations and Disclaimers for configurations.

With Self-Speculative Decoding, we observed significant latency improvement for FP16 inference (compared to without Self-Speculative Decoding). The graph below compares the inference latency for Llama2 7B/13B and Mistral 7B on Intel Data Center GPU Max 1550, under INT4 and FP16 using BigDL-LLM. In average, Self-Speculative Decoding brings about 35% improvements for FP16 on next token latency.

intel Ai summit

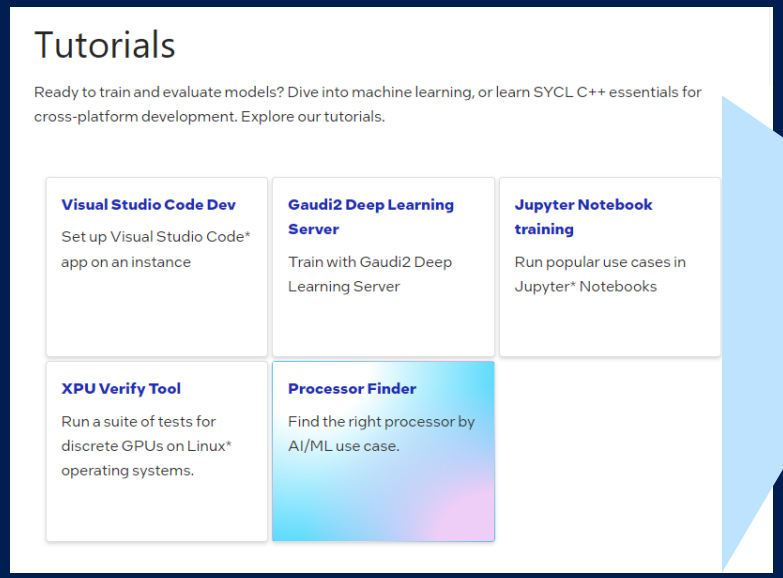
BigDL LLM repository: https://github.com/intel-analytics/BigDL

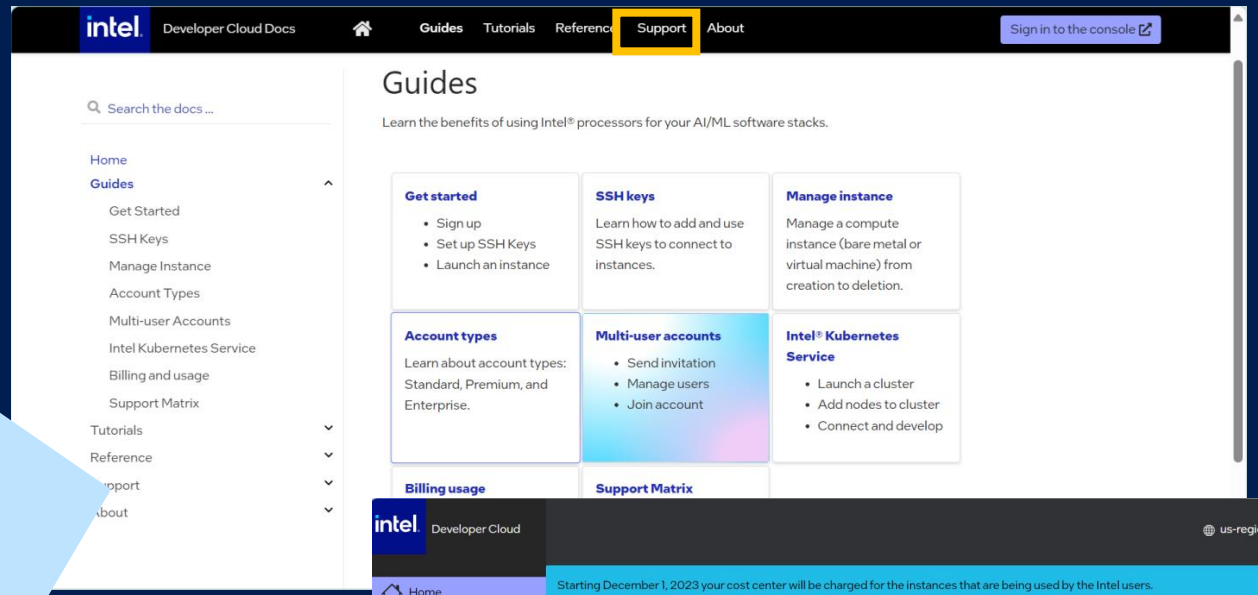
Intel® Developer Cloud

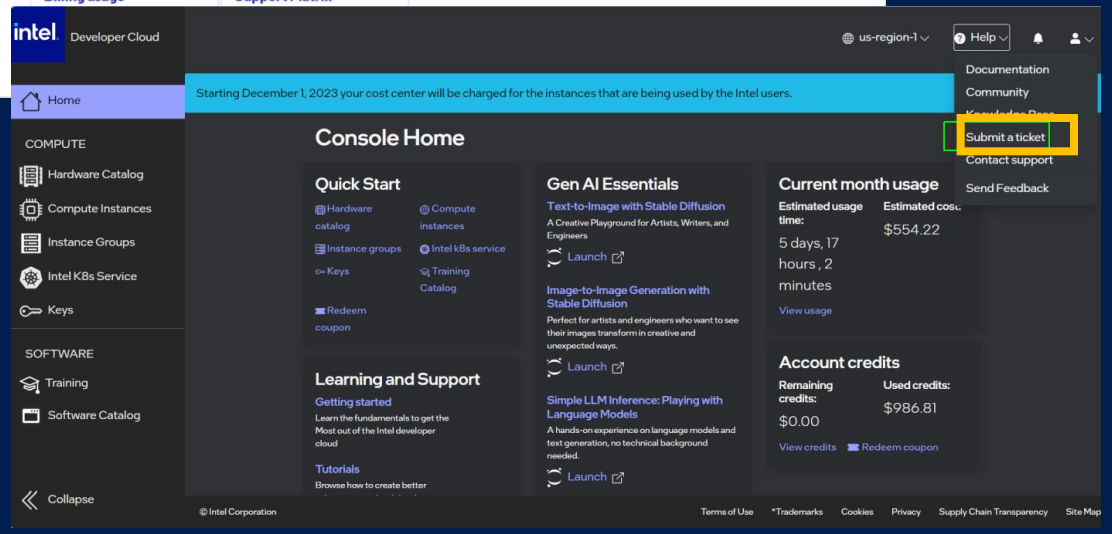
Online Documentation/Community Support/Submit a ticket

[Guides — Developer Cloud Docs documentation (intel.com)]

- Popular topics:
  - User account types
  - How to use SSH keys
  - Tutorials







intel Ai summit

Coupon for accessing Intel Developer Cloud:3ZM1-GFG9-ANS7

More Questions?
Professional and Community Support Available

Priority Support for Intel Toolkits

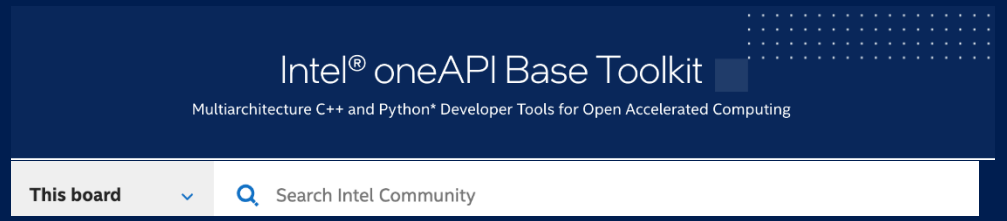
Every paid version of Intel® oneAPI Base, HPC, and Rendering Toolkit products includes Priority Support

- **Direct and private interaction** with Intel's support engineers, including the ability to submit confidential support requests
- **Accelerated response time** for toolkit-related technical questions and other product needs
- **Free download access** to all new product updates and continued access to older versions of the product
- **Ability to influence** product features and quality
- **Priority Support** for escalated defects
- **Access to a vast library** of self-help documentation that builds off decades of experience in creating high-performance code
- **Additional services at reduced cost**, including on-site or online training and consultation by Intel technical consulting engineers

Free Community Support

Connect with the Intel Community in public **Developer Software Forums**

- Supported by community technical experts and monitored by Intel Engineers
- Answers to commonly asked questions
- Access to online tutorials and self-help forums
- Troubleshooting guidance from fellow developers



intel Ai summit

您是否有興趣申請
Intel® Developer Cloud 優惠券？

掃描右側 QR Code
將能獲得一組使用優惠代碼
搶先體驗 Intel® 的硬體和軟體雲端 AI 服務

價值美金 $250，有效期限至 2024 年 7 月 30 日

intel Ai summit

Notices and Disclaimers

For notices, disclaimers, and details about performance claims, visit www.intel.com/PerformanceIndex or scan the QR code:

intel Ai summit

Thank You!