Cross TAB
oneAPI Community Forum
12/14/22

Purpose of the meeting

This brings together members of all the TABs for a single meeting to deliver an update on the 2022 activities and successes, and to set out a vision for the future of oneAPI and the community forum direction.

Goals for the meeting

- Highlight the activities and successes of 2022
- Introduce the new community forum organization
- Inspire increased collaboration

Agenda

Topic	Detail	Speaker	Approximate Timing
oneAPI Community Forum Update	Get a roundup of some of the highlights from 2022 as we close off the year. Find out about some changes to the organization of the oneAPI TABs and how the specification is managed. We'll also explain how to get involved in these new groups and how you can influence the direction of oneAPI.	Rod Burns, Codeplay Software	7:30am Pacific Time
Breakout Room Discussions	We will break out into rooms on a range of topics for discussion.	- AI - Hardware Abstraction - Language	7:50am Pacific Time

Stories from 2022	Hear from some of our members on what they have contributed to oneAPI this year.	Kevin Harms, Gordon Brown, Kentaro Kawakami and Robert Cohn	8:20am Pacific Time
The future - a cross vendor, industry standard programming	Find out what the future can look like for heterogeneous computing	Andrew Richards, Codeplay CEO	8:40am Pacific Time

Attendees:

Rod Burns, Codeplay

Alison Richards, Intel

Sanjiv Shah, Intel

Andrew Richards, Intel

Umar (?)

Xianmen Tian, Intel

Aksel Alpay, Univ of Heidelberg

Alexey Kukanov, Intel

Ronan Keryell, AMD

Andrew Barker, Intel

Andrey Alekseenko, KTH

Andrey Fedorov, Intel

Antonio Pena, BSC

Ben Ashbaugh, Intel

Christian Trott, Sandia National Laboratory

Daniel Chitnis, University of Edinburgh

En Shao, Chinese Academy of Science

Geoff Lowney, Intel

Gergana Slavova, Intel

Ioannis Sakiotis, Old Dominion University

James Reinders, Intel

John Melonakos, Intel

Juan Fumero, University of Manchester

Kentaro Kawakami, Fujitsu Lab

Kenneth Benzie, Codeplay

Kevin Harms, ANL

Kumadha Narasimhan, Codeplay

Medhi Goli, Codeplay

Mike Kinsner, Intel

Pablo Reble, Intel

Pat Quillen, Mathworks

Paul Petersen, Intel

Penporn Koanantakool, Google

Piotor Luszczek, ICL UTK (education)

Robert Cohn, Intel

Romain Dobleau, SiPearl

Sarah Knepper, Intel

Sastori

Stephanie Even, Mercedes Benz

Sunita Nadampalli, AWS (Graviton team)

Piezhao Qui, University of Edinburgh

Terry Cojean, Karlsruhe Institute of Technology (KIT)

Zack Waters, Intel

Notes from oneAPI Future Breakout:

Rod Burns, Codeplay

Antonio Pena, BSC

John Melonakos, Intel

Danial, University of Edinburgh

Piezhao Qui, University of Edinburgh

What are the perceptions of oneAPI

It needs to be clear where to start with oneAPI, at the moment there is a bit of confusion for this and compared to SYCL.

What is provided to developers needs to work and have user friendly documentation, there was a general feeling that this could be improved.

It needs to be giving a feeling of stability and reassurance it will be around a long time.

There needs to be a full commitment to an open strategy and roadmap.

There are pitfalls, it was felt other open standard groups ended up with more features than were needed because they did not want to say no to things. It's about finding the balance between open and keeping on track with what users need.

Languages and frameworks

There are a large set of languages used by researchers especially using supercomputer clusters. Need to understand how things like Python, Matlab and so on fit with oneAPI. How to bring seamless adoption of oneAPI within these languages and frameworks.

Notes from oneAPI Hardware Abstraction Breakout: (see attached pdf)

Multiple Layers of Hardware Abstraction

Currently, oneAPI supports various points of intercept for hardware abstraction. See Figure 1.

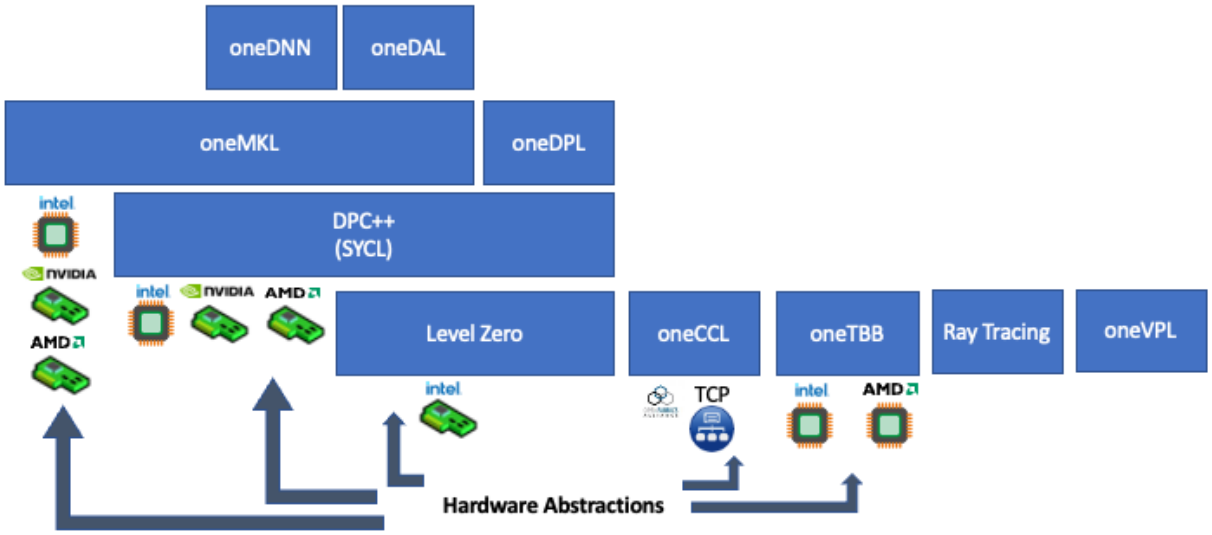

Figure 1

The ability to intercept at multiple layers was thought to be positive.

Level Zero offers a low-level device abstraction

The SYCL backend offers a higher level abstraction

oneMKL directly couples to optimized vendor libraries

This allows the implementers with robust, well-developed drivers and tool chains to intercept oneAPI at high

layers.

An implementer with an immature driver stack may prefer to intercept with Level Zero + SPIRV to reuse as much of the existing infrastructure of oneAPI.

A layered abstraction makes it easier to bring up new device hardware.

Applications users would like to reuse existing tools for performance profiling and debugging, oneAPI abstractions should allow for this.

Plan to include Unified Runtime in SIG

What are the minimum hardware requirements oneAPI should require?

The breakout discussed what minimum level of hardware requirements does oneAPI currently require and what should it require? What will it require in the future?

oneAPI is trending toward a rich set of features with lots of flexibility which implies a full features device

Could oneAPI support more narrowly focused devices?

What changes would propagate through the various specifications to make this happen?

A more focused device may imply some of the interfaces are supported via hardware

oneAPI layers could provide emulation of any missing hardware features

What level of device information must the dependent layers know about to understand device

capabilities?

Perhaps reference 64-bit atomic discussion

Future topics

Ideas for future breakout topic discussions.

Should the scope of the SIG be limited to Level Zero or cover Unified Runtime as well?

How to interact with other SIG working groups

Christian Trott asked who will maintain the backends for all of the DPC++ backends? Gordon said Codeplay will be doing that support and keeping up with the backends.