

# oneAPI DevSummit 2021 レポート

すがわら きよふみ  
iSUS 編集部

# 最初に注意を

このセッションで紹介する内容は、インテル社の公式な情報および見解ではなく、iSUS 編集部がセッションを視聴して独自に内容をまとめたものです

# はじめに

## イベントのサマリー

### SYCL\* 2020 の注目する機能

### 注目するセッションのサマリー

# ソフトウェアにおける 7 つ目のトレンド?

それぞれ 1958 年から 1973 年の間に誕生し、1990 年代/2000 年代に台頭し始めて、5 年強をかけて成熟したツール/言語/フレームワーク/ランタイム体系が構築されました

GUI オブジェクト GC ジェネリック ネットワーク



# ソフトウェア全体に与える影響の比較

	GUI	オブジェクト	GC	ジェネリック	ウェブ	並列化	XPU
アプリケーション・プログラミング・モデル	●●●	●●●	●	●	●	●●●	●●
ライブラリーとフレームワーク	●●●	●●●		●●	●●●	●●●	●●●
言語とコンパイラー	●●	●●●	●	●●		●●	●●●
ランタイムと OS	●●		●●		●	●●●	●●●
ツール (設計、測定、テスト)	●●●	●	●		●●	●●	●●●

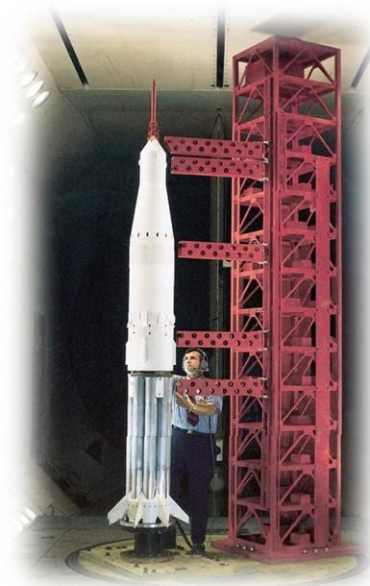
## 影響の範囲と重要性

- = 多少。1つの主要な製品リリース
- = 重要。1つ以上の製品リリース
- = 新しい考え方/必須。複数の主要リリース

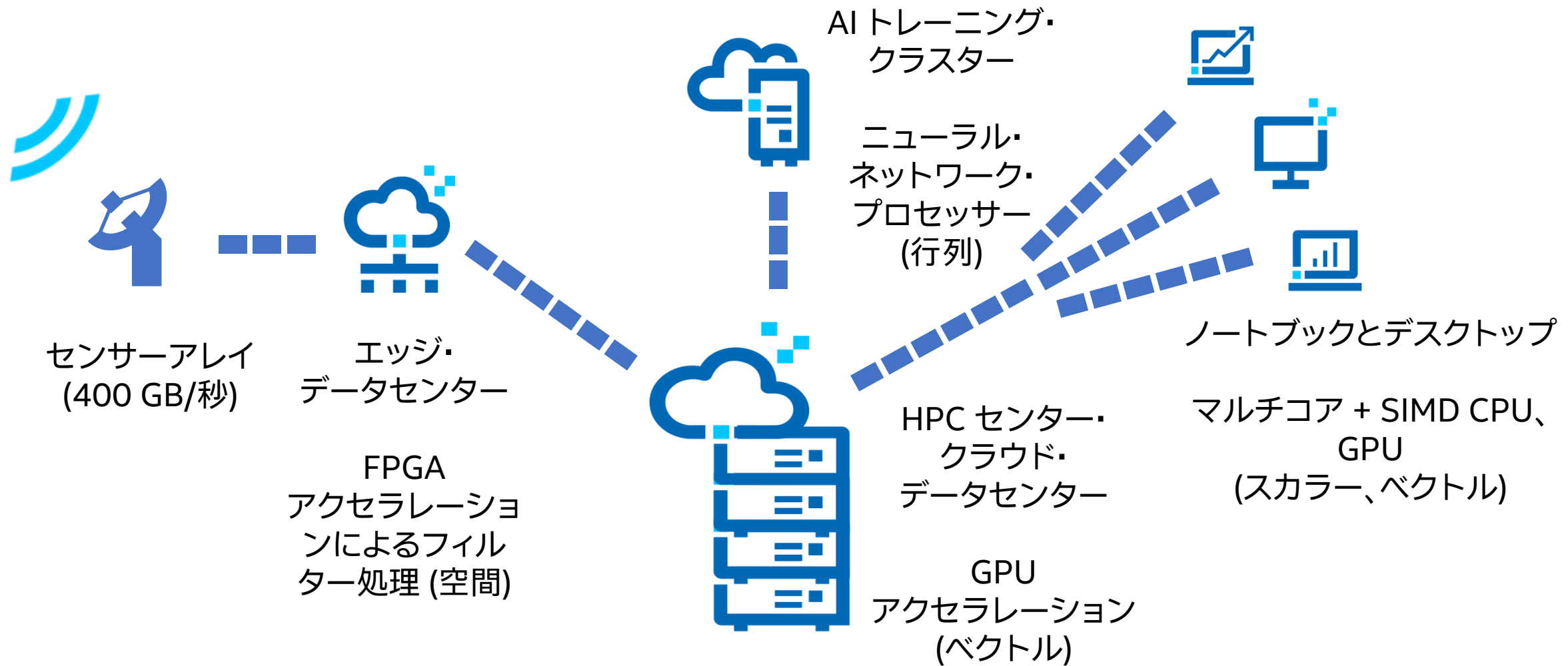
# XPU を活用するには

XPU の目的を達成するにはさまざまな手段が必要

- C/C++、または Fortran コンパイラーのみ
- コンパイラーと OpenMP\*
- コンパイラー、OpenMP\*、oneAPI と DPC++



# データ主導型の世界では、アクセラレーションが始まる...



多くのアプリケーション、多くのコンピューター、多くの場所...

# 多くの多様なプログラミング・モデル

1,000,000

1,000

抽象化プログラミング  
暗黙のパフォーマンス  
汎用性

ダイレクト・プログラミング  
明示的なパフォーマンス  
マイクロアーキテクチャー向けに最適化

Node.js\*  
Python\*

C/C++  
FORTRAN

OpenMP\*  
Parallel-For  
SIMD

OpenMP\* Target  
OpenACC\*

OpenCL\*  
CUDA\*

ベクトル組込み関数  
GPU 組込み関数  
Verilog\*



問題は?

1. 多くの言語やツールチェーンを学習し、展開して、保守する必要があります
2. パフォーマンスが重要なコードは、時として特定のマイクロアーキテクチャー (ベンダー) に依存します
3. パフォーマンスが重要なコードは、専門の開発者が必要になります
4. 多様なアーキテクチャーで高いパフォーマンスを実現する抽象化が必要です

はじめに

# イベントのサマリー

SYCL\* 2020 の注目する機能  
注目するセッションのサマリー

# oneAPI DevSummit at IWOCCL '2021

セッションカテゴリー	内容
VENDOR UPDATE	SYCL 2021 Vendor Update
DEV CLOUD UPDATE	Developer tools to get you started on oneAPI
HANDS-ON SESSION	Application optimization with Cache-aware Roofline Model and Intel oneAPI tools
<b>TECH TALK</b>	AI > A Deep Dive into a Deep Learning Library for the A64FX Fugaku CPU – Meet the Developer
LIGHTNING TALK	Great Cross-Architecture Challenge Application Showcase
KEYNOTE	SYCL 2020 in hipSYCL: DPC++ features on AMD GPUs, NVIDIA GPUs and CPUs
<b>LIGHTNING TALK</b>	Bringing SYCL to Super Computers with Celerity
LIGHTNING TALK	Great Cross-Architecture Challenge Application Showcase
TECH TALK	It's Acceleration but Faster! A Business Perspective on FPGA Development.
<b>TECH TALK</b>	Migrating and tuning a CUDA-based stencil computation to DPC++
TECH TALK	Comparative Analysis of Intel HLS Design Tools on a Case Study in Neuromorphic
TECH TALK	TAU Performance System

全セッションのビデオが公開されています: <https://www.oneapi.com/events/devcon2021iwocl/> (英語)

# 参加企業および団体



UNIVERSITÄT  
HEIDELBERG



中国科学院深圳先进技术研究院  
SHENZHEN INSTITUTES OF ADVANCED TECHNOLOGY  
CHINESE ACADEMY OF SCIENCES



UNIVERSITÀ DEGLI STUDI DI SALERNO



Sistema FIEB



PELO FUTURO DA INOVAÇÃO



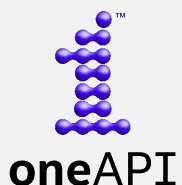
ParaTools

# oneAPI 業界イニシアチブ

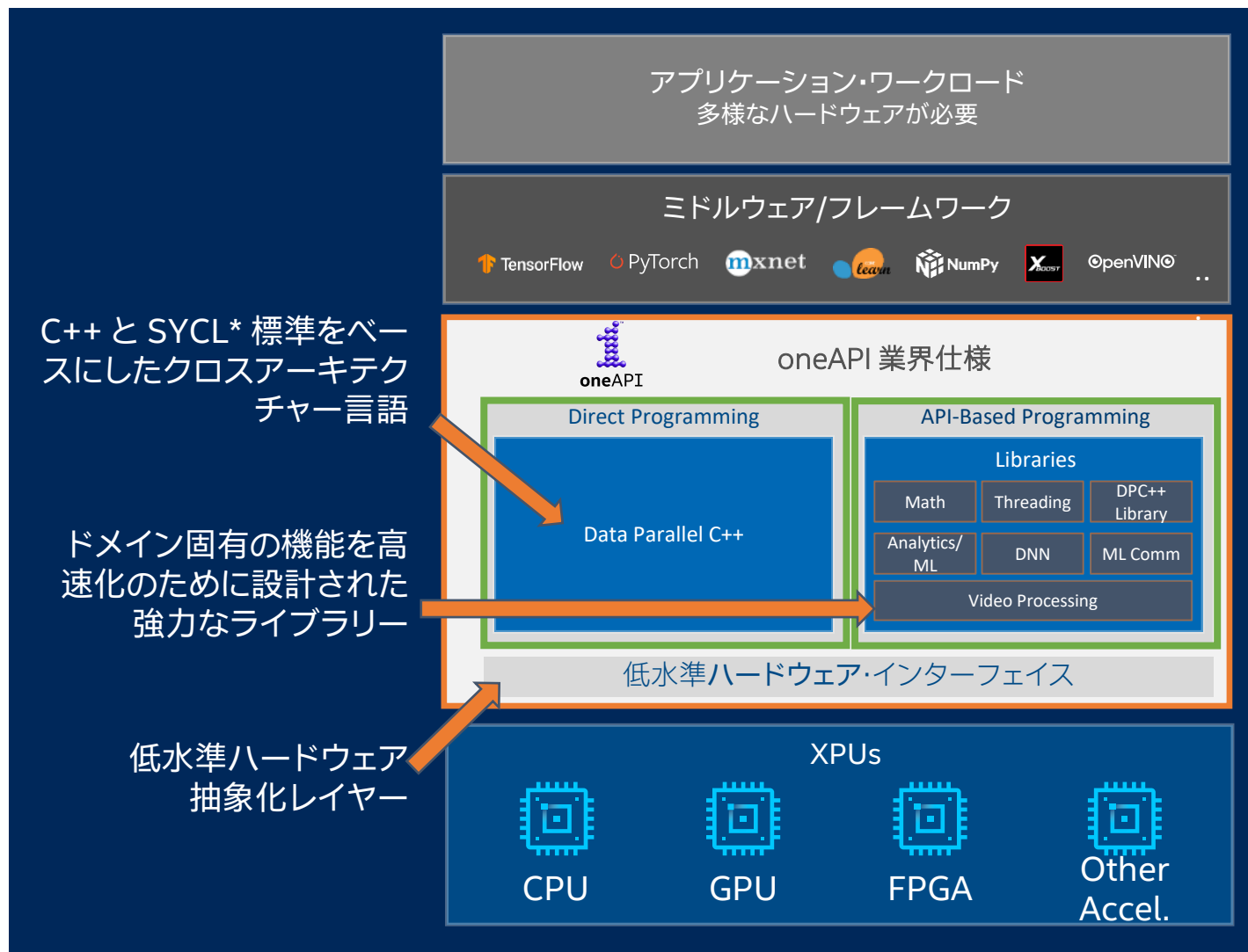
XPU で計算を行う自由度

コミュニティと業界の協力を促進するオープンな環境

複数のアーキテクチャーとベンダーでコードを再利用可能

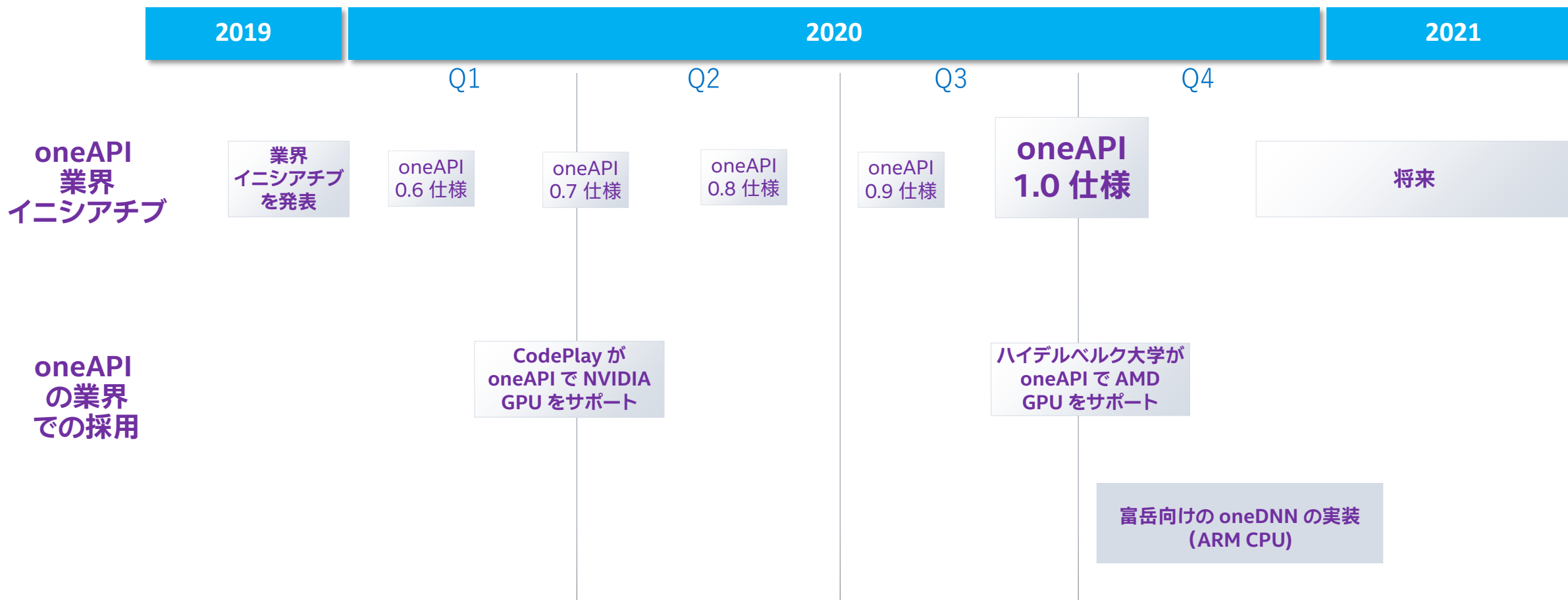
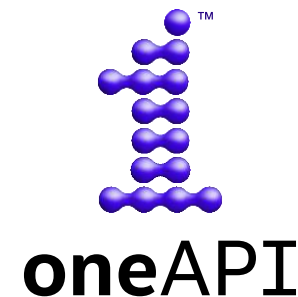


経済的および技術的な負担を伴う独自のプログラミング・モデルに代わって、高速なコンピューティングを実現する生産性に優れたスマートなパス



\* 社名、製品名などは、一般に各社の表示、商標または登録商標です。

# oneAPI 業界イニシアチブの歩み



\* 社名、製品名などは、一般に各社の表示、商標または登録商標です。

はじめに  
イベントのサマリー

# SYCL\* 2020 の注目する機能

注目するセッションのサマリー

このセッションのスライド配布はありません

はじめに  
イベントのサマリー  
SYCL\* 2020 の注目する機能  
**注目するセッションのサマリー**

# A64FX 富岳 CPU のディープ・ラーニング・ライブラリーの詳細

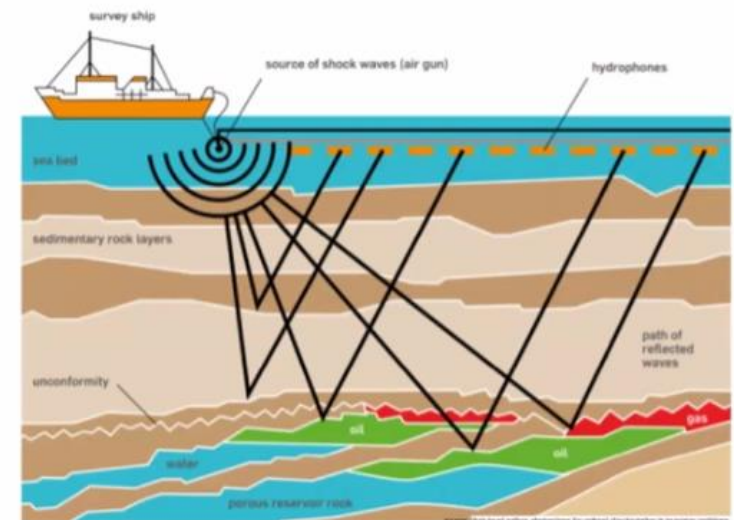
- oneDNN を Armv8-A 命令セットに対応
- 既存の汎用数値演算ライブラリーに比べ大幅にパフォーマンス向上

詳しくは、<https://blog.fltech.dev/entry/2020/11/18/fugaku-onednn-deep-dive-ja>

# CUDA\* ベースのステンシル計算を DPC++ へ移行およびチューニング

地震イメージング法を使用したリバーズ・タイム・マイグレーションによる資源調査

- CUDA\* ベースのコードをインテル® DPC++ 互換性ツールを使用して、CUDA\* から DPC++ ソースへ移行し、インテル® Advisor を使用して計算リソースとメモリー管理を最適化
- 有限差分の計算にアクセラレーターを活用
- 移行から最適化に要した日数は 19 日
- パフォーマンスを 2 倍向上 (計算集約型の部分は 4.9 倍)
- インテル® DevCloud のリソースを活用



# インテル® DPC++ 互換性ツール

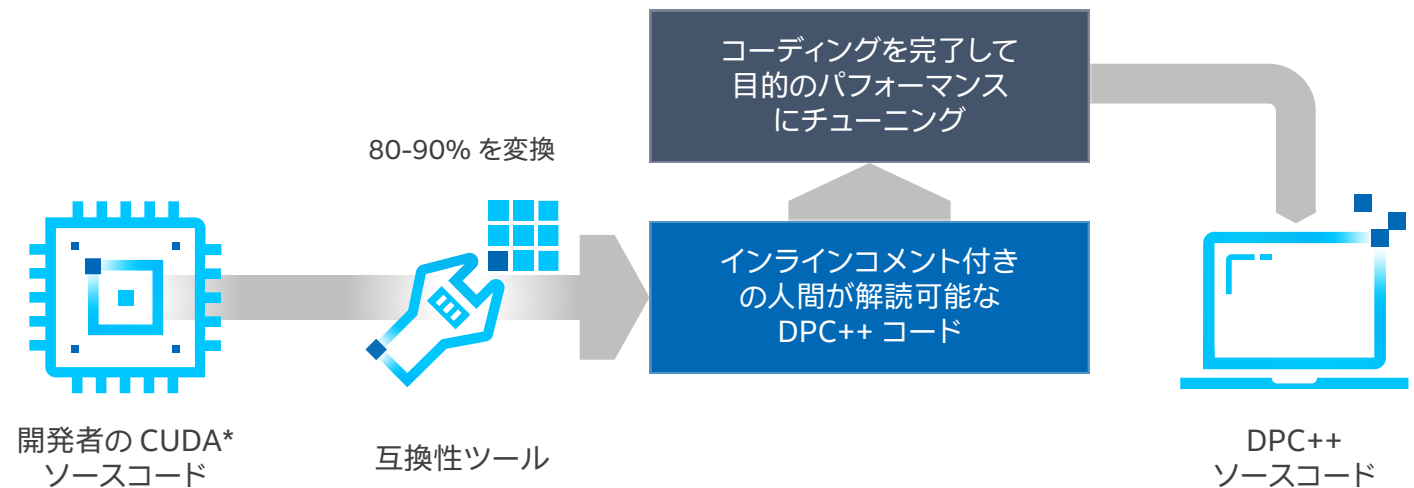
コードの移行時間を最小化

すでに CUDA\* で記述されている  
コードを DPC++ に移行する開発  
者を支援、可能な場合は**人間が解  
読可能なコード**を生成

通常はコードの 80-90% を自動的  
に移行

開発者がアプリケーションの移植  
を完了できるように支援するインラ  
インコメントを提供

インテル® DPC++ 互換性ツールの使用フロー



# インテル® Advisor

最新ハードウェア向けの設計を支援

## オフロードのモデル化

アクセラレーターにオフロードした際のパフォーマンスを推定

## ルーフライン解析

CPU/GPU コードの計算とメモリーを最適化

## ベクトル化アドバイザー

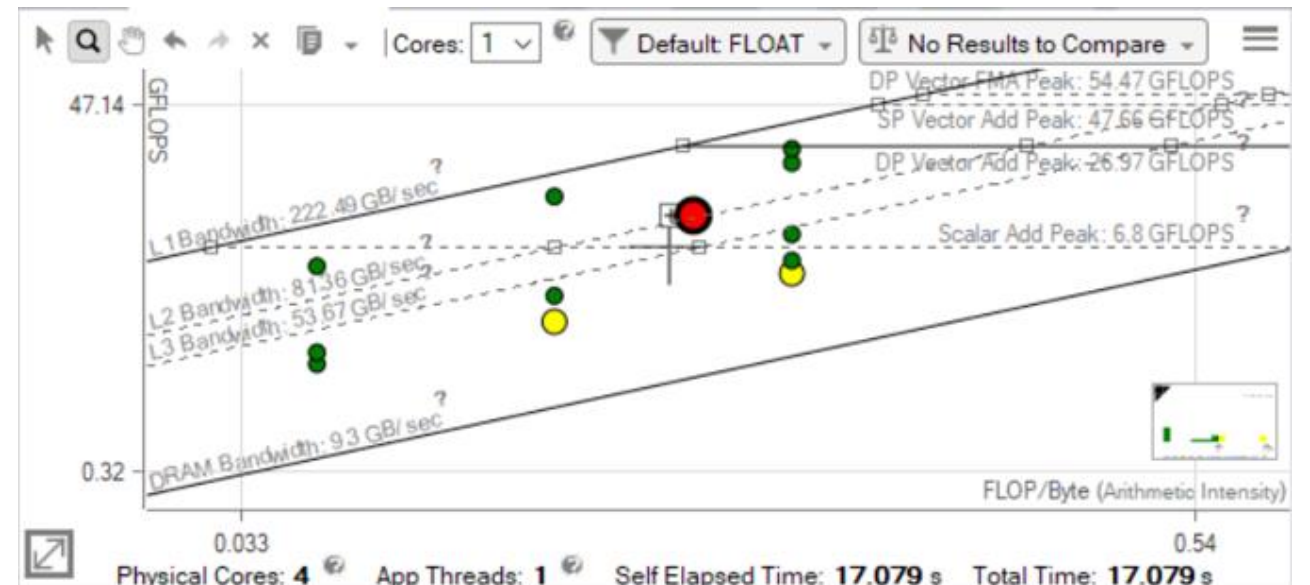
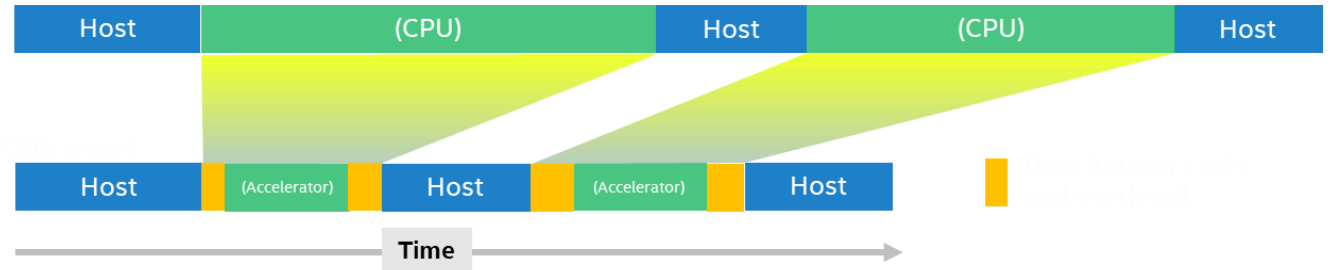
ベクトル化とその最適化

## スレッド化アドバイザー

スレッド化されていないアプリケーションに効率的なスレッド化を追加

## フローグラフ・アナライザー

効率的なフローグラフを作成して解析



# Celerity\* を採用するスーパーコンピューター に SYCL\* を導入

SYCL\* は異なるデバイス間の移植性をもたらしますが、プログラムを複数ノードに分散する必要があります

## MPI + SYCL\*

- 開発者は、複雑な分散コンピューティングを管理する必要があります
- さらにデータのパーティション化とスケジュールを考慮しなければなりません
- 明示的な通信も必要です

## Celerity\*

- SYCL\* コードの最小限の拡張です
- データのパーティション化とスケジュールは、分散されたランタイムシステムで管理されます
- 通信は暗黙的です

# Celerity\* とは?

Khronos の SYCL\* 業界標準ベースのハイレベル API

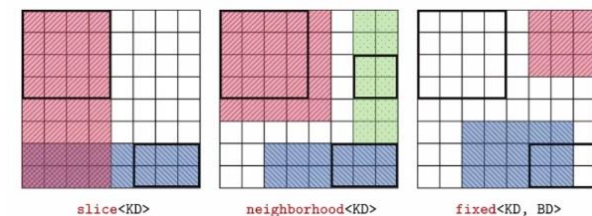
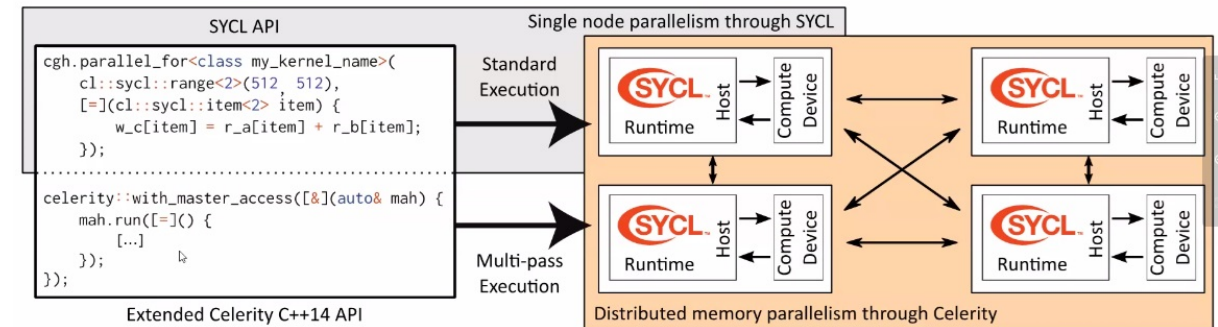
- 分散プログラミング向けのセマンティクスを導入
- SYCL\* 上に最小限の拡張セットとして実装
  - 現在は SYCL\* 1.2.1 のみをサポートしますが、間もなく SYCL\* 2020 に対応予定
- 一般的な分散メモリー・プログラミングの複雑性を排除

## 分散ランタイムシステム

- マルチパスの完全な非同期実行
- タスクグラフとコマンドグラフを実装

## レンジマッパーをサポート

- K 次元の実行レンジチャンクから N 次元のバッファボリュームへの任意の関数マッピング:  
celerity::chunk<KD> -> celerity::subrange<BD>
- Celerity\* API で事前に構築された抽象化の一般的なケース



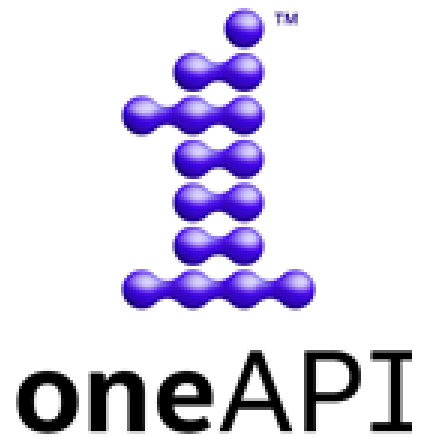
# これから …

- oneAPI を使用したクロスアーキテクチャーの並列アプリケーションを開発しましょう
  - 開かれています
  - 簡単です
  - わくわくします
- インテル® DevCloud for oneAPI で最新のインテル XPU とインテル® oneAPI ツールキットを体験してください
  - 無料です
  - そして便利
  - 柔軟性があります

# 日本語ドキュメント

- SYCL\* 2020 API リファレンス・ガイド  
<https://www.isus.jp/isus-translated-documents/>
- SYCL\* 2020 仕様 (リビジョン3) 翻訳中  
<https://www.khronos.org/registry/SYCL/> (英語)
- oneAPI 仕様リリース 1.0 Rev3 Part1  
<https://www.isus.jp/products/oneapi/oneapi-spec-japanese/>
- インテル® oneAPI プログラミング・ガイド  
<https://www.isus.jp/products/oneapi/oneapi-programming-guide-released/>
- インテル® oneAPI ポーティング・ガイド  
<https://www.isus.jp/products/c-compilers/oneapi-porting-guide-japanese/>
- oneAPI GPU 最適化ガイド  
<https://www.isus.jp/products/oneapi/oneapi-gpu-optimization-guide-released/>





## iSUS は oneAPI ベースのソフトウェア開発を支援します

Intel、インテル、Intel ロゴ、その他のインテルの名称やロゴは、Intel Corporation またはその子会社の商標です。

\*その他の社名、製品名などは、一般に各社の商標または登録商標です。

製品および性能に関する情報: 性能は、使用状況、構成、その他の要因によって異なります。詳細については、<http://www.intel.com/PerformanceIndex/> (英語) を参照してください。注意事項の改訂 #20201201

© 2021 Intel Corporation. 無断での引用、転載を禁じます。