



Why a single C++ API makes sense for Heterogeneous Compute

STAC Summit, 13 Nov 2019, London

INDUSTRY TRENDS IMPACTING FINANCIAL SERVICES

Higher performance

Focus on costs

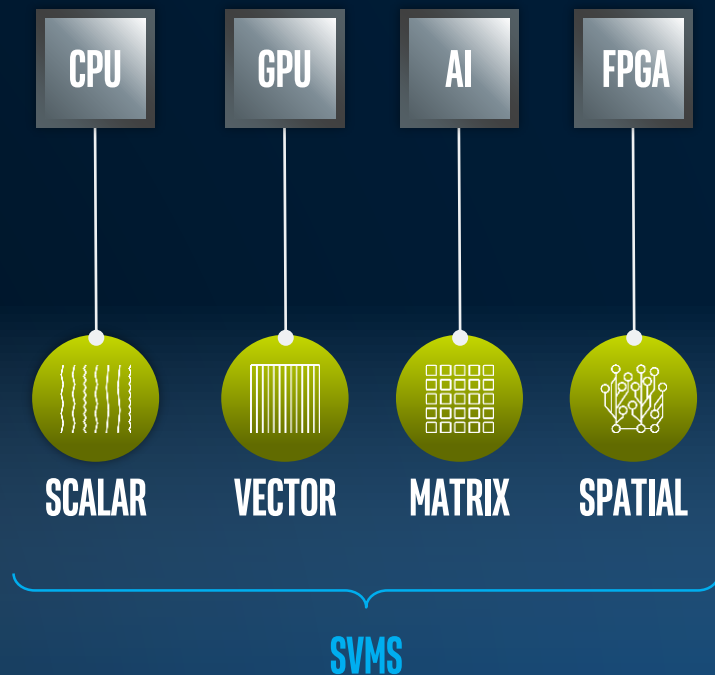
Energy efficiency

THE NEED FOR EFFICIENT COMPUTE



DIVERSE WORKLOADS REQUIRE DIVERSE ARCHITECTURES

The future is a **diverse** mix of scalar, vector, matrix, and spatial architectures deployed in CPU, GPU, AI, FPGA and other accelerators



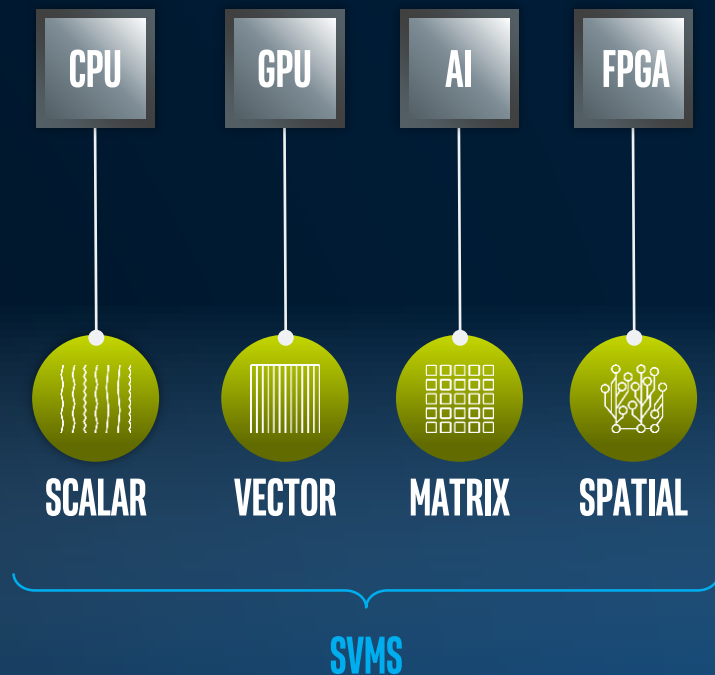
PROGRAMMING CHALLENGE

Diverse set of data-centric hardware

No common programming language or APIs

Each platform requires unique software investment

Inconsistent tool support across platforms



WHY NOT AN EXISTING LANGUAGE?

No common programming language or APIs exist today to address a diverse set of data-centric hardware

- Lack of commonality in code-bases and methodology, resulting in extra cost and delays

- Each platform requires unique software investment

Available data parallel languages are either proprietary or low-level, while there still exists inconsistent tool support across platform

UNIFIED PROGRAMMING MODEL ADVANTAGES

Cross-architecture support with extensibility for increased portability

Performance closer to respective native model/language/arch

Open standards based for increased productivity

DATA PARALLEL C++

STANDARDS-BASED, CROSS-ARCHITECTURE LANGUAGE

Language to deliver uncompromised parallel programming productivity and performance across CPUs and accelerators

Code reuse across hardware targets with custom tuning for a specific accelerator

Open, cross-industry alternative to single architecture proprietary language

Based on C++

Delivers C++ productivity benefits, using common C++ constructs

Incorporates SYCL* from the Khronos* Group to support data parallelism and heterogeneous programming with automatic scheduling of data movement

Community Project to drive language enhancements

Extensions to simplify data parallel programming

Open and cooperative development for continued evolution, e.g. unified shared memory support

Builds upon Intel's years of experience in architecture and compilers

Data Parallel C++

DPC++ Frontend

LLVM Runtime

DPC++ Frontend

New features

SYCL

C++

ONEAPI: SINGLE UNIFIED PROGRAMMING MODEL TO DELIVER CROSS - ARCHITECTURE PERFORMANCE

INTEL ONEAPI CORE CONCEPT

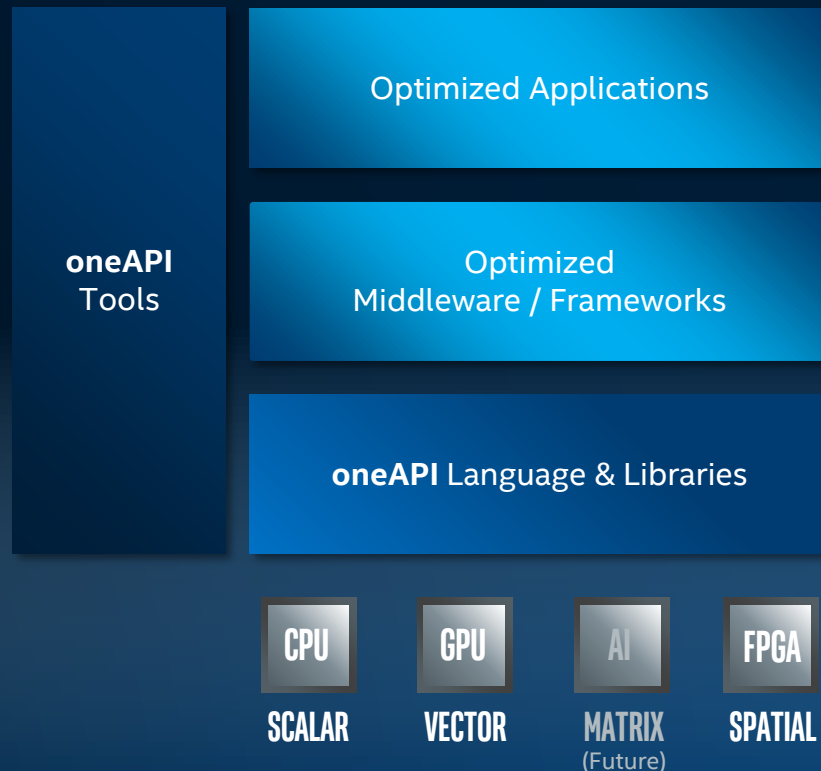
Project oneAPI delivers a unified programming model to simplify development across diverse architectures

Common developer experience across Scalar, Vector, Matrix and Spatial architectures (CPU, GPU, AI and FPGA)

Uncompromised native high-level language performance

Device-specific tuning will still be required to maximize performance

Based on industry standards and open specifications



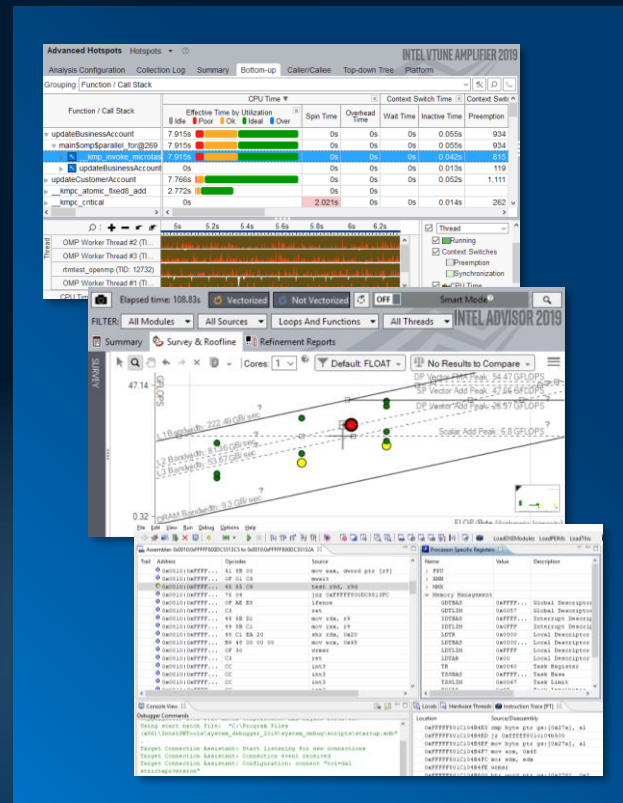
ADVANCED DEBUG & ANALYSIS TOOLS

Productive performance analysis across SVMs architectures

Performance Profiler

Parallelization Assistant

Debugger



WHERE TO LEARN MORE



Intel[®] oneAPI workshop

London, UK - December 11th 2019

Details on Data Parallel C++ fundamentals and programming model

Libraries and inter-operability, task parallel and data parallel working on concert

Programming with FPGA

[Register Now](#)

[Optimization Notice](#)

Copyright © 2019, Intel Corporation. All rights reserved.

*Other names and brands may be claimed as the property of others.



SUMMARY

Diverse workloads for data-centric computing are driving the need for diverse compute architectures including CPUs, GPUs, FPGAs, and AI accelerators

oneAPI unifies and simplifies programming of Intel CPUs and accelerators, delivering developer productivity and full native language performance

oneAPI is based on industry standards and open specifications to encourage ecosystem collaboration and innovation

NOTICES & DISCLAIMERS

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at [intel.com](https://www.intel.com), or from the OEM or retailer.

Copyright © 2019, Intel Corporation. All rights reserved. Intel, the Intel logo, Xeon, and VTune, are trademarks of Intel Corporation or its subsidiaries in the U.S. and other countries.

Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804