Success Story

Intel® Academic Program for oneAPI

Building Modern HPC Curriculum to Spur Adoption of Modern Heterogeneous Programming Methods

At Loyola University Chicago, Professor George K. Thiruvathukal is developing the curriculum codenamed unoAPI to provide a developer experience with oneAPI that is more akin to general application development in popular HPC languages such as Python.

Factors including AI and machine learning are making heterogeneous, accelerator-equipped hardware a critical class of target systems. oneAPI provides a unified API abstraction layer that enables developers to code once and run across heterogeneous collections of CPUs, GPUs and FPGAs. At the same time, however, computer science programs retain a tight focus on modern, higher-level languages such as Java and Python.

This competition for attention tends to create a gap between emerging hardware and the developers emerging to program for it. Professor George K. Thiruvathukal is helping to advance modernization of programming methods for heterogeneous architectures with a oneAPI-powered curriculum codenamed unoAPI. Dr. Thiruvathukal relates, "I chose "uno" to give the curriculum a world flavor. As someone who works to broaden participation in computing, my hope is that our curricular efforts can inspire a new, diverse group of faculty and students to adopt HPC."

Professor George K. Thiruvathukal, Loyola University Chicago.

While capturing oneAPI's ability to spare both students and their instructors from tedious low-level coding paradigms, the curriculum also provides coding exemplars and software engineering methods that facilitate adoption.

In particular, Dr. Thiruvathukal's work seeks to expand the reach of oneAPI to a generation of developers accustomed to the highly developed ecosystems of libraries and repositories available for Java, Python and Go. It also incorporates relatively recent additions to modern C++, such as lambda expressions and other ideas from functional programming. Established techniques to work with external dependencies help avoid the need for ad hoc solutions to common tasks, so programmers can focus instead on core scientific and technical problems.

"We're excited about the oneAPI curriculum we're creating, and the collaborative nature of Intel's academic programs to educate the future technologists."

– Professor George K. Thiruvathukal, Department of Computer Science, Loyola University

Building on foundational resources to teach oneAPI

The Intel® Academic Program for oneAPI has proven to be a vital set of assets for Dr. Thiruvathukal as he builds out the unoAPI curriculum. He draws on a range of resources that range from books and documentation to online examples and tools, which supplement his own adaptations of existing courseware for oneAPI. Access to Intel® Developer Cloud through the Program gives students an indispensable testbed for understanding the capabilities and constraints of different Intel hardware platforms, to explore firsthand how various accelerators can enable their oneAPI/SYCL codes.

The relationship between Dr. Thiruvathukal, his education collaborators (Konstantin Läufer), his students and Intel extends to the Intel architecture-based Aurora supercomputer at Argonne National Laboratory, where he is a visiting computer scientist. Thiruvathukal summarizes this opportunity, "The Aurora computer is an exciting development (and partnership with Intel) that will need a new pool of talent to develop programs that take advantage of it." With a forward-looking vision that incorporates the power of oneAPI for heterogeneous programming, his students are a cohort well situated to provide that talent.

oneAPI: Programming for the New Era of Heterogeneous Computing

Hardware accelerators are critical for maximizing throughput and energy efficiency while driving down workload latency and cost on commercial-off-the-shelf (COTS) servers. Developers have used performance engines such as GPUs and FPGAs to supplement the CPU for years, although proprietary programming models such as CUDA have limited the reach of those efforts.

oneAPI changes all that, with a single, open model for code that can execute on CPU cores as well as various hardware accelerators. **Intel oneAPI toolkits** provide best-in-class compilers, performance libraries, frameworks and analysis and debug tools, so developers can code once and run anywhere, from the largest supercomputers to compute nodes on the distributed edge.

Extending modern paradigms to define the future of programming

As the industry adapts programming methods to meet the emerging reality of pervasive heterogeneous hardware, it is also shedding older inflexible approaches such as proprietary GPGPUs. Instead, Dr. Thiruvathukal's unoAPI curriculum builds on oneAPI support for diverse platforms — including non-Intel ones — with the advantages on novel architectures afforded by ahead-of-time compiled languages such as C++.

This goal drives Dr. Thiruvathukal as he develops course materials for modern C++ and oneAPI. Dr. Thiruvathukal sees in oneAPI an enormous potential to transform the industry. He likens it to the early Java value proposition at Sun Microsystems, to run a single codebase with uniformly excellent results across platforms, but even better without the overhead of a virtual machine. In oneAPI, Dr. Thiruvathukal sees a similar realm of potential for HPC and tomorrow's implementations of technical computing based on future AI and machine learning.

Using a curriculum holistically constructed to look beyond language and framework to an ecosystem focus backed by solid software engineering principles, he is engaging the emerging generation of developers. Connecting HPC to familiar programming concepts and techniques, Dr. Thiruvathukal intends to "help build a pipeline of students who come to HPC research with proven, state-of-the-art software engineering and modern systems programming habits." His call to the future is clearly given.

Learn More:

- Intel® Academic Program for oneAPI
- The UnoAPI Curricular Project
- UnoAPI: Balancing Performance, Portability, and Productivity (P3) in HPC Education in HPC Education

0723/OT/MESH/PDF 353922-001US