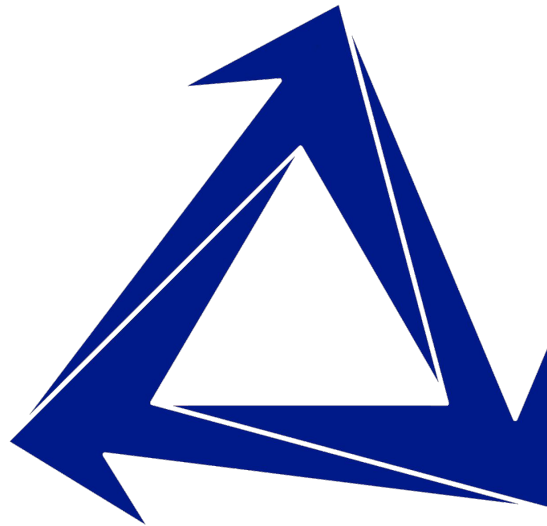


Inspira Crea Transforma



APOLO

**Centro de
computación científica**

Universidad EAFIT

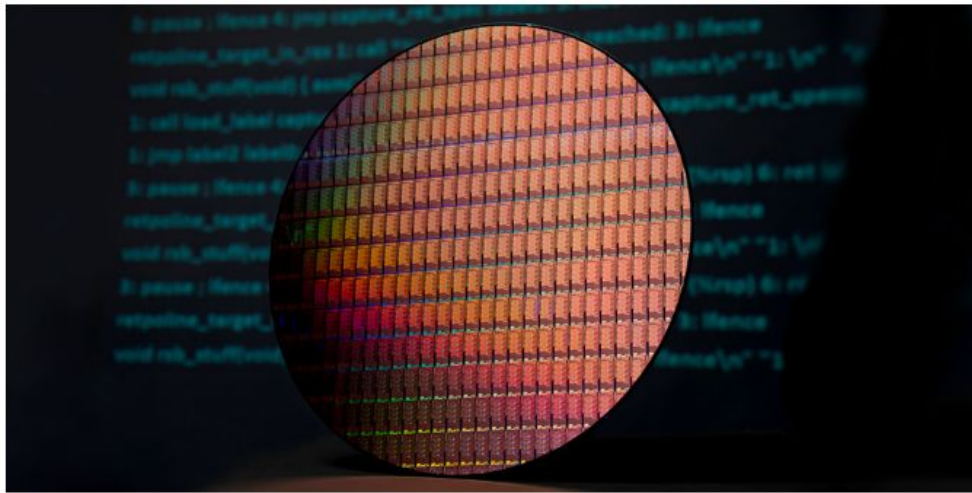
Inspira Crea Transforma

**UNIVERSIDAD
EAFIT[®]**

Optimizando su código serial y paralelo con algunas de las herramientas de desarrollo de Intel

Juan David Pineda-Cárdenas
<jpineda2@eafit.edu.co>

Ciclo de Conferencias
Centro de Computación Científica APOLO
Universidad EAFIT



ADVANCING SECURITY AT THE SILICON LEVEL

Hardware-based Protection Coming to Data Center and PC Products Later this Year



By Brian Krzanich

In addressing the vulnerabilities reported by Google Project Zero earlier this year, Intel and the technology industry have faced a significant challenge. Thousands of people across the industry have worked tirelessly to make sure we delivered on our collective priority: protecting customers and their data. I am humbled and thankful for the commitment and effort shown by so many people around the globe. And, I am reassured that when the need is great, companies – and even competitors – will work together to address that need.

But there is still work to do. The security landscape is constantly evolving and we know that there will always be new threats. This was the impetus for the [Security-First Pledge](#) I penned in January. Intel has a long history of focusing on security, and now, more than ever, we are committed to the

Tomado de:
<https://newsroom.intel.com/editorials/advancing-security-silicon-level/>

Avance de la seguridad a nivel del silicio

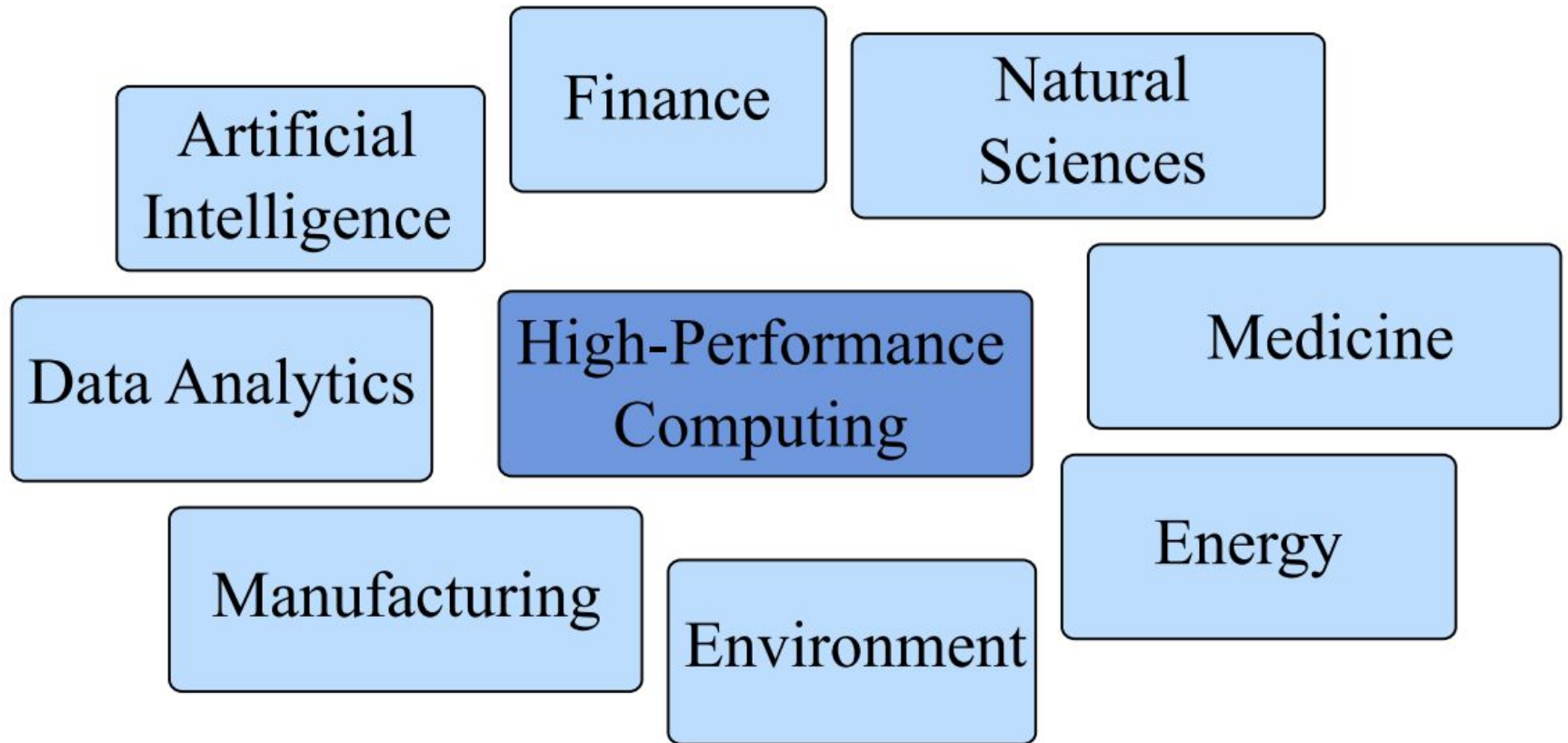
*“Today, I want to provide several updates that show continued progress to fulfill that pledge. First, **we have now released microcode updates for 100 percent of Intel products launched in the past five years that require protection against the side-channel method vulnerabilities discovered by Google.** As part of this, I want to recognize and express my appreciation to all of the industry partners who worked closely with us to develop and test these updates, and make sure they were ready for production.”*

By Brian Krzanich - CEO Intel Corp.

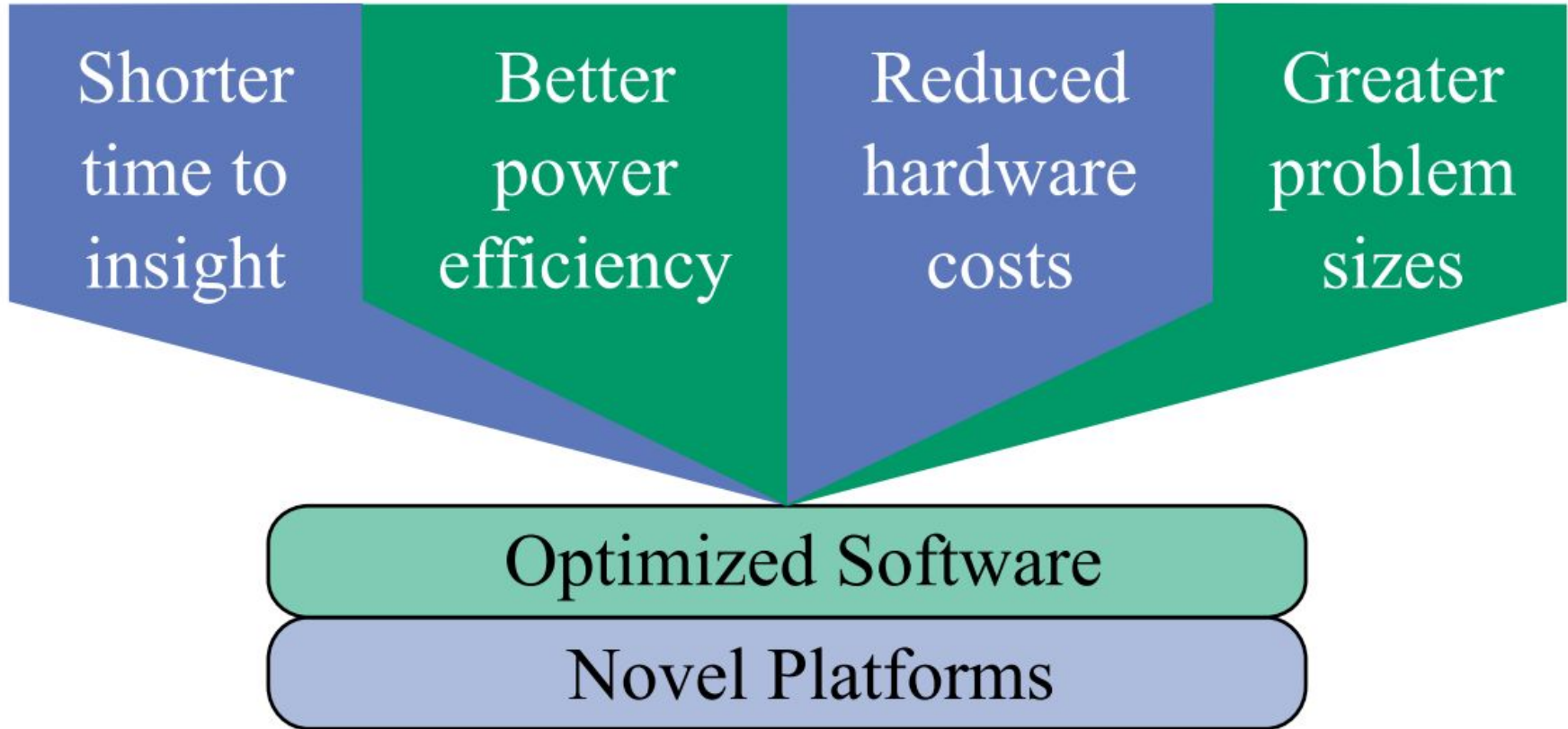
Avance de la seguridad a nivel del silicio

“While Variant 1 will continue to be addressed via software mitigations, we are making changes to our hardware design to further address the other two. We have redesigned parts of the processor to introduce new levels of protection through partitioning that will protect against both Variants 2 and 3. Think of this partitioning as additional “protective walls” between applications and user privilege levels to create an obstacle for bad actors.”

By Brian Krzanich - CEO Intel Corp.



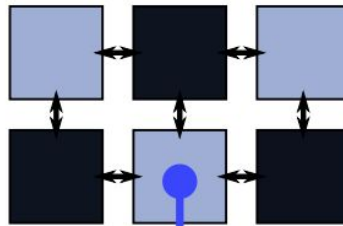
Tomado de: Fundamentals of Parallelism on Intel Architecture. Colfax Research.
Intel. Coursera. <https://es.coursera.org/learn/parallelism-ia>



Tomado de: Fundamentals of Parallelism on Intel Architecture. Colfax Research.
Intel. Coursera. <https://es.coursera.org/learn/parallelism-ia>

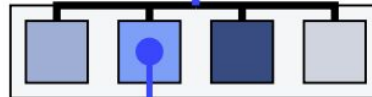
Capas de Programación Paralela

CLUSTER COMPUTING
in distributed memory



```
MPI_Sendrecv(data, k,  
MPI_DOUBLE, data2,  
... );
```

MULTITHREADING
in shared memory



```
#pragma omp parallel for  
for (j = 0; j < m; j++)  
  ComputeSubset(j);
```

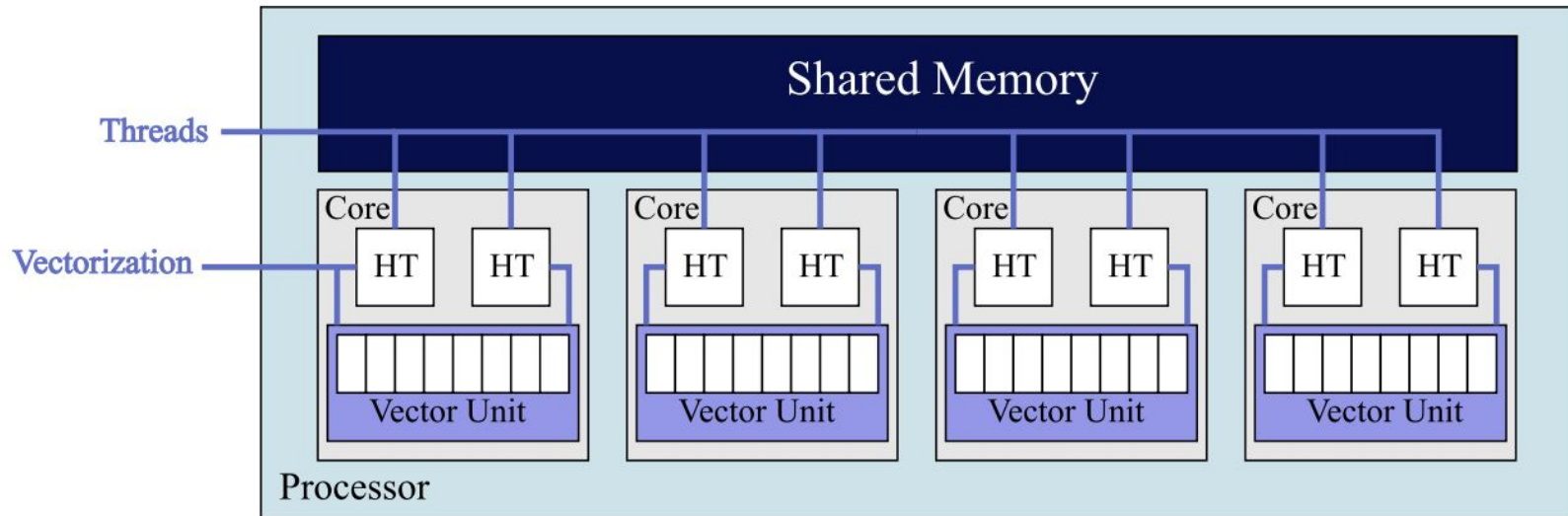
VECTORIZATION
of floating-point math



```
#pragma omp simd  
for (i = 0; i < n; i++)  
  A[i] += B[i];
```

Tomado de: Fundamentals of Parallelism on Intel Architecture. Colfax Research.
Intel. Coursera. <https://es.coursera.org/learn/parallelism-ia>

Paralelismo: Núcleos y vectores



Unbounded growth opportunity, but **not automatic**

Tomado de: Fundamentals of Parallelism on Intel Architecture.
<https://es.coursera.org/learn/parallelism-ia>

Plataformas de Cómputo de Intel

General-Purpose Processors

Intel® Xeon®
Intel® Core™
Intel® Atom™, ...



Specialized Processors

Intel® Xeon Phi™
processors
and coprocessors



Computing Accelerators

Intel® VCA (x86)
Intel® Nervana™ Platform
Intel® DLIA™ (FPGAs)



Network Interconnects

Intel® Omni-Path™
Architecture



Tomado de: Fundamentals of Parallelism on Intel Architecture.
<https://es.coursera.org/learn/parallelism-ia>

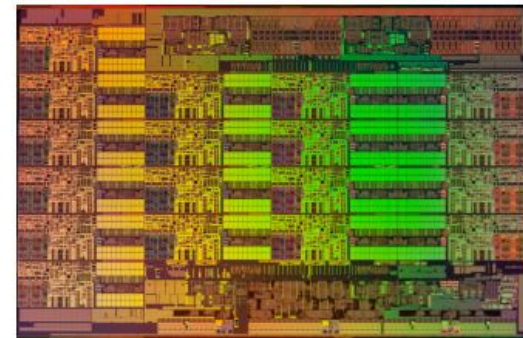
Intel Xeon

- 1-, 2-, 4 CPUs por Servidor
- Propósito General
- Altamente Paralelos (44 núcleos*)
- Rico en recursos (+ cores + cache)
- “*Forgiving Performance*”
- ~ 1.0 TFLOP/s en DP* (Teórico)
- ~ 154 GB/s de ancho de banda* (Medido)

*2-way Intel Xeon, Broadwell architecture (2016), top of the line. (e.g. E5-2699 V4)

Tomado de: Fundamentals of Parallelism on Intel Architecture.

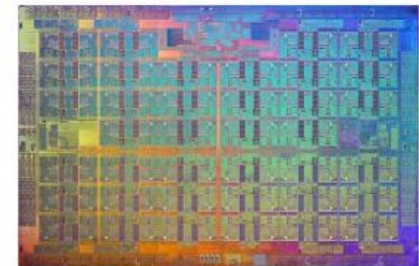
<https://es.coursera.org/learn/parallelism-ia>



Intel Xeon Phi (2a. Generación)

- Booteable PCIe add-in card
- Especializado para HPC
- Altamente paralelo (72 núcleos*)
- Balanceado para cómputo
- “*Less forgiving than Xeon*”
- ~ 3.0 TFLOP/s en DP* (Teórico)
- ~ 490 GB/s de ancho de banda* (Medido)

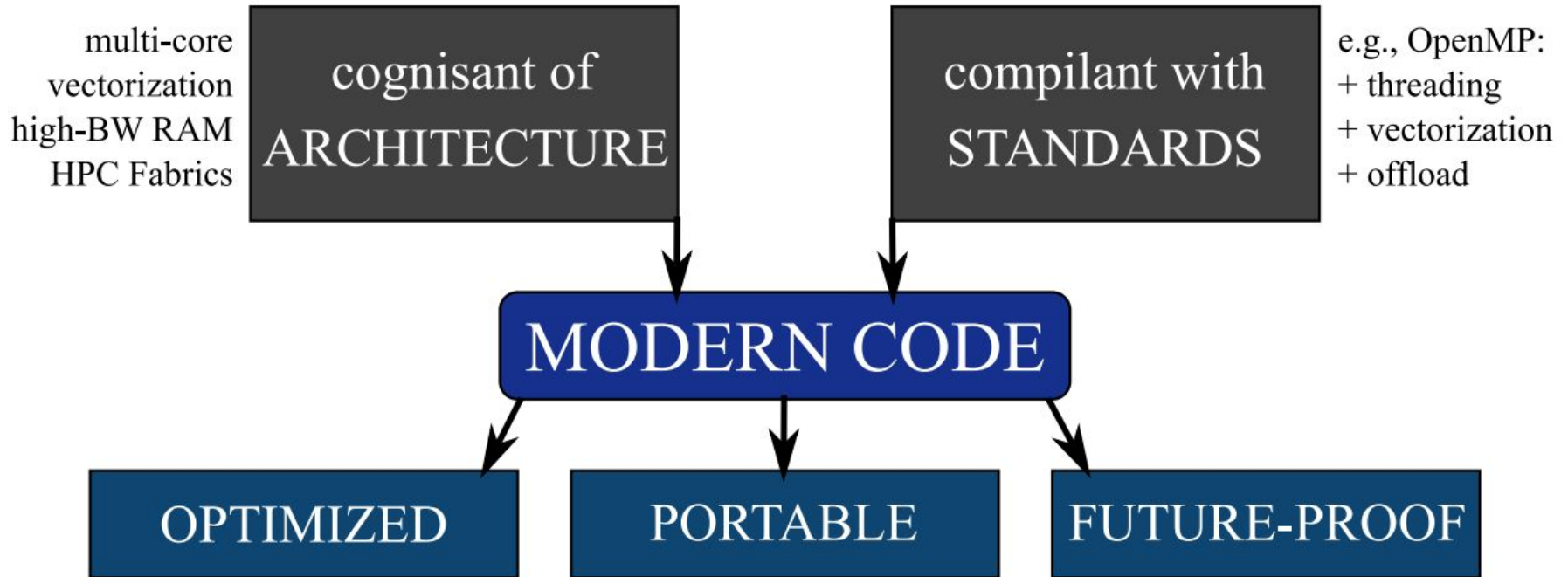
* Intel Xeon Phi, Knights Landing architecture (2016), top of the line. (e.g. 7290P)



Tomado de: Fundamentals of Parallelism on Intel Architecture.

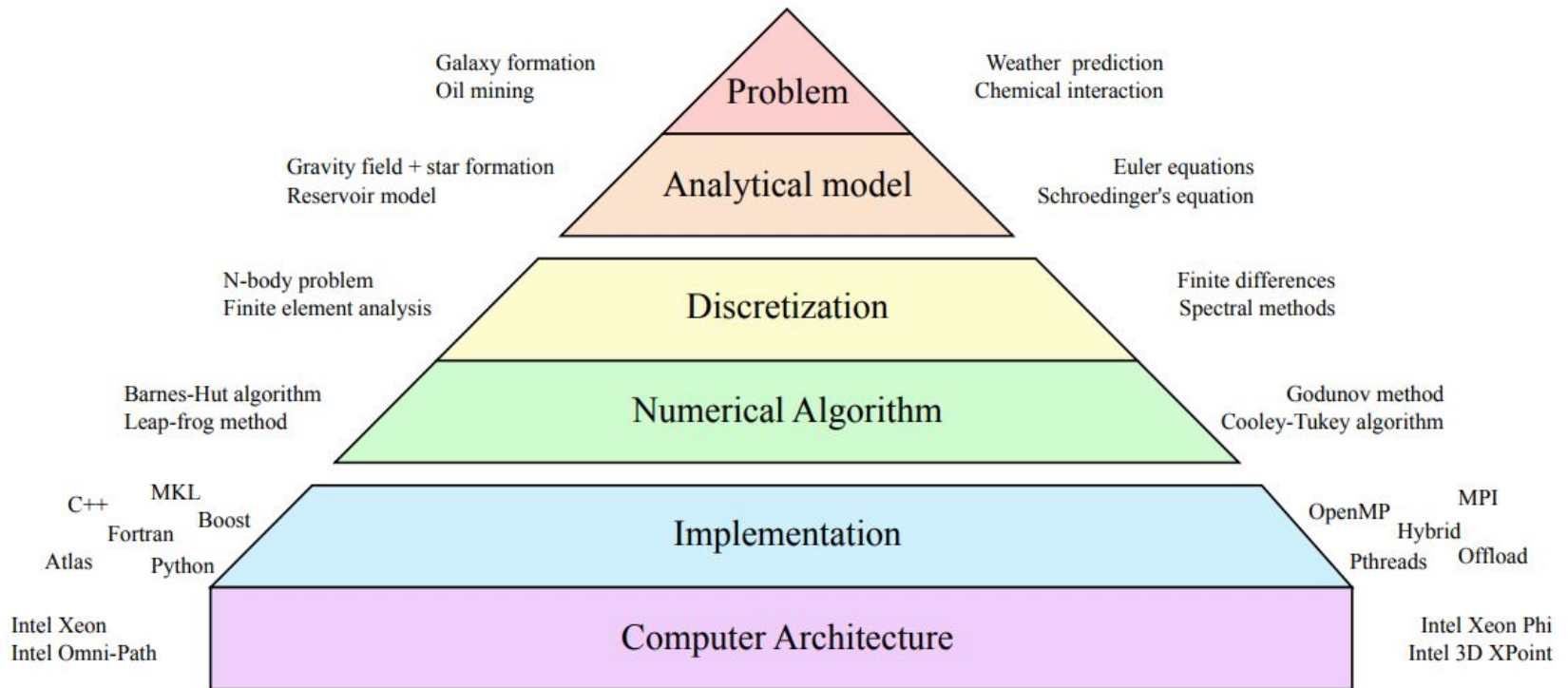
<https://es.coursera.org/learn/parallelism-ia>

Un código para todas las plataformas



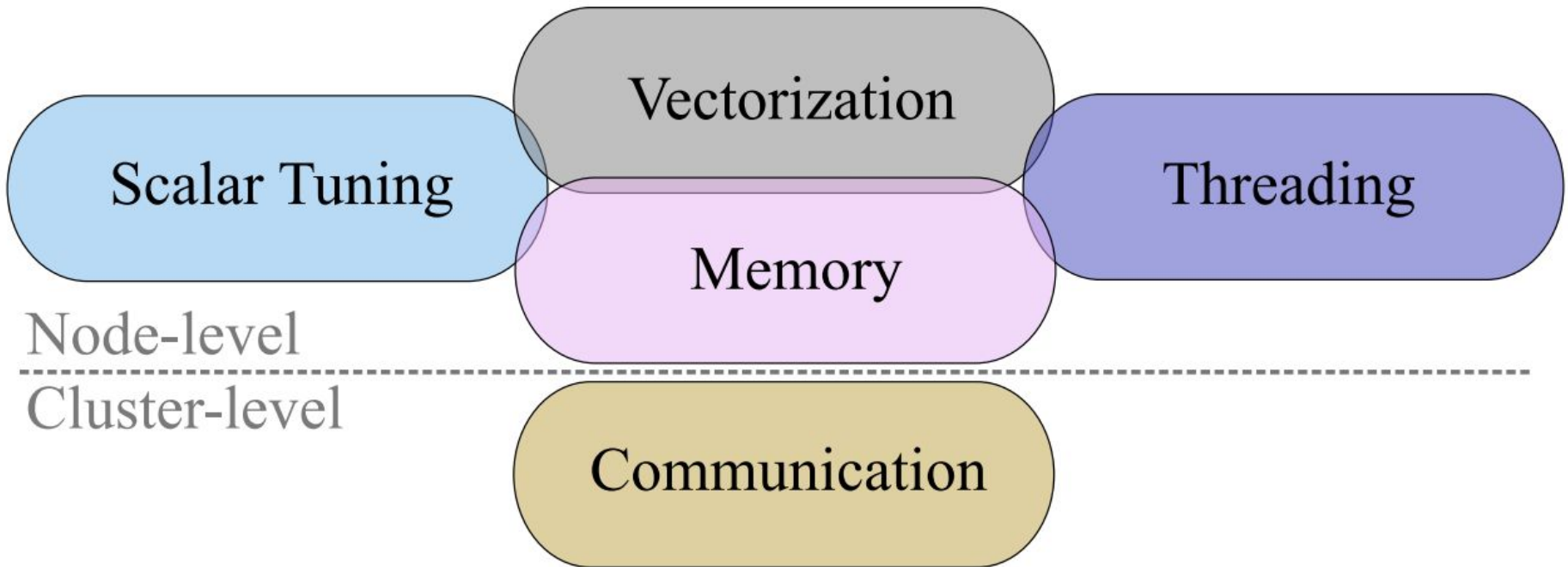
Tomado de: Fundamentals of Parallelism on Intel Architecture.
<https://es.coursera.org/learn/parallelism-ia>

Computación en Ciencia e Ingeniería



Tomado de: Fundamentals of Parallelism on Intel Architecture.
<https://es.coursera.org/learn/parallelism-ia>

Áreas de Optimización

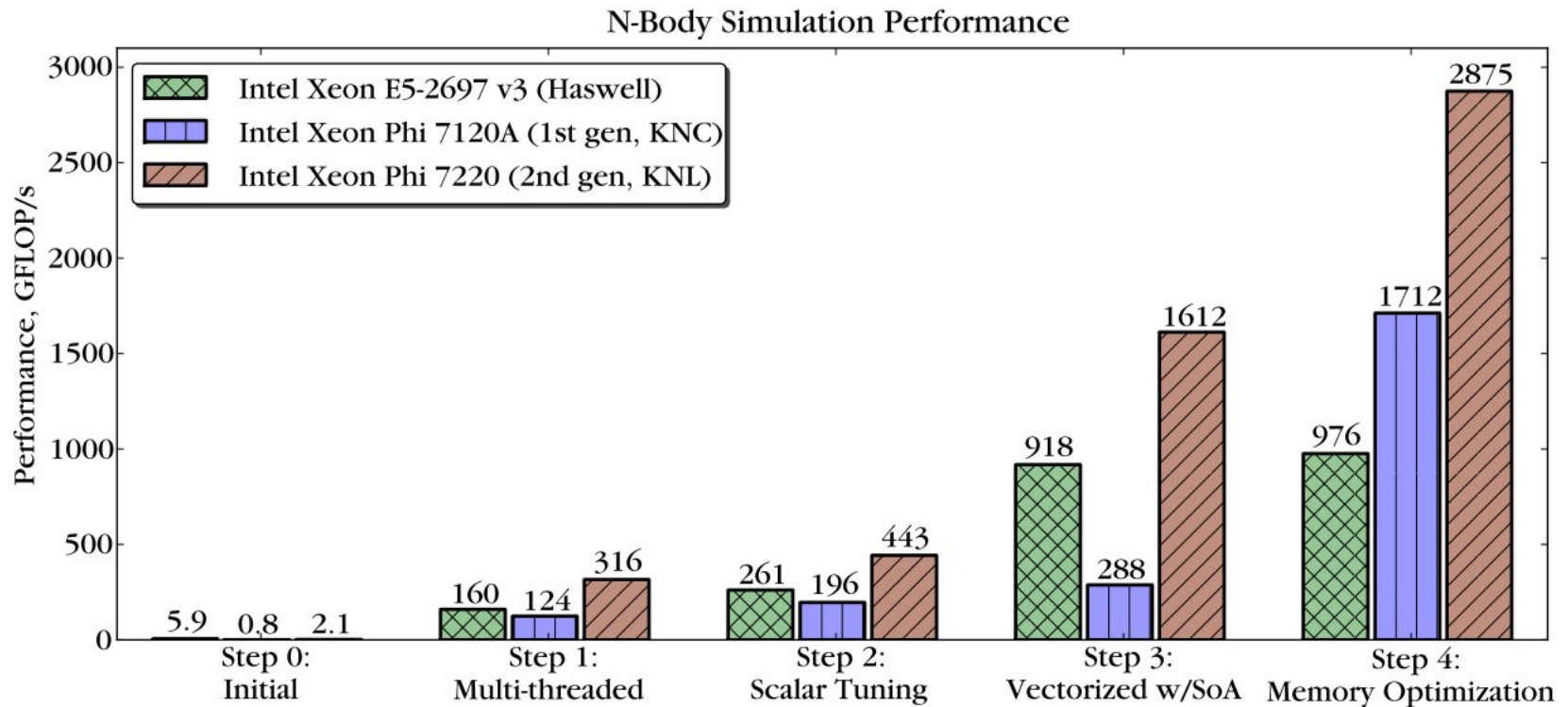


Node-level

Cluster-level

Tomado de: Fundamentals of Parallelism on Intel Architecture.
<https://es.coursera.org/learn/parallelism-ia>

Experiencia en Modernización de Código Común

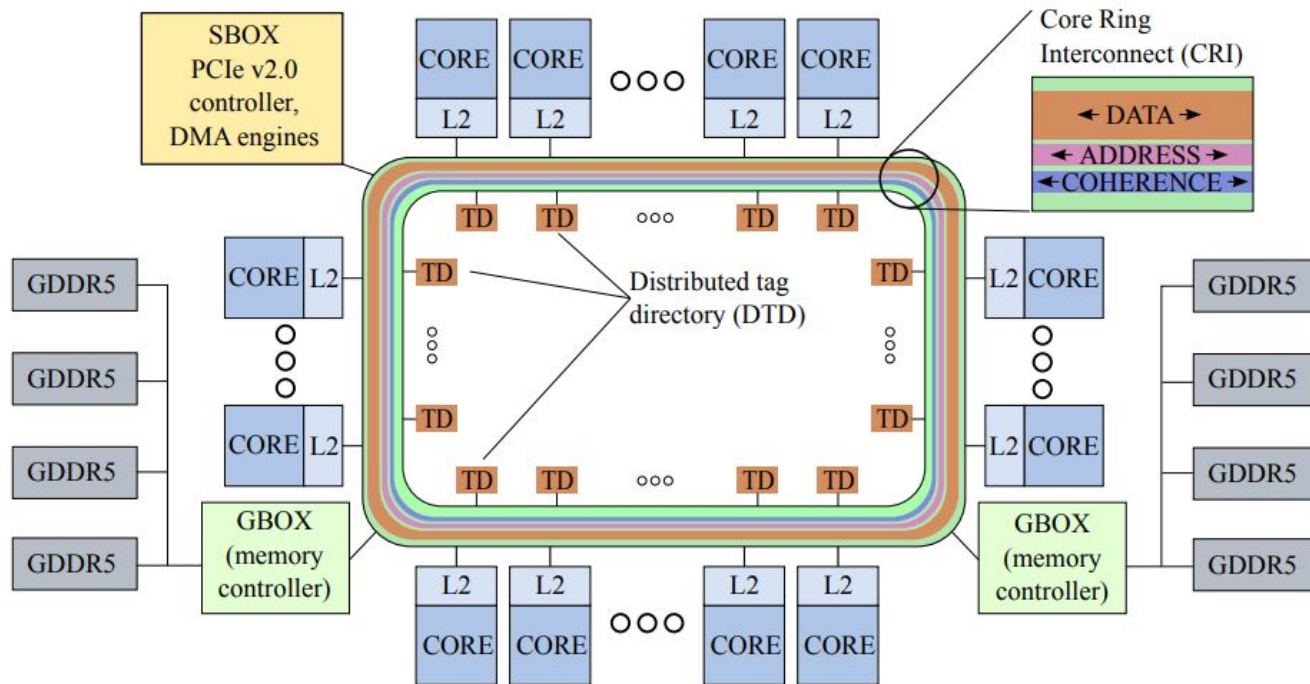


Tomado de: Fundamentals of Parallelism on Intel Architecture.

<https://es.coursera.org/learn/parallelism-ia>

Multihilos con OpenMP

Cores implement MIMD (Multiple Instruction Multiple Data) arch



Tomado de: Fundamentals of Parallelism on Intel Architecture.
<https://es.coursera.org/learn/parallelism-ia>

Vectorización con Compiladores Intel

Vectors – form Single Instruction Multiple Data (SIMD) architecture

Scalar Instructions

$$\begin{array}{r} 4 + 1 = 5 \\ 0 + 3 = 3 \\ -2 + 8 = 6 \\ 9 + -7 = 2 \end{array}$$

Vector Instructions

$$\begin{array}{r} 4 \\ 0 \\ -2 \\ 9 \end{array} + \begin{array}{r} 1 \\ 3 \\ 8 \\ -7 \end{array} = \begin{array}{r} 5 \\ 3 \\ 6 \\ 2 \end{array}$$

Vector Length

Tomado de: Fundamentals of Parallelism on Intel Architecture.
<https://es.coursera.org/learn/parallelism-ia>

Tráfico de memoria

Caches facilitate data re-use

RAM is optimized for streaming

Tiling

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

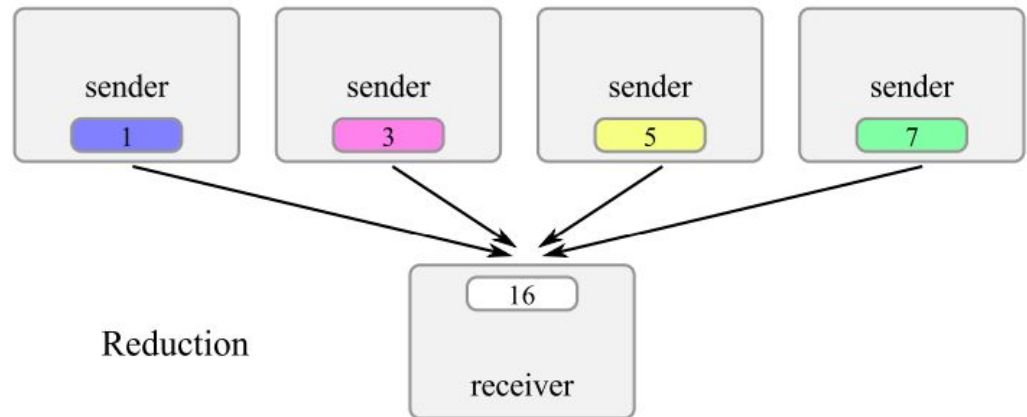
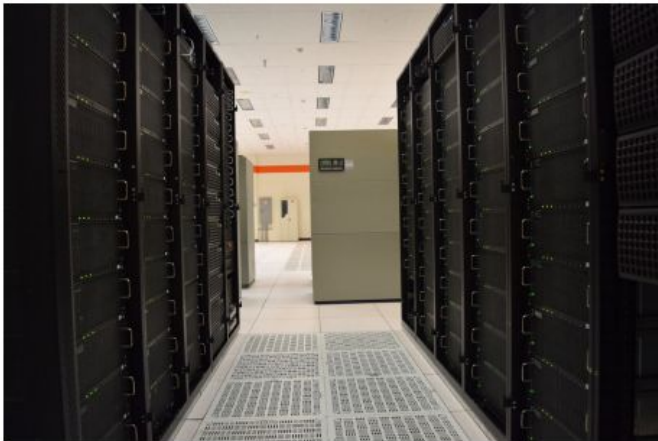
Cache-Oblivious Recursion

1	3	9	11
2	4	10	12
5	7	13	15
6	8	14	16

Tomado de: Fundamentals of Parallelism on Intel Architecture.
<https://es.coursera.org/learn/parallelism-ia>

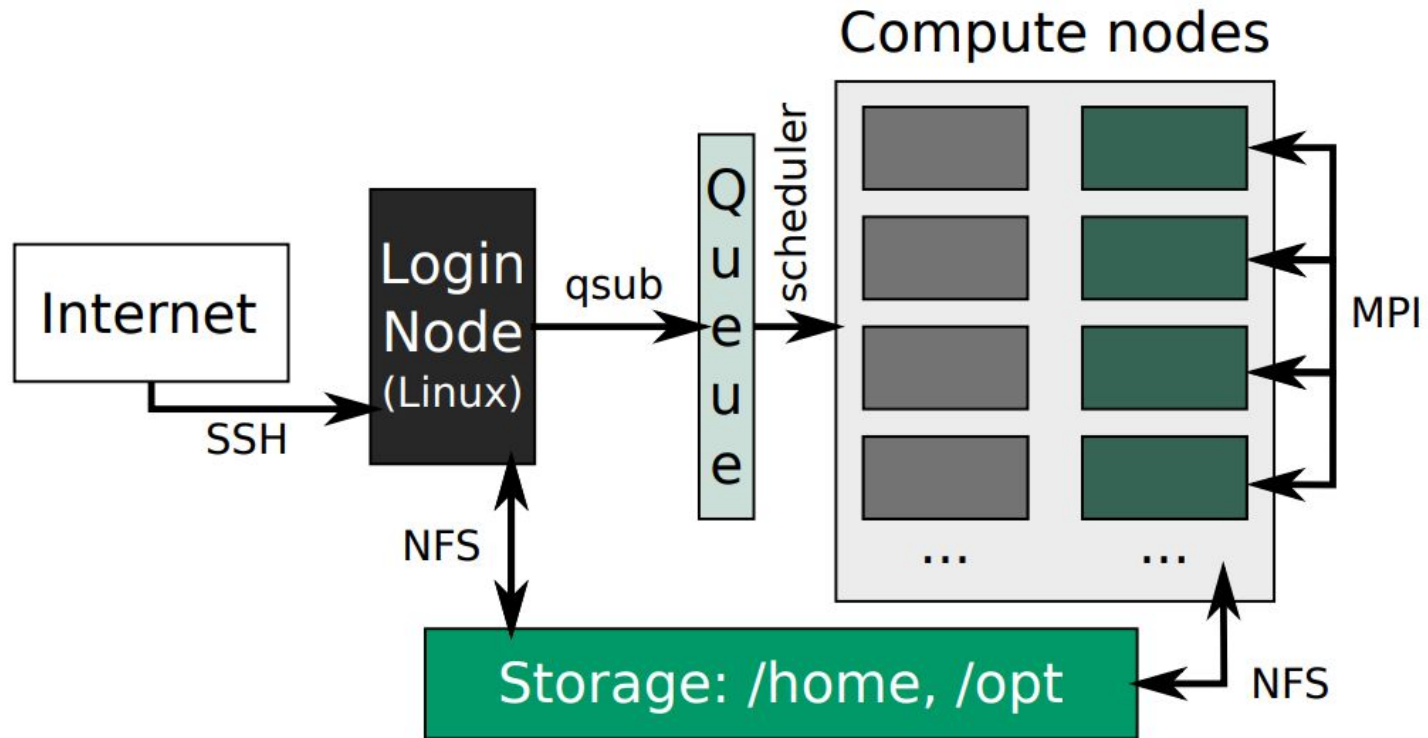
Clusters y MPI

Clusters form distributed-memory systems with network interconnects



Tomado de: Fundamentals of Parallelism on Intel Architecture.
<https://es.coursera.org/learn/parallelism-ia>

Intel en Apolo



Tomado de: Fundamentals of Parallelism on Intel Architecture.
<https://es.coursera.org/learn/parallelism-ia>

Opciones generales para optimización

- -O1
 - Optimizar tamaño del código, no vectorización
- -O2 - Por defecto
 - Inlining, vectorización
 - Optimizaciones básicas de ciclos
- -O3
 - Aplicaciones intensivas en ciclos
 - Incluye todas las de O2
 - Transformaciones de ciclos más agresiva

Tomado de: Optimization for Intel C++ and Fortran Compilers
<https://software.intel.com/en-us/articles/intel-c-compiler-introduction>

Optimización para Multi-Núcleo y Multi-procesamiento

- -parallel
 - Optimización Auto-paralelismo
- Lenguajes soportados para paralelismo explícito
 - OpenMP* 4.0 y la mayor parte de 4.5
 - Disponible para C/C++ y Fortran
 - Paralelismo basado en Pragmas/Directivas
 - Reducciones
- -xHost
 - Le dice al compilador que genere instrucciones para el conjunto de instrucciones más alto que esté disponible

Tomado de: Optimization for Intel C++ and Fortran Compilers
<https://software.intel.com/en-us/articles/intel-c-compiler-introduction>

Opciones específicas por procesador

- -m<target>
 - Le dice al compilador cuales características puede usar, incluyendo cuál conjunto de instrucciones puede generar según <target>
 - Targets:
 - SSE
 - SSE2 (default)
 - SSE3
 - SSSE3
 - SSE4.1
 - SSE4.2
 - AVX
 - AVX2
 - AVX512
 - IA32

Tomado de: Optimization for Intel C++ and Fortran Compilers
<https://software.intel.com/en-us/articles/intel-c-compiler-introduction>

Opciones específicas por procesador

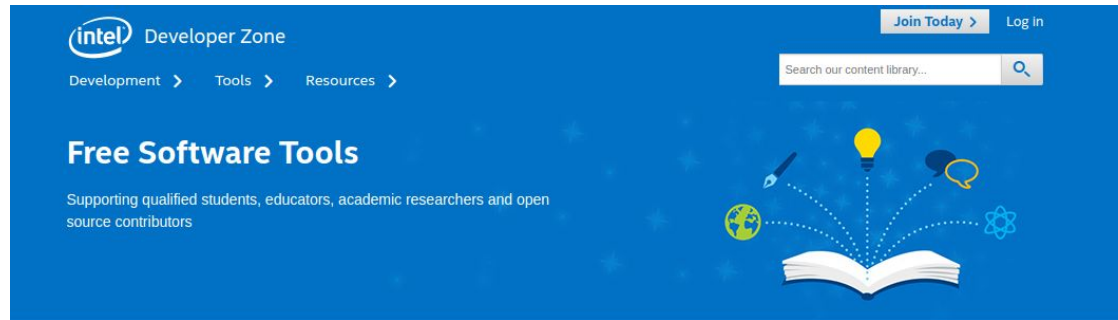
- `-x<target>`
 - Le dice al compilador cuales características puede usar, incluyendo cuál conjunto de instrucciones y optimizaciones puede generar según `<target>`
 - Targets:
 - COMMON-AVX512
 - MIC-AVX512
 - CORE-AVX512
 - CORE-AVX2
 - CORE-AVX-I
 - AVX
 - SSE4.2
 - SSE4.1
 - ATOM_SSE4.2
 - ATOM_SSSE4.2
 - ATOM_SSSE3
 - ...

Tomado de: Optimization for Intel C++ and Fortran Compilers
<https://software.intel.com/en-us/articles/intel-c-compiler-introduction>

Math Kernel Library

- Acelera rutinas matemáticas, incrementa el rendimiento de las aplicaciones y reduce el tiempo de desarrollo
- Librería lista para usar que incluye:
 - Álgebra Lineal
 - Transformadas rápidas de Fourier
 - Redes neuronales profundas
 - Estadística vectorial y ajuste de datos
 - Matemática vectorial
- Usa la API de C y Fortran para Compatibilidad con BLAS, LAPACK y FFTW. No se requieren cambios de código

¿Dónde Obtenerlo?



Free Tools for Students



Intel supports the next generation of scientists and engineers by ensuring that students have free access to the select Intel® Software Development Products. All products include user forum support.

[Get More Information](#)

Qualifications



- Full or part-time matriculated students that are currently enrolled at a degree-granting higher education institution and have a school email address.
- Students cannot be paid and/or compensated for software development. The products below are licensed to students for personal use to support their education, under the terms of the Non-Commercial License.

Student

[Educator](#) >

[Open Source Contributor](#) >

Quick Links

[User Support Forums](#) >

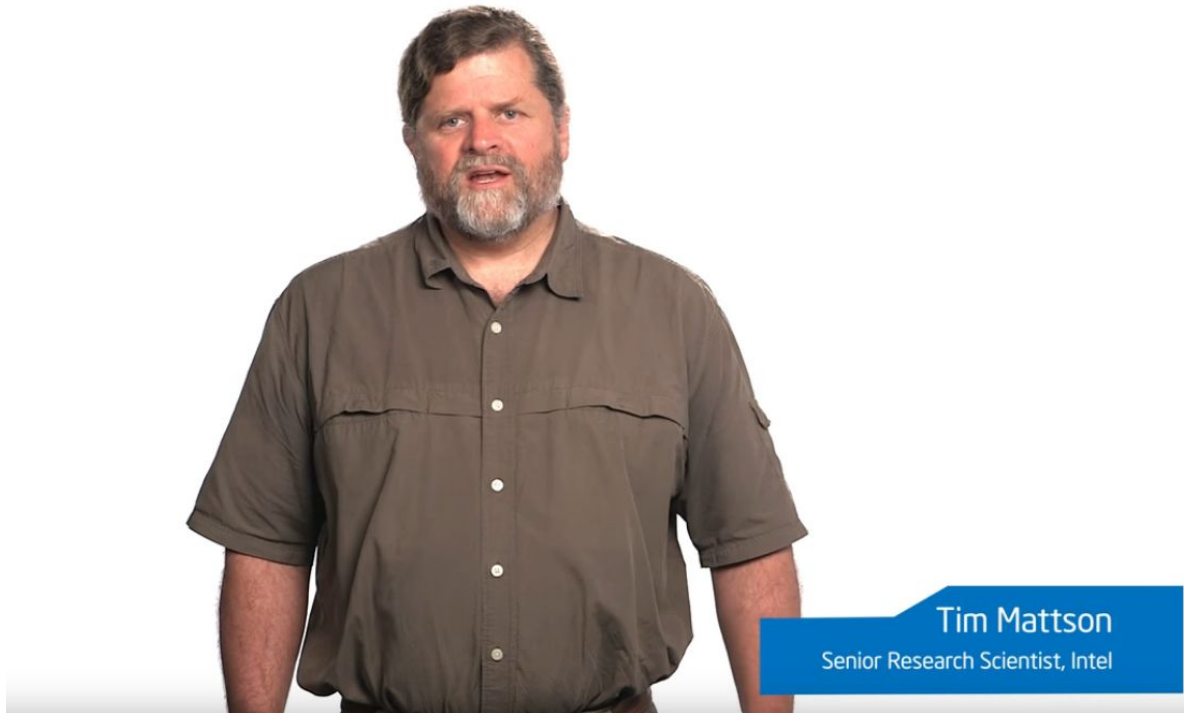
[Product Information](#) >

[Software End-User License](#) >

[Free Intel® Performance Libraries](#) >

- <https://software.intel.com/en-us/qualify-for-free-software/student>
- <https://software.intel.com/en-us/qualify-for-free-software/educator>

Recurso: Introducción a OpenMP



Tim Mattson

Senior Research Scientist, Intel

<https://youtu.be/nE-xN4Bf8XI>

Python Intel Distribution

ACCELERATE PYTHON* PERFORMANCE POWERED BY ANACONDA*

Supercharge applications and speed up core computational packages with this performance-oriented distribution.

Free Download



<https://software.intel.com/en-us/distribution-for-python>

Inspira Crea Transforma

UNIVERSIDAD
EAFIT[®]

Fundamentals of Parallelism on Intel Architecture

by Intel



Andrey Vladimirov



Welcome to Fundamentals of Parallelism on Intel Architecture! You're joining thousands of learners currently enrolled in the course. I'm excited to have you in the class and look forward to your contributions to the learning community.

To begin, I recommend taking a few minutes to explore the course site. Review the material we'll cover each week, and preview the assignments you'll need to complete to pass the course. Click **Discussions** to see forums where you can discuss the course material with fellow students taking the class.

If you have questions about course content, please post them in the forums to get help from others in the course community. For technical problems with the Coursera platform, visit the [Learner Help Center](#).

Good luck as you get started, and I hope you enjoy the course!

Less



<http://coursera.org/learn/parallelism-ia/>

Referencias y Recursos

- Compiler options: <https://software.intel.com/en-us/node/677967>
- Math Kernel Library: <https://software.intel.com/en-us/mkl>
- Fundamentals of Parallelism on Intel Architecture: <https://www.coursera.org/learn/parallelism-ia/home/welcome>
- Intel Software Development Products Samples and Tutorials: [https://software.intel.com/en-us/product-code-samples?field_software_product_tid\[\]=20813](https://software.intel.com/en-us/product-code-samples?field_software_product_tid[]=20813)
- Performance essentials using OpenMP* 4.0 vectorization with C/C++ <https://software.intel.com/en-us/videos/performance-essentials-using-openmp-40-vectorization>
- Vectorization Essentials: <https://software.intel.com/en-us/articles/vectorization-essential>
- Intel C++ Compiler Introduction: <https://software.intel.com/en-us/articles/intel-c-compiler-introduction>
- Tutorial: Using Auto Vectorization: <https://software.intel.com/en-us/intel-cplusplus-compiler-auto-vectorization-tutorial>
- Explor our documentation: https://software.intel.com/en-us/documentation?field_software_product_tid%5B0%5D=20813&field_software_product_tid%5B1%5D=20813&field_article_type_tid%5B0%5D=20783&field_article_type_tid%5B1%5D=20783
- Intel C++ Compiler in Intel Parallel Studio XE Support: <https://software.intel.com/en-us/c-compilers/ipsxe-support/training>