Mooc HPC: Fundamentals of Parallelism on Intel Architecture

Marco Antonio Franco Montoya mfranc18@eafit.edu.co

Este curso se enfoca en los aspectos básicos de la computación de alto rendimiento utilizando arquitecturas Intel (todos los ejemplos del curso son en estas plataformas) y a través de sus 5 semanas se analiza cómo se pueden optimizar códigos, más allá de una arquitectura específica, a través del uso de vectorización, hilos, comunicación entre múltiples nodos de cómputo, entre otros aspectos.

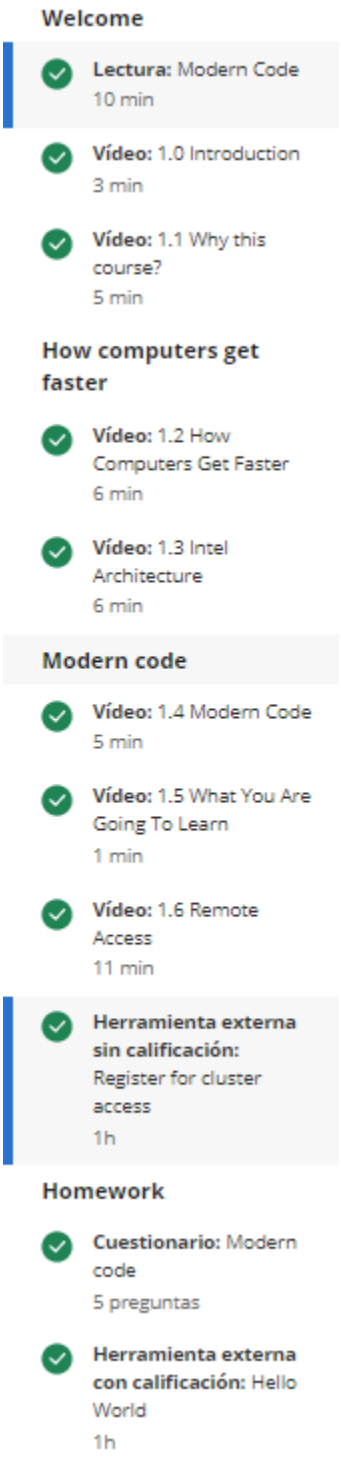
Los códigos ejecutados para este proceso se encuentran en el siguiente Repositorio

Contenido

Semana 1

En la primera semana se muestran múltiples aspectos de las arquitecturas Intel y su avance a través de los años, además de mostrar cómo se puede acceder al cluster de

COLFAX, utilizado para todas las actividades evaluadas. Se finaliza esta semana con un Hello World ejecutado de forma local y en un nodo.



Quiz:

Modern code

PUNTOS TOTALES DE 5

1. Intel Xeon Phi processors are a... 1 punto

- ( ) ...computing platform for legacy applications
- (•) ...more efficient computing platform than traditional CPUs for applications that have good thread and vector parallelism
- ( ) ...specialized family of processors for applications requiring high clock speeds

2. Multiple cores can share data in system memory 1 punto

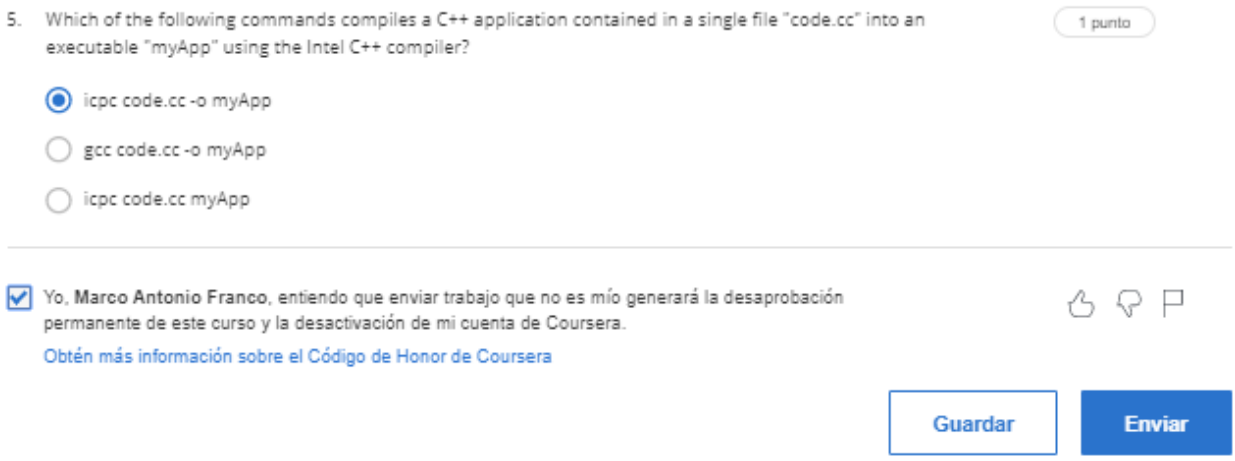
- (•) True
- ( ) False

3. In vector units, you can apply: 1 punto

- ( ) A single stream of instructions to a single data element in a vector
- ( ) Multiple streams of instructions to multiple data elements in a vector
- (•) A single stream of instructions to multiple data elements in a short vector

4. There are no mainstream processors with clock speeds above 4 GHz. What is the reason for that? (pick all answers that apply) 1 punto

- [x] Adding parallel processing capabilities to CPUs gives better performance per watt than increasing clock speeds
- [x] Increasing the clock speed would increase the power requirement, which make the required cooling solutions expensive and impractical
- [ ] Clock speed above 4GHz would result in slower computational speed



Semana 2

En la semana 2 se analizan los procesos de vectorización de un código, cómo a través de una instrucción se pueden procesar múltiples datos, en instancias como la suma de matrices, entre otras se encuentra su funcionalidad

Vector Arithmetics in Intel Architecture

- **Lectura:** Vectorization 10 min
- **Vídeo:** 2.1 Vector Operations 5 min
- **Vídeo:** 2.2 Vectorizing Your Code 5 min

Automatic Vectorization

- **Vídeo:** 2.3.1 Automatic Vectorization 5 min
- **Vídeo:** 2.3.2 Will This Vectorize? 15 min
- **Vídeo:** 2.4 Guided Automatic Vectorization 4 min

Demo: Stencil Kernel

- **Vídeo:** 2.8.1 Stencil Introduction 2 min
- **Vídeo:** 2.8 Stencil 9 min
- **Lectura:** Code Download 10 min

Designing Vectorizable Code

- **Vídeo:** 2.5 SIMD-Enabled Functions 3 min
- **Vídeo:** 2.6 Vector Dependence 5 min
- **Vídeo:** 2.7 Strip Mining 1 min

Demo: Numerical Integration

- **Vídeo:** Numerical Integration Introduction 2 min
- **Vídeo:** 2.9 Integral Vectorization 8 min
- **Lectura:** Code Download 10 min

Homework

- **Vídeo:** 2.10 Learn More 1 min
- **Cuestionario:** Vectorization 5 preguntas
- **Herramienta externa con calificación:** Vectorizing Monte-Carlo Diffusion 1h

Quiz

Vectorization

PUNTOS TOTALES DE 5

1. Which of the two loops is safe to be vectorized with AVX-512 vector instructions acting on 512-bit vector registers? By "safe" we mean that the vectorized loop will produce the same results as the scalar loop. Here A is an array of type "double". — 1 punto

   Code A:

```
1  for (int i = 0; i < n-1; i++)
2    A[i] += A[i+1];
```

   Code B:

```
1  for (int i = 4; i < n; i++)
2    A[i] += A[i-4];
```

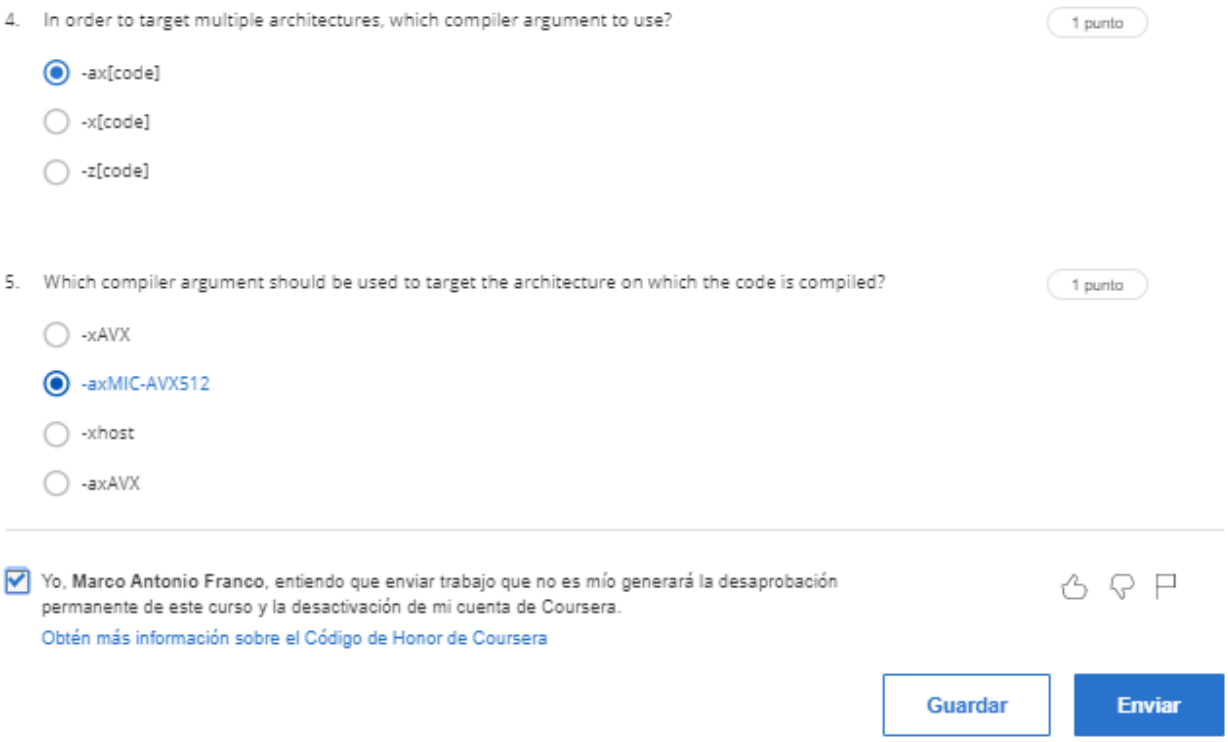
   - ○ Code B
   - ○ Both
   - ◉ Code A
   - ○ Neither

2. Which of the following cannot be automatically vectorized by the compiler? — 1 punto

   - ○ For loop
   - ◉ While loop

3. What compiler argument do you have to use with the Intel C++ compiler to produce an optimization report in Linux? — 1 punto

   - ○ /Qopt-report
   - ◉ -qopt-report
   - ○ -opt-report



Semana 3

La semana 3 se enfoca en OpenMP, sus usos y el tipo de instrucciones que se pueden utilizar para crear hilos dentro de una solución determinada. Se analizan las operaciones Mutex, los ciclos paralelos, etc.

Multiple Cores in Intel Architecture

- **Lectura:** Multithreading with OpenMP 10 min
- **Vídeo:** 3.1 Cores and Threads 3 min
- **Vídeo:** Demo: Forks 9 min
- **Lectura:** Code Download 10 min

OpenMP

- **Vídeo:** 3.2 Creating Threads 4 min
- **Vídeo:** 3.3 Variable Sharing 1 min
- **Vídeo:** 3.4 Parallel Loops 5 min
- **Vídeo:** 3.5 Data Races Mutexes 4 min
- **Vídeo:** 3.7 Parallel Reduction 4 min

Demo

- **Vídeo:** Stencil Introduction 2 min
- **Vídeo:** Stencil Demonstration 3 min
- **Lectura:** Code Download 10 min

Homework

- **Vídeo:** Learn More 1 min
- **Cuestionario:** OpenMP 7 preguntas
- **Herramienta externa con calificación:** Multithreaded Filtering 1h

Quiz

OpenMP

PUNTOS TOTALES DE 7

1. OpenMP is a framework for... 1 punto

   - ○ multiprocessing in distributed-memory systems (clusters)
   - ◉ multithreading and vectorization in shared-memory systems

2. Which flag is required to compile your OpenMP application with Intel compiler? 1 punto

   - ○ -fopenmp
   - ○ -qopt-report
   - ○ -xhost
   - ◉ -qopenmp
   - ○ -openmp

3. Based on this snippet, which of the following is correct about the number of copies of a variable in memory at runtime if we are using 4 threads? 1 punto

```
1  int A, B;
2  #pragma omp parallel private(B)
3  {
4    int C;
5    //code to be executed
6    }
```

   - ◉ There is a single copy of A and there are 4 copies of both B and C
   - ○ There is a single copy of A and B and there are 4 copies of C
   - ○ There are 4 copies of A, B and C

4. Which of these mutexes has the highest performance penalty? 1 punto

- (●) #pragma omp critical
- ( ) #pragma omp parallel
- ( ) #pragma omp atomic

5. What is the default number of OpenMP threads in an application running on an Intel Xeon Phi processor 7210 with 4-way hyper-threading? Feel free to look up the technical specifications of this processor model online 1 punto

- ( ) 64
- ( ) 1
- ( ) 192
- ( ) 128
- (●) 256

6. Which of the following code snippets has a data race ("race condition"): 1 punto

Code A:

```
#pragma omp parallel for

for (int i = 0; i < n-1; i++)

A[i] += A[i+1];
```

Code B:

```
#pragma omp parallel for

for (int i = 1; i < n; i++)

A[i] += A[i-1];
```

- ( ) Code B
- (●) Code A
- ( ) Both

7. Consider the following piece of code for computing (incorrectly?) the mean and standard deviation of values in an array: 1 punto

```
void ComputeStats(int n, double* x, double & mean, double & stdev) {
  mean = 0.0;
  stdev = 0.0;

  #pragma omp parallel for

  for (int i = 0; i < n; i++) {
    mean += x[i];
    stdev += x[i]*x[i];

  }

  mean /= double(n);
  stdev = sqrt(stdev/double(n) - mean*mean);
}
```

What should you change in this code to make it correct and efficient?

- (●) Add a reduction clause in the parallel pragma
- ( ) Add a #pragma omp critical clause
- ( ) Add a #pragma omp atomic clause

☑ Yo, **Marco Antonio Franco**, entiendo que enviar trabajo que no es mío generará la desaprobación permanente de este curso y la desactivación de mi cuenta de Coursera.

Obtén más información sobre el Código de Honor de Coursera

Guardar Enviar

Semana 4

En la semana 4 se habla sobre el acceso a la memoria por parte del procesador, y cómo esta puede ser optimizada dependiendo del caso de uso. En el ejemplo se muestra cómo para un escenario óptimo se pueden distribuir los datos para evitar múltiples lecturas y así suprimir penalidades al tener que leer los datos directamente desde su origen.

Memory Organization in Intel Architecture

- **Lectura:** Memory Traffic 10 min
- **Video:** 4.1 Cheap Flops 6 min
- **Video:** 4.2 Memory Hierarchy 7 min

High-Bandwidth Memory

- **Video:** 4.3 High Bandwidth Memory 3 min
- **Video:** 4.4 Memory Allocation 1 min
- **Video:** Stencil Introduction 2 min
- **Video:** Demo: Stencil with numactl 3 min
- **Video:** Demo: Stencil with Memkind 3 min
- **Video:** 4.5 Bypassing Caches 2 min
- **Video:** Stencil Demonstration-Nontemporal 1 min
- **Video:** Stencil Demonstration-Char 3 min
- **Lectura:** Code Download 10 min

Caches and Data Locality

- **Video:** 4.6 Locality in Space 7 min
- **Video:** 4.7 Locality in Time 8 min

Practical Demonstration- Numerical Integration

- **Video:** Integral Introduction 2 min
- **Video:** Integral Demonstration 28 seg
- **Lectura:** Code Download 10 min

Homework

- **Cuestionario:** Memory traffic 8 preguntas
- **Herramienta externa con calificación:** Batch FFTs in HBM 1h

Quiz

Memory traffic

PUNTOS TOTALES DE 8

1. In KNL memory architecture, Cache mode for MCDRAM means: — 1 punto

   - (●) MCDRAM acts as a cache between L2 cache and DDR4 RAM
   - ( ) MCDRAM acts as a cache between L1 cache and L2 cache
   - ( ) DDR4 RAM acts as a cache between MCDRAM and L2 cache

2. Which memory has the lowest access latency in KNL memory organization? — 1 punto

   - (●) L1 cache
   - ( ) L2 cache
   - ( ) MCDRAM
   - ( ) DDR4 RAM

3. What is the flat mode in KNL memory architecture? — 1 punto

   - (●) MCDRAM is treated as a separate NUMA node
   - ( ) DDR4 is treated as LLC
   - ( ) MCDRAM is treated as LLC

4. If your application is bandwidth-limited and requires less than 16 GB of memory, which is the easiest and the most efficient way of running it in high-bandwidth memory (HBM) of an Intel Xeon Phi processor? — 1 punto

   - (●) Run the whole program in the HBM using numactl
   - ( ) Allocate bandwidth-critical data in HBM using memkind library
   - ( ) Set the processor in the cache mode and run the application without modification

5. If your bandwidth-limited application requires more than 16 GB of memory, what are the options that can be used to run your application in the high-bandwidth memory of an Intel Xeon Phi processor? — 1 punto

   - [ ] Use numactl
   - [x] Use cache mode
   - [x] Use memkind library

6. How to make the Intel compiler implement streaming stores in a loop? — 1 punto

   - [x] -qopt-streaming-stores=always
   - [x] #pragma vector nontemporal
   - [ ] #pragma vector temporal

6. How to make the Intel compiler implement streaming stores in a loop? 1 punto

- [x] -qopt-streaming-stores=always
- [x] #pragma vector nontemporal
- [ ] #pragma vector temporal

7. A gas simulation needs to detect collisions between ball-shaped gas particles. The simulation runs on an Intel Xeon processor with 256-bit vector units. Your goal is to achieve the optimum performance in such a simulation. With this in mind, rank the following options for data structures from best to worst. 1 punto

Case A:

```
struct Ball {
  // Coordinates of the center of the ball:
  float x, y, z;
  float R;
  // Radius of the ball
};

Ball b[1000];
```

Case B:

```
class AllBalls {
  // Coordinates of the centers of the balls:
  float x[1000], y[1000], z[1000];
  float R[1000];
  // Radii of the balls
};

AllBalls b;
```

Case C:

```
struct BallVector {
  float x[8], y[8], z[8];
  float R[8];
};

BallVector b[125];
```

- (•) B is the best, A is the worst
- ( ) A is the best, B is the worst
- ( ) C is the best, A is the worst

8. When an application reads and updates a single 32-bit floating-point number residing in the main memory, how much data is physically sent from the main memory to the core? 1 punto

- (•) 64 bytes
- ( ) 32 bytes
- ( ) 16 bytes

[x] Yo, **Marco Antonio Franco**, entiendo que enviar trabajo que no es mío generará la desaprobación permanente de este curso y la desactivación de mi cuenta de Coursera.

Obtén más información sobre el Código de Honor de Coursera

Guardar Enviar

Semana 5

En la semana 5 se revisa la clusterización y los retos que representa. Cuando múltiples nodos están ejecutando aspectos del mismo programa, estos deben comunicarse para segmentar de forma efectiva los datos, además de facilitar la información requerida por otros nodos. Para funciones del curso se usa la biblioteca MPI, específicamente con el compilador de Intel para ejecutar dichos procesos de comunicación. Se advierte que se debe tener cuidado al ejecutar estos procesos puesto que por defecto son operaciones bloqueantes.

(dado que se completó al tercer intento esta evaluación, no permite mostrar las respuestas dadas)

MPI

CALIFICACIÓN DEL ÚLTIMO ENVÍO

100%

1. In distributed-memory systems, the _____ framework can be used for communicating data between processes. — 1 / 1 punto

✓ Correcto

2. The function MPI_Comm_size returns the _____ — 1 / 1 punto

✓ Correcto

3. The MPI function that sends data from a group of processes to one process is____ — 1 / 1 punto

✓ Correcto

4. In a hybrid OpenMP+MPI application for a modern multicore processor, how do you manage the code of processes and threads? (choose the best answer) — 1 / 1 punto

✓ Correcto

5. In this snippet, 2 processes are communicating, one is sending a message and the other is receiving it, in order to prevent this application from blocking, what are the correct values of DESTINATION and SOURCE ? — 1 / 1 punto

```
if (myid == 1)
{
  buffer="Hello world";
  MPI_Send(&buffer, count, MPI_INT, DESTINATION, tag, MPI_COMM_WORLD);
  printf("processor %d sent %d integers\n",myid,count);
}
if (myid == 2)
{
  MPI_Recv(&buffer, count, MPI_INT, SOURCE, tag, MPI_COMM_WORLD, &status);
  printf("processor %d received %d integers\n",myid,count);
}
```

✓ Correcto

Certificado y conclusiones

A través de este curso se lograron obtener las nociones básicas para implementar soluciones flexibles que permitan el uso optimizado de procesadores diseñados para la interacción con datos vectorizados, la comunicación entre nodos de estos

procesadores, entre otros aspectos. Sin mencionar el uso de las bibliotecas OpenMP y MPI para la paralelización interna y entre nodos de la solución creada.

Al finalizar el curso se puede obtener el siguiente certificado