Computación de Altas Prestaciones 2023

Intel OneApi

OneApi (Parallel Studio)

- El paquete OneApi (antes Parallel Studio) de Intel es un paquete que incluye compiladores, herramientas de depuración, profilers, asistentes para Optimización, librerías optimizadas para cálculos de muchos tipos, etc.
- La versión actual es de libre acceso tanto para Windows (por primera vez) como para Linux
- Existen varias versiones, versión "normal" o edición "cluster"



Librerías para paralelización

- Todos los compiladores (C, C++, Fortran) soportan OpenMP
- La versión cluster lleva una librería MPI de alto rendimiento
- Hay varias librerías de programación paralela de propósito general: TBB (Threading Building Blocks), IPP (Intel Primitives..)
- Intel MKL : Librería de funciones matemáticas paralelizadas, incluye Blas, Lapack, Pardiso, FFts, etc.
- Distribución Python incluida



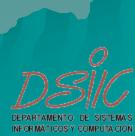
Herramientas

- Depurador gdb; versión para computación heterogenea (Xeon Phi, GPUs)
- Intel Vtune amplifier (profiler)
- ◆ Intel Trace Analyzer and Collector
- Intel Advisor
- Intel Inspector (depurador de memoria y de threads)
- ◆ Intel Cluster Checker



Librerías específicas

- Librería daal (Data analytics)
- ◆ IPP para criptografía



Threading Building Blocks

- ◆ Intel® Threading Building Blocks (Intel® TBB) is a runtime-based parallel programming model for C++ code that uses threads. It consists of a template-based runtime library to help you harness the latent performance of multicore processors. Use Intel TBB to write scalable applications that:
- Specify logical parallel structure instead of threads.
- Emphasize data parallel programming
- Take advantage of concurrent collections and parallel algorithms
- Incluye parallel STL (Standard Template Library; https://es.wikipedia.org/wiki/Standard_Template_Library); https://software.intel.com/en-us/get-started-with-pstl
- https://software.intel.com/en-us/tbb-user-guide
- Tutorial: https://software.intel.com/en-us/tbb-tutorial



Ejemplo

```
void SerialApplyFoo( float a[], size_t n ) {
   for(size_t i=0; i!=n; ++i)
       Foo(a[i]);
                                     Versión con tbb
#include "tbb/tbb.h"
using namespace tbb;
class ApplyFoo {
 float *const my a;
public:
 void operator()( const blocked_range<size_t>& r ) const {
   float *a = my a;
   for( size t i=r.begin(); i!=r.end(); ++i )
     Foo(a[i]);
 ApplyFoo(float a[]):
   my a(a)
void ParallelApplyFoo( float a[], size_t n ) {
   parallel_for(blocked_range<size_t>(0,n), ApplyFoo(a));
```



Intel® Integrated Performance Primitives

Intel® Integrated Performance Primitives (Intel® IPP) is a software library that provides a broad range of functionality, including general signal and image processing, computer vision, data compression, and string manipulation.

Tutorial: https://software.intel.com/en-us/ipp-tutorial

Referencia: https://software.intel.com/en-us/ipp-dev-reference



Intel® Data Analytics Acceleration Library

Intel® Data Analytics Acceleration Library (Intel® DAAL) is the library of Intel® architecture optimized building blocks covering all stages of data analytics: data acquisition from a data source, preprocessing, transformation, data mining, modeling, validation, and decision making.

https://software.intel.com/en-us/getstarted-with-daal-for-linux



Intel® Vtune

Profiler de Intel; Soporta programas paralelos, también en memoria distribuida

https://software.intel.com/en-us/articles/intel-vtune-amplifier-tutorials

Para alguno de estos tutoriales (hardware) hay que hacer una instalación específica como root.

Es interesante hacer el primer tutorial, y luego repetir el proceso usando el programa de producto de matrices paralelizado con OpenMp

Arrancar vtune (versión gráfica): /opt/intel/vtune_amplifier_xe/bin64/vtune-gui



Intel® Advisor

Herramienta que analiza el código del usuario y aconseja sobre como paralelizar, vectorizar, y como usar computación heterogenea (coprocesadores Xeon Phi, GPUs)

Arrancar inspector (versión gráfica): /opt/intel/advisor/bin64/advixe-gui



Intel® Advisor; Tutoriales:

https://software.intel.com/enus/articles/advisor-tutorials



Intel® Inspector

Herramienta para localizar errores de memoria (memory leaks, accesos indebidos, ...) y de threading (race conditions) en programas C, C++, Fortran

1) Arrancar inspector (versión gráfica): /opt/intel/inspector/bin64/inspxe-gui



Intel® Inspector; Tutoriales:

https://software.intel.com/en-us/articles/inspectortutorials

Similar a valgrind. (visto en seminario anterior)

Los programas que se inspeccionan bajo Inspector o bajo valgrind se ralentizan mucho. Si puede ser, no lanzar casos muy grandes.



Videos tutoriales en castellano

https://www.danysoft.com/webinar-intel-software-2019/

- Introducción a parallel studio
- 2. Intel compiler HPClab
- 3. Intel Advisor
- 4. Vtune
- 5. Introducción a herramientas Intel para IA
- Demo de herramientas Intel para IA

