

UNIVERSIDAD NACIONAL AUTÓNOMA DE HONDURAS
Departamento de Matemática
Sílabo de Programación de Alto Rendimiento (MMM-680)

Información del Curso

Código: MMM-680

Créditos: 4

Total de horas: 60

Requisitos: MMM-670

Aula: 302 edif. 1847

1 Objetivos de Aprendizaje

Al terminar el semestre el estudiante será capaz de:

1. Entender los principios fundamentales de las arquitecturas paralelas.
2. Diseñar algoritmos paralelos para resolver problemas prácticos.
3. Diseñar programas usando las librerías OpenMP, MPI4PY y MPI.
4. Manejar las librerías más utilizadas para el procesamiento numérico de problemas de gran escala: BLAS, LAPACK, ARPACK.
5. Desarrollar e implementar algoritmos eficientes utilizando la librería STL.

Contenido

1. Principios de arquitecturas paralelas y distribuidas.
2. Ejemplos de algoritmos paralelos notables:
 - Optimización por SGD
 - Solución numérica de la ecuación de Laplace $\Delta f = 0$.
3. Ejemplos importantes de Programación de Alto Desempeño:
 - <https://github.com/tmikolov/word2vec>
 - <https://github.com/inneme/pcg-c>
4. Usando *parallel* desde de la linea de comando unix.
5. Usando *threads* en C/C++ y Python.
6. Llamando código de alto desempeño desde Python.
7. Programación paralela en usando el MPI.

8. Uso de MPI en Python usando mpi4py.
9. Utilizando Contenedores Docker.
10. Uso de rutinas asincronas: async en Python.
11. Análisis y diseño de algoritmos paralelos.
12. Librerías numéricas BLAS STL y LAPACK.
13. Librerías numérica CUDA: ejemplo de generacion de numeros aleatorios en GPU.
14. Optimización y depuración de código secuencial con *gprof*.
15. Optimización y depuración de código paralelo con *mpiP*.
16. Comparación con lenguajes de programación funcionales, LEAN y Haskell.

2 Calendarización de Clases

3 Criterios de Evaluación

3.1 Tareas

Se asignará una tarea corta semanalmente, las soluciones se discutirán los días jueves de cada semana. (60%)

3.2 Exámenes

Los exámenes son problemas para resolver en casa usando el contenido visto en clase:

- Jueves 26 de junio. (10%)
- Jueves 31 de julio. (10%)

3.3 Proyecto Final

El estudiante deberá proponer un proyecto de programación que involucre su tema de investigación que utilice herramientas de programación de alto desempeño. Se evaluará tres etapas del proyecto: Propuesta, Avance y presentación Final. (20%)

4 Referencias

Bibliografía

- Eijkhout, Victor. Introduction to high performance scientific computing. Lulu. com, 2010. APA
- Anderson, E.; Bai, Z; Bischof, C; Blackford, S.; Demmel, J.; Dongarra, J.; Du Croz, J.; Greenbaum, A.; Hammarling, S.; McKenney, A. y Sorensen, D. LA-PACK Users' Guide, Third Edition, Society of Industrial and Applied Mathematics, 1999.
- Chapman, B.; Jost, G. y Van der Pas, R. Using OpenMP: Portable Shared Memory Parallel Programming, MIT Press, 2007.
- Dongarra, J.; Foster, I.; Fox, G.; Gropp, W.; Kennedy, K.; Torczon, L. y White, A. (eds) Sourcebook of Parallel Programming, Morgan Kaufmann, 2002.
- Jordan, H. y Alagband, G. Fundamentals of Parallel Processing, Prentice Hall, 2002.
- Josuttis, N.M. The C++ Standard Library: A Tutorial and Reference, Addison-Wesley, 2008.
- Karniadakis, G. y Kirby II, R. Parallel Scientific Computing in C++ and MPI, Cambridge University Press, 2003.