Meta Volante 1

David Calle Gonzalez
dept. of science
EAFIT
Medellín, Colombia
dcalleg@eafit.edu.co

Santiago Gil Zapata
dept. of science
EAFIT
Medellín, Colombia
sgilz@eafit.edu.co

Sebastian Obando
dept. of science
EAM
Medellín, Colombia
sebastian.obando.8888@eam.edu.co

Juan Manuel Young Hoyos
dept. of science
EAFIT
Medellín, Colombia
jmyoungh@eafit.edu.co

***Abstract*—El presente documento tiene por objetivo demostrar-los procesos necesarios para poder ejecutar HPL en nuestro cluster de 2 nodos en Cronos.**

***Index Terms*—HPC, HPL, MPI.**

I. Introducción

La idea del proyecto es mostrar cómo logramos una eficiencia de 45.28% **usando sólo 16 cores**.

II. Objetivos

Lograr una eficiencia entre 35% y 45% usando solo 16 cores y los 2 nodos de Cronos que tenemos a nuestra disposición.

III. Ambiente de ejecución

En este apartado se mostrará qué se ha usado para realizar estas pruebas.

A. Hardware

- **Cantidad de nodos:** 2.
- **Procesadores por nodo:** Intel Xeon E5-2670 0 (16) @ 2.989GHz.
- **Controlador Ethernet:** Intel Corporation I350 Gi-gabit Network Connection.
- **Memoria por nodo:** 16 x 4GB DIMM DDR3 1333MT/s.

B. Software

- **HPL:** 2.3 [1].
- **Sistema Operativo:** CentOS Linux 8 (Core) $\times 86_6 4$.
- **BLAS:** 3.10.0 [2].
- **MPI:** icc (ICC) 2021.2.0 20210228.
- **Compiler:** icc (ICC) 2021.2.0 20210228.
- **NFSv3**.

IV. HPL Heaven

La definición de que tenemos de este concepto abarca varios razonamientos que hace complicado dar una definición concreta de este mismo. De una forma más acertada y general podríamos decir que "Heaven" hace referencia a el valor teórico más optimo al que podríamos llegar; sin embargo, este valor en la práctica es imposible de alcanzar por lo que siempre intentamos tener algo lo más cercano porsible a este mismo.

V. Optimizaciones de HPL

A. HPL.dat

Los valores de parametros que se encuentran definidos en *HPL.dat* tiene un alto impacto en el performace del mismo, especialmente 3 de estos que son:

1) El tamaño de la matriz de nuestro problema (N): Este valor debe seleccionarse basados en el sistema de hardware y software que se tenga; Se debe decidir teniendo en cuenta la eficiencia computacional y la capacidad en memoria. A mayor el tamaño de la matriz, mayor serán los Flops.
2) El tamaño del bloque (NB): Este valor es usualmente determinado por medio de experimentación. Sin embargo, debe de ser cuidadosamente seleccionado para no ser ni muy grande ni muy pequeño, suelen ser números menores a 512 y normalmente multiplos de 64.
3) La matriz de procesos bidimencional (P X Q): El resultado del producto debe ser equivalente al total del número de procesos asignados en el parametro -n al ejecutar hpl. Como recomendación general se tendría a $P < Q$. Donde P debería de tomar el valor más pequeño que sea posible.

B. Make.Linux_Centos

En el archivo adjunto llamado *Make.Linux_Centos* añadimos algunos *flgas* como:

- **-xHost**
- **-Ofast**
- **-ansi-alias**
- **-z noexecstack**

VI. Dependencias

Se realizó intalación de:

- **NFS (Network File System:)** Posibilita que distintos sistemas conectados a una misma red accedan a ficheros remotos como si se tratara de locales.
- **SLURM:** Es un sistema de programación de trabajos y administración de clústeres de código abierto, tolerante a fallas y altamente escalable para clústeres de Linux grandes y pequeños.
- **Infiniband:** Es un bus de comunicaciones serie de alta velocidad, baja latencia y de baja sobrecarga de CPU, diseñado tanto para conexiones internas como externas.

Sus especificaciones son desarrolladas y mantenidas por la Infiniband Trade Association.

- **MPI:** Es un estándar que define la sintaxis y la semántica de las funciones contenidas en una biblioteca de paso de mensajes diseñada para ser usada en programas que exploten la existencia de múltiples procesadores. Escritos generalmente C, C++, Fortran y Ada.
- **BLAS:** Basic Linear Algebra Subprograms. Es una especificación que define un conjunto de rutinas de bajo nivel para realizar operaciones comunes de álgebra lineal tales como la suma de vectores, multiplicación escalar, producto escalar, combinaciones lineales y multiplicación de matrices.
- **HPL:** is a High-Performance Linpack benchmark implementation. The code solves a uniformely random system of linear equations and reports time and floating-point execution rate using a standard formula for operation count.

VII. Resultados

Para nuestro sistema, tenemos el siguiente R_{peak}:

$$R_{peak} = 2 \times 2 \times 8 \times 2.6 \times 10^9 \times 8 = 665.6\ GFLOP/s$$

Usando solo **16 *cores*** logramos $3.0145e+02$ Gflops, por lo que nuestra eficiencia fue de:

$$Eficiencia = \frac{301.45}{665.6} * 100\% = 45.2899639423\%$$

References

[1] J. D. A. C. P. L. Antoine Petitet, Clint Whaley, "Hpl 2.3 - aportable implementation of the high-performance linpack." [Online]. Available at: http://www.netlib.org/benchmark/hpl/software.html.

[2] [Online]. Available at: http://www.netlib.org/blas/.