

# Meta volante #1

Jhon Jairo Chavarria Gaviria  
Juan Manuel Villanueva Aristizabal  
Julián David Ramírez Lopera  
Santiago Alzate Cardona

## Introducción

Este documento tiene como objetivo mostrar cuáles fueron los puntos más importantes que nos llevaron a obtener los mejores resultados en los benchmark de hpl.

### 1. Entorno de ejecución:

Para poder realizar la configuración del HPL la primera pregunta que nos realizamos fue: ¿Qué características físicas tienen los nodos en los que vamos a ejecutar las prueba de HPL? Para dar solución a esta pregunta miramos nuestro entorno de ejecución y buscamos entenderlo para poder llegar al mejor performance posible, las características del hardware eran:

**Cantidad de nodos:** 2

**Procesadores:** Cada nodo posee 2 procesadores **Intel Xeon CPU. E5-2670**, de 8 núcleos y 2.6GHz, para dar un total de 16 núcleos por nodo y 32 núcleos para ambos nodos.

**Controlador Infinidad:** El *Network Controller* que poseen los nodos es **Mellanox Technologies MT27500 Family [ConectX-3]**

**Memoria:** Por cada nodo se poseen 16 módulos de memoria de **4GB DIMM DDR3 1333MT/s**

### 2. RPEAK

$$\begin{aligned} R_{peak} &= \#nodos \times \frac{\#cpus}{nodo} \times \frac{\#nucleos}{cpu} \times \frac{\#ciclos}{segundo} \times \frac{\#Flops}{ciclo} \\ R_{peak} &= 2 \times 2 \times 8 \times 2.6e9 \times 8 \\ R_{peak} &= 665600000000 \text{ Flops} \\ R_{peak} &= 6.656e + 02 \text{ GFlops} \end{aligned}$$

### 3. Instalación de NFS

Instalamos NFS porque nos permite acceder a los mismos datos optimizando los datos de almacenamiento debido a que podíamos compartir así archivos por red. Así también ambos cluster podrían acceder a una misma instalación de un software. Decidimos montar un directorio compartido al que llamamos **/shared** donde instalamos todas las aplicaciones.

## 4. Selección de Software:

Una vez entendimos el entorno de ejecución que poseíamos procedimos a realizar la selección de las implementaciones que usamos de las dependencias de **HPL**. Estas dependencias eran alguna implementación de **MPI** y otra de **BLAS** o **VSIPL**.

Al buscar diferentes implementaciones de **MPI** como **IMPI**, **OPENMPI** y **MPICH** llegamos a la conclusión de que la mejor implementación para nuestro cluster sería la de **IMPI** (MPI de Intel) debido a la naturaleza de los procesadores, y esta implementación podría suponer una mayor optimización en las instrucciones para los procesadores.

Con implementaciones de **BLAS** encontramos dos que nos llamaron la atención, **OPENBLAS** e Intel **MKL** (Intel Math kernel Library) la cual incluía implementación de funciones matemáticas de **BLAS**. Es por esto que decidimos utilizar las 2 implementaciones para comparar el performance.

## 5. Instalación de dependencias:

### 5.1 Instalación de Intel MPI

Tras esto decidimos seguir con la descarga e instalación del **MPI**, en este caso optamos por la implementación de **MPI de Intel**, ya que nuestros procesadores eran de esta misma empresa y llegamos a la conclusión de que esta implementación vendría mejor optimizada para estos procesadores, llegando a un mejor performance. En cuanto a la versión, optamos por la última versión estable al estar lo más actualizada y libre de errores posible.

### 5.2 Instalación de BLAS - MKL

Tras la instalación del **Intel MPI** seguimos con la descarga e instalación de la librería que íbamos a usar para las subrutinas matemáticas. La primera opción por la que nos decantamos fue por **OPENBLAS** y su versión 0.3.17, ya que era la recomendada. Pero no solo nos quedamos con esta librería, también instalamos la versión de **MKL** que venía en el **Intel OneApi Base**, ya que en este paquete venían las versiones compatibles con el **IMPI** instalado. Con estas dos librerías se realizaron pruebas en la ejecución de **HPL** para ver cuál era más óptima y nos daba mejor resultado en cuanto al performance.

## 6. Configuración de Infiniband

En cuanto a lo que **Infiniband** se refiere se optó por descargar e instalar el driver de Mellanox en su última versión estable, siendo esta la v4.9, con el Firmware de HPE que correspondía a su *part-number* (649282-B21\_B2).

También se configuró la red en ambos nodos creando las conexiones necesarias.

## 7. Tuning de HPL

En cuanto a la instalación de **HPL**, en este caso se instaló la versión 2.3, ya que era la más actualizada. Tras su instalación procedimos a realizar la modificación del archivo **HPL.dat** para buscar la mejor configuración para nuestros nodos.

Para realizar el tuning lo primero que hicimos fue comprender los parámetros que íbamos a modificar, entender que representaban dentro de la matriz y que papel cumplían dentro del programa de **HPL** para la solución de problemas de álgebra lineal.

El primer dato modificado fue el *Device Out* al cual se le asignó el valor 1 para poder así obtener los resultados de la prueba en el archivo **HPL.out**. Luego decidimos de que solo íbamos a realizar un problema con una sola configuración de **Ns**, **NBs** **P** y **Q**, así que a continuación vamos a exponer los valores que se les asignaron a cada una de estas variables:

Para **Ns** se optó por un valor cercano a 112.048 para las dimensiones de la matriz, llegamos a este valor a partir de la memoria disponible (124Gb entre los dos nodos) en Bytes ( $1,24e+11$ ) dividido entre 8 para una doble precisión y luego se le sacó raíz cuadrada para obtener la dimensión máxima de matriz que soporta esa capacidad de memoria (124.498) y luego decidimos que solo íbamos a usar el 90% de estas dimensiones por lo que llegamos al resultado de 112.048. El mejor resultado lo obtuvimos realizando varias pruebas llegando a un valor de 114.240, el cual se parece mucho a la estimación inicial.

Para **NBs** conocemos que los valores que puede tomar cada celda de la matriz están comprendidos entre 32-256 pudiendo tomar valores múltiplos de 8, por lo tanto, debido a que nuestra configuración de **Ns** estaba en el 90% se decidió que para **NBs** también íbamos a optar por usar el 90% del valor máximo posible, siendo este 230.4, ya que este número no es múltiplo de 8 optamos por los múltiplos de 8 más cercano a este, siendo estos últimos los valores 232 y 224.

En cuanto a **P** y **Q** decidimos tomar los valores de **P=4** y **Q=8**, esto debido a que **PxQ** debe dar como resultado el número de núcleos que van a ser usados en la ejecución de **HPL**, y como mencionamos anteriormente en el entorno de ejecución, entre los 2 nodos poseen un total de 32 núcleos, para obtener este número se puede realizar varias combinaciones (2x16, 16x2, 8x4, 4x8, 1x32, 32x1) pero en la teoría se tiene de que se deben buscar los valores que forman la matriz más cuadrada posible y que **P** sea menor que **Q**, debido a estas restricciones llegamos a la conclusión de que la mejor configuración para **P** y **Q** a la que podíamos llegar cumpliendo estas restricciones era la de **P=4** y **Q=8**.

## 8. Resultados

Luego de haber instalado los software anteriormente mencionados y teniendo en cuenta el tuning que hicimos a HPL con el archivo HPL.dat adjunto en la entrega. Obtuvimos un Rmax de **6.1027e+02 GFlops**, comparando esto con el Rpeak del cluster que eran unos **6.656e+02 GFlops** obtenemos un rendimiento de aproximadamente **91.68%**.

## Referencias

- *HPL Benchmark* (no date) *Netlib.org*. Available at: <https://www.netlib.org/benchmark/hpl/> (Accessed: August 8, 2021).
- *8.6. Configuring the NFS Server Red Hat Enterprise Linux 7* (no date) *Redhat.com*. Available at: [https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/7/html/storage\\_administration\\_guide/nfs-serverconfig](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/storage_administration_guide/nfs-serverconfig) (Accessed: August 8, 2021).
- *Configuring InfiniBand and RDMA networks red hat enterprise Linux 8* (no date) *Redhat.com*. Available at: [https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/8/html-single/configuring\\_infiniband\\_and\\_rdma\\_networks/index](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/configuring_infiniband_and_rdma_networks/index) (Accessed: August 8, 2021).
- *Download the Intel® oneAPI Base Toolkit* (no date) *Intel.com*. Available at: <https://software.intel.com/content/www/us/en/develop/tools/oneapi/base-toolkit/download.html> (Accessed: August 8, 2021).
- *Hewlett Packard Enterprise* (no date) *Software Details, Hpe.com*. Available at: [https://support.hpe.com/hpesc/public/swd/detail?swItemId=MTX\\_1ee5f6fa00b14116b2a1107f72](https://support.hpe.com/hpesc/public/swd/detail?swItemId=MTX_1ee5f6fa00b14116b2a1107f72) (Accessed: August 8, 2021).
- *Intel® oneAPI HPC Toolkit* (no date) *Intel.com*. Available at: <https://software.intel.com/content/www/us/en/develop/tools/oneapi/hpc-toolkit.html> (Accessed: August 8, 2021).
- *Mellanox* (no date a) *Mellanox.com*. Available at: <https://docs.mellanox.com/display/ConnectX5ENOC3/Linux+Driver+Installation> (Accessed: August 8, 2021).

- *Mellanox* (no date b) *Mellanox.com*. Available at: <https://docs.mellanox.com/display/MLNXOFEDv461000/Installing+Mellanox+OFED> (Accessed: August 8, 2021).
- *Mellanox* (no date c) *Mellanox.com*. Available at: [http://www.mellanox.com/downloads/firmware/mlxup/4.17.0/SFX/linux\\_x64/mlxup](http://www.mellanox.com/downloads/firmware/mlxup/4.17.0/SFX/linux_x64/mlxup) (Accessed: August 8, 2021).