Meta volante #1

Aurelio Vivas, Stiven Agudelo and Miguel Correa

***Abstract*—In this article we give the information about the servers we worked on and a brief description of the libraries that were useful to perform the HPL benchmark.**

***Index Terms*—High Performance Linpack, Intel, High Performance Computing, Infiniband, Network File System, Intel OneAPI.**

I. Introduction

The architecture considered for the deployment of the intel HPL benchmark is shown in Figure 1. Software and hardware details are given below.

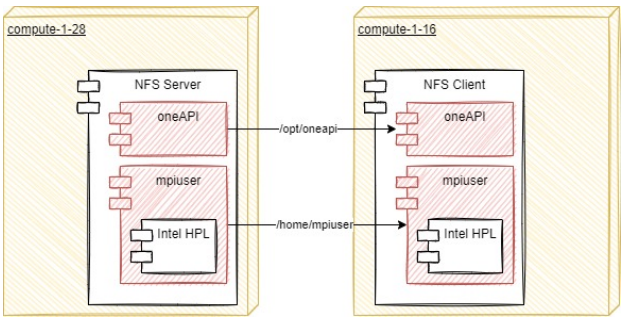

Fig. 1. Deployment digram

A. Hardware

- **CPU(s)** 16
- **Thread(s) per core:** 1
- **Core(s) per socket:** 8
- **Socket(s):** 2
- **NUMA node(s):** 2
- **Model name:** Intel(R) Xeon(R) CPU E5-2670
- **CPU MHz:** 3292.090
- **Network controller:** Mellanox Technologies MT27500 Family [ConnectX-3]

B. Software

- **Operative System:** Centos 8.2.
- **OneAPI Base Toolkit** [1], [2].
- **OneAPI HPC Toolkit** [3], [4].
  - **Benchmark:** HPLinpack 2.3 – High-Performance Linpack benchmark (December 2, 2018) [5].
  - **Numerical Math Library:** oneAPI Math Kernel Library for Linux [6].
  - **C compiler:** gcc (GCC) 8.4.1 20200928 (Red Hat 8.4.1-1), icc (ICC) 2021.3.0 20210609 [3], [4]
  - **MPI Library:** Intel(R) MPI Library for Linux* OS, Version 2021.3 Build 20210601 (id: 6f90181f1)
- **Network File System:** NFSv4.

A. Vivas was with the Universidad de los Andes, Bogotá, Colombia e-mail: aa.vivas@uniandes.edu.co.

S. Agudelo and M. Correa are with Universidad EAFIT, Medellín, Colombia.

II. Intel OneAPI

Intel's oneAPI is an open set of libraries that spands several domains that benefit from acceleration, including an interface for deep learning; general libraries for linear algebra math, video, and media processing; and others. The OneAPI base toolkit comprises the libraries shown in Table I [7]. In particular, we used the Math Kernel Library comprising BLAS, LAPACK, sparse solvers, fast Fourier transforms (FFT), random number generator functions (RNG), summary statistics, data fitting, and vector math [8], [9]. The Math Kernel Library also comprises the HPL benchmark optimized for Intel processors [10]. In addition, the OneAPI HPC toolkit [11] was installed. The toolkit comprises the C/C++ compiler and the MPI Library optimized by Intel [11]. Installation details of the OneAPI base and HPC toolkits are available in [2] and [4] respectively. Once you have installed OneAPI base and HPC toolkits, you will be able to access the intel optimized HPL benchmark on */opt/intel/oneapi/mkl/2021.3.0/benchmarks/mp_linpack/*. Here you will find, (1) a ready to use HPL benchmark binary file, *xhpl_intel64_dynamic*; (2) a shell script, *runme_intel64_dynamic*, which sets the required environment variables (Number of MPI processes, Network Fabric, etc) and execute the benchmark binary file; (3) the benchmark parameters file, *HPL.dat*; and other auxiliary files [12].

TABLE I

Library Name	Short Name	Description
oneAPI DPC++ Library	oneDPL	Algorithms and functions to speed DPC++ kernel programming
oneAPI Math Kernel Library	oneMKL	Math routines including matrix algebra, FFT, and vector math
oneAPI Data Analytics Library	oneDAL	Machine learning and data analytics functions
oneAPI Deep Neural Network Library	oneDNN	Neural networks functions for deep learning training and inference
oneAPI Collective Communications Library	oneCCL	Communication patterns for distributed deep learning
oneAPI Threading Building Blocks	oneTBB	Threading and memory management template library
oneAPI Video Processing Library	oneVPL	Real-time video encode, decode, transcode, and processing

III. Network File system

Network File System (NFS) is a distributed file system protocol originally developed by Sun Microsystems (Sun) in 1984, allowing a user on a client computer to access files over a computer network much like local storage is accessed [13]. In this case, we used the NFSV4 on the client and service sides, both included in the package `nfs-utils`. As depicted in 1, we shared the */opt* and */home/mpiuser* directories over the Ethernet network, the *mpiuser* was created beforehand. We the ssh keys for the

mpiuser in order to enable passwordless inter-nodes communication for MPI applications. In addition, we copied the */opt/intel/oneapi/mkl/2021.3.0/benchmarks/mp_linpack/* directory into the *mpiuser* home with the purpose of execute all the experiments in the *mpiuser* home which is visible by all the cluster computing nodes.

IV. Infiniband

Infiniband is a standard for high-performance communications networks. At the hardware level, the standard defines the architecture of switches, routers, channel adapters (Host Channer Adapter and Target Channer Adapter) and subnet managers that are part of an Infiniband-based network. [14], [15], [16], [17]. At the software level, the standard defines a series of interfaces or verbs *(verbs)* whose implementation depends on the hardware vendor. The word "verb" was used in this context since a verb describes an action. In this aspect, an application that makes use of the Infiniband network is developed based on these verbs or interfaces. The way in which these interfaces perform the underlying action depends on the implementation that each manufacturer has developed in the lower level libraries.

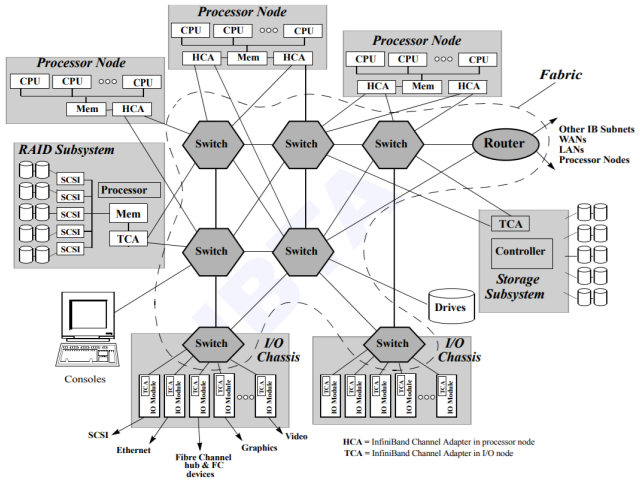

Fig. 2. Red basada en la especificación Infiniband [16]

The verbs defined by the standard are contained in the *libibibverbs* library. This library must be installed in order to enable the verbs. Applications developed in C/C++ must import the verbs, *infiniband/verbs.h*, in order to access interfaces such as *ibv_open_device*— and *ibv_post_send*, which allow manipulation of the network device [18]. Consequently, the *libibverbs* library is responsible for accessing specific libraries of the hardware manufacturer to find the specific implementation of each interface, and thus execute the implementation on the device. In this way, the *libibverbs* library makes the implementation of applications that use the network device agnostic to network devices created by different manufacturers [19].

Infiniband requires installation/configuration of kernel-level drivers (*kernel drivers*) and user-space-level drivers (*user-space drivers*) [20], [19]. That is, to use the network device with Infiniband support from any vendor, the hardware-level drivers and user-space-level drivers from the specific vendor are required. For example, in the case of `mlx4` devices, the kernel-level driver `mlx4_core` and the user-space-level driver `mlx4_ib` are required. In general the user-level and kernel-level drivers for Infiniband are compiled into the Centos 8.2 kernel. Additional libraries required to setup Infiniband are mentioned in [19] [21]. In particular, we installed *libibverbs-utils*, *librdmacm*, *infiniband-diags* and *rdma* and their dependencies. ,

V. HPL Parameters and Results

We mainly use the Netlib [22] and University of Luxembourg [23] documentations to understand the HPL parameters. The Intel documentation was used for the initial parameters tuning. Here we found the recommended values for P, Q, N, and NB which are described in [24]. With the above we defined the problem size as is shown by Equation 1.

$$N = NB * LCM(P, Q) * cores * x \tag{1}$$

Where x was used as a scale factor, $cores$ is the total number of cores in the cluster, and NB was set to 256 as recommended in the Intel documentation. For a single node, we considered $P = 1$, $Q = 2$, $cores = 16$, and assign different values for N when varying x. On the other hand, in two nodes, we set $P = 2$, $Q = 2$, $cores = 32$, and $N = 116736$ where N was computed from an online *HPL.dat* file calculator available in [25]. This configuration enabled us to achieve $6.44557e + 02$ GFlops in performance as reported in Equiation 2. The theoretical performance of the cluster are shown in Equations 4 and 5, describing the performance for the normal clock speed and turbo clock speed respectivelly.

$$R_{max} = 6.44557e + 02GFlops \tag{2}$$

$$R_{peak} = \#Nodes * \frac{\#CPU}{Node} * \frac{\#Cores}{CPU} * \frac{Frequency}{Core} * \frac{\#FLOPs}{Cycle} \tag{3}$$

$$R_{peak} = 2 * 2 * 8 * 2.60 * 8 = 665.6GFLOPS \tag{4}$$

$$R_{peak} = 2 * 2 * 8 * 3.30 * 8 = 844.6GFLOPS \tag{5}$$

VI. Conclusion

One main aspect to be considered when optimizing the execution of the HPL benchmark is the awareness of the number of threads and number of MPI processes that the benchmark creates. In this regard, the use of the *htop* tool was very important because at the beginning we noted that the application starts with a fixed number of processes, then it spawned several threads that were oversubscribing the computing cores causing a significant performance degradation. Finally, the Intel documentation recommends to create as many MPI processes as NUMA domains are available in the cluster [26]. This because a great amount of MPI processes may cause inter-node communication issues due to the NUMA architecture in cluster computing nodes.

REFERENCES

[1] Intel Corporation, "Get started with the intel® oneapi base toolkit for linux*," https://software.intel.com/content/www/us/en/develop/documentation/get-started-with-intel-oneapi-base-linux/top.html, (Accessed on 08/08/2021).

[2] ——, "Download the intel® oneapi base toolkit," https://software.intel.com/content/www/us/en/develop/tools/oneapi/base-toolkit/download.html, (Accessed on 08/08/2021).

[3] ——, "Get started with the intel® oneapi hpc toolkit for linux*," https://software.intel.com/content/www/us/en/develop/documentation/get-started-with-intel-oneapi-hpc-linux/top.html, (Accessed on 08/08/2021).

[4] ——, "Download the intel oneapi hpc toolkit," https://software.intel.com/content/www/us/en/develop/tools/oneapi/hpc-toolkit/download.html, (Accessed on 08/08/2021).

[5] ——, "Overview," https://software.intel.com/content/www/us/en/develop/documentation/onemkl-linux-developer-guide/top/intel-oneapi-math-kernel-library-benchmarks/intel-distribution-for-linpack-benchmark/overview-of-the-intel-distribution-for-linpack-benchmark.html, (Accessed on 08/08/2021).

[6] ——, "Developer guide for intel® oneapi math kernel library for linux*," https://software.intel.com/content/www/us/en/develop/documentation/onemkl-linux-developer-guide/top.html, (Accessed on 08/08/2021).

[7] Wikipedia, "oneapi (compute acceleration) - wikipedia," https://en.wikipedia.org/wiki/OneAPI_(compute_acceleration), (Accessed on 08/08/2021).

[8] Intel Corporation, "Accelerate fast math with intel® oneapi math kernel library," https://software.intel.com/content/www/us/en/develop/tools/oneapi/components/onemkl.html#gs.7yjhl0, (Accessed on 08/08/2021).

[9] ——, "Get started with intel oneapi math kernel library," https://software.intel.com/content/www/us/en/develop/documentation/get-started-with-mkl-for-dpcpp/top.html, (Accessed on 08/08/2021).

[10] ——, "Intel optimized linpack benchmark for linux*," https://software.intel.com/content/www/us/en/develop/documentation/onemkl-linux-developer-guide/top/intel-oneapi-math-kernel-library-benchmarks/intel-optimized-linpack-benchmark-for-linux.html, (Accessed on 08/08/2021).

[11] ——, "Optimized tools for hpc apps using intel® oneapi hpc toolkit," https://software.intel.com/content/www/us/en/develop/tools/oneapi/hpc-toolkit.html#gs.7yja4w, (Accessed on 08/08/2021).

[12] ——, "Contents," https://software.intel.com/content/www/us/en/develop/documentation/onemkl-linux-developer-guide/top/intel-oneapi-math-kernel-library-benchmarks/intel-distribution-for-linpack-benchmark/contents-of-the-intel-distribution-for-linpack-benchmark.html, (Accessed on 08/09/2021).

[13] Wikipedia, "Network file system," https://en.wikipedia.org/wiki/Network_File_System, (Accessed on 08/08/2021).

[14] Mellanox Tecnologies, "Introduction to infiniband," https://www.mellanox.com/pdf/whitepapers/IB_Intro_WP_190.pdf, (Accessed on 08/06/2021).

[15] Infiniband Trade Association, "Infiniband specification frequently asked questions," https://www.infinibandta.org/ibta-specification/, (Accessed on 08/07/2021).

[16] ——, "Infiniband architecture specification," https://www.afs.enea.it/asantoro/V1r1_2_1.Release_12062007.pdf, (Accessed on 08/07/2021).

[17] Mellanox Technologies, "Introduction to infiniband for end users," https://www.mellanox.com/pdf/whitepapers/Intro_to_IB_for_End_Users.pdf, (Accessed on 08/07/2021).

[18] D. Barak, "libibverbs - rdmamojo rdmamojo," https://www.rdmamojo.com/2012/05/18/libibverbs/#Library_API, (Accessed on 08/08/2021).

[19] R. Hat, "Introduction to infiniband," https://people.redhat.com/dledford/infiniband_get_started.html, (Accessed on 08/07/2021).

[20] R. H. C. Portal, "Chapter 13. configure infiniband and rdma networks red hat enterprise linux 7 — red hat customer portal," https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/networking_guide/ch-configure_infiniband_and_rdma_networks, (Accessed on 08/07/2021).

[21] Red Hat, "13.4. infiniband and rdma related software packages red hat enterprise linux 7 — red hat customer portal," https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/networking_guide/sec-infiniband_and_rdma_related_software_packages, (Accessed on 08/08/2021).

[22] Netlib, "Hpl tuning," https://www.netlib.org/benchmark/hpl/tuning.html, (Accessed on 08/08/2021).

[23] "High perf. linpack (hpl) - ul hpc tutorials," https://ulhpc-tutorials.readthedocs.io/en/latest/parallel/mpi/HPL/#high-performance-linpack-hpl-benchmarking-on-ul-hpc-platform, (Accessed on 08/08/2021).

[24] Intel Corporation, "Configuring parameters," https://software.intel.com/content/www/us/en/develop/documentation/onemkl-linux-developer-guide/top/intel-oneapi-math-kernel-library-benchmarks/intel-distribution-for-linpack-benchmark/configuring-parameters.html, (Accessed on 08/09/2021).

[25] Advanced Clustering Technologies, Inc., "How do i tune my hpl.dat file?" https://www.advancedclustering.com/act_kb/tune-hpl-dat-file/, (Accessed on 08/08/2021).

[26] Intel Corporation, "Running the intel optimized mp linpack benchmark," https://software.intel.com/content/www/us/en/develop/documentation/onemkl-linux-developer-guide/top/intel-oneapi-math-kernel-library-benchmarks/intel-distribution-for-linpack-benchmark/running-the-intel-distribution-for-linpack-benchmark.html, (Accessed on 08/09/2021).