Geospatial Benchmark

Derda Kaymak
Computer Science Dept.
Marquette University
derda.kaymak@marquette.edu

Abstract

Data in the world, and our work with data is increasing day by day. One of the areas where we use data a lot is geospatial operations. Therefore, being able to perform these heavy operations more effectively will provide us with many benefits in daily life. In addition, new hardware is being developed to handle this accumulated data, one of which is data processing units (DPUs). In this study, the optimizations that can be achieved by performing these operations on data processing units and using high performance computing are discussed. For this, we have prepared a benchmarking tool that measures the performance of geospatial operations on different hardware. Then we ran this tool on this CPU and DPU using different data and parameters and compared the results. As a result of this study, we observed that although the CPU works better than the DPU, performance can be increased by offloading the CPU to the DPU.

1 Introduction

Today, the data created around the world has reached a very large volume and continues to increase rapidly. This data contains very important information waiting to be discovered and attracts the attention of researchers in order to find solutions to the issues they are working on. In addition, the processing of this large amount of data requires significant computing power, so the importance of high performance computing, which allows multiple computers to process the same problem synchronously, is increasing and this method is becoming more and more popular.

Data communication and computation are crucial areas in HPC, and researchers are doing several studies to improve the performance of these two fields. Two of the areas that can be emphasized in order to increase performance in HPC are communication and load balancing. There are various libraries for nodes to communicate with each other, and some libraries are more advantageous depending on the type of tasks. In addition, correct distribution of the workload among the nodes is very important in terms of performance. For this, the given problem needs to be divided into smaller tasks that can be processed by different nodes and distributed to these nodes. The size of the task and the suitability of the architecture for this task are highly effective on performance.

In addition, some companies have released various hardware products to make certain communication and computation processes more effective, one of the most popular products in this field is the BlueField-2 DPU released by Nvidia. BlueField-2 is an improved and more flexible version of SmartNICs, it has its own ARM processors and memory, and has accelerators for some operations such as storage and cryptography. Considering these specifications, it can be seen that besides providing data communication from the network, it has some computation capabilities that can reduce the workload of the CPU. Various experiments have been carried out to measure the performance of the DPU (Liu et al., 2021) and increase the processing power by working with the CPU.

One of the areas where HPC is used extensively is geospatial analytics. Since spatial data is often very large and requires complex operations to be processed, multiple processors must work together to process this data in a reasonable amount of time. Due to the capabilities of BlueField-2 in the data communication area and the ability to reduce geospatial operations to some simpler operations, the use of BlueField-2 in geospatial analytics can be effective in offloading the CPU and reducing the redundancy in memory. Moreover, the inter-node communication libraries used in this area allow the data and operations to be shared between the nodes and the nodes to communicate with each other, and this makes it possible for CPUs and DPUs to work together even if they have different architectures.

Availability of geospatial data is critical for studying geospatial analytics. The reason for this is that although the studies usually require large data, the available data is limited. Therefore, there are some studies carried out to increase the availability of spatial data and to create artificial data for use in studies. In addition, another important subject is the data structures that will be used to store this data. There are various data structures implementations that enable to efficiently process large size geospatial data.

BlueField-2 has other usage areas besides geospatial operations. For example, in the field of artificial intelligence, Jain et al. (2022)'s study shows that up to 17.5% performance increase has been achieved by using it together with the CPU in the training of deep neural network models. Similarly, in molecular dynamics area, BlueField-2 DPU was used to offload the CPU and up to 20% speed increase was achieved with some parallelization operations (Karamati et al., 2022). These studies show that the inclusion of DPUs in computing systems can make valuable contributions to increasing the processing performance. However, Karamati et al. (2022) also showed that the BlueField-2 does not perform well in some situations due to the new release of this hardware and its limited features in some ways. In this regard, performance evaluation of DPUs is important in order to make more consistent analyzes to increase efficiency.

We have created a benchmarking tool to measure the performance of the BlueField-2 DPU, which has the potential for ambitious performance improvements as mentioned above, and to compare the performance with currently used processors. In our benchmarking tool, we used the geos library to perform geospatial operations, and the hpcx module and MPI interface to divide the workload into different nodes. Then, we measured the performance of the processors by running this benchmarking tool using various datasets, in several architectures and with different parameters.

In this paper, the performance of various architectures including CPU and DPU will be compared. In addition, the features of the benchmarking tool created for performance evaluation will be shared.

2 Experimental Setup

A benchmarking tool has been created to measure the geospatial processing performance of various hardware. In this tool, generally two files containing geospatial data are taken as input, various geospatial operations are performed on these data and the elapsed time is calculated. The geospatial operations mentioned here are R-tree iteration, Query (only includes filtering), Intersects, Overlaps, Touches, Crosses, Contains, Equals, Equals Exact (with 0.3 tolerance), Covers, and Covered By. The time measured for each operation includes creating the R-tree with the Base Layer, making a query for each geometry in the Query Layer using the MBRs in the R-tree and performing the specified operation with the candidate geometries obtained as a result of the filtering. The algorithm of the benchmarking tool is shown below.

Algorithm 1 Benchmark Algorithm

```
Input: Base File (B), Query File (Q), # of processes
Get or Create File Partitions for Layers
for each process (synchronously) do
    Create timer
    for each file partition(i) of the process do
        Get Base Layer B(i)
        Get Query Layer Q(i)
        for each operation do
            Create R-tree using B(i)
            for each geometry in Q(i) do
                Query MBR of geometry
                if MBR matches R-tree then
                    Perform operation
                end if
            end for
        end for
        Add elapsed time to timer
    end for
    Get total elapsed time
end for
return maximum elapsed time
```

With the help of various parameters that can be used, the tests to be performed can be varied. Thanks to the MPI library, this benchmarking tool can be run by selecting different number of nodes and number of processes per node. Since the number of hardware and processes to be used varies, some options have been created for using files containing data. For example, when a data file is given as an input, this file can be divided into as many parts as the number of processes containing an equal number of geometry, allowing each process to process a single file containing its own rank. As another option, when a folder containing pre-divided and numbered data is given as input to the tool, the processes continue until the files are finished using the round robin approach, and the elapsed time is calculated by summing the times obtained for different files. Finally, since it is sufficient to compare the matching parts of the Base and Query layers in spatially partitioned data, when the folders containing these partitioned data are given as input, the tests are conducted by matching files with the same number. There is also a parameter in this tool that allows the test to be repeated for the selected number of times, and the average of the times obtained from it to be given as a result.

The benchmark was tested on the Thor cluster of the HPC Advisory Council, which has 32 CPU and 32 DPU nodes. The technical specifications of this cluster are given in Table 1.

3 Benchmark Results

The benchmarking tool specified in the previous section was run on Thor cluster's CPU (Intel Broadwell E5-2697A) and DPU (BlueField-2) using different data and parameters, and various test results were obtained. The features of the data used are shown in Table 2.

Experiment #1: As a first experiment, the base layer data is divided into 128 pieces with equal number of geometry and tested with the round robin method. Cores of a single node were used in this experiment. The experiment was carried out using 1,2,4,8 cores and processes on DPU and CPU, and only the Intersects operation was performed. The results are shown in Figure 1 in comparison.

Considering the results, it can be seen that the CPU performs the same geospatial operation approximately 2.6 times faster than the DPU. In addition, it has been observed that the performance increases more than twice when the number of cores in the same processor unit is doubled. This can be thought of as the cache increase with the use of more cores, although the size of the R-tree remains constant due to the creation of one tree per file partition.

Experiment #2: Then, the Intersects operation is again performed on a single DPU and CPU node, but with the base layer data divided by the number

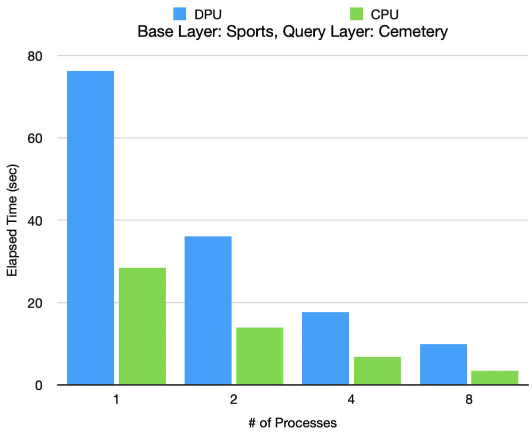

Figure 1: Intersects performance of single node using data divided into 128 partitions

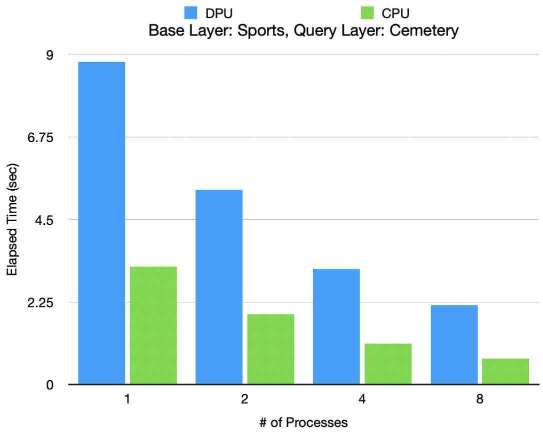

Figure 2: Intersects performance of single node using data divided into number of processes

of processes. The graph comparing CPU and DPU performances is given in Figure 1.

In the results of this experiment, similar to the previous experiment, it was observed that the CPU was 2.6 times faster than the DPU. However, when the number of processes for both hardware is doubled, it is seen that the performance increase is limited to approximately 1.7 this time. This is because when the data is less split, the generated R-tree is larger and the time complexity of the queries is reduced. For this reason, in addition to processing power, even if the increase in cache provides an advantage, the benefit obtained from the data structure that is optimized for these operations decreases.

Experiment #3: As another experiment, a test was conducted using multiple nodes with a single core for each node and including all geospatial

Hardware	Description
Chassis	56 Dell PowerEdge R730/R630 36-node cluster
CPU	Dual Socket Intel® Xeon® 16-core CPUs E5-2697A V4 @ 2.60 GHz (Broadwell)
Disk	1TB 7.2K RPM SATA 2.5” hard drives per node
Memory	256GB DDR4 2400MHz RDIMMs per node
Adapter	ConnectX-6 HDR100 100Gb/s InfiniBand adapters - BlueField-2 HDR100 100Gb/s adapters
Switch	Mellanox HDR Quantum Switch QM7800 40-Port 200Gb/s HDR InfiniBand

Table 1: Specifications of Thor Cluster

Dataset	Size	Records
Cemetery	56 MB	193 K
Sports	590 MB	1.8 M
Lakes	9 GB	8.4 M

Table 2: Datasets

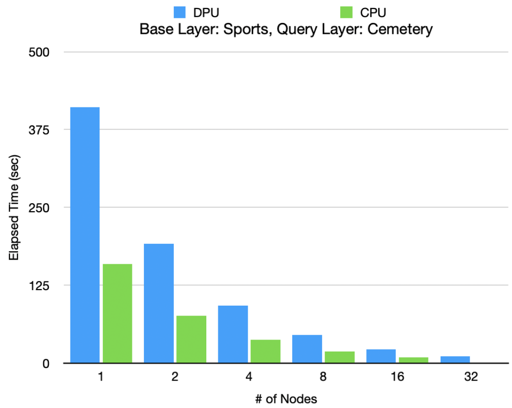

Figure 3: Total performance of multiple nodes using data divided into 128 partitions

operations mentioned in the paper. The test was repeated for 1,2,4,8,16 and 32 nodes and the results were noted. You can see the comparative results in Figure 3.

Looking at the results of this study, similarly, we can say that the CPU performs better than the DPU, and when the number of nodes is increased, the performance increases more than twice due to the increase in cache. Despite the increase in the processing time, it is seen that the ratio of the elapsed time between the hardware and the performance increase rate according to the number of nodes are similar to the results of the previous experiment using the same number of data partitions.

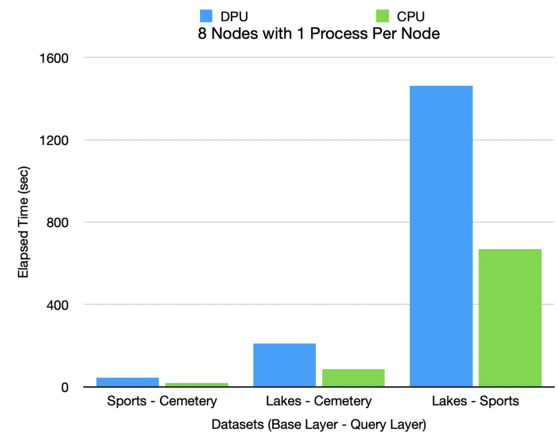

Figure 4: Total performance of multiple nodes using data partitions each containing 14014 geometries

Experiment #4: The experiment with the above-mentioned configurations was performed multiple times using various data. In Figure 4, you can see the 8-node performances of the processing units on different data. Sports and Lakes data, which is used as Base Layer here, is divided into partitions, each containing 14014 geometry. Therefore, although the R-tree sizes created in the experiments are equal, the number of files read differs.

When we look at the results given in Figure 4, we see that the CPU performance for the Sports-Cemetery file pair is about 2.4 times better than that of the DPU, but this ratio decreases slightly as the size of the data pairs used increases. Although the difference is small, this may be an indication that the DPU can be more scalable than the CPU.

Experiment #5: In addition, experiments with an equal number of processes were conducted using multiple cores of a single DPU node and using multiple DPU nodes using a single core of each. The results of this study are shown in Figure 5.

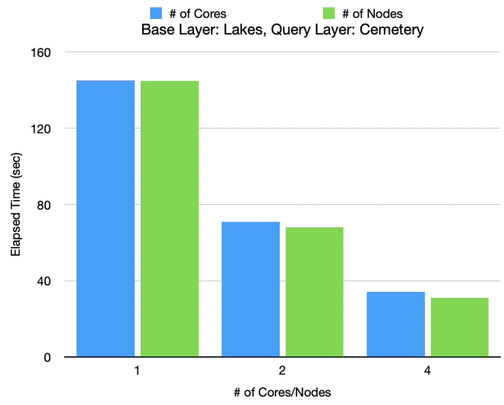

Figure 5: Intersects performance of DPU with single node-multiple core vs multiple node-single core using data divided into 128 partitions

When we look at the results, although the result is almost the same as the same test is repeated for the single process, when the number of processes increases, it is seen that the cases where the number of nodes are increased have a very small advantage over the cases where the number of cores is increased. This could be related to the memory and cache size used, but the difference is so small that it would not be accurate to draw any firm conclusions.

Experiment #6: Finally, tests were carried out when CPU and DPU were used together. Dynamic load balancing is used here, and when each processing unit finishes the file it is processing, it takes the new file from the queue. In this experiment, a single node and a single process are used for each hardware. Also, instead of MPI, gRPC library is used to provide communication between the CPU and DPU. The results of this study are shown in Figure 5.

Considering the results of this experiment, although the same process takes about 76 seconds when done on the DPU and about 28 when done on the CPU, the process takes about 16 seconds when the CPU and DPU are used together. According to these data, it is seen that the performance increase is above the expected.

4 Conclusions

In this study, a benchmarking tool that measures the geospatial operation performance of the hardware has been created and the performances have been measured especially by running it on a new hardware, BlueField-2 DPU, and an Intel CPU.

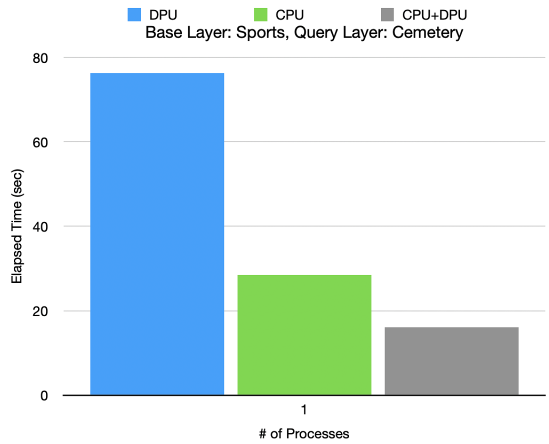

Figure 6: Intersects performance of DPU and CPU using data divided into 128 partitions

Considering the results, it is seen that the performance of the CPU is about 2.5 times better than the DPU. In addition, when the number of file partitions is kept constant for both hardware and the number of processes is doubled, the performance increases more than twice due to the increase in cache. As a result of the experiments, it is seen that although DPUs are less powerful than CPUs, they have the capacity to increase performance by offloading CPUs.

References

Arpan Jain, Nawras Alnaasan, Aamir Shafi, Hari Subramoni, and Dhabaleswar K. Panda. 2022. Optimizing distributed dnn training using cpus and bluefield-2 dpus. *IEEE Micro*, 42(2):53–60.

Sara Karamati, Clayton Hughes, K. Scott Hemmert, Ryan E. Grant, W. Whit Schonbein, Scott Levy, Thomas M. Conte, Jeffrey Young, and Richard W. Vuduc. 2022. "smarter" nics for faster molecular dynamics: a case study.

Jianshen Liu, Carlos Maltzahn, Craig D. Ulmer, and Matthew Leon Curry. 2021. Performance characteristics of the bluefield-2 smartnic. *CoRR*, abs/2105.06619.