PERFORMANCE OF DATA PROCESSING UNIT AND HIGH PERFORMANCE COMPUTING IN GEOSPATIAL OPERATIONS

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. Since the system to be used for this purpose has various components including hardware, algorithms and several approaches, the areas to be optimized also vary. Therefore, several methods used in this field are provided together with examples from different studies, and the potential performance increases that can be achieved by the combination of these methods are examined. As a result of the researches examined, it has been seen that the newly developed DPUs can significantly reduce the computation time for heavy operations by offloading the CPU.

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.

This paper provides information on the advantages and challenges of performing geospatial operations using DPUs and parallel programming. Considering the above information, this project is at the intersection of several fields of research and requires the collation of knowledge gained in these fields. Therefore, this research was conducted by dividing it into six different subtopics. These topics are shown in Figure 1.

Topics
Performance Characteristics of DPU
Geospatial Operations
Data and Specialized Data Structures
Communication for HPC
Load Balancing
Benchmarking Metrics

Figure 1: Subtopics of the research

2 Performance Characteristics of DPU

To better understand the function of DPUs, we first need to look at Network Interface Cards (NICs). NICs are hardware that provides the connection between our computers and the network. Usually data is processed by CPUs and GPUs and sent to and received from other computers on the network with the help of NICs. Over time, these devices were developed and SmartNICs were created. These devices have gained some computing capabilities in addition to the features of NICs. DPUs are the latest examples of this development, thanks to their programmable pipelines, they have important improvements for computation in addition to connection. In brief, DPUs are improved version of SmartNICs, and NICs are essential hardware for our computers at the moment. So, these devices can replace NICs if it is realized that DPUs will also provide considerable efficiencies in computing.

One of the topics being worked on in the development of SmartNICs is to give these devices some programmable capabilities. (Hoefler et al., 2017) have developed a vendor-independent and portable interface called sPIN to improve the networking performance of computers. With sPIN, users can write their own handler methods and decrease the need to access the host memory and also reduce the redundancy in the memory thanks to the fast buffer memory system. Fast buffer memory is a system that allows the SmartNIC to store data using the memory on its own, and in this way, the operations that the SmartNIC can process can be done directly on the device without being sent to the host memory and CPU. You can see the logic of this approach comparatively in Figure 2. In addition, the study has demonstrated that using this interface, significant speedups are provided to reference methods in real applications. As a limitation in this

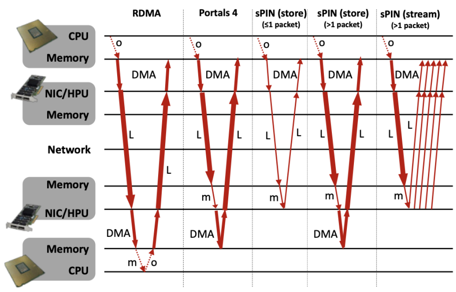

Figure 2: Packet transmission between computers using sPIN (Hoefler et al., 2017)

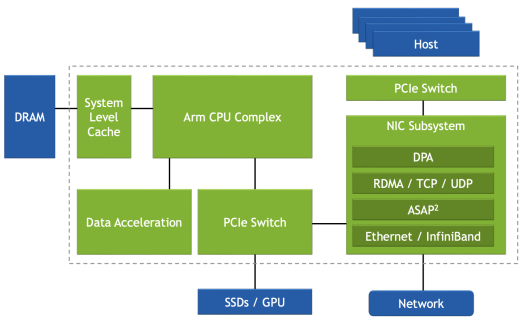

Figure 3: BlueField-2's architecture (Burstein, 2021)

work, data movement and packet processing are done by handlers, but the design of the handlers is left to the programmers, and since there is no limit to the code length that can be written, possible long codes may cause the system to work inefficiently.

DPUs are newly developed technologies and various studies have been done to evaluate their performance. One of the devices frequently used in this field is Nvidia's BlueField 2 DPU. Figure 4 shows the architecture of BlueField-2. Liu et al. published a comprehensive study showing the performance characteristics of the BlueField-2. You can find the technical specifications of BlueField-2 in Figure 4. They did the research under two headings as communication and computation, and they evaluated the performances of the DPU for these two by comparing them with other modes and hardware. For communication, they concluded that in in-transit computations, embedded mode, which relies on positioning the DPU to assist the CPU, outperforms the separated mode, which allows it to function as a separate node. For computation, the researchers performed a test called stress-ng using the BlueField-2 DPU and various servers to measure the processing size they can handle by performing various operations on the processors. Some logic and arithmetic operations, memory and caching algorithms, sorting and search-

CPU	8 ARMv8 A72 cores (64-bit) @ 2.5 GHz, 1MB L2 cache per 2 cores, 6MB L3 cache
DRAM	16 GB on-board DDR4-1600
Storage	eMMC flash memory
Network	Ethernet or InfiniBand: dual ports of 10/25/50/100 Gb/s, or a single port of 200 Gb/s
Accelerators	• Hardware root of trust • RegEx • IPsec/TLS data-in-motion encryption • AES-XTS 256/512-bit data-at-rest encryption • SHA 256-bit hardware acceleration • Hardware public key accelerator • True random number generator
PCIe	Gen 4.0 x16
OS	Ubuntu 20.04 (kernel 5.4.0-1007-bluefield)

Figure 4: Technical specifications of the BlueField-2 (Liu et al., 2021)

ing algorithms and cryptography operations can be given as examples to these operations. As a result of this test, it was found that the computing performance of the DPU lags behind that of general purpose CPUs, but with the help of the accelerators in the DPU, which are also shown in Figure 4, the DPU has been shown to perform much better in encryption operations, memory operations under contention, and on-card IPC operations.

Various universities and institutions provide hardware support for users doing research on DPU usage. For example, Cloudlab and HPC Advisory Council High-Performance Center (HPCAC) have clusters with multiple DPUs and are frequently used by researchers working in this field.

3 Geospatial Operations

While this research can provide insight for many applications that require high processing power, more specifically, the focus of this survey is geospatial operations. Geospatial Operations' usage area is increasing day by day and since they are both compute-intensive and communication-intensive, they need to be performed faster. Moreover, there are different types of operations. In Jackpine, Ray et al. (2011) examined these operations under two classes as Micro and Macro. Micro operations are based on comparison of two geometries such as finding overlap, intersection, cross, touch, equal, within, contain, disjoint and spatial join. And, operations such as geocoding, map search, toxic spill, land information management and flood risk analysis can be given as examples to Macro operations.

In geospatial operations, data is usually pro-

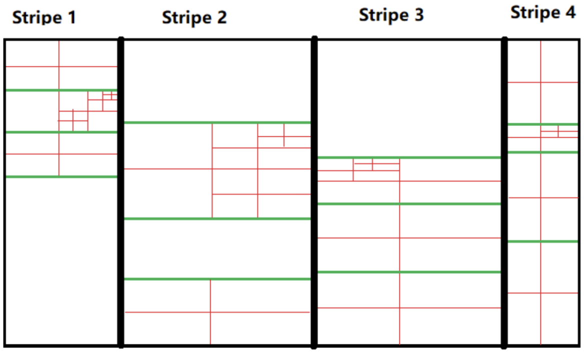

Figure 5: Representation of a data partitioned using ADP (Yang and Puri, 2020)

cessed by dividing it into pieces. There are two basic approaches here. One of them is file based partitioning. In this approach, data is usually processed by splitting it into pieces of equal file size. The other approach is spatially partitioning, where data is partitioned using the positions of the polygons. These approaches offer some advantages over the type of the given task. In addition, Yang and Puri (2020) offer a new data partitioning algorithm, called Adaptive Data Partitioning (ADP), that divides data by locations and makes it more suitable for load balancing. In this method, the data is divided into pieces of different sizes, which vary according to the data density, instead of grids of equal size. They also provide faster partitioning by parallelizing this process. Figure 5 shows a representation of the partitioning logic of ADP. Vertical thick lines separate the different nodes used in parallel computing, and horizontal medium-thick lines separate the cores of each node.

Geospatial operations are often complex operations, but there are some methods that can improve performance by simplifying certain parts of these operations. As Liu et al. (2019b) mentioned, one of these methods is hierarchical filtering and refinement. Geographic shapes are represented by points, lines, and polygons. In most cases where these operations are needed, a large number of complex polygons must be compared with each other using the above mentioned Micro operations. However, it requires a lot of time and processing power. Instead, the smallest boxes that can contain polygons, called minimum bounding rectangles (MBRs), are created. Then, a list of candidate polygons is created by comparing these rectangles instead of the actual polygons, and the polygons in this list are used to compare the actual polygons. Since MBRs

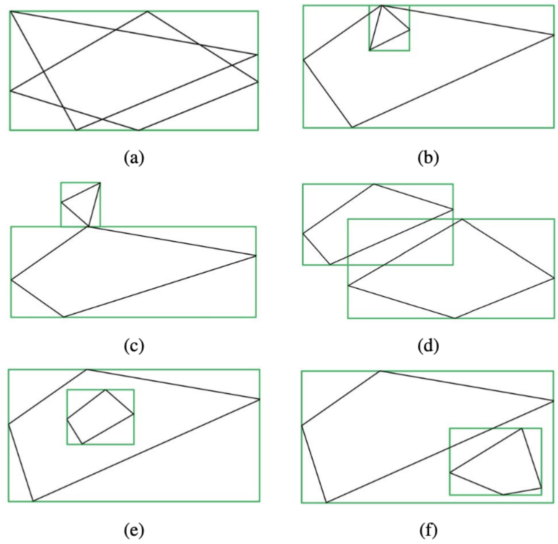

Figure 6: Minimum bounding rectangles and several cases (Liu et al., 2019b)

have a much simpler structure than actual polygons, they are much easier to compare and take up less memory space. With this method, comparisons that require high computing are limited to a small list of candidates and a significant performance increase is achieved in this way. Figure 6 shows example MBRs and the various cases that can be inferred using them. For example, the six candidates shown here are extracted by looking at the intersection status of the MBRs to find out if two polygons intersect. Then, the same process is done on the candidate actual polygons one by one and non-intersecting cases such as (d) and (f) are eliminated. Liu also extends this method and provides a more advanced version. With the method called PolySketch, polygons are divided into line segments and MBRs of these segments are created. Then, these rectangles are then used for the comparisons. It has been stated that with this method, spatial join, which is one of the most used operations, can be performed much faster.

4 Data and Specialized Data Structures

Another subject to examine the performance of geospatial operations is data and the data structures that provide some advantages over others in storing the spatial data. Geospatial data is often hard to find. Although there are some real datasets, these data are limited and difficult to diversify. At this point, the University of California Riverside offers a comprehensive dataset that includes various landforms around the world (Ghosh et al.). This

Dataset	Size	Records
Cemetery	56 MB	193 M
Sports	573 MB	2 M
Parks	7.7 GB	10 M
Lakes	7.4 GB	8 M
Buildings	29.8 GB	115 M

Figure 7: Sample Data from UCR-STAR Geospatial Dataset

resource contains data of various geometric shapes such as cemetery, parks, lakes, roads, buildings, airports all over the world and continues to be updated as new data are obtained. In this source, where datasets of different sizes can be obtained, the size of some data reaches hundreds of GB. Since the data here is from the real world, it shows how large the data is handled by real applications that are using spatial data and the importance of optimizing these operations. In Figure 6, you can see the features of some of the data provided by UCR-STAR. This dataset is updated as new data comes in. In addition, a spatial data generation tool called Spider has been developed to generate artificial data for people working in this field to use it in their research (Vu et al., 2021). With this tool, datasets with various distributions (uniform, gaussian, diagonal etc.), cardinalities and geometries can be created. Spider allows the generated data to be downloaded in various formats such as CSV, WKT and GeoJSON. Also, the tool provides a link that allows the created data to be shared among the research team members. Considering that various and large spatial data should be used to evaluate performance, it can be said that this tool is especially useful for researchers working in this field.

In addition, the data structure to be used for geospatial operations must be well designed to allow this data to be processed effectively. One of the popular data structures used for storing spatial data is R-tree which is invented by Antonin Guttman (Guttman, 1984). Similar to B-tree, Rtree is eight balanced index structure where its leaf nodes contain pointers to the objects, which are polygons, lines or points for geospatial data. Moreover, internal nodes contain geometric shapes sized to cover their chids. Thanks to this structure of R-tree, nodes that are in relations with each other such as intersection, overlap, contain each other are kept in R-tree in an indexed manner. In this way, frequently used queries for geospatial operations can be performed efficiently. An example of R-tree is shown in Figure 8.

There are several studies that examine the efficiency of R-tree. Agarwal et al. (2012) performed geospatial operations by comparing geometries from two spatial data source using large spatial data. In their study, first they created R-tree from one the data sources, called Overlay Layer, and for each polygon in the other data source, called Base Layer, they performed intersection query on the R-tree. Then, they use set of polygons in Overlay Layer by sorting them instead of using R-tree. They concluded that the scenario using R-tree outperforms the sorting based approach.

There are also different versions of R-tree such as Hilbert R-tree which is used for multidimensional data and R*-tree which is harder to construct yet has a better query performance. Moreover, there are optimized implementation of R-tree for different architectures. For example, Cho et al. (2021) re-designed the R-Tree algorithms to be failure-atomic and byte-addressable to use it with persistent memories such as NVRAMs. They created their own in-place rebalancing algorithm for R-tree which is optimized for persistent memory and compared it to the existing rebalancing algorithm. Finally, they showed that fine-grained control of failure-atomic design makes it possible to perform lock free searches and they showed that it provides a performance increase of up to 13.5 times by increasing concurrency. These studies show that R-trees can also work in harmony with emerging technologies.

5 Communication for HPC

High performance computing (HPC) refers to the use of supercomputers and parallel processing techniques to solve complex computational problems. HPC systems are characterized by their high speed and ability to process large amounts of data quickly. HPC is used in a variety of fields, including science, engineering, finance, and medicine. HPC systems are typically built using specialized hardware, such as high-end processors and large numbers of interconnected computers, and they often require specialized software and programming techniques to take advantage of their capabilities. They are often used to solve problems that require the simultaneous execution of many small, independent

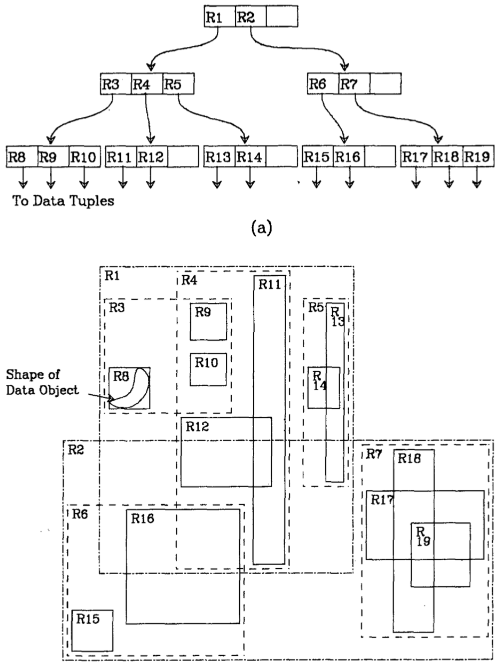

Figure 8: R-tree structure (Guttman, 1984)

tasks, or to perform simulations that require a large number of calculations to be performed in parallel. HPC is an essential tool for many scientific and industrial applications, and it is constantly evolving as technology advances and new applications are developed.

In their study on the use of HPC for remote sensing, Lee et al. (2011) examined the recent advances of high performance computing in terms of fields such as multiprocessor systems, large-scale and heterogeneous networks of computers, grid and cloud computing environments. According to Lee et al. (2011), the structure in which high performance computing is performed, aka distributed system, should have the following capabilities,

1. Resource discovery and catalogues
2. Data interoperability
3. Service/Job/Workflow Management
4. Resource instantiation and provisioning
5. Monitoring
6. Event Notification
7. Security
8. Accounting and Auditing

Since geospatial operations are complex operations that are usually performed using large data, parallel programming is also used to compute these operations. In parallel programming, the workload is distributed to different nodes, and in the end, the results of each node are gathered in one place. Therefore, the communication of nodes with each other, has a great impact on performance, like processing power.

One of the problems experienced in HPC communication is that the data shared between nodes is often very large. Agarwal et al. (2012) drew attention to this issue in his study, since there is a high serialization cost even in the transmission of a data that can be considered small in terms of Geospatial data, the metadata is sent instead of sending and receiving the actual data, and then the nodes make redundant file reads themselves.

There are various libraries used in the HPC field for inter-node communication. MPI (Message Passing Interface) is the one of these libraries which is a standardized programming library for parallel and distributed computing. It provides a set of functions and routines that allow developers to write programs that can communicate with each other and exchange data over a network.

MPI is widely used in high performance computing (HPC) environments, where it is used to write programs that can run on large clusters of computers and take advantage of their parallel processing power. It is a flexible and powerful tool that can be used to write programs that can run on a wide range of hardware and software platforms. As a shortcoming, it can be said that MPI's support for heterogeneous systems is low and it needs improvement.

gRPC can be shown as another library used in this field. gRPC, short for Google Remote Procedure Calls, is a modern, open-source remote procedure call framework that can run anywhere. It enables client and server applications to communicate transparently, and makes it easier to build connected systems.

At a high level, gRPC allows a client application to call a server application's methods as if they were local. This is accomplished through the use of a network protocol that defines the format of the messages that are exchanged between the client and server. gRPC is based on the HTTP protocol and uses Protocol Buffers, a language- and platform-agnostic data serialization format, to define the structure of the messages that are exchanged between the client and server. This allows gRPC to support a wide range of programming languages and platforms. Although gRPC is also used in high performance computing, it is a general purpose communication library. Being independent of the platform makes it easy to implement, but due to the complex structure of the TCP/IP stack it uses, it may cause unnecessary overhead when used in local systems.

Overall, MPI and gRPC are both useful tools for building distributed and parallel systems, but they are intended for different purposes and have different strengths and capabilities. MPI is particularly well-suited for HPC applications, while gRPC is a more general-purpose RPC framework that can be used for a wide range of applications.

6 Load Balancing

In order to maximize the benefit from using hardware together on the same problem, the workload must be well distributed according to the processing power of the different nodes. For this, operations can be done by dividing the workload by the number of hardware. Load balancing is a technique used for this purpose to distribute incoming requests or workloads across multiple servers or resources in order to optimize performance, reliability, and scalability.

In a load-balanced system, incoming requests are forwarded to a load balancer, which is responsible for distributing the requests across a group of servers or resources. The load balancer uses a variety of algorithms to determine the best server or resource to handle each request, based on factors such as server performance, workload, and availability. There are several benefits to using load balancing. For instance, if one server or resource fails, a load-balanced system can continue to operate by routing requests to other servers or resources. This helps to improve the overall reliability of the system. Moreover, load balancing makes it easier to scale a system by allowing new servers or resources to be added or removed as needed. This makes it easier to respond to changing workloads and demand. It is commonly used in a variety of applications, including web servers, databases, and cloud computing environments.

There are two main approach for load balancing,

1. Static Load Balancing
2. Dynamic Load Balancing

The main difference between the two is how the load balancer determines which server or resource to use for each request.

In static load balancing, the load balancer uses a fixed set of rules or a predetermined algorithm to determine which server or resource to use for each request. This means that the mapping between incoming requests and servers or resources is fixed, and does not change over time.

Dynamic load balancing, on the other hand, uses real-time data and feedback to determine which server or resource to use for each request. This means that the mapping between incoming requests and servers or resources can change over time, based on factors such as server performance, workload, and availability.

Both static and dynamic load balancing have their own advantages and disadvantages. Static load balancing is generally simpler to implement and may be more suitable for systems with predictable workloads. Dynamic load balancing, on the other hand, is more flexible and can adapt to changing workloads and resource availability. Therefore, the choice between static and dynamic load balancing depends on the specific needs and requirements of the system being built.

7 Benchmarking Metrics

In computer science, benchmarking refers to the practice of evaluating the performance of a system or algorithm using a standardized dataset or test. Benchmarking metrics are used to measure the performance of a system including hardware and software performance against a set of predetermined standards or benchmarks. It is important to carefully consider which benchmarking metrics are most relevant for a particular task, as different metrics may be more or less appropriate depending on the characteristics of the system and the data. Additionally, it is often helpful to compare the performance of a system to baseline, in order to understand how much improvement a particular model provides. In addition, the variables to be used in the test must be well chosen in order to obtain consistent results.

As a benchmarking study, Ray et al. (2011) created a benchmarking tool called Jackpine to measure the performance of spatial databases. Ray et al. (2011) performed the benchmarking test by choosing different types of operations, which are also mentioned in the ”Geospatial Operations” section, as test variables, and database types such as MySQL, PostgreSQL and Informix. They then evaluated the results they obtained from the tests using various metrics and came up with an overall score. This research also provides a general perspective on how the benchmarking test should be done.

In this research field, the cases that are wanted to be calculated are generally the performance of the CPU, the performance of the DPU, and the performance when both are used together. When we look at the related studies; the number of nodes, data size, and geospatial operation types can be good test variables.

In addition, there are several benchmarking metrics that may be relevant for evaluating the performance of geospatial operations on DPUs and HPC systems. Some potential variables to consider might include,

1. Computation time: This is the amount of time it takes to complete a particular geospatial operation, and it is often a key factor in determining the performance of a system.

2. Data transfer speed: In geospatial operations, large amounts of data may need to be transferred between different nodes, so the communication is an important factor. The speed at which this data can be transferred can significantly impact the overall performance of the system.

3. Memory usage: The amount of memory required to perform a particular geospatial operation can be an important factor in determining the performance of a system. Systems with limited memory may struggle to perform certain operations, or may need to use slower disk-based storage as a result.

4. Energy consumption: The amount of energy required to perform a particular geospatial operation can be an important factor, particularly in HPC systems where energy consumption may be a significant cost.

5. Scalability: The ability of a system to handle increasing amounts of data or computational workload can be an important benchmarking metric. Systems that are able to scale well may be able to handle larger or more complex geospatial operations more efficiently.

It may be helpful to consider a combination of these and other variables when evaluating the performance of geospatial operations on a DPU and HPC system.

8 Applications In Various Fields

DPUs have a wide variety of uses. They are used to efficiently perform data processing tasks, such as machine learning and deep learning, data analytics, data storage and retrieval, networking, and edge computing for the Internet of Things (IoT). DPUs can be used to accelerate a variety of tasks and optimize data storage and retrieval in various applications. Generally, performance increase can be achieved by making optimizations in the methods given in the titles mentioned above. Below are examples of some studies where the DPU is used together with the mentioned methods to increase performance.

For instance, Jain et al. (2022) aimed to increase performance by using CPU and DPU together in Deep Neural Network area. There are several steps to create a model trained with deep learning, such as data augmentation, training, and validation of the trained model. These operations are usually compute intensive due to the large amount of data used and are performed on the CPU and GPU. In this study, different stages of deep learning were offloaded to DPU, in other words, load balancing was made between CPU and DPU, and computation time was tried to be reduced. Experiments were conducted on the HPC Advisory Council cluster using the MPI library which is modified by these researchers.

First of all, the data loading and data augmentation parts are performed in the DPU and the processed data is sent to the CPU and the training and validation stages are processed there. Then, instead of data loading and data augmentation parts, model validation was done on the DPU and all the remaining operations were performed on the CPU. In this section, it was seen that model validation in DPU takes longer time than model training in CPU, and the experiment has been updated so that the model validation task is split into two parts and shared between the CPU and DPU. As a final experiment, a hybrid design was created by combining the first two designs. Data augmentation and some of the model validation were performed on the DPU. Although model validation takes longer to process in the DPU, the reason for adding a new process is

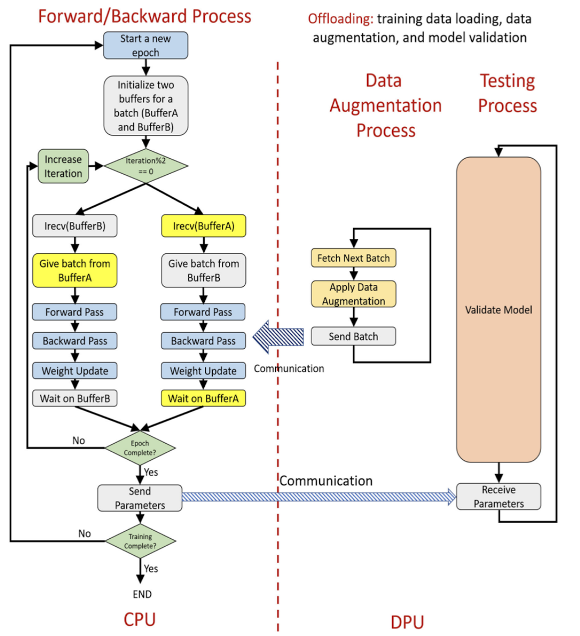

Figure 9: Hybrid design for DL training (Jain et al., 2022)

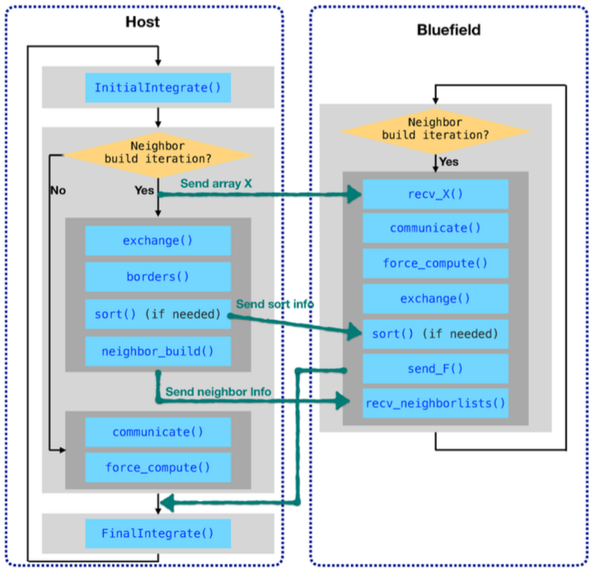

Figure 10: Work sharing between CPU and DPU in MiniMD (Karamati et al., 2022)

that since the data preprocessing, model training and model validation parts in deep learning are sequential processes, one process cannot be started before the other is finished. Therefore, the augmented data processed in the DPU is sent to the CPU and trained, and then some of the test data is sent to the DPU and the trained model is validated. With this approach, the waiting time of the processors at any time interval is minimized and maximum efficiency is obtained from the processors. You can see the details of the hybrid design in Figure 9.

These designs have been tested with CNN models and Transfer learning and various performance evaluations have been made. In performance evaluations using various parameters such as datasets, neural network models, batch sizes, number of nodes and file systems, it was observed that up to 17.5% performance increase was achieved in some scenarios thanks to proper load balancing.

As future work, since it is widely used in deep learning training, GPU can be included in this system and updates can be made on processes that will be offloaded to DPU due to the computing power difference in the processing units. With this way, the capabilities of the DPU in this area can also be better tested, as data communication becomes even more important in this case.

As an example from another research area, Karamati et al. (2022) conducted a study on the "molecular dynamics application" case study to determine whether it is possible to increase the processing power using the BlueField-2 DPU. With using MiniMD molecular dynamics proxy application, they compared the performance of Intel host and BlueField-2 DPU in terms of data communication and computation, including power consumption. In addition, the researchers redesigned and parallelized the sequential processes in MiniMD and distributed them to the CPU and DPU. Similar to Jain et al. (2022), this study also aims to minimize the idle time of processing units. Moreover, and they tested the networking and data movement performance of BlueField-2. Thanks to the parallelized MiniMD algorithm and the use of DPU, they have achieved a performance increase of up to 20%. Information about the processes that the CPU and DPU perform and the communication between them are shown in Figure 10.

The researchers use MPI for inter-node communication. The CPU used in this study has x86 architecture and BlueField-2 has arm architecture. Therefore, this study can be given as an example of high performance computing with heterogeneous systems.

Liu et al. (2019a), built a framework called iPipe for offloading distributed applications onto DPUs. They used different test applications to evaluate the performance such as real-time data analytics engine, distributed transaction system, and repli-

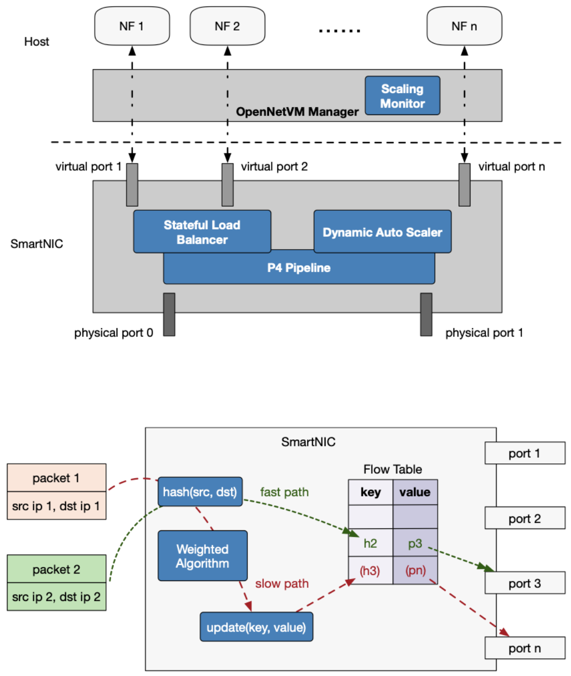

Figure 11: Load-balancer on DPU (Ni et al., 2021)

cated key-value store. The results clearly show that SmartNICs play an important role in both data processing and reducing latency in network.

DPUs are also used in security field. Diamond et al. (2022) have tried to take advantage of DPUs by using them in network security because of the hardware accelerators they have which are given in Figure 4. In their research, they demonstrated the benefits of offloading the encryption of Ethernet traffic to hardware accelerators of BlueField-2 DPU. They showed that BlueField-2's IPSec/TLS accelerator is capable to encrypt IPsec datagrams at nearly 16 Gb/s which is much more than many CPUs. They concluded that BlueField-2 has the ability to both improve network performance and help with Ethernet security in confidentiality, integrity, and authentication.

There are also studies suggesting that load-balancers should be offloaded to the DPU (Cui et al., 2021),(Ni et al., 2021). The results obtained in these studies suggest that DPU has the potential to do load balancing effectively and this topic should be continued to be researched. In Figure 11, you see an example of the structure of a load balancer running on the DPU.

Considering these studies, it can be seen that when load balancing, communication and data usage are done correctly, DPU provides benefits by reducing computation time in many areas. Therefore, performance increase can be achieved with the use of DPU in geospatial operations.

9 Conclusion and Future Work

In this study, the issues related to the measurement and improvement of the geospatial operations performance of data processing units and high performance computing were discussed by giving examples of various studies in different research fields. Due to the fact that the research covers a wide area, it has been examined under various titles.

The first of these was to determine the specifications and performance characteristics of the DPU, which is the hardware on which geospatial operations will be run. Information on the development of the DPU and approaches to its use in different fields were expressed there.Afterwards, information was given about the types of geospatial operations that the research focused on specifically, their characteristics and which classes they were examined under. In addition, the properties of the data used in geospatial operations and the data structures used in the processing and storage of this data are presented. As other subtitles, communication and load balancing, two important areas of high performance computing, were handled. The effect of communication on overall performance and the good and bad aspects of the libraries used in communication were evaluated. In addition, information was given about the benefits of load balancing and its different types. Finally, evaluations on how the test variables and benchmarking metrics to be used for performance evaluation should be selected according to the research type were presented. In addition to these titles, it is aimed to make inferences about the benefits of using geospatial operations on DPU and HPC by giving examples from studies in which these titles are used together.

As a result, DPUs are emerging hardware and have the potential to provide various optimizations in different areas. Considering the benefits it provides in other areas, it is seen that DPUs can also provide significant improvements for geospatial operations that are compute and communication intensive. As the studies in this field increase, we can see these units frequently in our personal computers or data centers. As future work, it can be studied on the use of DPUs in more areas. In addition, since DPUs are still under development, studies can be made on which accelerators can be added to these units in the future to optimize such heavy operations. Optimizing the communication methods used in heterogeneous systems is also an area that needs improvements.

References

Dinesh Agarwal, Satish Puri, Xi He, and Sushil K. Prasad. 2012. A system for gis polygonal overlay computation on linux cluster - an experience and performance report. *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum*, pages 1433–1439.

Idan Burstein. 2021. Nvidia data center processing unit (dpu) architecture. In *2021 IEEE Hot Chips 33 Symposium (HCS)*, pages 1–20.

Soojeong Cho, Wonbae Kim, Sehyeon Oh, Changdae Kim, Kwangwon Koh, and Beomseok Nam. 2021. Failure-atomic byte-addressable r-tree for persistent memory. *IEEE Transactions on Parallel and Distributed Systems*, 32:601–614.

Tianyi Cui, Wei Zhang, Kaiyuan Zhang, and Arvind Krishnamurthy. 2021. Offloading load balancers onto smartnics. In *Proceedings of the 12th ACM SIGOPS Asia-Pacific Workshop on Systems*, APSys '21, page 56–62, New York, NY, USA. Association for Computing Machinery.

Noah Diamond, Scott Graham, and Gilbert Clark. 2022. Securing infiniband networks with the bluefield-2 data processing unit. In *International Conference on Cyber Warfare and Security*, volume 17, pages 459–468.

Saheli Ghosh, Tin Vu, Mehrad Amin Eskandari, and Ahmed Eldawy. UCR-STAR: The UCR Spatio-Temporal Active Repository. *SIGSPATIAL Special*, 11(2):34–40.

Antonin Guttman. 1984. R-trees: A dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, SIGMOD '84, page 47–57, New York, NY, USA. Association for Computing Machinery.

Torsten Hoefler, Salvatore Di Girolamo, Konstantin Taranov, Ryan E. Grant, and Ron Brightwell. 2017. Spin: High-performance streaming processing in the network. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '17, New York, NY, USA. Association for Computing Machinery.

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.

Craig A. Lee, Samuel D. Gasster, Antonio Plaza, Chein-I Chang, and Bormin Huang. 2011. Recent developments in high performance computing for remote sensing: A review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 4(3):508–527.

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

Ming Liu, Tianyi Cui, Henry Schuh, Arvind Krishnamurthy, Simon Peter, and Karan Gupta. 2019a. Offloading distributed applications onto smartnics using ipipe. In *Proceedings of the ACM Special Interest Group on Data Communication*, SIGCOMM '19, page 318–333, New York, NY, USA. Association for Computing Machinery.

Yiming Liu, Jie Yang, and Satish Puri. 2019b. Hierarchical filter and refinement system over large polygonal datasets on cpu-gpu. In *2019 IEEE 26th International Conference on High Performance Computing, Data, and Analytics (HiPC)*, pages 141–151.

Zhen Ni, Cuidi Wei, Timothy Wood, and Nakjung Choi. 2021. A smartnic-based load balancing and auto scaling framework for middlebox edge server. In *2021 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pages 21–27.

Suprio Ray, Bogdan Simion, and Angela Demke Brown. 2011. Jackpine: A benchmark to evaluate spatial database performance. In *Proceedings of the 2011 IEEE 27th International Conference on Data Engineering*, ICDE '11, page 1139–1150, USA. IEEE Computer Society.

Tin Vu, Sara Migliorini, Ahmed Eldawy, and Alberto Belussi. 2021. Spatial data generators.

Jie Yang and Satish Puri. 2020. Efficient parallel and adaptive partitioning for load-balancing in spatial join. In *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 810–820.