

Integrated Geo Cloud Solution for Seismic Data Processing

Pruthvi Raj Venkatesh¹, Chaitanya Kanchibhotla², DVLN Somayajulu², Radhakrishna P²

¹*Ambedkar Institute of Technology*

²*NIT Warangal*

pruthviraj_v@hotmail.com, ckanchibhotla@gmail.com, soma@nitw.ac.in, prkrishna@nitw.ac.in

Abstract: Oil industries generate an enormous volume of digitized data (e.g., seismic data) as a part of their seismic study and move it to the cloud for downstream applications. Moving massive data into the cloud can pose many challenges, especially to Commercial-off-the-shelf geoscience applications as they require very high compute and disk throughput. This paper proposes a digital transformation framework for efficient seismic data processing and storage comprising of: (a) Novel Data storage options, (b) Cloud-based HPC framework for efficient seismic data processing, and (c) MD5 hash calculation using the MapReduce pattern with Hadoop clusters. Azure cloud platform is used to validate the proposed framework and compare it with the existing process. Experimental results show a significant improvement in execution time, throughput, efficiency, and cost. The proposed framework can be used in any domain which deals with extensive data requiring high compute and throughput.

Keywords: *Seismic, HPC, Azure Blob, Geoscience, MapReduce, Hadoop, Azure Files, NetApp, MD5*

1. Introduction

Seismic study is an essential part of a vertical in the oil industry called Upstream that deals with exploring natural reserves like oil and gas under the earth's crust. Exploration study involves high risk and money and the generation of a massive amount of data that need to be ingested into the various geoscience applications to gather valuable insights. Most oil industries adopt digital transformation (DT) that involves using emerging techniques to solve complex business problems associated with the seismic study. These digital solutions enhance the performance of existing software and workflows for seismic study as they use extensive cloud computing capabilities. Integrated cloud-based solutions in the oil and gas industry have the following advantages: (a) Significant reduction of

overall time to first well (which refers to the identification of the exact location of drilling), (b) High availability of seismic data (raw data in archival mode and processed data is ready to use mode) to all Geoscience applications, (c) Data integration (quick access to all structured/unstructured data), (d) High availability of Geo-Science and subsurface applications, and (f) Data Safety (tapes and hard disks are not physically safe).

Several works in the literature are focused on using a vast amount of data for computations in cloud computing [1][4] [8][11]. Works in [6][16] [18] uses the Azure cloud platform for analyzing the performance of cloud platforms in scenarios where extensive data is processed for computational results. Srirama et al. [14] built a platform called SciCloud deployed on the HPC computing platform for the researchers to use the resources for solving computationally intensive operations efficiently. Ekanayake et al. [10] designed a map-reduce framework called twister, an enhanced MapReduce runtime on Azure, which supports massive computations. I. Saeed et al. [19] showcase the different security measures implemented in storage services available in cloud providers. They have also compared the security measure in both the Amazon Web Service(AWS) cloud and Microsoft Azure Cloud. H. Dewan and Hansdah[20] provide details of storage services provided by three popular cloud provides and guidelines for using appropriate storage services in each cloud provider. E. Bocchi et al. [21] proposed a framework for testing cloud storage performance from a wide variety of storage providers from different cloud providers. M. Meena and Bharadi [22] talk about performance challenges in an application involving huge images and talk about a technique called CBIR to handle them in the cloud. P. Matri et al. [23] show how relevant is the usage of a blob in scenarios involving HPC and Big data in cloud platforms. They have shown how blob storage can be an effective replacement for HPC clusters using file system storage. While the above research papers present effective utilization of the

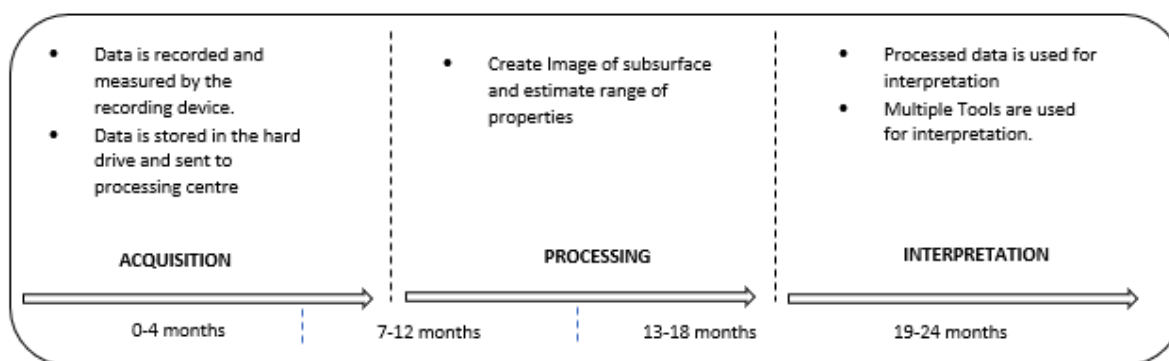


Figure 1. Steps in the Traditional Approach

computational capacity of the cloud, there is a need for a framework that supports:

1. High computational requirements
2. Effective utilization of the various storage tiers available in the cloud
3. Constraints related to data access protocol
4. Throughput.

The paper presents a novel solution for seismic data processing by utilizing cloud services related to storage, computational capacity, and access protocols. The following are the main contributions of the paper:

1. Proposed an integrated GeoCloud solution framework High-Performance Computing (HPC) architecture for seismic data processing that reduces time to first discovery by 50% (compared to traditional exploration life cycle).
2. Proposed cost-effective storage options for Simple Messaging Block (SMB) file access requirement of seismic data processing applications.
3. Introduced a novel approach for MD5 calculation for huge files using the MapReduce pattern on Hadoop clusters to reduce the overall verification process.
4. Presented methodologies to measure the efficiency of proposed data storage options.

The paper is organized in the following order. The background and motivation are described in the next subsequent section, along with the concepts used in Seismic data processing. Followed by the current process and challenges in seismic data processing. The later section presents the proposed framework to address the challenges. Test Bed and Experimental

Results are presented afterward, showing the performance and efficiency, and the last section concludes the paper.

2. Background and Motivation

The seismic study generates massive data (typically from 21 to 200 TB of data per week), which requires analysis. The study is carried out by sending seismic waves under the earth's surface, and the waves reflected are captured and analyzed. Suppose an area on earth is identified as a potential natural oil reserve. In that case, a seismic study is carried out in that area by dividing the area into blocks that are further broken down into multiple swaths (a broad strip of land or sea). Each swath can generate 7TB of data during the processing. The data is generated one swath at a time or can be generated simultaneously (based on exploration vendor). The data involved in this process contains structured and unstructured data, thus leading to big data (due to variety and volume). Out of the vast data generated, only a small chunk of filtered data can be used for further analysis. As raw data cannot be used, it must be refined, filtered, processed, and analyzed to generate *value*. Figure 1 explains the three necessary steps: acquisition, processing, and interpretation carried out in an exploratory study. The traditional approach usually takes up to 2 years to complete the entire cycle consisting of these three phases. Also, processing such massive data using traditional approaches is a challenge. It is impractical to provision machines or storage that could process such a massive volume of data. As a part of digital transformation, the seismic data generated in the exploration cycle is transferred to the on-premises environment for further processing. The following are critical challenges in the traditional approach.

1. **Data Availability:** The data should be available for the G&G (geological and geophysical) community in all the phases (refer to figure 1) of the exploration cycle. Existing traditional approaches use tape drives to transfer data from exploration sites to the on-premises environment. Tapes are then converted to usable digital format through a process called Tape Transcription. This process is time-consuming and poses an additional cost to the company.
2. **Tools:** Using terabytes of data in the cloud poses many challenges, especially to Commercial-off-the-shelf (COTS) geoscience applications such as Petrel, Eclipse, Promax. G&G users use these applications to analyze the exploration data for which the data has to be downloaded from the cloud into their machines. These applications require very high compute and disk throughput for efficient usage.
3. **Cost:** The use of high throughput disks in high computing platforms is costly in any public cloud, and moving to the cloud is not beneficial if cost concerns are not addressed.
4. **Data Integrity:** As the data moves through multiple stages in the exploration process, data must not be manipulated due to data movement errors, and care should be taken that integrity is maintained.

3. Seismic Data Processing: Current Process And Problems

Seismic analysis of an oil block typically takes approximately a year as it depends on both the acquisition partner's velocity and size of the exploration block. The four stages of seismic data are Data Acquisition, Processing, Interpretation, and Archival. Note that at each stage, the necessary throughput and the mode of data access can be different.

3.1 Phases of Seismic data processing

3.1.1 Data Acquisition: Seismic Data is generated in remote exploration blocks. Each exploration block is a combination of multiple swats. Acquisition partners use seismic and shock waves to record seismic data from each swat. Approximately 6 to 7 TB of data is generated per swat, and a complete swath analysis would take nearly two weeks. Seismic data accumulated

in this period will be approximately 13 TB. This generated data must be transferred to on-premise data centers for further processing and is usually done by storing and transferring the data in disks or tape drives, which involves significant time. There should be a process or methodology to consume this data, ingest, and return whenever and wherever required to decrease the overall time in the seismic data processing cycle.

3.1.2 Raw Data Load: Data received as-is from acquisition partners is referred to as raw data in traditional systems. This data is usually loaded into Network Access storage (NAS) drives, which have limited space capacity. Storing data in NAS brings additional challenges that affect seismic data analysis. These challenges are mainly due to the limitations of NAS drives, such as scalability and performance. In addition to this, there is a requirement to scale up and down the utilization capacity based on the size of data ingested into the system. The data stored on the NAS drive is shared between various team members where the users (typically Geology & Geoscience team) copy the data from the NAS drive to the local VMs, which they use, generate output, and transfer data to the shared drives so that other team members can access. If the users concurrently access the content in shared drives, there is a high chance of reducing the overall throughput. Along with this, there are several other problems, such as:

1. High cost for procuring the drives
2. Limitation to support only five concurrent users compared to the expectation of a minimum of 50 concurrent users in a real-time scenario.
3. No backup or disaster recovery solutions.

There is a requirement to provide a shared drive accessible over popular platforms such as Windows, Linux, and macOS to address the above problems.

3.1.3 Processing: Processing seismic data requires COTS tools like Permedia [13] and Geolite [9] to read and process the data. Usually, these tools are desktop applications requiring huge computing power and high throughput disks. The minimum memory required for these applications ranges from 100 GB to 150 GB, and the minimum throughput from local VM disks is expected to be 250 Mbps. Each of these tools has a diverse OS requirement, typically either Windows or Linux. Hence, every user maintains two machines with Windows and Linux operating systems and is used depending on the COTS tool OS requirement. Multiple computers for each user bring out additional

costs for hardware, software, and implementation as they are used only at specific times during the seismic data processing cycle. There is a requirement to set up a provisioning model where these machines are set up on a need basis and disposed of when the tool's usage is completed. This "pay as you go" model can be used for optimizing costs to the company.

3.1.4 Processed Data Copy: The output of this step is processed data. This data is generated in local disks of VMs and copied to shared drives for further processing.

3.1.5 Interpretation: The data from the previous step is fed into interpretation software to derive the location of the first well. This phase is highly critical as it determines the decision of continuing drilling operations at a prospective well. This phase also requires high computing machines (having both Windows and Linux operating systems) and COTS tools to interpret the seismic information.

3.1.6 Archival: After the data is interpreted and the well's location is identified, the project can be considered closed. The data generated in the entire cycle should be moved after a year to low-cost storage like tape drives for future reference and legal requirements. This action also frees up storage for a future project and reduces the storage cost. Every year, approximately 500 TB of seismic data is generated and is stored on-premises. Information retrieval, when needed, is a challenge if archived. Also, storing such massive historical data adds additional storage costs. There is a requirement for a system that can provide low-cost storage with the ease of retrieving content when needed.

3.2 Infrastructure Problems in Seismic Data Processing

Some of the activities in Seismic data processing consume a significant amount of CPU time to complete. Generating a processed file from raw data on a single machine can take several days to complete depending on the size and complexity of the data. There is a requirement to process the data on multiple high-end machines to speed up the process. Certain G&G(Geology and Geoscience) applications require the High-Performance Computing (HPC) framework for completing operations such as simulations, 3-D model generation for some geo-cellular models, and reservoir models. In traditional systems, the HPC

clusters are set up on-premises, which adds high costs for cluster setup and the scaling process. Moreover, these clusters are not frequently used. Hence, as a part of DT, setting up HPC clusters on the cloud is recommended. One of the key differences between the on-premises HPC framework and cloud HPC framework is the ability to add and remove resources when required dynamically. In this work, we present:

1. Cloud-based HPC cluster solution that can be used with G & G COTS applications having SMB data access requirements
2. Recommended HPC configurations for seismic data processing (refer to HPC Cluster Simulation).

3.3 Data Integrity Problems

Since data is transferred through multiple phases and storage locations in the seismic data processing cycle, there is a high risk of both data manipulation and data corruption during the process. One of the effective ways the data manipulation can be detected is by calculating the MD5 hash. Although MD5 has cryptographic vulnerabilities (mainly for digital signatures and password storage), MD5 hash is the most frequently used and preferred in the industry to verify data integrity for the following reasons [2].

1. Less computationally expensive when compared to other algorithms such as SHA-2
2. Faster and can generate millions of checksums very quickly.
3. Return accurate results for data integrity when it is well implemented.
4. Well suited when hashes are sent through a most secure channel.

Data integrity for files can be checked by calculating the MD5 hash for a file before transferring to the cloud. When the transfer is completed, the MD5 hash is recalculated and compared with MD5 generated before data transfer. In current systems, this process is time-consuming. Moreover, an MD5 checksum is not calculated for files larger than 4 MB files in cloud providers such as Azure. This paper presents an approach for MD5 calculation using the Map-Reduce pattern on Hadoop clusters by splitting huge files (in chunks of 4 MB) to evaluate the data integrity.

4 Proposed Integrated Geo Cloud Framework

This section addresses the drawbacks mentioned in section "Seismic Data Processing: Current Process And Problems" using an integrated Azure-based cloud solution and recommends a novel combination of technologies for each phase of the seismic data processing cycle.

1. **Data Acquisition:** Use of tools like MS DataBox, DataBox Disk, Azure Import Export to efficiently transfer data from exploration blocks to Microsoft (MS) data centers.
2. **Raw Data:** Use Azure blob storage with Hot Tier to address bandwidth requirement and replication as Geo-Redundant Storage (GRS) to address data redundancy and disaster recovery.
3. **Processing:** Use of NV series machines with the Graphical Processing Unit (GPU) capabilities for handling high resolution and high compute requirements.
4. **Processed Data:** Use of technologies like NetApp, Xendata, Azure File share to provide a high throughput of low-cost storage and GRS replication for disaster recovery. The setup will also have the following key features:
 - a. File system-based access to big data
 - b. High throughput with a minimum of 100 Mbps speed
 - c. Use of low-cost storage like Azure blob
 - d. Accessibility of shared data between both Linux and Windows systems.
5. **Interpretation:** Use of NV series machines as indicated in the processing layer.
6. **Archival:** Use of Azure Blob with Archival data tier and GRS replication.

The details for the above approaches are given below.

4.1 Storage Types

The following four storage types are recommended to ensure the most effective usage of storage and achieve optimal cost and throughput requirements.

1. **Type 1- User Drive:** This type of storage is dedicated storage provisioned as Managed

Solid State Drives (SSD) [3] disks attached to the Virtual Machines (VM). These disks are available as one of the disks (for example, E: drive) within a VM. They can be used by users to store raw seismic data and for processing using the required software. There are multiple managed disk options available, but P30 is recommended to provide a throughput of 250 Mbps. This storage option also facilitates file system-based access to data. As this disk is local to a user's VM, the VM's size can go up to 1 TB depending on the user requirement.

2. **Type 2 - Shared Drive:** This type of storage is shared storage provisioned for storing raw and processed data. This type of storage enables sharing data between teams during the different phases of the seismic data processing cycle. Typically, around 40-150TB of this storage is required during the different processing cycle phases. This storage is the most critical part of the architecture as this bypasses the concurrent users' access limitation (which is existing in the traditional system) by allowing up to 50 users to access the data concurrently. The storage also provides very high disk I/O. It supports high throughput with a minimum of 100 MB (even for 50 or more concurrent users).
3. **Type 3 - Azure Blob:** This storage uses Hot Block Blob [3] storage that holds raw data obtained from seismic acquisition partners. Data is first loaded into this type of storage until it is copied to the Type 2 storage used by teams to process seismic data. This type of storage option is also suitable for data not frequently used during the processing cycle for cost reduction.
4. **Type 4 - Archive Blob:** This type of storage uses Archive Block Blob [3] storage to hold legacy data. Typically, after the data processing is completed, the data may no longer be required. Old and unused content can be moved to archive storage to reduce storage costs.

4.2 Data Flow

Figure 2 presents the process flow between the different storage types during the seismic data processing cycle. The steps are:

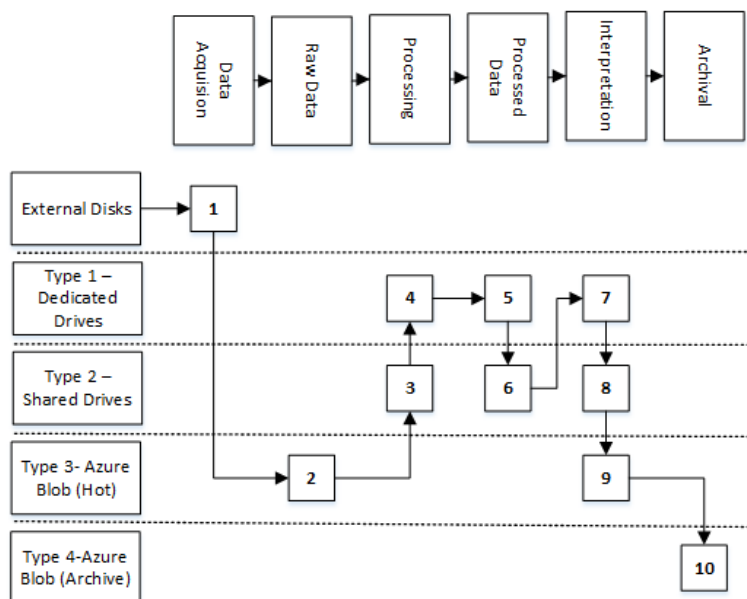


Figure 2. Data Flow in Seismic data processing cycle

1. Data from external partners is first loaded into external disks like MS Disk, MS Databox [7] and shipped to MS data center.
2. Data from an external device is first loaded into Azure Blob (Type 3) and then moved to Shared drives (Type 2).
3. From Shared drives (Type 2), data is copied to the Local drive (Type 1) for processing.
4. The processed data is moved back to Shared Drive (Type 2) so that other team members can access it for further processing.
5. Processed data from Shared drives (Type 2) will be moved again to Local Drives (Type 1) for interpretation.
6. Data generated from interpretation software is then moved to Shared Drives (Type 2) for

any discussion and further analysis (if required).

7. Once the analysis is completed, data is moved to Azure Blob (Type 3), and it is moved to Azure Blob (Type 4) for archiving. "Type 1", "Type 3", and "Type 4" storage options are standard in general. For "Type 2" (shared drive) storage option, we used (a) Azure Files, (b) Azure NetApp Files (ANF) and XenData.

4.3 HPC Cluster for Seismic Data Processing

Multiple seismic data processing scenarios involve large computations and require massive infrastructure to generate simulations. This process requires a high-performance computing setup to process massive data. In this section, we present a

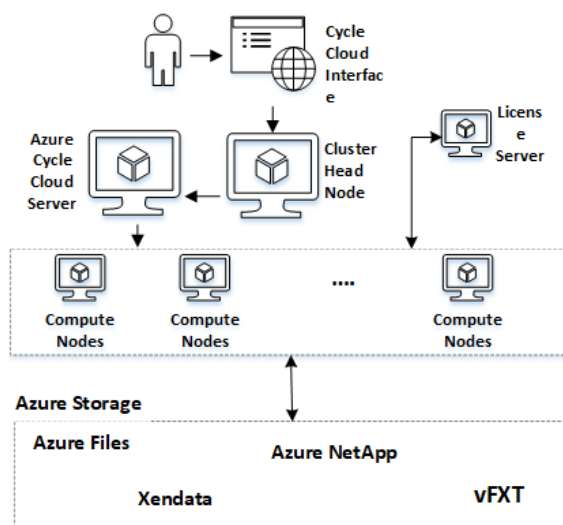


Figure 3. HPC Cluster setup

cloud-based HPC setup that enables efficient implementation of the processing scenarios. The advantage of the proposed method is the flexibility of upscaling and downscaling the compute nodes and swift infrastructure set up by using Azure Resource Manager templates that automate the setup and configuration of HPC components. Figure 3 shows the proposed HPC cluster setup using Azure. Following is the process flow of the proposed solution.

1. The user creates an HPC cluster consisting of head nodes and compute nodes using the Azure cycle cloud.
2. Jobs are created and configured to run in the head node to read the Azure storage data. The head node auto-scales the compute nodes depending on the number of jobs and the total cores allocated to the head node.
3. The license server should be set up in the same VNET as the compute nodes. The license server allocates licenses to the utilities, which are required for seismic data analysis.
4. The following two types of scripts are placed in network file shares (NFS) accessible for all the compute nodes.
 - a. Script files containing the business logic.
 - b. Script files for installing the required seismic utility.
5. The scripts are read from NFS and are uploaded into Compute nodes during the compute nodes creation.

6. Compute nodes are installed with the required seismic data processing utilities using scripts.
7. The compute nodes read the data from any of the proposed storage options and use it in the jobs for processing.
8. Post execution, the status of the jobs is written to the Azure cycle cloud monitoring system.
9. The alert system present in the Azure cycle cloud frequently checks for any alerts. It sends notifications to the user if required.

The key characteristics of the proposed approach are as below:

1. **Non-attended operation:** Since all the operations such as creating clusters, installing the required utilities, moving data from azure storage are done using commands and configuration files, this setup does not require any manual intervention.
2. **License Server Integration:** As there is a need for multiple utilities in the seismic data processing cycle that needs a license, the license server checks for every utility and provides it to the compute node when needed.
3. **Parallel Processing:** Since, end to end sequential processing of seismic data processing takes considerable time and also involves very high reads and writes, HPC parallel processing capability is used to cut down the overall time by processing multiple files in parallel. As the raw files are available over shared drives, parallel

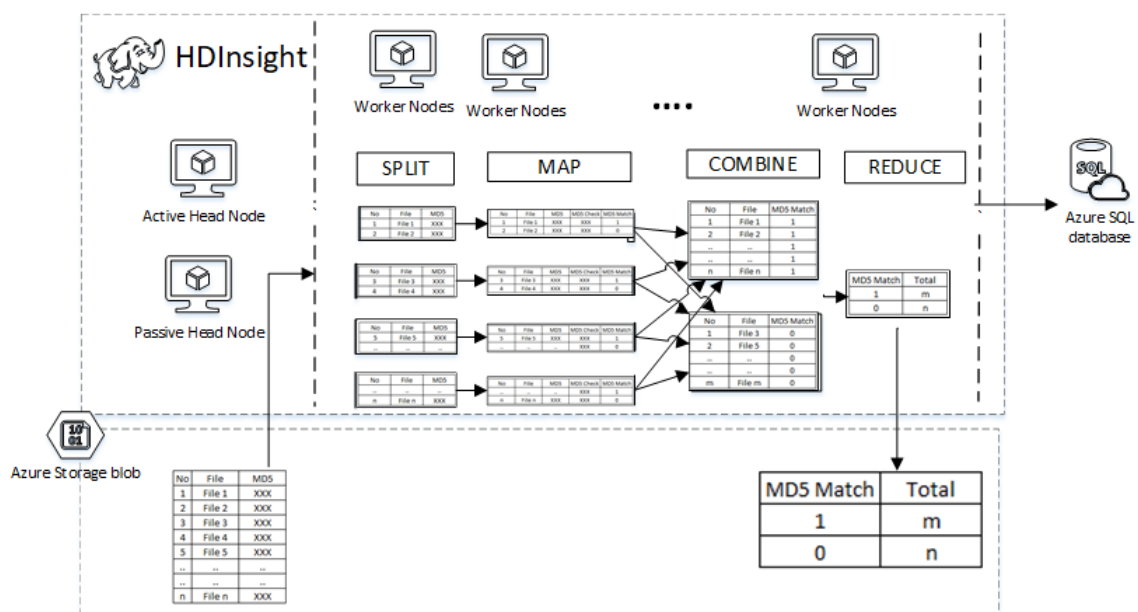


Figure 4. MD5 Calculation Flow

Procedure 1. Map Function for MD5 Hash Calculation

Input: Manifest file for split files containing Filename, MD5 hash values

Output: Split files updated with MD5 Hash Boolean value

```

1. While inputLine is not NULL do
2.     inputValues [] = inputLine. Split (,)
3.     /*Split input line by “,” */
4.     inputFile = inputValues [1]
5.     /*Get full file path in the inputFile field*/
6.     inputMD5 = inputValues [2]
7.     /*Get precalculated MD5 has of the inputFile */
8.     Download inputFile to local directory
9.     MD5Check = Calculate MD5(inputFile)
10. /*Set MD5 for the inputFile */
11. If (inputMD5 = MD5Check)
12.     Set MD5Match = 1
13. Else
14.     Set MD5Match = 0
15. Endif
16. Update inputFile, inputMD5, MD5Check and MD5Match value to SQL Database for future reference.
17. Write inputFile and MD5Match to STDOUT
18. end while

```

processing of the files can be done without any interdependency during the processing lifecycle stages.

4.4 MD5 calculation using Hadoop cluster

Seismic data is moved across multiple environments after it is generated in the exploration block until it reaches the cloud. First, the data is cataloged into files containing details such as File Name, Path, and MD5 hash, post which the data is shipped to the Microsoft data centers for data upload. Then the data is read from the data disk and is upload into the cloud environment. During this process, there is a high risk of intentional or unintentional data manipulation. One of the methods in which the data integrity can be checked is by computing the MD5 hash. This section discusses an MD5 hash computing technique based on the MapReduce pattern on the Hadoop cluster. Figure 4 shows the MD5 calculation flow. Usually, the MD5 hash is calculated for each file during the cataloging process. It is used as a reference to check the data integrity after moving the data to the cloud. When the data load is completed at the Data centers, the MD5 hash is calculated again and compared with the precalculated value to check the integrity. This process is both time and resource consuming

especially for big data. It mainly depends on the exploration center's compute capability. Also, as a data integrity check is mandatory during seismic data processing, it should take minimal time as possible. Figure 4 shows the flow of MD5 hash calculation using the MapReduce pattern on HD Insight Hadoop Cluster. HDInsight configuration involved creating two head nodes with 4 Cores and 240 GB RAM, and the six worker nodes are 8 Cores with 56 GB RAM. Mapper and Reducer modules were implemented in .NET framework 4.5 using Microsoft.Hadoop.Mapreduce library. The following are the detailed steps in the proposed process.

1. **Load input files:** Catalogue comma-separated file(.CSV) consisting of File Name, Path, and MD5 hash is loaded into HDFS. Then it is consumed by the Map-Reduce job.
2. **Split Input files:** The contents of the catalog file are split into multiple rows called input splits. The input splits will be the input to the map function.
3. **Map:** The Map phase is the first phase in the execution of the map-reduce program. In this phase, the data in each split is passed to the mapping function. The mapping function's main functionality is to download the split

Procedure 2. Reducer function for MD5 hash calculation

Input: Filename, MD5 hash values

Output: Total count showing total number of matches and mismatches

1. **while** inputKey is not NULL **do**
2. Group Values by inputKey
3. Write inputKey,Count(Values) to STDOUT separated by tab.
4. **end while**

file from the blob and calculate the MD5 hash. A Boolean flag called "*MD5 Match*" is set to *true* if the precalculated MD5 hash is matched with the MD5 hash calculated by the Map function; otherwise, it is set to *false*. Both the recalculated MD5 hash and Boolean flag values are sent to the next phase, the reducer function. The results are also written to the SQL database for future reference. Procedure 1 shows the steps in the map function for MD5 calculation.

4. **Combine:** This step consumes the Map phase's output and consolidates the output based on the MD5 match value from the Map phase. This phase's output is the results segregated into two groups, where the first group contains the value that is marked as '1' and another group where the value is marked as '0'.
5. **Reduce:** In this step, the output values are read from the Combine phase and are aggregated based on the MD5 match value. The output of this phase is the total count showing total matched and unmatched files. Procedure 2 shows the steps in the reduce function for MD5 calculation.

5 Test Bed and Experimental Results

The test bed was created using the Azure cloud to show the viability of the proposed framework. We present performance test results for the three proposed Type 2 – Shared Drive storage options that are implemented using Azure files[7], NetApp [7], and XenData [17].

1. **Azure Files:** Azure files are managed file share services provided by Microsoft [7]. The service was configured to provide SMB protocol endpoints that can be used for mounting in Windows machines. Premium Azure Files service was used to provide very high throughput since SSD drives support it in the backend.

2. **XenData:** Xendata[17] is a cost-effective and secure cloud-based edge software in which the files are stored in cloud object storage and exposed as named drives in windows machines. Xendata file shares were configured with the back end as block services on cloud VMs. Feature for frequently used file caching was turned on for fast access to locally cached data.
3. **Azure NetApp:** ANF is a NetApp service offering provided by Microsoft. ANF account was configured with single storage pools of fixed capacity and a single shared volume for the experiment. ANF Volumes network share address was added to VMs as shared drives. We have simulated results using the premium tier since it offers better throughput with optimal cost.

Benchmarking tests were done on the different proposed storages to measure the throughput/download times with different user loads. We performed experiments using Azure Batch to spawn multiple VMs to simulate simultaneous users. Multiple tasks were triggered to simulate the data transfer between the various storage options, and the results of the transfer were pushed into App Insight[7]. The upload and download speeds were captured for each storage type. Table 1 shows the total data transfer time between different storage options (refer to figure 2). The data is collected for ten concurrent users (wherever it is applicable). It can be observed that the time taken to move the data from disk to Azure blob is always constant, i.e., 1-2 days as the upload is done at the Microsoft data centers. The time taken for the disk shipment procedure is not considered here. It can be observed that for data size less than 10 GB, all the proposed options took almost a similar time. For the extensive data, i.e., data size more than 100GB, NetApp took less time when compared to the other two storage options. On average, NetApp is 15% faster when compared to the other two proposed storage options. From the results, it can be concluded that any of the three proposed storage options can be used for data less

Table 1. Table showing the average time taken (in mins) to move the files between storage options for 10 to 50 concurrent users

Type	Size (GB)	Data Transfer Time (in Minutes)				Total Users
		2	5	10	100	
	MS Disk to Type 3 - Azure Blob	1- 2 days	1- 2 days	1- 2 days	1- 2 days	NA
Azure Net App	Type 3 - Azure Blob to Type 2 - Shared Drive	< 1	< 1	1 to 2	9 to 11	NA
	Type 2 - Shared Drive to Type 1 - Dedicated Drive	< 1	1 to 2	1 to 2	20 to 25	10
	Type 1 - Dedicated Drive to Type 2 - Shared Drive	< 1	< 1	1 to 2	9 to 11	10
	Type 2 - Shared Drive to Type 3 - Azure Blob	< 1	< 1	1 to 2	9 to 11	NA
Azure Files	Type 3 - Azure Blob to Type 2 - Shared Drive	< 1	< 1	1 to 2	11 to 13	NA
	Type 2 - Shared Drive to Type 1 - Dedicated Drive	< 1	1 to 2	1 to 3	19 to 21	10
	Type 1 - Dedicated Drive to Type 2 - Shared Drive	< 1	< 1	1 to 2	11 to 13	10
	Type 2 - Shared Drive to Type 3 - Azure Blob	< 1	1 to 2	2 to 4	12 to 14	10
Xendata	Type 3 - Azure Blob to Type 2 - Shared Drive	< 1	< 1	1 to 2	10 to 12	NA
	Type 2 - Shared Drive to Type 1 - Dedicated Drive	< 1	1 to 3	1 to 3	22 to 26	10
	Type 1 - Dedicated Drive to Type 2 - Shared Drive	< 1	< 1	1 to 2	10 to 11	10
	Type 2 - Shared Drive to Type 3 - Azure Blob	< 1	< 1	1 to 2	10 to 13	NA
	Type 3 - Azure Blob to Type 4 - Archive Blob	< 1	< 1	< 1	< 5	NA

than 100 GB. However, NetApp is preferable for data, which is more than 100 GB.

5.1 Efficiency Tests for Type 2 Data Storage-Concurrency

Table 2 shows the throughput values of the type 2 storage option performed in a DS13_v2[3] server having a mounted disk of 1 TB. Tests were performed for varying numbers of concurrent users, i.e., 1, 5, and 10. Experiments were carried using software called CrystalDiskMark, which measures the efficiency of storage. The table captures the four read-write modes. The provisioned capacity for NetApp and Azure Files in the experiment was 32 TB premium tier, and the dedicated drive was P30 with 1 TB premium storage.

1. SEQ1MQ8T1: Simulates a sequential 1 MiB read and write operation with a read-write queue length of 8 and 1 thread.
2. SEQ1MQ1T1: Simulates a sequential 1 MiB read and write operation with a queue length of 1 and thread size of 1.
3. RND4K-Q32T16: Simulates a random read and writes of 4 KiB data sizes with a queue length of 32 and a total of 16 threads.
4. RND4K-Q1T1: Simulates a random read and writes of 4 KiB data sizes with a 1 and 1 thread queue length.

It can be observed that throughput for all the nodes is almost the same for small data size and small queue length. NetApp performs the best for bigger file sizes

Table 2. Table showing throughput of type 2 storage in the proposed solution

Solution	Number of Users	SEQ1M-Q8T1 READ	SEQ1M-Q8T1 WRITE	SEQ1M-Q1T1 READ	SEQ1M-Q1T1 WRITE	RND4K-Q32T16 READ	RND4K-Q32T16 WRITE	RND4K-Q1T1 READ	RND4K-Q1T1 WRITE
Tier 1 - Dedicated Drive	1	248.97	265.16	270.13	255.87	135.02	63.24	60.41	55.26
Type 2 - Azure NetApp File	1	1190.95	752.1	320.66	436.39	376.22	366.67	19.45	13.33
	5	827.35	602.238	324.252	290.648	363.856	294.656	18.268	12.822
	10	517.897	420.996	262.651	186.201	340.66	185.956	15.229	12.039
Type 2 - Azure Files	1	1190.95	752.1	320.66	436.39	376.22	366.67	19.45	13.33
	5	978.4	676.4	346.56	364.6	256.3	245.2	17.4	10.4
	10	756.7	567.2	235.67	245.7	256.2	154.6	14.67	7.4
Type 2 - Xendata	1	536.25	1803.99	540.21	1823.44	34.71	116.99	45.5	111.51
	5	456.67	1503.3	456.2	1634.5	31.4	135.5	40.4	98.4
	10	345.1	1456.3	346.6	1456.2	27.3	94.5	36.6	90.5

and queue length, and Xendata should be scaled up to match the throughput of NetApp and File Shares.

5.2 Performance Tests Setup for "Type 2" Data Storage

In this section, the procedure for performance testing of "Type 2" Storage is described.

5.2.1 Azure Files: 32TB of cloud storage was provisioned in the Premium tier to test the Azure Files' performance. The throughput of Azure Files was expected to increase propositionally depending on the provisioned capacity [7]. Figure 5 represents the proposed approach using Azure application insights, Azure Batch service, and Azure Files. The below approach was followed to test the performance of Azure files

Azure Files storage was created with a total storage capacity of 32 TB capacity in the premium tier. The premium tier was chosen to achieve maximum throughput and IOPS, with an optimal cost for the experiment.

1. Fifty folders were created in Azure Files storage.
2. Files with a total size of 32 TB were taken for the experiment. The data was split into random sizes ranging from 100 MB to 5 GB and was loaded into the folders created in the Azure Files storage.
3. A pool of 50 DS14_V2 VMs [3] with a P30 Premium disk VM was created using the Azure Batch service. Each VM in the pool

was configured with PowerShell scripts having functionalities for download and upload.

4. PowerShell commands were executed through Azure Batch jobs to download the files into VM's named drive from Azure Files storage folders.
5. The total time taken for the copying was noted using the PowerShell script, which was further used to calculate the throughput.
6. Metrics such as time taken and throughput were logged into Azure App Insight.

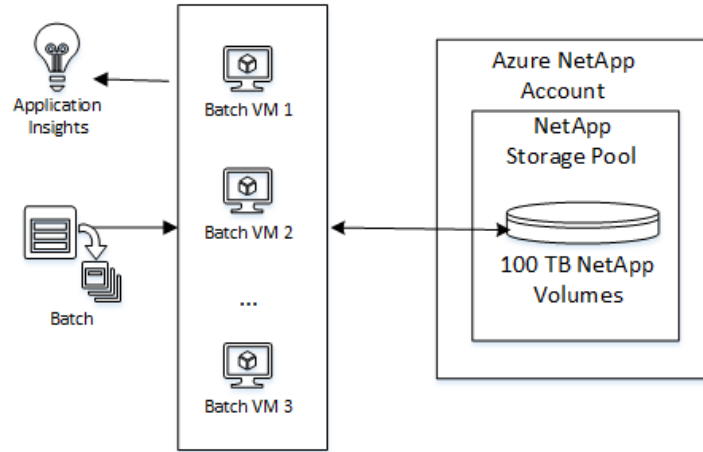


Figure 6. Azure NetApp Testing Procedures

7. After download functionality, upload functionality was triggered using the PowerShell scripts to upload the files to Azure Files from Azure Batch VMs.

5.2.2 NetApp Files: Figure 6 represents the Azure NetApp Files(ANF) file setup for testing the performance of ANF.

1. Azure NetApp Files volume having a capacity of 32 TB is created in a NetApp storage pool hosted in Azure NetApp Account.
2. The same set of procedures used for testing the Azure file share performance (refer section Performance Tests Setup for Type 2 Data Storage/Azure Files) were followed for calculating the time required for uploading

and downloading the files from NetApp volume.

5.2.3 Xendata: Figure 7 represents the XenData setup that was used in the experiment

1. Single VM(DS13_v2, 8 Core, 56 GB RAM, 1 TB P30 HDD) containing Xendata setup(referred to as Xendata cluster) with Cloud gateway software was configured with Azure Hot Blob storage.
2. The Azure Load balancer was set up with port 445 to enable secure SMB protocol. The VM was configured behind a load balancer to enable the flexibility of adding multiple VMs for read-only loads.
3. DNS is set up for the load balancer to provide a user-friendly name to the share exposed by the Xendata cluster.

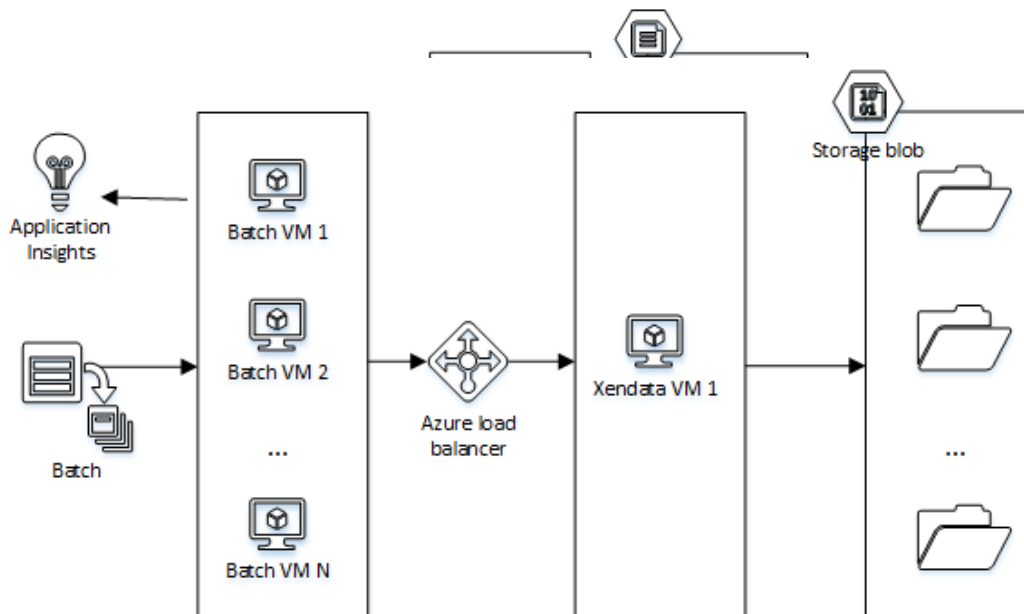


Figure 7. Xendata Testing Procedures

Table 3. Performance results for the storage options

	NetApp	Azure Files	XenData
Time taken by batch process (in mins)	12	11	13.2
Average Throughput MiB/s (Read/Write)	250.7	201	81.3
Average time taken for download files	21.42	18.4	27.83
Average time taken for uploading files	15.42	17.4	20.83
Average Time taken for the overall Process (in mins)	66.83	60.48	80.12

4. A batch pool of 50 DS14_V2 VMs with a P30 Premium disk VM was created using Azure Batch service to achieve max throughput. Each VM in the pool was loaded with PowerShell scripts having file download and upload functionalities.
5. The network share path(from step 3) was configured in individual batch VMs to upload and download to the batch VM.
6. The same procedures used for testing Azure File service using Azure Batch VMs were followed for calculating the time taken for uploading and downloading files from Xendata VM.
7. The throughput provided by this setup will be similar to the throughput provided by premium disks configured within the VM.
8. Additional time was required for downloading the files from the Azure blob to the local VM before servicing the content to the requested user.

5.3 Performance Test Results of Type 2 Storage Options

Table 3 shows the performance test results for the proposed storage options. It can be observed that batch creation took almost identical time as it was created by azure service. Throughput was calculated using a tool called Vdbench [15], which is a command-line utility for storage benchmarking. The block size was taken as 8 KiB. It can be observed that Azure NetApp has the highest average values for throughput. The average upload and download time using the storage options are also presented in the table. It can also be seen that the overall time taken is less for NetApp when compared to the other storage options

5.4 Cost Analysis

Figure 8 shows the cost comparison between the proposed approach and the traditional approach. The chart's primary axis shows the cost incurred by the proposed storage options for various data sizes, starting from 100 GB to 500 GB. The secondary axis shows the cost incurred by the traditional approach.

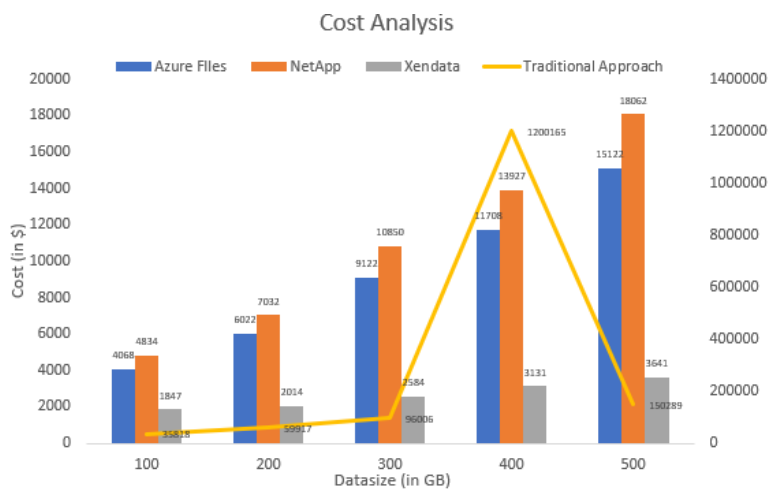


Figure 8: Cost Analysis

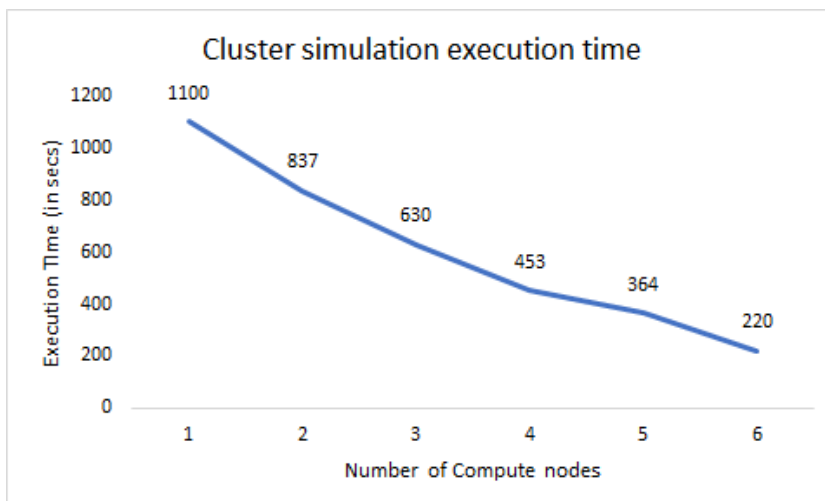


Figure 9. HPC Cluster simulation time for OPM

From the comparison of the proposed options, it can be found that Xendata has the lowest cost out of all the three options for all the data sizes. On average, there is more than 97.5 percentage of cost savings when any of the proposed "Type 2" Storage options are used when compared to the traditional approach.

5.5 HPC Cluster Simulation

HPC clusters in Azure were created using the Azure cycle cloud. The clusters created consisted of Linux CentOS 7.6 head node running on PBS Pro 19.1, which is open-source software for optimizing the workload in HPC environments. The memory of the head node was 240GB. Six compute nodes were created with CentOS 7.6 HB60rs. The configuration chosen for the compute node was Standard_HB60rs

machines having 60 cores CPU, 240 GB memory, and 100 Gbps RDMA network. The cluster setup was done in the Azure South Central (United States) region. A software called Open Porous Media (OPM) [12] was installed in all the compute nodes.

OPM is an open-source project for the development of applications for modeling and simulations for porous media data. The installation of the software was done using scripts. The scripts and the installable software were placed in the 4TB network file share (NFS) space. The "norne" dataset used for the simulation was downloaded from the OPM Open dataset [12]. The dataset was then uploaded into Azure storage. The dataset consisted of real-time simulations related to 49 wells, three exploration units for an oil field in the Norwegian

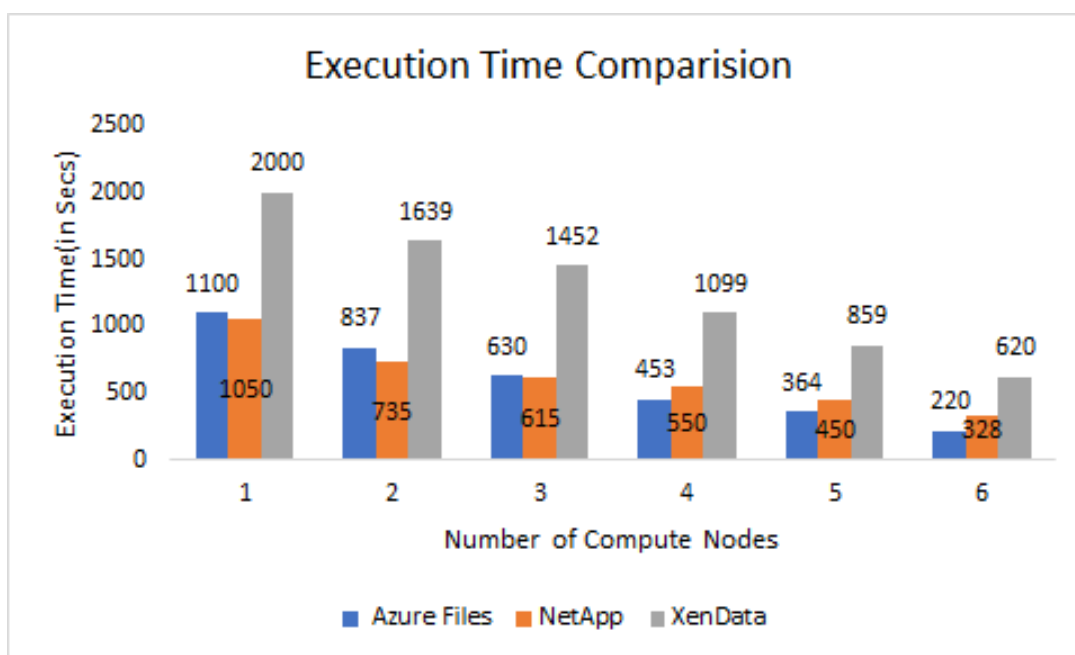


Figure 10. Execution Time Comparison

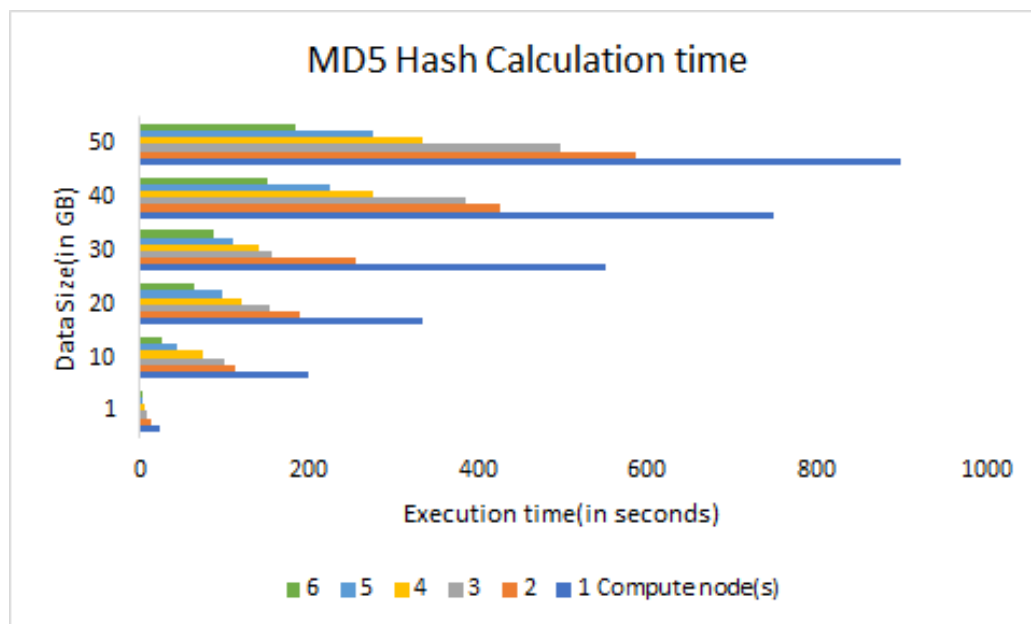


Figure 11. MD5 Hash calculation time on HPC clusters

sea. Simulations were carried using a variable number of compute nodes starting from 1 to 6. Figure 9 shows the average execution time calculated for ten simulations by varying the number of compute nodes. NetApp was preferred out of three storage options to read the data from azure storage as it provided better throughput when compared to the other two. It can be observed that the clusters scaled as expected. Average simulation time was also measured when data was read from Azure files, NetApp, and Xendata. Figure 10 shows the comparison of execution times between the three storage options. It can be observed that access using NetApp had taken a comparatively less average time of 600.6 seconds when compared to the other two options, Azure files (621.3 seconds) and Xendata (1278.16 seconds).

5.6 MD5 Hash Calculation

Tests were performed to calculate the overall execution time for calculating the MD5 hash value for dataset size ranging from 1GB to 50 GB. The dataset was taken from a famous seismic data processing company. The clusters were created on Azure HDInsight. Azure HDInsight is a platform offered by Azure to execute open-source frameworks such as Apache Hadoop, Spark, and Kafka. The clusters consisted of 2 head nodes and a variable number of compute nodes starting from 1 and later upgraded to 6. The clusters use Azure storage as the cluster storage. The head nodes were 4 Cores with 240 GB RAM, and the worker nodes

were 8 Cores with 56 GB RAM. The clusters were created with Linux CentOS as the operating system. Figure 11 shows the overall execution time for the worker nodes starting from 1 to 6. It can be observed that the overall execution time decreased as the number of compute nodes is scaled-up.

6 Conclusion

Many companies that deal with enormous data like seismic data processing today have embarked on a Digital Transformation journey. Digital transformation can be brought about by moving existing data into the cloud and implementing new practices within the cloud computing platform. This paper talks about the digital transformation in the Oil industries for seismic data processing and data management through integrated cloud-based solutions. This paper proposes an innovative solution for data storage of seismic data throughout the project's life cycle. Additionally, it suggests a setup for the HPC framework to process seismic data involving COTS tools and a MapReduce pattern on a Hadoop cluster for MD5 hash calculation. All the solutions are deployed on Azure (a popular cloud-based platform). The data storage option indicated in the paper has significant cost benefits reducing an overall cost reduction of more than 90% of the traditional cost by using edge software like Xendata and also a reduction in overall integrity check by using the inventive approach discussed in the paper

References

- [1] A. Thakar and A. Szalay, "Migrating a (large) science database to the cloud". In Proceedings of the 19th ACM International Symposium on High-Performance Distributed Computing. HPDC '10, (2010), pp. 430–434.
- [2] D. Cao and B. Yang, "Design and implementation for MD5-based data integrity checking system", In Proceedings of the 2nd IEEE International Conference on Information Management and Engineering, Chengdu, (2010), pp. 608-611.
- [3] Azure documentation | Microsoft Docs. <https://docs.microsoft.com/en-us/azure/>.
- [4] C. A. Lee, "A perspective on scientific cloud computing". In Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing. HPDC '10 (2010), pp. 451–459).
- [5] CrystalDiskMark – Crystal Dew World. <https://crystalmark.info/en/software/crystaldiskmark/>.
- [6] D. Kossmann, T. Kraska, and S. Loesing, "An evaluation of alternative architectures for transaction processing in the cloud", In Proceedings of the 2010 international conference on Management of Data. SIGMOD '10, (2010), pp. 579–590.
- [7] Directory of Azure Cloud Services | Microsoft Azure. <https://azure.microsoft.com/en-in/services/>.
- [8] G. Turcu, I. Foster, and S. Nestorov, "Reshaping text data for efficient processing on amazon ec2", In Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing. HPDC '10, (2010), pp. 435–444.
- [9] GEO Lite for the Wellsite Geologist Geologix. <https://www.geologix.com/geolite/>.
- [10] Jaliya Ekanayake, Hui Li, Bingjing Zhang, Thilina Gunarathne, Seung-Hee Bae, Judy Qiu, and Geoffrey Fox, "Twister: a runtime for iterative MapReduce", In Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing. HPDC '10, (2010), pp. 810–818.
- [11] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D.A Patterson, A. Rabkin, I. Stoica, and M. Zaharia, Above the clouds: A Berkeley view of cloud computing. Electrical Engineering and Computer Sciences Department, University of California, Technical Report, (2009) UCB/EECS-2009-28.
- [12] OPM | The Open Porous Media Initiative. <https://opm-project.org/>.
- [13] Permedia Petroleum Systems Modeling. <https://www.landmark.solutions/Permedia-Petroleum-Systems-Modeling>.
- [14] S. Srirama, O. Batrashev, and E. Vainikko, Scicloud: Scientific computing on the cloud, In Proceedings of 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing. CCGRID '10, (2010), pp. 579–580.
- [15] Vdbench: A storage benchmarking tool. <https://opensourceforu.com/2016/07/vdbench-storage-benchmarking-tool/>.
- [16] W. Lu, J. Jackson, J. Ekanayake, R. Barga, and N. Araujo, "Performing large science experiments on Azure: Pitfalls and solutions", In Proceedings of 2010 IEEE Second International Conference on Cloud Computing Technology and Science, (2010), pp. 209–217.
- [17] XenData Hybrid Cloud Storage Overview. <https://xendata.com/xendata-hybrid-cloud-storage/>.
- [18] Z. Hill, J. Li, M. Mao, A. Ruiz-Alvarez, and M. Humphrey, Early observations on the performance of windows azure. Scientific Programming, vol. 19, (2010), pp. 367-376.
- [19] I. Saeed, S. Baras and H. Hajjdiab, "Security and Privacy of AWS S3 and Azure Blob Storage Services," 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), Singapore, (2019), pp. 388-394
- [20] H. Dewan and R. C. Hansdah, "A Survey of Cloud Storage Facilities," 2011 IEEE World Congress on Services, Washington, DC, USA, (2011), pp. 224-231
- [21] E. Bocchi, M. Mellia and S. Sarni, "Cloud storage service benchmarking: Methodologies and experimentations," 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet), Luxembourg, Luxembourg, (2014), pp. 395-400.
- [22] M. Meena and V. A. Bharadi, "Performance analysis of cloud based software as a service (SaaS) model on public and hybrid cloud," 2016 Symposium on Colossal Data Analysis and Networking (CDAN), Indore, India, (2016), pp. 1-6.
- [23] P. Matri, Y. Alforov, Á. Brandon, M. Kuhn, P. Carns and T. Ludwig, "Could Blobs Fuel Storage-Based Convergence Between HPC and Big Data?," 2017 IEEE International Conference on Cluster Computing (CLUSTER), Honolulu, HI, USA, (2017), pp. 81-86