Smart Data Placement for Big Data Pipelines: An Approach based on the Storage-as-a-Service Model

Akif Quddus Khan*, Nikolay Nikolov†, Mihhail Matskin‡, Radu Prodan§
Hui Song†, Dumitru Roman†, and Ahmet Soylu‖
*Norwegian University of Science and Technology, Norway
akif.q.khan@ntnu.no
†SINTEF Digital, Norway
‡ KTH Royal Institute of Technology, Sweden
§ University of Klagenfurt, Austria
‖ OsloMet – Oslo Metropolitan University, Norway

***Abstract*—The development of big data pipelines is a challenging task, especially when data storage is considered as part of the data pipelines. Local storage is expensive, hard to maintain, comes with several challenges (e.g., data availability, data security, and backup). The use of cloud storage, i.e., Storage-as-a-Service (StaaS), instead of local storage has the potential of providing more flexibility in terms of such as scalability, fault tolerance, and availability. In this paper, we propose a generic approach to integrate StaaS with data pipelines, i.e., computation on an on-premise server or on a specific cloud, but integration with StaaS, and develop a ranking method for available storage options based on five key parameters: cost, proximity, network performance, the impact of server-side encryption, and user weights. The evaluation carried out demonstrates the effectiveness of the proposed approach in terms of data transfer performance and the feasibility of dynamic selection of a storage option based on four primary user scenarios.**

***Index Terms*—Storage-as-a-service, big data pipelines, data locality, data placement strategies, software containers**

I. Introduction

Big data pipelines are designed to support one or more of the three big data features commonly known as the three Vs (volume, velocity, and variety), while processing data through a series of data processing steps. The implementation of a big data pipeline includes several aspects of the computing continuum such as computing resources, data transmission channels, triggers, data transfer methods, integration of message queues, etc., making the design and implementation process difficult. This process becomes even more complex if a data pipeline is coupled to data storage, such as a distributed file system, which comes with additional challenges such as data maintenance, security, scalability, etc. [1]. Cloud storage systems (e.g., Amazon S3, Elastic Block Store, or EBS, Azure Blob Storage, Google Cloud Storage) offer very large storage with high fault tolerance, addressing several big data related storage concerns [2]. Moving data to cloud storage, i.e., Storage-as-a-service (StaaS), moves the extra overhead of data redundancy, backup, scalability, security, etc. to the cloud service provider.

In this respect, the objective of this paper is to demonstrate that the integration of StaaS with big data pipelines is a promising direction. This necessitates a one-of-a-kind solution for data pipeline design and a method for real-time data placement with unknown data volumes, availability, location, data security, etc. constraints. To this end, we first propose an approach to realize big data pipelines with hybrid infrastructure, i.e., computation on an on-premise server or on a specific cloud, but integration with StaaS; and secondly develop a ranking method to find the most suitable storage facility dynamically based on the user's requirements, including cost, proximity, network performance, impact of server-side encryption, and user weights [3].

The rest of the paper is organized as follows. Section II provides the related work. Section III presents the proposed approach and ranking method. Section IV provides an evaluation, while Section V concludes the paper.

II. Related Work

The scientific community has extensively acknowledged the necessity to use cloud computing to execute scientific workflows/pipelines [4]. Many studies investigated and demonstrated the viability of employing cloud computing for deploying big data pipelines in terms of both cost [5] and performance [6].

Abouelhoda et al. propose Tavaxy, a system that enables seamless integration of the Taverna system with Galaxy processes based on hierarchical workflows and workflow patterns [7]. Wang et al. [8] present early results and experiences in enabling interaction between Kepler SWFMS and the EC2 cloud. Antonio et al. [9] analyses hybrid multi-cloud storage systems and different data transfer techniques in general. These approaches discuss the benefits and possibilities of deploying big data pipelines in cloud infrastructure; however, they do not discuss the possibility of hybrid big data pipelines in a multi-cloud environment.

Zhang et al. [10] describes BerryStore, a distributed object storage system suited for the storing of huge quantities of small files. In a large Web application, file sharing generates a large number of requests. BerryStore is built to manage these requests. The essential insight is that extraneous disk operations should be avoided when reading metadata. The

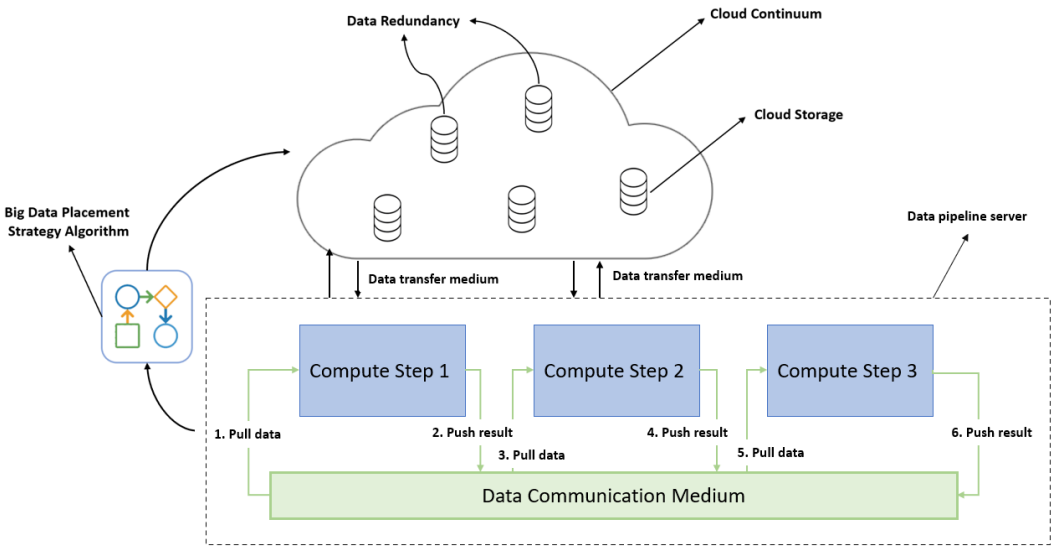

Fig. 1. Proposed approach for Big data pipelines with StaaS.

proposed mechanism does not provide any support for integration with the big data pipelines. Yuan et al. [11] studied the unique characteristics of scientific cloud operations and proposed a clustering data placement strategy capable of dynamically moving application data across data centers based on dependencies. Simulations on their cloud workflow system SwinDeW-C shown that their data placement strategy may significantly reduce data traffic during the execution of the process. The suggested system is heavily platform dependant since it only works for Hadoop. There are several other data placement techniques developed for Hadoop [12]–[14].

Er-dun et al. [15] addresses the issues connected with scientific workflows in the cloud computing environment, specifically the load balancing of datacentres. After examining the storage capacity of data centers, data transit patterns, and datacentre loads, they came with a workable data placement strategy using a genetic algorithm. In compared to other data placement strategies, the genetic algorithm-based data placement methodology performs well in terms of data center load balancing and data movement volume. While the proposed technique is effective, it does not address any functional requirements from the owners or developers of big data pipelines. It is limited to the number of datasets and the number of movements.

Our proposed approach incorporates multi-cloud storage providers, focuses on big data pipelines, provides results dynamically, and is platform independent.

III. Integration of Data Pipelines and StaaS

The proposed approach is shown in Figure 1, where the compute steps of a data pipeline are encapsulated in software containers as suggested by [16], and deployed on a server. A communication medium is setup for inter-step communication, for example pulling and pushing information about different pipeline events (trigger events) and execution results. The local

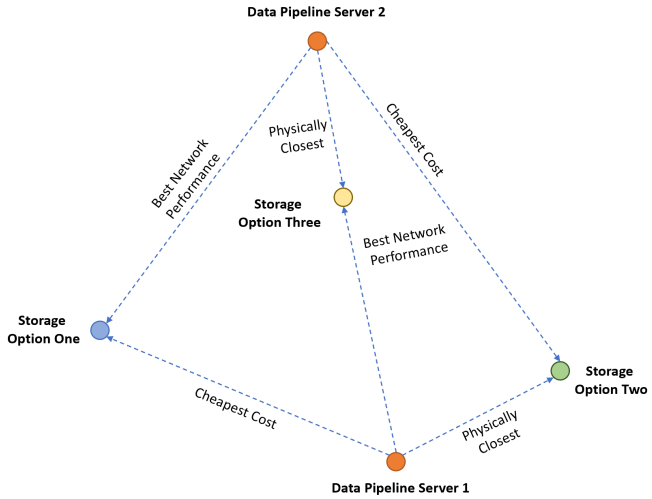

Fig. 2. Relation between cost, network performance, and physical distance.

storage, in this proposed approach, is replaced with hybrid cloud storage as a service. To reduce the cost of network egress in the case of on-premise computation servers, the inter-step data storage concept is also introduced. Data placement method finds the most suitable storage facility to store data from data pipeline server using a ranking method.

There are several parameters that affects the choice of cloud storage such as the cost of storage space, security, performance, and the location. These parameters are interdependent on each other. So there is a possibility that the cloud service provider that fulfills security requirements does not have the best network performance, or the one with closest to the data pipeline server is more expensive. This scenario is visually represented in Figure 2. The focus of the proposed method

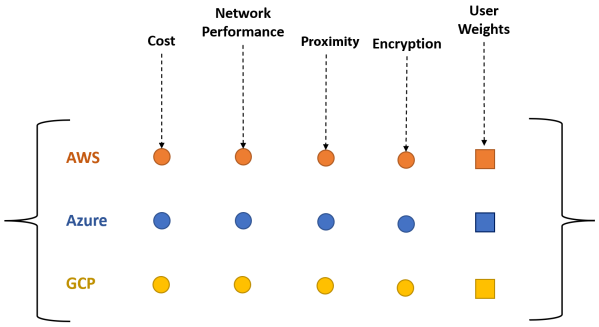

Fig. 3. Evaluation matrix; each row is allocated to a cloud service provider and column represents the parameters.

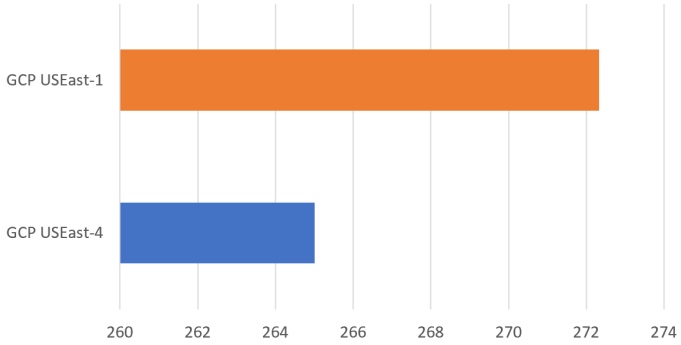

Fig. 4. Google region pair comparison in parallel execution. The x-axis shows the number of seconds taken.

is mainly on data locality, that is smart data placement to achieve maximum performance output. But different locations with different cloud storage providers have different costs. In addition to that, all data centers have different network infrastructure, so that network performance can vary between different data centers. The challenge is to decide which criteria is best suited to the situation and user requirements.

VIKOR, a multi-criteria decision analysis (MCDA) method [17], is used to rank different scenarios based on the user weights given by the experts or decision makers. VIKOR method is selected based on the framework developed by Jankowski et al. [18]. It is an innovative tool for choosing the MCDA approach that is most appropriate for the decision issue. For this very evaluation model, four different parameters are selected in addition to the user weights. These parameters are as follows: cost (i.e., based on storage, bandwidth, and READ and WRITE operations – see [3]), proximity (i.e., using IP ranges provided by the cloud service providers and GeoIP), network performance (i.e., throughput), the impact of server-side encryption (i.e., performance). Figure 3 shows the evaluation matrix for the proposed method. Values for the parameters are calculated using the independent software tools we developed. Server-side encryption is implemented on the stored data in each cloud storage provider and it's affect is tested on the performance. The last column shows the weights set by the user. The evaluation matrix explained above is given as an input to the MCDA VIKOR algorithm, and the output is the ranking of the cloud service providers.

IV. Evaluation

We evaluated our approach by comparing the data transfer performance of the storage option selected by our method against to the best guess region and demonstrated the feasibility of dynamic selection storage options based on four primary user scenarios. A data pipeline is deployed and tested with all the above mentioned characteristics. Five parameters are used as evaluation criteria to rank different cloud storage options based on the user's requirements and software tools are developed accordingly. The results from each software tool are then put into the evaluation matrix and used as input for the ranking method. For this paper, we have considered three cloud storage providers: Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP).

A. Performance evaluation

A data pipeline is deployed in the USEast region. The data pipeline is then integrated with six different storage facilities. Two from AWS, two from Azure, and two from GCP. Since the pipeline server is located in the USEast region, one can choose between available regions, *us-east-1* to *us-east-4*. Our method suggested the *us-east-4* region instead of the first region, with the sole wight on the proximity. We setup storage on both region pairs and tested the performance. A data pipeline with a total of 3,601 WRITE operations were carried out, with a total size of 3.2 Megabytes (MB). To avoid temporary downtime, the operation was repeated three times. Execution time with region suggested outranks the other. The results for parallel execution for GCP is shown in Figure 4.

B. Scenarios

Underlying workload characteristics play an important role for the selection of storage options, as there is often a trade-off. The application portfolio needs to be considered as a whole, as well as individually. The following are the user scenarios those are served well by the public cloud. Requirements are given for each scenario in terms of space (GB), bandwidth (GB), and the number of write and read operations.

Scenario 1 - Temporary requirements (Req.:1000, 15000, 5000, 5000): As the cloud uses a pay-as-you-go, utility-based pricing model, it is well-suited to short-term and transient workloads and projects. Example use cases are proof of concepts, pilots, application testing, etc. Table I ranks the cloud providers when a user puts equal weights for all parameters, that is 25% for cost, proximity, the impact of encryption, and network performance. Based on these input and the momentary conditions, GCP is ranked first by the algorithm[1].

[1]S and R represent the maximum utility of the majority and a minimum individual regret of the opponent respectively. Q integrates R and S for a compromise solution [17].

TABLE I
SCENARIO 1: TEMPORARY REQUIREMENTS

Rank	Alternatives	Si	Ri	Qi
1	AWS	0.25	0.15	0
2	GCP	0.28	0.25	0.52
3	Azure	0.75	0.25	1

Scenario 2 - Highly variable workloads (Req.: 2000, 25000, 10000, 10000): Demand variability comes in two distinct flavours: predictable (seasonal, cyclical, etc.) and unpredictable, e.g., month-end processing and on-season vs. off-season. Table II ranks the cloud providers when a user puts 70% weight on the network performance, 10% on the cost, 10% on the proximity, and 10% on the impact of encryption. In this case, cost is a secondary factor, and network performance has priority. Based on these input and the momentary conditions, Azure is ranked first by the algorithm.

TABLE II
SCENARIO 2: HIGHLY VARIABLE WORKLOADS

Rank	Alternatives	Si	Ri	Qi
1	Azure	0.3	0.1	0
2	AWS	0.45	0.42	0.45
3	GCP	0.71	0.70	0

Scenario 3 - High security, low scale/volume solutions (Req.: 1000, 15000, 5000, 5000): Although many customers fear the absence of security of the cloud, there are many capabilities within cloud storage to restrict and monitor access to resources. Table III ranks the cloud service providers when a user puts 80% weight on the impact of encryption, 10% on network performance, and only 5% on the cost and proximity. Based on these input and the momentary conditions, GCP is ranked first by the algorithm.

TABLE III
SCENARIO 3: HIGH SECURITY, LOW SCALE/VOLUME SOLUTIONS

Rank	Alternatives	Si	Ri	Qi
1	AWS	0.08	0.06	0
2	GCP	0.10	0.10	0.04
3	Azure	0.90	0.80	1

Scenario 4 - Dormant workloads (Req.: 5000, 1000, 2000, 2000): A dormant workload occupies no compute capacity and generates no network traffic, reducing the running costs to just storage. Example use cases are test/development, user acceptance testing, unit and system testing, etc. Table IV ranks the cloud service providers when a user puts 70% weight on cost, 10% on proximity, network performance, and the impact of encryption. Based on these input and the momentary conditions, AWS is ranked first by the algorithm.

TABLE IV
SCENARIO 4: DORMANT WORKLOADS

Rank	Alternatives	Si	Ri	Qi
1	GCP	0.11	0.1	0.01
2	AWS	0.14	0.08	0.02
3	Azure	0.90	0.70	1

V. CONCLUSIONS

We proposed a generic approach for implementing big data pipelines with StaaS integration. It allows on-premise processing and on-cloud and local storage temporarily for inter-step data input and output. We tested our approach in terms of the data transfer performance and demonstrated its feasibility through four different representative scenarios. Regarding the future work, more parameters could be added into the evaluation matrix. The results of the evaluation matrix could also be compared against actual decision makers.

Acknowledgments. Partially funded by enRichMyData (HE 101070284), DataCloud (H2020 101016835), BigDataMine (NFR 309691), and SINTEF SEP-DataPipes.

REFERENCES

[1] S. Robinson and R. Ferguson, "The storage and transfer challenges of big data," *MIT Sloan Management Review*, vol. 7, 2012.
[2] C. Yang *et al.*, "Redefining the possibility of digital earth and geosciences with spatial cloud computing," *International Journal of Digital Earth*, vol. 6, no. 4, pp. 297–312, 2013.
[3] A. Q. Khan, "Smart data placement for big data pipelines with storage-as-a-service integration," Master's thesis, Norwegian University of Science and Technology, Norway, 2022.
[4] Y. Zhao *et al.*, "Opportunities and challenges in running scientific workflows on the cloud," in *Proc. of the CyberC 2011*. IEEE, 2011, pp. 455–462.
[5] E. Deelman *et al.*, "The cost of doing science on the cloud: the montage example," in *Proc. of the SC 2008*. IEEE, 2008, pp. 1–12.
[6] A. Iosup *et al.*, "Performance analysis of cloud computing services for many-tasks scientific computing," *IEEE Transactions on Parallel and Distributed systems*, vol. 22, no. 6, pp. 931–945, 2011.
[7] M. Abouelhoda *et al.*, "Tavaxy: Integrating taverna and galaxy workflows with cloud computing support," *BMC Bioinformatics*, vol. 13, no. 1, pp. 1–19, 2012.
[8] J. Wang and I. Altintas, "Early cloud experiences with the kepler scientific workflow system," *Procedia Computer Science*, vol. 9, pp. 1630–1634, 2012.
[9] A. Celesti *et al.*, "Towards hybrid multi-cloud storage systems: understanding how to perform data transfer," *Big Data Research*, vol. 16, pp. 1–17, 2019.
[10] Y. Zhang *et al.*, "A novel solution of distributed file storage for cloud service," in *Proc. of the COMPSACW 2012*. IEEE, 2012, pp. 26–31.
[11] D. Yuan *et al.*, "A data placement strategy in scientific cloud workflows," *Future Generation Computer Systems*, vol. 26, no. 8, pp. 1200–1214, 2010.
[12] C.-W. Lee *et al.*, "A dynamic data placement strategy for hadoop in heterogeneous environments," *Big Data Research*, vol. 1, pp. 14–22, 2014.
[13] L. Wei-Wei, "An improved data placement strategy for hadoop," *Journal of South China University of Technology (Natural Science Edition)*, vol. 1, p. 028, 2012.
[14] J. Xie *et al.*, "Improving mapreduce performance through data placement in heterogeneous hadoop clusters," in *Proc. of the IPDPSW 2012*. IEEE, 2010, pp. 1–9.
[15] Z. Er-Dun *et al.*, "A data placement strategy based on genetic algorithm for scientific workflows," in *Proc. of the CIS 2012*. IEEE, 2012, pp. 146–149.
[16] A.-A. Corodescu *et al.*, "Big data workflows: Locality-aware orchestration using software containers," *Sensors*, vol. 21, no. 24, p. 8212, 2021.
[17] A. Ishizaka and P. Nemery, *Multi-criteria decision analysis: methods and software*. John Wiley & Sons, 2013.
[18] J. Wątrobski *et al.*, "Generalised framework for multi-criteria method selection," *Omega*, vol. 86, pp. 107–124, 2019.