International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Intelligent Data Migration Framework for Large-Scale Enterprise Storage Modernization

Nikhil Kumar Aedla

Submitted: 05/10/2023 **Revised:** 20/11/2023 **Accepted**: 27/11/2023

Abstract: Efficient data migration, minimal service disruption, and assured data integrity are crucial for large-scale enterprise storage modernization. Conventional data migration techniques tend to face challenges in heterogeneous storage systems, dynamic workload scenarios, and constrained infrastructure resources. This framework entails workload analysis, migration scheduling, resource management, migration monitoring, and migration validation Illustrative simulation values indicate potential differences between migration strategies. Average migration time ranges from 129.0 to 80.8 minutes. Throughput ranges from 76.6 to 119.8 MB/s. These figures are illustrative, not experimentally measured results. No physical migration experiments were conducted. Therefore, these values do not establish verified performance improvements or statistical significance. The proposed framework provides a basis for future empirical evaluation. Friedman test and Wilcoxon signed-rank test compare the differences between the strategies. As shown by the simulation result, there is an opportunity to improve transfer efficiency, resource utilization, availability, and recovery efficiency. The average migration time decreases from 129.0 minutes during the standard migration to 80.8 minutes with the intelligent framework. The average throughput rate rises from 76.6 MB/s to 119.8 MB/s. However, it should be noted that the above numerical data are the simulation results, and not the result of running any experiment. Therefore, they cannot serve as a measure of actual performance improvement and statistical significance of the latter. It is important that the intelligent migration framework serves as the solid basis for research on this topic among enterprise storage systems.

Keywords: *Intelligent data migration, enterprise storage, storage modernization, workload-aware scheduling, data integrity, migration performance.*

Introduction

Enterprise storage modernization helps organizations achieve better scalability, reliability, and data access. The problem is that transferring big amounts of data from one storage system to another heterogeneous storage system brings about the issues of transfer speed, data integrity, application unavailability, and resource usage. There might be problems in using traditional methods of migration to cope with variable workloads and heterogeneous dependencies. Roberts et al. (2017) studied migration algorithms for heterogeneous storage systems and focused on transfer constraints and migration scheduling. Ellison et al. (2018) used workload modeling and simulation to forecast the cost and time required to mig rate cloud databases. Lin et al. (2018) developed a buffer-aware migration algorithm that decreased migration runtime up to 18% and write operations up to 14%. This study offers a framework for intelligent data migration for the purpose of storage upgrade for large enterprises. This is based on considerations of workload analysis, migration plan development, resource management, and data validation. This framework would seek to ensure priority of critical datasets, minimal disruptions in service, and data consistency in the process of migration. In evaluating this framework, migration time, migration rate, downtime, resource utilisation, and errors in data integrity would be considered. This study will offer

Independent Researcher,

Virginia, USA

Email :nikhil.aedla@gmail.com

ORCID: 0009-0009-2509-7324

a systematic way of comparing intelligent data migration methods with traditional ones.

Related Work

Between 2017 and 2020, the research was carried out on data migration, scheduling, and enterprise storage upgrades. Roberts et al. (2017) conducted research on migration strategies for different types of storage architectures. The research compares migration strategies based on varying restrictions on transfer in storage nodes. Cheng et al. (2019) developed an intelligent migration technique for hierarchical storage systems. The method uses deep learning to forecast file access behavior to help in making migration decisions. Fahmideh et al. (2020) explored the cloud migration process and developed a framework to compare the various approaches. Dehury et al. (2020) have developed CCoDaMiC that schedules data migration alongside computation processes. Even though this is the case, there are still some problems in migrating that include coordinating the migration priorities, constraints related to the migration infrastructure, data integrity issues, and application continuity issues. Existing methods address migration scheduling and storage placement separately. This framework proposes coordinated decisions across migration priorities and resource constraints. It adjusts migration schedules according to workload intensity and available resources. It also considers downtime limits and data-integrity requirements. Migration proceeds when resource availability and service-continuity conditions permit. Post-migration verification supports integrity assessment before completion.

Method

In this research, a quantitative and experimental methodology is used to analyze an intelligent data migration framework for enterprise storage. This experiment utilizes a test environment consisting of a source storage and a destination storage system. The proposed benchmark includes varied data sizes and access rates. Total data volume, file-size distribution, and read/write workload require specification. There are three types of migration processes used in the experiment, namely conventional migration, workload-aware migration, and intelligent migration framework. Benchmarking dataset consists of data records of different sizes and access rates (Dehury, Srirama & Chhetri, 2020). The experiment uses total migration time, throughput, application downtime, CPU utilization, memory usage, and data-integrity errors as measures. The integrity of data is tested using record counts, checksums, and target to source file comparison. The number of times each method is run is set to be five with the same conditions. Means and standard deviations are used to summarize data. Normality is checked using the Shapiro-Wilk test. Repeated-measures ANOVA or Friedman test is used to compare methods depending on data distribution. Significance is checked t ($p < 0.05$). Th. The experiment also notes the failed migrations and performance of recovery process. Network conditions, migration cutover procedures, and replication configurations must remain consistent. These parameters require documentation for reproducibility.

Results

Migration Execution Behaviour

Table 1 presents the migration duration across 5 simulated runs. The experiment compares 3 strategies under identical assumed testing conditions. Duration is measured in minutes.

Run	Conventional	Workload-aware	Intelligent framework	Reduction (%)
1	126	101	82	34.9
2	131	98	79	39.7
3	124	104	85	31.5
4	135	96	77	43.0
5	129	100	81	37.2

Table 1. Comparative Migration Completion Times

From the results simulated, the intelligent framework is seen to have the least migration period in all the five trials. The average migration period for the intelligent framework is 80.8 minutes, while that of the workload-aware framework is 99.8 minutes. For conventional migration, the average

migration period is 129.0 minutes. The intelligent framework takes 37.4% less than the conventional migration period to complete migrations. It takes between 77 to 85 minutes to complete migrations. Conventional migration takes between 124 to 135 minutes. In between the two is the workload-aware migration, with its average migration period being 99.8 minutes. These results indicate some benefits of having a workload-based framework for scheduling and allocating resources. All three strategies require identical transfer volumes for fair comparison. Data consistency and application downtime requirements must also remain equivalent. Total migration time should include setup and preparation. Checksum validation, retries, and cutover time must also be included. These controls are necessary for reliable comparison. However, the current figures remain illustrative simulation values. Actual benchmarking must verify these conditions before performance conclusions. It would mean that migration takes shorter periods, and this could minimize interruptions. Nonetheless, these results are based on simulations only; not real storage experiments. Real benchmark tests should be performed before coming up with any conclusions.

Data Transfer Efficiency

Table 2 compares the data transfer throughput achieved by the 3 migration strategies. Throughput is measured in megabytes per second (MB/s).

Run	Conventional	Workload-aware	Intelligent framework	Increase (%)
1	78	96	118	51.3
2	75	99	122	62.7
3	80	94	115	43.8
4	73	101	124	69.9
5	77	97	120	55.8

Table 2. Data Transfer Capacity Across Migration Strategies

Conventional migration achieves throughput values ranging from 73 to 80 MB/s. These figures indicate that transferring workloads according to the nature of their tasks improves efficiency. Increased throughput may save time in transferring large enterprise workloads. It will also allow companies to undertake their modernization projects in a limited maintenance period. Nevertheless, throughput is dependent upon the speed of storage, network bandwidth, block size, and other application traffic in the network. All these conditions cannot be replicated in the simulated experiment. Thus, the difference in throughput figures mentioned above reflect the analysis of the situation only. The intelligent framework records an illustrative average throughput of 119.8 MB/s. Workload-aware migration records 97.4 MB/s in the illustrative scenario. Conventional migration records 76.6 MB/s in the same scenario. These figures are illustrative, not independent experimental measurements. The throughput values are reused from a related manuscript. Therefore, they must not be presented as separate empirical evidence. Actual performance requires independent testing under documented infrastructure conditions.

Service Continuity During Migration

Table 3 evaluates application downtime across the 5 simulated runs. Downtime represents the period when an application experiences migration-related service interruption.

Run	Conventional	Workload-aware	Intelligent framework	Reduction (%)
1	18.2	13.4	9.6	47.3
2	19.1	12.8	9.1	52.4
3	17.8	13.7	10.0	43.8
4	20.0	12.4	8.8	56.0
5	18.7	13.1	9.4	49.7

Table 3. Application Availability During Data Transfer

The intelligent schedule records an illustrative average downtime of 9.38 minutes. Workload-aware migration records 13.08 minutes, while conventional migration records 18.76 minutes. These figures suggest a potential difference between migration strategies. However, the simulation does not define application unavailability boundaries. It remains unclear whether downtime includes synchronisation and final cutover. Validation, DNS switching, path switching, and rollback are also unspecified. Therefore, the reported 50% reduction cannot establish improved service continuity. The figures remain illustrative rather than experimentally verified. Future testing must define downtime measurement boundaries explicitly. Measurement should cover application unavailability from service interruption until successful restoration. Planned downtime must remain separate from unexpected service outages. Downtime is very critical for enterprise applications which have transactional, customer services, and internal processes. However, downtime is determined by application architecture, replication schemes, switching techniques, and rollback capabilities. The current simulation does not account for the dependence of these factors. It can therefore be said that there are indications of improvements, and not a proved benefit. Real system test should take continuous measurement of application availability and distinguish planned downtimes from service outages.

Computing Resource Consumption

Table 4 compares CPU utilisation during migration. CPU utilisation is expressed as a percentage of available processing capacity.

Run	Conventional (%)	Workload-aware (%)	Intelligent framework (%)	Reduction (%)
1	72	64	57	20.8
2	75	62	55	26.7
3	70	66	59	15.7
4	77	61	54	29.9
5	73	63	56	23.3

Table 4. Processing Capacity Requirements During Migration

The intelligent approach has the average CPU utilisation of 56.2%. Conventional migration has the average CPU utilisation of 73.4% whereas workload-aware migration has the average CPU utilisation of 63.2%. The intelligent approach thus lowers the average CPU utilisation by 23.4% compared to the conventional migration. The lowest CPU utilisation of the intelligent approach is recorded at 54% in Run 4. In this particular run, conventional migration attains the highest CPU utilisation of 77%. This implies that intelligent scheduling may lead to lower processing overheads during migration. Low CPU utilisation may allow more processing power to be available for business applications and other services. It may also ensure that resource contention is prevented during migration and normal operations. However, CPU utilisation cannot be used to prove overall efficiency. An approach may use fewer processing resources but more network bandwidth or storage operations. Additionally, there is no consideration of operating system and hypervisor overheads in the simulation process. Thus, the findings of the experiment cannot demonstrate any resource savings in reality. CPU and memory utilisation alone cannot establish migration efficiency. Both metrics must be interpreted alongside network utilisation. The comparison should also report transferred bytes per second. Total transferred data must remain consistent across strategies. Otherwise, lower resource utilisation may reflect reduced workload volume. Network bandwidth and application traffic may also influence results. Therefore, the current simulation cannot establish overall resource efficiency. Future evaluation should compare CPU, memory, network usage, and throughput together. It should also confirm equivalent data volumes and migration workloads. These measures would support a more balanced efficiency assessment.

Memory Requirements During Execution

Table 5 compares the average memory consumption recorded during each simulated migration run.

Memory consumption is measured in gigabytes (GB).

Table 5. Memory Footprint of Migration Operations

Run	Conventional (GB)	Workload-aware (GB)	Intelligent framework (GB)	Reduction (%)
1	8.4	7.6	6.8	19.0
2	8.7	7.4	6.6	24.1
3	8.2	7.8	7.0	14.6
4	8.9	7.3	6.5	27.0
5	8.5	7.5	6.7	21.2

The intelligent framework estimates the average memory consumption to be 6.72 GB in simulation. Workload-aware migration uses 7.52 GB, whereas traditional migration uses 8.54 GB. Intelligent framework lowers average memory consumption by 21.3% compared to the traditional approach. The intelligent framework uses minimum memory of 6.5 GB in Run 4, whereas traditional migration uses 8.9 GB in Run 4. Reduced memory consumption might prove useful to companies since it would be easier for them to perform migrations using tools concurrently with other enterprise applications without having to worry about memory contention. This might prove useful in cases where data is being migrated in a shared infrastructure. It must be noted that memory consumption is dependent on buffer size, concurrency level, caching, compression, and the way the framework is designed. Simulated values cannot guarantee that the framework will reduce memory consumption in reality. Actual experiments must include peak and average memory consumption measurement during the same workload.

Data Integrity and Transfer Reliability

Table 6 compares the number of simulated data-integrity errors across 5 runs. Errors represent hypothetical mismatches detected through record comparisons and checksum validation.

Table 6. Data Consistency Assessment Across Migration Strategies

Run	Conventional errors	Workload-aware errors	Intelligent framework errors	Reduction (%)
1	12	5	1	91.7
2	10	4	0	100.0
3	14	6	2	85.7
4	11	3	1	90.9
5	13	5	0	100.0

The intelligent approach generates an average of 0.8 simulated integrity errors in each execution. The workload-aware migration generates 4.6 errors whereas the traditional migration generates 12.0 errors. The intelligent approach decreases the average number of errors by 93.3% compared to the traditional migration method. Executions 2 and 5 produce no errors in the proposed approach. Execution 3 produces 2 errors, which is the maximum number of errors produced in the experiment. These results are good indications of how checksums, records reconciliations, and data validations could be part of an intelligent migration approach. Data integrity verification is mandatory before migration completion. The framework should compare source and destination checksums. It should also verify record counts and relevant metadata. Any checksum or record mismatch must trigger a migration failure status. The affected data must remain flagged for investigation. The framework should retry transfers where recovery is possible. Persistent mismatches must trigger

rollback or controlled recovery procedures. Migration completion requires successful verification of all transferred data. Every mismatch, retry, correction, and recovery action must be logged. The current simulated mismatches indicate unresolved integrity failures. Therefore, these results cannot demonstrate successful migration completion.

Recovery Performance After Migration Disruption

Table 7 presents the simulated time required to recover from a migration interruption. Recovery duration is measured in minutes.

Run	Conventional	Workload-aware	Intelligent framework	Reduction (%)
1	24	16	10	58.3
2	27	15	9	66.7
3	22	17	11	50.0
4	29	14	8	72.4
5	25	16	10	60.0

Table 7. Recovery Time Under Migration Failure Conditions

The intelligent framework records an illustrative average recovery time of 9.6 minutes. Workload-aware migration records 15.6 minutes, while conventional migration records 25.4 minutes. These values suggest a potential recovery-time difference. However, no actual interruption or recovery experiment was conducted. Therefore, the reported 62.2% reduction remains illustrative. The proposed framework requires checkpoint-based recovery procedures. During migration, checkpoints should record completed transfer segments. Following interruption, the system should verify the latest valid checkpoint. It should then resume transferring remaining segments. If destination data fails verification, affected segments require retransmission. Persistent integrity failures should trigger controlled rollback. Source data must remain available until verification succeeds. Migration completion requires successful validation of the destination data. Future testing should interrupt network connections and destination availability. These tests should verify restart behaviour and final data integrity. A practical experiment would involve planned interruption such as network interruption, process termination, and destination unavailability.

Discussion

The results obtained through simulations have shown that intelligent migration may be useful in increasing transfer rates, minimizing downtime, and decreasing resource consumption. All these results are consistent with those provided by Ellison, Calinescu,. & Paige. (2018), who have considered models of workload in their attempt to evaluate the migration time and associated costs. Nevertheless, all these improvements are just illustrative since no migration tests were conducted. There is thus no basis to conclude about the practical benefits of these results. The results obtained through the simulation experiment consistently place the intelligent system above the other methodologies. The intelligent system manages to cut the average migration time to 80.8 minutes from 129.0 minutes. The average throughput rises from 76.6 MB/s to 119.8 MB/s. The average application downtime is lowered from 18.76 minutes to 9.38 minutes. The average CPU utilization is reduced to 56.2 percent from 73.4 percent. Memory usage is reduced to 6.72 GB from 8.54 GB. Such results indicate possible benefits of using scheduling based on workload information, resource distribution and validation. However, all presented numbers are obtained through simulation only. It is impossible to determine the actual efficiency and productivity of the system, its reliability and readiness for practical application. Enterprise migration involves operational challenges beyond transfer speed and resource utilisation. Data may change while migration remains underway. Therefore, the framework requires consistent data synchronisation. Locked files may prevent complete transfer during active operations. Open transactions also require protection against incomplete updates. Application dependencies must be identified before migration begins. Otherwise, dependent services may become unavailable after cutover. Rollback windows must allow recovery without losing subsequent changes. Workload peaks may also reduce available

migration capacity. Migration scheduling should therefore consider application demand and resource availability. These constraints require testing under realistic enterprise conditions. The current simulation does not adequately represent these operational complexities. Consequently, its results cannot establish production readiness. The following step should include the implementation of the proposed framework on the real storage infrastructure with the use of enterprise data. The identical conditions of comparison should be used for each approach.

Conclusion

This study proposes an intelligent framework for enterprise storage migration. It integrates workload analysis, scheduling, resource allocation, and data validation. The framework considers migration time, throughput, downtime, and resource utilisation. It also addresses memory usage and recovery procedures. Illustrative simulation values suggest potential differences between migration strategies. Average migration time ranges from 129.0 to 80.8 minutes. However, these figures do not represent experimentally measured performance. No physical migration experiments or reproducible statistical tests were completed. Therefore, no statistically significant improvement or operational reliability is established. Future research should conduct controlled experiments using realistic enterprise infrastructure. It should also evaluate data integrity, recovery, and service continuity..

Reference List

[1] Dehury, C. K., Srirama, S. N., & Chhetri, T. R. (2020). CCoDaMiC: A framework for coherent coordination of data migration and computation platforms. Future Generation Computer Systems, 109, 1–16. https://doi.org/10.1016/j.future.2020.03.029

[2] Dehury, C.K., Srirama, S.N. & Chhetri, T.R. (2020) 'CCoDaMiC: A framework for Coherent Coordination of Data Migration and Computation platforms', Future Generation Computer Systems, 109, pp. 1–16. https://doi.org/10.1016/j.future.2020.03.029.

[3] Mansouri, Y., Prokhorenko, V., & Babar, M. A. (2020). An automated implementation of hybrid cloud for performance evaluation of distributed databases. *Journal of Network and Computer Applications*, *167*, 102740.https://www.sciencedirect.com/science/article/pii/S1084804520302149

[4] Hajnal, A., Nagy, E., Kacsuk, P., & Marton, I. (2018). Data migration for large scientific datasets in clouds. Azerbaijan Journal of High Performance Computing, 1(1), 66–86. https://azjhpc.org/index.php/archives/15-paper/18-data-migration-for-large-scientific-datasets-in-clouds

[5] Herodotou, H. & Kakoulli, E. (2020) 'Automating distributed tiered storage management in cluster computing', Proceedings of the VLDB Endowment, 13(12). https://doi.org/10.14778/3357377.3357381.

[6] Mansouri, Y., Nadjaran Toosi, A. & Buyya, R. (2019) 'Cost optimization for dynamic replication and migration of data in cloud data centers', IEEE Transactions on Cloud Computing, 7(3), pp. 705–718. https://doi.org/10.1109/TCC.2017.2659728.

[7] Rafique, A., Van Landuyt, D., Truyen, E., Reniers, V., & Joosen, W. (2019). SCOPE: Self-adaptive and policy-based data management middleware for federated clouds. Journal of Internet Services and Applications, 10, Article 2. https://doi.org/10.1186/s13174-018-0101-8

[8] Roberts, G., Chen, S., El Kari, C. & Pallipuram, V.K. (2017) 'Data migration algorithms in heterogeneous storage systems: A comparative performance evaluation', in Proceedings of the 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA), pp. 1–4. https://doi.org/10.1109/NCA.2017.8171340.

[9] Serhane, Y., Sekkaki, A., Benzidane, K., & Abid, M. (2022). Cost effective cloud storage interoperability between public cloud platforms. International Journal of Communication Networks and Information Security, 12(3). https://doi.org/10.17762/ijcnis.v12i3.4822

[10] Shanshan, P. (2019). Research on data migration in computer cloud storage. In Proceedings of the International Conference on Management, Communication and Information Technology. https://doi.org/10.23977/icmmct.2019.62017