# Platform Migration: Data Centers to Cloud Architectures

**Prajwal V. Atreyas[1], S. Yamuna[2], Pramod Khadse[3], and S.B. Prapulla[4]**

[1]Student, RV College of Engineering, Bengaluru, Karnataka, India
[2]Manager Data Engineering, Mast Global Technologies, Bengaluru, Karnataka, India
[3]Senior Manager Data Engineering, Mast Global Technologies, Bengaluru, Karnataka, India
[4]Assistant Professor, RV College of Engineering, Bengaluru, Karnataka, India
*Email: [1]prajwal.va2008@gmail.com, [2]ys@mast.com, [3]pkhadse@mast.com and [4]prapullasb@rvce.edu.in*

***Abstract* —** With the current trend where organizations are moving towards cloud services and hybrid cloud technologies, the objective of this study is to develop a seamless data pipeline to perform data integration as part of platform migration, i.e. from data centers to cloud architecture. The proposed methodology is to implement these jobs by employing the Extract-Transform-Load (ETL) procedures to develop interfaces in Talend Open Studio, viz., a data integration tool. First, the data is extracted from multiple sources, such as, databases and flat files. Then, multiple transformations such as filtering, sorting and joining are done on the data. Finally, the transformed data is loaded into the staging tables of the Enterprise Data Warehouse. This is achieved by migrating the interfaces from the tool currently in use, IBM Infosphere DataStage, to re-create the functionalities. The comparison between the features of the two tools, Talend and DataStage, resulted in the identification of the pros and cons of each tool. It was inferred that Talend is equivalent to DataStage in most of the cases but with enhancements and tweaks in Talend, the execution time of few interfaces were reduced by half.

***Keywords* —** Talend, DataStage, Cloud migration, Data integration, ETL.

## I. INTRODUCTION

By 2022, more than 90% of global organizations will be using hybrid cloud, according to a March 2020 estimate. According to the study in [1] with 50 CIOs (chief information officer) conducted in April 2020, the proportion of total workload done on-premises was predicted to reduce from 59% in 2019 to 38% in 2021, a 41% decrease. Most businesses have adapted to this trend by transitioning from data-center focused architectures, in which all data is stored in-house and monitored or outsourced to a data center provider, to cloud-based designs, in which all data is stored and managed by harnessing the power of the cloud. This trend has been widely generalized and termed as 'Platform Migration'. Because of the long-term benefits of cloud architectures, platform migration is now a requirement for any organization. Platform migration is critical, since it is a prerequisite for modernizing or consolidating server and storage infrastructure, as well as incorporating data-intensive applications like databases, data warehouses, and data lakes, as well as, large-scale virtualization schemes [2].

Cloud native migration refers to moving an application and accompanying data to the cloud with few or no changes. Applications are refactored from their current surroundings and deployed to a new hosting location, i.e. the cloud. As a result, there are usually no major changes to the application design, data flow, or authentication processes [3].

Hybrid cloud approach to platform migration is the current trend. Instead of becoming cloud native, which has the disadvantage of being non-portable, meaning once the code is refactored to one specific cloud it is difficult to port to a different cloud. Hybrid cloud solves this issue by employing cloud-aligned architectures. Companies are looking into distributing the workloads across different platforms and different cloud service providers to utilize the best of each of them [4].

## II. LITERATURE REVIEW

Organizations may distribute their software systems over a pool of resources by utilizing cloud services. Organizations, on the other hand, rely largely on their mission-critical systems, which have been created through time. The on-premise deployments are common for older applications. Cloud migration has been the subject of research in recent years. The goal of the work in [5] is to discover, taxonomically classify, and compare current cloud migration studies. The analysis demonstrates that cloud migration research is still in its infancy, but it is progressing. This analysis reveals that there is a dearth of tool support for automating migration processes.

In the data warehousing process, the ETL phase is crucial. This is by far the most time-consuming and costly step of a decision-making system installation project. This ETL phase's adaptation to the big data/cloud environment, which is defined by massive amounts of heterogeneous and distant data, has therefore become a must. The authors in [6] propose a newer model in terms of volume and velocity, a state-of-the-art on the performance of ETL procedures in the cloud. The recommended solutions do not consider the flexibility or even the dynamic nature of data in the cloud.

The performance of a real-world TPS application is examined in [7], both on-premises and after migration to a cloud environment with both a public and private base. The performance characteristics assessed on Amazon EC2 are compared to their counterparts on premise and in private clouds. Measurements of selected performance indicators such as total throughput, total hits, average transaction response time, average throughput, and average hits per second are proposed in [8]. The findings of the performance metrics reveal that a public cloud, with its advanced services, outperforms a private cloud and on-premise.

With the growing popularity of cloud computing, the need to transfer legacy applications to the cloud in a timely and effective manner has become critical. Existing methods have several drawbacks, such as the requirement of legacy application source code or the inability to tweak or mash up the moved application. In [9], the authors propose a brand-new Application Migration Solution (AMS) that uses GUI recognition and reconstruction technologies to quickly transfer old programs to web applications. The fundamental technologies are described and evaluation results are supplied for technological validation. Enterprises may quickly deploy their old apps to the cloud using AMS. The authors conclude by suggesting that advanced corporate requirements, such as application customization or mashup with other applications to create more integrated and powerful applications, are also met by this solution.

Migration from an application to a new product, or from one product to another, is seen as a high-risk endeavor. Due to intellectual property rights, each product will not reveal its underlying database. The article [10], tries to explain how to migrate historical data from a risk management product to a data warehouse product and set up a daily feed from the product into the data warehouse so that prior day reports are available for current day trading.

The study in [11] entails the details of data integration and the Extract-Transform-Load (ETL) process. The authors have explored talend as an ETL tool and to lay out the differences between the processes involving data transfers between databases and data-warehouses. Data engineering is defined as a process which involves data extraction into a staging table, integration using ETL tools, and ingestion into a Data Warehouse, where business intelligence analysts utilize the information to make choices.

## III. PROPOSED METHODOLOGY

The implementation in this study uses Talend Open Studio as the development platform. Because the study is primarily concerned with the ETL process and data integration, Talend is the most appropriate product and has been chosen as the product to be utilised as part of the organization's transition to cloud architecture. The interfaces to be developed are present in DataStage and the same has to be re-created in Talend. By utilizing pre-built components in Talend meant for data integration purposes, the data pipeline is created.

The initial step is to develop the components that access the input data from either the source data or read input files. This requires access to the organization’s proprietary servers. The input data is extracted from these sources into the data staging area. Because the extracted data is in numerous formats and may be destroyed, it is necessary to extract it from several source systems and store it in the staging area. A collection of rules or functions is applied to the obtained data in order to convert it into a single standard format. Filtering, cleaning, joining, separating, sorting, and/or aggregating are all examples of transformation. Finally, the transformed data is put into the destination database. Few jobs require a report to be generated as a summary of the data is essential to provide details to serve business demands.

## IV. SYSTEM ARCHITECTURE

The entire system is split up into two modules, the master job and the sub job.

1. Master jobs: They handle the loading of context variables to receive the database and job specific configurations. They establish the connection with the databases and pass the job specific context variables to the sub jobs appropriately. Master job handles the sequencing of the sub jobs, i.e. the order in which the sub jobs need to be executed is determined and controlled by the master job.
2. Sub jobs: They perform several functions such as extracting input data from source database, read input files, transform the data, load data into destination database and also generate reports with the resultant data. This system architecture is a generic template for all the interfaces developed in Talend, the system architecture diagram is shown in Fig 1.

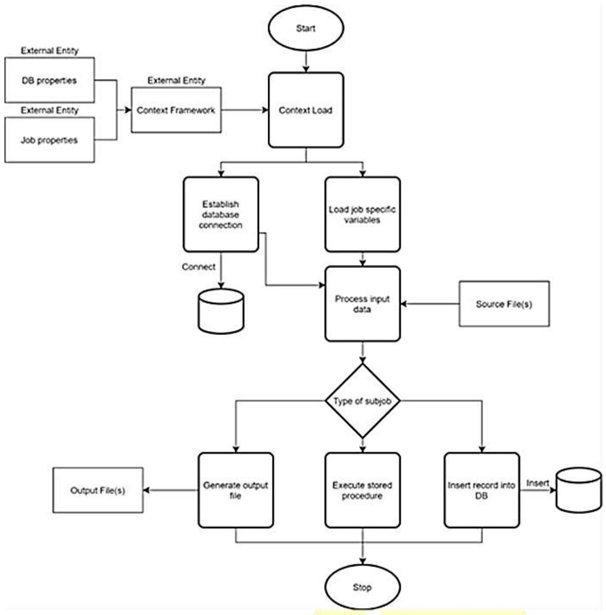

***Fig 1:** System architecture diagram of a typical Talend interface*

### *A. Context variables*

The initial step while running an interface is to load the context variables corresponding to that interface. These variables can be classified as 2 types, database variables which contain the database configuration for connection, and job specific variables which contain input and reject file path and file names, table and schema names required for the sub job, etc. These variables are used to get the code ready for production. It implies that by utilizing context variables, you may move code across development, QA, and production environments and have it ready to be executed in any of them. These variables are passed down to the sub jobs, and sub jobs are called in the required sequence as set in the master job.

### *B. ETL process*

ETL stands for Extract, Transform, and Load, and it is a Data Warehousing procedure. An ETL tool collects data from multiple data source systems, transforms it in the staging area, and then loads it into the Data Warehouse system.

1. *Extract:* Data is retrieved from numerous source systems and stored in the staging area in various forms such as relational databases, No SQL, XML, and flat files. Because the extracted data is in multiple forms and might be damaged, it is critical to extract it from several source systems and store it first in the staging area rather than straight in the data warehouse.
2. *Transform:* To transform the retrieved data into a single standard format, a set of rules or functions are applied to it. Transformation can be of any form, it can be filtering, cleaning, joining, splitting, sorting and/or aggregating.
3. Load: Finally, the converted data is put into the data warehouse. The pace and duration of loading are totally determined by the needs and differ from system to system.

## V. RESULTS AND ANALYSIS

Comparative analysis of both the tools includes the comparison based on the features present in both the tools and the results obtained by executing the same interfaces in both. The feature based comparison of Talend and DataStage is show in Table-1.

***Table 1:** Feature based comparison of Talend and DataStage*

| Talend | DataStage |
|---|---|
| Talend offers robust data integration in an open and scalable architecture to maximize its value to your business. | IBM Infosphere DataStage integrates data across multiple systems using a high-performance parallel framework, and it supports extended metadata management and enterprise connectivity |
| Easier tool for development and it is faster to implement jobs. | The GUI is difficult to learn and get a hang of for a beginner and needs more work to be user-friendly. |
| Talend is well documented for all the basic components with rich examples, and the in-application help is very powerful. Though it gets complicated with complicated components. | The documentation and in-application help for this tool is not sufficient and needs more detail especially for the new features. |
| The stability of Talend is good but the performance when dealing with huge amounts of data needs improvement. | DataStage can perform better when huge quantity of data is involved and is a very stable tool. |

| | |
|---|---|
| Talend outperforms DataStage when the interfaces that deal with less sizeable data is considered, in terms of time and efficiency. | DataStage is developed to handle big data and needs improvement when the data input is less. |
| Talend generates its executable through Java code generation engine. | DataStage executable generation involves the generation of one or more SQL script files and an XML metadata file. |
| Talend has extensive support to cloud services such as SnowFlake and is very easy to integrate them. | DataStage supports operations and integration with cloud services but it is not as good as Talend. |

For evaluating the interfaces developed in Talend, the reference chosen is the interfaces developed in DataStage. The time to complete execution of an interface is an essential metric used in the comparative analysis of the two platforms. This metric is measured by running the same interface in both Talend and DataStage. The complexity of each interface is also taken into consideration when evaluating the two tools. The metrics chosen to evaluate the interfaces are the complexity of the query, the number of sub jobs involved and the complexity of transformations performed on the data as shown in Chart 1.

The results of the evaluation of interfaces in both tools, Talend and DataStage are recorded and tabulated in Table-2. It can be inferred that Talend is equivalent to DataStage in terms of handling interfaces and time of execution as depicted in Chart 2. In some interfaces, Talend performs much better than DataStage. A few interfaces have been enhanced by careful examination to reduce the complexities involved. This has been proven in the case of Interface 1 and Interface 5 where Talend outperforms DataStage due to the enhancements implemented in Talend.

*Table 2: Interface wise comparison of Talend and DataStage*

| Interface No | Query complexities | Transformation complexity | No of sub jobs | Talend | DataStage |
|---|---|---|---|---|---|
| **1** | Low | Low | 24 | 38s | 70s |
| **2** | Low | Low | 1 | 9s | 16s |
| **3** | Low | Low | 1 | 13s | 17s |
| **4** | Low | Medium | 1 | 16s | 10s |
| **5** | Medium | High | 3 | 3s | 7s |
| **6** | High | High | 4 | 49s | 40s |

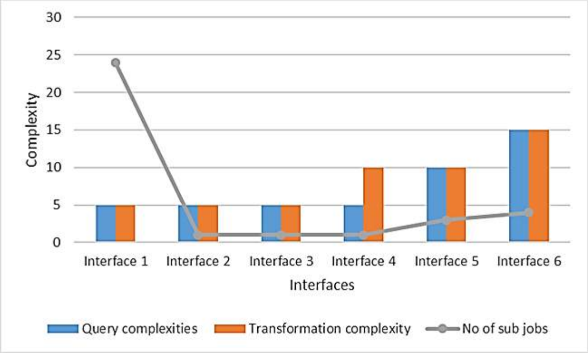

*Chart-1: Interface complexity comparison*

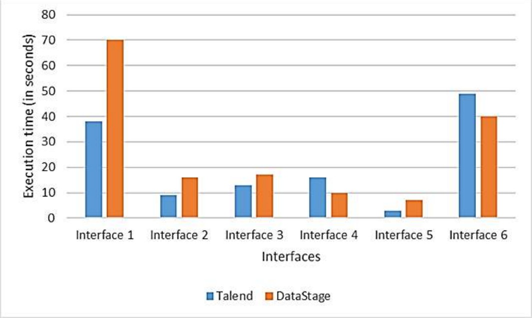

*Chart-2: Performance comparison*

## VI. CONCLUSION

With the experimentation results, and the comparison of the pros and cons of both the tools with respect to all the features, it can be concluded that Talend is an optimal choice for developing data integration projects. Although DataStage has certain benefits, the overall advantages of Talend outweighs its setbacks. The interfaces present in DataStage have been re-created in Talend and with a few tweaks and enhancement the execution time of a few interfaces was halved, giving it a bigger edge over DataStage. Thus, a seamless data pipeline was created successfully, to perform data integration using the ETL (Extract-Transform-Load) process, that extracts the data from a data lake which can be from multiple sources and in multiple formats, transformed and formatted to specified requirements and loaded into the staging tables of the data warehouse.

This study is mainly concerned with cloud migration and the benefits of platform migration. Hence, it can be concluded that the choice of migrating all the present interfaces from DataStage to Talend is justifiable when an organization decides to shift to cloud architecture.

## REFERENCES

[1] Duncan Stewart, Nobuo Okubo, "The cloud migration forecast: Cloudy with a chance of clouds. TMT Predictions 2021", 07 Dec, 2020, Accessed on 11 May, 2021, [Online] Available: https://www2.deloitte.com/xe/en/insights/industry/technology/technology-media-and-telecom-predictions/2021/cloud-migration-trends-and-forecast.html.

[2] Anurag, "Everything you need to know about Platform Migration", 7 May, 2020. Accessed on 11 May, 2021, [Online] Available: https://www.newgenapps.com/blog/everything-you-need-to-know-about-platform-migration/.

[3] D. S. Linthicum, "Cloud-Native Applications and Cloud Migration: The Good, the Bad, and the Points Between," in IEEE Cloud Computing, vol. 4, no. 5, pp. 12-14, September/October 2017, doi: 10.1109/MCC.2017.4250932.

[4] Yifat Perry. "Cloud MigrationWhat Is a Lift and Shift Cloud Migration?". 7 Mar, 2020. Accessed on 11 May, 2021 [Online] Available: https://cloud.netapp.com/blog/what-is-a-lift-and-shift-cloud-migration.

[5] P. Jamshidi, A. Ahmad and C. Pahl, "Cloud Migration Research: A Systematic Review," in IEEE Transactions on Cloud Computing, vol. 1, no. 2, pp. 142-157, July-December 2013, doi: 10.1109/TCC.2013.10.

[6] P. S. Diouf, A. Boly and S. Ndiaye, "Variety of data in the ETL processes in the cloud: State of the art," 2018 IEEE International Conference on Innovative Research and Development (ICIRD), 2018, pp. 1-5, doi: 10.1109/ICIRD.2018.8376308.

[7] A. Kandil and H. El-Deeb, "Exploration of application migration to cloud environment," 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence), 2016, pp. 109-114, doi: 10.1109/CONFLUENCE.2016.7508097.

[8] B. Rongen, "Making the case for migration of information systems to the cloud", 16thTwente Student Conference on IT, Enschede, The Netherlands, 2012.

[9] X. Meng, J. Shi, X. Liu, H. Liu and L. Wang, "Legacy Application Migration to Cloud," 2011 IEEE 4th International Conference on Cloud Computing, 2011, pp. 750-751, doi: 10.1109/CLOUD.2011.56.

[10] Shrinivasan, C, "Data Migration from a Product to a Data Warehouse Using ETL Tool", Proceedings of the Euromicro Conference on Software Maintenance and Reengineering, CSMR, 2010, pp 63 – 65, doi: 10.1109/CSMR.2010.25.

[11] J. Sreemathy, I. Joseph V., S. Nisha, C. Prabha I. and G. Priya R.M., "Data Integration in ETL Using TALEND," 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2020, pp. 1444-1448, doi: 10.1109/ICACCS48705.2020.9074186.

[12] M. A. Chauhan and M. A. Babar, "Migrating Service-Oriented System to Cloud Computing: An Experience Report," 2011 IEEE 4th International Conference on Cloud Computing, 2011, pp. 404-411, doi: 10.1109/CLOUD.2011.46.

[13] Ali, Syed Muhammad Fawad, "Next-generation ETL Framework to Address the Challenges Posed by Big Data.", DOLAP, 2018

[14] L. Munoz, J. Mazon and J. Trujillo, "ETL Process Modeling Conceptual for Data Warehouses: A Systematic Mapping Study," in IEEE Latin America Transactions, vol. 9, no. 3, pp. 358-363, June 2011, doi: 10.1109/TLA.2011.5893784

[15] A. Bansel, H. González-Vélez and A. E. Chis, "Cloud-Based NoSQL Data Migration," 2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP), 2016, pp. 224-231, doi: 10.1109/PDP.2016.111.