Technology Migration Challenges in a Big Data Architecture Stack

Rekha Singhal
TCS Research
Mumbai, India.
rekha.singhal@tcs.com

Shruti Kunde
TCS Research
Mumbai, India.
shruti.kunde@tcs.com

ABSTRACT

Application and/or data migration is a result of limitations in existing system architecture to handle new requirements and the availability of newer, more efficient technology. In any big data architecture, technology migration is staggered across multiple levels and poses functional (related to components of the architecture and underlying infrastructure) and non-functional (QoS) challenges such as availability, reliability and performance guarantees in the target architecture. In this paper, (1) we outline a big data architecture stack and identify research problems arising out of the technology migration in this scenario (2) we propose a smart rule engine system which facilitates the decision making process for the technology to be used at different layers in the architecture during migration.

Keywords

Big data, Migration, Performance

1. INTRODUCTION

Application or data migration [3] generally results owing to the introduction of more efficient, cost effective technology. Migration impacts the overall system architecture from various perspectives. (1) **The business impact** defines the business processes and re-architects organizational needs (2) **The cost impact**, outlines the total cost of ownership (3) **The technology impact**, defines impact on the application performance, software stack, hardware and data storage. Technology migration is triggered due to reasons such as change in business demands or functions, availability of new types of data, scalability, platform consolidation, limitation of existing technology, increase in workload and increase in total cost of ownership (TCO).

The ultimate aim of technology migration is to improve the overall system performance and TCO to ensure scalability and reliability over time, while meeting new types of data processing needs. The foremost challenge is to detect which application component(s) need to be migrated and then to decide the architecture, comprising big data processing platforms, for migration while ensuring performance and security during physical technology migration to newer platforms. The big data architecture we envision (Figure 1) is composed of multiple layers such as data acquisition layer, data messaging layer (Kafka, RabbitMQ, ZeroMQ [2]), stream data processing layer (Storm, Redis), data storage layer (relational, columnar), parallel data processing engine (Ignite, Memsql, Hbase, Hadoop), SQL query processing engine (Phoenix, Shark, Hive) and data visualization layer (elastic search, data visualization tools [1]). Each layer has various implementations available commercially as well as in open source [4]. The challenge lies in choosing the *most appropriate* component at each layer to guarantee *performance* to end users with growing data and workload size across all application components. Here, *performance* becomes a multi-objective function of latency, throughput, energy, reliability and total cost of ownership.

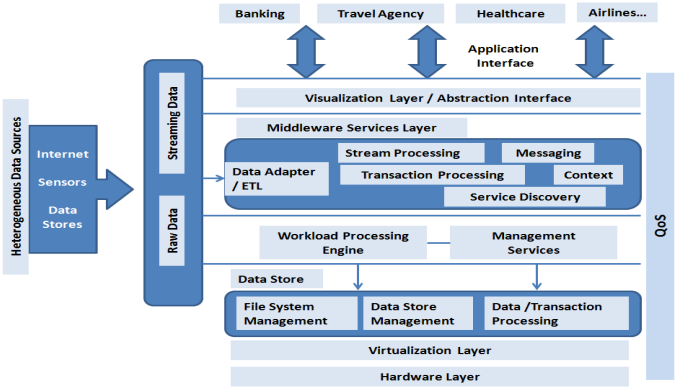

Figure 1: Big Data Architecture

2. MIGRATION PROCESS AND CHALLENGES

A migration process (Figure 2) happens only when a business experiences degradation in performance or increase in TCO. Some significant questions arising are - Should all or only critical components of a system migrate? What does the target architecture for components to be migrated look like? This decision making phase for (1) finalizing the technology in the big data architecture (2) satisfying the required performance and TCO objectives for migration, is referred to as the *Analysis Phase*. The actual (or physical) migration involves preparing the *migration plan* (How to migrate

ICPE'17 April 22-26, 2017, L'Aquila, Italy

ACM ISBN 978-1-4503-4404-3/17/04.

DOI: http://dx.doi.org/10.1145/3030207.3053670

Table 1: Rule Grammar

Constructs	Objects	Operations	Operators
IF, THEN, AND, OR	Event/Data properties, Performance metrics	Assert, Set, Average, Min, Max	$\geq, \leq, =, <, >, \neq, \cap, \cup$

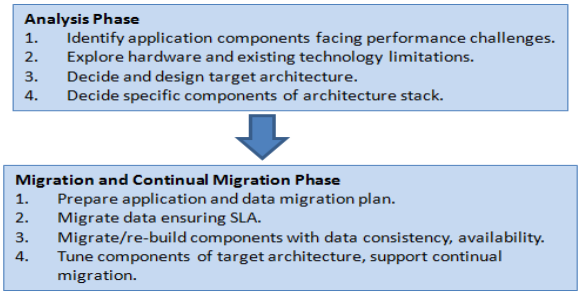

Figure 2: Process of Migration

application/ data while guaranteeing performance and security?) is the *Migration and Continual Migration Phase*. It deals with challenges such as maintaining performance, reliability and correctness of application and data during and after migration.

Some of the research issues we tackle are :

1. Define multi-objective function based on desirable cost, performance, reliability, growth of data and users in the system.

2. Identifying workload characteristics based on the application domain such as the load (arrival rate/size of requests), format of input data (streaming, continuous queries, raw data), data volume, data delivery semantics (exactly once, at most once, at least once) and message ordering.

3. Identifying relevant benchmarks or building new ones for evaluating performance and availability of applications deployed on the stack (Figure 1). Benchmarks should be able to handle a mix of different types of workloads such on-line transaction, business intelligence (adhoc and interactive queries), reporting, exploratory, continuous queries and queries on streaming data.

4. Multi objective performance and capacity planning models to guarantee QoS on the target architecture in terms of workload throughput, average latency for large scale systems (data, cluster, workload), availability and reliability using measurements conducted during POC (Proof of Concept) on small scale systems.

3. SMART RULE BASE

We propose a smart rule engine system (Figure 3), which will store rules formulated using the rule grammar outlined in Table 1. The rules are created by experts based on a literature survey (survey snapshot in Table 2), practitioner experiences and by benchmarking big data architecture stack. The benchmarking results enable the decision making process for matching performance guarantees required in the target architecture, while the state of the art study, facilitates feature set matching. The rule engine incorporates the intelligence to determine compatibility of the technologies that will be recommended at different layers in the big data architecture stack and to make a holistic recommendation based on the overall target architecture during the migration process. The rule engine is designed to be flexible and will evolve over a period of time to incorporate new experiences and thus provide improved recommendations.

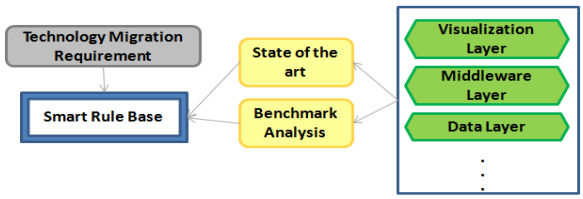

Figure 3: Smart Rule Engine System

Table 2: State of the art (A snapshot)

Query Engine	Cluster Size	Data Size	WorkLoad Type	Scalability
Shark	5+1 nodes, 68GB, 8 cores	5GB-525GB	Scan 10 sec, Agg 10min	Cluster scale till 4
Impala	5+1 nodes	5GB-525GB	Scan 40 sec 12hr-30min, Agg 5-10min	Performance bad on 6 nodes

4. CONCLUSIONS

In this paper we have presented a framework of a big data architecture stack, while highlighting the challenges involved in technology migration at each layer. We also propose a solution in the form of a smart rule engine system, which facilitates the decision making process of selecting the appropriate technology at various layers in the target architecture. Advances in newer technologies, performance guarantees and heterogeneous nature of incoming data, have all given rise to several non-trivial research challenges in the area of performance, which have been addressed in this paper. We conclude with an underlying need for having an auto migration process in which the system can continually learn, decide and migrate to a target big data architecture.

5. REFERENCES

[1] Open Souce Visualization Tools. `https://opensource.com/life/15/6/eight-open-source-data-visualization-tools/`.

[2] Performance Evaluation of Messaging Brokers. `http://www.eharmony.com/engineering/in-pursuit-of-messaging-brokers/`.

[3] Z. Bian, K. Wang, Z. Wang, G. Munce, I. Cremer, W. Zhou, Q. Chen, and G. Xu. Simulating big data clusters for system planning, evaluation and optimization. In *Proceedings of 43rd International Conference on Parallel Processing*, 2014.

[4] S. Mazumder. *Big Data Tools and Platforms*. Springer, 2016.