Abyss Storage Cluster 기반의 DataLake Framework의 설계

(Draft Design of DataLake Framework based on Abyss Storage Cluster)

차병래*, 박선*, 신병춘**, 김종원*

(ByungRae Cha, Sun Park, Byeong-Chun Shin and JongWon Kim)

요 약

기관 또는 조직은 비즈니스 시스템의 규모가 커지면서 이들과 관련된 서로 다른 시스템에서 다양한 대량의 데이터들이 생성되고 있다. 이와 같이 비즈니스 환경에서 서로 다른 시스템에서 데이터를 보다 스마트하게 처리하여 효율성을 높일 수 있는 방법이 필요하다. 이를 위한 가장 기본적인 접근 방법 중 하나는 DataLake와 같이 데이터를 정확하게 설명하고 전체 비즈니스에 대한 가장 중요한 데이터를 나타낼 수 있는 단일 도메인 모델을 만드는 것이다. DataLake의 장점을 구현하기 위해서는 다양하게 요구되어진 기능들을 어떤 구조로, 어떻게 작동 할 것인지에 대한 DataLake의 구성 요소들을 정의하는 게 중요하며, DataLake의 구성 요소들에 의해서 데이터 흐름에 따른 라이프 사이클을 갖게 된다. 또한 데이터 획득 시점에서 DataLake로 유입되는 동안 메타 데이터는 데이터 추적 가능성, 데이터 계보 및 라이프 사이클 전반의 데이터 민감도에 기반 한 보안 측면과 함께 캡처 및 관리되어야 하며, 이러한 이유로 Abyss Storage Cluster 기반의 DataLake Framework를 설계하였다.

■ 중심어 : 데이터레이크 프레임워크; Abyss Storage Cluster;

Abstract

As an organization or organization grows in size, many different types of data are being generated in different systems. There is a need for a way to improve efficiency by processing data smarter in different systems. Just like DataLake, we are creating a single domain model that accurately describes the data and can represent the most important data for the entire business. In order to realize the benefits of a DataLake, it is import to know how a DataLake may be expected to work and what components architecturally may help to build a fully functional DataLake. DataLake components have a life cycle according to the data flow. And while th data flows into a DataLake from the point of acquisition, its meta-data is captured and managed along with data traceability, data lineage, and security aspects based on data sensitivity across its life cycle. According to this reason, we have designed the DataLake Framework based on Abyss Storage Cluster.

■ keywords : DataLake Framework; Abyss Storage Cluster;

Ⅰ. 서 론

데이터는 기업운영의 여러 측면에서 중추적인 역할을 담당하면서 많은 기업들에게 중요성이 높아지고 있으며, 점차 기업의 가치는 데이터 중심으로 변화하고 있다. 기업에서는 초창기에 저장소(repository)에 자료를 저장하여 사용하는 단순한 방법을 사용하였다. 좀 더 발전된 형태로, 사용자의 의사결정에 도움을 주기 위하여, 기관 시스템의 데이터베이스에 축적된 데이터를 공통의 형식으로 변환해서 관리하는 데이터베이스인 데이터-웨어하우스(Data Warehouse)을 이용하고 있다. 오늘날 우리는 빅데이터(Big Data)를 활용하는 유스케이스(use case)들이 증가하고 있으며, 기업에서 처리되는 데이터들도 다양한 형태로 대량화되어짐에 따라서 이를 효율적으로 처리할 수 있는 데이터 레이크(Data Lake)가 점차 주목을 받고 있는 상황이다. z

기관 또는 조직은 비즈니스 시스템 전반에 걸쳐 방대한 양

* 정회원, 광주과학기술원 전지전자컴퓨터공학부
** 정회원, 전남대학교 수학과
※ This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (2017R1E1A1A03070059).
※ 본 연구는 중소벤처기업부와 한국산업기술진흥원의 “지역특화산업육성사업(과제번호 R0006020)”으로 수행된 연구결과 입니다.

접수일자 : 2018년 01월 08일 게재확정일 : 2018년 02월 20일
수정일자 : 2018년 01월 24일
교신저자 : 김종원 e-mail : jongwon@gist.ac.kr

Smart Media Journal / Vol.7, No.1 / ISSN:2287-1322
http://dx.doi.org/10.30693/SMJ.2018.7.1.10

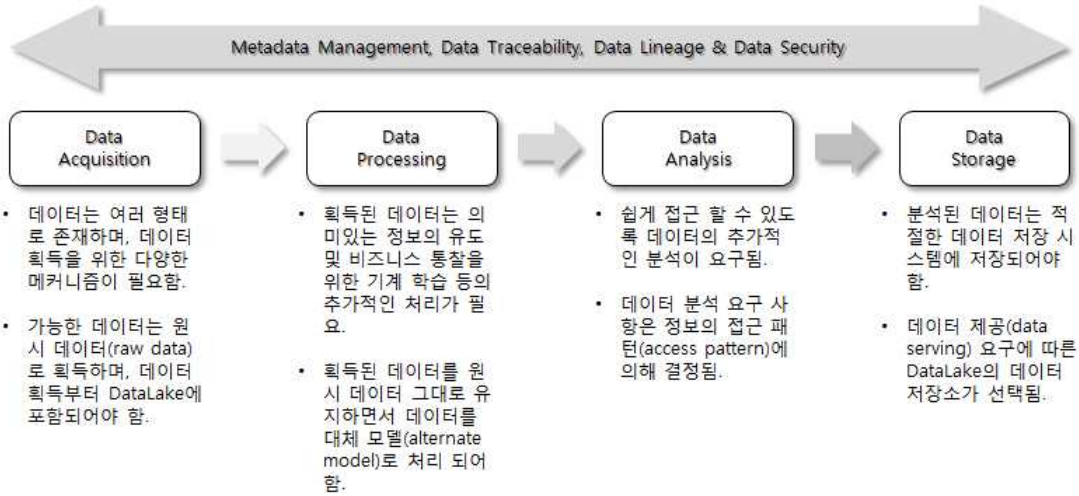

그림 1. DataLake의 Life Cycle

의 데이터를 생성하며, 규모가 커지면 서로 다른 시스템에서 데이터를 보다 스마트하게 처리하기를 원한다. 이를 위한 가장 기본적인 접근 방법 중 하나는 정보를 엔터프라이즈 데이터(Enterprise Data)화 하여서 데이터를 정확하게 설명하고 전체 비즈니스에 대한 가장 중요한 데이터를 나타낼 수 있는 단일 도메인 모델(single domain model)을 만드는 것이다. 이러한 일반적인 Data Lake 모델들은 데이터를 캡처링(capturing), 처리(processing), 분석(analyzing), 그리고 사용자들(users) 또는 데이터를 소비하는 시스템들(consuming systems)에 제공할 수 있어야 한다.

이를 위해서는 전사적 데이터 레이크(Enterprise-wide DataLake)를 구축해야 하며, 다음의 항목들을 고려해야 한다 [1]:

- 데이터 거버넌스(Data Governance) 및 데이터 계보(Data Lineage)
- 비즈니스 인텔리전스(Business Intelligence)를 유도하기 위한 인공 지능 및 기계 학습의 적용
- 예측 분석(Predictive Analysis)
- 정보의 추적성 및 일관성(Information Traceability and Consistency)
- 차원 데이터를 도출하기위한 과거 분석(Historical Analysis to derive Dimensional Data)
- 다양한 엔터프라이즈 데이터 전달을 위한 중앙 집중식 데이터 소스 기반의 최적화 된 데이터 서비스를 제공
- 빅데이터의 배치(Batch) 및 스트리밍(Streaming) 처리를 위한 람다 아키텍처(Lambda Architecture)의 채용

DataLake의 장점을 구현하기 위해서는 위에 기술된 DataLake의 요구되어진 기능들을 어떤 구조로, 어떻게 작동할 것인지에 대한 DataLake의 구성 요소들을 정의하는 게 중요하다. DataLake의 구성 요소들에 의해서 데이터 흐름(Data Flow)에 따른 라이프 사이클(Life Cycle)을 [그림 1]과 같이 Data Acquisition, Data Processing, Data Analysis, Data Storage을 갖게 된다. 또한 데이터 획득(Data Acquisition) 시점에서 DataLake로 유입되는 동안 메타 데이터(Meta data)는 데이터 추적 가능성, 데이터 계보(data lineage) 및 라이프 사이클 전반의 데이터 민감도(data sensitivity)에 기반 한 보안 측면과 함께 캡처 및 관리된다.

본 논문에서는 ICT 분야의 메가 트렌드인 Big Data, IoT, Cloud, AI 등의 시대적 상황에 따른 전사적 데이터 레이크(Enterprise-wide DataLake)를 구축하기 위한 여러 요구조건들의 사전 조사와 새롭게 정의되고 있는 다양한 Data Lake 솔루션 등을 고려하여 Abyss Storage Cluster 기반의 DataLake Framework를 설계 및 제안하고자 한다.

Ⅱ. 관련 연구

빅데이터 정의와 빅데이터를 처리하기 위한 다양한 아키텍처와 프레임워크들이 새롭게 계속적으로 정의되고 있다. 본 관련 연구에서는 빅데이터 정의와 빅데이터를 처리하기 위한 다양한 아키텍처/프레임워크, 그리고 Abyss Storage Cluster에 대하여 간략하게 소개한다.

1. 빅데이터와 실시간 처리 시스템

빅데이터 정의와 빅데이터를 처리하기 위한 다양한 아키텍처와 프레임워크들이 계속적으로 새롭게 정의되고 있으며, 특히 IBM은 [그림 2]와 같이 빅데이터를 4V(Volume, Variety, Velocity, Veracity)로 정의하고 있으며, 최근에는 7V로 4V에

추가적으로 3V(Vision, Visualization, Value)를 더하여 새롭게 재정의 하고 있다[1].

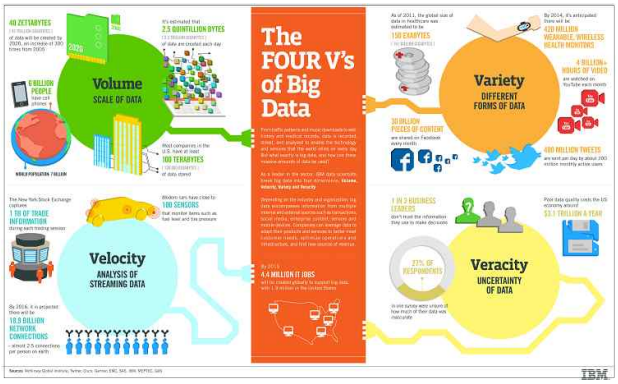

그림 2. IBM의 빅데이터의 정의[2]

실시간 처리를 위한 빅데이터 아키텍처는 빅데이터의 근원지(source)로부터 데이터를 수집하는 단계, 이를 적재 및 보관하고 보안 처리하는 단계, 데이터를 조회하는 단계, 데이터를 분석하는 단계, 그리고 데이터를 시각화하는 단계로 나눌 수 있으며, [그림 3]에 간략하게 나타낸다[3].

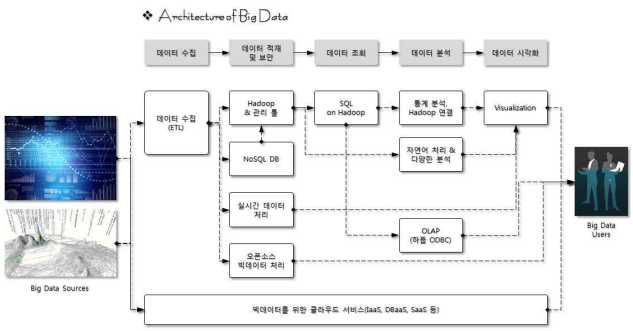

그림 3. 빅데이터 아키텍처

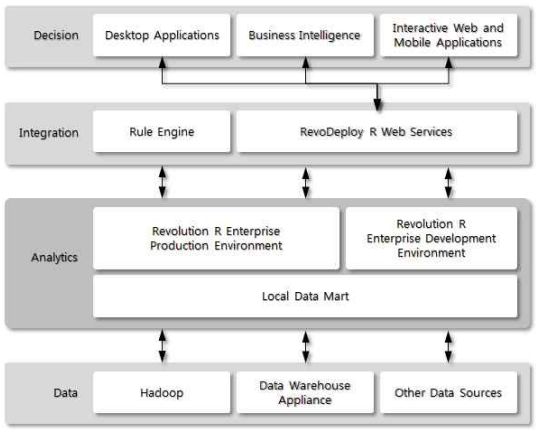

그림 4. 실시간 빅데이터의 예측분석 스택

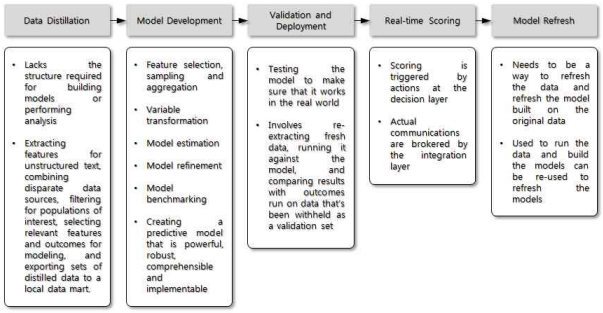

그림 5. 실시간 데이터 처리를 위한 5 단계 절차

실시간의 빅데이터 분석을 위한 David Smith가 제안한 RTBDA (Real-Time Big Data Analytics) 아키텍처는 4 계층(Data layer, Analytics layer, Integration layer, Decision layer)으로 [그림 4]와 같이 나타내며, 실시간 데이터 분석을 위한 RTBDA의 예측 분석을 위한 프레임워크는 5 단계 절차(Data distillation, Model development, Validation and deployment, Real-time scoring, and Model refresh)는 [그림 5]에 나타낸다[4].

2. Pradeep & Beulah의 Data Lake 개념 구조

Pradeep & Beulah은 Data Lake의 개념적인 구조를 [그림 6]과 같이 나타냈으며, Security and Governance Layer, Metadata Layer, 그리고 Information Life-cycle Management Layer의 3개 계층과 Intake Tier, Management Tier, 그리고 Consumption Tier의 3개 티어로 구성하였다[5]. Intake Tier는 Source System Zone, Transient Zone, 그리고 Raw Zone으로 구성되며, Management Tier는 Integration Zone, Enrichment Zone, 그리고 Data Hube Zone으로 구성된다. 마지막으로, Consumption Tier는 Data Discovery, Data Provisioning, External Access, 그리고 Internal Processing으로 구성되었다.

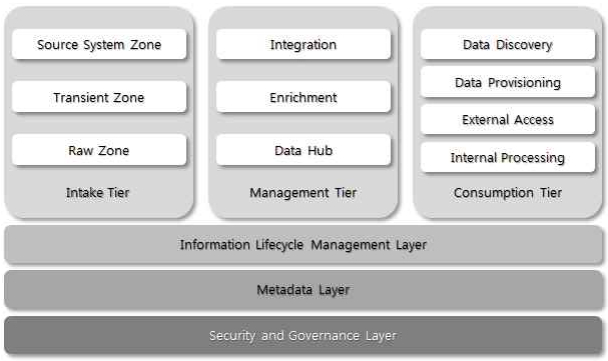

그림 6. Pradeep & Beulah의 Data Lake 개념 구조

Smart Media Journal / Vol.7, No.1 / ISSN:2287-1322
http://dx.doi.org/10.30693/SMJ.2018.7.1.12

3. AWS 기반의 Data Lake 솔루션 아키텍처

기업 또는 조직에서는 기존 솔루션으로 더 이상의 속도를 낼 수 없는 상황이며, 특히 Data Silo의 특수성에 의하여 보다 포괄적이고 효율적인 분석을 위한 스토리지 통합을 어렵게 만든다. 이러한 상황의 타개책 중의 하나로 아마존 AWS(Amazon Web Service)는 Big Data 처리 가능한 Data Lake를 제안 및 BI를 위한 다양한 분석 모듈 및 라이브러리를 서비스로 제공하고 있다[6]. AWS의 Data Lake 솔루션은 Amazon S3(Simple Storage Service)를 기본 스토리지 플랫폼으로 사용하며, Amazon S3는 사실상 무제한적인 확장성으로 인해 데이터레이크에 대한 최적의 기반과 내구성을 제공하게 설계되었다. 확장 가능한 성능, 사용 편의 기능 및 기본 암호화 및 액세스 제어 기능을 갖추고 있으며, Amazon S3는 AWS 및 타사 ISV(Independent Software Vendors) 데이터 처리 도구의 광범위한 포트폴리오와 통합이 가능하며, [그림 7]은 AWS의 Data Lake 솔루션의 아키텍처를 나타낸 것이다[7]. AWS 기반의 Data Lake 솔루션의 장점으로는 Flexibility, Agility, Security and Compliance, 그리고 Broad and Deep Capability 등이며, 또한 다음 사항들을 지원한다[8].

- 모든 유형의 데이터를 어떤 규모라도 저렴한 비용으로 수집 및 저장 가능
- 데이터 보안 및 무단 액세스 방지
- 중앙 저장소에서 관련 데이터를 카탈로그화, 검색 및 발견
- 신속하고 간편하게 새로운 유형의 데이터 분석 수행
- 애드혹 (ad hoc) 분석, 실시간 스트리밍, 예측 분석, 인공지능 (AI) 및 기계 학습을 위한 광범위한 분석 엔진 사용

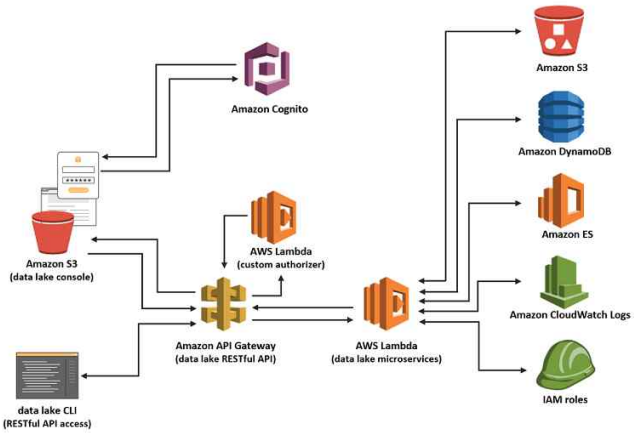

그림 7. AWS의 Data Lake 솔루션 아키텍처의 개요도

4. Abyss Storage Cluster

SMB를 위한 대용량 Abyss Storage Cluster의 구성은 [그림 8]과 같이 나타낼 수 있으며, 실제적으로 Abyss Storage Cluster의 H/W 프로토타입 개발과 제품의 양산이 가능하다. 또한 Abyss Storage의 성능 향상을 위하여 스토리지의 디스크 매체별 성능 테스트와 스토리지의 내부 네트워크의 가속화를 위한 본딩(Bonding), 그리고 KOREN 네트워크를 이용한 국내외 네트워크 트래픽 테스트를 완료한 상태이다[9, 10].

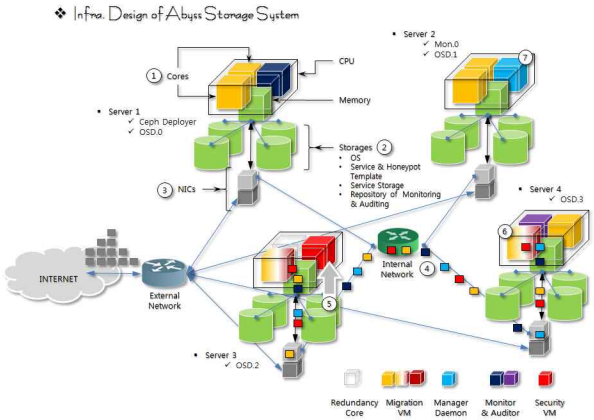

그림 8. Abyss Storage Cluster의 개념도

III. Abyss Storage Cluster 기반의 DataLake Framework의 설계

Abyss Storage Cluster 기반의 DataLake Framework는 전사적 데이터 레이크(Enterprise-wide DataLake)를 구축하기 위한 요구조건들을 고려하여 다음의 항목들과 [그림 9]와 같이 구성 된다:

- Physical Layer
- Distributed Storage Layer
- Security Layer
- Data Acquisition Layer
- Messaging Layer
- Ingestion Layer
- Lambda Architecture
- Serving Layer

DataLake의 Physical Layer는 DataLake Framework의 최하단에 위치하며 DataLake의 논리적 기능들을 지원하기 위한 컴퓨팅, 스토리지, 그리고 네트워킹 등의 물리적인 자원들로 구성되며, 본 연구에서는 Abyss Storage Cluster로 대응한다.

DataLake의 Distributed Storage Layer는 들어오는 이벤트 및 데이터 스트림에 대한 람다 아키텍처[11]의 전반적인 반응성(reactivity)을 정의하게 되며, 연결된 시스템의 이론(theory

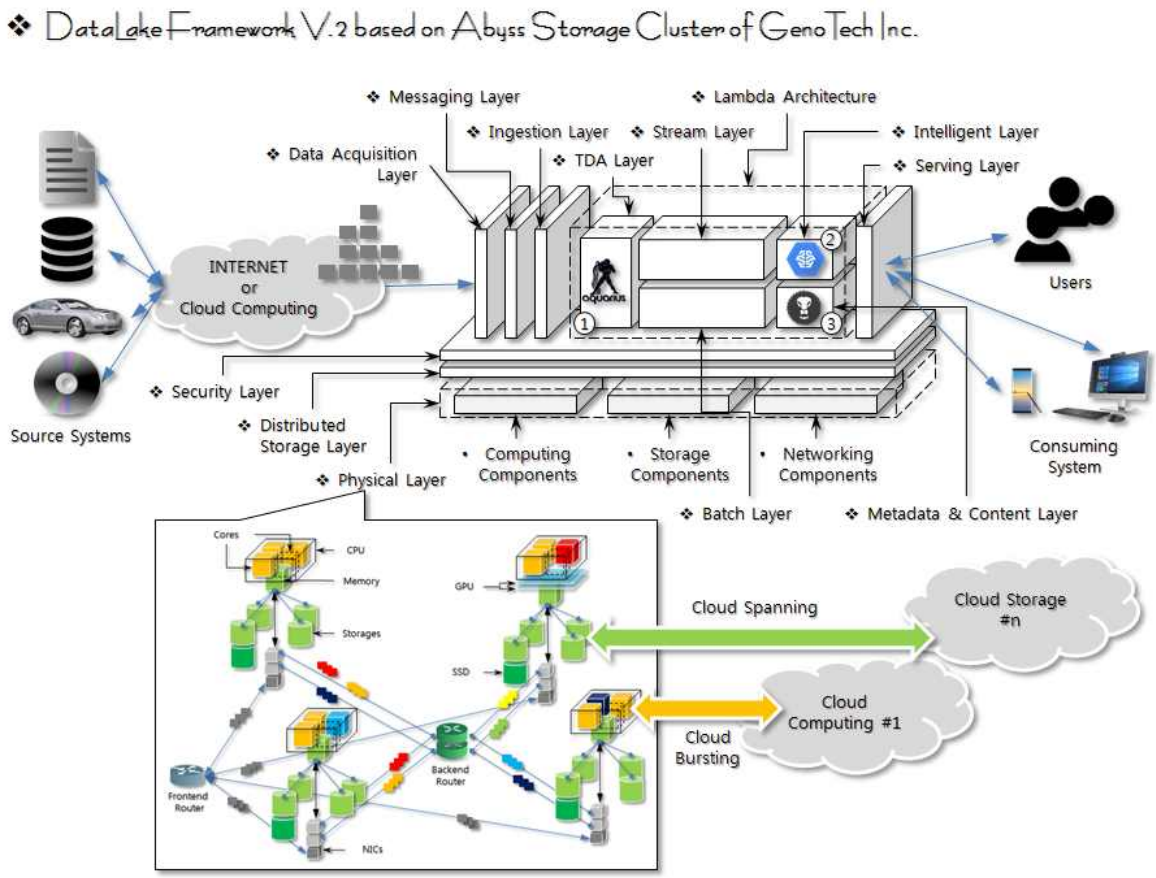

그림 9. DataLake Framework와 Abyss Storage Cluster 기반의 Cloud Bursting/Spanning

of connected systems)에 따르면 시스템은 연결 체인에서 가장 느린 시스템에 맞춰 반응성이 갖게 되므로, Distributed Storage Layer가 충분히 빠르지 않은 경우, 람다 아키텍처의 Stream Layer에 의해 수행되는 동작은 느려서 람다 아키텍처의 준 실시간(near rela-time) 특성을 방해하게 될 수 있다.

DataLake의 Security Layer는 각 계층의 구성 요소들 간의 인증(Authentication), Authorization, Network Isolation, Data Protection, 그리고 Auditing/Diagnose 등의 다양한 보안 기능들을 제공하여야 한다.

DataLake의 Data Acquisition Layer에서 기대되는 주요한 역할 중 하나는 데이터를 DataLake에서 추가 처리 할 수 있는 메시지로 변환 할 수 있다는 것이며, 따라서 Data Acquisition Layer는 다양한 스키마 사양을 수용 할 수 있는 유연성(flexibility)이 있어야 하며, 동시에 모든 변환된 데이터 메시지(translated data messages)를 DataLake에 원활하게 푸시 할 수 있는 빠른 연결 메커니즘을 제공할 수 있어야 한다.

DataLake의 Messaging Layer는 Data Acquisition Layer에 연결된 외부의 여러 연결들을 분리하는 주요한 계층임과 동시에 메시지의 전달을 보장하여야 한다. 메시지의 전달을 보장하기 위하여 메시지의 지속성(persistent)을 가져야 하며, 메시지의 지속성은 대개 스토리지 매체에서 지원하게 된다.

DataLake의 빠른 Ingestion Layer는 람다 아키텍처를 위한 주요한 계층이며, 이 계층은 람다 아키텍처의 작업 모델(working model)로 데이터를 전달하는 속도를 제어할 수 있어야 한다.

람다 아키텍처는 배치 데이터 처리와 실시간 데이터 처리의 병합 문제를 해결할 수 있으며, 고성능 분산 컴퓨팅과 확장성으로 대규모 데이터 세트를 일괄적으로 준 실시간 처리를 통해 일관된 데이터를 제공할 수 있는 방법을 제공한다. 특히 람다 아키텍처는 저 지연(low latency)에 의한 엔터프라이즈 데이터의 다양한 데이터로드 프로파일(data load profile)을 통한 스케일-아웃(scale-out) 아키텍처를 구현 가능한 방법을 정의하고 있다. 람다 아키텍처의 내부에는 제안하는 DataLake Framework의 주요한 특성을 포함하고 있으며, TDA Layer([그림 9]의 ① 참조), Batch Layer, Stream Layer, Intelligent Layer([그림 9]의 ② 참조), 그리고 Metadata & content

Smart Media Journal / Vol.7, No.1 / ISSN:2287-1322
http://dx.doi.org/10.30693/SMJ.2018.7.1.14

Layer([그림 9]의 ③ 참조)로 구성된다. 람다 아키텍처의 TDA Layer는 Topology data analysis[12]를 통한 원시 데이터를 모델링된 데이터(modeled data)로 변환하는 것이 이 계층의 주요 책임과 동시에 다른 Data Lake 모델들과의 차별화된 핵심 기능이며, 모델링된 데이터는 람다 아키텍처의 Serving Layer에 의한 제공 가능한 데이터 모델을 의미한다. 람다 아키텍처의 Intelligent Layer는 고품질의 모델링된 데이터를 생성하기 위해 수집된 원시 데이터를 기반으로 기계 학습 및 데이터 과학 처리를 Batch Layer와 Stram Layer에 지원하게 된다. 또한 고품질의 모델링된 데이터를 Distributed Storage Layer에 저장 및 사용자들과 소비 시스템을 위한 Service Layer에 제공하게 된다. 람다 아키텍처의 Metadata & Content Layer는 DataLake 운영을 위한 생성된 메타데이터의 저장/관리/검색 기능 등의 메타 데이터 관리 기능을 수행하며, DataLake를 구성하는 여러 Layer들의 다양한 역할을 수행할 S/W들을 가상화(Virtualization) 기술에 의한 CI(content integration)/CD(content delivery)/CD(content deployment) 기능을 제공하게 되며, 이러한 기능은 다른 Data Lake 모델들과의 차별화를 제공한다. 그리고 람다 아키텍처의 Metadata & Content Layer에서 제공되는 Cloud Bursting[13]과 Cloud Spanning[14]은 애플리케이션 작업 부하에 대한 원활한 운영과 확장성을 보장하도록 설계되었다.

DataLake의 Serving Layer는 DataLake로부터 데이터를 제공하는 주요 책임을 가지며, 특히 람다 아키텍처로부터의 생성된 데이터를 사용자 또는 소비하는 애플리케이션에게 데이터를 어떻게 전달 및 제공하는가가 중요하게 된다. 데이터는 시스템들 간의 다중 방법으로 전송이 가능하며, 데이터 전송을 위한 일반적인 방법은 서비스를 통하는 방법이며, 이러한 서비스는 주로 데이터를 제공 할 수 있는 데이터 서비스(data services)라고 한다.

Ⅳ. Abyss Storage Cluster 기반 DataLake Framework의 Cloud Spanning의 가능성 검증

Abyss Storage Cluster의 DataLake Framework의 Metadata & content Layer의 정의된 기능들([그림 9] 참조) 중의 하나인 Cloud Spanning의 가능성 검증을 위한 데이터 전송 테스트는 Dell T-620 서버와 Google 클라우드 스토리지 간의 원격지 데이터 전송 테스트를 수행하였다. Google Storage의 버킷을 통한 데이터 크기별로 10MB, 100MB, 1GB, 2GB, 5GB의 파일 업로드 및 다운로드 테스트를 실시하며, 테스트 결과를 [그림 10]에 나타내었다.

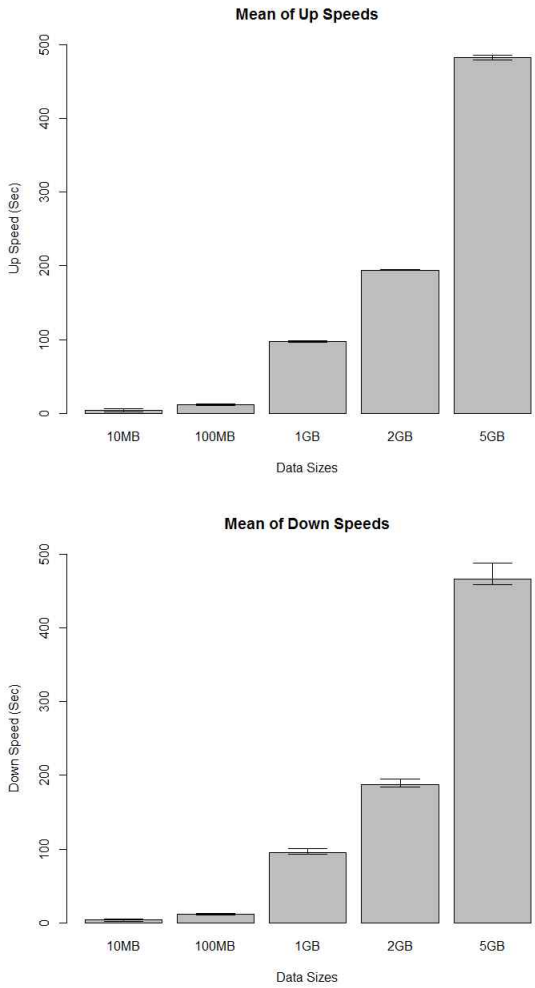

그림 10. Google Storage 버킷의 데이터 업로드와 다운로드 테스트 결과의 분석

[그림 10]은 통계 패키지 R[15]을 이용한 Google Storage의 데이터 업로드와 다운로드 테스트 결과를 분석하였으며, Bar 차트는 평균 속도를 의미하며, 상단의 두 개의 선은 데이터 전송 속도의 최댓값과 최솟값을 나타낸 것이다. 대체적으로 데이터 전송 용량이 커짐에 따른 속도의 분산(variance) 또한 커지는 경향을 보였으며, 이것은 Metadata & content Layer의 Cloud Spanning 구현을 위한 데이터 전송 측면에서 네트워크 상황에 따른 영향력이 매우 크다는 것을 간접적으로 입증하고 있다.

Ⅴ. 결 론

디지털 시대에는 고객의 특성이 끊임없이 변하며, 비즈니스적인 요구 사항을 이해하고 이를 기술적 요구 사항으로 변환이 필수적이다. BI(Business Intelligent)를 지원하기 위한 전단부의 Desktop 또는 모바일의 BI 상황보다는 후단부의 데이

터 모델링을 위한 빅데이터 기반 데이터레이크 플랫폼에 관한 연구가 중요하다. 본 논문에서는 Abyss Storage Cluster 기반의 DataLake Framework의 구성 요소들을 설계하였으며, 각각의 기능들을 간략하게 기술하였다. 또한 본 설계의 특징으로는 클라우드 컴퓨팅의 장점 중의 하나인 Cloud Bursting과 Cloud Spanning 기술을 접목하였다. 그리고 제안된 DataLake Framework의 기능 중의 하나인 Cloud Spanning의 가능성을 Google Storage의 버킷을 이용하여 검증 및 분석하였다.

REFERENCES

[1] Tomcy John and Pankaj Misra, "Data Lake for Enterprises - Leveraging Lambda Architecture for Building Enterprise Data Lake," Packt Publishing, May 2017.
[2] IBM의 빅데이터 정의, http://www.ibmbigdatahub.com/infographic/four-vs-big-data
[3] 장동인, "빅데이터로 일하는 기술," 한빛미디어, 2014년 12월 16일.
[4] Mike barlow, "Real-Time Big Data Analytics: Emerging Architecture," 1st Edition, O'Reilly, Feb. 2013.
[5] Pradeep Pasupuleti, Beulah Salome Purra, "Data Lake Development with Big Data," PACKT Publishing, 2015.
[6] John Mallory and Robbie Wright, "Building Big Data Storage Solutions (Data Lakes) for Maximum Flexibility," Amazon Web Service, July 2017.
[7] AWS, http://docs.aws.amazon.com/solutions/latest/data-lake- solution/architecture.html
[8] AWS, https://aws.amazon.com/ko/big-data/data-lake-on-aws/
[9] 차윤석 외 4인, "Abyss Storage의 Disk 타입에 의한 Ceph RADOS의 Benchmarking," 2017 한국통신학회 동계학술대회.
[10] 차병래 외 4인, "대용량 Abyss Storage의 KOREN 네트워크 기반 국내 및 해외 실증 테스트," 스마트미디어학회저널 Vol.6, no.1, pp.9-15, 2017년 3월호.
[11] Lambda Architecture, http://searchbusinessanalytics.techtarget.com/definition/Lambda-architecture
[12] 차병래 외 4인, "Idea Sketch to Improvement Image Learning based on Machine Learning using Topology Theory," SMA 2017.
[13] Cloud Bursting, http://searchcloudcomputing.techtarget.com/definition/cloud-bursting
[14] Cloud Spanning, http://searchcloudcomputing.techtarget.com/definition/cloud-spanning
[15] R, https://www.r-project.org/

저 자 소 개

차병래

2004년 목포대학교 대학원 컴퓨터공학과 졸업(공학박사)
2005년 호남대학교 컴퓨터공학과 전임강사
2009년 ~ 현재 광주과학기술원 전기전자컴퓨터공학부 연구조교수
2012년 ~ 현재 제노테크(주) 대표
<주관심분야: 정보보안, IDS, Neural Network, Cloud Computing, VoIP, NFC, 대용량 스토리지 기술 등>

박 선

2007년 인하대학교 컴퓨터정보공학과 공학박사
2008년 호남대학교 컴퓨터공학과 전임강사
2010년 전북대학교 인력양성사업단 박사후 과정
2010년 목포대학교 정보산업연구소 연구전임교수
2013년 ~현재 광주과학기술원 NetCS연구실 연구교수
<주관심분야: 정보검색, 데이터마이닝, 해양IT정보융합, 클라우드 컴퓨팅, IoT, 스토리지 시스템>

신병춘

2002년 전남대학교 수학과 조교수
2011년 전남대학교 수학과 교수
<주관심분야 : 수치해석, 인공신경망, 컴퓨터 비전>

김종원

1997년 University of Southern California 연구 조교수
1999년 Technology Consultant for VProtect Systems Inc.
2000년 Technology Consultant for Southern California Division of InterVideo Inc.
2001년 광주과학기술원 전기전자컴퓨터공학부 교수
2008년 ~ 현재 광주과학기술원 전기전자컴퓨터공학부 교수
<주관심분야: Future Internet, SDN & NFV, SDI>