한국정보전자통신기술학회논문지(jkiiect)'18-02, Vol.11 No.1
논문 18-11-01-70 http://dx.doi.org/10.17661/jkiiect.2018.11.1.70

스마트 팩토리 환경에서의 GlusterFS 기반 빅데이터 분산 처리 시스템 설계

이협건*, 김영운, 김기영, 최종석

Design of GlusterFS Based Big Data Distributed Processing System in Smart Factory

Hyeop-Geon Lee*, Young-Woon Kim, Ki-Young Kim

요 약 스마트 팩토리는 설계·개발, 제조, 유통·물류 등 생산 전체 과정에 정보 통신 기술을 적용하여 생산성, 품질, 고객만족도 등을 향상시킬 수 있는 지능형 공장이다. 스마트 팩토리에서 발생되는 데이터의 양은 공장의 규모 및 시설 수준에 따라 많은 차이를 보이지만, 기존의 생산관리시스템을 활용하여 방대한 양의 데이터를 발생시키는 스마트 팩토리 환경에 적용하기에 어려움이 있다. 이로 인해 방대한 양의 빅데이터 처리할 수 있는 빅데이터 분산 처리 시스템의 필요성이 요구되고 있다. 따라서 본 논문에서는 스마트 팩토리 환경에서의 GlusterFS 기반 빅데이터 분산 처리 시스템 설계하였다. 제안하는 빅데이터 분산 처리 시스템은 기존 분산 처리 시스템에 비해 네트워크 트래픽 분산 및 관리를 통해 부하와 데이터 소실 위험도를 감소시켰다.

Abstract Smart Factory is an intelligent factory that can enhance productivity, quality, customer satisfaction, etc. by applying information and communications technology to the entire production process including design & development, manufacture, and distribution & logistics. The precise amount of data generated in a smart factory varies depending on the factory's size and state of facilities. Regardless, it would be difficult to apply traditional production management systems to a smart factory environment, as it generates vast amounts of data. For this reason, the need for a distributed big-data processing system has risen, which can process a large amount of data. Therefore, this article has designed a Gluster File System (GlusterFS)-based distributed big-data processing system that can be used in a smart factory environment. Compared to existing distributed processing systems, the proposed distributed big-data processing system reduces the system load and the risk of data loss through the distribution and management of network traffic.

Key Words : Big Data, Hadoop, GlusterFS, Smart Factory, Big Data Distributed Processing System

1. 서론

최근 4차 산업혁명으로 인해 제조업은 스마트 팩토리 환경으로 변화하고 있으며, 설계·개발, 제조, 유통·물류 등 생산 전체 과정에 정보 통신 기술을 적용하고 있다. 스마트 팩토리는 빅데이터 및 사물인터넷 기술들을 활용하여 생산성, 품질, 고객만족도 등을 향상시키고 있다. 이러한 스마트 팩토리는 공정마다 방대한 양의 데이터를 발생하고 있으며, 이 데이터를 수집, 처리,

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(NRF-2017R1D1A1B03034729)
*Department of Data Analysis, Seoul Gangseo Campus of Korea Polytech
Department of Data Analysis, Seoil University
Department of Computer, Soongsil University
**Corresponding Author : Department of Data Analysis, Seoul Gangseo Campus of Korea Polytech(hglee67@kopo.ac.kr)
Received January 29, 2018 Revised February 06, 2018 Accepted February 15, 2018

분석에 중요성이 대두되고 있다. 따라서 방대양의 데이터를 처리할 수 있는 빅데이터 분산 처리 시스템의 필요성이 요구되고 있다[1, 2].

그러나 기존의 스마트 팩토리 환경에 적용되는 대부분 분산 처리 시스템들은 데이터 규모 및 분석에 대해 크게 고려하지 않고 구현하였으며, 특히 데이터 손실에 대처하기 위한 기술인 데이터 다중화를 지원하지 않거나, 일배치를 통해 다중화를 수행하기 때문에 물리적 저장소 파괴 현상과 같은 예기치 못한 장애에 대응이 어렵다.

이에 본 논문에서는 빅데이터 파일 관리를 위한 GlusterFS 기반 분산 파일 관리 시스템을 설계하였다. GlusterFS 기반 분산 파일 관리 시스템은 대용량 트래픽에 대처하기 위해 Scale Out으로 서버를 구축하고 로드 밸런싱을 트래픽을 분산시켰으며, 네트워크 부하로 인한 서비스를 장애를 방지하기 위해 트래픽 속도 관리하였다. 또한 GlusterFS 기반 분산 파일 관리 시스템은 빈번한 IO(Input/Output)로 인한 물리적 저장소의 파괴 현상에 대처하기 위해 GlusterFS를 이용하여 저장되는 파일들을 3중화 처리하여 파일 소실에 대한 위험도를 감소시켰으며, DB를 통해 저장되는 파일정보의 분산 처리 및 데이터 보호를 위해 MySql의 마스터-슬레이브 구조로 구현하였다.

본 논문의 구성은 다음과 같다. 2장에서는 파일 관리 시스템의 요구사항과 GlusterFS에 대해 살펴본다. 3장에서는 본 논문에서 제안하는 대용량 파일 전송을 위한 파일 관리 시스템의 설계한다. 마지막 4장에서는 결론 및 향후 연구 과제를 제시한다.

2. 관련 연구

본 장에서는 빅데이터 분산 처리 시스템에 적용될 분산 파일 시스템을 연구하기 위해 GlusterFS에 대해 살펴보고, 이 결과를 기반으로 빅데이터 분산 처리 시스템에 필요한 요구사항을 도출한다.

2.1 GlusterFS

GlusterFS[3]는 오픈 소소로서, 파일 시스템들을 저장 서버들을 클러스터링하여 페타바이트 급 데이터 저장 및 처리가 가능한 분산 파일 시스템이다. 또한 GlusterFS는 네트워크 대역폭만 가능하다면, 동시에 수 천개이상 클라이언트들이 접속하여 사용할 수 있다.

GlusterFS의 특징은 Hadoop Distributed File System과 같이 다중화(Replication) 구성이 가능하기에 데이터의 안정성이 뛰어나다. 이 외에도 GlusterFS는 여러 OS와의 호환성 등이 우수하기 때문에 다양한 클라우드 인프라 구성에 많이 사용된다[4].

GlusterFS[5, 6]와 Hadoop Distributed File System의 가장 큰 차이점은 인프라 환경 구성과 메타데이터의 사용 유무이다. 인프라 환경 구성은 앞서 설명한 Hadoop Distributed File System에 비해 GlusterFS가 비교적 인프라 환경 구성이 간단하며, 대용량, 대규모의 I/O 처리 능력이 뛰어나다. 메타데이터는 분산 파일 시스템에 저장된 파일의 위치 및 관련 정보들이 저장되는 데이터로 Hadoop Distributed File System에 사용되지만, GlusterFS는 메타데이터를 사용하지 않는다. 이로 인해 GlusterFS는 별도의 메타데이터를 관리하기 위한 데이터 저장소가 필요하다. 그러나 이러한 특징으로 인해 메타데이터 관리를 다양한 플랫폼 및 기술에 적용이 가능하여 Hadoop Distributed File System에 비해 유연하게 사용할 수 있다.

2.2 빅데이터 분산 처리 시스템 요구사항

스마트 팩토리 환경에서의 빅데이터 분산 처리 시스템이 갖추어야 할 사항들은 인프라 환경 측면에서 크게 4가지로 구분된다.

첫째, 스마트 팩토리 환경에서의 빅데이터 분산 처리 시스템들이 처리하는 데이터의 양과 서버의 규모는 기존의 분산 파일 시스템들과는 비교가 되지 않을 정도로 규모가 크다. 또한 빅데이터 분산 처리 시스템의 대부분의 핵심적인 사항들은 비용적인 측면에서의 효율성, 지속적으로 증가하는 데이터들의 수용, 빈번하게 발생하는 장애에 대한 대처와 인프라 관리의 편리성이 요구된다.

둘째, 스마트 팩토리 환경에서의 빅데이터 분산 처리 시스템은 다양한 사용자들의 요구를 충족시키기 위해 데이터의 입·출력과 높은 처리 성능이 요구된다. 또한 빅데이터 파일 관리를 위한 분산 파일 관리 시스템

은 대용량 파일에 대한 신속한 입출력 성능, 네트워크 구조 인식을 통한 저장되는 데이터들의 최적 배치, 보다 빠른 처리를 위한 효과적인 캐시의 사용, 순간적으로 집중되는 부하의 유연한 대처와 같은 사항들이 요구된다.

셋째, 스마트 팩토리 환경에서의 빅데이터 분산 처리 시스템은 데이터에 대한 암호화, 데이터 영역에 대한 사용자간 엄격한 접근 제어, 사용자 데이터에 대한 관리자의 접근 제어와 같은 사항들이 요구된다.

넷째, 스마트 팩토리 환경에서의 빅데이터 분산 처리 시스템은 사용자가 저장한 데이터에 오류가 발생하지 않도록 방지하고, 저장 공간을 최적으로 사용하기 위한 방법들이 요구된다.

3. 제안하는 빅데이터 분산 처리 시스템

[그림 1]은 본 논문에서 제안하는 분산 파일 관리 시스템의 구조를 나타낸다.

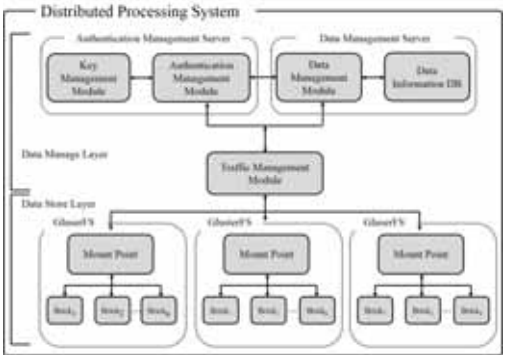

그림 1. 제안하는 빅데이터 분산 처리 시스템 구조
Fig. 1. Architecture of proposal big data distributed processing system

제안하는 빅데이터 분산 처리 시스템은 크게 데이터 관리 영역과 데이터 저장 영역으로 구성된다. 데이터 관리 영역에는 인증 관리 서버, 저장 데이터 관리 서버와 트래픽 관리 서버로 구성되며, 데이터 저장 영역은 데이터 저장 서버들로 구성된다.

3.1 데이터 관리 영역

데이터 관리 영역은 제안하는 빅데이터 분산 처리 시스템에 보안 관련한 접근 제어를 위한 인증 관리 서버, 저장된 데이터들의 메타정보가 저장 및 관리되는 저장 데이터 관리 서버와 데이터 관리 영역의 데이터 저장 서버들의 네트워크 부하를 관리하기 위한 트래픽 관리 서버로 구성된다.

3.1.1 인증 관리 서버

인증 관리 서버는 제안하는 빅데이터 분산 처리 시스템에 보안 관련한 접근 제어 기능을 수행한다.

인증 관리 서버는 키 관리 모듈과 인증 관리 모듈로 구성된다. 키 관리 모듈은 인증을 위해 사용되는 보안키를 관리하는 역할을 수행한다. 키 관리 모듈에서 사용되는 보안키는 빅데이터 분산 처리 시스템의 관리자에 의해 주기적으로 변경되며, 이전에 사용된 보안키는 최대 3회 이상 변경이 이루어진 경우에만 사용할 수 있도록 설계하였다. 인증 관리 모듈은 키 관리 모듈로부터 관리되는 보안키를 활용하여 빅데이터 분산 처리 시스템에 접근하는 사용자들의 접근 제어를 수행한다. 특히 인증 관리 모듈은 불법 접근 및 비정상적인 패턴을 보이는 사용자들에 대해 블랙리스트로 등록 및 관리를 수행한다.

3.1.2 데이터 관리 서버

데이터 관리 서버는 데이터 저장 영역에서 사용되는 저장데이터들의 메타정보를 저장 및 관리를 수행한다. 저장되는 메타정보는 무료 관계형 데이터베이스관리시스템인 MySQL에 저장된다. 데이터 관리 서버는 빅데이터의 특성상 많은 데이터의 저장으로 인한 트래픽 부하 발생에 대처하기 위해 반드시 메타정보의 분산 처리가 필요하다. 메타정보의 분산 처리는 MySQL을 마스터/슬레이브로 구성한 뒤, 이 세트들을 3중화하여 구성하여 Hadoop Distributed File System의 단점인 메타정보의 물리적 취약점을 해결하며, 예기치 못한 장애에 대처한다. 또한 메타정보의 분산 처리는 마스터 서버에서 읽기 트래픽만 처리하고, 슬레이브 서버에서 쓰기 트래픽만 처리하도록 역할을 분리한다.

3.1.3 트래픽 관리 서버

트래픽 관리 서버는 데이터 저장 영역의 데이터 저장 서버에 저장되는 데이터들이 하나의 데이터 저장 서버에 집중적으로 저장되지 않도록 관리하는 역할을 수행한다. 트래픽 관리 서버는 네트워크 부하 및 데이터 저장 서버의 현재 진행 중인 저장 인스턴스의 수, 저장된 데이터의 크기와 파일의 개수를 분석하여 최적화된 저장 경로를 정의한다. [그림 3]은 트랙픽 관리를 위한 최적화된 저장 경로 알고리즘의 Pseudo 코드를 나타낸다.

트랙픽 관리를 위한 최적화된 저장 경로 알고리즘은 앞서 설명한 저장 서버별 진행중 저장 인스턴스 수, 저장 서버별 저장된 데이터 크기와 저장 서버별 저장된 블릭 수를 기반으로 가중치를 부여하여 저장될 서버를 선정한다. 선정된 서버는 파일을 저장한다.

3.2 데이터 저장 영역

데이터 저장 영역은 데이터 저장 서버들로 구성된 영역이다. 데이터 저장 서버들에 적용된 분산 파일 시스템은 GlusterFS를 적용한다. 제안하는 시스템은 데이터 손실률을 최소화하기 위해 GlusterFS에서 제공하는 복제 기능을 이용하여 3중화 처리를 한다. 2중화가 아닌 3중화를 선택한 이유는 운영 중인 블릭 중 1개의 1블릭에 문제가 생겼을 경우, 모든 트래픽이 남은 1개의 블릭으로 몰려, 장비 복구가 완료되지 않은 상황에서 남은 1개의 블릭마저 장애가 발생하는 현상을 줄이기 위해서이다. 따라서 3중화는 1개의 블릭에 문제가 발생해도 나머지 2개의 블릭에서 처리하기 때문에 장애의 위험도가 2중화보다 뛰어나다.

```
1  StorePathAlgorithm{
2     long iCnt = DataStoreServers,getStoreInstance
3     long iDataSize = DataStoreServers,getDataSize
4     long iFileCnt = DataStoreServers,getBrickCount
5
6     //3가지 변수를 기반으로 가중치 계산
7     weight(iCnt, iDataSize, iFileCnt);
8     //파일 저장
9     storeData();
10 }
11
12 //저장 서버별 진행중 저장 인스턴스 수 가져오기
13 DataStoreServers,getStoreInstance(
14    for(int i=0; i<DataStoreServers.getCount;i++){
15       List instance = i.getStoreInstrance();
16    }
17 );
18
19 //저장 서버별 저장된 데이터 크기 가져오기
20 DataStoreServers,getDataSize(
21    for(int i=0; i<DataStoreServers.getCount;i++){
22       List instance = i.getStoreDataSize();
23    }
24 );
25
26 //저장 서버별 저장된 블릭 수 가져오기
27 DataStoreServers,getBrickCount(
28    for(int i=0; i<DataStoreServers.getCount;i++){
29       List instance = i.getStoreBrickCount();
30    }
31 );
```

그림 2. 저장 경로 알고리즘의 Pseudo 코드
Fig. 2. Pseudo code of the store path algorithm

4. 성능평가

본 논문에서의 성능평가는 제안하는 시스템의 네트워크 부하율과 데이터 손실률을 예측하여 분석한다.

4.1 네트워크 부하율

네트워크는 부하율은 데이터 저장시 발생되는 지연시간을 측정과 저장되는 데이터의 크기를 점층적으로 증가시켜 제안하는 시스템의 네트워크 부하를 Poisson 분포를 활용하여 분석한다.

지연시간은 T_d, 저장 경로 알고리즘 수행시간은 T_a, 데이터 저장시간은 T_{store}, 저장되는 데이터는 $\sum_{d=1}^{n} 3d = \frac{d^3}{n}(n+1)$ 로 정의하며 지연시간은 수식(1)

과 같이 나타낸다.

$$T_d = T_a + T_{store} \quad \text{수식(1)}$$

증가되는 데이터의 크기에 따른 네트워크 지연시간은 수식(2)와 같이 나타낸다.

$$T_d = \frac{T_a + T_{store} \times \frac{d^3}{n}(n+1)}{x!} \quad \text{수식(2)}$$

[그림 4]와 [표 1]은 네트워크 지연시간을 나타낸다.

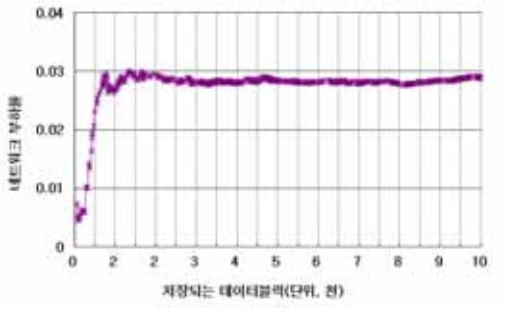

그림 3. 네트워크 부하율
Fig. 3. Rate of network load

표 1. 네트워크 부하율
Table 1. Rate of network load

데이터 블럭	0-1	1-2	2-3	3-4	4-5
네트워크 부하율	0.0128	0.0298	0.0289	0.0291	0.0273
데이터 블럭	5-6	6-7	7-8	8-9	9-10
네트워크 부하율	0.0284	0.0281	0.0291	0.0288	0.0289

4.2 데이터 손실률

데이터 손실률은 다중화 및 초기 블릭에 정상적으로 저장되지 않는 확률을 계산한 것으로 앞서 설명한 네트워크 부하율과 동일하게 Poisson 분포를 활용하여 분석한다. 데이터가 저장되는 동안 손실되는 데이터는 l, 평균적으로 발생하는 손실 횟수는 α, 저장되는 블릭 수는 n, 데이터 손실률은 p로 정의한다. 데이터 손실률은 수식(3)과 같이 나타낸다.

$$P_l = \frac{\alpha^{l-\alpha}}{l!} = e^{-\alpha}\prod_{l=1}^{l}(\frac{\alpha}{l}) \;\; \alpha = np \quad \text{수식(3)}$$

[그림 5]와 [표 2]는 데이터 손실률을 나타낸다.

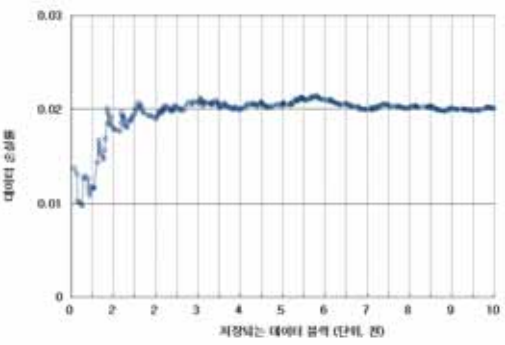

그림 4. 데이터 손실률
Fig. 4. Rate of brick loss

표 2. 데이터 손실률
Table 2. Rate of brick loss

데이터 블럭	0-1	1-2	2-3	3-4	4-5
데이터 손실률	0.0113	0.0149	0.0198	0.0211	0.0215
데이터 블럭	5-6	6-7	7-8	8-9	9-10
데이터 손실률	0.0221	0.0225	0.0233	0.0221	0.0211

분석 결과, 데이터 손실률은 초반에 저장되는 블릭이 적었으나 데이터 손실률은 급격하게 증가되었다. 그러나 저장되는 블릭이 2,000개 이상이 될 때 데이터 손실률은 최대 0.0241로 크게 변화가 없었다. 따라서 제안하는 시스템은 스마트 팩토리 환경과 같이 동시에 저장되는 블릭이 많은 환경에 적용함에 있어 적합하다.

5. 결론

본 논문에서는 스마트 팩토리 환경에서의 GlusterFS 기반 빅데이터 분산 처리 시스템을 설계하였다. 제안하는 시스템은 스마트 팩토리 환경에서의 방대한 양의 트

래픽에 대처하기 위해 Scale Out으로 서버를 구축하여 트래픽을 분산시켰으며, 네트워크 부하로 인한 서비스를 장애를 방지하기 위해 트래픽 속도 관리하였다. 또한 GlusterFS 기반 분산 파일 관리 시스템은 빈번한 IO로 인한 물리적 저장소의 파괴 현상에 대처하기 위해 GlusterFS를 이용하여 저장되는 파일들을 3중화 처리하여 파일 소실에 대한 위험도를 감소시켰으며, DB를 통해 저장되는 파일정보의 분산 처리 및 데이터 보호를 위해 MySql의 마스터/슬레이브 구조로 구현하였다. 향후 본 설계를 기반으로 시스템이 구현이 필요하다.

REFERENCES

[1] Hye-Jung Chang and Do-Nyun Kim, "A Study on big data utilization for implementation of the resident participation type safe community planning of the smart city," Journal of Korea Institute of Information, Electronics, and Communication Technology, Vol. 9, No. 5, pp. 478~495, Oct, 2016.

[2] In-Hak Joo, "Spatial Big Data Query Processing System Supporting SQL-based Query Language in Hadoop," Journal of Korea Institute of Information, Electronics, and Communication Technology, Vol. 10, No. 1, pp. 1~8, Feb, 2017.

[3] Eun-Hee Jeong and Byung-Kwan Lee, "A Design of Hadoop Security Protocol using One Time Key based on Hash-chain," Journal of Korea Institute of Information, Electronics, and Communication Technology, Vol. 10, No. 4, pp. 340~349, Aug, 2017.

[4] Jae-Hyuck Kwak, Sangwan Kim, Taesang Huh and Soonwook Hwang, "Implementation and Performance Analysis of Hadoop MapReduce over Lustre Filesystem," KIISE Transactions on Computing Practices, Vol. 21, No. 8, pp. 561~566, Aug, 2015.

[5] Deoksang Kim, Hyeonsang Eom and Heonyoung Yeom, "Performance Optimization in GlusterFS on SSDs," KIISE Transactions on Computing Practices, Vol. 22, No. 2, pp. 95~100, Feb, 2016.

저자약력

이 협 건(Hyeop-Geon Lee) [종신회원]

2011.03 – 2015.08,
숭실대학교 일반대학원
컴퓨터학과 공학박사
2015.12 – 현재,
한국폴리텍대학 서울강서캠퍼스
데이터분석과 조교수

<관심분야>
빅데이터, 실시간 분석, 데이터분석

김 영 운(Young-Woon Kim) [종신회원]

2004.09 – 2018.08,
숭실대학교 일반대학원
컴퓨터학과 공학박사
2015.12 – 현재,
한국폴리텍대학 서울강서캠퍼스
데이터분석과 조교수

<관심분야>
빅데이터, 실시간 분산 처리, 데이터분석

김 기 영(Ki-Young Kim) [종신회원]

2004.03 - 현재,
서일대학교 소프트웨어공학과
부교수

<관심분야>
빅데이터, 무선통신, 사물인터넷

최 종 석(Jong-Seok Choi) [정회원]

2010.03 – 2012.02,
숭실대학교 일반대학원
공학석사
2013.03 – 2015.02,
숭실대학교 일반대학원
공학박사수료

<관심분야>
빅데이터, 무선통신, 5G