KIPS Tr. Software and Data Eng.
Vol.5, No.6 pp.283~288 pISSN: 2287-5905

http://dx.doi.org/10.3745/KTSDE.2016.5.6.283

Anomaly Detection of Hadoop Log Data Using Moving Average and 3-Sigma

Siwoon Son[†] · Myeong-Seon Gil[††] · Yang-Sae Moon[†††] · Hee-Sun Won[††††]

ABSTRACT

In recent years, there have been many research efforts on Big Data, and many companies developed a variety of relevant products. Accordingly, we are able to store and analyze a large volume of log data, which have been difficult to be handled in the traditional computing environment. To handle a large volume of log data, which rapidly occur in multiple servers, in this paper we design a new data storage architecture to efficiently analyze those big log data through Apache Hive. We then design and implement anomaly detection methods, which identify abnormal status of servers from log data, based on moving average and 3-sigma techniques. We also show effectiveness of the proposed detection methods by demonstrating that our methods identifies anomalies correctly. These results show that our anomaly detection is an excellent approach for properly detecting anomalies from Hadoop log data.

Keywords : Big Data, Apache Hadoop, Apache Hive, Log Data, Anomaly Detection

이동 평균과 3-시그마를 이용한 하둡 로그 데이터의 이상 탐지

손 시 운[†] · 길 명 선[††] · 문 양 세[†††] · 원 희 선[††††]

요 약

최근 빅데이터 처리를 위한 연구들이 활발히 진행 중이며, 관련된 다양한 제품들이 개발되고 있다. 이에 따라, 기존 환경에서는 처리가 어려웠던 대용량 로그 데이터의 저장 및 분석이 가능해졌다. 본 논문은 다수의 서버에서 빠르게 생성되는 대량의 로그 데이터를 Apache Hive에서 분석할 수 있는 데이터 저장 구조를 제안한다. 그리고 저장된 로그 데이터로부터 특정 서버의 이상 유무를 판단하기 위해, 이동 평균 및 3-시그마 기반의 이상 탐지 기술을 설계 및 구현한다. 또한, 실험을 통해 로그 데이터의 급격한 증가폭을 나타내는 구간을 이상으로 판단하여, 제안한 이상 탐지 기술의 유효성을 보인다. 이 같은 결과를 볼 때, 본 연구는 하둡 기반으로 로그 데이터를 분석하여 이상치를 바르게 탐지할 수 있는 우수한 결과라 사료된다.

키워드 : 빅데이터, 아파치 하둡, 아파치 하이브, 로그 데이터, 이상 탐지

1. 서 론

최근 소셜 네트워크 서비스(SNS, social network service), 사물 인터넷(IoT, Internet of Things) 등 대용량 데이터를 발생시키는 서비스가 급증함에 따라 빅데이터[1-3]가 화두에 올랐다. 빅데이터란 다양한 형태로 빠르게 발생하는 대용량 데이터, 또는 이러한 데이터를 저장·가공하기 위한 기술을 의미한다. 하둡(Hadoop)[4-6]은 이러한 빅데이터를 처리하기 위해 사용되는 대표적인 소프트웨어 프레임워크로, 데이터를 분산 저장하는 HDFS(Hadoop distributed file system) [7-9]와 저장된 데이터를 분산 환경에서 처리하는 맵리듀스(MapReduce)[10-15]로 구성된다. 그러나 하둡을 단독으로 데이터 처리에 사용하기 위해서는 기존 코드를 맵리듀스로 변환해야 하는 어려움이 있다. 따라서 일반적으로 Apache HBase나 Apache Hive[16-18] 등 기존 RDBMS(relational database management system)와 유사한 환경을 지원하는 별도의 시스템과 함께 하둡 에코시스템을 구성하여 사용한다.

빅데이터는 대부분 비정형 데이터로 구성되며, 이는 텍스트, 문서, 이미지, 동영상 등의 여러 형태를 포함한다. 이러한 비정형 데이터 중에서도 시스템의 상태를 기록하는 로그(log)는 대표적인 빅데이터에 해당한다. 중소규모 이상의 기

※ 이 논문은 2014년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2014R1A2A2A01002548).
※ 2015년도 강원대학교 대학회계 학술연구조성비로 연구하였음 (관리번호-D1000463-01-01).
† 준 회 원 : 강원대학교 컴퓨터과학과 석사과정
†† 준 회 원 : 강원대학교 컴퓨터과학과 박사과정
††† 종신회원 : 강원대학교 컴퓨터과학과 교수
†††† 정 회 원 : 한국전자통신연구원 책임연구원
Manuscript Received : May 17, 2016
First Revision : May 30, 2016
Accepted : May 30, 2016
* Corresponding Author : Yang-Sae Moon(ysmoon@kangwon.ac.kr)

업과 연구 기관에서는 필요에 따라 다수의 서버를 구축하여 사용하는데, 서버의 증가에 따라 로그를 기반으로 한 모니터링에 많은 비용이 요구된다. 시스템 모니터링의 주요 목적은 분석을 통한 이상 탐지, 시스템 사용량 파악 등이 있으며, 특히 이상 탐지의 수행을 위한 여러 가지 방법들이 연구되어 왔다. 본 논문은 이 중 이동 평균(moving average)[19-22] 및 3-시그마[23, 24] 기반의 이상 탐지 기법을 로그 데이터에 적용하고, 이를 위해 Hive를 사용한다. 또한, 해당 이상 탐지 기법에 가중치를 부여하여 더 정확한 탐지를 수행하는 기능을 설계·개발한다. 이를 통해 여러 시스템에서 이상이 발생한 시점을 효과적으로 파악할 수 있으며, 앞으로 발생 가능한 이상을 예측할 수 있으므로 시스템 관리자가 미리 대처할 수 있다.

본 논문의 구성은 다음과 같다. 먼저 제2절은 본 논문에서 사용하는 기술의 관련 연구를 설명한다. 그리고 제3절은 이동 평균 및 3-시그마 기반의 로그 데이터 이상 탐지 기법을 Hive에서 사용할 수 있도록 설계하고 제4절에서 이 이상 탐지 기법을 실험 및 평가한다. 마지막으로 제5절은 결론 및 향후 연구로 본 논문을 마친다.

2. 관련 연구

하둡은 대용량 데이터를 저장하기 위한 HDFS와 저장된 데이터를 처리 및 분석하기 위한 맵리듀스로 구성된다. HDFS에서 데이터는 네임노드(NameNode)에 의해 특정 크기(기본 128MB)의 블록으로 나뉘어져, 여러 대의 데이터노드(DataNode)에 분산 저장된다. 여기서 네임노드는 HDFS의 메타데이터를 관리하고, 클라이언트가 HDFS에 저장된 파일에 접근할 수 있도록 마스터 역할을 수행한다. 데이터노드는 블록으로 나뉘어진 데이터를 로컬 파일 시스템에 저장하며, 주기적으로 네임노드에게 하트비트(heartbeat)와 블록 리포트(block report)를 전송한다. 이를 통해 네임노드는 데이터노드의 고장 여부 및 저장된 블록에 대한 정보를 얻을 수 있다. 이렇게 HDFS에 저장된 대용량 데이터는 기본적으로 맵리듀스를 통해 처리가 가능하다. 맵리듀스는 분산 환경에서의 병렬 데이터 처리 기법이자 프로그래밍 모델로써, 입력 데이터를 정제하는 Mapper와 정제된 값을 전달받아 통합하는 Reducer로 구성된다[5-6].

그러나, 하둡 환경에서 데이터를 분석하기 위해 맵리듀스 알고리즘을 설계 및 구현하는 것은 기존 개발자들에게도 매우 복잡하고 어려운 일이기 때문에, 프로그래밍에 능숙하지 않은 데이터 과학자에게는 더욱 큰 부담이 된다. Hive[16]는 이러한 하둡의 사용 방안을 개선하기 위해 개발된 프로그램으로, HDFS를 저장 구조로 하여 RDBMS와 유사하게 데이터를 테이블 형태로 파악할 수 있도록 한다. 또한, 질의 및 분석 기능 역시 제공하는데, 이는 SQL-like한 언어인 HiveQL로 사용이 가능하다. Hive에서는 내부적으로 사용자가 작성한 HiveQL을 맵리듀스 알고리즘으로 변환하여 실행하기 때문에, 데이터 과학자는 이를 통해 보다 쉽게 데이터를 처리하고 분석할 수 있다.

로그 데이터는 해당 시간의 시스템 상태를 수치적으로 표현하는 반정형 데이터(semi-structured data)로, 보통 크기는 작지만 빠른 속도로 생성되기 때문에 빅데이터로 분류할 수 있다. 로그 데이터를 수집하는 방법은 여러 가지가 있으나, 여러 대의 서버로 구성된 시스템을 각각 모니터링하여 필요한 자원이나 상태의 로그 데이터를 특정 시간 단위로 수집하는 방법이 가장 많이 사용된다. 이 같은 방법을 사용하는 대표적인 시스템으로는 Ganglia 모니터링 시스템[25]이 있으며, 본 연구에 사용된 로그 데이터 역시 Ganglia 모니터링 시스템을 통해 수집되었다. 이와 같은 로그는 특정 시간마다 시스템 상태를 저장하고 있으므로, 시스템이 정상적으로 운용 중이거나 비정상적인 상태가 발생하는 경우 모두 로그 데이터로 남겨진다. 로그 데이터는 그 특성상 데이터 자체에서 의미 있는 결과를 찾기에는 어려움이 있으며, 분석 없이 로그 데이터를 누적하여 저장하면 데이터 공간의 낭비를 초래할 수 있다. 따라서, 로그 데이터를 활용하기 위해서는 적절한 처리와 분석을 통해 패턴이나 특징, 의미 등을 찾아내기 위한 추가적인 기법들이 필요하다.

본 논문에서는 로그 데이터 활용을 위해 이상 탐지 기능을 구현한다. 이상 탐지에 사용되는 기법이나 알고리즘들은 여러 가지가 있으나, 본 연구에서는 이동 평균[19-22]과 3-시그마[23, 24]를 사용한다. 이동 평균은 시계열 데이터로부터 각 시간에 대해 일정 기간의 평균을 계산하여 변화를 추적하는 기술이다[19]. 이러한 이동 평균은 데이터로부터 불규칙한 변동을 제거하여, 데이터의 추세를 파악하기 위한 지표로 사용된다. 3-시그마는 정규분포 곡선으로부터 평균에서 ±3 표준편차 이내에 대부분의 값(99.7%)이 포함된다는 경험적인 규칙을 의미한다[23]. 본 논문은 이러한 이동 평균과 3-시그마 기술을 사용하여, 각 시간의 이동 평균으로부터 3-시그마를 벗어나는 값을 이상 값으로 판단한다.

3. 이동 평균 및 3-시그마 기반 로그 데이터 이상 탐지

본 절에서는 이동 평균 및 3-시그마 기반의 로그 데이터 이상 탐지 구조를 설계한다. Fig. 1은 이상 탐지 구조를 간단히 도식화한 것이다. 먼저 Ganglia 모니터링 시스템은 하둡의 각 서버로부터 로그 데이터를 수집하여 Hive에 전달한

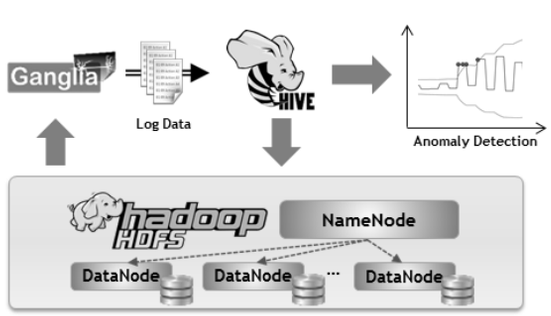

Fig. 1. Architecture of anomaly detection

다. Hive는 미리 정의한 테이블에 로그 데이터를 저장, 즉 HDFS에 로그 데이터를 저장한다. 마지막으로 Hive는 이상 탐지 기술을 사용하여 저장된 로그 데이터의 이상 탐지를 수행한다. 다음은 로그 데이터의 저장 및 이상 탐지 기술을 상세히 설계한다.

3.1 로그 데이터 저장 구조 모델링

Fig. 2는 Ganglia 모니터링 시스템을 통해 수집된 로그 데이터의 일부이다. Ganglia는 여러 대의 서버를 클러스터 형태로 묶고, 다수의 클러스터로부터 로그 데이터를 수집한다. Fig. 2에서 각 행의 첫 번째 항목 'etc'는 로그 데이터가 수집된 해당 클러스터를 의미하며, 두 번째 항목 'agent1'은 해당 서버의 호스트 명을 의미한다. 세 번째 항목 'bytes_out'은 수집된 로그 데이터의 종류이며 네트워크를 통해 출력된 바이트 크기를 의미한다. 마지막으로 네 번째 항목은 측정된 값을, 다섯 번째 항목은 측정된 시간을 각각 의미한다.

```
etc,agent1,bytes_out,1403.47,20150114000001
etc,agent1,bytes_out,1403.47,20150114000101
etc,agent1,bytes_out,1126.54,20150114000201
etc,agent1,bytes_out,1126.54,20150114000301
etc,agent1,bytes_out,1126.54,20150114000401
etc,agent1,bytes_out,1126.54,20150114000501
etc,agent1,bytes_out,1126.54,20150114000601
```

Fig. 2. Log data collected from Ganglia

이렇게 Ganglia를 통해 수집된 로그 데이터는 CSV(comma separated value) 형태의 텍스트 파일로 제공되므로, 이를 Hive에서 다루기 위해서는 테이블 구조를 설계하여야 한다. Fig. 3은 로그 데이터를 저장하기 위한 Hive의 테이블을 설계한 것이다. 테이블 구조는 Ganglia를 통해 수집된 CSV 형태의 로그 데이터와 같은 구조를 사용하였으며, 총 다섯 개의 항목으로 구성되어 있다.

Log_table

CLUSTER_NAME (STRING)	HOST_NAME (STRING)	METRIC_KEY (STRING)	METRIC_VALUE (STRING)	TIME_STAMP (STRING)

Fig. 3. Hive table scheme for storing log data

3.2 단순 이상 탐지와 가중치 적용 이상 탐지

앞서 언급한 바와 같이, 본 논문에서는 이상 탐지를 위해 이동 평균 및 3-시그마를 활용하며, 이는 특정 윈도우(window)의 이동 평균으로부터 ±3 표준편차 이외의 값을 이상치로 판단하는 방식이다. Fig. 4는 이와 같은 이상 탐지에 필요한 특정 윈도우의 이동 평균(SimpleMovingAverage)과 이동 표준편차(SimpleMovingStandardDeviation)를 계산하는 알고리즘이다. 알고리즘의 입력은 이전 단계에서 계산한 윈도우, 이상 탐지를 수행할 값, 그리고 윈도우의 크기이며, 출력은 각각 이동 평균 및 이동 표준편차이다. 먼저 두 알고리즘 모두 공통적으로 라인 1-5에서 이전 단계의 윈도우로부터 가장 오래된 값을 제거한 후, 이상 탐지를 수행할 값을 저장한다. 이동 평균은 라인 6-10에 의해 윈도우에 속한 모든 값의 합을 계산한 뒤, 윈도우의 크기로 나눔으로써 이동 평균을 계산한다. 그리고 이동 표준편차는 라인 6-13에서 윈도우에 속한 모든 값의 합과 제곱의 합을 계산하여 제곱의 평균과 평균의 제곱에 대한 차를 통해 분산을 구하고, 이에 제곱근을 취해 이동 표준편차를 계산한다.

```
Procedure SimpleMovingAverage()
Input:
  W: a window of previous step.
  V: a current positioned value.
  N: the window size.
Output:
  M: moving average.
1.  begin
2.    if (W equals N) then
3.      pop(W);
4.    end-if
5.    push(W, V);
6.    accum := 0;
7.    for each value v of W do
8.      accum := accum + v;
9.    end-for
10.   M := accum / N;
11. end
```

```
Procedure SimpleMovingStandardDeviation()
Input:
  W: a window of previous step.
  V: a current positioned value.
  N: the window size.
Output:
  SD: moving standard deviation.
1.  begin
2.    if (W equals N) then
3.      pop(W);
4.    end-if
5.    push(W, V);
6.    accum := 0;
7.    square_accum := 0;
8.    for each value v of W do
9.      accum := accum + v;
10.     square_accum := square_accum + square(v);
11.   end-for
12.   variance = (square_accum / N) - square(accum / N);
13.   SD := square_root(variance);
14. end
```

Fig. 4. Simple algorithms of computing moving average and moving standard deviation

Fig. 4와 같이 단순한 이상 탐지는 윈도우에 속한 모든 값이 동등하게 취급된다. 이는 곧 가장 오래된 값과 가장 최근 값이 같은 가치를 가진다는 것을 의미한다. 그러나, 일반적으로 로그 데이터는 시간적으로 가까운 데이터에 더 의존적인 경향이 있다. 즉, 이상 탐지할 기준 시간에 발생한 로그 데이터와 오래된 로그 간의 차이보다는 기준 로그와 최근 로그 간의 차이가 더 평가할 가치가 높다고 할 수 있다.

이를 해결하기 위해, 이동 평균 및 이동 표준편차에 가중치를 부여하는 방안을 사용한다. 본 논문에서는 가중치를 선형으로 부여하였는데, 윈도우에서 가장 오래된 값의 가중치는 1, 다음 값의 가중치는 2, 그리고 종내에는 이상 탐지를 수행할 값의 가중치에 윈도우의 크기와 같은 가중치를 부여한다. 이러한 과정은 Fig. 5의 가중치를 적용한 이동 평균(WeightedMovingAverage)과 이동 표준 편차(Weighted MovingStandardDeviation)의 알고리즘으로 나타내었다. 가중치 적용 이동 평균은 라인 10에서 가중치를 적용하고, 라인 12, 13에서 모든 가중치의 합에 대한 평균을 구한다. 그리고 가중치 적용 이동 표준편차는 라인 11, 12에서 가중치를 적용하고, 라인 14-16에서 가중치의 합에 대한 표준 편차를 구한다.

이러한 이동 평균 및 이동 표준편차의 알고리즘을 Hive에서 사용하기 위해 Hive의 사용자 정의 함수(UDF, user defined function)로 구현하였다. 사용자 정의 함수는 Hive Java API에서 제공하는 UDF 클래스를 기반으로 구현하며, 이는 HiveQL에서 불러와 SQL 함수 형태로 사용할 수 있다. 본 논문에서는 HiveQL에서 이동 평균을 MOVING_AVG 함수로, 이동 표준편차를 MOVING_STD 함수로 사용하였다.

```
Procedure WeightedMovingAverage()
Input:
  W: a window of previous step.
  V: a current positioned value.
  N: the window size.
Output:
  M: moving average.
1.  begin
2.    if (W equals N) then
3.      pop(W);
4.    end-if
5.    push(W, V);
6.    accum := 0;
7.    weight := 0;
8.    for each value v of W do
9.      weight := weight + 1;
10.     accum := accum + v * weight;
11.   end-for
12.   weight_sum := weight * (weight + 1) / 2;
13.   M := accum / weight_sum;
14. end
```

```
Procedure WeightedMovingStandardDeviation()
Input:
  W: a window of previous step.
  V: a current positioned value.
  N: the window size.
Output:
  SD: moving standard deviation.
1.  begin
2.    if (W equals N) then
3.      pop(W);
4.    end-if
5.    push(W, V);
6.    accum := 0;
7.    square_accum := 0;
8.    weight := 0;
9.    for each value v of W do
10.     weight := weight + 1;
11.     accum := accum + v * weight;
12.     square_accum := square_accum + square(v * weight);
13.   end-for
14.   weight_sum := weight * (weight + 1) / 2;
15.   variance := (square_accum / weight_sum) - square(accum / weight_sum);
16.   SD := square_root(variance);
17. end
```

Fig. 5. Advanced algorithms of computing weighted moving average and weighted moving standard deviation

3.3 이상 탐지 질의 설계

본 절에서는 Hive에 저장된 데이터로부터 이상 탐지를 수행하기 위해 이상 탐지 질의를 설계하는 방안을 설명한다. Fig. 6은 이동 평균 및 3-시그마 기반 이상 탐지를 위한 질의이다. 먼저, 라인 6-10의 내부 질의는 로그 데이터가 저장되어 있는 테이블 'log_table'로부터 이동 평균 및 이동 표준편차를 계산하는 과정이다. 여기서 MOVING_AVG 함수와 MOVING_STD 함수는 각각 특정 윈도우의 이동 평균 및 이동 표준편차를 계산하기 위한 UDF로, 윈도우의 크기(본 논문에서는 60분을 사용)를 매개 변수로 전달하여 계산할 수 있다.

```
SELECT tt.time_stamp, tt.metric_value,
CASE WHEN tt.metric_value IS NULL THEN 'TRUE'
     WHEN tt.metric_value NOT BETWEEN tt.moving_avg - 3*tt.moving_std
          AND tt.moving_avg + 3*tt.moving_std THEN 'TRUE'
     ELSE 'FALSE' END is_anomal
FROM (
     SELECT t.time_stamp, t.metric_value,
     ROUND(MOVING_AVG(t.metric_value, 60), 3) as moving_avg,
     ROUND(MOVING_STD(t.metric_value, 60), 3) as moving_std
     FROM log_table as t
    ORDER BY time_stamp) tt;
```

Fig. 6. Anomaly detection query based on moving average and 3-sigma

다음으로, 라인 1-4의 외부 질의는 계산된 이동 평균 및 이동 표준편차를 사용하여 이상 탐지를 수행하는 과정이다. 라인 2는 해당 로그 데이터 값이 NULL일 때 이상치로 판단하여 TRUE를 반환하며, 라인 4는 이상치로 판단되지 않은 항목에 대해서 FALSE를 반환한다. 라인 3은 이동 평균 및 3-시그마 기법을 통해 이상 탐지를 수행하는 핵심 과정으로, 해당 로그 데이터의 값이 평균으로부터 ±3 표준편차를 벗어나는 항목은 이상치로 판단하여 TRUE를 반환한다.

4. 실험 및 평가

본 절에서는 제3절에서 설계한 이상 탐지 기술을 실제 실험을 통해 그 유효성을 평가한다. 실험은 한 대의 네임노드 서버와 네 대의 데이터노드 서버를 사용하였다. 하드웨어 플랫폼으로 네임노드는 Intel Xeon CPU E3-1240 v3 3.40GHz, 16GB RAM이고, 데이터노드는 Intel Core i3-4350 CPU 3.50GHz, 4GB RAM이다. 소프트웨어 플랫폼으로는 Hadoop-2.6.0을 사용하였으며, 구현 언어는 HiveQL (Hive-1.1.0 버전)을 사용하였다. 실험에 사용된 로그 데이터는 'etc' 클러스터의 'agent1' 서버에서 하루 동안 수집된 'bytes_out' 로그 데이터(1,440개)이다.

4.1 단순 이상 탐지

Fig. 7은 단순 이동 평균 및 3-시그마 기반 이상 탐지 결과를 차트로 시각화한 것이다. Fig. 7에서 실선은 로그 데이터 값을 의미하며, 0시부터 24시까지 1,400 전후로 급격한 변화를 보이고 있다. 이러한 로그 데이터로부터 제3.2절의 이상 탐지 질의를 수행한 결과, 점선의 형태로 이동 평균 및 3-시그마의 밴드(band)가 형성됨을 알 수 있다. 여기서 로그 데이터가 밴드를 벗어나는 경우를 이상치로 판단하는데, 실험에서는 크게 네 구간에서 나타났으며, 이를 세로 막대로 표시하였다.

이러한 이상 탐지 결과를 통해, 이동 평균 및 3-시그마 기술을 다음과 같이 평가할 수 있다. 먼저, 로그 데이터의

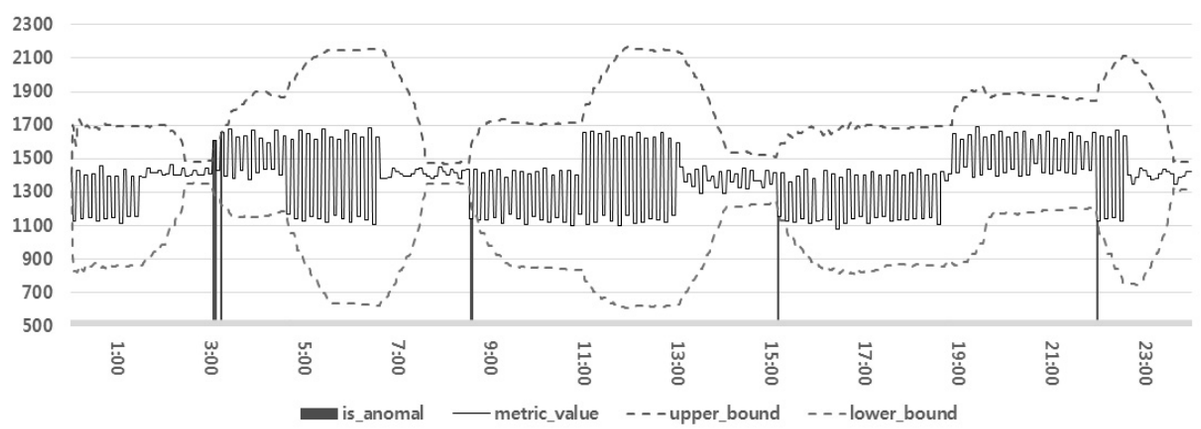

Fig. 7. Visualization of simple anomaly detection results

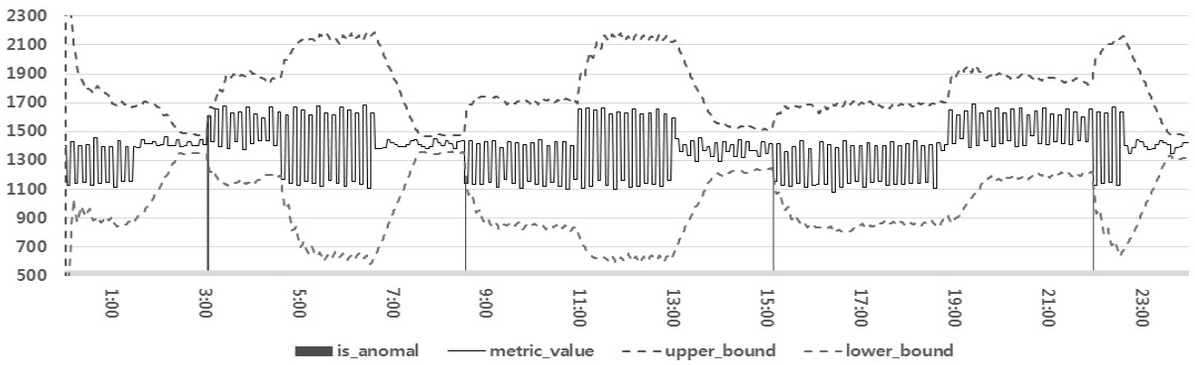

Fig. 8. Visualization of weighted anomaly detection results

변화율이 급격히 증가하는 경우를 올바르게 이상으로 판단한다. 즉, 15시 이전의 로그 데이터의 변화율은 크지 않으나, 15시 이후의 변화율이 급격히 증가함에 따라 대략 15시의 로그 데이터를 이상으로 판단한다. 또한 11시부터 13시까지의 구간에서 로그 데이터의 변화가 크지만, 이 구간에서 변화율은 크지 않으므로 이상으로 판단하지 않는다. 하지만 이러한 이상 탐지 기술은 급격한 변화율이 발생하는 시점에서 불필요한 이상 값도 추가적으로 나타나는 문제점이 있다. 예를 들어, 대략 3시의 이상 탐지 결과에서 7개의 이상 값이 나타났으며, 대략 9시에는 5개의 이상 값이 나타났다. 이는 단순 이상 탐지 기법의 한계라 할 수 있다.

4.2 가중치 적용 이상 탐지

Fig. 8은 가중치를 적용한 이동 평균 및 3-시그마 기반 이상 탐지 결과를 차트로 시각화한 것이다. 이는 제4.1절과 같은 로그 데이터를 실험에 사용하였으며 제3.2절의 이상 탐지 질의를 수행한 결과, Fig. 7과 유사한 점선의 형태로 이동 평균 및 3-시그마의 밴드가 형성되었다. 또한 로그 데이터가 밴드를 벗어나는 이상치를 세로 막대로 표시하였으며, Fig. 7과 같이 네 구간에서 이상치가 탐지되었다.

그러나, 이상 탐지의 결과가 이전 실험 결과인 Fig. 7과는 차이를 보인다. 그림을 보면, 이상치가 나타난 구간은 같으나, 구간 내에서 발생한 이상치의 수가 확연히 줄어든 것을 알 수 있다. 예를 들어, Fig. 6에서는 대략 3시의 이상 구간에서 7개의 이상치가, 대략 9시의 이상 구간에서 5개의 이상치가 발견되었다. 하지만 Fig. 7에서는 대략 3시의 이상 구간에서 2개의 이상치가, 대략 9시의 이상 구간에서 2개의 이상치가 발견되었다. 이러한 결과는 곧 가중치를 적용한 이상 탐지가 그렇지 않은 이상 탐지에 비해 이상치를 정확하게 찾아낼 뿐만 아니라, 단순 이상 탐지에 비해 불필요한 이상치를 탐지하는 빈도 역시 낮아진다는 것을 알 수 있다.

5. 결론 및 향후 연구

본 논문에서는 로그 데이터의 이상 값을 탐지하기 위해, Hive를 기반으로 하는 로그 데이터 저장 구조를 설계하였다. 또한, HiveQL을 사용하여 이동 평균 및 3-시그마 기반 이상 탐지 기술을 설계 및 구현하고, 실험 결과를 시각화하여 이상 탐지 기술에 대한 적정성을 논의하였다. 그리고, 단순 이상 탐지의 성능 향상을 위해, 로그 데이터에 가중치를 부여하여 더 정확한 이상 탐지를 수행하도록 하였다. 실험 결과, 로그 데이터의 변화율이 급격하게 증가하는 부분을 이상으로 올바르게 탐지함을 보였으며, 가중치 기법을 통해 이상 탐지의 정확성 역시 향상된 것을 알 수 있다. 본 논문의 향후 연구로 다수의 서버에서 수집되는 로그 데이터에서도 이상치를 탐지할 수 있도록 이상 탐지 기술을 개선하고, 속도 향상을 위해 Hive의 저장 구조를 재설계하고자 한다.

References

[1] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. Byers, “Big Data: The Next Frontier for Innovation, Competition, and Productivity,” Technical Report, McKinsey Global Institute, 2011.

[2] T. Rabl, M. Sadoghi, H.-A. Jacobsen, S. Gómez-Villamor, V. Muntés-Mulero, and S. Mankowskii, “Solving Big Data Challenges for Enterprise Application Performance Management,” in *Proc. of the VLDB Endowment*, Vol.5, No. 12, pp.1724-1735, Aug., 2012.

[3] M. Saecker and V. Markl, “Big Data Analytics on Modern Hardware Architectures: A Technology Survey,” *Springer Lecture Notes in Business Information Processing*, Vol.138, pp.125-149, 2013.

[4] Hadoop [Internet], http://hadoop.apache.org/.

[5] C. Lam and J. warren, “Hadoop in Action,” Manning Publications, 2010.

[6] T. White, “Hadoop: The Definitive Guide,” O'Reilly Media, Yahoo! Press, June, 2009.

[7] HDFS [Internet], http://hadoop.apache.org/hdfs/.

[8] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, “The Hadoop Distributed File System,” in *Proc. of the 26th IEEE Symp. on Mass Storage Systems and Technologies(MSST)*, Lake Tahoe, Nevada, pp.1-10, May, 2010.

[9] Dhruba Borthakur, “The Hadoop Distributed File System: Architecture and Design,” Technical Report, pp.1-14, 2007, http://hadoop.apache.org/core.
[10] J. Dean and S. Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters,” *Communications of the ACM*, Vol.51, No.1, pp.107-113, Jan., 2008.
[11] J. Dean and S. Ghemawat, “MapReduce: a Flexible Data Processing Tool,” *Communications of the ACM*, Vol.54, No.1, pp.72-77, Jan., 2010.
[12] S. Lee, J. Kim, Y.-S. Moon, and W.-K. Loh, “Iceberg Cube Parallel Computation using MapReduce,” *Korea Computer Congress*, Vol.37, No.1(A), pp.25-26, June, 2010.
[13] H. Lee, M. Kim, H. Lee, and H. Yoon, “Design and Implementation of an Analysis module based on MapReduce for Large-scalable Social Data,” *Korea Computer Congress*, Vol.38, No.1(B), pp.357-360, June, 2011.
[14] G. Kim, G. Nam, and U. Kim, “Analysis and Statistics of Domestic Dam Based on MapReduce,” *Korean Society for Internet Information*, pp.131-132, Nov., 2013.
[15] D.-S. Choi, G.-J. Mun, Y.-M. Kim, and B.-N. Noh, “An Analysis of Large-Scale Security Log using MapReduce,” *Korean Institute of Information Technology*, Vol.9, No.8, pp. 125-132, Aug., 2011.
[16] Hive [Internet], https://hive.apache.org/.
[17] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Authony, H. Liu, P. Wyckoff, and R. Murthy, “Hive: a Warehousing Solution over a Map-Reduce Framework,” in *Proc. of the VLDB Endowment*, Vol.2, Issue 2, Aug., 2009.
[18] J. S. Sarma, N. Jain, Z. Shao, P. Chakka, N. Zhang, S. Antony, H. Liu, and R. Murthy, “Hive - a petabyte scale data warehouse using Hadoop,” in *Proc. of the 26th IEEE International Conference on Data Engineering*, pp.996-1005, Mar., 2010.
[19] Y.-S. Moon and J. Kim, “Efficient Moving Average Transform-Based Subsequence Matching Algorithms in Time-Series Databases,” *Information Sciences*, Vol.177, No. 23, pp.5415-5431, Dec., 2007.
[20] J. M. Lucas and M. S. Saccucci, “Exponentially Weighted Moving Average Control Schemes: Properties and Enhancements,” *Technometircs*, Vol.32, Issue 1, 1990.
[21] J. S. Hunter, “The exponentially Weighted Moving Average,” *Journal of Quality Technology*, Vol.18, No.4, Oct., 1986.
[22] William W. S. Wei, “Time Series Analysis Univariate And Multivariate Methods,” Addison-Wesley, 2005.
[23] F. Pukelsheim, “The three sigma rule,” *The American Statistician*, Vol.48, Issue 2, pp.88-91, 1994.
[24] H.-P. Kriegel, P. Kroger, E. Schubert, A. Zimek, “LoOP: local outlier probabilities,” in *Proc. of the 18th ACM Conference on Information and Knowledge Management*, pp.1649-1652, Nov., 2009.
[25] Ganglia Monitoring System [Internet], http://ganglia.info/.

손 시 운

e-mail : ssw5176@kangwon.ac.kr
2014년 강원대학교 컴퓨터과학과(학사)
2014년~현 재 강원대학교 컴퓨터과학과 석사과정
관심분야 : 데이터마이닝, 빅데이터, 하둡 에코시스템

길 명 선

e-mail : gils@kangwon.ac.kr
2007년 강원대학교 컴퓨터과학과(학사)
2009년 강원대학교 컴퓨터과학과(석사)
2009년~2012년 강원대학교 중앙정보전산원
2012년~현 재 강원대학교 컴퓨터과학과 박사과정
관심분야 : 데이터마이닝, 시계열 분석, 빅데이터 분석, 하둡 에코시스템

문 양 세

e-mail : ysmoon@kangwon.ac.kr
1991년 한국과학기술원 전산학과(학사)
1993년 한국과학기술원 전산학과(석사)
2001년 한국과학기술원 전자전산학과 전산학전공(박사)
1993년~1997년 현대전자산업(주) 주임연구원
2001년~2002년 (주)현대시스콤 선임연구원
2002년~2005년 (주)인프라밸리 기술위원(이사)
2005년~2008년 한국과학기술원 첨단정보기술연구센터 연구원
2008년~2009년 미국 퍼듀대학교 방문연구원
2012년~2013년 강원대학교 기획부처장
2014년~2016년 강원대학교 IT대학 부학장
2005년~현 재 강원대학교 컴퓨터과학과 교수
관심분야 : 데이터마이닝, 스트림데이터, 저장 시스템, 데이터베이스 응용, 빅데이터 분석, 프라이버시 보호 마이닝

원 희 선

e-mail : hswon@etri.re.kr
1990년 연세대학교 전산과학과(학사)
1992년 한국과학기술원 전산학과(석사)
1992년~1999년 KBS 기술연구소 연구원
2000년~현 재 한국전자통신연구원 책임연구원
관심분야 : 빅데이터, 멀티테넌트 플랫폼, 프라이버시 보호 분석, 멀티테넌트 하둡