빅데이터 보안이벤트 처리를 위한 NoSQL 기반 분산 처리 시스템

한효준*, 강지원*, 정용환**, 김양우*
*동국대학교 정보통신공학과
**한국과학기술정보연구원
e-mail : han6343@dongguk.edu

NoSQL-based Distributed Processing System for Processing BigData Security Events

HyoJoon Han*, JiWon Kang*, Yong-Hwan Jung**, Yangwoo Kim*
*Dept. of Information Communication Engineering, Dongguk University
**Korea Institute of Science and Technology Information

요 약

인터넷과 클라우드 서비스 사용이 증가하면서 패킷의 양과 사이버 위협이 증가하였다. 본 논문에서는 빅데이터를 처리하기 위해 사용되는 NoSQL 을 보안이벤트의 신속한 처리를 위한 침입탐지 시스템에 적용하였다. 다양한 데이터 모델 유형의 NoSQL 데이터베이스 중에서 빅데이터 보안이벤트를 처리하는데 가장 적합한 시스템을 찾기 위해 세 가지 유형의 Snort 룰 기반 보안이벤트 분산 처리 프로토타입 시스템들을 구축하였고 각 시스템의 성능을 평가하였다. 그 결과로 MongoDB 기반의 보안이벤트 분산 처리 시스템이 가장 속도가 빠른 것을 확인하였다.

1. 서론

인터넷과 클라우드 서비스 사용이 증가하면서 이에 따른 패킷의 양도 급속도로 많아졌고 사이버 위협 또한 증가하였다. 빅데이터를 빠르게 처리하기 위해서 NoSQL 을 사용하는데[1], 이는 빅데이터화 되는 보안이벤트를 신속하게 처리하기 위한 침입탐지시스템에 적용 할 수 있다. NoSQL 데이터베이스는 다양한 유형의 데이터 모델이 존재하기 때문에, 보안이벤트의 특성에 따라 알맞은 데이터베이스를 선택하여 사이버 위협으로부터 빠르게 대처해야한다. 본 논문에서는 보안이벤트의 신속한 처리에 가장 알맞은 NoSQL 데이터베이스를 찾기 위해 Snort 룰 기반의 보안이벤트 처리 프로토타입 시스템을 설계, 구축하고 각 시스템의 성능을 비교 평가한다.

본 논문은 다음과 같이 구성된다. 2 장에서는 NoSQL 과 Snort 에 대해 살펴본다. 3 장에서는 Snort 룰 기반의 NoSQL 을 사용한 보안이벤트 처리 프로토타입 시스템을 설계하고 구축한다. 4 장에서는 구축한 프로토타입 시스템의 성능을 비교 평가한다. 마지막으로 5 장에서는 성능 평가 결과를 가지고 결론을 도출한다.

2. 관련연구

2.1 NoSQL

관계형데이터베이스의 데이터 저장 구조는 저장된 전체 데이터베이스에 접근하기에는 처리 속도가 늦고 관리가 어려웠다. 또한 빅데이터를 저장하기에 읽기/쓰기에서 관계형데이터베이스의 수평적 확장과 데이터 저장 구조로 인한 제약이 있었다. 관계형데이터베이스의 문제를 해결하기 위해 새로 고안된 기술이 NoSQL(Not only SQL) 이다. NoSQL 은 테이블 스키마(Table Schema)가 고정되지 않고, 테이블 간 'JOIN' 연산을 지원하지 않으며, 수평적 확장이 용이하다는 특징을 가진다. NoSQL 은 데이터 모델에 따라서 크게 세 가지로 구분되며, 각 데이터 모델의 구조와 특징은 다음 표와 같다.

<표 1> NoSQL 데이터 모델 특징

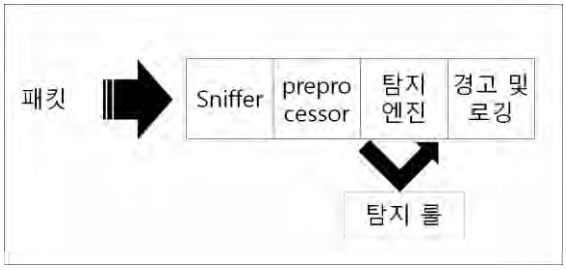
데이터 모델	설명
<키, 값> 저장 구조	· 가장 간단한 데이터 모델 · 범위 질의는 사용이 어려움
열 기반 저장 구조	· 연관된 데이터 위주로 읽는 데 유리한 구조 · 범위 질의에 유리
문서 저장 구조	· 스키마가 없고 확장성이 높다. · 레코드 간의 관계 설명이 가능 · 개념적으로 관계형데이터베이스와 유사

2.2 Snort

Snort[2]는 'Sniffer and More'라는 용어에서 유래되었는데, 초창기에는 단순한 패킷 Sniffer 프로그램에서 1999 년 룰 기반의 분석 기능이 추가되고 개발자 커뮤니티를 통한 계속 적인 기능 보완과 향상 그리고 'Sourcefire'에 인수됨에 따라 사실상 침입

탐지시스템의 기술 표준이 되었다. Snort 의 기능은 간단히 다음과 같이 정의할 수 있다.

- 패킷 Sniffer : 네트워크의 패킷을 읽어 보여주는 기능
- 패킷 Logger : 모니터링한 패킷을 저장하고 로그에 남기는 기능
- 네트워크 침입탐지시스템 : 네트워크 트래픽을 분석하여 공격을 탐지하는 기능



(그림 1) Snort 의 구조

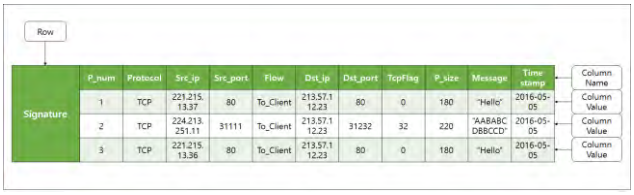
3. 본론

NoSQL 의 가장 큰 장점이라고 하면 수평적 확장을 통한 분산처리 시스템이라고 볼 수 있을 것이다. 본 논문에서는 NoSQL 데이터베이스 데이터모델의 대표적인 제품들을 이용해 빅데이터 보안이벤트를 분산처리하는 프로토타입 시스템을 설계하고 구축하였다.

3.1 Cassandra

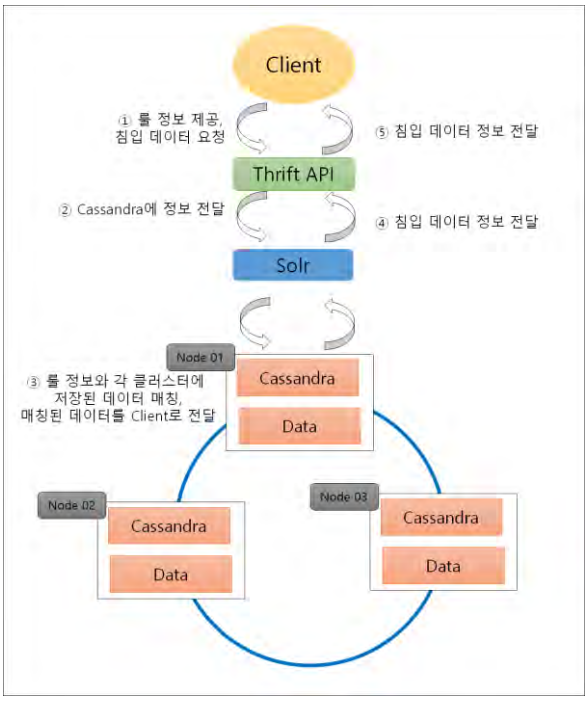
Cassandra[3][4]는 <키, 값> 저장 구조의 대표적인 제품으로 데이터 분산과 가용성 면에서 뛰어난 제품이다. Cassandra 는 일관적인 해싱(hashing)을 사용하여 데이터를 분산한다. Cassandra 는 데이터 복제 시 발생하는 오류에 대해 복구성이 좋고 쉽게 노드를 추가, 삭제 할 수 있는 장점이 있다. 따라서, Cassandra 를 이용하여 분산 처리 침입탐지시스템 구축 시, 침입탐지시스템이 감당해야 할 보안이벤트의 양에 따라 노드의 개수를 조정하며 최적화된 환경에서 처리할 수 있고, Cassandra 의 고가용성으로 시스템 고장 없이 운용할 수 있다.

Cassandra 는 <키, 값> 저장 구조이기 때문에 한 키에 하나의 값이 들어가는 방식이다. 다음은 Snort 의 룰에 맞게 변환한 보안이벤트 더미데이터 구조이다.



(그림 2) Cassandra 에서의 Snort 보안이벤트 구조

(그림 2)와 같은 구조의 더미데이터를 생성하여 다음과 같은 처리방식에 따라 보안이벤트를 처리하였다.

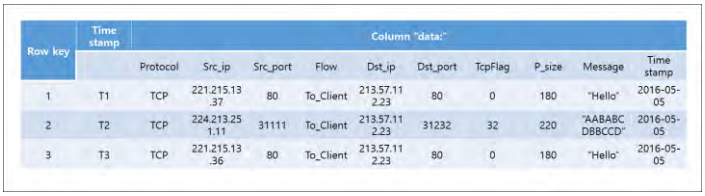


(그림 3) Cassandra 보안이벤트 분산처리 구조와 순서

3.2 HBase

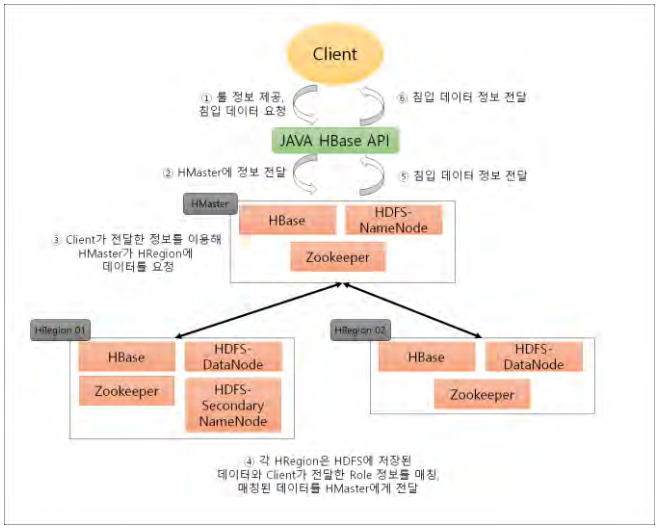
HBase[3][5]는 열 기반 저장 구조의 대표적인 제품으로 하둡(Hadoop)의 HDFS 와 Map-Reduce 를 통해 분산처리 환경을 구축할 수 있다. HBase 는 비정형, 반정형의 대규모 데이터를 비용과 시간을 절감하며 처리하고 분석할 수 있다. 시스템을 중단하지 않고 장비를 쉽게 추가하거나 삭제할 수 있고, 일부 장비에 장애가 발생하더라도 전체 시스템에 큰 영향을 주지 않는다. 따라서, HBase 분산처리 시스템을 사용하면 고가용성이 보장되는 환경에서 안정적인 침입탐지시스템을 구축할 수 있다. 또한 데이터의 복제본을 저장하기 때문에 서버에 장애가 발생해 데이터 손실이 생겨도, 본제본을 통한 데이터 복구가 가능하여 문제없이 침입을 탐지할 수 있다.

HBase 는 열 기반 저장 구조이기 때문에 연관된 데이터 위주로 읽는데 유리하다. 다음은 Snort 의 룰에 맞게 변환한 보안이벤트 더미데이터 구조이다.



(그림 4) HBase 에서의 Snort 보안이벤트 구조

(그림 4)와 같은 구조의 더미데이터를 생성하여 다음과 같은 처리방식에 따라 보안이벤트를 처리한다.



(그림 5) HBase 보안이벤트 분산처리 구조와 순서

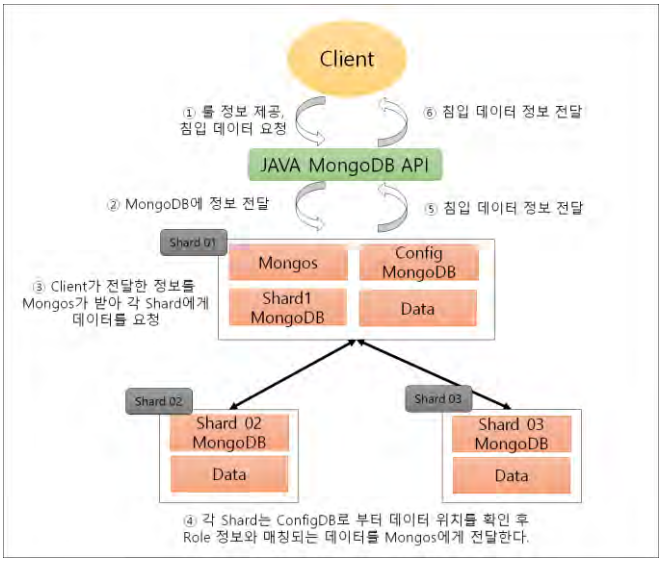
3.3 MongoDB

MongoDB[3][6]는 문서 기반 저장 구조의 대표적인 제품으로 데이터를 분할하고 서로 다른 서버에 나누어 저장하는 샤딩 기능을 이용해 분산처리 시스템을 구축할 수 있다. 자동-샤딩 기능을 통해 분산되지 않은 환경에서 작업하다가 용량이 부족하거나 데이터의 분할 저장, 처리가 필요한 경우 바로 분산환경을 구축할 수 있다. 시스템의 변화에 적극적으로 대응이 가능하다는 장점이 있으며, 마스터에서 슬레이브로 데이터를 비동기 방식으로 복제하기 때문에 마스터의 성능 저하가 거의 없어서 슬레이브를 추가해도 서비스 성능이 떨어지지 않는다.

MongoDB 는 문서 기반 저장 구조이기 때문에 레코드 간의 관계 설명이 가능하다. 다음은 Snort 의 룰에 맞게 변환한 보안이벤트 더미데이터 구조이다.

(그림 6) MongoDB 에서의 Snort 보안이벤트 구조

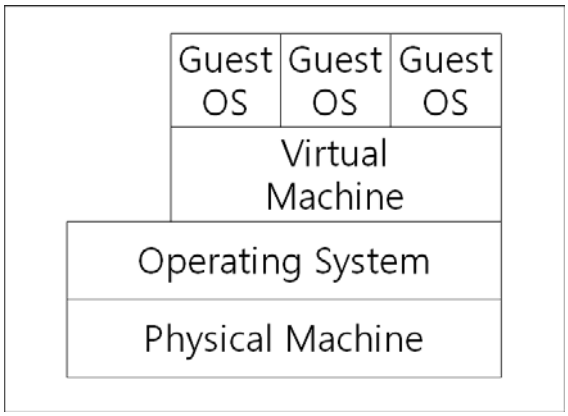
(그림 6)과 같은 구조의 더미데이터를 생성하여 다음과 같은 처리방식에 따라 보안이벤트를 처리한다.



(그림 7) MongoDB 보안이벤트 분산처리 구조와 순서

4. 실험 및 결과

각 NoSQL 제품 별로 분산처리 환경을 구축하기 위해서 시스템 당 3 대의 가상머신을 이용하여 프로토타입 시스템을 구축하였다.



(그림 8) 분산처리 프로토타입 시스템 구축 환경

각 노드의 사양은 다음 표와 같다.

<표 2> 분산처리 시스템 가상 노드 사양

항목	사양
CPU	I3-4170(1 Core)
Memory	4GB
HDD	200GB
OS	CentOS 7 x64

분산처리 환경은 위 표와 같은 사양의 가상 노드가 총 3 대로 구성 되어 있다. 실험은 각 제품의 데이터 저장 구조에 맞게 변환된 보안이벤트 더미데이터를 생성하고 데이터의 개수를 늘려가면서 수행 속도를 측정하였다. 각 데이터베이스에 저장된 데이터의 10 분의 1 은 침입 데이터이다. 각 데이터베이스는 각자의 추출, 검색 기능을 이용하여 데이터베이스에서 침

입 데이터를 추출, 검색 한다.
다음 표는 각 데이터베이스에서 실행한 데이터 추출 속도 측정 표이다.

<표 3> Cassandra 분산 처리 성능 표

총 데이터 수 (Packet)	추출 데이터 수 (Packet)	수행시간(Sec)
10,000,000	1,000,000	29.576
20,000,000	2,000,000	58.247
100,000,000	10,000,000	394.619

<표 4> HBase 분산 처리 성능 표

총 데이터 수 (Packet)	추출 데이터 수 (Packet)	수행시간(Sec)
10,000,000	1,000,000	21.287
20,000,000	2,000,000	34.148
100,000,000	10,000,000	236.254

<표 5> MongoDB 분산 처리 성능 표

총 데이터 수 (Packet)	추출 데이터 수 (Packet)	수행시간(Sec)
10,000,000	1,000,000	5.623
20,000,000	2,000,000	9.774
100,000,000	10,000,000	122.377

5. 결론 및 향후연구

실험 결과 세 가지 유형의 분산처리 프로토타입 시스템 중에서 MongoDB 를 사용한 분산처리 프로토타입 시스템의 성능이 가장 우수하다는 것을 확인하였다. 하지만 MongoDB 의 성능이 데이터 수에 비해 아주 빠르지 않음을 볼 수 있는데, 그 이유는 보안이벤트의 특성상 'Full Text Search'가 아니라 'Partial Word Search' 으로 진행됨에 따라 MongoDB 가 인덱싱 (Indexing) 처리를 할 수 없기 때문이다.
향후 연구에서는 MongoDB 를 사용한 침입탐지시스템과 기존에 사용하던 MySQL 를 사용한 침입탐지시스템과의 성능 비교를 통해 MongoDB 를 사용한 침입탐지시스템의 현장적용 가능성을 검토해 볼 것이다.

Acknowledgement

본 연구는 한국과학기술정보연구원(KISTI)의 위탁연구 과제로 수행한 것입니다.

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 대학 ICT 연구센터육성지원사업의 연구결과로 수행되었음 (IITP-2016-R2720-16-0004)

참고문헌

[1] Jing Han, Haihong E, Guan Le. "Survey on NoSQL Database", IEEE Xplore 2011.
[2] Martin Roesch, "Snort – Lightweight Intrusion Detection for Networks", USENIX 1999.
[3] Changlin He, "Survey on NoSQL Database Technology", Journal of Applied Science and Engineering Innovation 2015.
[4] Cassandra, http://cassandra.apache.org/
[5] HBase, https://hbase.apache.org
[6] MongoDB, https://www.mongodb.com/