JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 19, No. 11 : 2629~2636 Nov. 2015

빅데이터 관리를 위한 문서형 DB 기반 로그관리 시스템 설계

류창주[1] · 한명호[2] · 한승조[3*]

Design of Log Management System based on Document Database for Big Data Management

Chang-ju Ryu[1] · Myeong-ho Han[2] · Seung-jo Han[3*]

[1]Department of Information and Communication Engineering, Chosun University, Gwangju 501-759, Korea
[2]Department of Information and Communication Engineering, Chosun University, Gwangju 501-759, Korea
[3*]Department of Information and Communication Engineering, Chosun University, Gwangju 501-759, Korea

요 약

최근 IT 분야에서 빅데이터 관리에 대한 관심이 급증하고 있으며, 빅데이터의 실시간 처리 문제를 해결하기 위해 많은 연구가 진행되고 있다. 네트워크상에서 주고받는 데이터를 실시간으로 저장하는 기능으로 인해 리소스가 많이 필요한 반면, 높은 비용적 측면 때문에 분석 시스템 도입에 문제가 야기 되고 있으며 이러한 문제점 해결을 위해 저비용 고효율성을 만족하는 시스템 재설계의 필요성이 증가되고 있다. 본 논문에서는 빅 데이터 관리를 위한 문서형 DB 기반 로그관리 시스템을 설계하기 위해서 문서형 데이터베이스인 MongoDB를 사용하였으며, 제안하는 로그관리 시스템을 통해 고효율의 로그 수집 및 처리와 위,변조에 안전한 로그 데이터 저장을 확인한다.

ABSTRACT

Recently Big Data management have a rapid increases interest in IT field, much research conducting to solve a problem of real-time processing to Big Data. Lots of resources are required for the ability to store data in real-time over the network but there is the problem of introducing an analyzing system due to aspect of high cost. Need of redesign of the system for low cost and high efficiency had been increasing to solve the problem. In this paper, the document type of database, MongoDB, is used for design a log management system based a document type of database, that is good at big data managing. The suggested log management system is more efficient than other method on log collection and processing, and it is strong on data forgery through the performance evaluation

키워드 : 로그관리, 데이터베이스, MongoDB, Log4j

Key word : Log management, Database, MongoDB, Log4j

Received 10 August 2015, **Revised** 06 September 2015, **Accepted** 21 September 2015

*** Corresponding Author** Seung-Jo Han(E-mail:rcjlove@naver.com, Tel:+82-62-230-7069)
Department of Information and Communication Engineering, Chosun University, Gwangju 501-759, Korea

Open Access http://dx.doi.org/10.6109/jkiice.2015.19.11.2629 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

최근 IT 분야에서 다양한 정보 채널과 스마트 기기의 발달로 인해 정보의 생산, 유통, 보유가 기하급수적으로 증가하고 있으며, 이러한 데이터를 관리하는 빅데이터 관리에 대한 관심이 급증하고 있다.

특히 소셜 네크워크 서비스 성장으로 인한 데이터량의 급증과 이를 처리하는 속도, 플랫폼확장, 프레임 구축 등에 대한 어려움이 대두되고 있으며, 멀티미디어 서비스 성장으로 고용량 데이터가 증가함에 따라 데이터의 저장과 관리, 분석에 대한 한계가 들어나고 있다.

구글은 2003년 GFS(Google File System), 2006년 BigTable 등을 발표하였으며, 특히 GFS는 저 사양의 하드웨어 장비들을 연결하여 재구성한 분산 네트워크 파일 시스템으로써 클라우드 기술 시장을 형성하였다. 또한 BigTable은 대용량의 데이터를 저장하고 이를 분석하는 기술로 빅데이터 산업의 핵심 기술로 대두되고 있다[1].

이러한 빅데이터를 사용하는 시대에 사용하는 정보의 양과 더불어 오류도 증가하고 있으며, 오류에 대한 해결책을 해결하는데 많은 시간이 소모된다. 따라서 모니터링 또는 통계를 위한 시스템의 요구가 증가되고 있으며, 대응 시간 단축을 위한 관련 업무 프로세서가 동작하면서 발생하는 Log 데이터를 기반으로 하여 기존 시스템 절차에 영향을 주지 않으면서 정보를 수집 할 수 있는 시스템의 필요성이 대두 되고 있다.

실시간 처리 문제를 해결하기 위해 기존의 방법은 Oracle의 Exadata Database Machine, IBM의 Netezza등의 Big Data Appliance 또는 많은 디스크로 구성이 된 SAN(Storage Area Network) 장비가 필요하다. 또한 네트워크에서 주고받는 데이터를 실시간으로 저장하는 기능이 필요하기 때문에 많은 리소스가 필요하다. 하지만 높은 비용적 측면 때문에 분석 시스템 도입에 문제점이 발생하였다[2].

이러한 여러 문제점을 해결하기 위해 저비용 고효율성을 만족하는 시스템 재설계의 필요성이 증가되고 있으며, 기존의 프로그램 설계를 크게 해치지 않으며 로그 수집 및 통계 그리고 실시간 분석이 가능한 시스템 구축이 필요하다.

본 논문에서는 유연한 데이터 스키마 구조가 가능하며 저비용 구축을 위해 오픈소스 제품 중에서 문서형 데이터베이스인 MongoDB를 적용하였다. 시스템 H/A 구성의 유연성을 만족하고 구축에 소용 되는 비용의 효율성이 높으며 운영 기술의 진입 장벽 또한 비교적 낮은 자바스크립트 언어를 기반으로 하고 있다. 또한 기존 로그 관리 구조를 많이 해치지 않는 Log4j를 지원하고, 기존의 버그 추적이나 잘못된 거래 데이터 분석을 적용하기 위한 관리 시스템을 제안한다.

Ⅱ. 본 론

2.1. 로그 관리 개요

2.1.1. 로그 관리 목적

로그정보 수집의 목적은 개발자와 품질관리자, 시스템 운영자가 발견한 소프트웨어의 버그 또는 공격자, 부정사용자를 포함하는 비정상 사용자를 추적하는 것을 돕기 위한 소프트웨어이다. 즉 운영 중에 부정 사용자나 공격자 같은 데이터, 개발 과정에서 발견된 소프트웨어 버그의 상태변화 과정을 추적하기 위한 소프트웨어이다[3].

2.1.2. 로그 관리의 필요성

컴퓨터의 모든 일련의 작업은 로그라는 정보로 저장되게 되며 컴퓨터 관련 범죄가 일어났을 경우 흔적을 찾기 위해 관리자는 시스템에 저장되어 있는 로그를 분석한다[4]. 따라서 이러한 공격기법을 로그분석을 통해 발견 할 수 있다면 시스템의 취약점 제거 및, 공격에 대한 근원을 찾는 기반이 될 수 있다. 보안 관리를 위해서는 시스템에서 발생할 수 있는 취약점 점검 및 제거와 같은 예방활동도 중요하지만 로그를 관리하고 주기적으로 분석하는 작업 또한 중요하다.

2.1.3. 로그 관리의 기술요소

로그 관리의 기술의 요소는 그림1과 같이 크게 4가지로 나누어진다. 수집, 저장, 검색, 분석으로 이루어져 있다. 시스템내의 로그들을 취합하는 작업을 수집이라 하며, 수집한 것을 관리하는 형태를 저장 또는 보관이며, 저장된 데이터를 기반으로 주기적 검사를 검색 또는 추적이라 하며, 이러한 모든 데이터를 기반으로 보고형태의 데이터로 만드는 과정을 분석 또는 모니터링 이라한다[5].

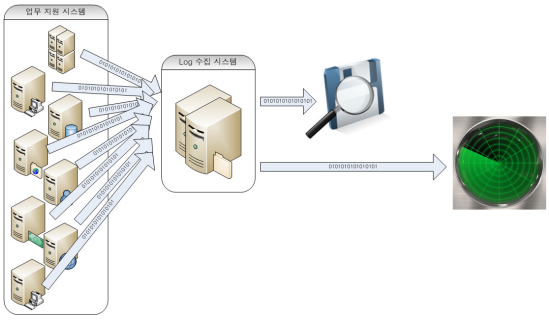

Fig. 1 Log Management four elements

2.2. 로그 관리 기술 동향

2.2.1. 국내 로그 관리 기술

국내 로그 관리 기술은 점진적으로 해외 클라우드 시스템들을 도입하고 있으며, 클라우드 관련 시장의 요구가 증가 되면서 관련 기술들이 개발되고 있으며, 여러 컨설팅 업체 및 제품 개발사에 의해 관련 기술의 보급과 산업 활성화가 진행되고 있다. 지난 H캐피탈과 N금융사의 전산망 장애로 금융권 로그관리·감독 강화의 필요성이 대두되고 있으며 금융권 및 기업시장에 통합로그관리 도입이 필수적인 상황이다.

Table. 1 Domestic log management status

product Nome	manufacturer	Describe
Anymon PLUS	U-net System	support parallel connections, data compression
Logsaver v2.0	TSN-tech	DVD-R recording support for tamper
Logcenter	innerbus	support compression Analysis

현재 많은 로그 관리 기술이 생겨나고 있으며, 개인정보를 포함한 내부 기밀정보가 담긴 시스템에서 발생하는 각종 로그 정보를 분석하여 다양한 내부 정보 유출에 대한 사전·사후 감사를 가능하게 해주는 진보된 형태의 통합로그관리 시스템 모델들이 개발되고 있다.

2.2.2. 국외 로그 관리 기술

서비스 서버상의 로그 데이터가 위·변조될 경우 정확한 정보를 얻을 수 없는 취약성에 대응하기 위한 관련 연구가 진행되고 있으며, 다양한 제품들이 생산되고 있다. 반영구적 보존, 위·변조가 불가능한 매체 실시간 저장, 데이터의 무결성, 가용성, 책임 추적성을 강화하여 내·외부자에 의한 불법적 접근시도 및 악의적인 침입을 분석하여 차단 기능의 위험을 감지하고, 보안 사고의 사전 모니터링이 가능하다. 또한 보안사고 및 불법 거래 발생 시 명확한 원인 규명 및 책임소재를 입증하는 증거 자료로 활용하기 위해 사용한다.

이와 관련하여 여러 저장 방법 및 가용성 연구 그리고 효율적 추적 기능을 위한 다양한 방법들이 연구되었으며, 특히 아마존의 EC2 기반의 기술이나 마이크로소프트사의 Windows Azure 등의 클라우드 기반 시스템, NoSQL 기반 시스템으로 전환 중에 있다.

지금 이 순간에도 많은 No SQL 기술이 생겨나고 있으며, 각각의 기술력 결합으로 인한 시너지를 이루려는 시도가 많이 있다. 현재 PaaS(Platform as a Service) 형태나 IaaS(Infrastructure as a Service) 형태의 서비스로도 양산되고 있다. 표 2는 PaaS 기업을 나타낸다.

Table. 2 PaaS corporation

service	manufacturer	Describe
AppEngine	Google	based on python or Django solution
Force.com	SalesForce	SalesForce of SaaS infra, based on Apex language solution
Bungee Connect	Bungee Labs	support of Java language, based on MVC Eclipse IDE support solution
Long Junp	Relational Network	support of Java language, based on MVC Eclipse IDE support solution
Wave Maker	VMware	Web browser-based develop -ment tools

2.3. 문서형 DB 기반 로그관리

2.3.1. 문서형 데이터베이스

문서형 데이터베이스는 테이블과 같은 경직된 구조에 데이터를 저장하지 않고 문서에 데이터를 저장한다. 관계형 데이터베이스 시스템 테이블에서는 열을 새로 추가하려면 테이블 자체의 정의를 변경해야 하며 null 값을 갖게 될지라도 열이 기존의 모든 레코드

에 추가된다.

하지만 문서를 사용하는 경우 기타 모든 문서를 변경하지 않고 개별 문서에 속성을 새로 추가할 수 있다. 문서형 데이터베이스에서는 “관련” 데이터를 별도의 스토리지 영역에 저장하는 대신 이를 문서 자체에 삽입한다. 각 참조는 추가 쿼리를 필요로 하기 때문에 관련 데이터가 저장된 또 다른 문서에 참조를 저장하는 것보다 처리 속도가 빠르며 데이터가 상위 문서 안에서 독립적으로 존재하는 것이 적합한 많은 애플리케이션에서 효율적으로 동작한다[5].

2.3.2. MongoDB

MongoDB는 강력하고 유연하며 확장성이 높은 데이터 저장소이다. JSON의 2진 버전인 BSON을 사용하여, 키 값 쌍으로 데이터를 유지하는 JSON 형태의 문서에 데이터를 저장하는 방식으로 MongoDB가 다른 문서 데이터베이스와 구별되는 한 가지 기능은 SQL문을 MongoDB 쿼리 함수 호출로 매우 간단하게 변환하는 기능이다. 이 기능을 이용하여 한 조직에서 현재 사용 중인 관계형 데이터베이스를 쉽게 마이그레이션 할 수 있다. 관련 데이터를 별도로 저장해야 하는 경우 MongoDB에서 별도의 컬렉션을 사용하여 쉽게 작업을 수행할 수 있다. 범위쿼리, 보조색인, 정렬 기능 같은 관계형 데이터베이스의 유용한 기능들과 함께 분산 확장 기능과 내장된 맵 리듀스 방식의 집계 연산이나 공간 정보 색인 같은 다양한 기능을 제공한다[6].

또한 MongoDB는 자체적으로 맵 리듀스를 지원하고 있으며 별도의 분산·병렬 컴퓨팅을 위한 플랫폼 없이 MongoDB에 저장된 데이터를 분산·병렬 처리로 빠르게 분석 할 수 있다. 하지만, 다른 컬렉션에서 데이터를 참조하는 대신에 문서 내부에 데이터를 삽입하면 성능을 대폭 개선할 수 있으므로 각각의 개별 유스 케이스를 대상으로 취해야 하는 접근 방식을 신중하게 결정해야 한다. 주요 운영 체제와 프로그래밍 언어에서 사용 가능한 2진 파일과 드라이버를 사용하여 매우 간단하게 설치하고 사용할 수 있다. MongoDB는 색인, 저장 자바스크립트, 집계, 고정 크기 컬렉션, 파일 저장소와 같은 다양한 기능을 가지고 있으며 표 3은 MongoDB의 기능을 나타낸다[7].

Table. 3 Function of MongoDB

property of MongoDB	Describe
index	unique index, composite index, spatial information index
Java script	save to server Java script functionand value
Aggregation	various aggregation tool map reduce
collection	support fixed size collection
storage	protocol provides that can save files
management	convenient database management

2.3.3. Log4j

Log for Java란 뜻으로 프로그래머가 로그문의 출력을 다양한 대상으로 할 수 있도록 도와주는 도구로써 오픈소스 운영 재단 중에 하나인 Apache Software Foundation에 있는 Java 언어 개발환경에서 로그 저장 편의성 및 테스트 품질 향상을 위해 작성된 Logging Framework이다. Log4j는 어플리케이션에 문제가 생겼을 경우 로깅을 활성화하면 문제의 위치를 찾을 수 있으므로 도움이 되고, 어플리케이션의 실행코드를 수정하지 않고 런타임에 로깅의 활성화를 진행 할 수 있으며 로깅의 속도는 중요하게 다루어지는 문제이다[8].

Log4j는 로그 그 자체를 의미하는 Logger와 로그를 출력하는 것을 정의하는 Appender, 출력의 형태를 결정하는 Layout으로 구성되어 있으며, 상부에서 출력 형태의 제어나 출력의 순위 제어가 중앙 집중 형태로 이루어지고 Logger의 계층적 구조를 사용하여 로그 기록을 사용자 정의된 형태로 출력 가능하다[9].

Ⅲ. 제안하는 문서형 데이터베이스 저장 메커니즘 설계

기존 Text File 시스템 기반 로그 저장 방법의 위·변조 문제를 해결하기 위해 문서형 데이터베이스 기반 로그관리 시스템을 제안한다.

3.1. Log4j 구성 설계

Appender의 저장 단위 결정은 사용자 접근성과 권한 설정 및 서버의 재배치를 고려해야 한다. Layout의 저장 단위 결정은 데이터의 저장 단위인 Row와 조회 성능 및 조건의 효율성에 의해 결정된다.

3.1.1. Pattern Layout 설계

Pattern Layout의 설계는 MongoDB 저장 형태의 결정 요인으로, 데이터의 조회 조건에 영향을 준다. 저장 단위의 분리를 고려하여 Database에는 각각의 서버들을 맵핑 하였으며, 제안하는 Pattern Layout은 차후 부하의 증가로 인한 문제, 통신 접근 제한, 권한 할당 문제를 쉽게 해결할 수 있도록 설계하였다. 또한 Collection의 단위는 사용자 또는 개발자의 혼란을 최소하기 현재 운영되고 있는 시스템의 Log Appender Name을 그대로 사용하였으며, 필드의 구조는 기본 MongoDB Pattern Layout을 통해 선언된 구조와 동일하게 설계하였다. 표 4는 MongoDB와 객체의 맵핑을 나타낸다.

Table. 4 Mapping to MongoDB and object

MongoDB	Object Definitions
Database	server name mapping
Collection	Log file name or Appender Name mapping
Document/field	Tree form to record a log message, include system PK to _id, Time Stamp

그림 2는 MongoDB Collectoin 필드 구조를 표현한 것으로 서버명과 Database를 동일하게 하였으며, Collection은 Log Name의 구조를 동일하게 하였다. 필드 중에 class는 전체 자바 클래스 명을 지칭하는 fully Qualified Class Name이 선언되어 있으며, 이외 별도로 자바 패키지명만을 배열형으로 구분하여 저장 하였다.

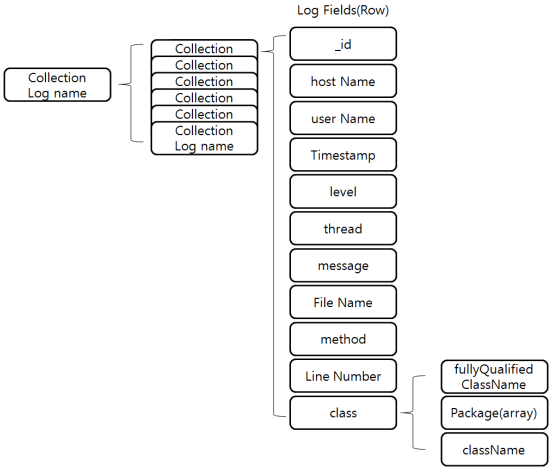

Fig. 2 MongoDB Collection field structure

배열형은 인덱스를 가지고 있는 구조이며, 해당 순서는 패키지 Depth 레벨과 동일하다. Class Name은 이벤트가 발생한 순수 패키지명을 제외한 클래스 명만을 가지고 있는 필드이다. 그림 3은 제안하는 Pattern Layout을 나타낸다.

```
protected String getRunHostname() {
    if (RunHostname == null) {
        try {
            InetAddress addr = InetAddress.getLocalHost();
            this.RunHostname = addr.getHostName();
        } catch (UnknownHostException e) {
            this.RunHostname = "localhost";
        }
    }
    return RunHostname;
}
protected String getRunUsername() {
    return System.getProperty("user.name");
}
private DBObject runinfo2bsonLoggingEvent(LoggingEvent loggingEvent) {
    DBObject oResult = null;
    if (loggingEvent != null) {
        oResult = new BasicDBObject();

        oResult.put("hostName", this.getRunHostname());
        oResult.put("userName", this.getRunUsername());
        oResult.put("timestamp", new Date(loggingEvent.getTimeStamp()));
        nullCheckerPut(oResult, "level", loggingEvent.getLevel().toString());
        nullCheckerPut(oResult, "thread", loggingEvent.getThreadName());
        nullCheckerPut(oResult, "message", loggingEvent.getMessage());
        addLocationInformation(oResult, loggingEvent.getLocationInformation());
        addThrowableInformation(oResult, loggingEvent.getThrowableInformation());
    }
    return oResult;
}
private void addLocationInformation(DBObject bson, LocationInfo locationInfo) {
    if (locationInfo != null) {
        nullCheckerPut(bson, "fileName", locationInfo.getFileName());
        nullCheckerPut(bson, "method", locationInfo.getMethodName());
        nullCheckerPut(bson, "lineNumber", locationInfo.getLineNumber());
        nullCheckerPut(bson,
        "class", class2bsonClassName(locationInfo.getClassName()));
    }
}
private DBObject message2bsonThrowable(Throwable throwable) {
    DBObject oResult = null;
    if (throwable != null) {
        oResult = new BasicDBObject();
        nullCheckerPut(oResult, "message", throwable.getMessage());
        nullCheckerPut(oResult,
        "stackTrace", stack2bsonStackTrace(throwable.getStackTrace()));
    }
    return oResult;
}
```

Fig. 3 Proposed Pattern Layout

3.2 MongoDB Collection 구성 설계

MongoDB 스키마 구조 설계를 하기 전에 MongoDB를 이루는 구조적 특징을 우선 알아보고자 한다. 표 5는 RDBMS와 MongoDB의 객체를 비교한 것으로 MongoDB를 이루고 있는 구성들을 나열하였다.

Table. 5 RDBMS and MongoDB's object

RDBMS	MongoDB
Database	Database
Table	Collection
Row	Document
Column	Field
Index	Index
table joins	Embedded documents and linking
Primary key	_id field(automatically)
collection	Framework support

MongoDB 에서의 Database는 Collection의 집합이며, 사용자 저장 접근의 최초 단위이다. 저장 단위의 큰 집합의 구분이기 때문에 Server 단위별 저장 단위를 결정하게 된다. Collection은 DBMS의 객체와 비교하였을 때 table 역할과 유사하다. 그림 4는 MongoDB Collection 구성을 나타낸다.


```
{ "_id" : ObjectId("50b2f6c7de9Bc4e6a6d1165d"),
"timestamp" : ISODate("2015-6-26T04:57:43.642Z"),
"level" : "DEBUG",
"thread" : "server.startup : 0",
"message" : "Looking for matching resources in jar file
[file:/WebSphere/installedApps/PAY/PAY.ear/UPAY.war
/WEB-INF/lib/egovframework.rte.fdl.cmmn-1.0.0.jar]",
"fileName" : "PathMatchingResourcePatternResolver.java",
"method" : "doFindPathMatchingJarResources",
"lineNumber" : "457",
"class": { "fullyQualifiedClassName" :
"org.springframework.core.io.support.PathMatchingResourcePatternResolver",
"package" : [ "org", "springframework", "core", "io", "support" ],
"className" : "PathMatchingResourcePatternResolver" } }
```


Fig. 4 MongoDB Collection configuration

기록 형태의 단위로 유사한 정보의 집합의 성격을 지니고 있기 때문에, Appender의 단위로 사용하였으며 정보 조회의 최소 단위는 DBMS의 Row와 유사한 필드 문서형 형태로 되어 있다. 이는 Tree 구조 형태로 확장이 가능하다는 장점이 있으며 저장의 특수성을 고려하여 차후에 데이터 추가를 용이하게 하기 위해 DBMS의 정규형을 거치지 않고 단일된 형태이다. MongoDB 에서의 필드의 특성을 충분히 고려하여, 확장성 있는 형태로 연속적인 구조변경이 가능하다.

Ⅳ. 성능분석 및 고찰

4.1. 검색

기존 로그 저장 방법은 Text File 시스템을 기반으로 하고 있다. Unix 시스템 기반의 환경이기 때문에 Text File 검색에 용이한 여러 명령어들을 기본으로 제공하고 있다. 검색 절차는 대상 파일명을 추출하고, 파일명들 중 해당 Keyword의 포함 여부를 확인한다. 해당 파일명, 데이터 위치인 LineNumber를 추출하며 대상 시각도 연, 원, 일 까지만 구분 할 수 있다. 반면 본 논문에서 제안하는 문서형 DB 기반 로그 관리 시스템은 MongoDB를 사용하여 기존 로그 저장 방식보다 구체적인 시간 정보까지 표현이 가능하다.

```
> query = { "timestamp" : { "$gte" : ISODate('2015-6-26T04:57:43.642Z') } }
> db.log.find(query);
```

Fig. 5 Proposed log management system's searching log

그림 5는 제안하는 로그 관리 시스템에서 시간 조건으로 로그를 검색한 결과이다. 위의 내용은 2015년 6월 26일 오후 04시 57분 43.642초 보다 큰 시간의 log Collection을 조회를 한다는 조건이다. 특정 시간대의 검색 기능은 기존 Text 파일 관리에서는 구현하기 어려운 조건이기도 하다. 이러한 검색 질의 이점이 매우 높기 때문에 검색 효율성이 매우 높다고 할 수 있다.

4.2. 저장

Log 데이터의 저장에 있어 위·변조 문제는 충분하게 고려되어야 한다. 하지만 기존 Text File 시스템은 단순 Text Edit에서 작업이 가능하기 때문에 위·변조의 위험성이 매우 높다고 할 수 있다. 제안하는 문서형 DB기반 로그 관리 시스템은 BSON 형태의 MongoDB를 사용하고 있으며, 일부 데이터 구조는 Hex Edit에서 볼 수 있으나, 저장된 필드의 데이터를 조작하는 것은 불가능하기 때문에 기존의 위·변조 문제를 해결할 수 있다. 그림 6은 Text 저장 방법 Log 데이터의 Hex Edit를 나타내며 그림 7은 MongoDB 저장 방법 Log 데이터의 Hex Edit를 나타낸다.

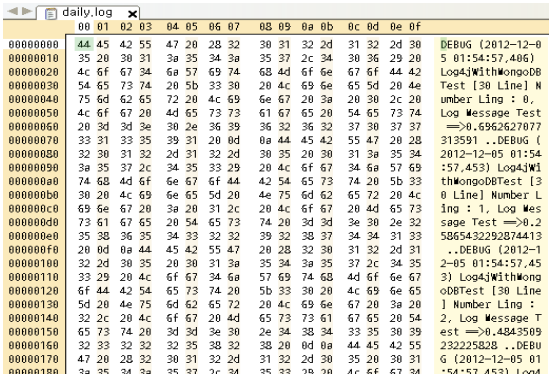

Fig. 6 Hex Edit log data Text storage process

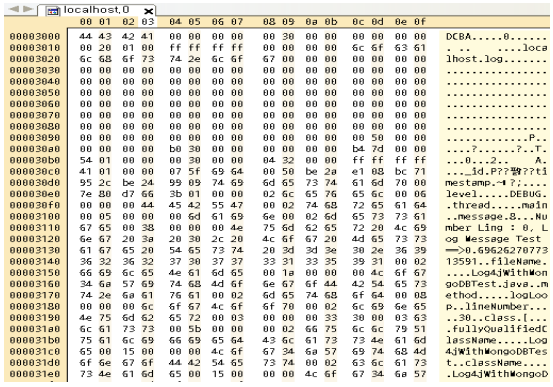

Fig. 7 Hex Edit log data MongoDB storage process

그림 6의 Text 데이터는 모든 데이터가 그대로 노출된 것을 알 수 있다. 그러나 그림 7의 MongoDB 데이터를 보면 중간 중간에 알 수 없는 문자로 표현된 데이터가 함께 섞여있는 것을 볼 수 있다. 이것은 MongoDB 만의 로그 데이터를 저장하는 정보이다. Text 저장 관리 시스템의 로그 정보는 공개되어 있어서 내용 변조가 가능하다. 그러나 MongoDB의 로그 정보를 수정하는 것은 데이터의 파괴를 일으킬 수 있기 때문에, Text 방식에 비해 보안성이 우수하다.

V. 결 론

데이터를 수집 한 후 분석하는 일련의 작업은 경영자의 결정권 시기를 지연 시키는 가장 큰 요인이다. 로그 수집을 단순한 보안 측면에서의 활용뿐만 아니라, 경영자의 결정에 도움이 되는 지표 산출로 사용되어야 하며, 실시간으로 대응할 수 있는 시스템으로의 확장이 필요하다. 이로 인하여 시스템 성능 확장과 효율적인 부하분산이 필요하게 되었다. 성능분석을 통해 MongoDB를 이용한 Log 관리 시스템은 향상된 보안성을 지님과 동시에 편리한 수평적 확장과 부하분산 시스템을 쉽게 구축할 수 있음을 확인하였다. 또한 기존의 여러 시스템에 나누어져 있는 Log 파일 분석에 있어서 많은 어려움이 있었으나, 제안하는 문서형 DB 기반 로그 관리 시스템을 통해 큰 비용 없이 시스템을 확장할 수 있고 기존 Text 기반 로그 관리 시스템보다 높은 효율성을 확인할 수 있었으며, Log 정보의 다양한 활용성에 대한 가능성도 알 수 있게 되었다.

ACKNOWLEDGMENTS

This work was supported by the funding of Chosun University intermural scientific research in 2015.

REFERENCES

[1] Yeong-hwan Yong, “Advent of an era of big data”, *Korea the Journal of information technology*, 2012.

[2] Pete Warden, “*Big Data Glossary*”, 2011.9.

[3] Hye-won Cho, Youg-koo Han, Young-koo Lee, “A Non-fixed Log Area Management Technique in Block for Flash Memory DBMS”, *Journal of KIISE* vol 5, 2010.10.

[4] Wan-jib Kim, Heung youl Youn, “Compliance with the integrated management for compliance in a log of System”, *Journal of KIISC*, vol 5, 2010.10.

[5] Kristina Chodorow, "*Scaling MongoDB*", 2011.3.

[6] Michael Dirolf, Kristina Chodorow, “*MongoDB The Definitive Guide*", O'Reilly Media, 2010.

[7] Chang-ju Ryu, Seung-jo Han, "The Design of Log Management System for MongoDB", *ICT conference*, vol2, 2013.6.

[8] Je-young Ryu, “Design and Implementation of Log Library for Embedded Software Development”, *KIISE conference*, 2011.11.

[9] Beom-gyun Choi, “*Jakarta Project*", 2004.3.

류창주(Chang-ju Ryu)

2012년 조선대학교 정보통신공학과(학사)
2014년 조선대학교 정보통신공학과(공학 석사)
현재 조선대학교 정보통신공학과 박사과정
※관심분야 : 네트워크 보안, 빅데이터 관리, 정보보호

한명호(Myeong-Ho Han)

2013년 조선대학교 IT공학과(공학 석사)
현재 조선대학교 정보통신공학과 박사과정
※관심분야 : 네트워크 보안, 빅데이터 관리

한승조(Seung-jo Han)

1980년 조선대학교 전자공학과(학사)
1982년 조선대학교 전자공학과(공학 석사)
1994년 충북대학교 전자계산학과 (공학 박사)
1986년 6월 ~ 1987년 3월 뉴올리언즈대학 객원교수
1995년 2월 ~ 1996년 1월 텍사스대학 객원교수
2000년 12월 ~ 2002년 3월 버클리대학 객원교수
1998년 3월 ~ 현재 조선대학교 전자정보통신공학부 교수
※관심분야 : 통신보안시스템설계, S/W 불법복제 방지시스템, ASIC 설계