

# 클라우드 환경에서 데이터베이스 통합을 위한 DBaaS 허브 시스템에 관한 연구

정계동\*, 황치곤\*\*, 이종용\*, 신호영\*\*\*  
광운대학교 교양학부\*, 광운대학교 컴퓨터과학과\*\*, 경북대학교 IT 보안학과\*\*\*

## The Study of DBaaS Hub System for Integration of Database In the Cloud Environment

Kye-Dong Jung\*, Chi-Gon Hwang\*\*, Jong-Yong Lee\*, Hyo-Young Shin\*\*\*  
Div. of General Education Information Engineering, Kwangwoon University\*  
Dept. of Computer Science, Kwangwoon University\*\*  
Dept. of IT Security, Kyungbuk University\*\*\*

**요약** 클라우드 환경에서 기업은 의사결정이나 기업의 정책을 결정하기 위해 데이터 통합 및 분석이 필요하다. 그러나 이 환경에서 새로운 시스템이 추가되면 데이터 통합 측면에서 기존 시스템이 가진 이질적인 특성 때문에 많은 비용과 시간이 필요로 하게 된다.

따라서 본 논문에서 멀티 데이터베이스 서비스를 위하여 DBaaS 허브 시스템을 제안한다. 이러한 DBaaS는 서로 다른 데이터베이스를 요구할 수도 있으며, 연관성이 있는 서비스를 위하여 데이터를 통합해야 할 필요가 있다.

DBaaS의 데이터 상호운용성 문제를 해결하기 위해 온톨로지를 이용한 메타 쿼리를 제안한다. 메타 쿼리는 실제 데이터에 접근하기 위한 쿼리가 아닌 상위의 쿼리이다. 이는 실제 데이터베이스에 접근할 때 온톨로지를 통하여 변환된 질의로 접근함으로써 데이터 통합을 기할 수 있다. 또한 메타 데이터를 이용하여 문서 지향 데이터베이스 시스템을 구축한다.

**주제어** : 클라우드 환경, 멀티 데이터베이스, DBaaS, 온톨로지, 메타 쿼리

**Abstract** In the cloud environment, the company needs data integration and analysis to make decision and policy. If new system is added to this environment, a lot of time and cost is needed due to disparate properties among systems when data is integrated. Therefore, in this paper, we propose a DBaaS hub system for multi-database service. The DBaaS may require a different database and need data integration for relevant service. Using the ontology, we propose a metadata query to resolve the interoperability issues between data of DBaaS. The meta-query is not a query to access the real data, but the query for the upper level. This method provides data integration by accessing the data with the converted query through an ontology when we access the actual database. We also constructs a document-oriented database system using the metadata.

**Key Words** : Cloud Environment, Multi-Database, DBaaS, Ontology, Meta-Query

\* 본 논문은 2014년 광운대학교의 학술연구비에 의하여 지원되었음

Received 5 July 2014, Revised 9 August 2014

Accepted 20 September 2014

Corresponding Author: Chi-Gon Hwang(Dept. of Computer Science, Kwangwoon University)

Email: duck1052@kw.ac.kr

ISSN: 1738-1916

© The Society of Digital Policy & Management. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. 서론

IT 시장의 새로운 패러다임으로 급부상 하면서 클라우드와 빅데이터는 시대의 흐름과, 변화를 주도하고 있다. 현재 클라우드 인프라의 효과적인 도입 및 빅데이터의 효과적인 관리에 집중하고 있지만, 실질적으로 더 중요한 것은 이러한 자산을 효과적으로 활용하는 것이다. 앞으로 클라우드와 빅데이터는 기술적인 관점이 아니라 비즈니스 관점에서 고민해야 하는 것이다. 이는 IT 인프라 중요 이해 관계자들의 요구사항을 통해서 기획하고 실행해야 한다. 그리고 엄청난 양의 데이터가 빠른 속도로 생성됨에 따라 과거 테이블 기반의 관계형 데이터베이스의 관점에서 유연한 데이터 처리는 어렵고 복잡하다. 따라서 빅데이터는 새로운 형태의 방대한 데이터의 출현을 의미하며, 기업들은 빅데이터 분석을 통해 가시적으로 드러나지 않는 변화의 흐름을 파악하고 새로운 형태의 데이터 분석을 통해 과거에 불가능했던 새로움을 창조하는 시대를 일컫는다[1].

본 논문의 멀티 데이터베이스에서 요구한 데이터 분석 및 추출은 다음과 같이 3단계로 나뉜다.

- STEP 1. 데이터 추출 : 메타 쿼리 형태에서 IS-A 관계와 HAS-A 관계를 확장하여 추출하는 단계이다.
- STEP 2. 데이터 변형 : 메타 쿼리를 로컬 쿼리로 변형하여 쿼리를 실행하고, 실행 결과는 메타 쿼리의 표준형으로 변형 문서 지향 파일로 생성한다.
- STEP 3. 데이터 전송 : 요구한 클라우드에게 문서 지향 형태로 파일 전송한다.

따라서 클라우드 환경에서 지원하는 여러 데이터베이스는 다양한 이질적인 특성으로 인해 많은 비용과 시간이 필요하다. 또한 이러한 빅데이터를 저장하고 관리할 수 있는 인프라를 구축하여야한다[2][3].

본 논문에서 클라우드 환경의 멀티데이터베이스 통합을 위한 데이터 수집 결과는 몽고 DB 형태의 문서 지향 데이터 형태로 만든다.

이에 따른 논문의 구성은 2장에서 관련연구로 DBaaS와 빅데이터 가상화 처리를 기술하고, 3장에서 쿼리 변환 기법 및 문서 지향 데이터 생성 방법론의 제시하고 이를 평가한다. 4장에서 결론을 기술한다.

## 2. 관련연구

### 2.1 DBaaS

협업 환경에서 시스템을 개발할 때, 예상 부하량과 향후 시스템의 성장률을 예상해야한다. 이때 데이터베이스 종류와 방법론에 대해 결정하게 된다. 이러한 과정은 다양한 예측을 통해서 진행되며, 그 과정에서 발생하는 불확실한 변수는 효과적으로 대응하기 어렵다. 이러한 문제를 해결할 수 있는 강점을 가진 것이 DBaaS이다. DBaaS는 Agility(민첩성)을 제공하므로 빠르게 애플리케이션을 기획하고 애플리케이션의 상태에 따라 효과적으로 데이터베이스를 사용할 수 있다.

클라우드에서 DBaaS의 유형은 두 가지가 있다. 첫째 유형은 스키마 레벨에서의 통합이고, 둘째 유형은 데이터베이스 레벨에서 RAC(Real Application Clusters)로 통합하는 것이다.

스키마 레벨에서의 통합은 스키마 표준화 문제로 인해 구현의 어려움이 많으나, 구현이 되면 가장 효율적으로 사용할 수 있다[4][5]. 또한 DBaaS의 전통적인 데이터베이스 기능이 제공되므로 SQL을 사용할 수 있다. 클라우드 환경에서 멀티데이터베이스는 ACID속성을 유지하기 힘들기 때문에 기본 로컬 데이터베이스는 그대로 유지하여야 한다[6]. 따라서 본 논문에서의 DBaaS 허브시스템은 데이터 웨어하우스 기반으로 하는 대용량 데이터들에 대한 접근, 데이터의 분할, 복제 및 분산처리가 가능하도록 하기위해 No-SQL 인터페이스를 제공할 수 있다.

### 2.2 빅데이터와 가상화

빅데이터란 데이터 양이 급격히 증가함에 따라 관리상 문제점을 유발시켜 별도의 관리가 필요한 데이터를 말하며, RDBMS 데이터베이스 관리 도구를 가지고 다루기에는 힘든 데이터 세트 규모를 말한다[7].

빅데이터 기술 도입 시 고려사항 중 정보 관리는 애플리케이션의 관점 보다 데이터의 관점에서 시스템을 봐야 한다. 실제로 데이터는 비즈니스의 중심으로 진출했으며, 기업들은 빅데이터를 처리하기 위해 보다 데이터 중심적인 플랫폼의 관점에서 IT 프로젝트를 새롭게 평가하고 있다. 데이터의 위치 또는 포맷에 독립적인 데이터 중심 기업에서는 분산형 의사 결정 및 운영 지원을 위해 데이터 가상화와 MDM(Master Data Management)에 대한

요구도 부각되고 있다. 따라서 데이터 가상화는 일반적인 가상화와 같이 이종으로 구성된 인프라를 논리적으로 통합된 하나의 리소스인 것처럼 접근, 관리, 최적화할 수 있게 해주는 방법이다. 이를 통해 일부 서비스, 기능 또는 기타 리소스의 내부적 구현에서 외부 인터페이스를 추상화할 수 있다[8].

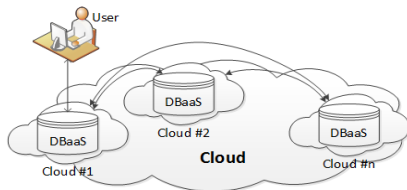
데이터 가상화[9]의 중심은 관계형, 하둡, NoSQL 등의 이질적 back-end 데이터 리포지토리에 대해 논리적으로 통합된 접근 및 기타 애플리케이션을 지원하는, SQL 가상화 접근 방법에 사용되는 것과 같은 추상화 계층이다. 물론 데이터 가상화는 경우에 따라 여러 지리적 위치와 클라우드 환경에 걸쳐 분산되어 있는 스토리지 및 서버 플랫폼과 같은 인프라 가상화의 다른 계층들에 의존할 수 있다[9][10][11].

따라서 빅데이터를 구축하기 위해 클라우드 컴퓨팅을 이용하는 것은 빅데이터 시스템 구성을 테스트하기 위한 방안이며, 가상화 상에서 빅데이터 시스템을 구현하여도 문제가 없는 시스템으로 설계하였을 경우에 바람직하다.

### 3. 클라우드 환경 DBaaS 허브 시스템

본 논문에서 제안하는 멀티데이터베이스 기반 클라우드 환경은 [Fig. 1]과 같다. DBaaS 허브시스템은 사용자가 선택한 애플리케이션을 실행하면서 필요로 하는 데이터베이스 리소스, 스토리지, 네트워크 장치를 포함한 인프라스트럭처에 대해 온 디맨드 방식으로 액세스를 제공한다.

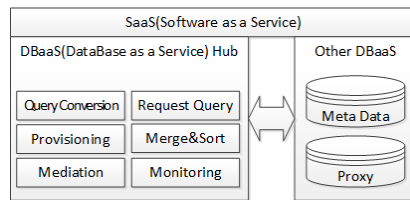
DBaaS 허브시스템은 데이터베이스 응용 프로그램의 가장 중요한 구성 요소이며, 가장 진보된 전문 지식을 동적 클라우드 환경에서 효과적으로 관리할 수 있고, 데이터베이스로서의 서비스(DBaaS) 솔루션과 PaaS를 솔루션에 묶여 DB의 가용성과 확장성을 보장 할 수 있다.



[Fig. 1] The proposed DBaaS hub system environment

### 3.1 DBaaS 시스템 허브 구성

최근 기업 시스템들은 데이터 공유와 협업을 위해 데이터 통합의 필요성을 제기하고 있다. 통합 방법은 서비스 지향 아키텍처와 서비스에 사용되는 데이터의 통합을 위한 마스터 데이터인 메타데이터 관리가 필요하다. 메타데이터는 데이터베이스 간의 매핑 정보 설정에 사용하고, 스키마 관리, 온톨로지 관리, 도메인 관리, 스크립트 관리로 구분된다. 스키마 관리는 어플리케이션 스키마 정보를 관리하는 기능으로 확장 컬럼을 설정한다. 온톨로지 관리는 데이터 간에 관계성인 카테고리 형태로 일반화(is-a)관계, 부품화(has-a)관계 및 데이터 이질성을 해결할 수 있다. 도메인 관리는 확장 컬럼의 데이터 타입을 설정하기 위한 기능으로, 도메인 타입을 설정함으로써 기본 데이터 타입, 코드 타입, 스크립트 타입을 매핑할 수 있다. 시스템 허브의 구성은 [Fig. 2]와 같다.



[Fig. 2] The DBaaS Framework

- Mediation : 데이터가 이질성으로 발생하는 문제 - 메타 데이터를 이용하여 쿼리 생성 및 조정
- provisioning : 수집하여 처리된 결과를 파일 형태로 공급
- Monitoring : 데이터와 쿼리 조정 처리 과정의 감시
- Merge&Sort : Proxy에 저장된 수집 결과를 Mediation의 도움으로 결합 및 정렬
- Collection : 대용량의 데이터의 수집과 네트워크 상의 이동을 위해 처리된 결과는 표준 스키마 형태의 몽고 DB에서 사용할수있는 파일형식으로 데이터 수집하여 Proxy에 저장
- Request Query : 변환된 쿼리를 협업된 데이터베이스 클라우드에게 요청한다
- Query Conversion : 수신된 추상화 쿼리를 Mediation의 지원으로 로컬 쿼리 생성
- Query Generation : Mediation을 통한 IS-A, HAS-A를 기반으로 하는 쿼리 생성

- Proxy Access : 요구된 데이터는 파일 시스템 형태로 Proxy에 접근하여 BULK Load 한다.

### 3.2 DBaaS를 위한 쿼리 관리자

이 장에서는 글로벌 쿼리의 생성, 글로벌 쿼리와 로컬 쿼리 간의 쿼리 변환에 대해서 기술하도록 한다.

글로벌 쿼리는 테이블을 지칭하는 FROM절이 없다는 것을 제외하는 SQL문과 유사하다. 그리고 필드명은 메타 데이터의 표준 필드명을 이용한다. 글로벌 쿼리의 기본형식은 다음과 같다.

```
SELECT <standard field list> WHERE
<condition>
```

여기서 <standard field list>는 표준 필드 리스트로 사용자 인터페이스에서 제공된다. <condition>은 사용자가 요구하는 정보를 추출하기 위한 검색조건이 된다. 검색조건은 글로벌 쿼리의 표준 필드와 사용자가 원하는 조건이고, 로컬 쿼리로 변환될 때는 추가적인 의미적 관계성이 포함된다. 만약 레거시 시스템에 존재하는 테이블이 두개 이상인 경우는 조인 조건도 추가된다. [Fig. 3]은 쿼리 형식의 규칙이다.

```
<standard field list> ::= <standard field list>|<standard field name>
<condition> ::= <comparison> AND <condition>|
<comparison> AND <condition>|
NOT <comparison>|
<comparison>
<comparison> ::= <standard field name> <op> <value>
<op> ::= =|<>|>|=|<|=
<value> ::= <standard field name>|<expr>
```

[Fig. 3] The Rule of SELECT of Global Query

위의 쿼리 규칙에 따라 쿼리를 수행하는 단계는 다음과 같다.

- 단계1. 글로벌 쿼리 생성.
- 단계2. 쿼리 파싱. 이 단계는 글로벌 쿼리에서 <standard field name>과 <value>항목을 추출한다.
- 단계3. 연관관계에서 IS-A, HAS-A, Equal 추가를 통한 글로벌 쿼리 확장하고 WHERE절을 확장함으로써, 검색의 효율을 향상시킨다.
- 단계4. 로컬항목 변환 및 테이블 정보 추가. <standard field name>는 추출된 로컬 항목으로 변경한다.

- 단계5. 프록시에 문서 지향 화일 생성. <standard field list>를 기반으로 하여 사용자 정보를 통해 PROXY에 결과를 생성한다.

<Table 1>은 상품목록 카테고리의 기본 항목이 데스크탑이며, IS-A 관계의 상품분류와 HAS-A 관계인 상품품목에 대한 분류이다.

<Table 1> The Category of Product List

Metadata Categories	Relation Property	Meta Value	Mapping Value
Product Class	Is-a	Desktop	Standard Desktop
Product Class	Is-a	Desktop	Sentinel Standard Desktop
Product Class	Is-a	Desktop	Sentinel Standard Desktop- Rugged
Material Class	Has-a	Standard Desktop	Manual Set
Material Class	Has-a	Standard Desktop	Monitor - 17"
Material Class	Has-a	Standard Desktop	Keyboard - 101 Key

<Table 2>Expansion of Extracted Query In the Process

Analysis Query	SELECT pcode, pname, indate, unit_price, amount WHERE factory = 'su-won' AND indate >= '2013-12-31' AND pname = 'desktop';
Include of Is-a Property	SELECT pcode, pname, indate, unit_price, amount WHERE factory = 'su-won' AND indate >= '2013-12-31' AND pname in ('desktop') AND pname in ('standard desktop', 'sentinel standard desktop');
Include of Has-a Property	SELECT pcode, pname, indate, unit_price, amount WHERE factory = 'su-won' AND indate >= '2013-12-31' AND pname in ('desktop','standard desktop', 'sentinel standard desktop') AND pname in ('manual set','monitor - 17");
Local Query	SELECT prod_info.pcode, prod_info.pname, to_char(prod_tr_info.date,'YYYY-MM-DD'), prod_tr_info.unit_price, prod_tr_info.amount FROM prod_info, prod_tr_info WHERE prod_info.pcode = prod_tr_info.pcode AND prod_tr_info.factory = 'chang-won' AND to_char(prod_tr_info.indate,'YYYY-MM-DD') >= '2013-12-31' AND prod_info.pname in ('desktop', 'standard desktop', 'sentinel standard desktop', 'manual set', 'monitor - 17");

prod_info	{ pcode: "com_123"
document-Oriented File	pname: "standard_desktop"
	indate: "2013-12-10"
	unit_price: "1,153,000"
	amount: "288,250,000"
	}

<Table 2>는 프로세스에서 추출된 질의 확장 및 문서 지향적인 저장 형식을 나타낸 것이다. 위의 1단계부터 3 단계까지 from 없는 메타쿼리로 질의를 확장하였으며, 4단계에서 로컬 쿼리로 변환하며 FROM 절을 생성한다. 단계5에서 단계4에서 만들어진 테이블을 문서를 <standard field list> 기반으로 생성한다.

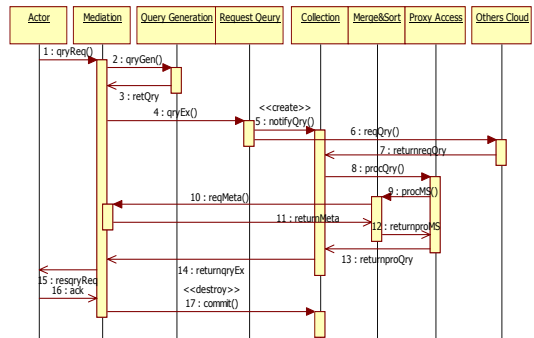
### 3.3 작업 흐름도

전체적인 시스템의 작업 흐름은 [Fig. 4]와 같으며, 각 단계에 발생하는 액션은 다음과 같고, 각 단계에 대한 결과를 포함한다.

- qryReq(): Actor->Mediation. Actor에서 사용자에게 의해 요구를 수용하기 위한 요구사항을 받아 Mediation에게 보낸다.
- qryGen(): Mediation->Query Generation. 사용자의 요구를 받아 수행하기 위한 질의문을 생성한다. 이때 생성되는 쿼리는 표준 형식의 글로벌 쿼리이다.
- qryEx(): Mediation->RequestQuery. 앞에서 생성된 쿼리를 실행한다.
- notifyQry(): RequestQuery->Collection. 쿼리가 수행된 후 결과를 수집하기 위한 환경을 조성한다. 해당 쿼리에 대한 결과를 각 클라우드에서 전송하면 이를 수행된 쿼리별로 결과를 축적한다. 이후 쿼리 수행과 관련된 사항은 Collection에 위임한다
- eqQry(): RequestQuery->Others Cloud. 실제 쿼리를 수행하기 위해 클라우드에 참여한 다른 시스템에게 이를 전송한다. 그 결과를 4의 작업에 의해 생성된 결과 영역에 일시적으로 축적되며, 수행된 쿼리 결과는 Mediation에 리턴된다.
- procQry(): Collection->Proxy Access. 쿼리 결과를 처리하여 결과를 사용자에게 리턴하기 위해 프록시 관리를 수행한다. 병합된 결과를 리턴받는다.
- procMS(): Proxy->Merge&Sort. 이 단계는 클라우드 상의 여러 시스템에서 수집된 결과를 병합을

요구한다. 그 결과를 리턴받는다.

- reqMeta(): Merge&Sort->Mediation. 이 단계는 병합을 위하여 표준 쿼리에서 사용된 글로벌 항을 요청한다. 각 쿼리에 대한 메타 정보는 Mediation에서 쿼리가 생성될 때부터 관리하는데 이 메타 정보를 리턴받는다.
- resqryReq: 1번의 쿼리에 대한 결과 사용자에게 제공하고, 사용자의 액션에 따라 commit()이 수행되고 프로세스는 종료된다.



[Fig. 4] Workflow of the System

<Table 3> Comparison with DBaaS system Document Document based file for NO-SQLbased file for NO-SQL

Compare Item	DBaaS	Proposed System
Integration	Schema level and application level using the master data of based on schema	Schema level and application level using DBaaS hub system
Data Scalability / Overcome Heterogeneity	Data standardization considering the integration and centralized system to collect information . It is hard to expand the data item.	Data standardization considering the integration and centralized system to collect information using ontologoy. It is easier to expand the data item through the IS-A, HAS-A, EQ relation.
Query	SQL	Meta SQL
Result	Table	Document based file for NO-SQL
Type	Static	Dynamic

위와 같은 과정을 통하여 작업은 사용자의 요구에 따라 표준 쿼리가 생성되고, 이 표준 쿼리는 로컬 쿼리로 변환되어 각 로컬 시스템에서 수행한다. 수행한 결과는 수집되어 표준항목으로 변환되어 사용자에게 제공된다.

이에 따라 사용자는 로컬 항목에 대한 내용과 무관하게 수행된 결과를 표준 항목으로 내용을 볼 수 있다. 이는 사용자에게 데이터에 대한 투명성을 제공한다.

### 3.4 비교평가

<Table 3>는 클라우드 환경에서 멀티데이터베이스 통합을 위한 DBaaS를 본 시스템과 비교한 것이다. 오라클 DBaaS는 데이터의 접근 및 교환기술로 스키마 기반의 마스터 데이터를 이용한다. 그러나 본 시스템은 DBaaS 허브 시스템을 이용하여 온톨로지 기반으로 데이터를 접근 및 교환할 수 있다. 첫 번째는 비교항목은 데이터 통합 방법으로 강 결합 스키마 통합방식과 약 결합 방식으로 어플리케이션 결합방식이 있다. 두 번째 항목은 데이터 확장성과 이질성 극복에 대한 항목이다. 제안된 DBaaS는 시스템은 수집정보 일원화뿐만 아니라 수집정보 확장을 위해 데이터 항목 추가 삭제 용이하다. 세 번째로 본 시스템의 질의 유형은 FROM 절이 없는 메타 SQL을 사용한다. 그리고 결과 데이터는 문서 지향 파일로 생성하며, 개발유형은 동적으로 생성을 할 수 있다.

## 4. 결론

본 논문에서는 멀티 데이터베이스 환경에서 시스템을 통합할 때, 로컬시스템의 영향을 최소화하기 위하여 DBaaS 허브 시스템을 제안했다. 본 시스템은 기존 시스템의 변경 없이 데이터를 통합함으로써 비용과 시간을 절감하고, 실시간 데이터 추출 및 정제 작업을 통한 일관성 있는 정보를 생성하여 정보의 품질을 향상시킬 수 있다. 이는 서비스에 따라 서로 다른 데이터베이스를 요구할 수 있으며, 연관된 서비스가 사용하는 데이터베이스는 서로 통합 할 수 있다.

제안된 DBaaS는 데이터 간의 상호운용성 문제를 해결하기 위해 온톨로지를 이용하여 상호운용을 위한 메타 쿼리를 제안하였으며, 이를 통해 위치 투명성을 제공하였고, 실제 데이터베이스 접근은 표준 메타 온톨로지를 통하여 변환된 질의로 접근하게 함으로써 데이터 통합을 기할 수 있다. 또한 표준 메타 데이터를 이용하여 문서 지향 데이터베이스 시스템을 구축하였다.

본 논문에서 설계된 시스템은 클라우드 환경에서 발

생할 수 있는 데이터베이스의 이질적인 문제를 DBaaS 메타데이터를 이용하여 해결하였다. 데이터의 이질성은 표준 스키마 문서와 로컬 데이터베이스의 접근 및 구조 정보를 사전에 등록한다. 다음은 쿼리 작성 및 분석기를 이용하여 표준 스키마 문서를 기준으로 메타 쿼리로 생성하였고, 메타 쿼리는 로컬 데이터베이스 시스템의 쿼리로 변환하여 이질성을 해결할 수 있도록 하였다. 이는 데이터 수집에 따른 문제를 프록시에 파일 형태로 저장하여 데이터 이질성을 해결할 수 있다. 추후 연구는 클라우드 환경에서 실시간 서비스를 위한 트랜잭션 처리 기법을 적용 할 수 있다.

## ACKNOWLEDGMENTS

This present has been conducted by the research grant of Kwangwoon University in 2014.

## REFERENCES

- [1] Aisling O'Driscolla, Jurate Daugelaiteb, Roy D. Sleator, 'Big data', Hadoop and cloud computing in genomics, *Journal of Biomedical Informatics*, 46, 5, pp.774 - 781, 2013.
- [2] Zohreh Sanaei, Saeid Abolfazli, Abdullah Gani,Rajkumar Buyya, Heterogeneity in Mobile Cloud Computing: Taxonomy and Open Challenges, *IEEE Communications Surveys & Tutorials*, 16, 1, pp.369-392, 2014.
- [3] Subashini, S., Kavitha, V., A survey on security issues in service delivery models of cloud computing, *Journal of Network and Computer Applications*. 34, 1, pp.1-11. 2011.
- [4] Soror Sahri, Rim Moussa, Darrell D. E. Long, Salima Benbernou, DBaaS-Expert: A Recommender for the Selection of the Right Cloud Database, *Foundations of Intelligent Systems, Lecture Notes in Computer Science*, 8502, pp.315-324, 2014.
- [5] Pathak, Ajeet Ram, and B. Padmavathi, Analysis of Security Techniques Applied in Database

Outsourcing, International Journal of Computer Science and Information Technologies, 5, 1, pp.665-670, 2014.

- [6] H. Hacigumus, B. Iyer and S. Mehrotra, Providing database as a service, in Proc. of IEEE 18th ICDE, pp. 29-38, 2002.
- [7] Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., and Byers, A. H, Big Data: The Next Frontier for Innovation, Competition, and Productivity, McKinsey Global Institute, 2011.
- [8] Sunil Soares, Big Data Governance, Sunilsoares, 2012.
- [9] Ward, Jonathan Stuart, and Adam Barker. "Undefined By Data: A Survey of Big Data Definitions." arXiv preprint arXiv:1309.5821, 2013.
- [10] Kian Win Ong, Yannis Papakonstantinou, Romain Vernoux, The SQL++ Unifying Semi-structured Query Language, and an Expressiveness Benchmark of SQL-on-Hadoop, NoSQL and NewSQL Databases, arXiv preprint arXiv:1405.3631, 2014.
- [11] Elif Dede, Madhusudhan Govindaraju, Daniel Gunter, Richard Shane Canon, Lavanya Ramakrishnan, Performance evaluation of a mongodb and hadoop platform for scientific data analysis, In Proceedings of the 4th ACM workshop on Scientific cloud computing, pp.13-20, 2013.
- [12] Jing Han, Haihong E, Guan Le, Jian Du, "Survey on NoSQL database", Pervasive Computing and Applications (ICPCA) 6th International Conference on(2011), pp. 363-366, 2011.

**정 계 동(Jung, Kye-dong)**



- 1985년 2월 : 광운대학교 전자계산학과(이학사)
- 1992년 2월 : 광운대학교 산업정보학과(이학석사)
- 2000년 2월 : 광운대학교 컴퓨터과학과(이학박사)
- 1993년 3월 ~ 2004년 12월 : 광운대학교 정보과학원 교수

- 2005년 3월 ~ 현재 : 광운대학교 교양학부 교수
- 관심분야 : XML, 분산시스템, 분산 컴퓨팅기술, 이동에이전

트, 클라우드 컴퓨팅, DBaaS  
 · E-Mail : gdchung@kw.ac.kr

**황 치 곤(Hwang, Chi-Gon)**



- 1995년 2월 : 창원대학교 경영학과(학사)
- 2004년 8월 : 광운대학교 정보통신학과(공학석사)
- 2012년 8월 : 광운대학교 컴퓨터과학과(공학박사)
- 2006년 1월 ~ 현재 : (주) 인천연구원

- 관심분야 : XMDR, 클라우드 컴퓨팅, DBaaS, 서비스 상호이용, 온톨로지, 멀티미디어
- E-Mail : duckl052@kw.ac.kr

**이 종 용(Lee, Jong Yong)**



- 1983년 2월 : 한양대학교 원자력공학과(공학사)
- 1988년 2월 : 광운대학교 전자공학과(공학석사)
- 1993년 2월 : 광운대학교 전자공학과(공학박사)
- 2005년 3월 ~ 현재 : 광운대학교 교수

- 관심분야 : 자동제어, 로보틱스, 영상인식
- E-Mail : jyonglee@kw.ac.kr

**신 효 영(Shin, Hyo-Young)**



- 1986년 2월 : 광운대학교 전자계산학과(이학사)
- 1988년 2월 : 광운대학교 전자계산학과(이학석사)
- 1998년 8월 : 광운대학교 전자계산학과(이학박사)
- 1988년 2월 ~ 1993년 8월 : LG 소프트웨어 연구소

- 1994년 2월 ~ 현재 : 경북대학교 IT보안과 부교수
- 관심분야 : 네트워크 보안, 분산 시스템
- E-Mail : hyshin@kbu.ac.kr