

클라우드 데이터 관리 인터페이스(CDMI) 시스템 설계 및 구현

Design and Implementation of a Cloud Data Management Interface(CDMI) System

안민제, 전인배, 손인국, 이하, 박용훈, 임종태, 복경수, 유재수
충북대학교 정보통신공학과

Minje Ahn(allgrudge86@gmail.com), Inbae Jeon(lovejib@naver.com),
Ingook Son(dlsrnr94422@nate.com), He Li(lihelol@gmail.com),
Yonghun Park(ffishh@nate.com), Jongtae Lim(jtlim@chungbuk.ac.kr),
Kyoungsoo Bok(ksbok@chungbuk.ac.kr), Jaesoo Yoo(yjs@chungbuk.ac.kr)

요약

최근 대규모의 데이터를 낮은 비용으로 효율적으로 처리 가능한 클라우드 컴퓨팅 기술의 발달로 클라우드 데이터를 관리하기 위한 연구들이 활발하게 진행되고 있다. 그러나 현재 연구되거나 서비스되고 있는 다양한 형태의 클라우드 컴퓨팅 플랫폼들은 기술적, 정책적으로 개방성을 띄지 않아 사용자의 구축 환경에 따른 상호 운용성을 보장하지 못한다. 즉, 사용자가 관련 서비스를 제공받기 위해서는 단일 업체의 서비스만을 제공받아야만 한다. 본 논문에서는 클라우드 환경에서 데이터 생성, 검색, 갱신, 삭제에 대해 CDMI 국제 표준을 준수하는 저장소 인터페이스를 설계하고 구현한다. 이러한 인터페이스는 특별한 제약 없이 클라우드 저장소를 구축하거나 사용하려는 개발자들에게 필요한 필수 기능들을 제공한다. 또한, 제한하는 시스템의 구현을 통해 CDMI 국제 표준의 활용성과 운용성을 검증한다.

■ 중심어 : | 클라우드 컴퓨팅 | 데이터 관리 | 저장소 |

Abstract

Recently, cloud data management has been actively studied along with the development of a cloud computing technology that can process large amounts of data at a lower cost. However, the existing cloud computing platforms do not guarantee interoperability according to the construction environments of users because they do not provide technical and political openness. In other words, in order for users to receive the related services, they use services provided by only one vendor. In this paper, we design and implement a storage interface that supports the international standard CDMI in order to retrieve, update, and delete data in cloud environments. These interfaces provide the functionality required for developers who want to build and use the cloud storage without special restrictions. In addition, we verify the operability and usability of CDMI international standard through the implementation of the proposed system.

■ keyword : | Cloud Computing | Data Management | Storage |

* 본 연구는 산업통상자원부의 출연금으로 수행한 산학융합지구조성사업 및 교육부와 한국연구재단의 지역혁신인력양성사업(No.2013H1B8A2032298)으로 수행된 연구결과임

접수일자 : 2013년 07월 15일

심사완료일 : 2013년 08월 06일

수정일자 : 2013년 08월 05일

교신저자 : 유재수, e-mail : yjs@chungbuk.ac.kr

I. 서론

최근 소셜 네트워크 서비스 등과 같은 소셜 미디어의 성장과 스마트 폰의 보급화로 인해 다양한 형태의 대용량 데이터가 급속도로 생산, 유통, 저장되고 있다. 이러한 대용량의 데이터를 저장, 관리 및 분석하기 위한 인프라를 구축하기 위해서는 막대한 비용과 시간이 소요된다. 따라서 대규모 투자 없이 효율적인 데이터 분석 및 관리가 가능한 클라우드 컴퓨팅 기술이 대두되고 있다[1]. 이와 함께 다양한 형태의 클라우드 컴퓨팅 서비스를 제공하기 위한 기술들이 연구되고 있다[2][3]. 특히, 클라우드 컴퓨팅을 통해 데이터 저장소를 사용하기 위한 저장 관리 기술에 대한 연구들이 활발하게 연구되고 있다[4-6].

현재 다양하게 연구되고 있는 클라우드 컴퓨팅 서비스는 크게 SaaS(Software as a Service), PaaS(Platform as a Service), IaaS(Infrastructure as a Service) 세 가지로 분류된다[1][7]. 구글, 아마존, 마이크로소프트, IBM, HP 등과 같은 IT 전문 업체들은 클라우드 서비스 자체 제공에서 클라우드 인프라를 구축하고 제공하기 위한 플랫폼 개발에 집중하고 있다[8]. 아마존은 AWS(Amazon Web Service)를 제공하면서 클라우드 서비스를 제공하기 시작하였다. 2006년 아마존은 EC2, S3 서비스를 제공하면서 자사의 전자상거래와 무관하게 가상의 컴퓨팅 환경을 제공하고 있다. 구글은 Google App Engine을 통해 파일 시스템과 데이터 저장소를 활용할 수 있는 응용 서비스를 개발을 지원하는 PaaS를 제공한다. 마이크로소프트는 IaaS, PaaS, SaaS를 모두 제공하고 있으며 Azure Services Platform를 통해 클라우드 시장을 공략하고 있다. 또한, 자사의 주요 솔루션을 SaaS로 전환하고 있다.

기존에 연구되거나 서비스되고 있는 클라우드 서비스들은 서로 다른 구조적, 기술적인 특성을 지니고 있어 이들 간의 상호운용성이 보장되지 못한다. 따라서 사용자가 클라우드 서비스를 제공받기 위해서는 단일 플랫폼으로 연동되어야만 하기 때문에 선택의 폭이 매우 제한적이다. 이러한 문제점을 해결하기 위해서는 클라우드 서비스들 간에 표준화된 인터페이스를 제공할

필요가 있다. SNIA에서는 클라우드 저장소를 구축하거나 사용하려는 개발자에게 클라우드 저장소를 접근하고 그 내부에 데이터를 관리를 위한 CDMI(Cloud Data Management Interface) 표준 인터페이스를 제시하였다[9]. 이를 통해 사용자에게는 상호 운용성을 통한 폭 넓은 선택을 제공하고 서비스 제공자에게는 공정한 상호 경쟁 속에 클라우드 컴퓨팅 산업의 활성화와 업계의 관심을 고조시킬 수 있다.

본 논문에서는 [10]에서 발표한 논문을 확장한 것으로 SNIA에서 제시한 CDMI 국제 표준에 따라 클라우드 저장소를 접근하고 데이터 관리를 수행하기 위한 시스템을 설계하고 구현한다. 개발하는 시스템은 CDMI 클라이언트, CDMI 서버, CDMI 메타데이터 서버로 구성된다. CDMI 서버는 CDMI 클라이언트로부터 요청을 받아 처리하며 CDMI 메타데이터 서버는 시스템 내부적인 파일시스템에서 CDMI를 제공하기 위해 사용되는 정보를 관리한다. 본 논문에서 제시한 인터페이스의 효율성을 검증하기 위해 표준 HTTP에 기반을 둔 별도의 전용 클라이언트를 구현한다. 이를 통해 CDMI 국제 표준의 활용성과 운용성을 확인하고, 국내 클라우드 업계의 관심과 국내 표준의 필요성을 부각시킬 수 있다.

본 논문의 나머지 구성은 다음과 같다. II장에서는 클라우드 서비스 제공을 위한 플랫폼 및 표준화 방향에 대해 기술하고 III장에서는 제안하는 CDMI 인터페이스 제공하기 위해 시스템 설계 내용을 기술한다. IV장에서는 CDMI 시스템 구현 내용을 기술하고 V장에서는 본 논문의 결론을 기술한다.

II. 관련 연구

클라우드 컴퓨팅 산업의 활성화로 다양한 형태의 클라우드 컴퓨팅 서비스가 서비스되거나 연구되고 있다. 서비스 되는 종류는 크게 세 가지로 구분된다. SaaS는 클라우드 서비스 제공자가 응용 소프트웨어를 제공하고 사용자가 이를 웹을 통해 이용하는 구조이다. 대표적인 적용사례로는 NTT(Nippon Telegraph and Telephone Corporation)의 IP-VPN(Virtual Private

Network) 서비스가 있다. NTT의 폐쇄 망을 이용하여 애플리케이션 서비스를 제공하고 있다[11]. PaaS는 사용자가 서비스 제공자로부터 소프트웨어를 개발할 수 있는 환경을 제공받는 서비스이다[1][7]. 대표적인 예로 Google의 AppEngine 서비스와 Microsoft의 Azure 등을 들 수 있다. AppEngine은 JAVA나 Python 등의 개발 환경을 Google에서 제공하는 인프라 상에서 이용할 수 있다[12]. Azure는 응용 소프트웨어뿐만 아니라 다양한 웹 개발 환경을 제공한다[13]. PaaS 특성상 개발 환경을 제공해주는 서비스이기 때문에 개발 상태를 유지 및 보수를 목적으로 IaaS를 수반하는 경우가 일반적이다. IaaS는 서버 환경을 서비스로 제공받는다. 사용자는 대용량의 저장장치나 고성능 컴퓨팅 자원을 서비스 제공자로부터 제공받을 수 있다. 최근 일반 사용자가 가장 쉽게 접근하거나 경험할 수 있는 서비스로, NHN의 N드라이브, 아마존 웹 서비스(AWS)의 S3, EC2 등을 예로 들 수 있다[7].

이들의 특징은 하드웨어적인 장치를 직접 판매하는 것이 아닌, 해당 자원을 웹을 통해서 제공한다는 점이다. 이를 통해 사용자는 자신이 필요로 하는 컴퓨팅 자원을 효율적으로 활용할 수 있다. 이와 같이 다양한 형태로 제공되는 기존 클라우드 컴퓨팅 기술들은 대부분 기업들이 서비스 판매를 목적으로 개발되었기 때문에 독자적인 구조를 지니고 있다. 따라서 클라우드 서비스들 간의 상호 운용성이 떨어진다. 즉, 특정 클라우드 서비스를 제공받는 사용자가 제공받는 인프라를 확장하기 위해서는 해당 서비스를 제공하는 업체로 이전하거나, 사용 중인 서비스를 확장하는 방법 밖에 존재하지 않는다.

최근 연구되고 있는 클라우드 데이터 관리 인터페이스들은 대부분 성능 향상을 목적으로 하는 연구가 대부분이다. 따라서 인터페이스 규격이 표준화보다는 성능 향상을 위한 독자적인 구조를 지니고 있다[14][15]. 이와 같은 독자 구조가 가지는 상호 운용성 저하의 문제점을 해결하기 위해서 SNIA에서는 2011년 클라우드 데이터관리를 위한 인터페이스를 규정하는 CDMI 표준안을 처음 제시하였고, 2012년 11월에 국제 표준으로 등재되었다[9]. CDMI는 서비스 제공자의 플랫폼과 사

용자가 접근하는 웹 사이를 연결해주는 표준 인터페이스이다. 이처럼 표준화된 인터페이스를 통해 사용자는 다양한 서비스들을 하나의 웹 화면 내에서 병행 또는 복합적으로 서비스를 제공받는 것이 가능해진다. 따라서 사용자는 필요에 따라 다양한 서비스를 활용할 수 있게 된다. 또한 본 논문에서는 국내에서 아직 CDMI 국내 표준화가 진행 중이고 업체들도 여전히 독립적인 구조를 지니고 있기 때문에, CDMI 국제 표준에 대한 성능과 활용성 검증 및 국내 표준의 필요성을 보이기 위해서 CDMI 인터페이스를 구현하고자 한다.

III. 제안하는 시스템

1. 제안하는 시스템의 구조

제안하는 시스템은 CDMI 클라이언트(CDMI Clients), CDMI 서버(CDMI Servers), CDMI 메타데이터 서버(CDMI Metadata Servers)로 구성된다. [그림 1]은 구현 시스템의 구조를 나타낸다. CDMI 클라이언트는 테스트를 효율적으로 수행하기 위해 자체적으로 구현한 것이고, 다른 restful을 지원하는 클라이언트를 이용하여도 된다. 다만 CDMI 표준을 따라야 한다. 실제 데이터는 전용 클라이언트 또는 웹을 통해 CDMI 표준 규격에 따라 CDMI 서버로 요청을 전달한다. CDMI 서버는 클라이언트 요청을 수신하고 응답을 수행한다.

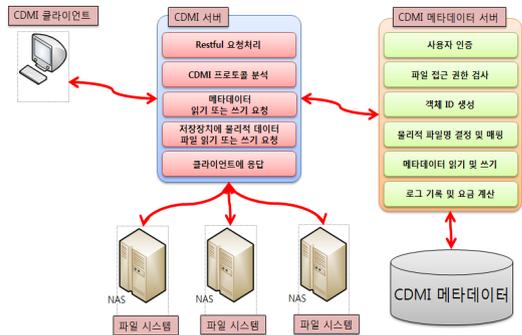


그림 1. 제안하는 시스템의 구조

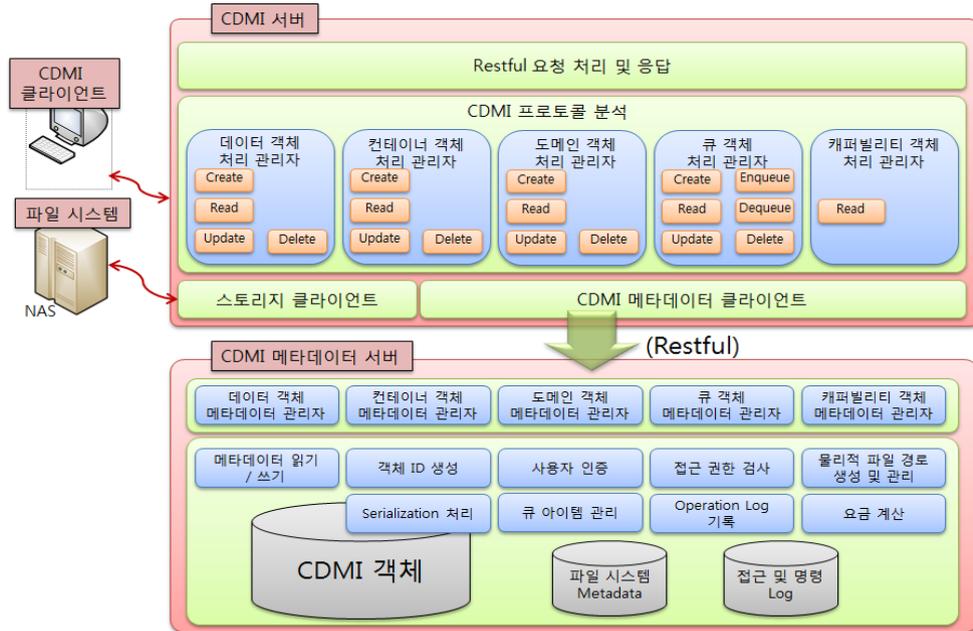


그림 2. CDMI 서버와 CDMI 메타데이터 서버의 역할

CDMI 메타데이터 서버는 클라이언트와 서버 사이의 데이터 송수신에 따라 발생하는 부가적인 로그 기록 및 메타데이터 정보들을 일괄적으로 유지하고 관리하는 서버이다. 또한, 데이터 접근에 필요한 물리적인 주소 정보와의 매핑 정보도 함께 보관하고 있게 된다. 따라서 사용자 요청이 CDMI 서버로 전송되면 모든 수행 절차에 따른 사용자 정보 및 저장 장치의 갱신 내역이 CDMI 메타데이터 서버로 전송되고 이를 누락 없이 기록한다. 만약 메타데이터 서버에 기록이 누락되거나 오류가 발생하게 되면 기존에 절차상 수행된 내용을 다시 이전 상태로 복구를 수행한다.

2. CDMI 서버 및 CDMI 메타데이터 서버

CDMI 서버는 CDMI 클라이언트로부터 요청을 수신하고 요청된 내용에 대한 표준 준수 여부 및 CDMI 버전 등을 검사한다. 서버는 요청된 내용을 분석하여 자체적인 시스템 명령으로 변환하고 클라이언트로부터 받은 명령을 처리한다. 클라이언트로부터 받은 명령 처리는 메타데이터에 대한 처리와 실제 콘텐츠에 대한 처리로

구분된다. 이러한 과정은 일반적인 파일시스템의 처리 과정을 유사하다. CDMI 클라이언트로부터 파일 생성, 수정, 삭제 명령이 요청되면 CDMI 서버는 메타데이터를 검사하여 해당 명령 처리가 가능하지 검사한다. 만약 명령 처리가 가능할 경우 메타데이터에 대한 정보를 갱신한 후 실제 콘텐츠에 대한 처리를 수행한다. 실제 콘텐츠에 대한 처리를 위해 해당 콘텐츠를 저장하기 위한 파일을 검색 기능과 해당 파일을 수정하는 기능을 수행한다. 메타데이터 처리는 CDMI 메타데이터 서버에 위임하고 콘텐츠에 대한 처리는 CDMI 서버에서 NAS를 직접 마운트하여 수행한다.

CDMI 메타데이터 서버는 내부적인 파일 시스템에서 CDMI 기능을 제공하기 위해 사용되는 정보를 관리한다. 즉, 사용자 인증, 권한 검사, 객체 ID 생성, CDMI 객체 이름과 실제 물리적인 파일의 이름 매핑, 메타데이터에 대한 효율적인 입출력, 시스템 접근 로그 기록 및 로그를 바탕으로 요금 계산을 수행한다. 이러한 기능은 파일 시스템 내부적으로 이루어지는 것과 유사하다.

3. 세부 처리 절차

[그림 2]는 CDMI 서버와 CDMI 메타데이터 서버의 기능을 세부적으로 나타낸 것이다. 데이터 객체, 그를 포함하는 컨테이너 객체, 계정을 관리하는 도메인 객체 등 클라우드 상의 데이터 관리를 위한 필수적인 기능들을 모두 명시하고 있다. CDMI 서버는 Restful 요청을 접수하고 처리 결과를 클라이언트에게 전달하는 역할을 수행한다. CDMI 서버는 클라이언트의 요청을 분석하기 위해 데이터 객체 관리자, 컨테이너 객체 관리자, 도메인 객체 관리자, 큐 객체 관리자, 캐퍼빌리티 객체 관리자로 구성된다.

각 객체 관리자는 프로토콜을 분석하여 요청된 기능을 수행하기 위한 내부적인 처리를 전달한다. 객체 및 컨테이너 객체 관리자는 실질적인 파일 처리가 이루어지기 때문에 대용량 파일 전송을 위한 데이터 스트리밍, 파일 생성, 복사 및 삭제 등의 부가적인 기능을 갖고 있다. 도메인 객체 관리자는 해당 프로그램을 서비스하는 측면에서 필수적인 요소이다. 실제 각 사용자의 사용량 등을 체크하여 기록하기 때문이다. 큐 객체 관리자는 서비스 제공자가 복수의 클라이언트로부터 받는 요청을 누락 없이 순차적으로 실행시키기 위해 존재한다. 넓은 범위의 큐를 지칭하고 있으며, 세부적으로 효율적인 시스템 구성을 위한 큐 설계에는 제약사항이 없다. 캐퍼빌리티 객체 관리자는 클라이언트와 CDMI 서버 간의 버전 확인 및 제공 가능 서비스 항목 정보 등을 얻을 수 있다. 이를 통해 이종의 플랫폼 간에도 동일한 형태의 제공 가능 서비스 항목을 읽어 올 수 있다. 메타데이터에 대한 처리는 CDMI 서버에 포함된 CDMI 메타데이터 클라이언트를 통해 CDMI 메타데이터 서버에게 요청되며 파일에 대한 처리는 저장소 클라이언트를 통해 저장소에게 요청한다. CDMI 메타데이터 클라이언트와 CDMI 메타데이터 서버와의 통신은 별도의 표준은 존재하지 않으나 본 시스템에서는 CDMI 클라이언트와 CDMI 서버 간의 통신 규격과 동일한 Restful 프로토콜을 기반으로 구현한다.

CDMI 메타데이터 서버는 각 객체 별로 메타데이터 관리자를 통해 메타데이터 처리를 수행한다. 메타데이터 관리자는 데이터 객체 메타데이터 관리자, 컨테이너

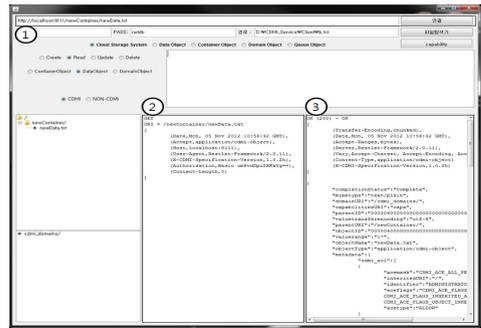
객체 메타데이터 관리자, 도메인 메타데이터 관리자, 큐 객체 메타데이터 관리자, 캐퍼빌리티 객체 메타데이터 관리자로 구성된다. 이러한 메타데이터 관리자는 각 객체에 대한 메타데이터 처리를 전담하고 처리된 결과를 CDMI 메타데이터 클라이언트에게 전달한다. 각 객체의 메타데이터 관리자는 메타데이터에 직접 접근하지 않는다. 이는 시스템의 안정성을 고려한 것으로 각 기능별 메타데이터를 처리하는 절차에 따라 기능을 설계하였다. 대표적인 기능으로서 객체 ID 생성, 사용자 인증, 접근 권한 검사, 파일의 물리적 경로 생성 및 관리 등이 있다. 또한 일관성을 유지하기 위해 통신 장애, 파일 오류, 읽기 장애 등이 발생할 경우 메타데이터를 복구할 수 있도록 설계하였다.

IV. 구현 및 예제

CDMI 서비스를 제공하는 서버 및 클라이언트는 모두 JAVA 6.0으로 구현되었으며, 리눅스와 윈도우즈 환경을 모두 지원한다. 구현에서 restful 프로토콜을 처리하기 위해 java 라이브러리 패키지인 restlet을 이용하였다. 또한, 메타데이터를 관리하기 위해 MySql DBMS를 이용하였고, 실제 콘텐츠가 저장되는 파일 시스템은 NAS를 이용하였다. 테스트를 위한 클라이언트는 Oracle에서 제공하는 HTTP 테스트 프로그램을 사용할 수 있으나 CDMI에 맞는 표준 규격을 매번 작성하여야 하는 불편함이 있어 동일한 통신 환경을 제공하는 전용 클라이언트를 구현하였다.

제안한 CDMI 서버의 효율성 및 처리 과정을 검증하기 위해 클라이언트에서 CDMI 표준을 준수한 명령을 전송하고 그 결과를 반환하는 과정으로 확인한다. [그림 3]의 (a)는 구현된 내용 중 캐퍼빌리티 객체를 통해 시스템의 구현된 기능을 조회한 결과이다. 1번의 선택 여부에 따라 캐퍼빌리티 객체에 대한 요청이 이루어진다. 2번과 같은 정보를 GET 방식으로 요청하면 3번과 같이 OK라는 상태 코드와 함께 현재 시스템 정보를 반환하는 것을 확인할 수 있다. [그림 3]의 (b)는 컨테이너 객체를 통해 사용자 도메인 내에 컨테이너를 생성한 결

파이다. 1번에 새롭게 생성하고 싶은 컨테이너 객체의 이름을 적어 host 다음에 /newContainer/라는 URI로 PUT요청을 3번과 같은 정보로 요청을 하면 4번과 같이 Created라는 상태 코드와 해당 CDMI 버전정보, 객체의 유형 정보 등 다양한 헤더 정보와 함께 생성된 컨테이너 객체에 대한 정보들이 반환된다. 실제로 2번을 통해 / 아래에 자식으로 newContainer라는 컨테이너 객체가 생성됨을 볼 수 있다. [그림 3]의 (c)는 생성된 컨테이너 내부에 데이터 객체를 추가적으로 생성하고, 해당 객체의 콘텐츠 내용 정보를 요청한 결과이다. 1번란에 조회하고 싶은 데이터 객체의 이름을 적어 host 다음에 /newContainer/newData.txt 라는 URI로 GET 요청을 2번과 같은 정보로 요청을 하면 3번과 같이 OK 라는 상태 코드와 해당 CDMI 버전정보, 객체의 유형 정보 등 다양한 헤더 정보와 요청한 데이터 객체에 대한 정보들이 반환된다. 요청한 데이터 객체가 텍스트 형태인 경우에는 내용이 텍스트 형태로 전송되어 오지만, 일반 파일인 경우 Base64로 변환되어져 있기 때문에 클라이언트 측에서 정상적으로 읽기 위해서는 디코딩 과정이 필요하다.



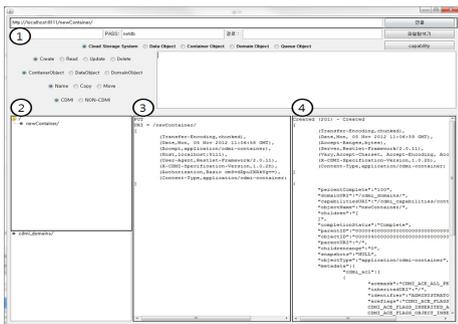
(c) 데이터 객체 읽기
그림 3. CDMI 표준 명령 처리 예제

V. 결론

본 논문에서는 클라우드 데이터를 관리하기 위해 CDMI 표준을 준수하는 인터페이스를 설계하고 구현하였다. 개발한 시스템은 클라우드 환경에서 데이터 생성, 검색, 갱신, 삭제에 위해 CDMI 클라이언트, CDMI 서버, CDMI 메타데이터 서버로 구성된다. 제한한 CDMI 서버의 효율성 및 처리 과정을 검증하기 위해 클라이언트에서 CDMI 표준을 준수한 명령을 전송하고 그 결과를 반환하는 과정으로 확인하였다. 구현을 통한 성능 평가를 토대로 SNIA에서 제시한 CDMI 국제 표준의 활용 가능성을 확인하였고, 이를 통해 사용자에게는 선택의 다양성을 보장하고, 서비스 제공자들에게는 동등한 기회를 통한 상호 경쟁 속에 클라우드 컴퓨팅 산업 전반에 활력을 불어 넣을 수 있을 것으로 기대된다. 향후 연구로 본 논문에서 제시한 인터페이스를 이용하여 빅 데이터 처리에 필요한 데이터를 저장 관리하기 위한 연구를 진행할 예정이다.



(a) 캐퍼빌리티 객체를 통한 구현된 기능을 조회



(b) 컨테이너 객체 생성

참고 문헌

[1] 민옥기, 김학영, 남궁한, “클라우드 컴퓨팅 기술 동향”, 전자통신동향분석, 제24권, 제4호, pp.1-13, 2009.
[2] 한재선, “컴퓨팅 플랫폼과 오픈 플랫폼 기술”, 정

보처리학회지, 제16권, 제2호, pp.39-50, 2009.

[3] J. Peng, X. Zhang, Z. Lei, B. Zhang, W. Zhang, and Q. Li, "Comparison of Several Cloud Computing Platforms," Proc. International Symposium on Information Science and Engineering, pp.23-29, 2009(12).

[4] G. Gu, Q. Li, X. Wen, Y. Gao, and X. Zhang, "An Overview of Newly Open-Source Cloud Storage Platforms," Proc. International Conference on Granular Computing, pp.142-147, 2012(8).

[5] 이병엽, 박준호, 유재수, "클라우드 서비스를 위한 고가용성 대용량 데이터 처리 아키텍처", 한국콘텐츠학회논문지, 제13권, 제2호, pp.32-43, 2013.

[6] 이병엽, 박준호, 유재수, "고가용성 클라우드 컴퓨팅 구축을 위한 그리드 소프트웨어 아키텍처", 한국콘텐츠학회논문지, 제12권, 제2호, pp.19-29, 2012.

[7] https://en.wikipedia.org/wiki/Cloud_computing/

[8] 정제호, "클라우드 컴퓨팅의 현재와 미래 그리고 시장전략", 한국소프트웨어진흥원 정책리포트, pp.56-85, 2008.

[9] SNIA, *Information technology -- Cloud Data Management Interface (CDMI)*, ISO/IEC 17826:2012, 2012.

[10] 안민제, 전인배, 손인국, 이하, 박용훈, 임종태, 북경수, 유재수, "클라우드 데이터 관리를 위한 CDMI 인터페이스의 설계 및 구현", 한국콘텐츠학회 춘계종합학술대회, pp.297-298, 2013.

[11] 윤용익, *미래 서비스를 위한 스마트 클라우드 모델*, 통신연합, 2010.

[12] <https://developers.google.com/appengine/>

[13] <http://www.windowsazure.com/>

[14] A. M. Lonea, D. E. Popescu, and O. Prosteian, "A survey of management interfaces for eucalyptus cloud," Proc. International Symposium on Applied Computational Intelligence and Informatics, pp.261-266, 2012.

[15] S. Venticinquie, L. Tasquier, and B. D. Martino,

"Agents Based Cloud Computing Interface for Resource Provisioning and Management," Proc. International Conference on Complex, Intelligent and Software Intensive Systems, pp.249-256, 2012.

저 자 소 개

안 민 제(Minje Ahn)

준회원



- 2012년 2월 : 충북대학교 정보통신공학과(공학사)
- 2012년 3월 ~ 현재 : 충북대학교 정보통신공학과 석사과정

<관심분야> : 데이터베이스 시스템, 소셜 네트워크 서비스, 빅데이터, 무선 센서 네트워크 등

전 인 배(Inbae Jeon)

준회원



- 2012년 2월 : 충북대학교 정보통신공학과(공학사)
- 2012년 3월 ~ 현재 : 충북대학교 정보통신공학과 석사과정

<관심분야> : 데이터베이스 시스템, 소셜 네트워크 서비스, 빅데이터, 무선 센서 네트워크

손 인 국(Ingook Son)

준회원



- 2012년 2월 : 충북대학교 정보통신공학과(공학사)
- 2012년 3월 ~ 현재 : 충북대학교 정보통신공학과 석사과정

<관심분야> : 모바일 네트워크, 무선 센서 네트워크, 위치 기반 서비스, 빅 데이터, 클라우드 컴퓨팅 등

이 하(He Li)

정회원



- 2006년 7월 : 운남대학교 컴퓨터 공학과(공학사)
- 2010년 8월 : 충북대학교 정보통신공학과(공학석사)
- 2010년 8월 ~ 현재 : 충북대학교 정보통신공학과 박사과정

<관심분야> : 데이터베이스 시스템, 모바일 소셜 네트워크 등

박 용 훈(Yonghun Park)

정회원



- 2005년 2월 : 호원대학교 정보통신공학과(공학사)
- 2007년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2012년 2월 : 충북대학교 정보통신공학과(공학박사)

- 2012년 9월 ~ 현재 : 조지아 공과대학교 전산학과 Postdoc

<관심분야> : 데이터베이스 시스템, 정보검색, 시공간 데이터베이스, 파일시스템, 위치기반서비스

임 중 태(Jongtae Lim)

정회원



- 2009년 2월 : 충북대학교 정보통신공학과(공학사)
- 2011년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2011년 3월 ~ 현재 : 충북대학교 정보통신공학과 박사과정

<관심분야> : 데이터베이스 시스템, 시공간 데이터베이스, 위치기반 서비스, 모바일 P2P 네트워크, 빅데이터

복 경 수(Kyungsoo Bok)

정회원



- 1998년 2월 : 충북대학교 수학과(이학사)
- 2000년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2005년 2월 : 충북대학교 정보통신공학과(공학박사)

- 2005년 3월 ~ 2008년 2월 : 한국과학기술원 정보전자연구소 Pstdoc

- 2008년 3월 ~ 2011년 2월 : 가인정보기술 연구소 차장

- 2011년 3월 ~ 현재 : 충북대학교 정보통신공학부 초빙부교수

<관심분야> : 데이터베이스 시스템, 자료저장시스템, 이동객체 데이터베이스, RFID 및 센서네트워크, 모바일 P2P 네트워크, 빅데이터

유 재 수(Jaesoo Yoo)

종신회원



- 1989년 2월 : 전북대학교컴퓨터공학과(공학사)
- 1991년 2월 : 한국과학기술원 전산학과(공학석사)
- 1995년 2월 : 한국과학기술원 전산학과(공학박사)

- 1995년 3월 ~ 1996년 8월 : 목포대학교 전산통계학과 전임강사

- 1996년 8월 ~ 현재 : 충북대학교 정보통신공학부 및 컴퓨터정보통신연구소 교수

<관심분야> : 데이터베이스시스템, 정보검색, 센서네트워크 및 RFID, 멀티미디어 데이터베이스, 분산객체컴퓨팅, 빅데이터