

다수의 클라우드 서비스의 효율적인 사용과 보안성 향상을 위한 브로커 서비스 구현

정상미*, 이윤호**, 조익환***, 조민재***, 이항복***, 황인원***, 위선민***

*안랩 SW QA 책임 연구원

**서울과학기술대학교 ITM 주임교수

*** 서울과학기술대학교 ITM 전공

Implement Broker Service for Efficient Use and Security of Multiple Cloud Services

Sang-Mi Jung*, Youn-Ho Lee**, Ik-Hwan Cho***, Min-Jae Cho***, Han-Bok Lee***,

In-Won Hwang***, Sun-Min Wi***

*AhnLab SW QA

**Seoul National University of Science & Technology, ITM Head of Professor

***Seoul National University of Science & Technology, ITM

요 약

본 논문에서는 구글 드라이브, 드롭박스 등 여러 회사에서 제공하고 있는 클라우드들을 하나의 플랫폼으로 통합하여 체계적이고 효율적으로 관리할 수 있는 시스템을 제안한다. 또한 해당 시스템을 통한 파일 업로드 및 다운로드를 진행할 경우, 파일을 여러 개의 패킷으로 분할하여 등록 된 클라우드에 각각 저장함으로써, 사용자의 데이터를 안전하게 보호할 수 있는 환경을 구축하고자 한다. 이를 위해 활용된 오픈 API 와 독자적으로 구축한 파일 시스템에 대해 설명하고, 실제 구현한 소프트웨어가 정상적으로 작동하는지 실험을 통해 검증하고자 한다. 최근 클라우드 보안 문제가 대두되는 시점에서 제안한 시스템이 향후 데이터를 보호하기 위한 기술로 활용될 수 있기를 기대한다.

1. 서론

무료 클라우드 서비스가 다양 해져 감에 따라 최근 여러 개의 클라우드 서비스를 이용하는 사용자가 증가하고 있다. 이는 데이터 유지 및 장소에 구애 받지 않고 정보를 관리할 수 있다는 장점이 있다. 하지만 다수의 클라우드 서비스를 효과적으로 관리하는 데에는 어려움이 있다. 여러 클라우드 서비스를 이용할 경우, 원하는 파일이 어디에 저장 되어 있는지 알지 못하거나 업로드 할 파일의 용량이 무료로 제공되는 저장 공간의 용량 보다 크다면 문제가 발생한다. 또한 사용자 데이터 유출과 같은 클라우드 관련 보안 문제가 사회적으로 큰 이슈가 되면서 관련 된 연구가 활발히 진행되고 있다.

본 논문에서는 무료로 제공되고 있는 클라우드들을 하나의 저장 공간으로 통합하여 사용자들이 효율적으로 관리하고 사용할 수 있는 플랫폼을 구축하고자 한다. 또한 기존에 제공 되어지는 무료 클라우드 서비스들을 연동하여, 각각의 클라우드에 분할 된 패킷을 업로드해 저장하는 파일 분할 시스템을 구축하고자 한다. 더불어 해당 시스템이 정상 작동하는지에 대해

검증 할 뿐만 아니라 구현하고자 하는 시스템의 동작 흐름을 소개하고, 시스템 동작에 필요한 API 및 기술을 설명하고자 한다.



그림 1. 메인 페이지. 웹 페이지 접속 시 화면으로 로그인 후 시보드로 이동할 수 있다.

2. 서비스의 특징 및 장점

클라우드 통합관리는 Open API 를 이용하여 사용자가

등록한 기존 클라우드 서비스와 동기화가 가능하도록 하여 기존 이용 서비스의 동시 제어를 보장한다. 특히 대시보드는 액세스 토큰(Access Token)을 이용하여 사용자가 기존 사용하고 있던 각 클라우드 서비스에서 제공하는 드라이브를 대시보드 상에서 쉽게 등록하고 삭제할 수 있다. 사용자는 등록되어 있는 각각의 드라이브를 ‘내 드라이브’ 기능을 통해 통합관리(예: 클라우드 활성화 / 비활성화)가 가능하다. 또한 데이터의 안정성을 보장하기 위해 패킷 분할을 통해 사용자의 파일이 기존에 등록 된 각 클라우드 서버에 분할 되어 업로드 된다. 또한 소프트웨어 추가 설치가 불필요하다는 장점이 있다. API 연동을 통해 Javascript 와 액세스 토큰을 통한 각 드라이브 서버와의 데이터 전송 방식은 REST 방식을 이용함과 동시에 서버 내에서의 Hash 함수를 이용하여 디렉토리 구조 및 중복을 방지한다. 결과적으로 로그파일을 생성함으로써 업로드 및 다운로드 시 필요한 데이터에 접근이 가능하고, 전체적인 서비스의 성능을 향상시킨다.

3. 서비스 구성도 및 기능

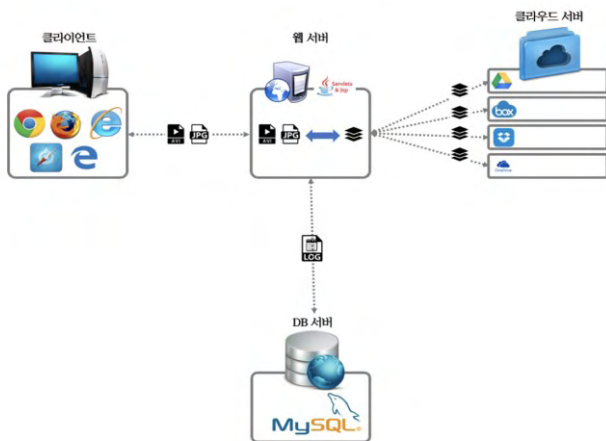


그림 2. 서비스 구성도

그림 2 는 서비스 구성도에 관한 사진이다. 사용자는 클라이언트 쪽에서 웹서버에서 제공하는 웹에 접속하여 사용자의 개인 정보를 통해 회원가입을 한다. 이후 웹서버는 그 정보를 DB 서버에 저장 한 다음 사용자가 연동한 드라이브 관련 정보 (Authentication 을 통한 Access Token 등)를 DB 에서 관리한다.

만일 사용자가 파일을 업로드 한다면 ZIP4J API 를 통해 분할 시킨 다음 서비스의 자체 플랫폼을 통해 각각 클라우드 서버에 병렬적으로 업로드를 시킨다. 그리고 각각 클라우드에 저장 된 파일들의 클라우드 자체 ID 값을 모두 저장한 Log 파일을 만들어 서비스 내 자체 Hash 함수를 통한 파일 관리 알고리즘에 의해 관리한다.

3.1 대시보드(Dashboard)

클라우드 관리, 설정, 파일 관리 및 분할 업로드, 다운로드 등 주요 기능들을 지원하는 통합 페이지이

다.



그림 3. 대시보드 페이지. 로그인 후 클라우드 관리, 설정 등 서비스 관리 기능을 제공한다.

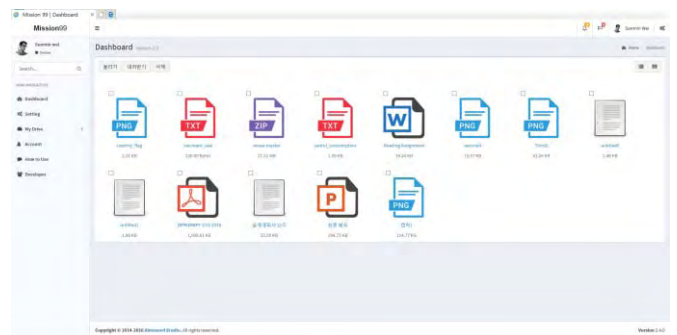


그림 4. 대시보드 페이지 #2. 우측 상단에 버튼을 통해 아이콘 모양의 UI 로 구성 된 대시보드를 제공한다.

3.2 분할 업로드 / 다운로드

분할 업로드 / 다운로드 기능은 클라이언트 쪽에서 파일을 업로드 시키면 MultiPartRequest 패키지를 이용해 서버 쪽에 업로드 된다. 그 후 Zip4J API 를 통해 파일을 업로드에 최적화 된 크기로 분할 시킨 다음 연동 된 클라우드의 액세스 토큰을 받아온 후 분할 된 각각의 파일을 업로드 시킨다. 파일에 대한 정보(id, name)를 Hash 함수를 통해 생성 된 디렉토리 내에 텍스트 파일의 형태로 저장 시킨다. 클라이언트 쪽에서 다운로드를 요청할 경우, 서버 쪽에서 파일에 대한 정보를 가지고 와서 각각의 클라우드 API 를 이용해 서버에 1 차적으로 저장을 시킨다. 그 후 Zip4J API 를 이용해 원본 파일을 생성하고 클라이언트 쪽으로 다운로드 한다.

4. 주요 적용 기술

4.1 Zip4J

Zip4J 는 Apache 라이선스가 적용된 API 로써, 원하는 용량을 Parameter 로 넣어주면 해당 byte 만큼 분할 압축 기능을 제공한다. 사용자가 원하는 파일을 서비스들을 통해 업로드시 웹 JSP 상에 연동된 Zip4J API 를 통하여 패킷 형식으로 자동 분할 압축되어 클라우드 서버로 전송되도록 구현하였고, 다운로드의 경우에도 마찬가지이다. 또한, API 가 구동된 직후, 각기 다른 클라우드 시스템으로 분할 전송된 패킷들의 경로를 저장하는 로그파일을 생성하여 성능 튜닝을 달성하였다

4.2 SHA-1 Hash 함수

Log 파일을 관리하는 알고리즘으로 SHA-1 Hash 함수를 사용하였다. 사용자의 이름과 파일명을 Parameter 값으로 하여 산출된 값을 바탕으로 해당 파일이 저장될 디렉토리를 생성하도록 하였습니다. 산출된 해쉬 값의 앞 두글자는 부모 디렉토리로, 나머지 글자들은 자식 디렉토리로 설정하였습니다.

4.3 Open API

통합 클라우드 관리 서비스에서 가장 중요한 Open API 로는 REST 방식으로 파일을 주고 받는 오픈 소스 API 형식으로 공개된 3개의 클라우드 서비스 (구글의 Google Drive, 드롭박스의 Dropbox, 박스의 Box)를 JSP 기반으로 구현하여, 서비스 상에서 연동될 수 있도록 하였다. 또한, 사용자는 대시보드에서 ‘클라우드 활성화 / 비활성화’, ‘클라우드 추가 / 제외’ 기능을 통하여 손쉬운 클라우드 관리가 가능하게 하였다.



그림 5. Open API 관련 사진.

5. 서비스의 기대 효과 및 활용 분야

5.1 효율적인 저장 공간 이용

독립적 클라우드 서비스들의 한정된 사용 가능 서비스들을 이용하여 마치 하나의 클라우드처럼 사용 가능하도록 하고 이를 통해 단일 클라우드 사용시 낭비되어왔던 애매한 공간들이 하나의 가상 클라우드 상의 용량처럼 사용되어지면서 효율적인 공간 활용이 가능해진다. 또한 해당 서비스에 연동된 모든 클라우드 서비스들은 사용자가 파일 업로드시 항상 동일한 크기의 분할 파일들이 저장되기 때문에 관리 측면에서도 큰 장점을 가진다.

5.2 보안성 강화

독립 클라우드 관리 플랫폼들과는 달리 Zip4j API 와 독자적인 알고리즘을 활용한 파일 분할시스템을 사용함으로써 사용자 데이터에 높은 안정성과 보안성을 보장해준다. 기존 단일 클라우드 사용자들은 모든 데이터가 중앙 집중형 서버에 저장되기 때문에 해커들의 집중 공격 대상이 되기 쉽다. 하지만 이와 다르게 본 프로젝트의 서비스는 분산저장 시스템을 사용하기 때문에 하나의 단일 클라우드가 공격당한다 하더라도 나머지 클라우드에 저장

된 패킷을 얻어내지 못하는 이상 공격자는 원본과 일을 복구해 낼 수 없기 때문에 단일 클라우드가 가지고 있던 취약점을 보완할 수 있게 된다.

5.3 성능 향상

분산파일들의 저장경로가 기록되는 Log 파일과 Hash 함수를 활용하여 만들어진 디렉토리들을 통해 서비스의 성능을 향상시킬 수 있다. 또한 큐 (Queue)형식과 같이 디렉토리 내의 모든 파일들을 검색하는 것이 아니라, 이미 분산 파일들의 Hash 경로가 저장되어 있는 Log 파일을 읽어 오므로써 파일 검색 시간을 줄일 수 있게 된다.

5.4 활용 분야

서비스들을 통한 활용분야는 크게 4 개로 나눌 수 있다. 개인의 중요 파일 백업, 기업의 중요 시스템 및 파일 백업, 기존 클라우드 서비스들을 통합 관리, 각각의 연동 클라우드 및 서비스 가용 공간 시각적으로 확인함으로써 좀더 다양하고 효율적으로 서비스들을 활용할 수 있다.

6. 관련 연구 및 제품

6.1 관련 연구

1. Depsky: Dependable and Secure Storage in a Cloud-of-clouds (2011, Alysson Bessani 외 4 명)

- 다중 클라우드 환경에서의 스토리지 복제 기법
- 현 다중 클라우드 환경의 문제점(신뢰성/ 보안성/ 효율성 부족)을 극복하기 위한 스토리지 복제 알고리즘
- 클라우드 상에 저장되는 파일의 메타데이터를 따로 저장

2. C3Ware: Cloud storage-based Collaborative Middleware (2012, 울산대학교)

- 협업 지원 미들웨어, 울산대학교에서 개발
- 여러 클라우드 스토리지를 백엔드 상에서 연동, 웹서비스 형태로 협업 서비스 제공

3. Erasure codes for storage systems a brief primer (James S. Plank, 2013.12)

- 분산 저장되는 데이터 세그먼트들의 데이터를 인코딩하여, 복구할 수 있는 Erasure Code, 즉 Parity 를 생성.
- 원본 데이터의 손실뿐만아니라, Parity 간의 손실도 복구 가능
- 분할 저장되는 디스크(스토리지) 개수에 상관없이 동일한 복구성능 보장
- 복구 정도의 차이는 없지만, 유연성 측면에서는 디스크가 많을수록 좋다.

4. Erasure Coding in Windows Azure Storage(Cheng Huang 외 7 명, 2012)

- 보다 발전된 Erasure Coding 기법인 Windows Azure

Storage 만의 독자적인 LRC(Local Reconstruction Code) 기법을 사용

- 복구를 위한 Parity 파일들을 다시 두 그룹(Local Parity & Global Parity)로 다시 한번 분산저장
- 일반적인 Erasure Coding 보다 적은 비용, 높은 Fault Tolerance, 빠른 성능을 보장

6.2 관련 제품 및 서비스

1) MultCloud

여러 클라우드 저장소 서비스를 한 장소에서 관리 해주고 각 서비스가 지원하는 기능은 모두 제공해 주고 이미지 파일과 같은 지원 대상 파일에 대해 미리보기 기능을 제공해준다.



그림 6. 관련 서비스 - MultCloud

2) 토스트 드라이브

8 월에 출시한 NHN 엔터테인먼트에서 출시한 어플리케이션으로 클라우드 서비스, 휴대폰, 윈도우 PC 와 무선 연결을 해서 파일 관리를 해주는 어플리케이션이다.

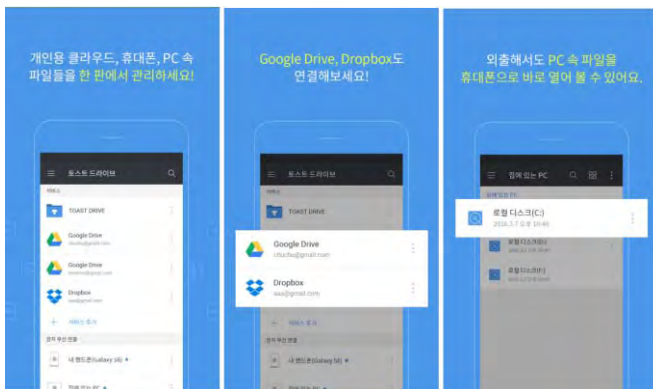


그림 7. 관련 제품 - 토스트 드라이브

중요한 요소라고 할 수 있다. 점점 데이터가 쌓이는 속도도 빨라지고 관리의 중요성이 대두되고 있다. 현재 많은 사용자들이 Google, Dropbox, 네이버, Box 등의 기업들이 제공해주는 무료 클라우드 저장 서비스를 이용하고 있다. 허나 무료인 만큼 제공해주는 드라이브의 용량이 적고 단일 클라우드 시스템의 ‘중앙 집중형 서버 저장방식’에 의한 개별적으로 관리를 해야 한다는 점에서 단점을 드러낸다. 이러한 단점에 대한 해결책으로 통합 클라우드 서비스를 제시할 수 있다. 통합 클라우드 서비스의 분산 저장 기능을 통해 확장성 및 유연성 측면에서 기존 서비스의 개별 관리에 따른 어려움을 해소할 수 있다. 또한 이 기능을 통해 클라우드에 업로드된 파일에 대해 보안성이 강화되고 파일을 분산 업로드 함으로서 업로드 시간을 줄여 효율적인 클라우드 관리가 된다고 할 수 있다. 또한 기재한 3 가지의 클라우드 뿐만 아니라 다른 무료 클라우드를 언제든지 연동 시킬 수 있는 플랫폼을 구성하여 접근성을 높였다. 이러한 기대 효과 기능은 기존의 단일 클라우드를 사용하면서 적은 용량과 관리에 불편함을 느낀 사용자들에게 해결책을 제시해준다.

참고 문헌

- [1] Alysso Bessani, Miguel Correia, Bruno Quaresma, Fernando Andre, Paulo Sousa, "DEPSKY : Depenable and Secure Storage in a Cloud-of-Clouds", University of Lisbon, Faculty of Sciences, Portugal, 2013.
- [2] 안형배, "다중 클라우드 스토리지의 효과적인 활용을 위한 스토리지 복제 기법 및 응용", 울산대학교 대학원, 2013
- [3] James S. Plank, (2013), "Erasure Codes for Storage Systems – A Brief Primer", ;login;, SYSADMIN ,38(6), 44-50
- [4] Cheng Huang, Huseyin Simitci, Yikang Xu, Aaron Ogus, Brad Calder, Parikshit Gopalan, Jin Li, and Sergey Yekhanin, "Erasure Coding in Windows Azure Storage", Microsoft Corporation, USENIX ATC, Boston, MA, 2012

7. 결론

우리는 데이터의 시대에 살고 있다고 해도 무방하다. 그만큼 데이터의 보안과 관리 및 확장성이