Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.31, NO.3, Jun. 2021 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2021.31.3.443

엣지 컴퓨팅 환경에서 추적 데이터 서버를 통한 데이터 추적*

임 한 울,[1†] 변 원 준,[1] 윤 주 범[2‡]
[1,2]세종대학교 (대학원생, 교수)

Tracking Data through Tracking Data Server in Edge Computing*

Han-wool Lim,[1†] Won-jun Byoun,[1] Joobeom Yun[2‡]
[1,2]Sejong University (Graduate student, Professor)

요 약

엣지 컴퓨팅(Edge Computing)의 핵심 기술 중 하나는 사용자의 움직임에 따라서 엣지 서버간에 데이터를 이동시켜 항상 사용자와 가까운 거리에서 서비스를 제공한다는 점이다. 그만큼 엣지 서버간의 데이터의 이동이 빈번하다. IoT 기술이 발전하고 사용영역이 확대됨에 따라 생성되는 데이터 또한 증가하기 때문에 각 데이터를 정확하게 추적하고 처리할 수 있는 기술이 필요하다. 개인정보와 같은 민감한 정보들에 대해서는 더욱 그러하다. 현재 클라우드 시스템 안에서 데이터들의 이동 및 유통에 대한 추적과 추적 기술에 기반한 데이터의 폐기 기술이 존재하지 않아 엣지 컴퓨팅 서비스의 사용자는 해당 데이터가 현재 어떤 곳에 위치하는지, 사용자가 데이터의 삭제를 요청할 경우 클라우드 시스템 내에서도 데이터가 제대로 제거되어 있는지 등을 확인할 수 없다. 본 논문에서는 엣지 컴퓨팅환경에서 각 엣지 서버와 중앙 클라우드에 저장되는 데이터들에 대해 데이터의 이동과 유통에 대한 추적 데이터를 생성, 관리하는 추적 데이터 서버를 구축하여 엣지 컴퓨팅환경에서 저장된 모든 데이터의 흐름을 정확하게 추적할 수 있는 기술과 추적 데이터를 활용하여 사용자의 움직임에 따라서 엣지 서버간의 이동하는 로컬 데이터와 분산 파일시스템에 저장된 데이터들을 정확하게 추적하고 이를 활용하여 데이터를 완벽하게 제거하는 기술을 제안한다.

ABSTRACT

One of the key technologies in edge computing is that it always provides services close to the user by moving data between edge servers according to the user's movements. As such, the movement of data between edge servers is frequent. As IoT technology advances and usage areas expand, the data generated also increases, requiring technology to accurately track and process each data to properly manage the data present in the edge computing environment. Currently, cloud systems do not have data disposal technology based on tracking technology for data movement and distribution in their environment, so users cannot see where it is now, whether it is properly removed or not left in the cloud system if users request it to be deleted. In this paper, we propose a tracking data server to create and manage the movement and distribution of data for each edge server and data stored in the central cloud in an edge computing environment.

Keywords: Edge Computing, IoT, Data Tracking, Distributed File System, Data Deletion

Received(02. 16. 2021), Modified(04. 12. 2021), Accepted(05. 04. 2021)

* 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터육성지원사업의 연구결과로 수행되었음 (IITP-2021-2018-0-01423)

† 주저자, nurilmaro@gmail.com
‡ 교신저자, jbyun@sejong.ac.kr(Corresponding author)

I. 서 론

오늘날 스마트폰, 사물 인터넷(Internet of Things, 이하 IoT) 기술의 발전에 따라 IoT 기기는 특정 공간이나 분야에서만 사용되는 것이 아닌 일상생활의 모든 공간에서 그리고 모든 사물 들이 인터넷에 연결되어 서로 통신하는 시대가 되었다. 5G 기술이 개발됨에 따라 IoT 기술의 발전과 영역의 확대는 더욱 가속화되고 있다. 하지만 현재 IoT 기기는 자체적인 특성으로 인해 클라이언트가 요구하는 기능을 전부 수행하기 위한 충분한 자원을 IoT 기기 내에 보유하고 있지 않는다는 한계가 존재한다. 이러한 IoT의 한계를 극복하기 위해 엣지 컴퓨팅(Edge Computing) 기술을 활용한다[1]-[3].

IoT 기술이 발전되고 사용영역이 확대됨에 따라서 클라우드에 연결되는 IoT 기기의 수와 각 서버가 처리하고 저장되는 데이터의 양 또한 그만큼 증가한다. 테 반셍 씨게이트 글로벌세일즈 수석 부사장은 2025년에 전 세계적으로 클라우드에 생성될 데이터의 양이 약 175ZB까지 이를 것으로 바라보고 있다[10]. 또한, 엣지 컴퓨팅기술의 중요한 특성 중 하나로 엣지 서버간의 데이터를 이동하는 기술인 서비스 마이그레이션(Service Migration)을 통하여 사용자에게 가장 최적의 거리에서 서비스를 제공하기 위해 사용자의 이동에 따라 데이터들이 여러 엣지 서버로 빈번하게 복사되거나 이동될 수 있다. 하지만 서비스를 이용하는 사용자는 현재 자신의 데이터가 어떤 엣지 서버에 위치하는지 엣지 서버에 데이터가 완벽하게 제거되었는지를 확인할 수 없다. 그렇기에 클라우드 환경에서 데이터들을 제대로 관리하고 활용하기 위해 각각의 데이터들을 정확하게 추적하고 올바르게 처리할 수 있는 기술이 필요하다. 특히, 개인정보, 의료정보와 같은 민감한 데이터들을 클라우드 환경에서 활용하기 위해 클라이언트의 요청에 따라 엣지 컴퓨팅환경에 존재하는 모든 데이터를 추적하여 완전히 폐기하는 것을 통해 클라이언트에게 해당 서비스에 대한 신뢰성을 주는 것이 더욱 필요하다. 본 논문에서는 엣지 컴퓨팅환경에서 저장, 복사, 이동되는 모든 데이터에 대한 추적 데이터를 생성하고 이를 관리하는 추적 데이터 서버를 구축하여 엣지 컴퓨팅 환경에 저장된 데이터들을 추적할 수 있는 기술과 추적 데이터 서버에 저장된 추적 데이터들을 활용하여 엣지 서버에 저장된 데이터의 흐름을 추적하고 이들을 확실히 제거하는 기술을 제안한다.

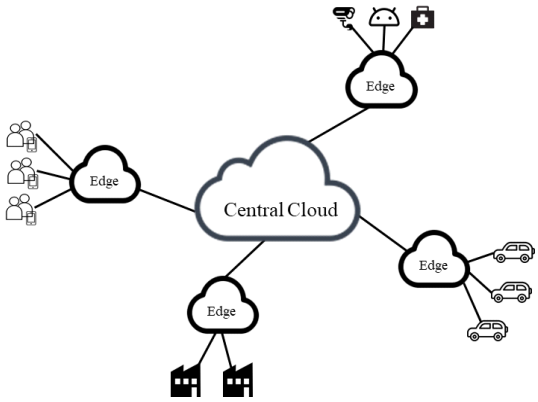


Fig. 1. Edge Computing

본 논문은 2장에서 데이터 추적과 관련하여 기존에 존재한 관련 연구와 본 논문에서 제시하는 추적 모델의 방향성을 비교하고, 3장에서 추적 데이터를 추적 데이터 서버에 저장하고 추적 데이터를 활용하여 데이터를 삭제하는 기법과 과정을 설명하고, 4장에서 구현의 결과와 제안한 기법들로 인해 발생하는 오버헤드와 효율성을 평가하고 끝으로 본 논문에 대한 결론으로 마무리한다.

II. 관련 연구

2.1 클라우드 데이터 추적기 (Cloud Data Tracker)

클라우드 데이터 추적기는 클라우드 시장이 확대됨에 따라서 클라우드 시스템의 보안을 강화하기 위해 클라우드 시스템에서 데이터 접근, 수정 등의 모든 데이터 관련 이벤트를 로깅하고 감시하기 위한 클라우드 보안 관련 대응 프레임워크다[12]. 클라우드 데이터 추적기는 클라우드 시스템 내 존재하는 데이터와 관련한 모든 이벤트를 로깅하고 감시함으로써 사고 발생 시 해커의 공격 경로와 사고의 책임자 등을 식별하는 것을 주된 목적으로 한다. 로그를 통해 사고의 발생 경위, IP, MAC 등을 파악하여 데이터 접근시간, 데이터에 접근한 클라이언트 등을 파악할 수 있다. 또한, 저장한 로그들을 시각적으로 표현할 수 있도록 인터페이스를 구축하여 이를 통해 더욱 직관적으로 추적할 수 있다.

하지만, 해당 논문에서 제안하는 추적 모델은 실시간으로 데이터를 추적하는 예방적 관점이 아닌 사고가 발생한 이후에 담당자가 기록된 로그들을 확인하여 대응하는 사후대응적 관점을 우선하여 고려하였

다. 또한, 공격 경로의 파악, 책임자 식별과 같은 작업은 어디까지나 담당자에 의해 수동적으로 분석된다. 이에 반해, 본 논문은 보안과 사후대처를 목적으로 이벤트를 로깅하는 방식이 아닌 완전폐기를 위해 추적 데이터를 저장하는 추적 데이터 서버를 별도로 구축하고 이를 통해서 사용자의 요청에 따라 실시간으로 추적 데이터를 통해 데이터의 위치를 파악하여 분산 파일시스템과 로컬로 저장된 데이터를 보유한 엣지 서버에 접근하여 데이터 폐기작업을 진행하는 것을 주된 목적으로 한다.

2.2 엣지 체인(EdgeChain)

엣지 체인은 블록체인 기술을 기반으로 엣지 컴퓨팅을 구축하는 프레임워크로서 IoT 기기의 보안 취약점과 확장성 문제를 극복하기 위해서 블록체인 기술을 적용한 엣지 컴퓨팅 기술이다[11]. 블록체인 기술을 활용하면 데이터의 소유, 이동 흐름 등과 같은 데이터의 추적도 원활하게 이루어질 수 있다. 하지만 블록체인 기술은 보안성과 같은 측면에서는 우수하지만 여러 노드에서 같은 작업들이 중복되어 이루어지기 때문에 효율성이 극도로 떨어진다는 단점이 존재한다. 또한, 데이터의 갱신과 저장에서 많은 자원과 시간이 소요되기 때문에 주로 실시간 작업을 수행하는 IoT 기기와 엣지 컴퓨팅 서비스에는 적합하지 않다. 본 논문은 이러한 블록체인 기반 방식의 단점으로 인해 블록체인 방식을 선택하지 않고 별도의 추적 서버를 구축하는 방식을 사용하여 엣지 컴퓨팅환경의 데이터에 추적성을 제공하였다.

2.3 분산형 동적 정보 흐름 추적기 (DDIFT, Decentralized Dynamic Information Flow Tracking)

분산형 동적 데이터 흐름 추적기(Decentralized Dynamic Information Flow Tracking, 이하 DDIFT)는 IoT의 애플릿 대상의 취약점에 대한 공격을 탐지하기 위해 기존 동적 데이터 흐름 추적 모델(Dynamic Information Flow Tracking, DIFT)을 개선한 모델이다[13]. 앞에서 언급한 클라우드 데이터 추적기와 마찬가지로 민감한 데이터의 노출, 삭제와 같은 위협으로부터 데이터를 보호하기 위함이 목적이다. 데이터에 다양한 타입의 태그를 생성하여 데이터의 무결성과 데이터의 흐름을 저장한

Table 1. Comparing Cloud Data Tracker and Tracking Server Method

Item	Cloud Data Tracker	Tracking Server
Proposed method	Log all events related to data in the cloud	Generates tracking data for data and stores it on a separate tracking data server
Approach	User interprets logs with Reactive approach	Server removes data with Real-time approach
Main Purpose	Security and follow-up	Real-time data tracking
Detailed Purpose	Identify attack paths and identify the person responsible for the damage	Complete data retirement with real-time data tracking
Major Differences	Logging is performed automatically, but post-response through logging is passive	Automatically extract/save trace data and work with trace data

Table 2. Comparing BlockChain and Tracking Server

Item	BlockChain	Tracking Server
Concept	Systems with multiple servers duplicating operations to increase reliability	Systems that handle all trace-related operations in Tracking Server
Pros	Maintains data integrity and reliability of job results through the same operations on each node	Easy to add and remove devices and no waste of storage space
cons	Takes a lot of resources, time, and less efficient	Increased security threats and difficulty in dealing with errors

다. DDIFT에서 제안하는 방식 중 다양한 타입의 태그 데이터를 생성한다는 점과 본 논문에서 제시하는 추적 데이터를 생성한다는 점은 유사하지만, DDIFT의 경우 IoT의 실행 중인 모든 애플릿이 대상이고 보안과 이후에 포렌식을 주된 목적인 부분에서 엣지 컴퓨팅의 전체에 저장된 데이터를 대상으로 완전폐기와 같은 작업을 위해 추적 데이터를 저장하고 활용하는 본 연구와는 방향의 차이가 존재한다.

III. 시스템 설계

본 논문에서는 엣지 컴퓨팅환경에서 저장되고 이동하는 모든 데이터의 위치를 실시간적으로 추적하기 위한 추적 데이터를 저장하는 추적 데이터 서버와 저장된 추적 데이터를 통해서 엣지 컴퓨팅환경에 존재하는 데이터들을 완벽하게 제거하는 방법을 제시한다. 저장된 추적 데이터는 후에 관리자나 클라이언트가 추적 데이터 서버에게 요청하여 요청을 받은 추적 데이터 서버가 현재 데이터가 저장된 엣지 서버와 분산 파일시스템의 위치를 파악한 다음 데이터를 완전폐기하는 작업을 수행한다.

3.1 로컬 파일에 대한 추적 시스템

추적 데이터를 저장하는 엣지 서버의 데이터 전체적인 방식은 Fig. 2와 같다. 저장되는 데이터의 유형은 크게 두 가지로 구분한다. 엣지 서버와 중앙 클라우드 서버에 로컬로 저장되는 데이터에 대한 추적 데이터를 저장하는 방식과 클라우드에 추가적으로 연결된 분산 파일시스템에 저장되는 데이터에 대한 추적 데이터를 저장하는 방식으로 구분된다.

엣지 서버와 중앙 클라우드 서버에 로컬로 저장되는 데이터의 경우 각 서버의 파일시스템에 변화가 발생하면 해당 데이터에 대한 추적 데이터를 추적 데이터 서버로 전송한다. 파일시스템의 변화를 감지하면서 데이터의 생성과 서비스 마이그레이션으로 인한 데이터의 복사, 이동에서도 데이터를 추적한다. 추적 데이터는 추적 데이터 서버의 Trace 테이블의 항목에 해당하는 데이터들을 전송한다. Trace 테이블 항목에 해당하는 데이터들은 Fig. 6와 같다. 데이터의 이름, 해당하는 엣지 서버의 아이디 등의 데이터를 전송한다. 엣지 서버의 아이디는 사전에 지정한다. 해당 데이터베이스의 Edge라는 테이블에 각 엣지 서버의 고유 아이디와 IP주소의 매핑 정보가 포함된다. 해당 매핑 정보를 기반으로 데이터 삭제 시 해당 엣지 서버에 삭제를 요청한다.

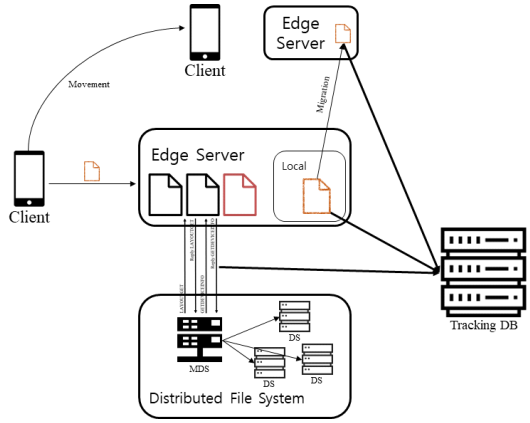


Fig. 2. System Design (Full System)

3.2 분산 파일시스템에 대한 추적 시스템

본 논문에서 사용하는 분산 파일시스템은 POSIX 표준을 준수하여 애플리케이션에서 분산 파일시스템을 로컬 파일시스템처럼 사용 가능한 병렬 네트워크 파일시스템(Parallel Network File System, 이하 pNFS)를 사용한다. pNFS의 동작 과정은 Fig. 3과 같다. 우선, 메타데이터 서버는 데이터 서버와 연결한다. 그리고 클라이언트는 메타데이터 서버에 연결한다. 이렇게 연결함으로 메타데이터 서버가 중간에서 연결된 클라이언트와 데이터 서버를 매칭시키는 것이다. 처음에 연결된 클라이언트는 메타데이터 서버와 LAYOUTGET 패킷을 주고받아 데이터 서버의 아이디와 같은 데이터 서버의 정보, 데이터가 저장된 패턴 등을 얻을 수 있다. LAYOUTGET 패

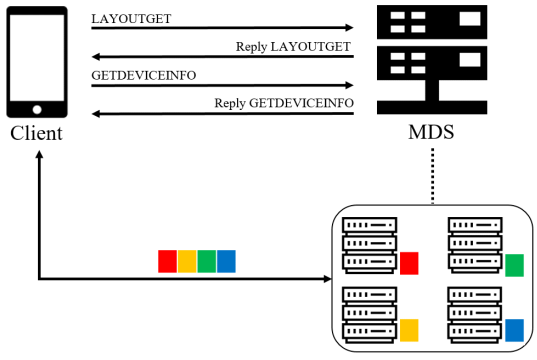


Fig. 3. pNFS(Parallel Network File System)

킷을 통해 얻은 데이터들을 토대로 다시 한번 클라이언트와 메타데이터 서버가 GETDEVICEINFO 패킷을 주고받아 데이터 서버의 실제 위치를 얻어서 이를 기반으로 클라이언트는 데이터 서버와 직접 통신하여 데이터 입출력을 수행한다[6]-[9].

본 논문에서 제시하는 클라우드와 연결된 분산 파일시스템의 데이터 저장에 대한 추적 시스템의 디자인은 Fig. 4와 같다. pNFS의 메타데이터 서버의 코드를 수정하여 클라이언트와 메타데이터 서버간의 OPEN 패킷을 통해 통신을 구축할 때 메타데이터 서버는 바로 추적 데이터 서버의 데이터베이스의 Trace 테이블과 원격으로 통신을 구축한다. 본 논문에서 진행한 실험은 엣지 서버, pNFS의 구성요소와 추적 데이터 서버는 서로 신뢰할 수 있다고 가정한 상태로 메타데이터 서버와 엣지 서버만 추적 데이터 서버의 Trace 테이블에만 접근하도록 제한함으로써 무결성을 보장한다.

메타데이터 서버와 클라이언트 간의 주고받는 LAYOUTGET 패킷에 대한 응답을 보낼 때 동시에 메타데이터 서버는 추적 데이터 서버에 저장되는 파일의 이름, 파일이 저장되는 데이터 서버의 주소 등과 같은 정보를 Trace 테이블에 전송한다. Trace 테이블에 전송하는 내용은 Fig. 6과 같다. 데이터의 수정이 발생하면 클라이언트가 수정된 파일을 저장할 때 메타데이터 서버에 대해 파일이 수정됨을 알려주는 LAYOUTCOMMIT 패킷을 통해서 추적 데이터를 갱신한다. 전송 내용은 데이터 생성 때 추적 데이터 서버에 보내는 데이터와 같다. 이후, 클라이언트가 데이터 서버와의 입출력을 마치고 통신을 종료하면 메타데이터 서버와 추적 데이터 서버와의 연결도 종료한다. 추적 데이터 서버와의 연결과 데이터 전송을 구분하는 이유는 LAYOUTGET 패킷을 주고받는 과정에서 추적 데이터 서버와의 세션을 열고, 닫는 작업을 같이 수행하는 경우 클라이언트가 데이터 서버에 데이터를 생성하고 접근할 때마다 LAYOUTGET 패킷을 통해 파일 핸들과 저장 패턴 등을 보내기 때문에 그만큼 메타데이터 서버는 데이터베이스와 연결을 생성하고 닫는 작업을 반복하는 과정에서 추가적인 오버헤드가 발생하기 때문이다.

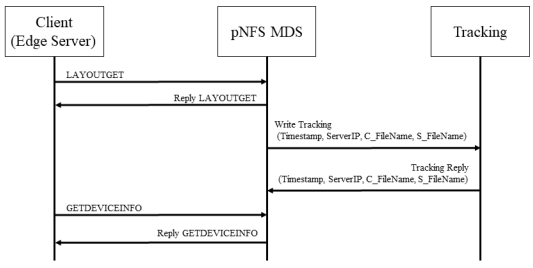


Fig. 4. System Design (Distributed File System)

3.3 데이터 완전 폐기

본 논문은 클라우드 컴퓨팅 서비스에 대한 사용자의 신뢰를 위해 저장된 추적 데이터를 사용하여 사용자가 원하는 시점에 실시간으로 데이터의 위치를 추적하여 클라우드 컴퓨팅 서비스 내에 존재하는 모든 데이터를 제거하는 방식을 제안한다. 데이터를 삭제하는 전체적인 흐름은 Fig. 5와 같다. 클라이언트로부터 데이터 삭제 요청을 받은 추적 데이터 서버는 먼저 pNFS 클라이언트로서 직접 접근한다. 추적 데이터 서버는 사전에 메타데이터 서버에서 추적 데이터 서버에 엣지 서버들보다 더 상위의 접근 권한을 가지도록 설정하였다. 추적 데이터 서버는 자신이 보유하고 있는 추적 데이터를 통해서 메타데이터 서버와 데이터 서버에 접근하여 저장된 데이터를 삭제한다. 다음으로 엣지 서버에 로컬로 저장된 데이터를 제거하기 위해서 추적 데이터 서버는 저장된 추적 데이터를 통해서 해당 데이터를 가지고 있는 그리고 데이터를 가지고 있었던 엣지 서버에 접근하여 데이터 삭제를 요청한다. 해당 데이터에 대한 삭제를 완료한 엣지 서버들은 추적 데이터 서버에 삭제가 완료되었음을 보고한다. 모든 엣지 서버로부터 완료 보고를 받은 추적 데이터 서버는 마지막으로 자신의 데이터베이스에 저장된 데이터를 삭제한 다음 처음에 데이터의 삭제를 요청한 클라이언트에게 삭제가 완료되었음을 보고한다.

IV. 구현 및 분석

본 장에서는 3장에서 제시한 설계를 구현한 추적 시스템과 추적 시스템을 활용한 완전폐기의 구현결과와 추적 시스템을 사용하여 사용자가 저장된 파일을 완전폐기하는 과정과 별도의 추적 데이터 서버를 구축하지 않은 엣지 컴퓨팅환경과 분산 파일시스템과의 오버헤드를 비교하여 해당 시스템의 추적 데이터 서버와의 추가적인 통신으로 인해 발생하는 오버헤드를 비교하는 실험을 진행하였다.

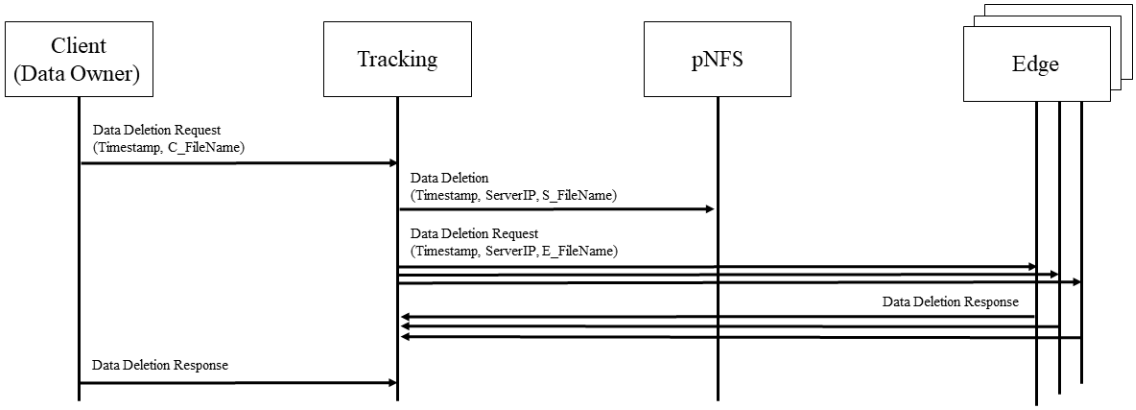


Fig. 5. System Design (Data Deletion)

4.1 실험 환경

본 논문의 실험을 진행한 환경은 CPU는 인텔 i7-10700k 3.80GHz이고 메모리는 1GB, 하드디스크는 80GB이다. 운영체제는 CentOS를 사용하였다. 서버와 클라이언트, 분산 파일시스템의 메타데이터 서버, 데이터 서버 모두 같은 환경에서 진행되었다. 추적 데이터 서버의 경우 CPU는 인텔 i7-10700k 3.80GHz이고 메모리는 4GB, 하드디스크는 80GB이다. 운영체제는 우분투 16.04를 사용하였다. 또한, MYSQL 데이터베이스를 사용하여 구축하였다.

4.2 추적 데이터 저장 구현결과

Fig. 6는 본 논문의 내용을 구현하여 생성된 데이터에 대한 추적 데이터를 추적 데이터 서버에 저장한 결과다. Fig. 6의 윗부분은 클라우드 컴퓨팅 환경에서 데이터가 생성되었을 때 저장되는 추적 데이터를 보여준다. 클라이언트에 저장되는 파일의 이름, pNFS의 데이터 서버에 저장되는 파일의 이름, 엣지 서버에 저장되는 파일의 이름, 저장된 엣지 서버의 위치, 생성날짜, 마지막 수정날짜로 구성되어있다. 파일 생성 시 분산 파일시스템에 생성된 파일과 엣지 서버에 로컬로 생성된 파일을 구분하기 위해 데이터베이스의 항목들을 구분한다. Fig. 6의 아랫부분은 데이터가 삭제되거나 다른 엣지 서버로 이동되었을 때 추적 데이터 서버에 적용된 것을 보여준다. 갱신, 이동, 복사, 제거와 같은 데이터의 변화가 발생하였을 때 추적 데이터 서버에서 해당 데이터에 대한 추적 데이터도 자동으로 적용된다. 예를 들어, 엣지 서버나 중앙 클라우드에 로컬로 저장된 데이터가 다른 위치로 이동하거나 데이터를 수정할 경우 엣지 서버의 아이디와 마지막 수정날짜를 갱신한다. Fig. 6에서는 test-nuri2 파일이 저장된 엣지 서버의 위치가 변경됨에 따라 엣지 서버와 수정날짜가 변경됨을 확인할 수 있다.

데이터베이스에 대한 항목은 현재 프로토타입이라 생성된 파일의 이름, 저장된 위치, 최초 파일 생성날짜와 마지막 갱신날짜에 대한 데이터를 저장하였지만, 후에 서비스 마이그레이션과 같은 전체적인 엣지 컴퓨팅 환경과 기능의 구현, 데이터에 접근할 클라이언트의 수와 유형의 증가, 분산 파일시스템의 메타데이터 서버/데이터 서버의 증가와 같은 이유로 추적 데이터에 다른 항목들이 추가되거나 변경될 수 있다. 후에 해당 데이터들을 사용하여 서버에 접근해 완전폐기와 같은 작업을 수행한다.

4.3 데이터 제거 구현결과

본 논문은 엣지 컴퓨팅환경에서 추적 데이터를 저장하는 방법뿐 아니라 추적 데이터를 활용하여 데이터를 완전폐기하는 방식까지 제안한다. Fig. 7은 Fig. 6의 윗부분에서 존재한 test-nuri5를 사용자로부터 삭제 요청을 받은 추적 데이터 서버의 화면이다. 저장된 데이터가 엣지 서버에 로컬로 저장되어있는지, pNFS로 저장되어있는지와 저장되어있는 곳의 파일명을 표시한다. pNFS에 저장된 데이터임을 확인한 추적 데이터 서버는 저장된 추적 데이터를 통해서 pNFS에 접근하여 저장된 데이터를 제거한다.

ID	name_C	name_S	name_E	DS_IP	Edge_ID	First_Reg	Last_Up
1	test-nuri	1835011.1566851269	NULL	192.168.163.158	NULL	2021-02-09 21:01:36	2021-02-09 21:01:36
2	test-nuri2	NULL	test-nuri2	NULL	001	2021-02-09 21:08:22	2021-02-09 21:08:22
3	test-nuri3	NULL	test-nuri3	NULL	002	2021-02-09 21:08:58	2021-02-09 21:08:58
4	test-nuri4	1835012.1566851276	NULL	192.168.163.158	NULL	2021-02-09 21:09:50	2021-02-09 21:09:50
5	test-nuri5	1835013.1566851277	NULL	192.168.163.158	NULL	2021-02-09 21:10:15	2021-02-09 21:10:15

ID	name_C	name_S	name_E	DS_IP	Edge_ID	First_Reg	Last_Up
1	test-nuri	1835011.1566851269	NULL	192.168.163.158	NULL	2021-02-09 21:01:36	2021-02-09 21:01:36
2	test-nuri2	NULL	test-nuri2	NULL	002	2021-02-09 21:08:22	2021-02-09 21:14:36
3	test-nuri3	NULL	test-nuri3	NULL	002	2021-02-09 21:08:58	2021-02-09 21:08:58
4	test-nuri4	1835012.1566851276	NULL	192.168.163.158	NULL	2021-02-09 21:09:50	2021-02-09 21:09:50

Fig. 6. Tracking System Implementation

제거가 완료되면 삭제가 완료되었다고 클라이언트에게 보고한다. 추적 데이터 서버에서도 확인을 위해 삭제되었다고 출력하였다.

```
test-nuri5 is stored as 1835013.1566851277 in pNFS
test-nuri5 is deleting

test-nuri5 stored in pNFS was deleted
```

Fig. 7. Data Deletion Implementation

4.4 추적 시스템 오버헤드 측정

3장에서 제시한 설계를 적용함으로써 엣지 컴퓨팅 환경에 데이터의 추적성을 제공할 수 있지만, 기존의 엣지 컴퓨팅환경에서 수행하는 통신 외 별도로 추적 데이터 서버와 추가적인 통신을 수행하기 때문에 그만큼 오버헤드가 발생한다. 본 절에서는 본 논문의 핵심기능을 구현하여 발생하는 오버헤드를 측정하여 본 논문의 핵심 기술의 실현 가능성을 설명한다. Table. 3과 Table. 4, Fig. 8과 Fig. 9는 각각 파일 생성횟수, 그리고 생성되는 파일의 크기에 따른 오버헤드에 대한 그래프이다. Table. 3과 Fig. 8은 파일 생성횟수에 따른 오버헤드, 그리고 Table. 4와 Fig. 9는 파일 크기에 따른 오버헤드에 대한 그래프이다. 두 가지 그래프 모두 생성횟수나 크기가 증가할수록 조금씩 오버헤드가 더 발생하지만 차이는 미미하다. 본 논문에서 제안한 모델의 경우 기존의 환경에서 소요된 시간보다 전체적으로 5초 이내로 증

Table 3. Overhead based on the number of file generations

number of times	Normal	Tracking	Overhead
100 times	3.879	6.286	2.407
1000 times	51.3	53.905	2.605

Table 4. Overhead based on file generation size

Size	Normal	Tracking	Overhead
10KB	5.952	8.233	2.281
1MB	51.3	53.905	2.605
5MB	84.988	88.089	3.101

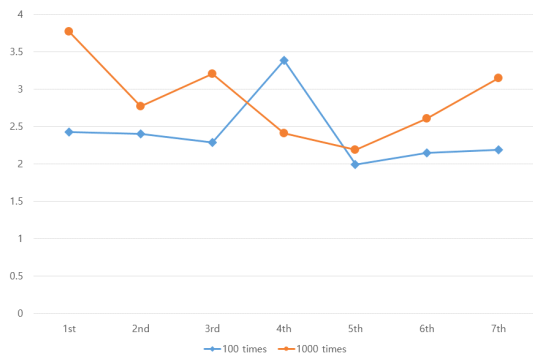


Fig. 8. Overhead based on the number of file generations

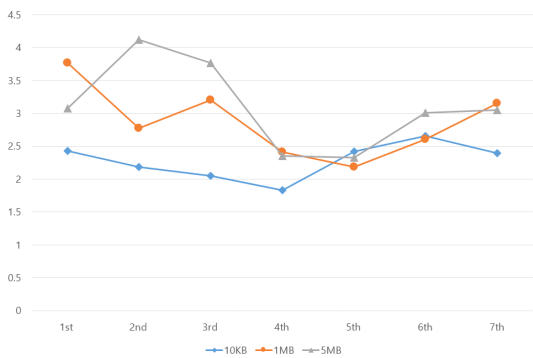


Fig. 9. Overhead based on file generation size

가하였음을 보여준다.

측정한 오버헤드는 변동될 변수들이 아직 존재한다. 이번 실험에 사용한 장비의 사양들보다 더욱 좋은 사양을 가진 장비들로 엣지 컴퓨팅환경을 구축하거나 vNFS와 같은 기존 NFS의 성능을 개선한 다른 분산 파일시스템을 사용할 경우 오버헤드가 감소할 수 있다는 긍정적인 부분이 존재한다.

V. 결 론

본 연구에서는 엣지 컴퓨팅환경에서 데이터를 저장하고 사용할 때 저장되는 모든 데이터를 추적하기 위한 엣지 컴퓨팅환경의 추적 데이터 서버 모델을 제안하였고 구현결과와 추적 데이터 저장으로 인해 발생하는 오버헤드를 측정하였다. 제안한 모델은 추적 데이터 서버를 사용하여 엣지 서버와 클라우드 서버에 로컬로 저장되는 데이터의 경우 각 서버에 파일시스템의 변화를 감지하여 파일의 추가가 발생하면 해당 데이터의 추적 데이터를 전송하고 파일을 제거하거나 각 엣지 서버 간 서비스 마이그레이션을 진행하면 추적 데이터 서버와 통신하여 해당 데이터를 제거/갱신하는 방법을 제시하였다. 엣지 서버와 중앙 클라우드 데이터의 대부분을 저장하는 분산 파일시스템의 경우 본 연구에서는 클라이언트와 메타데이터 서버 간 패킷의 내용을 추출하여 이를 기반으로 추적 데이터를 생성하고 저장하는 방법을 제시하였다. 이와 같은 모델을 통해서 통신에 대한 오버헤드가 최대 5초까지만 증가함을 확인하였다. 이와 같은 모델을 통해서 향후 엣지 컴퓨팅환경에서 데이터의 완전폐기와 같은 작업을 추가로 구현할 계획이다. 끝으로 본 논문에서 설명한 과정에서 보안적인 부분과 다른 프로그램들과의 호환성 부분을 더 연구하면 성능이 더욱 개선될 것으로 기대된다.

References

[1] Hyuk-joon Seo. “IoT security trends and IoT security technologies at home and abroad.” Weekly technology trends, pp. 2-14, Aug. 2017.

[2] Weisong Shi, Jie. Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. "Edge computing: Vision and challenges." IEEE IoT Journal, vol. 3, no. 5, pp. 637 - 646, Oct. 2016.

[3] Wazir Zada Khan, Ejaz Ahmed, Saqib Hakak, Ibrar Yaqoob and Arif Ahmed. “Edge computing: A survey” Future Generation Computer System, 97(2019), pp. 219-235, Aug. 2019.

[4] Sang-chul Moon, Hyung-joon Kim. “Distributed data management systems and data service technologies for cloud computing” KIPS Transactions on Software and Data Engineering, pp. 51-63, Mar. 2009.

[5] Young-soo Min, Ki-sung Jin, Hong-yeon Kim, Young-kyun Kim. “Distributed File System Technology Trends for Cloud Computing” Electronic communication trend analysis, Vol. 24, No. 4, pp. 55-68, Aug. 2009.

[6] Jung-sook Park, Soo-young Kim, Myung-hoon Cha, Dong-oh Kim, Young-chang Kim, Hong-yeon Kim. “PNFS Standardization and R&D Trends” Electronic communication trend analysis, Vol. 26, No. 5, pp. 55-65, Oct. 2011.

[7] RFC 5661 : Network File System Version 4 Minor Version 1 Protocol. Accessed: Sep 20, 2020 [Online]. Available : https://tools.ietf.org/html/rfc5661

[8] RFC 5663 : Parallel NFS (pNFS) Block/Volume Layout. Accessed: Sep 20, 2020 [Online]. Available : https://tools.ietf.org/html/rfc5663

[9] RFC 5664 : Object-Based Parallel NFS (pNFS) Operations. Accessed: Sep 20. 2020 [Online]. Available : https://tools.ietf.org/html/rfc5664

[10] World Data Generation 175ZB in 2025, Edge computing is a Big Opportunities. Accessed: Jan 12, 2021 [Online]. Available : https://zdnet.co.kr/view/?no=20190215094821

[11] Jianli Pan, Jianyu Wang, Austin Hester, Ismail Alqerm, Yuanni Liu and Ying Zhao, "EdgeChain: An Edge-IoT Framework and Prototype Based on Blockchain and Smart Contracts", IEEE IoT Journal, vol. 6, No. 3, pp. 4719-4732, Jun. 2019.

[12] Yu Shyang Tan, Ryan K. L. ko, Peter Jagadpramana, Chun Hui Suen, Markus Kirchberg, Teck Hooi Lim, Bu Sung Lee, Anurag Singla, Ken Mermoud, Doron Keller, Ha Duc, "Tracking of Data Leaving the Cloud", IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, pp. 137 - 144, Jun. 2012.

[13] Nikolaos Sapountzis, Ruimin Sun, Daniela Oliveira, "DDIFT: Decentralized Dynamic Information Flow Tracking for IoT Privacy and Security", The Network and Distributed System Security Symposium, Feb. 2019.

[14] Dharavath Ramesh, Neeraj Patidar, Gaurav Kumar, Teja Vunnam. "Evolution and Analysis of Distributed File Systems in Cloud Stoarge: Analytical Survey". International Conference on Computing, Communication and Automation, Apr. 2016.

[15] Kuo Sheng Deng, Chin Feng Lee, Jerry Chou, Yi Chen Shih, Shang Hao Chuang, Po Hsuan Wu. "pNFS-based Software-Defined Storage for Information Lifecycle Management" International Conference on Cloud Computing and Big Data, Nov. 2015.

[16] Ming Chen, Dean Hildebrand, Henry Nelson, Jasmit Saluja, Erez Zadok. "vNFS: Maximizing NFS Performance with Compounds and Vectorized I/O", USENIX, Feb. 2017.

[17] Ghania Al Sadi. "Tuning and Optimizing Network File System Server Performance" International Journal of Computer Applications, Jan. 2016.

[18] Philippe Deniel, Thomas Leibovici, Jacques-Charles Lafoucriere. "GANESHA, a multi-usage with large cache NFSv4 server" Linux Symposium, Jun. 2007.

[19] J. Blomer. "A Survey on Distributed File System Technology" Advanced Computing and Analysis Techniques in Physics Research, Sep. 2014.

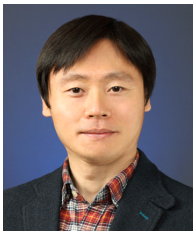
[20] Ejaz Ahmed, Mubashir Husain Rehmani. "Mobile Edge Computing: Opportunities, solutions, and challenges", 70(2017), pp. 59-63, May. 2017

[21] Tzong-Jye Liu, Chun-Yan Chung, Chia-Lin Lee. "A High Performance and Low Cost Distributed File System". IEEE 2nd International Conference on Software Engineering and Service Science, Jun. 2011.

〈저 자 소 개〉

임 한 울 (Han-wool Lim) 학생회원
2020년 2월: 세종대학교 정보보호학과 졸업
2020년 3월~현재: 세종대학교 정보보호학과 석사과정
〈관심분야〉 정보보호, 엣지 컴퓨팅, IoT 보안

변 원 준 (Won-jun Byoun) 학생회원
2018년 2월: 선문대학교 컴퓨터공학과 졸업
2020년 9월~현재: 세종대학교 정보보호학과 석사과정
〈관심분야〉 정보보호, 엣지 컴퓨팅, IoT 보안



윤 주 범 (Joobeom Yun) 종신회원
1999년 2월: 고려대학교 컴퓨터학과 학사
2001년 2월: 서울대학교 컴퓨터공학과 석사
2012년 2월: KAIST 전산학과 박사
2001년 3월~2015년 2월 : ETRI부설연구소 선임연구원
2015년 3월~현재: 세종대학교 정보보호학과 부교수
〈관심분야〉 네트워크 보안, 시스템 보안, 인공지능 보안