

ScienceDMZ에서의 초고속 전송을 위한 REST API기반 3자 접근 모델

이 상 권*, 석 우 진°, 문 정 훈*

REST API Based 3rd-Party Access Model for High-Speed Transmission in ScienceDMZ

Sang-gwon Lee*, Woo-Jin Seok°, Jeong-hoon Moon*

요 약

데이터의 크기가 날마다 커져가는 빅데이터 시대에 이러한 데이터를 활용할 수 있도록 다양한 연구가 활발히 진행되고 있다. 그 중에서는 대용량 데이터의 효율적인 전송을 위한 초고속 전송망을 구축하고자 노력하고 있다. 현재 국내외 연구기관에서는 협업을 위해 ScienceDMZ라고 불리는 초고속 전송 전용망을 이용하여 연구데이터를 상호 공유하고 있으며 고속 데이터 전송을 위해서 GridFTP 같은 고성능 전송 프로그램을 사용하고 있다. 하지만 이러한 방법은 광범위한 접근을 위해 ScienceDMZ 외부의 장소로 부터 접근을 하고자 할 때, ScienceDMZ가 초고속 전송을 위해 별도로 분리되어 있는 네트워크라는 점 때문에 접근이 어렵다는 문제점을 가지고 있다. 이러한 문제를 해결하고자 REST API를 이용하는 3자 전송 서버를 구축하여 REST API의 요청을 통해 전용망 접근이 가능하도록 하여 문제를 해결하였다. 3자 전송이 가능한 고성능 전송 환경 구축 후에 전송 테스트를 진행한 뒤 전송 결과를 분석하여 REST API를 이용한 3자 전송방법이 기존의 전송 방법에 비해 전송 속도의 감소가 없으며 외부 네트워크 노드의 접근이 가능하다는 점을 확인하였다.

Key Words : ScienceDMZ, REST API, 3rd-party transfer, DTN, SSD RAID, Network Tuning

ABSTRACT

Various studies have been actively conducted to utilize such data in the era of big data, where the size of data is increasing day by day. Among them, we are trying to build a high-speed transmission network for efficient transmission of large amounts of data. At present, research institutes in Korea and abroad share research data with each other through high-speed transmission network called ScienceDMZ for collaboration and use high-performance transmission programs such as GridFTP for high-speed data transmission. However, this method has a problem that when accessing from a place outside of ScienceDMZ for broad access, ScienceDMZ is difficult to access because it is a separate network for high-speed transmission. We solved the problem by constructing a third-party transfer server using the REST API and enabling access to the dedicated network through the request of REST API. After constructing high performance transmission environment capable of

* 본 연구는 한국과학기술정보연구원 ‘글로벌 협업연구 지원 국가 과학기술연구망 구축 및 서비스(K-19-L02-C02-S01)’ 과제의 지원을 받아 수행되었음

◆ First Author : UST(University of Science and Technology), sglee@kisti.re.kr, 학생회원

° Corresponding Author : Korea Institute of Science and Technology Information, wjseok@kisti.re.kr, 정회원

* Corresponding Author : Korea Institute of Science and Technology Information, jhmoon@kisti.re.kr, 정회원

논문번호 : 201904-038-D-RN, Received April 3, 2019; Revised May 16, 2019; Accepted May 21, 2019

third-party transmission, we test transmission and analyze the transmission results to confirm that the third-party transmission method using the REST API has no reduction in transmission speed compared to the existing transmission method and that the external network node can be accessed.

I. 서 론

오늘날 컴퓨터 및 인터넷의 발달로 생성되는 데이터의 크기는 이전에 비해 계속해서 증가되고 있다. 이렇게 생성되어진 방대한 양의 데이터를 빅데이터^[1-3]라고 부른다. 굉장히 큰 용량 단위인 페타바이트도 빅데이터의 시대에서는 부족해지는 날이 점점 다가오고 있을 정도로 빅데이터는 핫 이슈이다. 이러한 경향에 맞추어 빅데이터를 활용하려는 연구가 과학계, 소셜, 의료, 범죄망의 분석등 다방면으로 활발히 진행되고 있다.^[4-7] 특히 연구 분야에서 LSST(Large Synoptic Survey Telescope)^[8,9]같이 기존에 비해 대용량 데이터의 처리 필요로 하는 대형 실험장치가 많이 생겨나게 되면서 이를 수용할 수 있는 데이터 전송 전용망을 구성하기 위한 연구의 필요성이 증가하고 있다.^[10,11]

현재 국내외 연구기관들은 연구 데이터의 공유를 통한 협업을 위해서 고성능 전송망 구축에 대한 프로젝트들을 진행하고 있다. 미국에서는 PRP(Pacific Research Platform) 프로젝트를 통해 50개가 넘는 연구기관들이 참여하고 있다.^[12] PRP는 입자 물리학, 천문학, 생물학, 지구 과학, 데이터 시각화 등 다양한 주제의 연구 데이터를 공유하기 위한 데이터 전송 전용망의 구성에 대한 연구를 진행하는 프로젝트이다. PRP에서는 미국 에너지부의 ESnet(Energy Sciences Network)에서 개발한 ScienceDMZ모델^[13]을 채용하여 전용망을 구성하고 있다. ScienceDMZ는 과학 데이터 전송을 위한 목적으로 대학이나 연구소에 설치되는 초고속 전송이 가능한 네트워크를 말한다. ScienceDMZ에서는 SDN같은 네트워크 가상화 기술을 통해 일반 사용목적의 비즈니스 트래픽과 과학 데이터 트래픽을 분리하여 패킷로스도 인한 성능 저하를 최소화하고 100Gbps 이더넷카드, 레이드로 구성된 SSD(Solid State Drive)스토리지 같은 고성능 하드웨어가 설치된 다수의 DTN(Data Transfer Node)을 상호 연결을 통해 구성하여 과학데이터의 초고속 전송을 가능하도록 하였다. DTN은 초고속 데이터 전송을 위해 고성능 하드웨어와 최적의 성능을 내기 위한 네트워크 튜닝이 되어있는 서버를 말한다.

본 논문에서는 고성능 장비를 장착한 2대의 DTN을 설치하고 DTN의 네트워크 전송 성능을 최대로 이

끌어 내기 위하여 시스템과 네트워크 장비의 옵션들을 적절한 값으로 설정하는 네트워크 튜닝 작업과 REST API를 이용한 3자 전송 서버를 구축하고 이를 활용하는 3자 전송 모델에 대해 설명한다.^[14-15]

DTN에 고성능의 장치를 설치한 후 기본 옵션으로 사용 시에는 장비의 최대 가용 성능의 20~30% 정도 밖에 사용하지 못하기 때문에 여기에는 네트워크 튜닝 작업이 필요하다. 네트워크 튜닝작업에는 CPU의 클럭이 항상 최대치로 작동되도록 설정하고 네트워크 버퍼의 크기를 늘리고 송신 큐의 크기를 변경하는 등의 작업들을 포함한다.

DTN 간 데이터 전송에는 GridFTP^[16]이라는 고성능 전송 전용 소프트웨어를 사용한다. GridFTP를 통해 설치되는 소프트웨어는 2가지 종류로 데이터 전송을 직접적으로 수행하는 서버 프로그램과 서버에 명령어를 보내 제어하는 역할을 하는 클라이언트 프로그램이 있다. 2개 이상의 서버와 1개의 클라이언트가 있다면 3자 전송을 이용한 데이터 전송이 가능하다. 3자 전송을 이용하여 전송하게 되면 하나의 클라이언트를 통해 모든 서버에 명령어를 보내 조작하는 가능하기 때문에 데이터 송신 또는 수신 서버에 직접 접속할 필요 없이 손쉽게 전송 경로 설정이 가능하다는 이점이 있다. 하지만 ScienceDMZ 같은 초고속 전송 전용망에서는 성능 향상을 위해 기본적으로 외부와 네트워크를 분리하여 사용하기 때문에 전용망 이외의 장소에서는 접근이 불가능하다. 그렇기 때문에 외부에 있는 노드를 통해 GridFTP의 3자 전송 기능을 사용하기가 어렵다는 문제가 있다. 이를 해결하고자 REST API(Representational State Transfer Application Program Interface)^[17-19]를 이용한 3자 전송 서버를 구축하여 REST API를 통해 외부에서도 3자 전송이 가능하도록 설계하여 앞서 얘기한 문제점을 해결하였다.

II. DTN 네트워크 튜닝과 3자 전송을 위한 REST API 서버

2.1 DTN 네트워크 튜닝

100Gbps 같은 높은 대역폭의 네트워크 환경에서 기본적인 설정만을 통해서 사용자가 얻을 수 있는 실질적인 전송속도는 기대했던 성능에 비해 상대적으로

났다. 네트워크 장비가 지원하는 대역폭을 최대한 활용하기 위해서는 DTN의 튜닝은 필수적인 요소이다.^[20] DTN 네트워크 튜닝에 있어서 하드웨어의 정보에 따라 네트워크 튜닝방법은 달라질 수 있다. 여기에서 설명하는 튜닝방법은 아래 표 1에서 기술하고 있는 한국과학기술정보연구원에 설치되어있는 DTN의 하드웨어 스펙에 따라 적용되었다.

DTN 튜닝설정에 있어서 전송 속도에 가장 큰 영향을 주는 요소는 점보프레임이다. 점보프레임은 한 번에 전송하는 패킷의 크기를 늘려서 고속 전송을 가능하게 해주는 기술이다. MTU(Maximum Transmission Unit)값을 통해 패킷의 크기를 변경 할 수 있다. MTU는 기본값으로 1500으로 설정되어 있는데 아래의 명령어를 통해서 9000으로 설정해 줄 수 있다. \$Network에는 변경하고 하는 네트워크 인터페이스명을 입력해주면 된다. 서버와 클라이언트 양쪽 모두 같은 MTU로 설정되어야 제대로 동작한다.

```
#ifconfig $Network mtu 9000
```

그 다음 튜닝 옵션으로는 송신 큐 크기 설정이 있다. 패킷 전송 시에 패킷은 그 즉시 상대방측에 보내지는 것이 아니고 일단 네트워크 인터페이스에 안에 있는 큐에 저장된다. 이 큐를 통해서 전송 속도 제한 같은 트래픽 제어를 할 수 있으나 큐가 가득 차게 되면 패킷 손실이 일어날 수 있기 때문에 큐의 크기를 늘려주는 튜닝이 필요하다. 리눅스에서는 txqueuelen라는 옵션을 통해 변경해줄 수 있다. txqueuelen은 기본값으로 1000으로 설정되어 있는데 아래의 명령어를 통해 10000으로 설정해 줄 수 있다.

```
#ifconfig $Network txqueuelen 10000
```

표 1. DTN 하드웨어 스펙
Table 1. DTN Hardware Specification

Classification	Description
Product name	PowerEdge R730
CPU	Intel Xeon E5-2640 v4 @ 2.40GHz X 2
Memory	128 GB PC4-17000
Storage	Samsung SSD 860 EVO X 8, Samsung SSD 970 PRO X 4
RAID Controller	PERC H730P Mini(RAID 0 with Samsung SSD 860 EVO X 8), Highpoint 7101A(RAID 0 with Samsung SSD 970 PRO X 4)
Network adapter	Mellanox ConnectX-5 Dual Port 100Gb Network Adapter
Kernel	Red Hat 4. 8. 5-28

2.2 DTN 하드웨어 튜닝

운영체제는 전력의 효율화를 위해 CPU 사용이 적을 시에 유휴 상태가 되도록 설정되어 있다. 이 때 고성능이 필요한 전송 작업을 진행하게 되면 CPU가 유휴 상태에서 최고 성능 상태로 진입하는데 오버헤드가 존재하기 때문에 성능이 제대로 나오지 않는다. 이를 방지하고자 CPU가 항상 최고 성능으로 작동되도록 Red hat 4.8.5-28 커널 사용 기준으로 아래의 명령어를 통해 설정해 줄 수 있다.

```
#cpupower frequency-set -g performance
```

DTN을 통해 실제 데이터 전송작업을 진행하면 네트워크단에서 속도를 테스트 했을 때 보다 훨씬 낮은 속도로 진행된다. 이는 메모리상에서 전송된 데이터를 스토리지로 옮기는 과정에서 스토리지의 속도가 메모리의 속도에 비해 현저히 낮기 때문에 나타나는 현상이다. 이러한 속도 차에 의한 전송 속도의 감소를 줄이기 위해서 스토리지의 읽기, 쓰기 속도를 높여줄 필요가 있다. 스토리지의 속도는 스토리지의 종류와 스토리지의 구성 방식에 따라 달라진다.

스토리지의 종류에는 HDD(Hard Disk Drive), Sata 방식의 SSD 그리고 m.2 규격을 사용하는 Nvme SSD(Non Volatile Memory Express Solid State Drive)가 있으며 언급된 스토리지들의 속도는 앞에서 부터 뒤로 갈수록 빠르다. 가장 느린 스토리지와 가장 빠른 스토리지의 속도를 비교해보면 HDD인 씨게이트 2TB 7200RPM 제품의 쓰기 속도가 120MB/s 인데 비해 Nvme 방식의 SSD인 삼성 970 PRO의 쓰기 속도는 2700MB/s로 약 22배 정도의 차이가 있다.

스토리지 종류 다음으로 쓰기 속도에 영향을 주는 요소는 레이드 컨트롤러이다. 레이드 컨트롤러는 다수의 스토리지를 관리하며 스토리지의 레이드 설정을 수행하는 장치를 말한다. 레이드는 스토리지 여러 개를 묶어 고용량 또는 고성능을 가지는 스토리지로 만들어 사용하기 위해 개발된 기법이다. 레이드 컨트롤러의 성능에 따라 동일한 스토리지를 가지고 레이드를 구성하여도 속도의 차이가 생긴다. 레이드 컨트롤러의 종류에는 2가지 있다. 하나는 운영체제에서 지원하는 소프트웨어 방식으로 작동되는 레이드 컨트롤러로 추가적인 비용 없이 레이드 구성이 가능한 반면 성능은 떨어진다는 단점을 가지고 있다. 나머지 하나는 하드웨어 방식으로 작동되는 레이드 컨트롤러로 시스템에 별도의 레이드 컨트롤러 장치를 장착해야하기 때문에 추가 비용이 발생한다는 단점이 있는 대신 성능이 우수하다. 레이드를 구성하고 있는 스토리지의 수를 늘일수록 읽기와 쓰기 속도가 증가된다. 추가 가

능한 최대 스토리지의 수는 레이더 컨트롤러가 가지고 있는 한계에 따라 달라진다.

레이드 컨트롤러의 옵션에서는 레이드 종류를 설정할 수 있다. 주로 사용되는 레이드의 종류에는 레이드 0, 1, 5, 6이 있다. 이 중 스토리지의 성능을 최대로 끌어올릴 수 있는 레이드는 레이드 0이다. 레이드 0에서는 디스크 스트라이핑(Disk Striping)이란 기술을 사용하는데 디스크 스트라이핑은 데이터를 저장할 때 여러 조각으로 만들어 다수의 디스크에 나누어 저장하는 기술로 구성되는 디스크의 수에 따라 입출력 속도가 늘어나는 특징을 가진다. 레이드 0의 경우 스토리지의 속도를 향상시켜준다는 장점을 가지고 있지만 구성된 디스크 중 하나라도 고장날 시에는 전체 스토리지를 사용할 수 없게 되며 데이터의 복구 또한 힘들기 때문에 안정성 측면에서 부족하다는 단점을 가지고 있다.

2.3 3자 전송 방식

3자 전송은 데이터 전송에 직접적으로 관여하는 데이터 송신 서버와 데이터 수신 서버 사이에 데이터 전송 명령어를 각 서버에 전달하는 역할을 하는 클라이언트를 추가한 후에 이를 이용하여 간접적으로 데이터를 전송시킬 수 있는 방법이다.

기존의 데이터 전송 방법인 서버 대 서버간의 전송 방식과 3자 전송 방식을 비교해보면 기존의 방법으로 데이터 전송을 진행하려면 데이터를 송신할 서버에 접속하여 전송 명령어에 목적지 서버 주소 정보와 디렉토리 그리고 전송할 파일명을 입력해 데이터 전송을 진행한다. 이후 데이터의 전송경로를 변경하고자 할 때 목적지의 변경의 경우 단지 전송 명령어에서 목적지 서버 주소 정보를 변경해주면 되지만 출발지 서버를 변경하고자 한다면 변경하고자 하는 서버의 주소로 다시 접속해야 한다. 출발지 주소의 변경을 통해 데이터 전송 경로를 변경할 때 마다 이러한 과정을 거치기 때문에 번거롭다.

반면 3자 전송 방식의 경우에는 데이터를 송신 또는 수신하는 서버를 조작하는 역할을 수행하는 클라이언트를 사용한다. 클라이언트를 통해 전송 명령어에 송신 서버와 수신 서버의 주소를 입력하여 데이터를 전송하게 되는데, 이 경우 데이터 전송 경로를 변경하더라도 전송 명령어 안에 입력되는 송신 또는 수신 서버의 주소만 변경하면 전송이 가능하기 때문에 재접속 같은 과정의 필요 없이 상대적으로 간편하게 조작할 수 있다는 장점을 가지고 있다.

2.4 REST API를 이용한 3자 전송 서버

DTN에서 사용하는 고속 전송 프로그램인 GridFTP에서 기본적으로 3자 전송 기능을 지원하지기 때문에 2대 이상의 GridFTP 서버와 1대 이상의 GridFTP 클라이언트가 설치되어 있다면 추가적인 작업 없이 자체적으로 3자 전송이 가능하다. GridFTP 클라이언트와 GridFTP 서버를 이용한 3자 전송은 아래의 그림 1과 같이 진행된다. 클라이언트에서 컨트롤 채널을 통해 데이터를 송신 또는 수신을 진행할 GridFTP 서버에 제어 명령어를 전달하면 GridFTP 서버에서는 명령어에 따라 데이터 채널을 통해 전송작업을 진행하게 된다. 그림 1에서 보여주는 것처럼 GridFTP 클라이언트를 다른 노드에 설치하기만 한다면 다른 위치에서도 3자 전송 작업을 진행하는 것이 가능하다. 이 경우 일반적인 네트워크에서의 사용에는 문제가 없지만 ScienceDMZ 모델과 같이 외부 네트워크로부터 분리되어 있는 전송망에서 사용하게 될 시에는 전송망 내에 위치한 GridFTP 서버 프로그램을 실행하는 DTN과 외부 네트워크에서 위치한 GridFTP 클라이언트 사이에 서로 통신이 막혀있어 접근이 안되기 때문에 3자 전송 방식을 이용하는 것이 불가능하다는 문제를 가지고 있다.

앞에서 언급된 문제점을 해결하고자 본 논문에서는 REST API를 이용한 3자 전송 서버를 제안하였다. 그림 1과 같은 방식으로 진행되는 GridFTP에서 지원하는 3자 전송 방법은 GridFTP 클라이언트와 DTN 모두 같은 전송망에 속해 있으면 문제가 없지만 앞에 언급한 문제처럼 ScienceDMZ 같은 전용망에서 GridFTP 클라이언트가 외부 네트워크에 위치하고 있는 경우에는 상호간 접근이 불가능하게 된다. 따라서 전용망에 속해있으면서도 외부 네트워크와 접근이 가

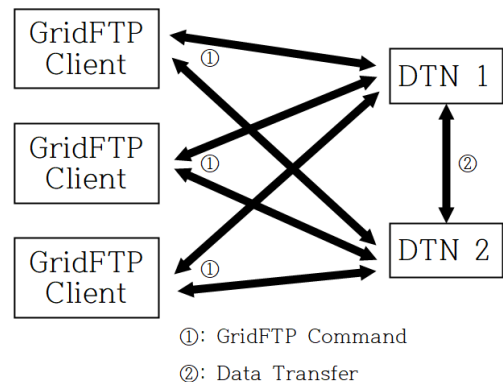


그림 1. GridFTP의 3자 전송방식
Fig. 1. 3rd-party Transmission method of GridFTP

능한 3자 전송 서버의 역할이 필요하다. REST API를 이용한 3자 전송 서버는 사용자로부터 REST API로 작성된 요청메시지를 받으면 이를 GridFTP의 전송 명령어로 변경시켜 실제 전송 작업을 진행할 서버에 전송해주거나 3자 전송 서버의 데이터베이스에 저장되어 있는 정보를 전달해주는 역할을 한다.

REST API 기반의 3자 전송 서버는 그림 2에서 나타나는 것처럼 사용자의 명령어를 받아서 GridFTP의 명령어로 변환하여 전달하는 중계역할을 해주기 때문에 전송망과 외부 네트워크 양쪽 모두로부터 통신이 가능하도록 하는 설정이 필요하다. 적절한 설정 없이 양쪽 네트워크를 사용하기 위하여 네트워크 설정에서 2개 이상의 게이트웨이를 입력하게 되면 default 게이트웨이로 설정된 게이트웨이만 작동되거나 경우에 따라서는 네트워크에 문제가 생겨 작동이 아예 안될 수도 있다. 따라서 다중 네트워크를 사용하기 위해서는 라우팅 테이블과 룰의 적절한 설정이 요구된다.

ScienceDMZ에서 데이터 전송에 사용되는 DTN은 VLAN(Virtual Local Area Network)을 사용하여 통신한다. VLAN은 가상 네트워크망을 말하여 이 기능을 통해 사용자가 원하는 네트워크망의 구조를 논리적으로 구성할 수 있다. 간단한 설정을 통해 네트워크를 논리적으로 분리할 수 있어 불필요한 트래픽을 줄여준다는 이점이 있어 DTN의 네트워크 구성에 사용되고 있다.^[21] VLAN은 스위치에서 제공되는 기능으로 설정은 주로 스위치를 통해 진행되고 서버에는 VLAN에서 사용될 IP 주소 같은 정보만 입력해 주면 된다. 스위치에서는 VLAN 태깅을 통해 네트워크를 분리할 수 있다. 스위치가 가지고 있는 각 포트마다 VLAN 태깅을 할 수 있다. 이 때 동일한 값으로 VLAN 태깅이 되어야만 서로 통신이 가능하고 다른 값으로 VLAN 태깅이 된다면 네트워크가 분리되어 통신이 되지 않게 된다. 그림 2에서 나타나는 것처럼 REST API를 이용한 3자 전송 서버는 VLAN을 통해 네트워크를 구성한 DTN 과 서로 통신이 되어야하기 때문에 3자 전송 서버와 DTN을 동일한 VLAN 태깅으로 설정해야한다.

3자 전송 서버에서 언급되는 REST는 웹 서비스를 통해 제공되는 이미지, 비디오, 데이터베이스 같은 자원에 각각의 URL를 부여하여 해당 자원을 HTTP Method를 통해 처리 가능하도록 구성된 아키텍처를 말한다. 이러한 과정을 API를 통해 수행할 수 있게 구성한 것을 REST API라고 한다. REST API의 형태로 웹 서비스를 구축하게 되면 사용자의 기기나 환경에 제약을 받지 않고 본래의 성능을 낼 수 있으며 다양한

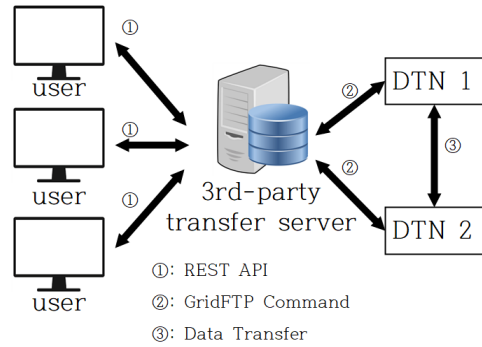


그림 2. 3자 전송 서버를 이용한 전송방식
Fig. 2. Transmission method using 3rd-party transfer server

서비스와도 쉽게 접목이 가능하다는 장점이 있다.^[22]

REST 아키텍처에는 6가지의 제약 사항이 있으며 각각의 제약 사항들에 대한 내용은 아래와 같다. 첫 번째는 인터페이스 일관성(Uniform Interface)로 URL로 지정된 자원에 대한 조작을 통일된 인터페이스를 통해 수행한다. 두 번째는 무상태성(Stateless)으로 REST에서 Stateless 특성을 가지는 HTTP 프로토콜을 사용하기 때문에 마찬가지로 무상태성을 가진다. 세 번째는 캐시 처리 가능(Cacheable)으로 대량의 요청들에 대해 빠른 속도로 처리하기 위해 캐시가 요구된다. 네 번째는 클라이언트 - 서버 구조(Client - Server Architecture)로 일관성 있는 인터페이스를 통한 서버와 클라이언트의 분리로 클라이언트는 더 이상 서버에 있는 데이터 스토리지에 대해 신경 쓰지 않아도 되고 서버 쪽에서는 사용자의 인터페이스나 상태에 대해 신경 쓰지 않아도 되기 때문에 각자 독립적으로 개발되거나 교체가 가능하다. 다섯 번째는 계층형 구조(Layered System)로 클라이언트가 최종 서버에 직접 연결되어 있는지 중간에 중계자를 통해 연결되었는지 상호간 차이가 존재하지 않기 때문에 중계 서버를 이용해서 로드 밸런싱을 사용 한다 던지, 공유 캐시를 제공한다거나 보안 정책을 강화시키는 등의 시스템을 확장 시키는 것이 가능하다. 여섯 번째는 주문형 코드(Code on Demand)로 서버가 임시로 클라이언트의 기능을 스크립트를 이용해 확장하거나 수정이 가능하다. 사용 예시로는 자바 애플릿이 있다. 여섯 번째 제약인 주문형 코드는 선택적인 옵션이기 때문에 이를 제외한 나머지 제약들을 모두 따르고 있는 REST API를 제공하는 웹 서비스를 RESTful하다고 말한다.

HTTP, URL를 이용하는 REST API를 통해 웹 서비스를 구현하는 방법을 설명하자면 다음과 같다.

표 2. REST API 설계
Table 2. Design of REST API

Method	URL	Payload	Action
GET	IP/info	N/A	Get DTN information
GET	IP/log	N/A	Get transfer result log
GET	IP/transfer	Source_IP, Destination_IP	Directory lookup
POST	IP/transfer	Source_IP_File, Destination_IP_Directory	File transfer

REST에서는 HTTP의 기본 메소드를 통해 자원에 대한 행위를 정의한다. HTTP 메소드는 클라이언트와 서버가 요청과 응답을 주고받을 수 있도록 설계된 방식이다. REST에서 주로 사용되는 HTTP 메소드는 4 가지 종류가 있다. 자원의 생성을 위해 사용하는 POST, 자원을 조회하기 위해 사용되는 GET, 자원을 수정하기 위해 사용되는 PUT 그리고 자원을 삭제하기 위한 DELETE가 있다.

REST API를 이용한 3자 전송 서버에서 사용되는 REST API의 디자인은 표 2와 같은 형태로 되어 있다. 가장 위에 있는 Action은 DTN 정보를 조회하는 것이다. 데이터베이스에 저장되는 DTN 정보에는 DTN의 IP 주소나 DTN이 설치된 위치에 대한 정보로 구성 되어있다. DTN 정보를 얻기 위해서는 3자 전송 서버의 IP 주소에 /info를 더한 URL에 GET 요청을 보내면 된다. 다음 Action은 데이터베이스에 저장되어 있는 전송 기록들을 조회하는데 사용되는 REST API이다. 전송된 기록에는 전송된 날짜와 파일 크기, 파일명, 출발지 IP, 도착지 IP가 저장된다. 이러한 기록들을 통해 평균 전송속도를 알 수 있다. 여기에 사용되는 URL은 3자 전송 서버 IP에 /log를 붙이면 되고 서버에 전송할 시에는 GET 요청으로 보내면 된다. 현재 서버의 디렉토리 상태를 보기위해서 또는 파일 전송하기 전과후의 결과를 확인하기 위해 디렉토리를 조회하는 Action이 있다. 여기에는 3자 전송 서버IP에 /transfer를 붙인 URL 에 GET 요청을 보내면 되는데 앞서 언급했던 Action들과 다르게 DTN과 상호작용이 필요하기 때문에 출발지의 DTN IP와 목적지의 DTN IP에 대한 정보를 같이 보내줘야 한다. 마지막은 파일을 전송하는 Action이다. 디렉토리를 조회하는 Action과 같은 URL에 GET 요청을 POST 요청으로 변경하여 보내면 된다. 여기에서 추가로 출발지의 파일명과 도착지의 디렉토리에 대한 정보를 추가하여 3자 전송 서버에 전달하면 데이터 전송이 시작된다.

III. 구현 및 시험

본 장에서는 1장과 2장에서 설명한 DTN을 설치와 네트워크 튜닝을 진행하고 REST API를 이용한 3자 전송 서버를 구현하여 GridFTP에서 지원하는 3자 전송 방법과 REST API를 이용하는 3자 전송 서버를 활용한 3자 전송 방법을 시험을 통해 각각 비교하였다.

3.1 DTN 장비 설치

시험에 사용되는 DTN은 DELL사의 PowerEdge R730 서버를 기반으로 하여 설치하였다. 서버의 운영 체제는 리눅스 Cent 7을 사용하였다. 여기에 초고속 전송을 위하여 Mellanox 사의 100Gbps의 전송속도를 가지는 고성능 네트워크 카드를 설치한다. 또한 실제 데이터 전송 속도에 있어서 속도 감소의 주된 원인인 낮은 디스크 읽기 쓰기 속도 문제를 해결하기 위해 Nvme 방식의 SSD를 여러 개를 설치한 뒤에 레이드 컨트롤러를 통해 레이드 0으로 설정하였다. 전체적인 하드웨어 스펙에 대해서는 표 1를 통해 상세히 나타내었다. DTN 설치가 있어서 반드시 위와 동일한 시스템을 구성할 필요는 없다. 일부는 보다 고성능을 위하여 값비싼 장비로 구성하기도 하는데 보통은 제한된 예산 문제로 그렇게 구성하기가 힘들기 때문에 ESnet에서는 보급형 버전의 DTN인 FIONA(Flash I/O Network Appliance)를 발표하여 적절한 가격에 준수한 성능을 갖춘 DTN을 만들 수 있도록 기준을 제시하였다. 또한 ESnet에서는 아프리카나 동남아시아 같은 저소득 국가에서도 DTN을 설치 할 수 있도록 지원하기 위해 저가형 버전의 DTN인 FIONNET을 발표하였다.

DTN의 설치 이후에는 제 성능을 내기 위해 하드웨어 및 네트워크 튜닝이 진행되어야 한다. 하드웨어 튜닝에서는 바이오스에서 CPU와 메모리가 고성능으로 동작하도록 클럭 속도를 높여주는 오버클럭을 진행한다. 운영체제부분에서는 커널 설정을 통해 CPU의 사용량이 적을 때 CPU의 성능을 낮추어 전력을 절약하는 전력 절약 옵션을 끄고 메모리의 페이지 크기를 늘려주는 설정을 한다. 네트워크 튜닝에서는 보다 빠른 전송을 위해 전송되는 패킷 사이즈를 늘리기 위해 MTU 사이즈를 높여주는 설정을 한다. 그리고 대용량 데이터를 전송 시 패킷 로스로 인한 속도 저하를 막기 위해 버퍼 사이즈를 높여주는 설정을 진행하였다.

3.2 REST API를 이용한 3자 전송 서버의 구현 REST API를 이용한 3자 전송 서버의 프로그램은

구현이 간단하고 생산성이 높은 프로그래밍 언어인 파이썬을 사용하여 개발하였다. 개발 환경은 파이썬에서 제공하고 있는 IDLE(Integrated Development and Learning Environment)을 사용하였고 여기에 파이썬 기반의 웹 개발을 위한 프레임워크인 플라스크(Flask)를 설치하여 표 2에서 설계하였던 REST API에 따라 각 Action을 처리할 수 있도록 함수를 작성하여 구현하였다. 구현된 함수들 중에서 가장 핵심적인 부분인 3자 전송 명령어를 다루는 함수를 아래의 그림 3과 같이 작성하였다. 이 함수에서는 사용자로부터 /transfer URL로 보내진 Json 형식을 따르는 요청 메시지를 받으면 여기에 저장되어 있는 목적지와 도착지 정보를 받아 저장한 뒤에 메소드가 GET인지 POST인지를 확인한다. 메소드가 POST로 설정되어 있다면 GridFTP 클라이언트의 전송 명령어를 실행하도록 구현하였다.

2장에서 설명한 REST API의 디자인을 사용하여 구현된 REST API를 이용한 3자 전송 서버의 아키텍처를 그림 4를 통해 쉽게 이해할 수 있도록 나타내었다.

REST API를 이용한 3자 전송 서버의 아키텍처에 대해 설명하자면 다음과 같다. 3자 전송을 사용하려는 사용자는 3자 전송 서버에 cURL 같은 소프트웨어를 사용해 요청메시지를 보내게 되는데 이 때 사용되는 요청메시지는 제안된 REST API의 디자인에 맞게 설계 되어있어 3자 전송 서버가 이를 확인하여 각 명령어에 맞는 작업을 진행하게 된다. 요청메시지를 받은 3자 전송 서버는 요청 메시지의 URL과 메소드를 확인하는데 이 때 요청 메시지의 메소드가 GET이고 URL이 /Info로 되어 있으면 3자 전송 서버의 데이터베이스에 저장되어 있는 DTN의 정보를 가져와 사용자에게 전달해준다. 전달 받은 DTN의 정보를 통해 사용자는 DTN서버의 위치 정보, IP 주소 그리고 하드웨어의 스펙을 알 수 있다. 요청 메시지의 메소드가 GET이고 URL이 /Log로 되어 있으면 3자 전송 서버는 데이터베이스에 저장되어 있는 전송 기록을 보여준다. 전송 기록에 저장되어 있는 정보에는 데이터 전송 날짜, 출발지, 목적지, 데이터 크기, 평균 전송 시간 그리고 평균 전송 속도가 있다. 요청메시지의 메소드가 POST이고 URL이 /Transfer로 되어 있으면 요청

```
@app.route("/transfer", methods=["GET", "POST"])
def transfer():
    content = request.json
    Source = content['Source']
    Dest = content['Dest']
    if request.method == "POST":
        a=os.popen("globus-url-copy -vb ftp://"+Source+" ftp://"+Dest).read()
```

그림 3. 3자 전송 명령어 함수
Fig. 3. 3rd-party transfer command function

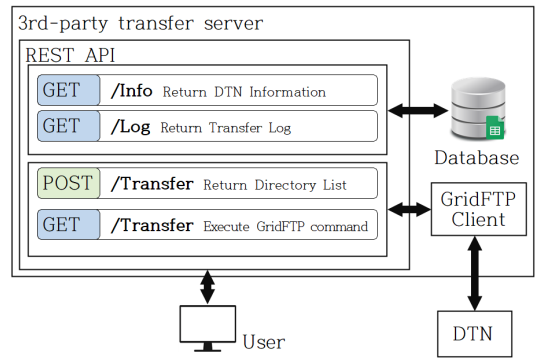


그림 4. REST API를 이용한 3자 전송 서버 아키텍처
Fig. 4. 3rd-party transfer server architecture using REST API

메시지에 저장되어 있는 출발지 DTN이나 도착지 DTN의 주소를 읽어 해당 DTN의 디렉토리 정보를 보내준다. 이를 통해 사용자는 현재 디렉토리에 어떠한 파일이 저장되어 있고 전송 후에는 전송된 데이터를 확인할 수 있다. 요청메시지의 메소드가 GET이고 URL이 /Transfer로 되어 있으면 요청메시지에 저장되어 있는 출발지 DTN, 도착지 DTN, 데이터 정보를 토대로 GridFTP 클라이언트를 이용한 3자 전송을 진행하게 된다.

3.3 3자 전송 방식의 예시

GridFTP 클라이언트를 이용한 3자 전송과 REST API를 이용한 3자 전송 서버를 통해 3자 전송을 진행하는 방법을 예시를 통해 설명하자면 다음과 같다.

우선 사용자가 GridFTP 클라이언트를 사용하기 위해서 GridFTP를 설치하는 작업이 필요하다. 설치에 필요한 명령어는 Cent7 기준으로 다음과 같다. 우선 GridFTP 설치에 필요한 저장소를 설치해야 하며 저장소의 설치 명령어는 다음과 같다. “yum install globus-connect-server-repo-latest.noarch.rpm” 저장소를 설치하고 나면 다음의 명령어를 통해 GridFTP를 설치할 수 있다. “yum install globus-connect-server” GridFTP의 설치가 완료되면 GridFTP 클라이언트를 사용할 수 있게 된다.

GridFTP 클라이언트를 사용하려면 커맨드 창에 globus-url-copy를 사용하면 된다. globus-url-copy를 사용할 때는 옵션으로 -vb를 붙여주면 좋다. -vb 옵션은 전송시 데이터의 크기나 평균 전송 속도, 초기 전송 속도를 사용자에게 제공해준다. -vb옵션 뒤에는 출발지 DTN의 IP 주소와 포트번호, 디렉토리 그리고 전송할 파일의 이름을 쓰고 이어서 도착지 DTN의 IP

```
[kisti@134 ssd]$ sudo globus-url-copy -vb ftp://134.75.115.133:2811/export/ssd/100GB
100GB ftp://134.75.115.131:2811/export/ssd/100GB
Source: ftp://134.75.115.133:2811/export/ssd/100GB
Dest: ftp://134.75.115.131:2811/export/ssd/100GB
104882765824 bytes 1127.67 MB/sec avg 1247.16 MB/sec inst
```

그림 5. GridFTP 클라이언트를 이용한 3자 전송 예시
Fig. 5. Example of 3rd-party transmission using GridFTP Client

주소와 포트번호, 디렉토리 그리고 저장할 파일의 이름을 쓰면 된다. 앞에서 설명한 대로 globus-url-copy 를 통해 GridFTP에서 제공하는 방법으로 3자 전송을 진행하면 그림 5와 같은 결과를 얻을 수 있다.

REST API를 이용한 3자 전송 서버를 사용해 3자 전송을 진행하려면 REST API를 쉽게 사용할 수 있는 cURL을 사용한다. cURL은 다양한 프로토콜을 지원하는 데이터 전송을 위한 도구로 커맨드라인 환경에서 많이 사용된다. cURL은 보통 리눅스 배포판에 기본적으로 포함되어 있어 설치가 불필요하다. cURL를 통해 REST API를 이용한 3자 전송 서버에 요청메시지를 전송하려면 우선 REST API의 메소드를 설정해야 한다. 메소드 설정은 -X 옵션 또는 -request 옵션을 통해서 설정할 수 있다. 요청메시지에 저장되는 데이터들은 Json 형식으로 보내지기 때문에 -H 옵션을 통해 보내는 데이터가 Json 형식으로 되어 있음을 서버 측에 알려준다. Json은 JavaScript Object Notation의 약자로 데이터를 저장할 때 속성-값의 쌍으로 저장하며 이렇게 속성-값의 쌍으로 이루어진 집합을 Json 객체라고 한다. Json 객체는 텍스트 형태로 이루어져 있어 사람과 컴퓨터 모두 읽고 쓰기가 간단하다는 특징을 가지고 있다. 요청메시지에서 데이터가 저장될 부분은 -d 옵션을 사용한 뒤에 Json 형식에 따라 3자 전송에 필요한 값을 저장하면 된다. 앞에서 설명한 방법으로 cURL을 사용해 REST API를 이용한 3자 전송 서버로 3자 전송을 하는 진행하는 예시는 아래의 그림 6와 같다.

```
[kisti@localhost ~]$ curl -H "Content-Type: application/json" -X POST -d '{"Source": "134.75.115.131:2811/export/ssd/100GB", "Dest": "134.75.115.133:2811/export/ssd/100GB"}' 203.253.235.158:22641/transfer
Source: ftp://134.75.115.131:2811/export/ssd/100GB
Dest: ftp://134.75.115.133:2811/export/ssd/100GB
104882765824 bytes 1171.24 MB/sec avg 967.50 MB/sec inst
```

그림 6. REST API를 이용한 3자 전송 예시
Fig. 6. Example of 3rd-party transfer using REST API

3.4 REST API를 활용한 3자 전송 방식의 성능 분석

REST API를 이용한 3자 전송 서버를 활용한 전송 테스트은 표 1에서 나타나있는 스펙의 DTN 2개를 대

전 KISTI에 설치한 후 진행하였다. 전송에 사용되는 파일은 100GB 크기의 데이터를 사용하였고 전송 방법은 4가지로 나누어 진행하였다. GridFTP에서 사용되는 전송 명령어인 globus-url-copy를 사용해 DTN1과 DTN2 양방향으로 전송해보고 이후에는 REST API를 이용한 3자 전송 서버를 설치하여 REST API 요청을 보낼 수 있는 툴인 cURL을 이용하여 DTN1과 DTN2 간의 REST API를 이용한 3자 전송을 이용하여 양방향으로 데이터 전송을 진행하였다.

각 전송 방법에 대해 총 10 차례 전송을 진행하였고 그 결과를 그림 7를 통해 나타내었다. DTN1->DTN2의 방향으로 GridFTP 클라이언트를 이용한 3자 전송을 진행하였고 그 결과 평균 전송 속도는 1087.925 MB/sec이고 최저 전송 속도는 1044.09 MB/sec 그리고 최고 전송 속도는 1115.09 MB/sec의 결과를 얻을 수 있었다. 같은 경로로 REST API를 이용한 3자 전송 서버를 통해 진행하였을 때의 전송 속

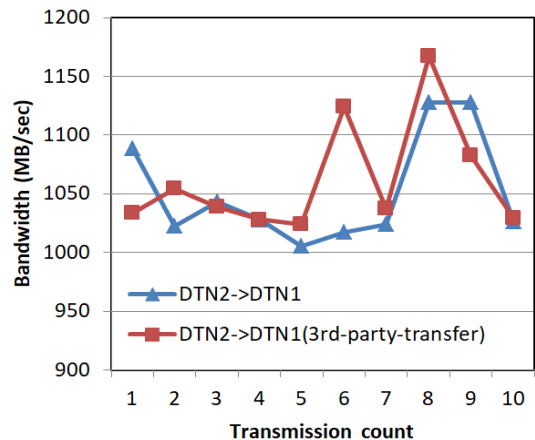
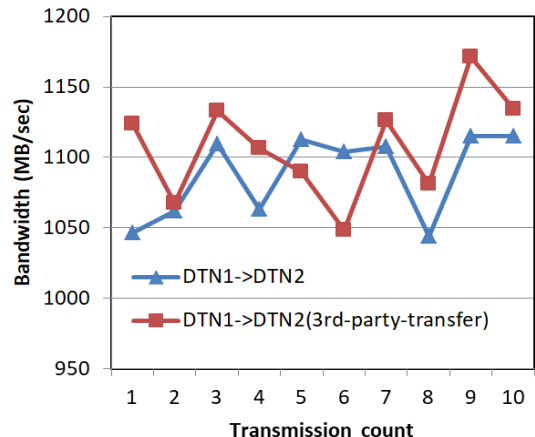


그림 7. 전송 결과
Fig. 7. Transmission result

도의 평균값은 1108.169 MB/sec 이고 최저 전송 속도는 1048.47 MB/sec 그리고 최고 전송 속도는 1171.24 MB/sec 로 기존의 방법에 비해 약간의 속도 상승이 있었다.

반대 방향으로 GridFTP 클라이언트를 사용한 데이터 전송의 평균 속도는 1050.998 MB/sec 이고 최저 속도는 1005.27 MB/sec 그리고 최고 속도는 1127.68 MB/sec의 결과를 얻을 수 있었다. 동일한 경로에서 REST API를 이용한 3자 전송 서버로 전송 작업을 진행했을 시에는 평균 속도로 1061.792 MB/sec 결과가 나왔고 최저 전송 속도는 1023.79 MB/sec 그리고 최고 전송 속도는 1167.14 MB/sec의 결과를 얻었다.

앞에서 테스트한 4가지 전송 방법에 대한 전송 결과를 비교해보면 전체적으로 볼 때 DTN1 -> DTN2의 경로가 DTN2 -> DTN1의 경로보다 평균 전송 속도의 비교 시 약간 빠른 편이었다. GridFTP 클라이언트를 사용한 3자 전송과 REST API를 이용한 3자 전송의 비교에서는 평균 및 최고 전송 속도가 REST API를 이용한 3자 전송에서 약간의 향상을 보이고 있지만 이 정도의 차이는 그림 6의 그래프에서 나타나는 것과 같이 전송 작업 시의 네트워크의 상태에 따라 달라지는 정도 오차라 REST API를 이용한 3자 전송을 통한 데이터 전송 시에도 전송 속도에는 변동이 없는 것을 확인하였다.

IV. 결 론

본 논문에서는 REST API를 이용한 3자 전송방법을 제안하였다. 기존의 3자 전송에서는 GridFTP 클라이언트인 globus-url-copy의 명령어를 사용하여 전송하는데 여기에 REST API를 사용하는 3자 전송 서버를 설치하여 REST API를 이용한 요청을 통해서도 전송이 가능하도록 하였다. REST API를 이용한 3자 전송을 통해 데이터를 전송해 본 결과 GridFTP에서 지원하는 3자 전송방법에 비해 전송속도가 감소되는 일 없이 어느 위치에서든 상관없이 3자 전송 서버에 REST API 요청을 보내면 전송할 수 있는 환경을 조성할 수 있었다. 그 결과 기존에는 불가능했던 전송망 외부에 위치한 다른 노드로부터의 3자 전송도 가능해지면서 효율성과 편의성 면에서 향상되는 결과를 얻을 수 있었다.

대용량 데이터 전송에 사용되는 전용 노드인 DTN은 100Gbps 성능을 가진 네트워크 인터페이스 카드를 장착하고 논문에서 언급한 DTN 튜닝을 통해 메모리 간 전송 시에는 98Gbps 정도의 성능을 낼 수 있지

만 실제 데이터 전송을 위한 스토리지 간 전송에서는 그보다 훨씬 못 미치는 속도인 9Gbps 정도 밖에 낼 수 없었다. 이러한 현상은 디스크의 쓰기 속도가 네트워크의 속도를 따라가지 못하기 때문인데 이를 해소하고자 고성능의 SSD로 레이드를 구성하였지만 여전히 부족한 상태이다. 향후 디스크 쓰기 속도를 향상시킬 수 있는 방안을 연구하여 이를 REST API 이용한 3자 전송에 활용할 수 있도록 하여 대용량 데이터를 쉽고 빠르게 보낼 수 있는 고성능 네트워크망 구축에 이바지할 수 있도록 할 계획이다.

References

- [1] A. Gandomi and H. Murtaza, "Beyond the hype: Big data concepts, methods, and analytics," *Int. J. Inf. Management*, vol. 35, no. 2, pp. 137-144, 2015.
- [2] S. Yu, M. Liu, W. Dou, X. Liu, and S. Zhou, "Networking for big data: A survey," *IEEE Commun. Surv. & Tuts.*, vol. 19, no. 1, pp. 531-549, 2017.
- [3] J. Chung, S. Donovan, J. Bezerra, H. Morgan, J. Ibarra, R. Clark, and H. Owen, "Novel network services for supporting big data science research," *Gateways 2017*, 2017.
- [4] M. Kiran, E. Pouyoul, A. Mercian, B. Tierney, C. Guok, and I. Monga, "Enabling intent to configure scientific networks for high performance demands," *Future Generation Computer Syst.*, vol. 79, no. 1, pp. 205-214, 2018.
- [5] M. Pramanik, W. Zhang, R. Lau, and C. Li, "A framework for criminal network analysis using big data," in *Proc. 2016 IEEE ICEBE*, pp. 17-23, 2016.
- [6] S. Gole and B. Tidke, "A survey of big data in social media using data mining techniques," in *Proc. 2015 IEEE ICACCS*, pp. 1-6, 2015.
- [7] D. Dimitrov, "Medical internet of things and big data in healthcare," *Healthcare Informatics Res.*, vol. 22, no. 3, pp. 156-163, 2016.
- [8] C. Stubbs, D. Sweeney, J. Tyson, and LSST Collaboration, "An overview of the large synoptic survey telescope (lsst) system," *Bulletin Am. Astronomical Soc.*, vol. 36, pp.

- 1527, 2004.
- [9] M. Juric and T. Tyson, "LSST data management: Entering the era of petascale optical astronomy," in *Proc. The Int. Astronomical Union*, vol. 10, no. H16, pp. 675-676, 2012.
- [10] A. Gonzalez, et al., "Big data and analysis of data transfers for international research networks using NetSage," *IEEE Int. Congress on Big Data*, pp. 344-351, Honolulu, USA, 2017.
- [11] W. Hong, J. Moon, W. Seok, and J. Chung, "Enhancing data transfer performance utilizing a DTN between cloud service providers," *Symmetry*, vol. 10, no. 4, pp. 110-110, 2018.
- [12] *The Pacific Research Platform, High-speed transmission network*, Retrieved 2019 [Online], from: <https://prp.ucsd.edu>
- [13] E. Dart, L. Rotman, B. Tierney, M. Hester, and J. Zurawski, "The ScienceDMZ: A network design pattern for data-intensive science," *Scientific Programming*, vol. 22, no. 2, pp. 173-185, 2014.
- [14] S. Lee and W. Seok, "3rd-party transmission method based on REST API for high-speed data transmission," in *Proc. KNOM*, pp. 39-40, 2018.
- [15] J. Moon, "A study on the high performance big data transfer for HPC and implementation of GridFTP based 3rd party transmission," in *Proc. KICS*, pp. 1324-1326, 2018.
- [16] *GridFTP, Transmission Program*, Retrieved 2019 [Online]. from: <http://toolkit.globus.org/toolkit>
- [17] R. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. Dissertation, Information and Computer Science, Univ. California, Irvine, USA, 2000.
- [18] S. Ong, S. Cholia, A. Jain, M. Brafman, D. Gunter G. Ceder, and K. Persson, "The materials application programming interface (API): A simple, flexible and efficient API for materials data based on REpresentational state transfer (REST) principles," *Computational Materials Sci.*, vol. 97, no. 1, pp. 209-215, 2015.
- [19] F. Haupt, "A framework for the structural analysis of REST APIs," in *Proc. 2017 IEEE ICISA*, pp. 55-58, 2017.
- [20] J. Park, J. Park, S. Kim, and M. Noh, "Performance enhancement method through Science DMZ data transfer node tuning parameters," *KTCCS*, vol. 7, no. 2, pp. 33-40, 2018.
- [21] V. Nguyen and Y. Kim, "SDN-based enterprise and campus networks: A case of VLAN management," *The J. Information Process. Syst.*, vol. 12, no. 3, 2016.
- [22] H. Kim, "REST API based server construction for web application performance analysis," *The J. Advanced Navig. Technol.*, vol. 22, no. 5, pp. 456-461, 2018.

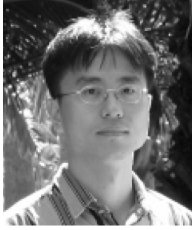
이 상 권 (Sang-gwon Lee)



2015년 2월 : 충북대학교 컴퓨터공학과 학사
 2017년 2월 : 충북대학교 컴퓨터공학과 석사
 2017년 3월~현재 : 과학기술연합대학원대학교 과학기술정보과학학과 박사과정

<관심분야> 네트워크 성능 테스트, 네트워크 튜닝, 클러스터 컴퓨팅, IoT 네트워크

석우진 (Woo-jin Kim)



1996년 2월 : 경북대학교 컴퓨터공학과 학사

2003년 2월 : Univ. North Carolina, Computer Science 석사

2008년 2월 : 충남대학교 컴퓨터공학과 박사

2003년 2월~현재 : 한국과학기술정보연구원 책임연구원

<관심분야> 네트워크 성능 향상, TCP 전송, 네트워크 테스트 방법론, 클라우드 및 컨테이너 기술, Science 네트워크/컴퓨팅

문정훈 (Jeong-hoon Moon)



1997년 2월 : 경일대학교 컴퓨터공학과 학사

1999년 2월 : 경북대학교 컴퓨터공학과 석사

2000년 3월~현재 : 한국과학기술정보연구원 선임연구원

<관심분야> 네트워크 QoS, 가상화, SDN, 클라우드 컴퓨팅, TCP 성능향상, ScienceDMZ