

Non-Disruptive Server Management for Sustainable Resource Service Based on On-Premise

Hyun-Woo Kim[†]

ABSTRACT

The rapid development of IT, many conventional passive jobs have been automated. This automation increases the leisure time of many people and various services are being developed for them. In addition, with the advent of smart devices that are compact and portable, it is possible to use various internet services without any time and place discretion. Various studies based on virtualization are under way to efficiently store and process large data generated by many devices and services. Desktop Storage Virtualization (DSV), which integrates and provides users with on-premise-based distributed desktop resources during these studies, uses virtualization to consolidate unused resources within distributed, legacy desktops. This DSV is very important for providing high reliability to users. In addition, research on hierarchical structure and resource integration for efficient data distribution storage processing in a distributed desktop-based resource integration environment is underway. However, there is a lack of research on efficient operation in case of server failure in on-premise resource integration environment. In this paper, we propose Non-disruptive Server Management (NSM) which can actively cope with the failure of desktop server in distributed desktop storage environment based on on-premise. NSM is easy to add and remove desktops in a desktop-based integrated environment. In addition, an alternative server is actively performed in response to a failure occurrence.

Keywords : Non-Disruptive Server, Sustainable Resource Service, High Availability

온-프레미스 기반 지속적인 자원 서비스를 위한 서버 무중단 기법

김 현 우[†]

요 약

최근 IT 기술의 급성장에 따라 종래의 많은 수동적인 업무가 자동화되었다. 이러한 자동화를 통해 많은 사람들의 여가 시간이 증대되고 이를 위한 다양한 서비스들이 개발되고 있다. 또한, 소형화되고 휴대에 편리한 스마트 디바이스의 등장으로 시간과 장소에 구애를 받지 않고 다양한 인터넷 서비스 이용이 가능하다. 많은 디바이스 및 서비스에서 발생하는 빅 데이터의 효율적인 저장 처리를 위해 가상화 기반의 다양한 연구들이 진행되고 있다. 이러한 연구 중에 온-프레미스 기반 분산된 데스크탑 자원을 통합하고 사용자에게 제공하는 데스크탑 스토리지 가상화는 가상화를 이용하여 분산된 레거시 데스크탑내에 사용되지 않는 자원을 통합한다. 이는 사용자에게 신뢰성 제공을 위한 고 가용성이 매우 중요시된다. 또한, 분산된 데스크탑 기반 자원 통합 환경에서 효율적인 데이터 분배 저장 처리를 위한 계층적 구조 및 자원 통합 연구들이 진행되고 있다. 그러나 온-프레미스 기반 자원 통합 환경에서 서버 고장 발생시 효율적인 대처를 위한 운영 연구가 미흡하다. 이러한 운영적 서버 고가용성을 위해 본 논문에서는 온-프레미스 기반 분산된 데스크탑내 스토리지 통합 환경에서 데스크탑 서버의 장애 발생에 따른 능동적인 대처가 가능한 무중단 서버 관리 기법을 제안한다. 이는 데스크탑 기반 통합 환경에서 데스크탑 추가 및 제거가 용이하다. 또한, 장애발생에 따라 대체 서버가 능동적으로 수행된다.

키워드 : 무중단 서버, 지속적인 자원 서비스, 고가용성

1. 서 론

최근 IT 기술의 급성장에 따라 다양한 스마트 디바이스가

개발되었다. 디지털 카메라, 스마트 TV, 스마트 폰, 태블릿 PC 등의 다양한 디바이스에서 생성되는 비정형 데이터와 문자 데이터, 영상 데이터, 위치 데이터 등의 폭발적인 데이터 증가로 빅 데이터 시대가 도래되었다. 또한 다양한 스마트 디바이스의 고성능화 및 소형화, 편리한 휴대성으로 개인의 여가시간 및 기업의 업무 효율성이 증대되었다. 이러한 디바이스의 증대는 데이터 생성 기점의 증가를 의미한다. 또한, 다

[†] 정 회 원 : 동국대학교 멀티미디어공학과 연구교수
Manuscript Received : September 20, 2018
Accepted : October 1, 2018

* Corresponding Author : Hyun-Woo Kim(hwkim@dongguk.edu)

양한 종류의 디바이스에서 제공되는 많은 서비스로 인하여 급속도로 데이터가 생성되고 있다. 현재 이러한 빅 데이터를 효율적으로 저장 및 관리하기 위한 가상화 기반의 많은 연구들이 진행되고 있다. 이러한 가상화의 종류로는 어플리케이션 가상화, 하드웨어 가상화, 데스크탑 가상화, 네트워크 가상화, 서버 가상화, 스토리지 가상화 등 다양하다. 이 중에 데스크탑 스토리지 가상화(Desktop Storage Virtualization; DSV)는 분산된 레거시 데스크탑의 스토리지 자원을 통합하고 다른 사용자의 저장 요청에 따라 제공된다[1-9].

이러한 DSV는 분산된 많은 수의 데스크탑이 연결되기 때문에 스토리지 사용자를 위한 신뢰성이 매우 중요시된다. 따라서 서버로 동작 중인 데스크탑의 결합 및 장애 등과 같은 예외 발생과 사용자의 잘못된 조작으로 인한 서버 다운에 따른 대처 기능이 필요하다. 기존의 desktop storage virtualization의 연구로는 데스크탑 기반 통합 환경에서 계층적 구조 및 자원 통합을 위한 연구들이 진행되었으나, 이러한 효율적인 운영 측면에서 연구가 미흡하다[7-15].

이러한 이유로 본 논문에서는 데스크탑 기반 스토리지 통합 환경에서 데스크탑 서버의 장애 발생에 따른 능동적인 대처가 가능한 무중단 서버 관리 기법을 제안한다. 이는 데스크탑 서버의 장애 발생에 따라 연결된 다른 데스크탑이 능동적으로 대체 서버를 수행한다. 또한 데스크탑의 추가 및 제거가 용이함에 따라 스토리지 사용자에게 높은 QoS를 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 자원 통합 기법 및 관리, 장애 발생에 따른 대처에 대한 연구를 살펴본다. 3장에서는 본 논문에서 제안하는 무중단 서버 관리 기법을 설명한다. 4장에서는 제안하는 스킴의 설계, 5장에서는 구현 및 성능평가를 설명한다. 성능 평가는 무중단 서버 관리 기법이 적용된 데스크탑 자원 통합 환경에서 서버 장애 대처에 대한 처리 속도를 평가한다. 또한, 데스크탑 서버를 인위적으로 장애를 발생시켜 최대 운영 환경을 평가한다. 마지막으로 6장에서는 전반적인 결론 요약 및 향후 연구로 구성되어 있다.

2. 관련 연구

기존의 자원 통합 기법에 대한 연구로 GFS[12], RISBD [13], CSTORE[14], Hadoop[15]에 대해 살펴보면 다음과 같다.

구글 파일 시스템(Google File System; GFS)은 접근하는 GFS 클라이언트, GFS에 저장된 접근 제어 정보, 파일과 청크의 위치 정보, 네임스페이스 등을 관리하는 GFS 마스터, 청크의 관리 및 GFS 클라이언트 요청에 따라 입출력을 담당하는 GFS 청크 서버로 이루어진 분산 파일 시스템이다. GFS 마스터를 통해 데이터 무결성을 위한 청크 복제본 생성 및 복제본 수 조정 및 사용되지 않는 저장 공간 관리 등을 수행한다. 또한, GFS 마스터의 장애 발생시 청크 서버데이터 손실을 방지하기 위해 원격의 다른 서버에 청크 서버로부터 전달 받은 메타데이터 및 변동 사항에 대한 메타데이터를 로컬 및 원격의 다른 서버에 복제한다. 그러나 연속적인 GFS 마스터의 장애 발생시 청크 서버의 데이터 손실이 발생된다.

자원 통합 시스템(Resource Integrated System for Big Data; RISBD)은 독립적으로 사용되고 있는 데스크탑 자원의 비가용 자원을 분석하고 이를 통합하여 스토리지 서비스를 제공하는 스킴을 제안하였다. 또한, 분산된 많은 데스크탑 자원의 통합으로 인해 발생할 수 있는 서버 장애의 대응 기법을 제안하였다. 그러나 IP 기반의 서버 선정으로 인해 불필요한 오버헤드가 많이 요구됨에 따라 성능 저하의 요인으로 작용되는 문제가 발생된다. 본 논문에서는 데스크탑 성능, 위치 및 저장된 자원을 고려함으로써 보다 효율적으로 서버를 선정한다.

데스크탑 지향 분산된 퍼블릭 클라우드 스토리지(Desktop-oriented Distributed Public Cloud Storage; CSTORE)는 분산된 데스크탑의 운영 측면에서 사용자의 독립적인 공간 및 데이터 보안을 보장하는 스킴을 제안하였다. 온-프레미스 기반 서버 자원 운영 측면에서 일반적인 퍼블릭 스토리지 시스템보다 확장성 및 성능 향상을 보인 반면, 서버 장애시 사용자가 접근하지 못하는 문제를 내제한다.

하둡(Hadoop Distributed File System; HDFS)은 분산된 데스크탑 각 자원인 데이터노드와 이를 관리하는 네임노드로 구성되어 있다. 이러한 구성은 수 천대의 데스크탑의 자원 통합이 가능하다. 내부적으로 데이터를 복제 수행하기 때문에 데이터 무결성이 가능하지만, 네임노드의 장애 발생시 데이터노드에 저장된 수 많은 데이터가 손실된다. 이러한 문제로 인해 네임노드의 장애 발생시 수행되는 세컨더리노드가 업데이트 되었지만, 연속적인 장애 발생시 데이터 손실을 막을 수 없는 문제를 가진다. 본 논문에서는 이러한 단점을 극복하고자 데이터 저장 위치를 가지는 메타데이터를 분산된 데스크탑에 주기적으로 동기화한다.

3. NSM 스킴

본 논문에서 제안하는 Non-disruptive Server Management (NSM) 기반 접속된 데스크탑은 서버와 클라이언트 기능을 모두 포함한다. 이러한 기능은 최초 실행시 동작 구성에 따라 서버 및 클라이언트로 구분되어 동작된다. 연결된 데스크탑은 서버의 역할을 하는 Desktop Master Node(DMN)와 사용자의 데이터 보관을 담당하는 Desktop Storage Node(DSN)로 구성된다. DMN은 동작시 데스크탑의 추가 및 삭제, 저장된 데이

Table 1. NSM Metadata

Factor	Description
IP	IP-based priorities.
Performance	CPU and the Memory should be set according to the user while ensuring that is equal to 100%.
Stored Data Amount	In operation-based on-premise environments of the NSM, connected desktops are normalized based on the desktop with the smallest amount of stored data.
Distance	Response time(Distance) is determined according to periodic heartbeats. normalization is conducted based on the minimum response time of heartbeats.

터 및 가용 스토리지 공간 등을 관리한다. DSN은 DMN에게 주기적으로 자신의 상태 정보 및 DMN의 동작 여부를 확인한다. DMN의 장애 발생시 대체 서버 선정을 위한 메타데이터는 Table 1과 같이 정의한다.

NSM이 적용된 온-프레미스 환경에서 DMN으로 수행되는 데스크탑은 Fig. 1과 같이 동작된다. 초기 서버 시작으로 데스크탑의 접속을 기다리는 Accept이 실행된다. 데스크탑이 접속됨에 따라 상태 정보인 CPU, Memory, Storage, IP, Distance를 분석하여 메타데이터를 생성하고 리스트에 반영한다. 이후에 사용자 설정에 따라 메타데이터를 정렬하고 접속된 모든 데스크탑에게 메타데이터를 전송한다. 이러한 과정은 NSM이 적용된 온-프레미스 기반 DSV 환경을 구성하는 데 있어서 초기 데스크탑 추가에 따라 반복적으로 수행된다.

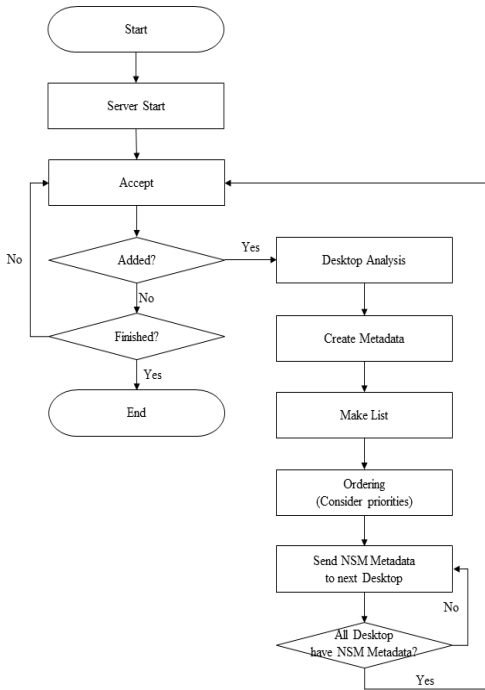


Fig 1. DMN operation flow

DSN으로 수행되는 데스크탑은 주기적으로 서버 작동 여부를 Fig. 2와 같이 확인한다. DSN은 DMN으로 접속하고 DMN으로부터 메타데이터를 수신한다. 서버의 상태 확인을 위해 하트비트를 송신하며, 서버 역할을 하는 DMN의 상태를 파악한다. 사용자가 설정한 N초 동안 주기적으로 확인한다. 기본적으로 5초마다 서버 작동 유무를 확인한다. 서버의 장애 발생에 따라 서버 수행이 불가능한 상태가 되면 Count 값이 증가한다. Count는 장애가 발생한 횟수를 의미하며 사용자가 설정한 M 횟수에 따라 이를 초과하면 서버 장애로 인식한다. 장애가 인식되면 메타데이터를 통해 자신이 대체 서버인지를 확인한다. 대체 서버라면 DMN 기능이 수행된다. 대체 서버가 아니라면 메타데이터를 통해 대체 서버로 수행되는 DMN에 접속을 시도한다. 이러한 수행은 최소 2대의 데스크탑까지 수행이 가능함에 따라 연속적인 장애 발생에 대처가 가능하다.

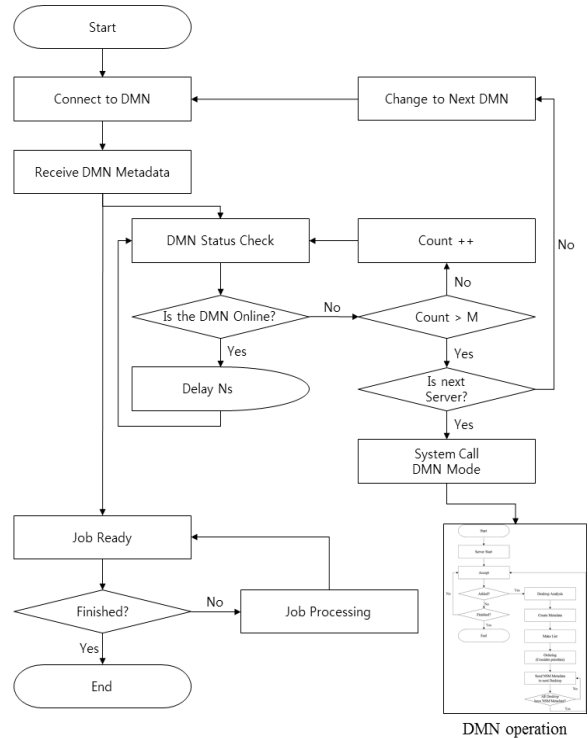


Fig 2. DSN operation flow

4. NSM 설계

User Interface는 세부적으로 Desktop Information, Duplication Count, Chunk Size, Mode, Connect Information, Priority Information을 나뉘며, Fig. 3과 같다. Desktop Information은 데스크탑의 허용 스토리지 크기, 포트 번호, 스토리지 제공으로 인한 데이터 저장 위치 등을 설정한다. Duplication Count는 접속되는 데스크탑에 비례하여 저장되는 데이터의 중복 저장 횟수를 설정한다. Chunk Size는 저장되는 데이터 크기 단위로 64MB, 128MB, 256MB, 512MB 등으로 설정한다. Connect Information은 초기 DMN에 접속하기 위해 DMN에 대한 IP, 포트 번호를 설정한다. 이러한 설정에는 DMN이 작동중인 경우인지를 확인 및 적용한다. Priority Information은 사용자가 DMN의 장애 발생시에 대체 DMN 선정에 대한 우선순위를 설정한다. 우선순위로는 IP, 성능, 저장된 데이터 크기, 거리를 고려하여 적용한다. 이때 거리는 연결된 데스크탑에 하트비트의 응답시간을 의미한다. IP는 인터넷 주소 체계 기반으로 설정된다. 성능은 데스크탑의 CPU, Memory를 고려한다. 즉, 처리 속도 및 데스크탑의 수용 수에 따라 100%를 기준으로 적용한다. 예를 들어, CPU 비중이 70%이고 Memory 비중이 30%라면 이 둘의 합이 100%인 경우에만 설정된다. 저장된 데이터 크기는 스토리지 서비스를 운영중인 경우에 저장된 데이터 총 크기가 최하인 데스크탑을 기준으로 사용자가 설정한 성능과 근접한 데스크탑을 대체 DMN으로 선정한다. 이러한 이유는 만일, DMN으로 선정된 데스크탑에 저장된 데이터의 크기가 많을 경우

에 Duplication Count를 유지하기 위해 내부적으로 컴퓨팅 및 네트워크 대역폭을 소비하기 때문이다. Mode는 최초 실행 시 사용자의 설정에 따라 DMN 또는 DSN 모드로 운영할 지를 설정한다.

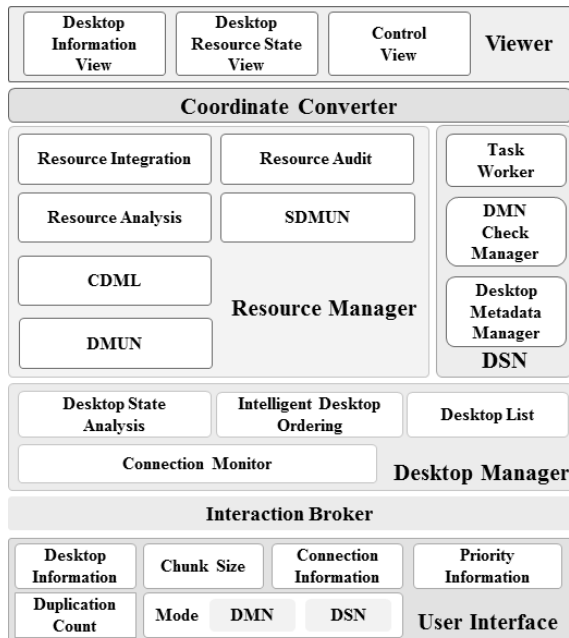


Fig 3. NSM Architecture

Interaction Broker는 User Interface를 통해 사용자로부터 전달 받은 데스크탑 기본 정보 및 장애 발생시 대응을 위한 우선순위 설정 정보를 Desktop Manager 및 Resource Manager로 전달하는 역할을 한다.

Desktop Manager는 세부적으로 Desktop State Analysis, Intelligent Desktop Ordering, Desktop List, Connection Monitor로 나뉜다. Desktop State Analysis는 Desktop의 저장된 데이터 크기 및 성능에 대해 정적 성능과 동적 성능을 분석한다. 분석된 데스크탑 상태 정보는 Intelligent Desktop Ordering으로 보내진다. Intelligent Desktop Ordering은 desktop state analysis로부터 전달받은 desktop state information를 통하여 대체 DMN을 선정하기 위한 대체 우선순위 순서를 부여한다. 정렬된 대체 DMN 리스트는 Desktop List에 저장된다. Desktop List는 대체 DMN 리스트를 Resource Manager로 보낸다.

Resource Manager는 세부적으로 Resource Analysis, Resource Integration, CDML(Create Desktop Meta Data List), DMUN(Desktop Metadata Update Notify), SDMUN(Stored Data Metadata Update Notify), Resource Audit으로 나뉜다. Resource Analysis는 데스크탑의 스토리지 정보를 분석한다. 분석으로는 DMN의 스토리지 저장 분배에 따라 저장될 스토리지 크기와 해당 DSN에 올바르게 데이터 저장이 수행되는 지 및 DSN에 설정된 허용 스토리지가 가용되는 지를 판별한다. Resource Integration은 resource analysis의 DSN의 가용 스토리지 판별에 따라 통합 스토리지에 포함한다.

다. CDML은 Desktop Manager의 Desktop List를 통해 전달 받은 DMN 리스트를 모든 데스크탑에 동기화하기 위해 DMN 메타데이터를 생성한다. 생성된 DMN 메타데이터는 DMUN에 전달된다. DMUN은 접속된 모든 데스크탑에게 DMN 메타데이터를 보내어 동기화를 수행한다. DSM은 사용자로부터 전달받은 데이터의 저장 위치를 포함하는 data 메타데이터를 모든 데스크탑에게 보내어 동기화를 수행한다. 이러한 방법은 DMN의 장애시 최적의 DSN을 선별하고 DSN에 저장된 DMN 메타데이터를 통해 무중단 서버 운영이 가능하다.

Coordinate Converter는 Viewer에 데스크탑의 상태 정보, 자원 상태를 가시화하기 위해 정보를 가공하는 역할을 한다. 가공된 정보는 Viewer에 전달된다.

Viewer는 Coordinate Converter로부터 전달받은 데스크탑 및 자원 정보와 제어 상태를 사용자에게 시각화 한다.

DSN(Desktop Storage Node)은 DMN으로부터 데이터 저장 요청을 수행하는 Task Worker와 DMN의 작동 상태를 확인하는 DMN Check Manager, DMN으로부터 전달받은 메타데이터를 관리하는 Desktop Metadata Manager로 구성되어 있다. DMN Check Manager를 통해 DMN의 서버 장애를 인지하는 경우에 Desktop Metadata Manager로부터 대체 서버를 판별하고 대체 서버를 수행함으로써 지속적인 운영이 가능하다.

5. NSM 구현 및 성능평가

본 논문에서 제안하는 NSM이 적용된 시스템의 초기 실행 화면은 Fig. 4와 같다. Fig. 4내의 ①은 초기 실행시 어떠한 모드로 수행할 지를 선택하며 Desktop Master Node (DMN)와 Desktop Storage Node (DSN)로 나뉜다. Fig. 4내의 ②는 DMN을 선택한 경우에 활성화되는 화면으로 현재의 데스크탑 정보인 IP와 포트 번호를 확인한다. 또한, Fig. 4내의 ②는 접속된 데스크탑 리스트 및 사용자의 설정 상태를 보여주며, 사용자에 의해 변경이 가능하다. Fig. 4내의 ③은 DSN으로 저장된 데이터 정보를 나타내는 데이터 리스트와 DMN 상태를 보여준다. DMN의 장애 발생에 대한 정보를 실시간으로 확인한다. 서버 장애시 대체 서버 가동을 위한 후부 DMN 리스트는 사용자가 DMN에서 설정한 대체 서버 우선순위에 따라 동기화된 메타데이터를 나타낸다. 또한, Fig. 4내의 ③은 설정을 통해 저장될 데이터의 경로 및 통합할 스토리지의 크기 설정이 가능하다.

Fig. 5는 DMN과 DSN을 포함하여 5대의 데스크탑이 통합된 환경으로써 DMN의 장애 발생에 따라 Candidate DMN List의 첫 번째 DSN이 DMN으로 동작하는 과정을 보인다. 이 때, 데스크탑은 인텔 코어 i7 3대(메모리 8GB 1대, 4GB 2대), 인텔 코어 i5 2대(메모리 4GB 1대, 메모리 6GB 1대)로 구성하였다. DMN의 Candidate DMN List는 CPU 점유율, 메모리 사용량, 저장된 데이터 크기 등이 중요시 된다. 거리의 경우에는 보통 동일한 네트워크 환경에서 구축되기 때문에 시간 차이가 크지 않다. 저장된 데이터 크기는 중복 저장된

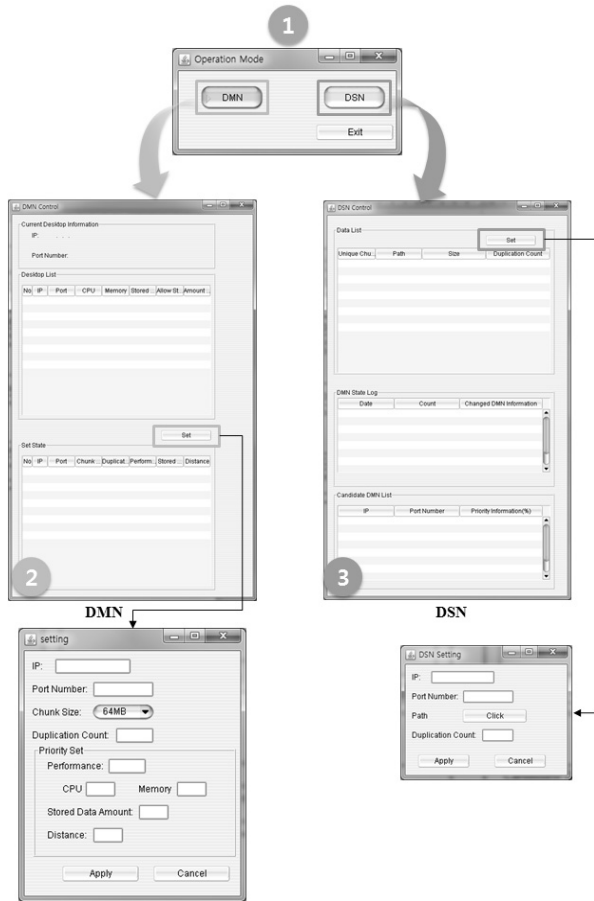


Fig 4. DMN and DSN executions according to DSM operation mode setting.

데이터 크기에 따라 성능이 좋더라도 DMN으로 선정시 데이터 복제 시간으로 서버 동작이 지연되기 때문에 청크 10개의 단위로 서버 우선순위를 적용하였다. 또한, DSN의 메타데이터 크기는 4MB로 설정하여 접속된 DSN의 메타데이터 수용 여부를 판단한다. Fig. 5내의 ①은 DMN이 정상적으로 동작하는 경우이다. Fig. 5내의 ②는 DMN의 장애 발생에 따라 DSN1, DSN2, DSN3, DSN4이 DMN과의 연결을 해제한다. Fig. 5내의 ③은 Candidate DMN List의 메타데이터 정보를 통해 1순위의 DSN을 판별한다. 1순위가 자신인 경우에 DMN으로 동작하기 위해 서버를 활성화한다. Fig. 5내의 ④는 DMN으로 선정된 DSN을 제외한 DSN이 새로운 DMN으로 연결된 모습을 나타낸다. 연결된 후에는 새로운 DMN을 통해 새로운 Candidate DMN List를 접속한 모든 DSN에게 전송하여 동기화한다. 또한, DMN은 사용자가 설정한 데이터 복제 횟수(duplication count)와 자신이 저장하고 있던 데이터의 중복 저장 수를 비교하여 조건에 만족한 지를 판별한다. 만일 만족하지 않는다면, 자신의 데이터와 동일한 데이터를 가지고 있는 DSN에게 호출하여 데이터 복제 횟수가 만족하도록 다른 DSN으로 데이터 복제를 명령한다. 이러한 반복적 과정을 통해 장애 발생시 대처가 가능하다. 서버를 삭제하는 경우에도 Fig. 5와 같이 동일하게 동작한다.



Fig 5. Non-disruptive server process in DSM

Fig. 6은 총 10개의 데스크탑이 연결된 상태로 HDFS의 마스터 노드 장애 발생시 동작 상태를 비교한다. HDFS의 경우에는 네임노드의 장애 발생시 세컨더리 서버가 수행되어 8대의 데이터 노드가 안정적으로 동작한다. 그러나, 연속적인 장애 발생에 따라 데이터 노드가 동작하지 못하는 상황이 발생된다. 본 논문에서 제안하는 무중단 서버 관리 기법은 연결된 모든 데스크탑이 서버의 기능을 내재하고 서로의 데이터 저장 정보 및 서버 우선 순위 리스트를 동기화함으로써 장애 발생시 지속적인 서버 운영이 가능하다. 이러한 기능은 DMN과 DSN이 이루어지는 최소 2대의 데스크탑 구성까지 가능하다. 이러한 무중단 서버 관리 기법은 많은 수의 데스크탑이 연결된 환경에서 서버 장애에 대한 고가용성을 제공한다.

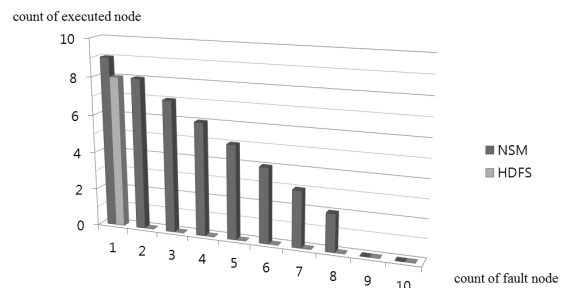


Fig 6. Comparison of the desktop operation states that master node fault has occurred.

6. 결론

본 논문에서는 종래의 연속적인 서버 장애에 대해 대처가 가능한 무중단 서버 관리 기법을 제안하였다. NSM은 DSV 환경에서 연결된 데스크탑이 서버의 기능을 내재하고 데스크탑의 성능, 거리, 저장된 데이터양을 기준으로 대체 서버를

선정을 위한 서버 메타데이터를 생성하였다. 이러한 방법은 연결된 데스크탑 중에 최적화된 서버를 선정함으로써 서버 재운영을 위한 지연시간을 줄인다.

향후에는 서버 고가용성뿐만 아니라, 데이터의 용도별 효율적인 저장 및 추출을 위한 알고리즘을 연구하고자 한다. 또한, 네트워크 스위치 구성에 있어서 알고리즘을 적용시키기 위한 연구를 하고자 한다.

References

[1] Mohammad Isam Malkawi, "The art of software systems development: Reliability, Availability, Maintainability, Performance (RAMP)," *Human-centric Computing and Information Sciences*, Vol.3, No.22, pp.1-17, Dec. 2013.

[2] Eun-Ha Song, Hyun-Woo Kim, and Young-Sik Jeong, "Visual Monitoring System of Multi-Hosts Behavior for Trustworthiness with Mobile Cloud," *Journal of Information Processing Systems*, Vol.8, No.2, Jun. 2012.

[3] Sun-Ho Lee and Im-Yeong Lee, "A Secure Index Management Scheme for Providing Data Sharing in Cloud Storage," *Journal of Information Processing Systems*, Vol.9, No.2, pp. 287-300, Jun. 2013.

[4] Nitesh Shrivastava and Ganesh Kumar, "A survey on cost effective multi-cloud storage in cloud computing," *International Journal of Advanced Research in Computer Engineering and Technology*, Vol.2, Issue 4, Apr. 2013.

[5] So-Yeon Kim, Hong-Chan Roh, Chi-Hyun Park, and Sang-Hyun Park, "Analysis of Metadata Server on Clustered File Systems," in *Proceedings of the Korea Computer Congress 2009*, KCC, Vol.36, No.1, pp. 64-69. Jul. 2009.

[6] Pradnya Eknath Gaonkar and Sachin Bojewar, Jayesh Ajit Das, "A Survey: Data Storage Technologies," *International Journal of Engineering Science and Innovative Technology*, Vol.2, No. 2, pp.547-554, Mar. 2013.

[7] Garth A. Gibson and Rodney Van Meter, "Network attached storage architecture," *Communications of the ACM*, Vol.43, No.11, pp.37-45, Nov. 2000.

[8] B. Dong, Q. Zheng, F. Tian, K. Chao, R. Ma, and R. Anane, "An optimized approach for storing and accessing small files on cloud storage," *Journal of Network and Computer Applications*, Vol.35, No.6, pp.1847-1862, Nov. 2012.

[9] Manojit Chattopadhyay, Pranab K. Dan, and Sitanath Mazumdar, "Comparison of visualization of optimal clustering using self-organizing map and growing hierarchical self-

organizing map in cellular manufacturing system," *Applied Soft Computing*, Vol.22, pp.528-543, Sep. 2014.

[10] José Luís Silva, José Creissac Campos, and Michael D. Harrison, "Prototyping and analysing ubiquitous computing environments using multiple layers," *International Journal of Human-Computer Studies*, Vol.72, No.5, pp.488-506, May. 2014.

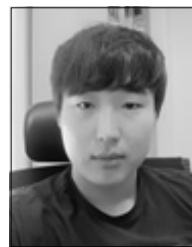
[11] Javed Mohammed, "Evolution of the Next Generation of Technologies: Mobile and Ubiquitous Computing," *International Journal of Advanced Research in Science, Engineering and Technology*, Vol.1, No.5, pp.247-253, Dec. 2014.

[12] Mengdi Wang, Bo Li, Yongxin Zhao, and Geguang Pu, "Formalizing Google File System," in *Proceedings of 2014 IEEE 20th Pacific Rim International Symposium Dependable Computing*, Singapore, Singapore, 18-21 Nov. 2014. pp.190-191

[13] Hyun-Woo Kim, Jong Hyuk Park, and Young-Sik Jeong, "Human-centric storage resource mechanism for big data on cloud service architecture," *Journal of Supercomputing*, Vol. 72, Issue 7, pp.2437-2452, Jul. 2016.

[14] Hancong Duan, Shiyao Yu, Mei Mei, Wenhan Zhan, and Lin Li, "CSTORE: A desktop-oriented distributed public cloud storage system," *Computers and Electrical Engineering*, Vol.42, pp.60-73, Feb. 2015.

[15] Jeffrey Shafer, Scott Rixner, and Alan L. Cox, "The Hadoop Distributed Filesystem: Balancing Portability and Performance," in *Proceedings of IEEE International Symposium on Performance Analysis of System and Software (ISPASS 2010)*, White Plains, New York, 28-30 Mar. 2010, pp.122-133.



김 현 우

<https://orcid.org/0000-0001-8295-2598>

e-mail : hwkim@dongguk.edu

2005년 원광대학교 전기전자 및

정보공학부(공학사)

2016년 동국대학교 멀티미디어공학과

(공학박사)

2018년~현 재 동국대학교 멀티미디어공학과 연구교수

관심분야 : Cloud Computing, Internet of Things, Mobile Edge

Computing, Information Security