KIPS Trans. Comp. and Comm. Sys.
Vol.12, No.4 pp.135~142
ISSN: 2287-5891 (Print), ISSN 2734-049X (Online)

https://doi.org/10.3745/KTCCS.2023.12.4.135

A Study on Optimizing Disk Utilization of Software-Defined Storage

Lee Jung Il† · Choi YoonA†† · Park Ju Eun†† · Jang Minyoung†††

ABSTRACT

Recently, many companies are using public cloud services or building their own data center because digital transformation is expanding. The software-defined storage is a key solution for storing data on the cloud platform and its use is expanding worldwide. Software-defined storage has the advantage of being able to virtualize and use all storage resources as a single storage device and supporting flexible scale-out. On the other hand, since the size of an object is variable, an imbalance occurs in the use of the disk and may cause a failure. In this study, a method of redistributing objects by optimizing disk weights based on storage state information was proposed to solve the imbalance problem of disk use, and the experimental results were presented. As a result of the experiment, it was confirmed that the maximum utilization rate of the disk decreased by 10% from 89% to 79%. Failures can be prevented, and more data can be stored by optimizing the use of disk.

Keywords : Software Defined Storage, Cloud Platform, Optimization

소프트웨어 정의 스토리지의 디스크 이용을 최적화하는 방법에 관한 연구

이 정 일† · 최 윤 아†† · 박 주 은†† · 장 민 영†††

요 약

최근에는 디지털 변환이 확대됨에 따라 많은 기업들이 퍼블릭 클라우드 서비스를 이용하거나 자체 데이터센터를 구축하고 있다. 소프트웨어 정의 스토리지는 클라우드 플랫폼에서 데이터를 저장하기 위한 핵심적인 솔루션으로 전세계적으로 이용이 확대되고 있다. 소프트웨어 정의 스토리지는 전체 스토리지 자원을 하나의 저장장치와 같이 가상화하여 사용할 수 있고 유연한 Scale-out을 지원하는 장점이 있는 반면에, 가변 크기의 오브젝트 방식으로 인한 디스크의 이용에 불균형이 발생하고, 장애를 유발할 수 있다. 본 연구에서는 디스크 이용의 불균형 문제를 해결하기 위하여 스토리지의 상태정보를 바탕으로 디스크의 가중치를 최적화하여 오브젝트를 재분배하는 방법에 대하여 제안하고, 그 실험 결과를 제시하였다. 실험을 수행한 결과, 디스크의 최대 이용률이 89%에서 79%로 10%만큼 감소한 것을 확인하였다. 디스크의 이용률을 최적화함으로써 장애를 예방하고, 더 많은 데이터를 균등하게 저장할 수 있어 효율적인 스토리지 이용이 가능할 것으로 기대된다.

키워드 : 소프트웨어 정의 스토리지, 클라우드 플랫폼, 최적화

1. 서 론

소프트웨어 정의 스토리지(SDS: Software Defined Storage)란 전체 스토리지 자원을 하나의 저장장치처럼 사용할 수 있도록 해주는 소프트웨어를 말한다[1,2]. 디스크를 늘리면 스토리지를 쉽게 확장할 수 있는 장점 때문에 ICT 자원의 유연한 운영을 지원하고, 활용도를 높일 수 있게 해준다[3,4]. 데스크탑 클라우드 또는 서버 클라우드 등 클라우드 플랫폼은 사용자 증가 시 유연한 Scale-out이 요구되기 때문에, 확장성이 뛰어난 소프트웨어 정의 스토리지를 이용한다[5].

소프트웨어 정의 스토리지에 저장되는 데이터는 가변 크기의 특성을 가지는 오브젝트로 변환된다. 오브젝트의 최대 크기는 (2MB, 4MB 등)으로 설정이 가능하며, 데이터 파일이 오브젝트의 최대 크기보다 큰 경우에는 여러 개의 오브젝트로 분할된다. 각 오브젝트는 해쉬 함수를 이용하여 어느 디스크에 저장될지 결정된다. 해쉬 함수의 출력은 균등하게 배분되기 때문에 디스크에 저장되는 오브젝트의 개수는 큰 차이가 없다. 하지만, 디스크별로 오브젝트의 크기가 다르기 때문

※ 이 논문은 한국전력공사 전력연구원의 디지털 플랫폼 정합화 및 전력 AI엔진 기술 개발 연구과제에 의하여 연구되었음.
† 정 회 원 : 전력연구원 선임연구원
†† 비 회 원 : 전력연구원 일반연구원
††† 비 회 원 : 전력연구원 선임연구원
Manuscript Received : December 16, 2022
First Revision : February 1, 2023
Accepted : February 28, 2023
* Corresponding Author : Park Ju Eun(jueun_park@kepco.co.kr)

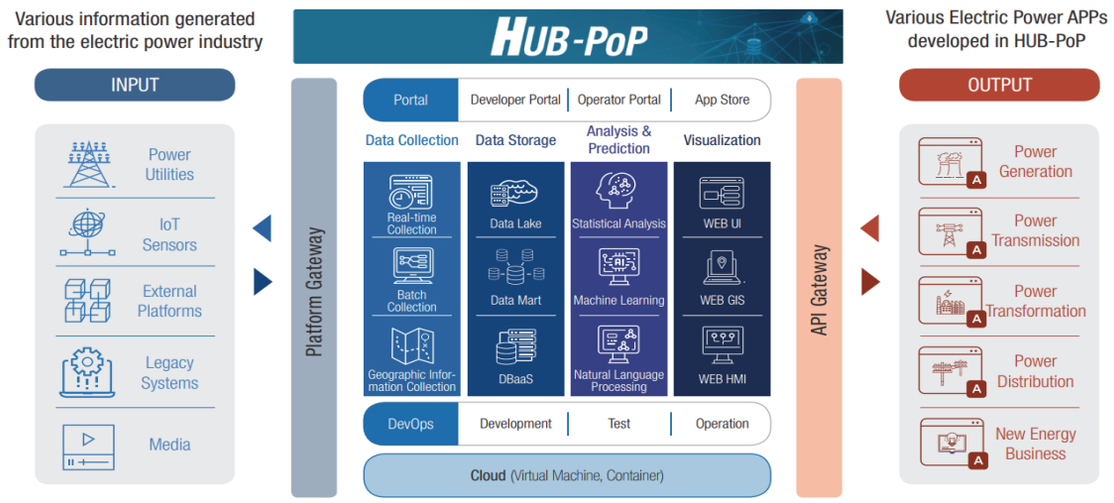


Fig. 1. HUB-PoP Cloud Platform

에, 디스크별로 데이터 이용량은 다를 수 있다.

스토리지의 이용률이 높은 경우(약 70% 이상)인 경우에는 크기가 큰 오브젝트가 더 많이 저장된 디스크의 이용률이 100%에 도달하여 에러가 발생할 수 있다. 디스크에 에러가 발생하면 클라우드 플랫폼의 가상서버에 접속할 수 없는 장애가 발생한다. 클라우드 플랫폼의 장애는 기업에 비용적인 손실을 초래할 수 있으므로 예방이 필요하다. 본 연구에서는 이러한 문제점을 해결하기 위하여 소프트웨어 정의 스토리지에 장착된 각 디스크의 이용률을 고려하여 가중치를 최적화하고, 오브젝트를 재분배함으로써 디스크 이용을 최적화할 수 있는 방법을 제안하고, 실험 결과를 제시한다.

2. 관련 연구

2.1 클라우드 플랫폼(HUB-PoP)

한국전력에서는 오픈소스로 공개된 Openstack[6]과 Kubernetes[7]를 이용하여 클라우드 플랫폼(HUB-PoP)[8]을 Fig. 1과 같이 자체적으로 구축하여 전사적인 연구개발에 활용 중이다. Openstack을 이용하여 서버 가상화를 구현하였으며, 각 프로젝트에서 요구하는 가상서버를 쉽게 생성하고 삭제하는 기능이 포함된다. 또한, Kubernetes를 이용하여 컨테이너 가상화 플랫폼[9]을 구현하였고, 마이크로 서비스 아키텍처[10] 기반으로 소프트웨어를 개발하여 배포하고 운영할 수 있는 환경을 제공한다. 클라우드 플랫폼의 인프라는 고성능 서버 89대와 네트워크 스위치 15대로 구성된다. Table 1은 클라우드 플랫폼에서 이용할 수 있는 가상자원 규모를 나타낸다.

한국전력에서는 클라우드 플랫폼의 운영을 위하여 소프트웨어 정의 스토리지인 Ceph 스토리지를 이용 중이다. 스토리지의 전체 디스크의 평균 이용률은 69%이지만, 전체 디스크의 이용률의 분포는 57%에서 89%로 넓게 분산되어 있다. 오브젝트의 가변 크기로 인하여 디스크별 이용량이 다르게 운영되기 때문이다. 디스크의 이용률이 85% 이상인 경우에는 경고 상태가 되며, 이용률이 85% 이상으로 높은 디스크에 크기가 큰 오브젝트가 추가로 저장되면 디스크가 가득차서 장애가 발생할 위험이 있다. 이러한 디스크 이용률 불균형 문제로 인하여 최근에 이용률이 높았던 디스크가 가득차서 클라우드 플랫폼의 장애가 발생한 사례가 있다. 장애가 발생한 원인은 클라우드 플랫폼의 데이터를 저장하는 스토리지의 특정 디스크 1개가 가득차서 해당 디스크를 사용하는 가상서버의 파일 시스템이 Read-only 모드로 전환되었기 때문이다[11]. 파일 시스템이 Read-only 상태가 되면 가상서버의 원격접속과 서비스가 정지된다. 가상서버의 모든 서비스는 파일 시스템에 이력을 저장하기 때문에, 파일 시스템이 쓰기 모드여야만 동작이 가능하다.

Table 1. Cloud Resource Status

Resources	CPU (Cores)	Memory (GB)	GPU (Devices)	Storage (TB)
Max capacity	3,088	17,269	62	833

2.2 Ceph Storage

Ceph[12]는 오픈소스로 공개된 소프트웨어 정의 스토리지이다. 주로 서버 가상화 플랫폼인 Openstack과 컨테이너 가상화 플랫폼인 Kubernetes의 스토리지로 이용된다. Ceph는 소프트웨어 정의 스토리지이기 때문에 서버의 제조사와 성능에 상관없이 설치하여 스토리지를 구성할 수 있다. 또한, 스토리지 서버와 디스크를 확장하면 쉽게 스토리지 용량을 늘릴 수 있다[13].

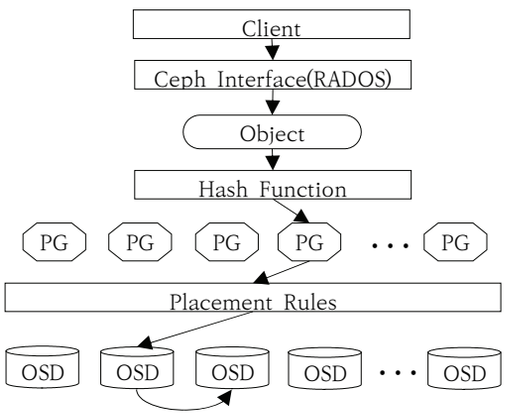


Fig. 2. Ceph Data Storage Process

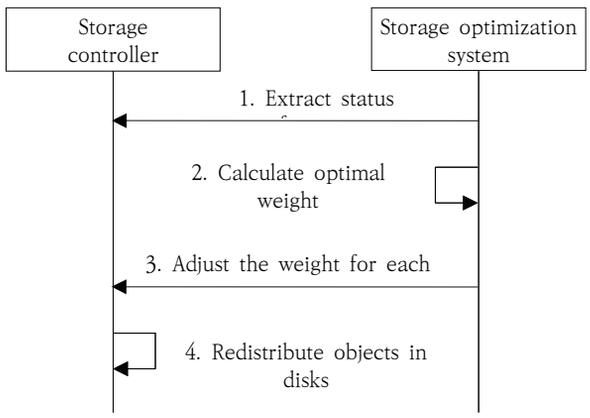


Fig. 3. System Process

Fig. 2와 같이 Client는 Ceph의 인터페이스인 RADOS (Reliable Automatic Distributed Object Store)[14]를 통하여 데이터를 입력하며, 입력된 데이터는 가변 크기인 오브젝트로 변환된다[15]. 각 오브젝트는 해쉬 함수를 통하여 어느 PG(Placement Group)에 포함될지가 결정된다. 그리고 각 PG는 배치와 복제 규칙에 따라 저장될 OSD(Object Storage Daemon)가 결정되며, 데이터 유실을 방지하기 위한 복제 기능인 Erasure code[16]도 지원한다.

2.3 소프트웨어 정의 스토리지 최적화

소프트웨어 정의 스토리지는 클라우드의 핵심 구성요소로 활용되고 있다. 세계적으로 가장 많이 이용되고 있는 Openstack에서도 Ceph를 소프트웨어 정의 스토리지로 이용한다[17]. 이러한 소프트웨어 정의 스토리지의 최적화를 위하여 다양한 연구가 존재한다. 먼저, 오픈스택 클라우드의 가용성과 효율성을 높이기 위해서 계층적인 클라우드 시스템이 제안되었으며, 클라우드 스토리지로 Ceph, HDFS, Swift를 통합하여 파일을 최적으로 배분하는 방법이 제안되었다[18]. Tianru Zhang 등은[19] 강화학습 기반으로 동적인 마이그레이션 정책을 통한 스토리지 최적화 연구를 진행하였다. Pieter-Jan Maenhaut 등은[20] 계층적인 클라우드 테넌트 트리를 기반으로 스토리지 재배치 알고리즘을 제안하였다.

3. 제안 방법

Fig. 3은 본 연구에서 제안하는 스토리지 디스크의 이용을 최적화하는 절차를 나타낸다. 첫째, 스토리지에서 제공하는 Restful API를 통하여 스토리지의 상태, 오브젝트 불일치율, 디스크 가중치, 디스크 이용현황(이용량, 가용용량)을 추출한다. 둘째, 디스크의 이용량 통계를 바탕으로 각 디스크의 가중치를 베이지안 최적화를 이용하여 최적으로 계산하고, API를 통하여 디스크별 가중치를 컨트롤러에 입력한다.

가중치가 입력되면 스토리지 내부적으로 각 디스크의 오브젝트를 재분배한다. 재분배는 가중치가 낮은 디스크의 오브젝트가 가중치가 높은 디스크로 이동하면서 이루어진다. 디스크 이용률이 평균 이용률보다 높아서 가중치가 낮게 조정된 디스크에 저장된 오브젝트는 가중치가 높은 디스크로 이동하게 된다. 이를 통하여 오브젝트가 재분배되며, 이용률이 높은 디스크의 이용률을 낮출 수 있다.

각 디스크의 이용률을 고려한 가중치를 계산하는 방법은 알고리즘 1과 같다. 디스크의 이용률 정보를 바탕으로 각 디스크의 가중치를 최적화 하기 위하여 베이지안 최적화[21]를

Algorithm 1 Optimize Weight of Disks

Input : Disk usage rate U_i

Initialize S ← 0, M ← []
P ← {a←[0.1 to 1.0], b←[2 to 10])
$\mu \leftarrow \frac{1}{n}\sum_{i}^{n} U_i$
X ← InitPoint(P)
for a_t, b_t in X do
  f_t ← Score(a_t, b_t, μ)
  M.insert($((a_t, b_t), f_t)$
end for
for t = 1, ..., N do
  a_t, b_t ← AcquisitionFunction(P, M)
  f_t ← Score(a_t, b_t, μ)
  M.insert($((a_t, b_t), f_t)$
  if f_t 〉 $S_{\max}$ then
    $S_{\max} \leftarrow f_t$, $\hat{a} \leftarrow a_t$, $\hat{b} \leftarrow b_t$
end for
$\widehat{W}_i \leftarrow \hat{a} \times U_i^{\hat{b}}$, **for** all i
return $\widehat{W}_i$

Table 2. Optimization Search Space

Variable	Min	Max	Interval
Coefficient (a)	0.1	2.0	0.0001
Degree (b)	2	20	1

이용한다. 베이지안 최적화란 입력 x에 대하여 목적함수 $f(x)$를 최대로 만드는 최적의 파라미터를 찾는 방법이다. 베이지안 이론[22]을 기반으로 현재까지 알려진 선행확률(Prior)과 실험을 통하여 구한 가능성(Likelihood)를 통해 후행확률(Posterior)를 추정한다.

베이지안 최적화를 이용하여 스토리지의 상태정보에 적합한 계수 a와 차수 b의 최적 해를 구하기 위하여, 먼저 최적의 계수 a와 차수 b를 탐색하기 위한 탐색 공간 P를 Table 2와 같이 정의한다. 초기 탐색 시에는 베이지안 최적화의 InitPoint 함수를 통하여 탐색 공간의 구간을 나누어 계수 a와 차수 b를 선정한다. 선정된 a_t와 b_t를 이용하여 목적함수 f_t를 구하고, 그 결과를 모델 M에 추가한다.

다음으로는 초기 탐색의 결과인 모델 M을 가우시안 프로세스[23]를 통하여 확률적인 추정을 강화하며, 이를 바탕으로 다음으로 탐색할 후보를 AcquisitionFunction 함수를 통하여 선정한다. N번 만큼의 반복 학습을 수행하며, 선정된 파라미터를 통하여 계산된 목적함수 f_t가 최대인 파라미터를 탐색하는 것을 목적으로 한다. 반복 학습을 통하여 최적으로 도출된 최적의 계수 $\hat{a}$와 최적의 차수 $\hat{b}$에 각 디스크의 이용률 U_i를 입력하여 가중치 $\widehat{W_i}$를 계산하는 식은 Equation (1)과 같다. Equation (1)에 각 디스크의 이용률을 대입하여 가중치를 계산하고, 계산된 가중치를 스토리지 컨트롤러에 입력하여 각 디스크의 오브젝트를 재분배한다.

$$\widehat{W_i} = 1 - \hat{a} \times U_i^{\hat{b}} \tag{1}$$

알고리즘 2는 베이지안 최적화 학습과정에서 최적의 해를 구하기 위한 최적화 점수를 계산하기 위한 Score 함수를 정의한다. 최적화 과정에서 추천된 계수 a와 차수 b, 그리고 스토리지 디스크의 이용률 U_i가 입력되고, 최적화 점수 S가 출력된다. 배열 X는 디스크 이용률을 나타내며, 0부터 1까지 0.01 단위로 세분화하여 선언한다.

배열 X에 포함된 모든 값 x에 대하여 가중치 y를 계산한다. x가 평균 μ보다 작거나 같으면 y를 Y_1 배열에 삽입하고, 평균 μ보다 크면 y를 Y_2 배열에 삽입한다. Y_2의 원소는 최소 가중치 $W_{\min}$보다는 크거나 같으면 최적화 점수를 계산하여 반환하고, 아니면 0을 반환한다. 최적화 점수 S의 계산 시 γ는 평균 이용률을 초과하는 값에 대한 우선 가중치이다.

Algorithm 2 Score

Input : Coefficient a, Degree b, Disk average usage rate μ

```
Initialize S ← 0
X ← {0, 0.01, 0.02, ..., 0.98, 0.99, 1.0}
for x in X do
    y ← 1 − a × x^b
    if x ≤ μ then
        Y1.insert(y)
    else
        Y2.insert(y)
end for
if min(Y2) ≥ W_min
    S ← −σ(Y1) n(Y1)/n(X) + σ(Y2) n(Y2)/n(X) (1+γ)
return S
```

γ가 높을수록, 평균을 초과하는 이용률의 표준편차에 높은 가중치를 부여한다. σ는 표준편차이며, 수식은 Equation 2와 같다.

$$\sigma(X) = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n}}, x \in X \tag{2}$$

최적으로 계산된 각 디스크의 가중치를 스토리지 컨트롤러의 Restful API를 통하여 입력함으로써 각 디스크의 오브젝트를 재분배할 수 있다. 스토리지의 상태는 수시로 변화하므로 주기적으로 오브젝트를 재분배할 것이다. 다만, 디스크의 가중치가 변경되면 디스크별로 오브젝트가 추가적으로 이동해야 하기 때문에 오브젝트 불일치율(Object misplaced ratio)이 증가할 수 있다. 오브젝트 불일치율이 증가하면 더 많은 오브젝트가 재분배 되기 때문에 스토리지의 성능을 저하시킬 수 있다. 따라서, 스토리지의 오브젝트 불일치율을 고려하여 가중치를 변경하여야 한다. 따라서, 가중치를 변경 가능한 오브젝트 불일치율 상한선을 정하고 이 상한선보다 작은 경우에만 가중치를 입력하여 오브젝트 재분배를 최소화한다. 예를 들어 오브젝트 불일치율 상한선이 3%라면, 현재 스토리지의 오브젝트 불일치율이 3% 이하인 경우에만 가중치를 입력하고 오브젝트를 재분배한다.

다음으로, 각 디스크별로 계산된 가중치가 기존 가중치와의 큰 차이가 없는 경우에 가중치를 수시로 변경하면 스토리지의 성능만 저하시키고, 오브젝트 재분배 효과는 미미할 것이다. 따라서, 각 디스크별로 계산된 가중치와 기존에 입력된

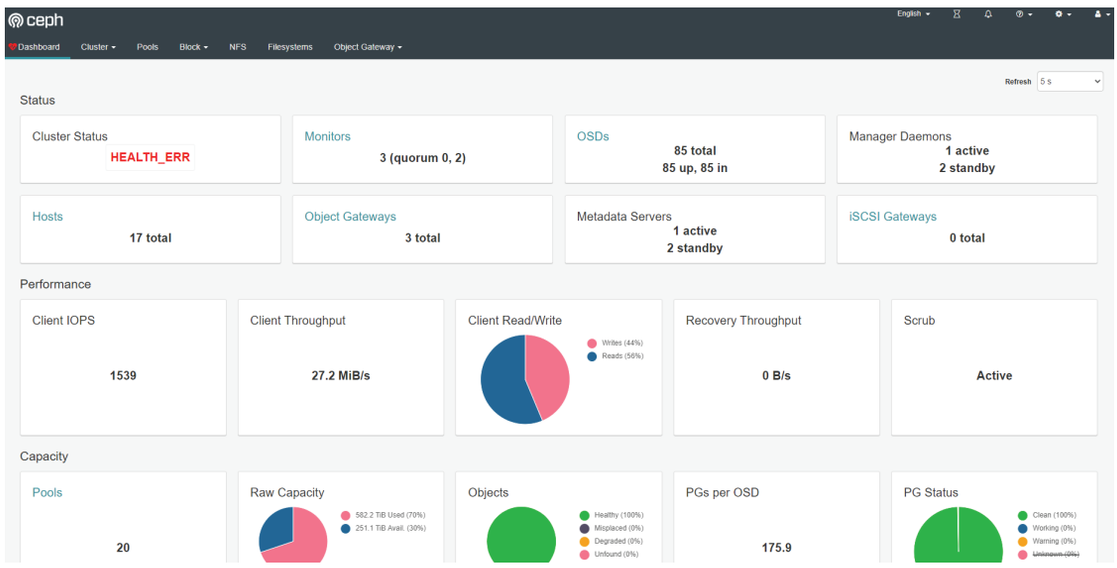


Fig. 4. Ceph Dashboard

가중치의 차이를 구하고, 그 차이가 하한선보다 큰 경우에만 가중치를 입력하여 오브젝트 재분배를 최소화한다. 예를 들어 디스크의 하한선이 0.01이라면, 가중치 차이가 0.01 이상인 경우에만 가중치를 입력하고 오브젝트를 재분배한다.

4. 실험 및 결과

4.1 실험 환경

본 연구에서 제안하는 디스크 이용 최적화 방법의 효과를 증명하기 위하여 실제 운영 중인 Ceph 스토리지 시스템을 대상으로 실험을 진행하였다. 실험에 사용된 Ceph 스토리지의 버전은 nautilus(14.2.5)이고, 서버 9식으로 구성되며, 장착된 디스크는 10.9TB HDD 61개, 7TB SSD 24개이고, 최대용량 833TB에서 582TB인 69%를 이용 중이다. 각 서버의 사양은 고속처리를 위하여 CPU 2.7Ghz 36Core, 256GB RAM, 10Gbps Network로 구축되어 있다. Fig. 4는 Ceph 스토리지의 대시보드이다.

4.2 실험 결과

실험을 위한 하이퍼 파라미터인 최적화 반복 횟수는 300회, 최소 가중치($W_{\min}$)는 0.5, 우선 가중치(γ)는 0.5로 설정하였다. 반복 학습 과정마다 최적화 점수(S)가 계산되며, 최적화 점수가 가장 높은 계수(a)와 차수(b)가 최적해로 선정된다. Fig. 5는 반복 학습과정에서의 최적화 점수를 기록한 결과이다. 146번째 학습에서 a가 0.5307이고, b가 8일 때 최적화 점수가 0.05762로 가장 높게 나타났다. 최적화 결과를 반영한 디스크의 가중치 계산식은 Equation (3)과 같다.

$$\widehat{W}_i = 1 - 0.5307 \times U_i^8 \tag{3}$$

Fig. 6은 Ceph의 각 디스크의 이용률(U_i)을 입력하여 계산된 조정 가중치($\widehat{W}_i$)의 분포를 나타내며, X축은 디스크 이용률이고, Y축은 가중치이다. X축이 0부터 디스크 이용률 최소값인 0.57까지는 1에 가깝고, 평균인 0.69부터 1까지 가중치가 가파르게 감소한다.

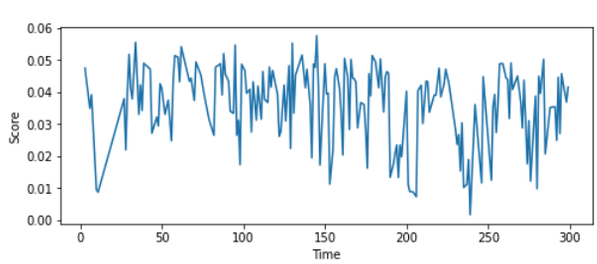


Fig. 5. Optimization Score History

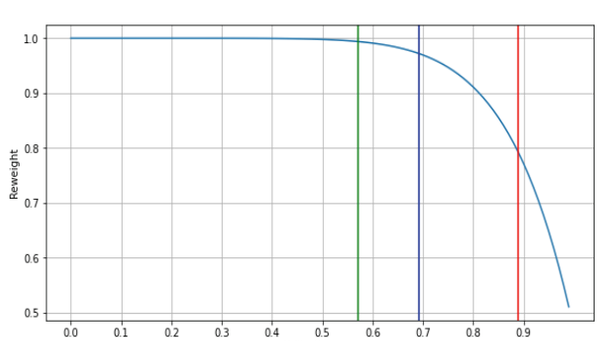
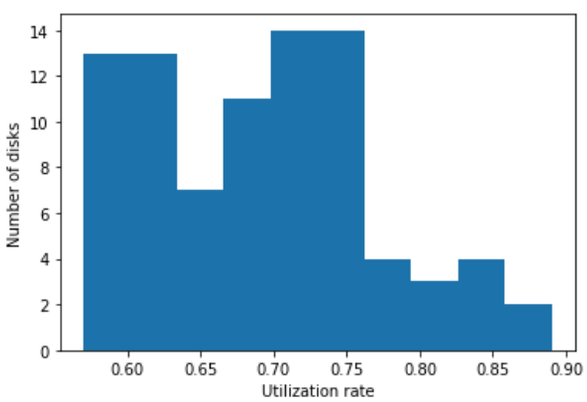


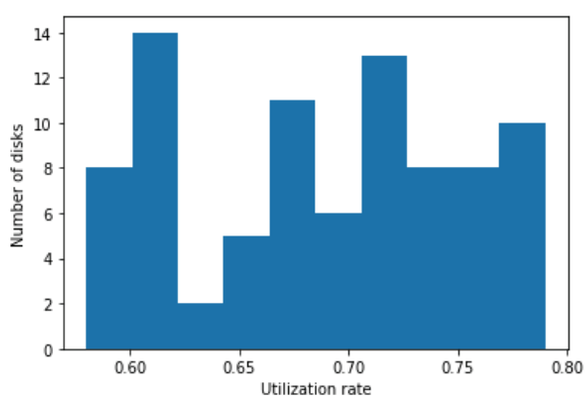
Fig. 6. Disk Weight Curve

Table 3. Disk Utilization Statistics

	Std	Min	Mean	Max
Before	0.0780	0.57	0.69	0.89
After	0.0606	0.58	0.69	0.79



(a)



(b)

Fig. 7. Disk Utilization Histogram: (a) Before (b) After

Table 3은 오브젝트 재분배 전후의 디스크 이용률에 대한 통계정보를 나타낸다. 디스크의 오브젝트를 재분배한 결과, 디스크 이용률의 표준편차는 0.078에서 0.0606으로 0.0174만큼 감소하였고, 최대 이용률은 89%에서 79%로 10%만큼 감소하였다. 가중치를 조정함으로써 각 디스크의 오브젝트가 재분배되었고, 디스크의 최대 이용률을 낮출 수 있다는 것을 확인하였다.

Fig. 7은 오브젝트 재분배 전후의 디스크의 이용률에 대한 빈도수를 나타내는 히스토그램이다. X축은 디스크 이용률이고, Y축은 각 디스크 이용률에 해당하는 디스크 개수를 나타낸다. 오브젝트를 재분배하기 전에 이용률이 80% 이상인 9개의 디스크의 이용률이 모두 80% 미만으로 조정되었다. 오브젝트 재분배를 통하여 디스크의 이용률의 분포가 57~89%에서 58~79%로 개선된 것을 확인하였다.

5. 결 론

소프트웨어 정의 스토리지는 전체 스토리지 자원을 하나의 저장장치와 같이 가상화하여 사용할 수 있고 유연한 Scale-out을 지원하는 장점이 있는 반면에, 가변 크기의 오브젝트 방식으로 인한 디스크의 이용에 불균형이 발생하고, 장애를 유발할 수 있는 단점이 있다. 본 연구에서는 디스크 이용의 불균형 문제를 해결하기 위하여 스토리지의 상태정보를 추출하고, 추출된 상태정보를 바탕으로 각 디스크의 가중치를 최적화하여 오브젝트를 재분배하는 방법에 대하여 제안하고, 그 실험 결과를 제시하였다. 실험환경으로는 한국전력공사 전력연구원에서 연구개발용으로 운영 중인 Ceph 스토리지 클러스터가 이용되었다. 스토리지의 전체용량은 833TB이고, 디스크의 평균 이용률은 69%이다. 실험을 수행한 결과, 디스크 이용률의 표준편차는 0.078에서 0.0606으로 0.0174만큼 감소하였고, 분포는 57~89%에서 58~79%로 개선되었으며, 최대 이용률은 89%에서 79%로 10%만큼 감소하였다. 다른 디스크에 비하여 이용률이 높은 디스크의 오브젝트를 이용률이 낮은 디스크로 재분배함으로써 최대 디스크 이용률을 낮출 수 있었다. 스토리지를 구성하는 디스크의 이용률을 최적화하면 클라우드 플랫폼의 장애를 예방하고, 더 많은 데이터를 저장할 수 있게 되어 효율적인 스토리지 이용이 가능하다. 향후에는 디스크 이용의 최적화 개선을 위하여 인공지능 기반의 강화학습을 이용한 디스크 이용 최적화 알고리즘의 적용에 대한 연구를 수행할 예정이다.

References

[1] Coraid, 2013, The Fundamentals of Software-Defined Storage [Internet], http://san.coraid.com/

[2] S. Robinson, 2013, Software-Defined Storage: The Reality Beneath the Hype [Internet], http://www.computerweekly.com/

[3] G. Joshi, E. Soljanin, and G. Wornell, "Efficient redundancy techniques for latency reduction in cloud systems," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, Vol.2, No.2, pp.12, 2017.

[4] H. X. Mao, X. L. Shu, K. Huang, and L. Zhang, "Research of data reliability technology based on erasure code redundancy technology in cloud storage," *Advanced Materials Research*, Vols.912-914, pp.1345-1348, 2014.

[5] M. Peters and M. Keane, "Key reasons to use software-defined storage and how to get started," *IBM Whitepaper, IBM.com website*, pp.1-8, 2015.

[6] T. Rosado and J. Bernardino, "An overview of openstack architecture," *Proceedings of 18th International Database Engineering & Application Symposium*, pp.366-367, 2014.
[7] Kubernetes Cluster Architecture [Internet], https://kubernetes.io/docs/concepts/architecture
[8] J. Lee et al., "Development of the KEPCO electric power software common platform technology," www.kepri.re.kr, 2021.
[9] D. Bernstein, "Containers and cloud: From lxc to docker to kubernetes," *IEEE Cloud Computing*, Vol.1, No.3, pp.81-84, 2014.
[10] T. Cerny, M. J. Donahoo, and M. Tmka, "Contextual understanding of microservice architecture: Current and future directions," *ACIM SIGAPP Applied Computing Review*, Vol.17, No.4, pp.29-45, 2018.
[11] Troubleshooting OSDS. [Internet], https://docs.ceph.com/en/quincy/rados/troubleshooting/troubleshooting-osd
[12] S. A., Weil, S. A. Brandt, E. L. Miller, D. D. Long, and C. Maltzahn, "Ceph: A scalable, high-performance distributed file system," *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, pp.307-320, 2006.
[13] C., Maltzahn, E., Molina-Estolano, A., Khurana, A. J., Nelson, S. A., Brandt, and S., Weil, "Ceph as a scalable alternative to the hadoop distributed file system," *The USENIX Magazine*, Vol.35, pp.38-49, 2010.
[14] S. A. Weil, A. W. Leung, S. A. Brandt, and C. Maltzahn, "Rados: A scalable, reliable storage service for petabyte-scale storage clusters," *Proceedings of the 2nd International Workshop on Petascale Data Storage: Held in Conjunctin with Supercomputing'07*, pp.35-44, 2007.
[15] Ceph Architecture [Internet], https://docs.ceph.com/en/latest/architecture
[16] W. K. Lin, D. M. Chiu, and Y. B. Lee, "Erasuere code replication revisited," *Proceedings of Fourth International Conference on Peer-to-Peer Computing*, pp.90-97, 2004.
[17] X. Zhang, S. Gaddam, and A. T. Chronopoulos, "Ceph distributed file system benchmarks on an openstack cloud," *IEEE International Conference on Cloud Computing in Emerging Markets(CCEM)*, pp.113-120, 2015.
[18] Chao-Tung Yang, W. H. Lien, Y. C. Shen, and F. Y. Leu, "Implementation of a Software-Defined Storage Service with Heterogeneous Storage Technologies." *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*, pp.102-107, 2015, doi:10.1109/WAINA.2015.50.
[19] T. Zhang, S. Toor, and A. Hellander, "Efficient hierarchical storage management framework empowered by reinforcement learning," *arXiv:2201.11668*, 2022.
[20] P. -J. Maenhaut, H. Moens, B. Volckaert, V. Ongenae, and F. De Turck, "Design of a hierarchical software-defined storage system for data-intensive multi-tenant cloud applications," *2015 11th International Conference on Network and Service Management (CNSM)*, pp.22-28, 2015, doi: 10.1109/CNSM.2015.7367334.
[21] J. Mockus, V. Tiesis, and A. Zilinskas, "The application of bayesian methods for seeking the extrenum," *Towards Global Optimization*, Vol.2, pp.117-129, 1978.
[22] J. M. Bernardo and A. F. Smith, "Bayesian theroy," John Wiley & Sons, 2009.
[23] E. Schulz, M. Speekenbrink, and A. Krause, "A tutorial on Gaussian process regression: Modeling, exploring, and exploiting functions," *Journal of Mathmatical Psychology*, Vol.85, pp.1-16, 2018.

이 정 일
https://orcid.org/0000-0002-0719-312X
e-mail : jlee@kepco.co.kr
2004년 전북대학교 컴퓨터공학과(학사)
2022년 충남대학교 컴퓨터공학과(석사)
2004년 ~ 2010년 전력연구원 일반연구원
2011년 ~ 현 재 전력연구원 선임연구원
관심분야 : 클라우드 컴퓨팅, 전력수요예측, 데이터 사이언스

최 윤 아
https://orcid.org/0000-0002-9787-3662
e-mail : yoona.choi@kepco.co.kr
2019년 고려대학교 컴퓨터정보학과(학사)
2021년 고려대학교 컴퓨터융합소프트웨어학과(석사)
2011년 ~ 현 재 전력연구원 일반연구원
관심분야 : 전력수요예측, 데이터 사이언스

박 주 은

https://orcid.org/0000-0001-7450-9510
e-mail : jueun_park@kepco.co.kr
2020년 제주대학교 경영정보학과(학사)
2021년 ~ 현 재 전력연구원 일반연구원
관심분야 : 클라우드 컴퓨팅, 데이터 사이언스

장 민 영

https://orcid.org/0000-0002-3605-3468
e-mail : dearjmy@kepco.co.kr
2008년 숭실대학교 정보통신전자공학과 (학사)
2021년 ~ 현 재 충남대학교 컴퓨터공학과 석사과정
2021년 ~ 현 재 전력연구원 선임연구원
관심분야 : 전력계통 운영효율 관련 데이터 사이언스