

Boot storm Reduction through Artificial Intelligence Driven System in Virtual Desktop Infrastructure

Heejin Lee*, Taeyoung Kim**

*Professor, Dept. of Computer Software Engineering, Dongyang Mirae University, Seoul, Korea

**Software engineer, Clcom, Gyeonggi, Korea

[Abstract]

In this paper, we propose BRAIDS, a boot storm mitigation plan consisting of an AI-based VDI usage prediction system and a virtual machine boot scheduler system, to alleviate boot storms and improve service stability. Virtual Desktop Infrastructure (VDI) is an important technology for improving an organization's work productivity and increasing IT infrastructure efficiency. Boot storms that occur when multiple virtual desktops boot simultaneously cause poor performance and increased latency. Using the xgboost algorithm, existing VDI usage data is used to predict future VDI usage. In addition, it receives the predicted usage as input, defines a boot storm considering the hardware specifications of the VDI server and virtual machine, and provides a schedule to sequentially boot virtual machines to alleviate boot storms. Through the case study, the VDI usage prediction model showed high prediction accuracy and performance improvement, and it was confirmed that the boot storm phenomenon in the virtual desktop environment can be alleviated and IT infrastructure can be utilized efficiently through the virtual machine boot scheduler.

▶ **Key words:** VDI, boot storm, virtual machine, AI, efficiency

[요 약]

본 연구에서는 부트 스톰을 완화하고 서비스 안정성 향상을 위하여 AI 기반 VDI 사용 예측 시스템, 가상머신 부팅 스케줄러 시스템으로 구성된 부트 스톰 완화 방안인 BRAIDS를 제안한다. 가상 데스크톱 인프라(Virtual Desktop Infrastructure, VDI)는 조직의 업무 생산성 향상과 IT 인프라 효율성 증대를 위한 중요한 기술이다. 다수의 가상 데스크톱이 동시 부팅될 때 발생하는 부트 스톰은 성능저하와 대기 시간 증가를 유발한다. xgboost 알고리즘을 사용하여, 기존 VDI 사용 데이터를 활용하여 향후 VDI 사용량을 예측한다. 또한 예측된 사용량을 입력으로 받아 VDI 서버와 가상머신의 하드웨어 사양을 고려하여 부트 스톰을 정의하고, 부트 스톰을 완화하기 위하여 순차적으로 가상머신을 부팅할 수 있는 스케줄을 제공한다. 사례연구를 통하여 VDI 사용 예측 모델은 높은 예측 정확도와 성능 향상을 보였으며, 가상머신 부팅 스케줄러를 통하여 가상 데스크톱 환경에서의 부트 스톰 현상을 완화하고 효율적으로 IT 인프라를 활용할 수 있음을 확인하였다.

▶ **주제어:** 가상 데스크톱 인프라, 부트 스톰, 가상머신, 인공지능, 효율화

- First Author: Heejin Lee, Corresponding Author: Heejin Lee
- *Heejin Lee (heejinlee@dongyang.ac.kr), Dept. of Computer Software Engineering, Dongyang Mirae University
- **Taeyoung Kim (tykim@clcom.kr), Clcom
- Received: 2024. 05. 14, Revised: 2024. 06. 24, Accepted: 2024. 07. 08.

I. Introduction

가상 데스크톱 인프라 (Virtual Desktop Infrastructure, VDI)는 중앙 데이터센터의 서버 자원과 가상화 기술을 활용하여 데스크톱 환경을 사용자에게 네트워크상에서 제공하는 기술이다[1]. 현대 조직에서 IT 인프라를 효율적으로 활용·관리하기 위한 기술로, 기존의 물리적인 데스크톱 환경이 아닌 중앙 서버에서 가상머신 형태로 사용자에게 데스크톱 환경이 제공된다. 중앙 집중화 방식으로 IT 인프라를 관리하고, 사용자에게는 높은 유연성과 접근성을 제공할 수 있다. VDI는 원격 작업, 업무 환경 구축, 보안 강화 등 다양한 목적으로 활용되며, 특히 COVID-19 대유행으로 원격 작업과 온라인 교육이 급증함에 따라, 기업에서 VDI 환경 도입이 점점 늘어나고 있다. [1.2] 사용자가 가상머신을 이용하여 권한에 맞는 애플리케이션과 데이터에 접근할 수 있으며, 이러한 특징은 효율적인 리소스 관리와 업무 생산성 향상을 가능케 한다.

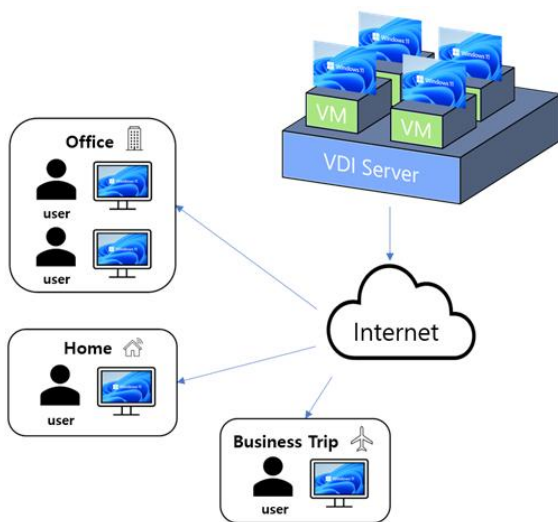


Fig. 1. Virtual Desktop Infrastructure

다수의 VDI 사용자가 동시에 가상머신을 부팅하는 상황에서 발생하는 부트 스톰[3]은 성능저하와 대기 시간 증가를 유발한다. 기존의 부트 스톰 방지 방법은 크게 두 가지이다. 첫 번째는 고성능 하드웨어를 추가하는 방안으로 비용이 많이 드는 단점이 있다. 두 번째는 정적 부팅 스케줄링을 통해 미리 가상머신을 부팅하는 방안으로 가상머신 사용이 일정하지 않은 상황에서 부트 스톰 해결이 어려운 단점이 있다. 동적 스케줄링을 하기 위해서는 VDI를 사용하는 조직의 구조와 직원들의 사용 패턴을 분석해야 하며, VDI를 사용하는 조직마다 직원의 구성이나, 휴일에 따라 VDI의 사용량을 정확히 예측하기 어려움이 있다.

본 논문에서는 VDI의 부트 스톰을 완화하기 위한

BRAIDS (Boot storm Reduction through Artificial Intelligence Driven System)를 제안한다. BRAIDS는 두 개의 시스템으로 구성되어 있다. 첫 번째로 VDI 사용 데이터 기반으로 기계학습을 통해 VDI 사용량을 예측하는 시스템이다. 두 번째로 부트 스톰이 최소화되도록 예측 사용량에 따라, 가상머신을 미리 부팅 해 놓는 동적 가상머신 스케줄링 방안을 제안하는 시스템이다. 이를 통하여 다양한 VDI 서버의 하드웨어 환경에 맞추어 다양한 VDI 사용 데이터 기반의 스케줄링을 할 수 있다. BRAIDS를 통하여 부트 스톰으로 인한 성능저하를 최소화하고, 다수의 가상머신을 안정적으로 제공할 수 있다.

본 논문의 구성은 다음과 같다. II 장은 부트 스톰 개선을 위한 기존 연구를 기술한다. III 장은 부트 스톰 감소를 위한 VDI 활용시스템인 BRAIDS를 소개한다. IV 장에서는 VDI 사용량 예측을 위한 AI 기계학습 모델을 제안한다. V 장에서는 VDI 사용 예측과 하드웨어 사양을 바탕으로 가상머신 동적 부팅 스케줄링 시스템에 관하여 기술한다. VI 장에서는 결론 및 향후 연구로 논문을 마무리한다.

II. Related works

부트 스톰은 VDI 서버에 대규모 사용자가 접속하여 다수의 가상머신을 동시에 기동시킬 때 발생하는 성능저하와 대기 시간이 길어지는 현상이다. [3] 부트 스톰은 주로 디스크 대역폭에 영향을 받아 발생한다. [4] 대규모 가상머신이 동시에 기동 되면, 이에 따른 디스크 입출력(Disk Input/Output) 작업이 증가하면서 중앙 서버에 부하가 집중되고, 이에 따라 부팅 시간이 급격히 증가하여 VDI 사용성을 저해하고, 전체 시스템의 효율성이 낮아진다.

부트 스톰 현상을 해결하기 위해 다양한 기술적 접근 방법이 제안되었다. Soundararajan et al.은 128명의 동시 부팅 시 스토리지 대역폭이 최소 10Gbps 이상이 권장했다[4,5]. SanDisk et al.은 부트 스톰과 안정성을 위해 IOPS 높은 SSD를 권장했다. [6, 7] 또한, Li et al.은 디스크 입출력 성능 및 데이터 최적화를 통해 트래픽 부하를 감소하여 부팅 대기 시간을 줄였다. [8] Austin et al.은 중복 데이터를 효율적으로 관리하고[9], Kaveh et al.와 Shi et al.은 캐싱 전략을 활용하여 가상머신의 기본 이미지를 서버 캐싱하거나 부팅 커널을 공유 메모리에 캐싱하여 데이터를 효율적으로 활용하는 방법을 제시했다. [10, 11] 부트 스톰을 완화하기 위하여 스토리지를 HDD에서 SSD로 변경하고, 스토리지 대역폭을 높이는 것은 비용이

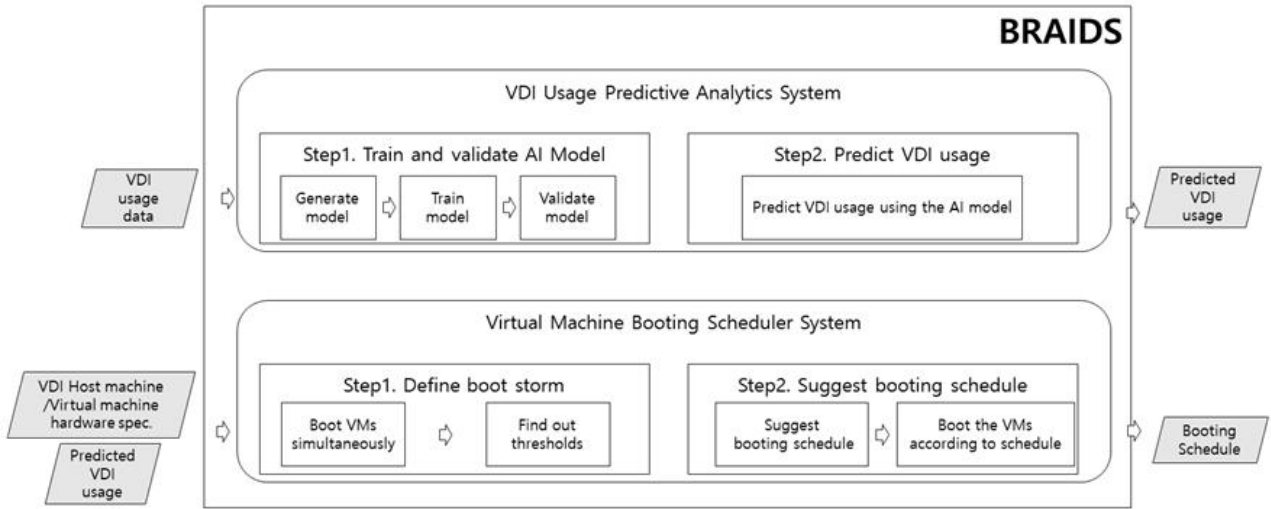


Fig. 2. BRAIDS (Boot storm Reduction through Artificial Intelligence Driven System)

많이 발생한다[12,13]. 가상머신은 부팅시 가장 많은 리소스를 사용하며, 애플리케이션의 평균 리소스 요구량은 부팅시 활용하는 리소스에 비하여 훨씬 작다. 본 논문에서는 부트 스톰 개선을 위해 비용이 많이 드는 하드웨어 성능 개선이 아닌, 비용이 추가로 들지 않는 가상머신 동적 부팅 방안을 제시한다.

클라우드 컴퓨팅에서 머신 러닝을 활용하여 리소스 관리를 하는 연구[14]들이 시도되고 있고, 이중 가상머신의 효율성을 위한 연구도 진행 중이다. 사용자가 원하는 가상머신 종류를 다르게 하여 최적의 효율을 높이는 연구 [15], 가상머신의 워크로드를 덤핑 기법을 활용하여 예측하여 가상머신을 배치하는 연구 [16,17], 예측 모델을 통한 가상머신의 사용 시점에 대한 예측을 보이는 연구[18]등이 존재한다. 본 연구에서는 이들 연구에서 사용한 예측 기법이 아닌 새로운 xgboost 알고리즘으로 예측하고, 현 하드웨어 환경에 따른 부트 스톰을 완화할수 있는 방안을 제안한다.

III. BRAIDS (Boot storm Reduction through Artificial Intelligence Driven System)

BRAIDS(Boot storm Reduction through Artificial Intelligence Driven System)는 부트 스톰을 완화하기 위해 AI를 활용하여 VDI 사용량을 예측하고, 가상머신 부팅 스케줄링을 제안하는 시스템이다. 그림 2와 같이 BRAIDS는 크게 VDI Usage Predictive Analytics System(VDI 사용 예측 시스템)과 Virtual Machine Booting Scheduler System(가상머신 부팅 스케줄러 시스템)으로 구성된다.

1. VDI Usage Predictive Analytics System

VDI 사용 예측 시스템은 기존 VDI 사용량을 기계학습으로 학습하고, 향후의 사용을 예측하는 시스템이다. 첫 번째 단계로는 VDI 사용량을 예측하는 모델을 생성, 학습, 검증한다. 이때 모델은 다양한 알고리즘을 활용할 수 있고, 본 연구에서는 xgboost 알고리즘을 활용하여 모델을 생성, 학습, 검증하였다. 모델 학습시 기존의 VDI 사용량을 입력으로 받으며, 이 VDI 사용량은 회사마다 유연근무제로 인한 다양한 시간대의 출근, 매년 변경되는 음력 공휴일이나 대체 공휴일등의 다양한 요소들에 영향을 받는다. 두번째 단계로는 검증이 완료된 모델을 사용하여 향후의 VDI 사용량을 예측한다. 예측된 VDI 사용량은 Virtual Machine Booting Scheduler System(가상머신 부팅 스케줄러 시스템)에 입력으로 제공한다.

2. Virtual Machine Booting Scheduler System

가상머신 부팅 스케줄러 시스템은 부트 스톰 상황을 완화하도록 동시기동 가상머신 사용량, VDI 서버의 하드웨어 사양, 네트워크 사양, 가상머신 이미지 크기 요소들을 고려하여 가상머신 부팅 일정을 제안한다.

첫 번째 단계는 부트 스톰을 시뮬레이션 한다. VDI 서버 환경, 가상머신의 자원 요구량에 따라서 기동 시간이 달라 지므로 여러 경우를 실험해 보고, 가상머신의 기동시간이 명확하게 느껴지는 시점을 찾아 부트스톰으로 정의한다. 부트 스톰은 자연 대기 시간의 명확한 기준이 존재하지 않는다. 즉 회사의 규모, VDI 서버의 하드웨어 비용 효율성, 부팅 대기시간의 적정 기준에 따라서 해당 상황에서의 부트 스톰을 정의한다. 부트 스톰 정의에 따라, 다음 단계에서 부트 스톰이 일어나지 않도록 스케줄을 정의한다.

두 번째 단계는 부팅 스케줄을 정하고, 그에 맞추어 가상머신을 기동한다. 첫 번째 단계에서 정의한 부트 스톱이 일어나지 않도록, 가상머신을 순차적으로 기동시킨다. 정해진 일정에 따라 사용자가 실제 사용하기 이전에 가상머신을 미리 기동시켜서 사용에 불편이 없도록 한다.

IV. VDI Usage Predictive Analytics System

VDI 사용 예측 시스템은 VDI의 동시간대 부팅 수요를 기계학습으로 지금까지의 사용 현황을 기반으로 예측하는 시스템이다. 예측 시스템의 결과로 나온 수요 예측은 부트 스톱을 예측하고 가상머신 부팅 스케줄링을 정할 때 입력으로 사용된다.

예측을 위하여 앙상블 부스팅 기법의 한 종류인 XGBoost(Extreme Gradient Boosting) [19] 회귀분석 알고리즘을 활용한다. XGBoost는 기계학습 분야에서 널리 사용되는 강력하고 효과적인 알고리즘 중 하나이다. 여러 결정 트리를 조합하여 강력한 앙상블 모델을 형성하고, 각 트리는 이전 트리의 오차를 최소화하는 방식으로 구성되어 전체 모델의 예측 성능을 향상한다. 데이터의 병렬 처리를 지원하여 대규모 데이터 세트에서도 빠르게 학습이 가능하고, 다양한 하이퍼파라미터를 조절하여 모델의 성능을 향상한다[19]. 모델 작성은 python xgboost [20] 라이브러리와 scikit-learn [21] 라이브러리를 활용하였다.

모델 학습은 가상머신의 대규모 동시기동이 필요한 시각과 사용량이 포함된 데이터를 직접 생성하여 활용하였다. 직원이 1,000명인 가상의 회사를 설정하고, 출근 시간은 7시, 8시, 9시, 10시로 구분하였으며, 시간대별 출근 인원은 10%, 40%, 30%, 10%로 설정하고, 오차범위 50명으로 랜덤으로 설정하였다. 휴일에는 VDI를 활용한 업무가 없으며, 회사에 출근 시 VDI에 접속하여 업무를 한다고 가정하였다. 데이터는 연도(Year), 월(Month), 일(Day), 요일(Wday), 시간(Hour), 휴일(Holiday), 음력 월(Lmonth), 음력 일(Lday), 출근 인원(attendance) 총 9개 컬럼으로 이루어져 있으며, 2013년 1월 1일부터 2023년 12월 31일 까지 11년의 데이터를 생성하였다. 총 16068개의 전체 데이터가 있으며, 학습은 2013년부터 2022년까지의 데이터(총 14608개)로 학습하고, 2023년 데이터(총 1460개)로 검증하였다.

Target은 동시 부팅 가상머신 수를 나타내는 출근 인원 변수인 attendance로 설정하고, Feature는 나머지 변수

로 설정하고 학습했다. 예측 모델 검증은 2023년 데이터를 이용했으며, R2 score(R Squared Score), MSE(Mean Square Deviation), RMSE(Root Mean Square Deviation)의 값은 표1과 같다. R2 score는 예측하는 종속 변수의 변동 중에서 실제 데이터의 변동을 얼마나 잘 설명하는지를 표현하는 비율이다. 훈련 데이터 및 평가 데이터 모두에서 높은 R2 Score를 가지며, 이는 모델이 실제 데이터와 예측값이 높은 일치도 및 상관관계를 가짐을 나타낸다. 또한 제안한 모델의 MSE와 RMSE를 확인하여 과적합을 피하며 효과적으로 실제 데이터를 예측한 것을 확인했다.

$$R^2 = 1 - \frac{\sum (y_i - \hat{y})^2}{\sum (y_i - \bar{y})^2} \quad (1)$$

, where \hat{y} = predicted values of y ,
 \bar{y} = mean value of y .

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (2)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (3)$$

Table 1. Model validation result

Metric	Value
R2 Score	0.9879393
MSE	275.34835
RMSE	16.593624

최적 하이퍼파라미터 구성을 찾기 위하여 scikit-learn python library인 GridSearchCV를 이용하여, 다섯 개의 파라미터와 값을 표 2와 같이 도출하였다.

Table 2. Best Hyper-Parameters

parameter	value
n_estimators	700
learning_rate	0.01
max_depth	3
min_child_weight	1
min_split_loss	2

그림 3은 2023년의 데이터의 예측값과 실제값을 가우시안 필터를 적용하여 노이즈를 제거하였고, 그 결과 매우

유사한 성향을 보이는 것을 확인 할 수 있다.

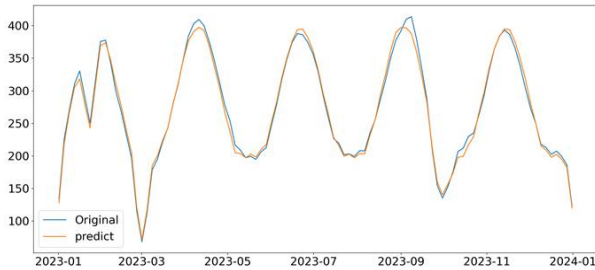


Fig. 3. Actual value vs Prediction value

V. Virtual Machine Booting Scheduler System

가상머신 부팅 스케줄러 시스템은 VDI 환경에 맞는 가상머신 부팅 스케줄을 제안하기 위하여 크게 두 단계로 나뉜다. 첫 번째 단계에서는 각 상황에 맞는 부트 스톰을 정의한다. 부트 스톰이란 가상머신의 대규모 동시 기동으로 성능저하가 일어나는 것으로, 이에 대하여 정량적으로 명확한 정의는 존재하지 않는다. 성능저하를 규정하는 임계치는 회사의 규모, VDI 서버 비용, 업무의 긴급성 등에 따라 정의되며, 상황의 중요도에 따라서 변경되기 때문이다. 두 번째 단계는 앞 단계에서 정의된 부트 스톰이 발생하지 않도록 가상머신을 스케줄링 하여 기동시킨다. VDI 사용 예측 시스템으로부터 받은 예측 사용량과 VDI 서버 하드웨어 사양, 가상머신 하드웨어 사양을 입력받아 부트 스톰이 일어나지 않을 가상머신 부팅 스케줄을 출력으로 제공한다.

1. Define boot storm

호스트머신과 가상 머신의 사양에 따라 디스크 I/O, CPU 사용율이 다르므로, 각 환경에서 여러 vm을 동시에 기동하여, 성능저하가 일어나는 시점을 찾아 부트스톰을 정의한다. 동시 기동해도 성능저하가 급격히 늘어나지 않는 가상머신 최대 개수를 임계치로 정의하고, 성능저하의 임계치도 정의한다.

부트 스톰에 영향을 주는 4가지를 고려하여 부트스톰 확인 매커니즘을 개발하였다.

1) 디스크 I/O

- 부트스톰은 디스크 I/O에 가장 많은 영향을 받는다. 이는 컴퓨터의 처리 과정 중 디스크에서 가장 많은 시간이 걸리기 때문이다.

- HDD가 하나인 환경에서 기동하는 가상머신의 개수와 전체 기동 시간은 비례한다. 이는 가상머신의 개수가 많을 수록 읽어야 하는 데이터 양이 비례하여 늘어나며, 데이터를 읽는 것이 순차적으로 진행되기 때문이다.

- HDD가 여러 개인 환경에서, 각 HDD에서 병렬로 디스크 I/O가 진행되므로 HDD의 개수만큼 동시에 가상머신이 기동된다.

- 하나의 가상머신을 기동시 부팅 완료 시점까지의 평균 초당 디스크 I/O를 얻으며, 이를 활용하여 스케줄링을 한다.

2) 네트워크의 대역폭

- 원격 서버로부터 받는 네트워크 대역폭이 전체 디스크 I/O보다 작으면 성능저하가 일어나므로, 성능저하가 일어나지 않도록 가상머신의 개수를 조절한다.

3) CPU 사용률

- 디스크 I/O가 과도하게 일어나면 CPU 사용률에 영향을 주어 높아지며, CPU 사용률이 100%에 도달하면 성능저하가 일어난다. 이 경우 가상머신 동시 기동 개수가 부트스톰을 일으키는 임계치이다.

4) 그 외 요인

- 상황에 따라서 성능저하보다 더 중요한 요인이 존재한다. VDI 서버의 사양, 비용, 회사의 규모, 업무의 긴급성에 따라서 성능저하를 감내하는 경우를 고려한다.

부트 스톰의 임계치를 구하는 매커니즘은 그림 4와 같다. 동시 기동하는 가상머신을 순차적으로 증가시키면서 임계치를 찾는다. CPU 사용률이 100% 넘는 경우, HDD의 개수가 가상머신의 개수보다 작은 경우, 네트워크 대역폭이 전체 디스크 I/O 보다 적은 경우, 그 상황에서 기동한 가상머신의 수가 임계치의 후보가 된다. 다만 회사 사정 등에 의한 다른 요인 또한 고려된다.

부트 스톰을 정의하기 위하여 가상 머신을 동시 기동 실험을 여러번 실행하였다. 실험에서 사용한 호스트 머신은 1개의 HDD를 가진 고사양 개인용 컴퓨터로, 호스트 머신과 가상머신의 하드웨어 사양은 표 3과 같다. 본 실험은 HDD 환경에서 가상머신을 동시에 여러개를 실행시켰으며, 제약 사항은 네트워크를 통하여 가상머신을 실행하지 않은 점과, HDD가 한 개인 점이다. 실험은 총 10회 진행하였고, 가상머신을 1개부터 12개 까지 동시에 부팅시키고, 모든 가상머신 부팅이 완료된 시점과 CPU 사용률을 측정하였다. 부팅 완료 시간은 가상머신이 부팅되어 OS의 사용자 로그인 화면으로 전환될 때까지의 시간을 측정했다.

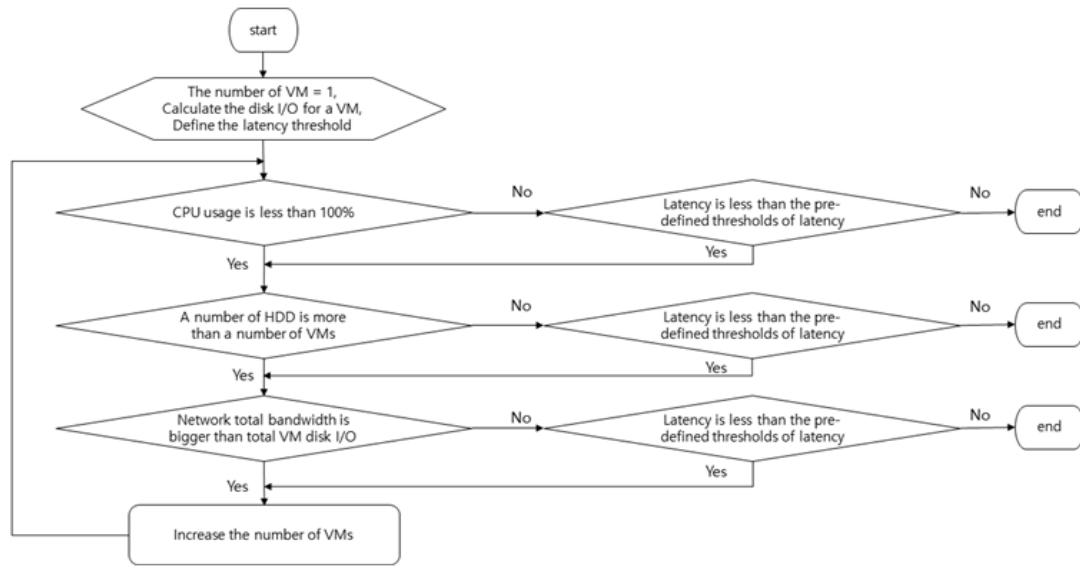


Fig. 4. Flow chart for finding out the thresholds of boot storm

Table 3. Host machine/Virtual machine specification

Host machine specification	
OS	Windows 11
CPU	AMD Ryzen 5 3600
	3.6 GHz, 6 Core, 12 Thread
RAM	DDR4-3600 32 GB
Storage 1	SATA3 HDD, 4TB
	5400 RPM, 256MB Buffer
Virtual machine specification	
OS	Ubuntu 22.04
CPU	1 Core
RAM	2 GB
Storage	AHCI(SATA) 25GB
VM image	630M

Table 4. Total time that all VMs are ready

VMs	1	2	4	6	8	10	12
Total time that all VMs are ready(s)	58	109	201	298	445	563	798
Average DISK I/O	10.9	11.6	12.5	12.7	11.3	11.2	9.5

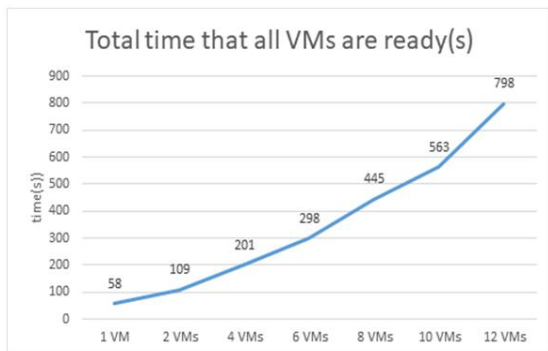


Fig. 5. Total time that all VMs are ready

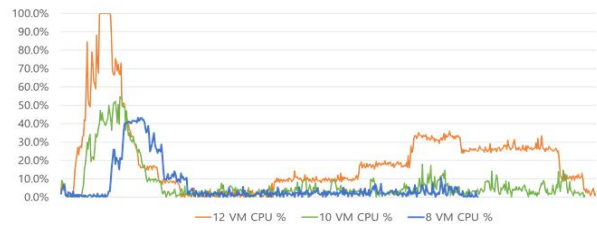


Fig. 6. CPU usage

실험 결과, 부트 스톰은 가상머신이 12개인 경우 나타나며, 부트 스톰이 일어나지 않게 하기 위해서는 동시 기동 가상머신을 12대 보다 적은 10대로 해야 한다. 표4와 그림 5에서 HDD가 1개인 상황에서는 순차적으로 데이터를 읽기 때문에, 가상머신의 개수와 부팅 완료 전체 시간이 비례함을 알 수 있다. 그림 6은 가상머신이 8대, 10대, 12대인 경우 CPU 사용률을 나타낸다. 12대의 가상머신을 부팅 시 CPU 사용률이 100%에 도달하였고, 이때 부팅 대기 시간이 증가한 것을 확인 할 수 있다. 디스크 I/O가 급증하면 이에 따라 CPU 사용률이 같이 증가하고, CPU 사용률이 일정 이상 높아지면, 컴퓨터 성능이 저하된다. 즉 CPU 사용률을 유심히 살펴 부트 스톰의 시작점을 찾을 수 있다.

가상머신 12개를 동시에 동작시킬 경우, CPU 사용률이 100%가 되고, 100%로 일정시간 유지된 후에 하강하였다. 이때 성능저하가 일어나, 12개 부팅시 전체 가상머신 부팅 완료 시간도 798초로 10개 이하의 가상머신 부팅 상황 보다 평균적으로 더 긴 시간이 소요되었다. 평균 디스크 I/O는 가상머신이 1대에서 10대까지는 10.9MB/s에서 12.7MB/s 사이에 있었는데, 가상머신이 12개일 때, 9.5MB/s로 떨어졌다. 이는 디스크 I/O가 과부하가 걸려 CPU 사용률이 임계치 이상 높아졌고, 그로 인하여 성능저

하가 생긴 것이다. 또한 가상머신 한 개가 생성될 때 하나의 HDD에서 걸리는 디스크 I/O는 10.9mb/s 이다.

2. Suggest booting schedule

가상머신 부팅 스케줄러 시스템은 가상머신이 부팅이 완료되어야 하는 시각을 기준으로 부트 스톰이 발생하지 않는 동시 기동 가능 최대 가상머신 개수만큼 순차적으로 기동시킨다. 앞 단계에서 정의한 부트스톰이 일어나지 않을 최대 부팅 가상머신 개수를 사용하며, 이를 순차적으로 기동하는 스케줄을 제공한다.

부트 스톰 예방 시스템 BRAIDS를 활용하여 그림 7와 같은 가상의 VDI 환경에서 특정한 개수의 가상머신을 기동할 때 부트 스톰이 일어나지 않기 위한 스케줄링을 제안한다. 해당 환경은 스토리지는 Network Attached Server(NAS)와 연결되어 있으며, HDD가 10개로 구성되어 있다. HDD가 여러개인 경우, HDD를 병렬로 효율적으로 가상환경을 기동시킬 수 있다. 스토리지 서버의 대역폭, 각 가상머신이 부팅 시 소비되는 대역폭, 가상머신이 부팅에 필요한 부팅 데이터 크기는 표 5과 같으며, 적용 환경에 따라 변경 가능하다.

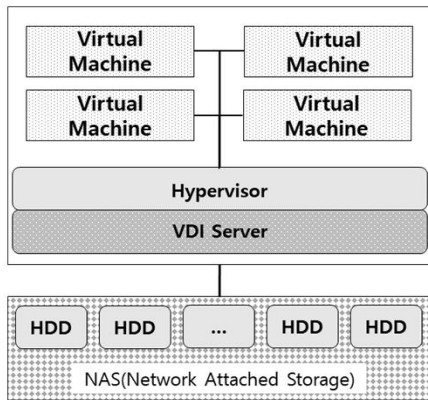


Fig. 7. VDI server

Table 5. Host machine, NAS and virtual machine hardware specification

Host machine specification	
OS	Ubuntu 22.04
Storage	SATA3 HDD 4TB*10 each
	5400 RPM, 256MB Buffer
NAS specification	
Total bandwidth	1Gbps
Virtual machine specification	
OS	Ubuntu 22.04
CPU	1 Core
RAM	2 GB
Storage	AHCI(SATA) 25GB
VM image size	630M
Booting time	58s
Disk I/O for a VM	10.9mb/s

Table 6. Predicted VDI usage

	Time	Number of required VMs
1	8:00 a.m.	70
2	9:00 a.m.	100
3	10:00 a.m.	60
4	11:00 a.m.	30
5

BRAIDS의 VDI 사용 예측 시스템을 통하여 예측된 표 6와 같은 가상머신 사용량 예측을 입력받는다. 오전 8시에 가상머신 70대가 가동되어야 하는 상황에 대해 스케줄링한다.

그림 4의 부트 스톰의 임계치를 찾는 매커니즘을 활용하여 최적의 동시 기동 가상머신 개수를 구하였다. CPU 사용율이 100%가 넘는 경우, 네트워크 대역폭보다 초당 디스크 I/O가 큰 경우, HDD 개수보다 가상머신의 개수가 더 많은 경우를 고려하였고, 결과값은 10개이다. 현 실험 환경에서 HDD의 개수가 10개이므로 동시에 병렬로 10개를 실행할 수 있다. 또한 네트워크 대역폭이 1Gbps 즉 125mb/s이므로 11.5개보다 적은 개수의 가상머신을 실행하는 경우, 1대를 실행했을 때와 10대를 실행했을 때 동일한 시간이 걸린다. 10개보다 더 많은 가상머신을 실행할 경우, HDD의 특성상 디스크량에 비례하여 전체 시간이 늘어난다.

만약 70대를 동시 실행한다면, HDD 환경에서는 읽을 디스크의 양(가상머신의 개수)에 따라 지연시간이 발생한다. 또한 CPU 사용율이 100%를 넘게 되고, 대역폭보다 디스크 I/O가 약 7배 더 높아져 발생하는 병목 현상이 일어난다. 70개의 가상머신을 10개의 HDD가 나누어서 병렬로 7개씩 기동한다고 가정하면, 표4에 의하면 6대 기동시 걸리는 시간인 298초와 8대 기동시 걸리는 445초 사이의 약 370초 정도 소요된다. CPU 사용률과 대역폭 요소를 제외하고도 약 6배 이상(370/58)의 시간이 걸리고, CPU 사용률과 네트워크 대역폭의 병목현상으로 인해 훨씬 더 오래 걸릴 것으로 예상된다.

표 7은 스케줄링 결과이며, 가상머신은 미리 계획된 시간에 맞춰 순차적으로 부팅을 한다. 7시 53분에 10대를 동시 기동을 시작하고, 10대 기동이 완료된 후 7시 54분에 다시 10대를 기동한다. 이처럼 7차례에 걸쳐 차례로 10개씩 부팅하여, 8시 정각에는 70대가 모두 부팅 완료 되어 사용할 준비가 된다. 이처럼 가상머신 부팅 스케줄러 시스템을 통하여 사용자들은 부트 스톰을 경험하지 않고, 미리 가상머신이 부팅되어 있으며, VDI 시스템도 효율적으로 미리 기동 되어 불필요한 전력 소모가 없다.

Table 7. A result of virtual machine scheduling

	Booting start time	Booting ready time	Num of VMs booted simul.	Num of required VMs	Num of VMs booted	Num. of VMs not booted
1	7:53	7:53:58	10	70	10	60
2	7:54	7:54:58	10	70	20	50
...
6	7:58	7:58:58	10	70	60	10
7	7:59	7:59:58	10	70	70	0
8	8:50	8:50:58	10	100	10	90

VI. Conclusions

원격 작업 및 온라인 교육 활성화로 인해 VDI의 중요성이 점점 커지지만, 대규모 가상머신의 동시 부팅 시 발생하는 부트 스톰으로 인한 VDI 성능저하 문제는 여전히 남아있다. 본 논문에서는 대규모 가상머신 동시 부팅 시 발생하는 성능저하와 대기 시간 증가를 최소화하고, 안정적이고 효율적인 서비스를 제공하고, VDI 운영 하드웨어 비용을 절감하고자 BRAIDS를 제안하였다.

VDI 사용 예측 시스템은 기계학습을 활용한 VDI 사용 예측 모델을 개발하여 사용량을 예측하였다. XGBoost 회귀 알고리즘을 사용하여 가상머신 동시 기동 패턴을 학습하여 가상머신 사용량을 높은 정확도로 예측하였다. 가상머신 부팅 스케줄러를 도입하여 예측된 가상머신 사용량, 서버의 하드웨어 사양에 따라 부트 스톰 현상을 완화하기 위해 동적 스케줄러를 제안하였다. 예측된 사용량에 따라 가상머신을 동적으로 부팅하고, 스토리지 대역폭을 효과적으로 활용하여 사용자가 겪는 부팅 대기 시간을 최소화하였다. 또한 BRAIDS를 적용한 사례연구를 통하여 시스템의 적합성을 확인하였다.

향후 연구로는 VDI 사용 예측 시스템을 더 개선하여, 사용자별 가상머신 기동 시간, 직군별/그룹별로 다른 가상머신 유형에 따라서 여러 상황을 예측할 수 있게 하겠다. 차기 모델은 학습 data에 개인 특성이 반영된 속성을 더 추가하여, 개인 맞춤형 가상머신 사용량을 예측하고, 그에 따른 부팅 스케줄을 제공하여 본 연구에서 제안한 방법을 보완 및 확장할 계획이다.

ACKNOWLEDGEMENT

Following are results of a study on the "Leaders in INdustry- university Cooperation 3.0"

Project, supported by the Ministry of Education and National Research Foundation of Korea.

REFERENCES

- [1] Soni, Vaishali, Deepika Kukreja, and Deepak Kumar Sharma. "Security vs. flexibility: Striking a balance in the pandemic era." Proceedings of the 2020 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS). IEEE, pp.1-5, Dec. 2020. DOI: <https://doi.org/10.1109/ANTS50601.2020.9342779>
- [2] Ahmad, Shah Nawaz, Shabana Mehruz, and Javed Beg. "Securely work from home with CASB policies under COVID-19 pandemic: a short review." Proceedings of the 9th International conference system modeling and advancement in research trends (SMART). IEEE, pp. 109-114, Dec. 2020, DOI: <https://doi.org/10.1109/SMA-RT50582.2020.9337121>
- [3] Samant, Durgesh, and Umesh Bellur. "Handling boot storms in virtualized data centers—a survey." ACM Computing Surveys (CSUR), Vol. 49, No. 1, pp. 1-36, June. 2016, DOI: <https://doi.org/10.1145/2932709>
- [4] Soundararajan, Vijayaraghavan, and Jennifer M. Anderson. "The impact of management operations on the virtualized datacenter." ACM SIGARCH Computer Architecture News, Vol. 38, No. 3, pp. 326-337, June. 2010 DOI: <https://doi.org/10.1145/1815961.1816003>
- [5] Soundararajan, Vijayaraghavan, et al. "Benchmarking a virtualization platform." Proceedings of the 2014 IEEE International Symposium on Workload Characterization (IISWC). pp. 99-109. IEEE, Oct. 2014. DOI: <https://doi.org/10.1109/IISWC.2014.6983049>
- [6] SanDisk, SanDisk® SSD Boot Storm Testing for Virtual Desktop Infrastructure (VDI), https://www.sandisk.com.br/content/dam/sandisk-main/en_us/assets/resources/enterprise/white-papers/sandisk-ssd-boot-storm-testing-for-virtual-desktop-infrastructure.pdf
- [7] SanDisk, vmware. High Performance VDI using SanDisk® SSDs, The VMware Horizon View™ and Virtual SAN - A Deployment and Technical Considerations Guide, https://www.sandisk.com/content/dam/sandisk-main/en_us/assets/resources/enterprise/white-papers/high-performance-vdi-using-sandisk-ssds-vmware-horizon-view-and-virtual-san.pdf
- [8] Li, Dingding, et al. "Triple-L: improving CPS disk I/O performance in a virtualized NAS environment." IEEE Systems Journal, Vol. 11, No. 1, pp. 152-162, March. 2017. DOI: <https://doi.org/10.1109/JSYST.2015.2456038>
- [9] Clements, Austin T., et al. "Decentralized Deduplication in SAN Cluster File Systems." Proceedings of the USENIX annual technical conference. Vol. 9. pp.101-114, June, 2009.

- [10] Razavi, Kaveh, and Thilo Kielmann. "Scalable virtual machine deployment using VM image caches." Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis. pp.1-12, Nov. 2013. DOI: <https://doi.org/10.1145/2503210.2503274>
- [11] Shi, Xiang, et al. "VMBKS: a shared memory cache system based on booting kernel in cloud." The Journal of Supercomputing, Vol. 75, pp. 4-19, Jan. 2019 DOI: <https://doi.org/10.1007/s11227-015-1554-1>
- [12] Kim, Il-Han, and Sun-Dong Kwon. "A Study on the Performance of Cloud-based VDI Adoption: Comparing between IS administrators and business users." Management & Information Systems Review, Vol. 37, No. 2, pp. 149-167, June. 2018, DOI: <https://doi.org/10.29214/damis.2018.37.2.008>
- [13] Nakhai, Pedram Hossein, and Nor Badrul Anuar. "Performance evaluation of virtual desktop operating systems in virtual desktop infrastructure." Proceedings of the 2017 IEEE Conference on Application, Information and Network Security (AINS). pp. 105-110, Nov. 2017, IEEE, DOI: <https://doi.org/10.1109/AINS.2017.8270433>
- [14] Goodarzy, Sepideh, et al. "Resource management in cloud computing using machine learning: A survey." Proceedings of the 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA). pp.811-816, Dec. 2020, DOI: <https://doi.org/10.1109/ICMLA51294.2020.00132>
- [15] Lee Changsung, "A Virtual Machine Allocation and Placement Technique for Efficient VDI Services in the Public Institutions Large Private Clouds", MS.thesis, sogang univ, 2020
- [16] Kim, Moon-Hyun, et al. "Min-max exclusive virtual machine placement in cloud computing for scientific data environment." Journal of Cloud Computing, Vol. 10, No. 2, pp.1-17, Jan.2021. DOI: <https://doi.org/10.1186/s13677-020-00221-7>
- [17] Kang, Yoo, Kim, Jung, Shin, Noh, "Dovetail Usage Prediction Model for Resource-Efficient Virtual Machine Placement in Cloud Computing Environment", Journal of KIISE(Korean Institute of Information Scientists and Engineers), Vol.50, No.12, pp. 1041 - 1047 Dec. 2023, DOI: <https://doi.org/10.5626/JOK.2023.50.12.1041>
- [18] Qiu, Feng, Bin Zhang, and Jun Guo. "A deep learning approach for VM workload prediction in the cloud." Proceedings of the 2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD). pp 319-324, May. 2016, DOI: <https://doi.org/10.1109/SNPD.2016.7515919>
- [19] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 785-794, Aug. 2016 DOI: <https://doi.org/10.1145/2939672.2939785>
- [20] Python XGboost library, <https://xgboost.readthedocs.io/>
- [21] Python Scikit-learn library, <https://scikit-learn.org/>

Authors



Heejin Lee received the B.S., M.S., Ph.D degrees in Computer Science from Korea Advanced Institute of Science and Technology in 2004, 2006, and 2017, respectively. Dr. Lee joined the faculty of the

Department of Computer Software Engineering at Dongyang Mirae University, Seoul, Korea, in 2023. She is currently an assistant professor in the Department of Computer Software Engineering in Dongyang Mirae University. She is interested in cloud computing, artificial intelligence and software engineering.



Taeyoung Kim received the B.S., M.S. degrees in Computer Science from Dongyang Mirae University in 2021 and 2024, respectively. He has been a software engineer in Clcom company since 2021.

He is interested in cloud computing and artificial intelligence.