PROJECT PROPOSAL

SERVER I/O PATTERN PREDICTION BASED ON MACHINE LEARNING

머신러닝 기반 서버 I/O 패턴 예측

STUDENT NAME:	Jinseo Choi
STUDENT NUMBER:	202340213
COURSE NAME:	Data Analysis based on Machine Learning
DEPARTMENT:	Dept. of Computer engineering, College of IT Convergence
SUPERVISOR:	Prof. Ok-Ran Jeong
DATE OF SUBMISSION:	2023. 04. 20

CONTENTS

ABSTRACT

Background

AWS(Amazon web service)의 EC2 와 같은 클라우드 서비스는 운영할 때 하나의 서버를 여러 클라이언트가 사용하게 된다. 클라이언트는 지불한 금액에 따라 서버의 리소스와 접근권한을 할당 받는다. 특히 QoS(Quality of service)의 경우 특정 사용자의 성능(e.g. latency, bandwidth)를 보장받아야 한다. 그러나, 기존 QoS 동작 메커니즘은 Latency critical 클라이언트의 Latency 가 지정된 임계 값을 초과하면 Best-effort 클라이언트의 성능을 제한하여 Latency critical 클라이언트의 성능을 유지한다. 이러한 기존 방식은 Latency critical 클라이언트 성능이 임계 값을 벗어난 이후에 동작한다는 문제점이 존재한다. 이러한 문제점을 해결하기 위해 본 연구에서는 머신러닝 기법을 사용하여 서버 상 클라이언트의 I/O 패턴을 미리 예측하여 효율적으로 Latency critical 클라이언트의 성능을 보장하는 QoS 메커니즘을 제안한다.

Methods

본 연구에서는 클라이언트의 I/O 패턴을 예측하기 위해 머신러닝 모델을 사용한다. 머신러닝 모델 훈련을 위한 데이터셋으로 SNIA 에서 제공하는 MSR Cambridge I/O trace 데이터셋을 사용한다. 이를 통해 I/O 패턴을 학습하여 I/O 요청 크기를 예측하는 모델을 생성한다. 이후, 각 모델의 정확도를 측정하여 성능을 평가한다.

Evaluation

제안기법을 평가를 위해 첫째, 서로 다른 종류의 애플리케이션에 대한 I/O trace 로 학습된 3 개의 머신러닝 모델을 생성한다. 이후, 각 테스트 데이터셋을 사용하여 예측 정확도를 측정한다. 이를 통해 머신러닝 기법의 I/O 패턴 예측 가능성을 평가한다.

Discussion and Conclusion

본 연구는 머신러닝 기반 QoS 매커니즘을 구현하기 이전에 머신러닝 기법을 통한 I/O 패턴 예측 가능성에 대해 살펴본다. 향후, QoS 매커니즘을 구현하여 시뮬레이션을 통해 기존방식과 성능평가를 진행하고자 한다.

INTRODUCTION

오늘날 딥 러닝과 같은 인공지능 기법이 높은 성능을 보이면서 다양 분야에 적용되고 있다[1-4]. 딥 러닝 모델은 심층신경망(Deep neural network)으로 구성되어 있으며 대량의 데이터셋을 통해 학습을 진행한다. 이는, 높은 컴퓨팅 연산을 요구한다. 따라서 최근에는 딥 러닝 모델 훈련을 위한 클라우드 컴퓨팅 서비스가 활발하게 사용되고 있다[2]. 클라우드 컴퓨팅 서비스는 고성능 컴퓨팅 서버의 리소스를 여러 사용자가 사용할 수 있도록 리소스를 분배한다. 따라서 리소스 분배를 위한 다양한 분배정책이 존재한다.

Quality of Service

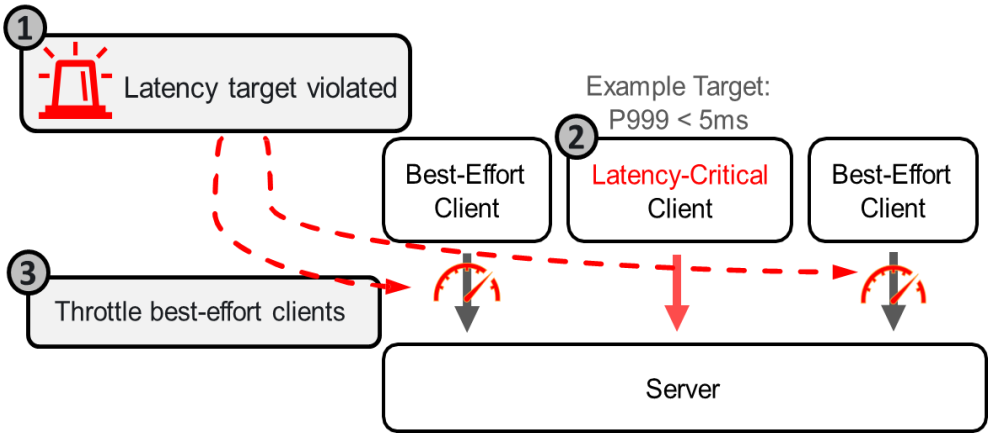

Fig.1 The example of the QoS mechanism

QoS 는 리소스 분배정책 중 하나로 특정사용자에 대해서 일정한 성능을 보장하는 분배정책이다. 크게 Critical 클라이언트와 Best-effort 클라이언트로 구분된다. Critical 클라이언트는 일정한 성능이 보장되는 클라이언트로 일반적으로 Latency 임계 값을 지정하여 임계 값 이하의 성능을 보장하도록 유지된다. Best-effort 클라이언트는 Critical 클라이언트의 성능을 보장하기 위해 성능에 제한을 받는 클라이언트이다. Fig.1 은 QoS 의 동작 예시를 보여준다. 2 개의 Best-effort 클라이언트와 1 개의 Latency critical 클라이언트로 구성된 서버가 존재할 때, Latency critical 클라이언트는 임계 값이 5ms 으로 설정되어 있다. QoS 는 클라이언트의 성능을 모니터링한다. ❶에서 Latency critical 클라이언트의 성능이 저하되는 상황이 발생할 경우 Latency critical 클라이언트의 성능이 임계 값을 넘어서게 된다❷. 이때 QoS 는 Best-effort 클라이언트의 Bandwidth 를 쓰로틀링하여 성능을 제한한다❸. 이로 인해 발생하는 유휴 Bandwidth 리소스를 Latency critical 클라이언트에 할당하여 기존 성능을 유지시킨다.

기존의 QoS 방식은 Latency Critical 클라이언트의 성능이 임계 값을 벗어난 이후에 동작한다는 문제점이 존재한다. 또한 높아진 Latency 가 다시 기준 성능을 회복하는데 시간이 소요된다. 이로 인해 일정 시간동안 성능 임계 값을 벗어난 상태로 유지된다. 이러한 문제점을 해결하기 위해 본

연구에서는 머신러닝 기법을 사용하여 클라이언트의 성능변화를 예측하여 임계 값을 벗어나기 전에 미리 쓰로틀링을 수행하여 성능을 유지하는 기법을 제안한다. 성능 변화를 예측하기 위해 우리는 I/O 패턴을 학습하여 I/O 요청 크기를 예측하는 머신러닝 모델을 설계한다.

Datasets

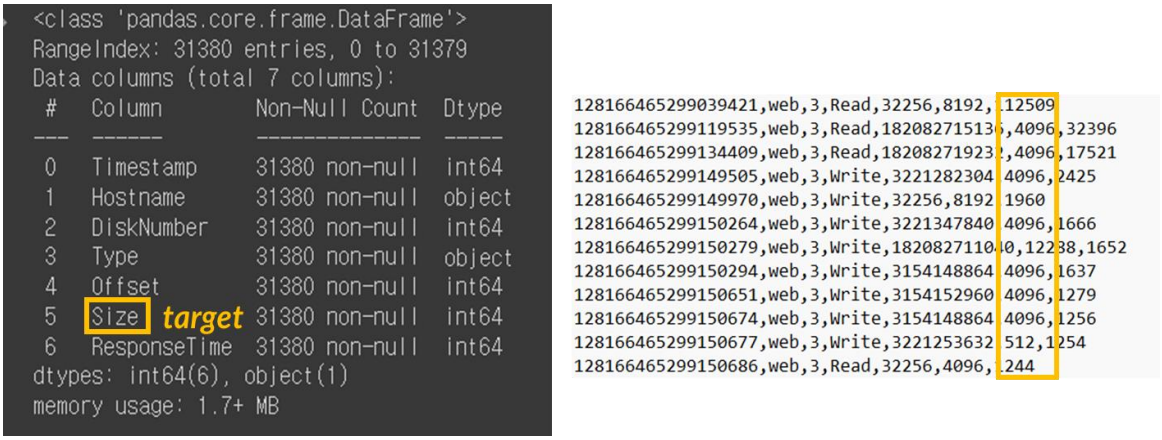

Fig.2 The I/O trace dataset description

머신러닝 모델의 훈련을 위한 데이터셋으로 우리는 SNIA 에서 제공하는 MSR Cambridge I/O trace 데이터셋을 사용한다[5]. 이는 실제 Microsoft 웹 서버의 I/O trace 데이터를 수집하였다. Fig.2 는 I/O trace 데이터셋의 컬럼 정보를 보여준다. 데이터셋은 I/O 를 요청한 시각인 Timestamp, I/O 를 요청한 애플리케이션을 의미하는 Host, 요청을 받은 서버 내 스토리지 번호, I/O 종류, I/O 주소 오프셋, I/O 요청 크기, 응답시간으로 구성되어 있다. 데이터셋은 여러종류의 애플리케이션에 대한 Trace 를 제공한다. 본 연구에서 제안하는 머신러닝 모델은 I/O 요청 크기를 예측하는 모델으로 설계한다.

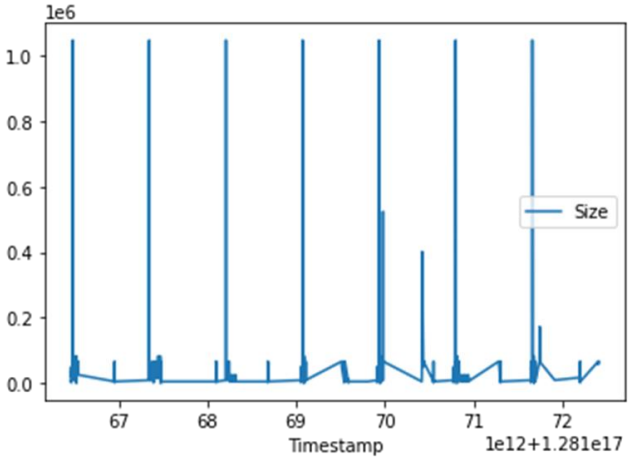

Fig.3 The example of the I/O pattern

Fig.3 은 웹 메일 애플리케이션에 대한 서버 I/O 패턴 그래프를 보여준다. 그래프에서 확인할 수 있듯이, 일정한 패턴을 가지는 것을 확인할 수 있다.

PROBLEM STATEMENT

Overview

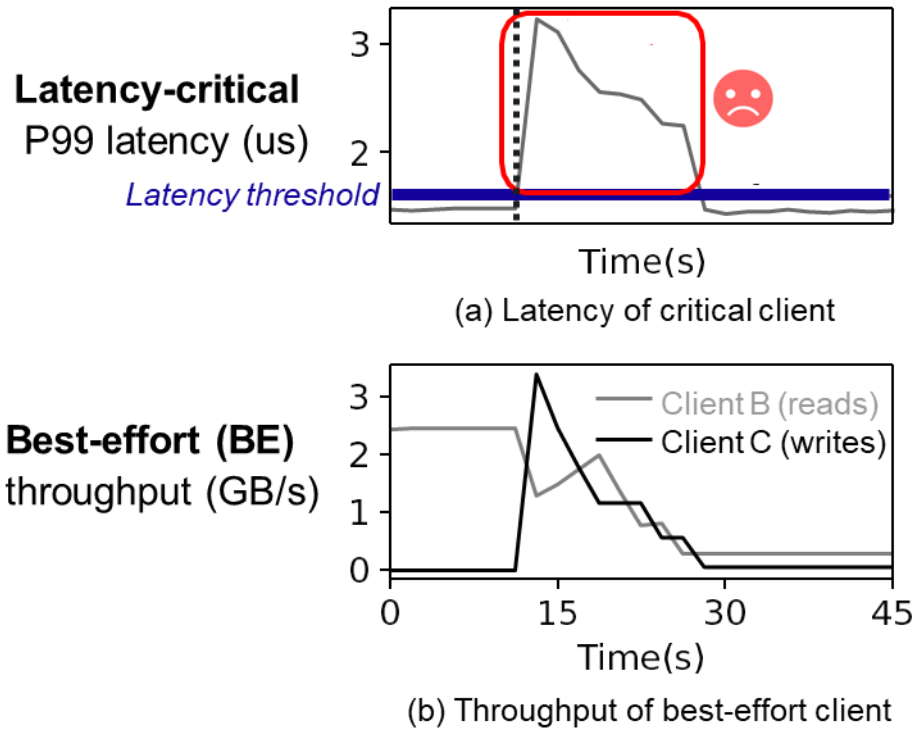

Fig.4 The example of the conventional QoS

앞서 언급한 것과 같이, 본 연구에서는 기존 QoS 의 문제점을 분석하였다. Fig. 4 는 기존 QoS 사용 시 발생하는 문제점에 대한 예시를 설명한다. (a)는 Latency critical 클라이언트의 latency 변화를, (b)는 Best-effort 클라이언트의 Throughput 변화를 보여준다. (b)에서 B 클라이언트의 Throughput 이 상승하면서 동시에 (a)의 Latency critical 클라이언트의 Latency 가 임계 값 보다 높아진다. 이후, QoS 의 메커니즘에 따라 Best-effort 클라이언트의 Bandwidth 를 제한하면서 Latency critical 클라이언트의 Latency 가 낮아진다. 이러한 동작방식은 Latency critical 클라이언트의 실제 성능이 임계 값을 넘어선 이후에 성능을 복구하기 위한 동작이 수행된다는 문제점이 존재한다. 또한 임계 값을 벗어난 이후 다시 성능이 회복될 때까지 시간이 소요된다.

Research Hypothesis

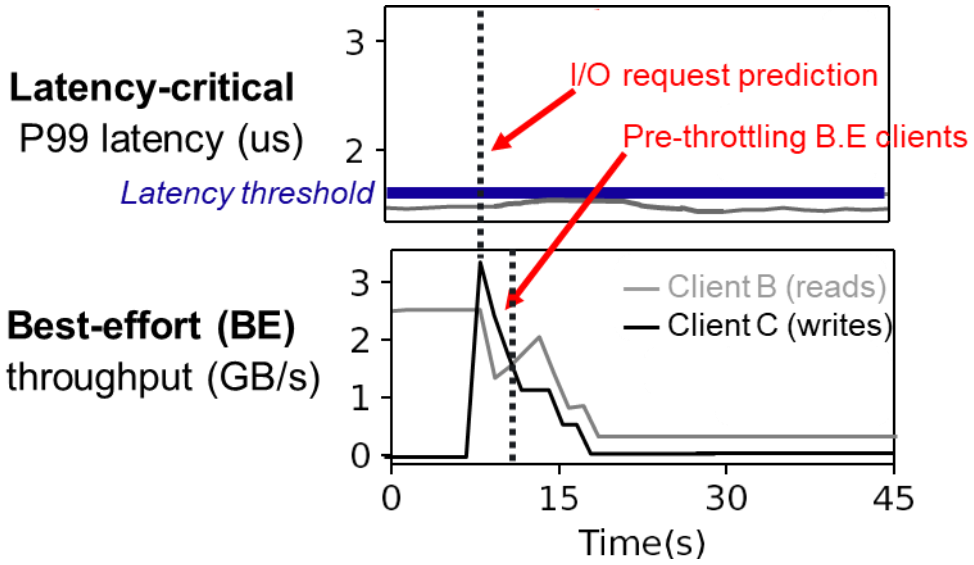

Fig.5 The hypothesis of the proposed QoS

Fig.5 는 본 연구의 제안기법인 머신러닝 기반 QoS 메커니즘을 적용했을 경우 예상되는 결과를 보여준다. 제안기법은 I/O 패턴을 학습한 머신러닝 모델을 통해 모든 클라이언트의 다음 I/O 요청 크기를 예측한다. 예측 값의 합계를 계산하여 Latency critical 클라이언트의 성능 임계 값 초과 여부를 판단한다. 임계 값을 초과할 것으로 판단할 경우, 현재 시점에서 미리 Best-effort 클라이언트의 성능을 제한한다. 이를 통해 다음 시점에서 Latency critical 클라이언트의 성능이 임계 값을 초과하는 것을 방지할 수 있다.

OBJECTIVES AND AIMS

Overall Objective

본 연구에서는 다양한 머신러닝 모델을 사용하여 성능비교를 통해 I/O 패턴 예측에 적합한 머신러닝 모델을 선택한다. Fig.2 은 모델 간 성능비교의 예시를 보여준다. 또한 다양한 튜닝기법을 적용하고 최적의 하이퍼 파라미터를 탐색하여 모델을 정확도를 향상시킨다. 이를 통해 QoS 적용 가능성을 보인다.

Specific Aims

1. RMSE(Root mean squared error)가 0.001 < RMSE < 0.01 구간의 성능을 보이는 머신러닝 모델 구현

DESIGN

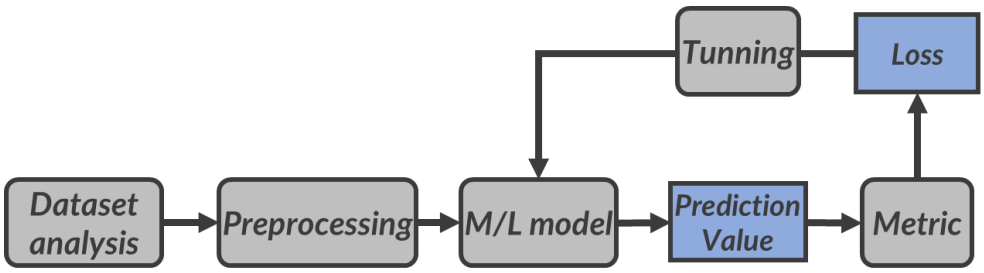

Fig.5 The hypothesis of the proposed QoS

본 연구에서는 Fig.5 와 같은 과정을 통해 예측 모델을 구성하여 실험을 진행한다. 또한 머신 러닝, 딥 러닝, 크게 2 가지 방법으로 모델을 구성하여 실험을 진행한다.

Dataset analysis

본 연구에서는 Fig.

REFERENCES

[1] Hao, M., Toksoz, L., Li, N., Halim, E. E., Hoffmann, H., & Gunawi, H. S. (2020, November). LinnOS: Predictability on Unpredictable Flash Storage with a Light Neural Network. In *OSDI* (pp. 173-190).

[2] YEUNG, Gingfung, et al. Towards GPU utilization prediction for cloud deep learning. In: *Proceedings of the 12th USENIX Conference on Hot Topics in Cloud Computing*. 2020. p. 6-6.

[3] OLY, James; REED, Daniel A. Markov model prediction of I/O requests for scientific applications. In: *Proceedings of the 16th international conference on Supercomputing*. 2002. p. 147-155.

[4] AGARWAL, Megha, et al. Active learning-based automatic tuning and prediction of parallel i/o performance. In: *2019 IEEE/ACM Fourth International Parallel Data Systems Workshop (PDSW)*. IEEE, 2019. p. 20-29.

[5] Campello, Daniel, et al. Filesystem SysCall Traces SNIA IOTTA Trace Set 5198, SNIA IOTTA Trace Repository, http://iotta.snia.org/traces/system-call?only=5198