쿠버네티스 환경에서 Cool Down Time 설정에 따른 CPU 사용량 및 서비스 응답시간 분석

권서은[1], 김동균[1], 유헌창[1]
[1]고려대학교 정보대학 컴퓨터학과
{thunnie920, kdonggyun97, yuhc}@korea.ac.kr

An Analysis of CPU Utilization and Service Latency Based on Cool Down Time Configuration in Kubernetes

Seoeun Kwon[1], Donggyun Kim[1], Heonchang Yu[1]
[1]Dept. of Computer Science and Engineering, Korea University

요 약

쿠버네티스의 HPA(Horizontal Pod Autoscaling) 기법은 워크로드의 규모에 따라 동적으로 컨테이너 리소스를 조정하여 시스템 성능을 최적화하고 자원 활용 효율성을 높이는 역할을 한다. 그러나 HPA의 주요 문제점인 pod flapping을 해결하기 위해 CDT(Cool Down Time)가 도입되었다. 본 논문은 다르게 설정한 CDT 값에 따라 각 CPU 자원 할당량 별 지연 시간과 자원 사용량을 분석했다. 실험 결과 CDT 설정값이 성능과 자원 사용량에 영향을 미치는 것을 파악했고, 이를 바탕으로 자원 사용량과 SLO(Service-level Objective) 만족률을 모두 고려한 최적의 CDT 설정의 필요성을 제고한다.

1. 서론

오토스케일링은 워크로드에 맞춰 자원을 자동으로 조정하여 과도 또는 과소 할당 된 리소스로 인한 비용 및 성능 문제를 해결하는 기술이다[1]. 컨테이너 오케스트레이션 플랫폼인 Kubernetes는 HPA(Horizontal Pod Autoscaling)를 통해 오토스케일링을 제공한다. 그러나, HPA는 짧은 시간 동안 부하가 자주 변화할 때, 잦은 리소스 해제 및 할당이 반복되는 Flapping 현상이 발생한다. 일정 시간 자원의 조정 빈도를 제한하여 자원 효율성을 높이고자 CDT(Cool Down Time)가 도입되었으나, 부적절한 CDT 설정은 자원 부족으로 인한 서비스 응답 지연, 또는 자원 과다로 인한 비용 증가 문제를 초래한다.

본 논문은 CDT 설정에 따른 응답시간과 자원 사용량의 상관관계를 분석하였다. CDT를 길게 설정할수록 응답 지연 시간은 낮았으나 CPU 사용량은 증가했다. 이를 통해 자원 사용량과 응답 시간을 모두 고려한 CDT 설정의 필요성을 제고한다.

2. HPA 및 CDT 관련 연구

HPA는 워크로드에 대한 부하가 증가하면 스케일 아웃을 통해 파드를 추가 생성하고, 부하가 감소하면 스케일 인을 통해 불필요한 리소스를 해제하여 리소스의 규모를 축소한다[2]. 그러나 워크로드의 급격한 증가와 감소에 따른 변동성으로 인해 HPA가 짧은 시간 내에 상반된 조치를 반복하는 문제를 방지하기 위해, CDT를 도입해 HPA가 빈번하게 작동하는 것을 방지한다. CDT는 이전에 생성된 파드의 효과가 나타나기 전까지 스케일링 작업 간 일정 시간을 두고, 해당 작업을 대기하게 한다[3]. 그러나, CDT가 너무 짧을 때는 파드가 자주 생성되고 종료되면서 처리하지 못하는 요청이 많이 발생한다. 반면에 CDT를 너무 길게 설정했을 때는 파드 유지 시간이 증가하며 CPU 자원을 과도하게 사용한다. 따라서, 본 논문에서는 다양한 CPU 할당 조건에서 CDT를 다양하게 설정하여 애플리케이션의 성능과 자원 할당량 관계를 비교 분석한다.

3. CDT 설정에 따른 CPU 사용량 및 서비스 응답 시간 분석

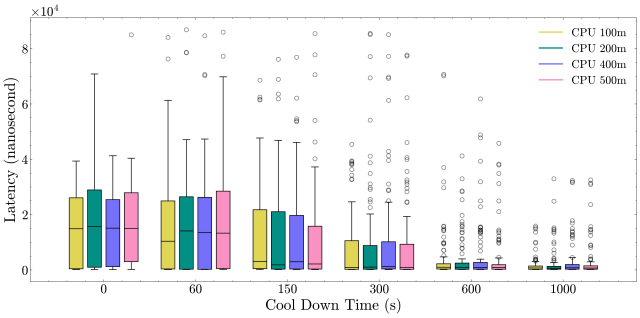
3.1 실험 환경 및 실험 설계

본 연구에서 사용되는 시스템 환경은 Ubuntu Server 22.04 LTS 운영체제를 갖는 마스터 노드 1개와 워커 노드 2개로 클러스터를 구성하였다. 각 노

드는 2 core 3.1GHz AMD EPYC™의 CPU와 4GB 메모리, 5Mbps의 대역폭을 갖는다.

실험에 사용된 애플리케이션은 PHP 기반 웹 애플리케이션이며, 스케일링이 가능한 범위로는 최소, 최대 각각 1개, 10개로 설정하였다. 애플리케이션 부하를 위해 부하 생성기를 통해 HTTP 요청을 보내어 15초당 25회부터 300회까지 GET 요청을 보냈다. 컨테이너 실행 시 최소로 필요한 자원인 CPU 할당량을 0.1, 0.2, 0.4, 0.5 코어로 변화를 주며 각 실험을 진행하였다.

3.2 CDT 설정 및 CPU 할당량에 따른 지연 시간과 자원 사용량 분석

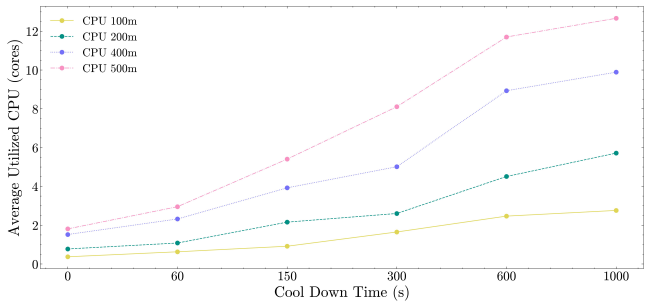


[그림 1] CPU 할당량 별 CDT에 따른 지연 시간 분포

[그림 1]은 전체 요청 중 지연 시간이 긴 상위 5%의 지연 시간에 대한 지연 시간 분포를 보여준다. CDT가 증가할수록 지연 시간이 감소하는 경향을 확인할 수 있다. 그러나, 일정 수준 이상으로 CDT가 증가할 경우, 지연 시간이 유의미하게 줄어들지 않았다. 이는 시스템 자원이 충분히 안정화된 상태에서 CDT 증가에 따른 추가적인 자원 할당은 성능 개선에 큰 영향을 주지 않기 때문이다.

[그림 2]에서는 CDT가 높게 설정될수록 워크로드에서 할당된 파드의 평균 CPU 자원 할당량이 많아지는 것을 확인할 수 있다. 동일한 CDT 설정 상황에서는 파드 하나당 기본 CPU 할당량이 많을수록 전체 평균 자원 사용량 역시 높게 나타났다. 특히 CPU 할당량이 더 높게 설정될수록 CDT 증가에 따른 평균 자원 사용량이 증가했으며, 0.1코어 대비 0.5코어 사용량이 최대 4.89배 증가했다.

이러한 현상은 자원의 효율적인 사용을 목표로 하는 오토스케일링 메커니즘과 관련이 있다. CDT가 길어질수록 파드의 생성 및 제거 횟수가 줄어들어 시스템 안정 기간이 길게 유지되고, 그에 따라 응답 지연 시간이 감소한다. 이는 HPA가 자원을 추가 및 해제하는 빈도를 줄이며 파드가 일정 기간 지속적으로 작업을 처리할 수 있기 때문이다. 그러나 CPU 사용량은 CDT가 길어짐에 따라 증가했는데, 이는 파드가 더 오랜 시간 유지되며 추가적인 자원을 소모했기 때문이다. 이러한 결과는 CDT 설정을 통한 HPA가 워크로드를 더 안정적으로 처리하게 도와주지만, 자원 사용량과 지연 시간 간의 trade-off가 발생함을 시사한다.



[그림 2] CPU 할당량 별 CDT에 따른 평균 자원 사용량

4. 결론

CDT가 길어질수록 스케일링 빈도가 줄어들어 성능이 안정화되고 응답 지연 시간이 줄어들며 SLO 만족률 향상에 도움이 되었으나, 자원 사용량이 증가하였다. 이는 CDT 설정을 통해 Flapping 현상을 줄일 수 있음을 시사한다. 그러나 CDT가 일정 수준 이상으로 증가하면 자원사용량은 증가하나, 응답 지연 측면에서 크게 개선되지 않았다. 이를 통해 적절한 CDT 설정이 시스템 성능을 개선하고 자원 낭비를 최소화하는 데 기여할 수 있을 것으로 기대된다.

Acknowledgement

본 연구는 2024년 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학사업 지원을 받아 수행되었음(2023-0-00044)

참고문헌

[1] Rampérez, Víctor, et al. "FLAS: A combination of proactive and reactive auto-scaling architecture for distributed services." Future Generation Computer Systems 118 (2021): 56-72.

[2] https://kubernetes.io/ko/docs/tasks/run-application/horizontal-pod-autoscale/

[3] Dogani, Javad, Farshad Khunjush, and Mehdi Seydali. "K-agrued: A container autoscaling technique for cloud-based web applications in kubernetes using attention-based gru encoder-decoder." Journal of Grid Computing 20.4 (2022): 40.