

지연 시간 감소와 안정적인 서비스 환경을 위한 HPA-VPA 혼합 오토스케일링 연구

한창희¹, 정현수², 장규창³, 길준민^{2,*}

¹제주대학교 물리학과, ²제주대학교 컴퓨터공학과, ³제주대학교 인공지능학과

{developersedna, jhs990909}@stu.jejunu.ac.kr, {kyuchang, *jmgil}@jejunu.ac.kr

A Study on HPA-VPA Hybrid Autoscaling for Reducing Latency and Ensuring a Stable Service Environment

Han Changhee¹, Jeong Hyunsu², Jang Kyuchang³, Gil Joon-Min^{2,*}

¹Dept. of Physics, Jeju National Univ.

²Dept. of Computer Engineerg, Jeju National Univ.

³Dept. of Artifical Intelligence, Jeju National Univ.

요 약

쿠버네티스(Kubernetes)는 현대적인 클라우드 네이티브 애플리케이션을 관리하기 위한 강력한 오케스트레이션 플랫폼으로 자리잡고 있으며, 그 핵심 기능 중 하나인 오토스케일링(Autoscaling)은 시스템 자원을 동적으로 조정하여 애플리케이션의 성능을 최적화 하는 역할을 수행한다. 본 연구에서는 쿠버네티스의 HPA(Horizontal Pod Autoscaler)와 VPA(Vertical Pod Autoscaler)를 결합한 혼합 전략이 시스템 성능에 미치는 영향을 분석하였다. 이를 위해, 기존 쿠버네티스의 기본 오토스케일링 옵션인 HPA 단독 사용과 HPA-VPA 혼합 전략을 비교하는 부하 테스트를 수행하였다.

I. 서 론

현대의 클라우드 네이티브 환경에서는 애플리케이션의 확장성과 자원 관리의 효율성이 핵심 이슈로 대두되고 있다. 특히, 예상하지 못한 급단적인 부하 상황에서 안정적인 서비스를 유지하지 못하는 것은 많은 클라우드 서비스 제공자들에게 도전 과제이다. 쿠버네티스[1]의 오토스케일링 전략인 HPA와 VPA는 각각 수평 및 수직 리소스 확장을 통해 이러한 문제를 해결하려 하지만, 단독으로 사용될 때에는 극한의 상황에서 성능 저하나 서비스 중단, 그리고 요청 실패율 증가와 같은 문제가 발생할 수 있다. [2]에서는 주로 리소스 사용량 최적화와 오버헤드 감소에 집중해 이론적인 오토스케일링 전략만을 다루었으나, 본 연구는 이러한 한계를 극복하고 요청 실패율을 줄이기 위한 목적으로 HPA와 VPA의 혼합 오토스케일링 전략을 제안한다.

본 논문의 구성은 다음과 같다: 2절에서는 쿠버네티스의 HPA와 VPA에 대한 개요를 설명하고, HPA-VPA 혼합 알고리즘을 제시한다. 그리고 k6 부하 테스트 도구[3]를 사용해 HPA 단독 사용과 HPA-VPA 혼합 전략을 적용한 경우의 성능을 비교하는 실험을 보여준다. 마지막으로, 3절에서는 본 연구의 결론과 함께 제안 전략의 시사점 및 한계점에 대해서 기술한다.

II. 제안 방법

2.1. HPA-VPA 혼합 전략

HPA는 클러스터 내에서 실행 중인 애플리케이션의 부하를 감지하고 파드(Pod)의 개수를 자동으로 조정하여 서비스의 가용성을 높이고 안정적인 운영을 가능하게 하는 오토스케일링 전략이다. HPA는 단일 파드에 장애가 생기더라도 복제본(Replica)을 생성해 서비스 중단을 방지한다. 그러나 많은 복제본이 생성되면, 이들 간의 데이터 통신과 특정 볼륨에 대한 접근이 빈번해져 오버헤드가 발생한다[3].

한편, VPA는 각 파드에 할당된 CPU 및 메모리 리소스를 자동으로 조정하여 애플리케이션이 요구하는 자원을 적절히 확보하는 전략이다. VPA는 파드의 CPU 및 메모리 리소스를 자동으로 조정해 리소스 사용 효율을 극대화하지만, 리소스 변경 중에는 서비스의 일시적 중단이 발생한다[4]. 이에 본 연구에서는 쿠버네티스에서의 효율적인 오토스케일링을 위해 HPA와 VPA를 결합한 전략을 제안한다. 이 전략은 HPA를 통해 부하가 발생하면 복제본 수를 먼저 증가시키고, 트래픽을 새로운 복제본으로 전환한다. 이후, 원래 파드에 VPA를 적용한 후, 트래픽을 다시 원래 파드로 되돌린다. 그 후 필요 없어진 복제본을 줄이거나 제거하여 리소스 효율성을 극대화한다. 이 전략은 HPA의 확장성과 VPA의 리소스 최적화를 결합하여 자원 사용을 보다 효과적으로 수행할 수 있도록 설계되었다.

2.2. HPA-VPA 혼합 알고리즘

알고리즘 1은 HPA와 VPA를 결합하여 단계적으로 오토스케일링을 수행하는 과정을 보여준다.

Algorithm 1 HPA-VPA Mixed Strategy Performance Test

```
1: Initialize HPA and VPA settings:
2:   HPA_MIN_REPLICAS  $\leftarrow$  1, HPA_MAX_REPLICAS  $\leftarrow$  10
3:   HPA_CPU_THRESHOLD  $\leftarrow$  80%
4:   VPA_CPU  $\leftarrow$  [100 millicores, 500 millicores]
5:   VPA_MEM  $\leftarrow$  [200 MiB, 1 GiB]
6: Apply HPA:
7:   Apply HPA configuration to the deployment
8: Apply VPA:
9:   Apply VPA configuration to the deployment
10: Perform performance test:
11:   Run k6 performance test with the following scenario:
12:     Gradually increase VUs from 0 to 2500 over the first 5 min.
13:     Gradually increase VUs from 2500 to 3000 over the next 10 min.
14:     Gradually decrease VUs from 3000 to 2500 over the following 10 min.
15:     Gradually decrease VUs from 2500 to 1500 over the final 5 min.
16: Restore original VPA settings:
17:   Apply original VPA configuration to the deployment
```

위 알고리즘은 다음과 같은 단계로 구성된다.

1. [설정 초기화] HPA와 VPA의 설정값을 초기화 및 각 파드의 최소 및 최대 리소스 한계를 정의한다.
2. [HPA 진행] 디플로이먼트에 HPA를 진행한 후 10초의 유휴 시간을 둔다.
3. [트래픽 이전] HPA로 생성된 복제본에 기존 파드에서 진행되는 서비스 트래픽을 옮긴다.
4. [VPA 진행] 기존 파드에 VPA를 진행하고 10초간 대기한 뒤, 서비스 트래픽을 다시 기존 파드로 옮긴다.
5. [성능 테스트] k6 도구를 사용하여 설정된 시나리오에 따라 성능 테스트를 수행한다.
6. [VPA 설정 복원] 테스트 완료 후, VPA의 설정을 원래 상태로 복원한다.

2.3. 실험 환경

본 연구에서는 Hyper-V 기반의 Ubuntu 가상 머신에서, k6 부하 테스트 도구의 동일한 시나리오로 두 개의 실험이 진행되었다. 각 실험은 1회 30분, 총 3회 진행되었고, 쿠버네티스의 오토스케일링 쿨다운은 기본 설정인 5분으로 설정되었다.

실험에 사용된 HPA의 조건은 최소 파드 수 1개, 최대 파드 수 10개이며, 각 CPU 100m, 메모리 200MiB가 할당되어 있다. VPA 조건은 CPU 최소치/최대치 각각 100millicores/500millicores, 메모리 최소치/최대치 각각 200MiB/1GiB로 설정하였다.

실험에서 사용된 k6 부하 테스트 시나리오는 가상 유저(Virtual Users, VU)의 수를 시간에 따라 변화시키는 형태로 구성하였다. 가상 유저는 노드 포트에 HTTP 요청을 전송하는 역할을 수행한다.

실험 시나리오는 다음과 같다: 먼저 처음 5분 동안 가상 유저수를 0에서 2,500까지 서서히 증가시키고, 이후 10분 동안 2,500에서 3,000까지 증가시킨다. 다음 10분 동안 다시 2,500까지 감소시키고, 마지막 5분 동안 1,500까지 감소하는 형태로 부하를 설정하였다.

이 실험은 동일한 조건 하에서 HPA와 HPA-VPA 혼합 전략의 성능을 비교하여 각 전략의 효율성을 평가하는 것을 목표로 한다.

2.3 실험 결과

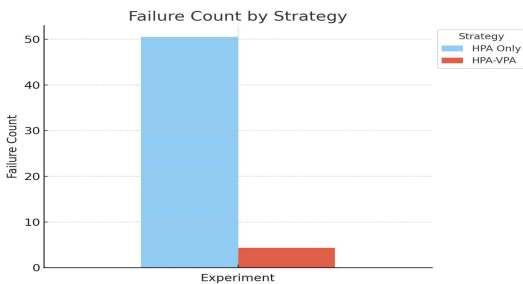


그림 1. http request 평균 실패 횟수

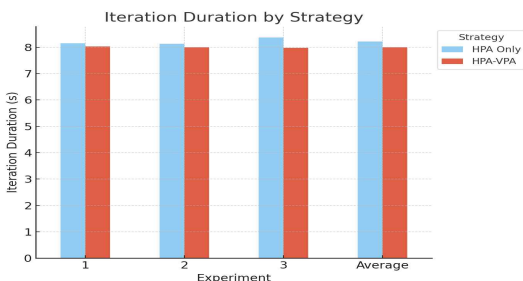


그림 2. 지연 시간 결과

그림 1은 HPA와 HPA-VPA 혼합 전략의 http request 실패 횟수를 보여준다. HPA-VPA 혼합 전략에서는 http request 실패 횟수가 평균 전체 요청 수 474,965건당 4.33회로, HPA 단독 사용의 평균 실패 횟수인 491,287건당 50.5회에 비해 현저히 낮았다. 이는 HPA-VPA 혼합 전략이 보다 높은 서비스 안정성을 확보할 수 있다는 것을 보여준다.

그림 2는 가상 사용자가 수행하는 작업 1회가 완료되는데 소요되는 시간을 나타내는 Iteration Duration을 보여준다. 본 연구에서는 세 차례의 실험을 통해 Iteration Duration을 측정하였으며, 그 결과 혼합 전략의 평균 소요 시간은 8초, 단독 사용의 경우 평균 8.22초로 나타났다. 이를 통해 혼합 전략이 단독 사용에 비해 평균적으로 더 빠른 응답 시간을 제공함을 확인할 수 있다. 이는 HPA와 VPA가 결합하여 리소스 사용을 최적화하고, 효율적인 오토스케일링을 통해 시스템 부하를 관리했기 때문으로 판단된다.

III. 결론

본 연구에서는 쿠버네티스 환경에서 극단적인 부하 상황에서도 안정적인 서비스를 제공하기 위해 HPA와 VPA를 결합한 혼합 오토스케일링 전략을 제안하였다.

HPA-VPA 혼합 전략은 평균적인 성능에서는 HPA 단독 전략보다 우수했지만, 서비스를 다른 파드로 이전하는 데 10초 단위의 시간이 매번 소요되는 점은 단점으로 작용할 수 있다.

결론적으로, VPA만을 사용할 때와 달리, HPA 혼합 전략을 사용하면 서비스의 중단 없이 파드 단위의 개별적인 리소스 관리가 가능하다는 점은 HPA 단독 사용으로는 감당하기 힘든 극단적인 부하가 들어오는 상황에서서도 VPA를 추가적으로 사용하여 대응할 수 있기 때문에, 트래픽을 이전하는 10초 단위의 오버헤드보다 서비스 중단이나 지연으로 생기는 문제가 더 큰 클라우드 환경에서는 고려할만한 여지가 있다고 판단된다. 이러한 결과는 클라우드 네이티브 환경에서 애플리케이션의 안정성과 성능 최적화를 동시에 달성하기 위한 오토스케일링 전략 설계에 중요한 시사점을 제공한다.

ACKNOWLEDGMENT

본 결과물은 2024년도 교육부의 재원으로 한국연구재단의 지원을 받아 수행된 지자체-대학 협력기반 지역혁신사업의 결과입니다(2023RIS-009). 그리고 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2022R1A2C1092934).

참 고 문 헌

- [1] <https://kubernetes.io/>
- [2] 이정민, 김정민, 강형빈, 노서영, “쿠버네티스 환경에서의 파드 스케일링 혼합 전략 및 가용 리소스 기반 스케일링 전략 선택 알고리즘,” 한국컴퓨터종합학술대회 논문집, pp. 1483-1485, 2023.
- [3] <https://k6.io/docs/using-k6/http-requests/>
- [4] 장용현, 정진원, 이재학, 유현창, 이은영, “오토스케일링 알고리즘 비교를 위한 요청 수 기반 쿠버네티스 Horizontal Pod Autoscaler 모델링,” 한국컴퓨터종합학술대회 논문집, pp. 1385-1387, 2001.