엣지 컴퓨팅 환경을 위한 쿠버네티스 오토 스케일링 기법 성능 분석

김주윤, 염성훈, 정세종, 우수연, 이경운
경북대학교

{juyun0229, saltcastle01, tpwhd3004, woosy123, kwlee87}@knu.ac.kr

A Study on the Auto-scaling technique of Kubernetes for Edge computing

Juyun Kim, Sunghoon Yeom, Se-Jong Jeong, Soo-Yeon Woo, Kyungwoon Lee
Kyungpook National Univ.

요 약

본 논문은 대표적인 컨테이너 오케스트레이션 툴인 쿠버네티스를 엣지 컴퓨팅 환경과 유사한 실험 환경에 적용하여 컴퓨팅 자원 관리를 위한 오토 스케일링 기법의 성능을 평가한다. 이를 통해, 엣지 컴퓨팅 환경에서 쿠버네티스의 오토 스케일링 기법을 활용하여 효율적인 자원 관리 및 워크로드 성능 향상을 위해 필요한 요소들을 파악한다.

Ⅰ. 서 론

쿠버네티스 [1]는 de-facto 컨테이너 오케스트레이션 툴로 클라우드 환경의 많은 서버 상에 컨테이너들을 생성하고 관리해주는 소프트웨어 플랫폼이다. 쿠버네티스에서는 컨테이너 간의 성능 간섭을 최소화하고 자원격리를 제공하기 위해 각 컨테이너에 할당되는 컴퓨팅 자원을 사용자로 하여금 설정할 수 있도록 하는 자원 관리 기법을 제공하고 있다. 이를 이용해 사용자는 각 컨테이너 단위로 CPU 코어의 개수 또는 메모리 공간을 할당할 수 있으며 할당 받은 컴퓨팅 자원을 기반으로 컨테이너 상에 수행되는 서비스는 안정적인 성능을 제공할 수 있다 [2].

엣지 컴퓨팅 환경에서는 풍부한 컴퓨팅 자원을 갖춘 클라우드 데이터센터와는 달리 제한적인 개수의 CPU 와 메모리 크기를 가지는 엣지 컴퓨팅 장치로 구성되므로 효율적인 컴퓨팅 자원에 대한 관리가 중요하다. 하지만, 기존의 자원 관리에 대한 연구들은 클라우드 데이터센터에 주로 초점을 맞추고 있어[3] 엣지 컴퓨팅 환경을 위한 컨테이너 자원 관리 연구가 필요하다.

본 논문에서는 엣지 컴퓨팅 환경과 유사한 실험 환경을 구성하고 실제 워크로드의 성능 평가를 통해 쿠버네티스의 자원 관리 기법이 워크로드의 성능에 미치는 영향을 분석한다. 실험에 적용하는 자원 관리 기법으로는 컨테이너가 생성 및 실행되는 과정에서 동적으로 할당되는 자원의 양을 조정해주는 오토 스케일링 (auto-scaling) 기법을 활용하였다. 오토 스케일링 기법은 시스템 상황에 따라 동적으로 컨테이너에 할당되는 CPU 와 메모리 자원의 양을 조절하는 기능으로 자원 활용율을 높이고 컨테이너 상에 동작하는 워크로드의 성능을 향상시키는 장점이 있다. 하지만, 본 논문의 실험 결과, 현재의 오토 스케일링 기법은 주로 CPU, 메모리에 초점을 맞추고 있고, 네트워크 대역폭에 대한 오토 스케일링은 아직 제공하고 있지 않아 자원 관리를 통한 워크로드의 성능 향상 효과가 크지 않음을 확인하였다.

네트워크 대역폭은 마이크로 서비스 구조 (micro-service architecture)와 같이 서로 다른 컨테이너들이 서로 통신하면서 사용자에게 서비스를 제공하는 경우 전체 서비스 품질을 결정하는 매우 중요한 요소로 작용한다. 따라서, 본 논문의 실험 결과를 통해 쿠버네티스의 오토 스케일링 기법이 엣지 컴퓨팅 환경에서 효율적인 자원 관리 및 워크로드의 성능 향상을 달성하기 위해서는 컨테이너의 네트워크 성능에 따라 동적으로 컨테이너의 할당되는 네트워크 대역폭 오토 스케일링 기법의 개발이 필요함을 확인하였다.

Ⅱ. 배경 지식

쿠버네티스의 오토 스케일링은 동적으로 컴퓨팅 자원을 조절하는 기능으로 조절 방식에 따라 Vertical Pod Autoscaling (VPA)와 Horizontal Pod Autoscaling (HPA)로 나뉜다. VPA 는 쿠버네티스의 최소 관리 단위인 파드(pod) [1] 의 자원 사용량을 모니터링하여 하나의 파드가 점유하는 컴퓨팅 자원의 양을 조절하는 기능이고, HPA 는 VPA 와 비슷하게 파드의 자원 사용량을 모니터링하지만 실행 중인 파드의 개수를 조절하는 기능이다. HPA 와 VPA 는 기본적으로 CPU 와 메모리 사용량과 같이 사전에 정의된 컴퓨팅 자원의 사용량 정보는 쿠버네티스의 서버 단위 에이전트인 kubelet 내의 cAdvisor 와 같은 메트릭 수집기에 의해 수집되고, 수집된 정보는 Metric API Server 를 거쳐서 HPA 와 VPA 에게 정보가 전달되어 오토스케일링이 수행된다. 오토 스케일링 기능은 파드의 기본 설정 값을 포함하는 manifest 파일에 기술되어 있는 request 값과 limit 값을

[1] 파드는 하나 또는 그 이상의 컨테이너로 이루어진다.

참고하여 스케일링을 수행한다. 이 때, request 는 파드에 할당되는 최소의 컴퓨팅 자원을 의미하며 limit 은 최대 값을 의미한다.

VPA 는 컴퓨팅 자원을 업데이트하는 방식으로 Auto 모드와 Manual 모드를 지원한다. Auto 모드는 VPA 가 파드의 자원 사용량을 모니터링한 후 메트릭을 기반으로 VPA 가 파드에 적절한 request 값과 limit 값을 도출하고, 새롭게 도출된 파드의 request 값과 limit 값으로 변경하며, 변경된 값을 적용되는 과정에서 파드는 재시작된다. Manual 모드는 Auto 모드와 비슷하게 파드에 적절한 request 값과 limit 값을 도출하지만, VPA 가 직접 변경하지 않고 적절한 파드의 request 값과 limit 값을 추천해준다..

Ⅲ. 실험 방법 및 결과

본 논문에서는 쿠버네티스의 현재 VPA 기반 오토 스케일링 기법의 한계점을 확인하기 위해 가상 머신으로 구성된 실제 실험 환경에 쿠버네티스 클러스터를 설치하여 자원 관리 기법과 오토 스케일링 기법 적용에 따른 워크로드의 성능과 자원 사용량을 측정하였다.

실험에 사용된 서버는 Intel Xeon Processor Gold 6146 @ 24 코어 3.2GHz CPU 와 DDR4 256GB 메모리를 탑재하여 4 대의 가상 머신을 운용한다. 특히, 엣지 컴퓨팅 환경과 유사한 실험 환경을 구축하기 위해 본 실험에서는 가상 머신의 CPU, 메모리와 같은 컴퓨팅 자원을 엣지 컴퓨팅 장치 수준으로 제한하였다(가상 CPU 4 개, 메모리 4 GB). 또한, 4 대의 가상 머신에 모두 우분투 20.04, 리눅스 커널 5.4 버전을 설치하고 쿠버네티스 버전 1.27.1 을 설치하였다. 4 대의 가상 머신 중 1 개는 쿠버네티스의 중앙 컨트롤러 역할인 마스터 노드 (master node)로 구성하였으며, 나머지 3 개의 실제 파드, 즉 컨테이너가 동작하는 작업 노드 (worker node)로 사용하였다.

본 실험에서는 쿠버네티스의 오토 스케일링 기법이 작업 성능에 미치는 영향을 확인하기 위해, 파드의 manifest 만을 이용하여 컴퓨팅 자원을 관리하는 상황과 추가로 VPA 가 추가로 실행되는 상황의 성능을 측정하여 비교한다. 실험에 사용한 워크로드로 대표적인 분산 키 밸류 (key-value) 저장소인 Memcached 를 활용하였다. 각 작업 노드에는 1 개의 memaslap 과 2 개의 memcached 를 하나씩 각각 배치하는데, memaslap 은 성능 측정을 위해 부하를 발생시키는 장치이고 memcached 는 memaslap 으로부터 작업 요청을 받아 이를 처리한 결과를 memaslap 에게 반환한다. 실험에서는 memcached 가 작업 요청을 받아 이를 반환하는데 걸리는 시간과 이때 사용한 컴퓨팅 자원양을 측정하였다. 실험에 사용한 메트릭은 operation per second(OPS)으로 단위 시간당 memcached 서버에서 처리하는 작업의 수를 말하며, 실험 결과로는 동일한 실험을 5 번씩 수행하여 측정한 OPS 값의 평균과 표준편차를 나타내었다.

그림 1 의 실험 결과에서 “Default”는 파드에 할당하는 컴퓨팅 자원에 대한 최대 값 (limit)과 최소 값 (request)을 정해두고 측정한 결과이고 “VPA”는 동일한 설정에서 오토 스케일링 기법을 적용한 결과이다. 컴퓨팅 자원에 대한 최대와 최소 값을 설정하는 경우 설정 값에 의해 컨테이너의 성능이 제어되는 것을 확인할 수 있다. 이 경우 OPS 는 평균 8505, 표준 편차는 51 로 나타났다. 즉, 실험을 반복하더라도 컨테이너의 성능은 거의 균일하게 달성된다. 하지만, 오토 스케일링 기법인

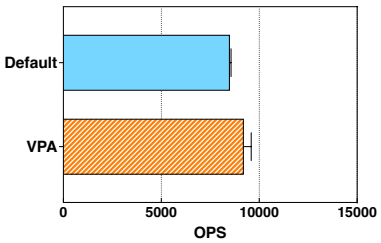

그림 1. memcached 평균 성능 및 편차

VPA 를 적용하는 경우 평균 성능 8.3%, 표준 편차가 6.6 배 늘어나는 것을 확인하였다. 그 이유는, VPA 가 적용되는 경우 컨테이너가 사용하는 CPU 나 메모리의 사용량이 소폭 변화하게 되어 컨테이너의 작업 성능에 영향을 미치기 때문이다.

그러나, 이로 인한 평균 성능의 차이는 8.3%로 그리 크지 않음을 볼 수 있다. 오토 스케일링을 적용했음에도 컨테이너의 작업 성능이 크게 증가하지 않은 이유는 현재의 VPA 가 CPU 와 메모리 자원에 대해서만 제어하기 때문이다. Memcached 와 같이 분산 처리 워크로드들을 각 서버에서 작업을 처리한 뒤 그 결과를 중앙의 컨트롤러 또는 다른 동료 작업자에 전달하기 때문에 전체 작업 성능이 네트워크 성능에 의해 영향을 받게 된다. 따라서, CPU 와 메모리 자원만을 다루는 현재의 오토 스케일링 기법은 이와 같은 분산 처리 작업의 성능 제어에는 효과가 크지 않음을 알 수 있다. 따라서, 엣지 컴퓨팅 환경에서 효율적인 작업 관리 및 제어를 위해서는 네트워크 성능 측면에서의 오토 스케일링 기법의 확장이 필요하다.

4. 결론

본 논문에서는 엣지 컴퓨팅 환경과 유사한 실제 실험 환경을 구성하여 컨테이너의 작업 성능을 측정함으로써 쿠버네티스의 오토 스케일링 기법인 VPA 를 적용하더라도 워크로드의 성능 향상이 미미함을 확인하였다. 이는 현재의 VPA 에서는 CPU 와 메모리 자원만을 고려하고 네트워크 대역폭에 대한 제어는 제공하지 않기 때문이다. 다양한 분산 처리 작업과 같은 서비스를 엣지 컴퓨팅 환경에서 효율적으로 구동하기 위해서는 쿠버네티스의 오토 스케일링 기법을 네트워크 대역폭에 대해 제어하도록 기능의 확장이 필요하다.

ACKNOWLEDGMENT

이 논문은 정부(과학기술정보통신부)의 재원으로 과학기술사업화진흥원의 지원을 받아 수행된 연구임 ('학연협력플랫폼 구축 시범사업' RS-2023-00304695).

참 고 문 헌

[1] “Kubernetes”, https://kubernetes.io/

[2] Kim, Eunsook, Kyungwoon Lee, and Chuck Yoo. "Network SLO-aware container scheduling in Kubernetes." *The Journal of Supercomputing* 79.10 (2023): 11478-11494.

[3] Xu, Cong, Karthick Rajamani, and Wesley Felter. "Nbwguard: Realizing network qos for kubernetes." Proceedings of the 19th international middleware conference industry. 2018.