

에지 클라우드에서 컨테이너 배포 속도 개선 방법

이승민¹, 노희준², 이원준³

¹고려대학교 사이버국방학과

²인하대학교 컴퓨터공학과

³고려대학교 정보보호대학원

smlee03@korea.ac.kr, hjroh@inha.ac.kr, wlee@korea.ac.kr

A method to speed up container deployments on edge clouds

Seungmin Lee¹, Heejun Roh², Wonjun Lee³

¹Dept. of Cyber Defense, Korea University

²Dept. of Computer Engineering, Inha University

³School of Cybersecurity, Korea University

요 약

컨테이너 오케스트레이션은 에지 클라우드에 컨테이너를 효율적으로 배포, 관리할 수 있게 해 주는 자동화 기술이다. 지리적으로 떨어져 있을 수 있는 에지 노드의 집합인 에지 클라우드의 규모와 복잡성이 커지면서 에지 클라우드에서 컨테이너의 고속 프로비저닝이 중요해졌다. 이때, 에지 클라우드에서 컨테이너의 배포 속도가 중요한 변수로 작용한다. 본고에서는 배포 속도를 개선하기 위해 이미지 내의 파일을 에지 노드 간에 공유하는 Peer-to-Peer 아키텍처에 기반을 둔 컨테이너 가속화 방법을 제시하고 단순화한 유량 모델을 통해 그 성능을 분석한다.

1. 서론

쿠버네티스(Kubernetes)와 같은 컨테이너 오케스트레이션 도구는 클라우드에 컨테이너(container)를 확장성 있고 빠르게 배포할 수 있게 한다. 하지만, 일반적인 클라우드 데이터 센터와 달리 지리적으로 떨어져 있을 수 있는 에지 노드의 집합인 에지 클라우드는 제한된 용량 및 사용자의 이동성과 같은 고유한 특징을 가지기 때문에, 에지 클라우드에서 컨테이너의 오케스트레이션에 대한 다양한 연구들[1, 2]이 진행 중에 있다.

특히, 에지 클라우드의 규모와 복잡성이 커지면서 에지 클라우드에서 컨테이너의 고속 프로비저닝(provisioning)이 중요해졌다. 최신 연구 중 하나인 Starlight[1]는 여러 개의 레이어가 적층된 구조로 되어 있던 기존의 도커 컨테이너 이미지 파일에서 반복적으로 등장할 수 있는 레이어들의 중복성을 피하는 방법을 제시하였다. 또한, 고비용의 왕복 요청을 필요로 하는 기존의 풀(pull) 기반 구조 대신 푸쉬(push) 기반 구조를 사용하여 워커(worker) 노드가 필요한 순서대로 지정된 파일을 전송할 수 있게 하였다. Starlight는 기존 솔루션(containerd)보다 컨테이너를 평균 3배 빠르게 프로비저닝한다.

하지만 Starlight에서는 에지 클라우드 내의 에지 노드 간의 협업을 통한 배포 시간 개선에 대해서는 고려하지 않았다. 노드 간의 협업이 워커의 부담을 증가시키고 네트워크 대역폭과 다른 자원의 오버헤드를 증가시킬 것을 우려하였기 때문이다. 그럼에도 불구하고 공개 레지스트리(registry)에서 배포하는 이미지 내 파일 중 겨우 3%만이 고유하다[2]는 점은 에지 노드 간 협업의 가능성을 시사한다. 따라서 본고에서는 에지 클라우드 내의 에지 노드 간에 이미지 내 파일 목록을 교환하여, 타 에지 노드가 파일 단위로 요청하는 경우 이미지에서 파일을 추출 후 압축하여 청크(chunk) 단위로 공유하는 컨테이너 배포 가속화 방법을 제안한다. 최근 연구인 BREAK[3]가 다소 유사하지만, BREAK는 이미지를 리팩토링 후 레이어 단위로 캐싱한다는 차이가 있다.

2. 제안 방법의 성능 분석

이미지를 배포하는 역할을 수행하는 하나의 레지스트리와, 지리적으로 근접한 에지 노드로 구성된 에지 클라우드로 구성된 시스템이 있다. 에지 노드는 배포 요청이 주어진 하나의 바인딩 노드와, 이미지를 가지고 있는 워커 노드로 구분한다. 이때 파일

목록의 공유 비용을 고려하지 않고, 파일의 전송은 [4]의 네트워크 유량 모델을 따른다고 가정한다. 또한 각 노드의 업로드 시의 속도는 제한되어 있지만 다운로드 속도는 충분히 크다고 가정한다.

이제 컨테이너 이미지 파일의 크기를 F 라고 하고, μ_R 을 레지스트리의 업로드 속도, u_w^i 는 i 번째 워커 노드의 업로드 속도라고 하자. 또, 바인딩 노드와 워커 노드는 자체 캐시에 이미지 파일의 일부를 가지고 있을 수 있으므로, 바인딩 노드 내의 캐시 히트율을 h_{local} , 타 워커 노드에서의 캐시 히트율은 h_{edge} 라고 하자. 그러면 바인딩 노드에 이미 존재하여 다운로드할 필요 없는 이미지 부분의 크기는 $F \times h_{local}$ 이며, 타 워커 노드에 존재하여 공유할 수 있는 이미지 부분의 크기는 $F \times h_{edge}$ 이다. 그러면 레지스트리로부터는 $F \times (1 - h_{local} - h_{edge})$ 만큼의 이미지 부분만 받아오면 되며, 그 다운로드 시간은 다음과 같다.

$$D_R = F \times (1 - h_{local} - h_{edge}) / \mu_R$$

이제 Peer-to-Peer 방식으로 타 워커 노드에서 이미지 부분을 공유받을 때의 시간을 구한다. 분석의 편의를 위해 공유할 이미지 부분을 워커 노드 별로 겹침 없이 분할(partition)했다고 가정하자. 이때 i 번째 워커 노드가 파일의 공유할 비율을 θ_i 이라고 하면 $\sum \theta_i = 1$ 이다. 그러면 워커 노드에서 공유한 부분을 바인딩 노드가 다운로드 하는데 걸리는 시간은 i 번째 워커 노드의 업로드 시간의 최댓값이다.

$$D_P = \max_{i \in \{1, \dots, N\}} (F \times h_{edge} \times \theta_i) / \mu_w^i$$

제안 방법에서 컨테이너 이미지는 레지스트리와 타 워커 노드들로부터 동시에 다운로드 받는 것이므로, 최소 배포 시간 D 는 두 값 중 큰 값으로 결정된다.

$$D = \max \{D_R, D_P\}$$

3. 제안 방법의 수치적 성능 평가

이 절에서는 2절에서의 분석 결과를 바탕으로 워커 노드의 수에 따른 캐시 알고리즘의 이론적 성능을 수치적으로 평가한다. 분석의 편의를 위해 θ_i 는 모두 같은 값이고, $F = 10\mu_R = \mu_w^i$ 라 가정하였다. 성능 계산에 사용된 h_{local} 과 h_{edge} 값은 IBM이 제공한 워크로드 데이터셋을 바탕으로 BREAK[3]에서 실험을 통해 구한 히트율을 차용하였다.

그림 1은 워커 노드의 개수 N 에 따라 레지스트리에서 모두 다운로드 받는 방법(fresh download)과 제안 방법(cache) 간의 파일 분배 시간을 비교한 그

래프이다. 제안 방법은 워커 노드의 개수 N 이 증가함에 따라 파일 분배 시간이 감소하지만 fresh download는 파일 분배 시간이 일정하다.

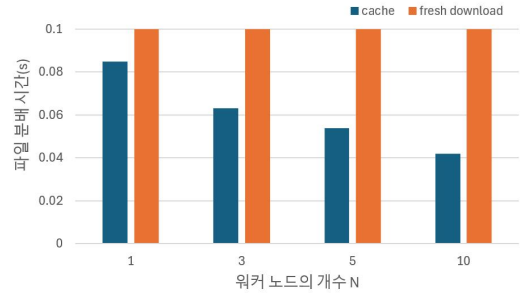


그림 1 워커 노드의 개수에 따른 파일 분배 시간

4. 결론

본고에서는 기존 컨테이너 배포 방식이 이미지 레이어의 중복된 사용을 충분히 활용하지 못하는 한계를 극복하기 위해 에지 클라우드 내의 워커 노드가 컨테이너 이미지를 파일 수준에서 공유하는 컨테이너 배포 방법을 제시하고 분석하였다.

Acknowledgment

이 논문은 2024년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구 (No. RS-2023-00234719, (SW스타랩) 서비스 연속형 지향 에지 Continuum SW 프레임워크)이며 과학기술정보통신부 및 정보통신기획평가원의 2023년도 암호화사이버위협대응기술연구개발사업(RS-2023-00235509, ICT융합 공공 서비스·인프라의 암호화 사이버위협에 대한 네트워크 행위기반 보안관계 기술 개발)의 연구결과로 수행되었음.

참고문헌

- [1] J. L. Chen *et al.*, "Starlight: Fast Container Provisioning on the Edge and over the WAN," in *Proc. of USENIX NSDI*, Apr. 2022.
- [2] N. Zhao, *et al.*, "Large-Scale Analysis of Docker Images and Performance Implications for Container Storage Systems," *IEEE TPDS*, Vol. 32, No. 4, Apr. 2021.
- [3] Y. Feng, *et al.*, "BREAK: A Holistic Approach for Efficient Container Deployment among Edge Clouds," in *Proc. of IEEE INFOCOM*, May 2024.
- [4] R. Kumar and K. W. Ross, "Peer-Assisted File Distribution: The Minimum Distribution Time," in *Proc. of IEEE HotWeb*, Nov. 2006.