



엣지 컴퓨팅 환경을 위한 경량화 컨테이너 런타임 설계 및 성능 검증

Changin Kim*, Youjeong Heo, Changbeom Choi
Department of Computer Engineering, Hanbat National University

엣지 컴퓨팅

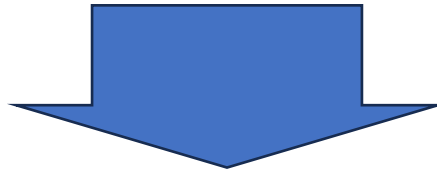
- 엣지 컴퓨팅은 데이터를 생성하는 기기 근처에서 처리하는 분산형 컴퓨팅 구조
- 자율 주행, IoT, 스마트 시티와 같은 실시간 응답성이 요구되는 분야에서 광범위하게 활용
- 엣지 장치의 **제한된 자원으로 높은 성능과 저비용 자원 관리를 수행하는 것이 요구**

컨테이너 기술

- 컨테이너 기술은 경량화된 가상화를 가능하게 하여 자원 소모를 줄이고, 가상 머신에 비해 더 빠른 실행 속도를 제공
- 엣지 환경에서 필요한 애플리케이션 격리와 유연성을 확보하는 데 적합한 솔루션으로 평가
- 그러나 Docker와 같은 기존 런타임은 클라우드에 최적화되어 있어 엣지 환경에서는 **과도한 자원 소모 문제가 발생**할 수 있습니다.

경량화 컨테이너 런타임

- 본 연구에서는 특수한 요구 사항을 충족하기 위해 최적화된 **경량화 컨테이너 런타임을 제안**
- 최적화된 컨테이너 기술의 가능성을 탐구하고, 실사용 환경에서의 응용 가능성을 제시하고자 함



제안한 컨테이너 런타임을 구현하고, ARM 기반의 저사양 장치(Raspberry 4B)에서 Docker와의 비교연구를 통해 애플리케이션 시작 속도, 메모리 사용량, CPU 사용량을 측정

Proposed System

System Architecture

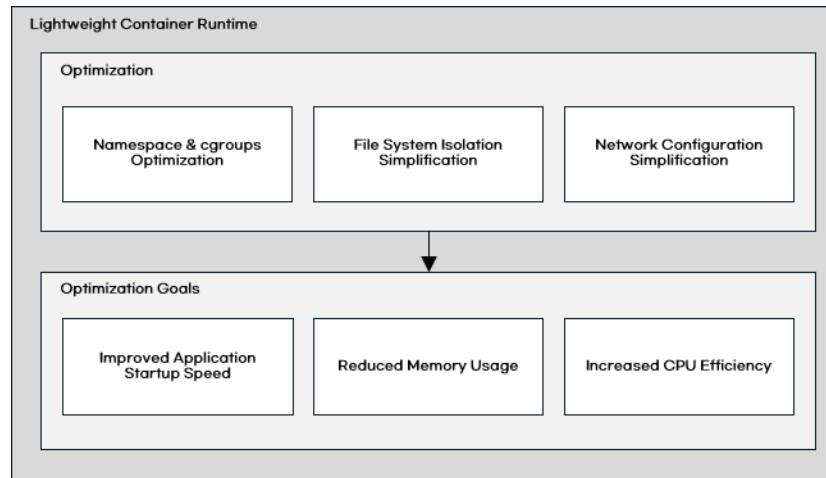


그림 1 경량형 컨테이너 런타임 설계도

- 네임스페이스 및 cgroups
 - 사용자 네임스페이스 격리를 단순화하고, 프로세스 네임스페이스 기본적인 독립성만을 유지하도록 하여 자원소모를 줄임
 - cgroups의 자원 할당을 제한하여 불필요한 CPU 및 메모리 소비를 줄임
- 파일 시스템
 - 일반적인 Docker 환경에서는 overlay2와 같은 파일 시스템 격리를 사용하여 다층적인 파일 시스템 구조를 유지함
 - 간단한 bind mount 방식을 사용하여 파일 시스템 격리를 단순화함
- 네트워크 설정
 - 다중 네트워크 브리지 및 가상 네트워크 인터페이스를 제거하고 단일 인터페이스로 엣지 장치에 직접 연결

성능 지표

Performance Metric	Measurement Tool	Unit
Startup Time	time command	sec
Memory Usage	cgroups monitoring feature	MB
CPU Usage	cgroups monitoring feature	%

표 1 성능 지표와 측정 방법

- 애플리케이션 시작 속도, 메모리 사용량, CPU 사용량을 비교
- 애플리케이션 시작 속도는 애플리케이션이 시작되는 데 걸리는 시간을 time 명령어로 측정
- 메모리 사용량과 CPU 사용량은 cgroups의 실시간 모니터링 기능을 통해 측정
- 각각 MB 및 % 단위로 기록

Case Study

실험 절차

```
changin@worker1:~$ time docker
ae398166b426334ec36cc0176771a64
real    0m0.273s    MEM USAGE
user    0m0.017s    18.2MiB /
sys     0m0.013s

● changin@worker1:~/come-capstone24
Container started successfully
Execution time: 20.938438ms
Maximum resident set size (kbytes): 7280
real    0m0.031s
user    0m0.009s
sys     0m0.004s
```

그림 2 실험 절차 일부

- Raspberry Pi 4B 장치 사용, 운영체제로는 Ubuntu Server 20.04를 설치
- 두 런타임에서 동일한 애플리케이션을 실행
- Docker 기준 데이터 기록, 경량화 런타임 실시간 데이터 기록 후 평균값을 산출하여 성능차이를 비교하였음

실험 결과 및 분석

Performance Metric	Docker	Lightweight Runtime	Improvement (%)
Startup Time (sec)	3.5	2.4	31.4
Memory Usage (MB)	120	90	25
CPU Usage (%)	75%	60	20

표 2 성능 평가 결과(평균값)

- 애플리케이션 시작 속도가 평균 30% 이상 개선
- 메모리와 CPU 사용량은 각각 25%와 20% 절감
- 실시간 응답성과 자원 절약 측면에서 기존의 한계 극복에 대한 가능성을 보여줌

Conclusion

- 본 연구는 엣지 컴퓨팅 환경에서의 자원 효율성과 빠른 애플리케이션 시작 속도를 실현하기 위한 경량화 컨테이너 런타임의 설계와 성능을 평가함
- 향후 연구로는 다양한 환경에서 경량화 컨테이너 런타임의 성능을 더욱 최적화하고 특정 애플리케이션 요구사항에 맞춘 맞춤형 자원 관리 방안을 제시하는 방향으로 확장할 것임

Ackowldgement

“본 연구는 2024년 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학사업의 연구결과로 수행되었음”(2022-0-01068)

Reference

- [1] H. M. Park and T. H. Hwang, "Changes and trends in edge computing technology", The Journal of The Korean Institute of Communication Sciences, vol. 36, no. 2, pp. 41-47, 2019.
- [2] J. H. Hong, K. C. Lee and S. Y. Lee, "Trends in edge computing technology", J. Electron. Telecommun. Trends Anal., vol. 35, no. 6, pp. 78-87, 2020. doi: 10.22648/ETRI.2020J.350608
- [3] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey and S. Linkman, "Systematic literature reviews in software engineering—a systematic literature review", Inf. Softw. Technol., vol. 51, no. 1, pp. 7-15, 2009.
- [4] K. Petersen, R. Feldt, S. Mujtaba and M. Mattsson, "Systematic mapping studies in software engineering", Proc. Int. Conf. Eval. Assessment Softw. Eng., pp. 68-77, 2008.