실시간 데이터 분석을 위한 컨테이너 기반 가상화 성능에 관한 연구†

최보아*, 한재덕*, 오다솜*, 박현국*, 김현아*, 서민관**, 이종혁*‡
*대구가톨릭대학교 인공지능·빅데이터공학과
**티쓰리큐(주)
e-mail : jonghyuk@cu.ac.kr

A Study on Performance Evaluation of Container-based Virtualization for Real-Time Data Analysis

BoAh Choi*, JaeDeok Han*, DaSom Oh*, HyunKook Park*, HyeonA Kim*, MinKwan Seo**, JongHyuk Lee*
*Dept. of Artificial Intelligence & Big Data Engineering, Daegu Catholic University
** T3Q

요 약

본 논문은 실시간 데이터 분석을 위한 컨테이너 가상화 기술 사용에 대한 효용성을 알아보기 위해 HDP 와 MapR 배포판에 포함된 Spark 를 도커라이징 전과 후 환경에 설치 후 HiBench 벤치마크 프로그램을 이용해 성능을 측정하였다. 그리고 성능 측정치에 대해 대응표본 t 검정을 이용하여 도커라이징 전과 후의 성능 차이가 있는지를 통계적으로 분석하였다. 분석 결과, HDP 는 도커라이징 전과 후에 대한 성능 차이가 있었지만 MapR 은 성능 차이가 없었다.

1. 서론

도커[1]는 운영체제 레벨 가상화인 컨테이너 기술을 제공하는 소프트웨어로서 PaaS(Platform as a Service)의 일종이다. 컨테이너 가상화 기술은 클라우드 컴퓨팅을 가능하게 하는 중요한 기반 기술로 경량성, 성능, 이식성 등의 측면에서 하드웨어 가상화 기술 대비 우수하다. 또한 애플리케이션 실행에 대한 동일 환경 제공이 가능하여 애플리케이션의 개발, 배포, 운영의 효율화를 위해 최근 널리 활용되고 있다.

스파크[2]는 대규모 데이터 처리를 위한 통합 분석 엔진이다. 스파크는 인 메모리 캐싱 등 하둡의 배치 처리보다 최적화된 실행을 지원하여 실시간 데이터 처리 및 분석에 많이 활용된다.

본 논문은 컨테이너 가상화 기술을 이용한 실시간 데이터 분석의 성능을 분석하여 컨테이너 가상화 기술의 효용성을 알아보고자 한다. 이를 위해 본 논문은 하둡 배포판 중 HDP[3]와 MapR[4] 내 포함된 스파크를 도커 적용 전과 후 환경에 각각 구축하고 벤치마킹 프로그램인 HiBench[5]를 이용하여 성능을 측정한다. 스파크 도커라이징 환경의 적용 전과 후의 비교를 위해 실행 시간과 처리량에 대해 통계적 검증 방법인 T-test 로 가설을 세워 성능 차이 유무를 비교 분석하여 컨테이너 가상화 기술의 효용성을 판단한다.

본 논문의 2 절에서는 HDP 와 MapR 기반 스파크 환경 도커라이징 전과 후의 성능 차이 존재 여부를 확인하기 위해 이용한 대응표본 t 검정을 설명하고 3 절에서는 실제 데이터를 이용하여 도커라이징 전과 후의 성능 차이가 있는지 통계 분석한 결과를 설명한다. 마지막으로 4 절에서는 결론을 맺는다.

2. 관련연구

t 검정은 t 분포에 의존하여 두 집단 간의 평균이 통계적으로 유의미한 차이를 보이는지 판별한다. 대응표본 t 검정은 한 집단내에 두 개의 변수를 비교하기 위해 일반적으로 한 집단의 사전, 사후에 대한 차이를 분석할 때 사용하는 통계적 기법이다. 본 논문은 HDP 와 MapR 기반으로 스파크 도커라이징 환경 적용 전과 후의 성능 차이를 분석하기 때문에 대응표본 t 검정(paired t-test) 사용이 적합하다.

3. 실험 환경 및 결과

본 논문은 스파크 도커라이징 환경 적용 전과 후

† 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW 중심대학지원사업의 연구결과로 수행되었음(2019-0-01056).
‡ 교신저자

성능의 차이가 존재하는지 분석하기 위해 Host OS 와 Docker 에 HDP 와 MapR 기반 Spark 환경을 <표 1>과 같이 각각 구축하였다. 그리고 HiBench 벤치마크 프로그램을 사용하여 각각의 워크로드를 10 번 실행 후 실행시간과 처리량을 수집하였다. HiBench 는 Micro, ML, SQL, Graph, Websearch 및 Streaming 의 6 가지 범주로 총 27 개의 워크로드로 구성되어 있다. 본 논문에서는 Micro 의 WordCount 와 Sort 워크로드를, Websearch 의 PageRank 워크로드를 사용하였다.

<표 1> 실험 환경

구분	상세 스펙 및 버전 Host OS / Docker
Hardware	AWS EC2 instance: t2.large (vCPUs: 2, RAM: 8GB, General Purpose SSD)
Software	Ubuntu 16.04 HDP 2.6.5 MapR 5.2.2 OpenJDK 1.8.0_222 Docker 18.09.7 HiBench 7.0

본 논문은 도커라이징 전과 후의 실행 시간과 처리량 성능에 대한 차이를 알아보기 위해 다음과 같이 가설을 세우고 대응표본 t 검정을 실행하여 분석하였다.

- H_0: 도커라이징 전과 후의 성능에 대한 차이가 없다.
- H_1: 도커라이징 전과 후의 성능에 대한 차이가 있다.

3.1 HDP 기반 Spark 의 도커라이징 전, 후 실행시간 차이 분석

HDP 기반 Spark 의 도커라이징 전과 후의 WordCount, Sort, PageRank 의 실행시간을 대응표본 t 검정 수행 결과, <표 2> 와 같이 HDP 기반 Spark 의 도커라이징 전 WordCount, Sort, PageRank 실행시간 평균은 각각 7.521, 7.294, 7.623 이고, 후의 평균은 각각 8.965, 8.838, 9.424 이다. 유의수준 α=0.05 에서 전과 후의 차이에 대한 유의확률을 검정 하였을 때, 유의확률은 각각 5.645e-07, 1.798e-07, 1.616e-06 이므로 모두 H_1 을 채택한다. 따라서, 도커라이징 전과 후의 성능에 대한 차이가 있다고 할 수 있다.

<표 2> HDP 기반 Spark 의 도커라이징 전, 후 실행시간 차이 분석 결과 (단위: 초)

	전 평균	후 평균	유의확률
WordCount	7.521	8.965	5.645e-07
Sort	7.294	8.838	1.798e-07
PageRank	7.623	9.424	1.616e-06

3.2 HDP 기반 Spark 의 도커라이징 전, 후 처리량 차이 분석

HDP 기반 Spark 의 도커라이징 전과 후의 처리량을 대응표본 t 검정 수행 결과, <표 3>과 같이 도커라이징 전 처리량 평균은 각각 4909.5, 4900.5, 1408.7 이고, 후의 평균은 각각 4054.1, 4098.6, 1149.5 이다. 유의수준 α=0.05 에서 전과 후의 차이에 대한 유의확률을 검정 하였을 때, 유의확률은 각각 4.543e-06, 3.695e-07, 2.663e-05 이므로 모두 H_1 을 채택한다. 따라서, 도커라이징 전과 후의 성능에 대한 차이가 있다고 할 수 있다.

<표 3> HDP 기반 Spark 의 도커라이징 전, 후 처리량 차이 분석 결과

	전 평균	후 평균	유의확률
WordCount	4909.5	4054.1	4.543e-06
Sort	4900.5	4098.6	3.695e-07
PageRank	1408.7	1149.5	2.663e-05

3.3 MapR 기반 Spark 의 도커라이징 전, 후 실행시간 차이 분석

MapR 기반 Spark 의 도커라이징 전과 후의 실행시간을 대응표본 t 검정 수행 결과, <표 4>와 같이 도커라이징 전 실행시간 평균은 각각 10.957, 10.987, 10.7481 이고, 후의 평균은 각각 10.803, 10.848, 10.930 이다. 유의수준 α=0.05 에서 전과 후의 차이에 대한 유의확률을 검정 하였을 때, 유의확률은 각각 0.4082, 0.3689, 0.3193 이므로 모두 H_0 을 채택한다. 따라서, 도커라이징 전과 후의 성능에 대한 차이가 없다고 할 수 있다.

<표 4> MapR 기반 Spark 의 도커라이징 전, 후 실행시간 차이 분석 결과 (단위: 초)

	전 평균	후 평균	유의확률
WordCount	10.957	10.803	0.4082
Sort	10.987	10.848	0.3689
PageRank	10.7481	10.930	0.3193

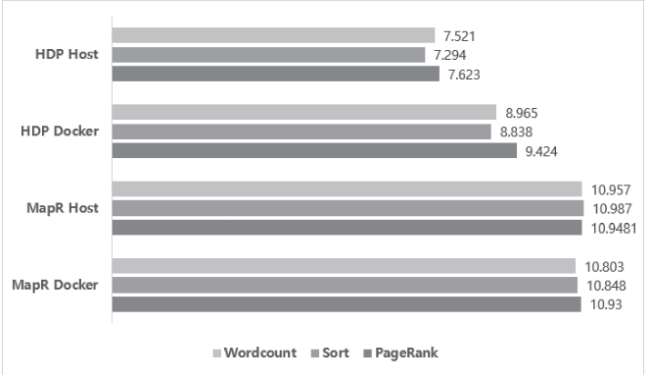
3.4 MapR 기반 Spark 의 도커라이징 전, 후 처리량 차이 분석

MapR 기반 Spark 의 도커라이징 전과 후의 처리량을 대응표본 t 검정 수행 결과, <표 5>와 같이 도커라이징 전 처리량 평균은 각각 3394.7, 3384.9, 1007.6 이고, 후의 평균은 각각 3296.5, 3318.1, 989.6 이다. 유의수준 α=0.05 에서 전과 후의 차이 대한 유의확률을 검정 하였을 때, 유의확률은 각각 0.1606, 0.1999, 0.2485 이므로 모두 H_0 을 채택한다. 따라서, 도커라이징 전과 후의 실행시간에 대한 차이가 없다고 할 수 있다.

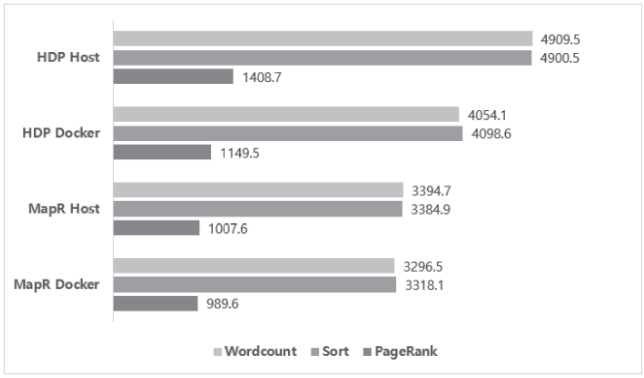
<표 5> MapR 기반 Spark 의 도커라이징 전, 후 처리량 차이 분석 결과

	전 평균	후 평균	유의확률
WordCount	3394.7	3296.5	0.1606
Sort	3384.9	3318.1	0.1999
PageRank	1007.6	989.6	0.2485

위의 분석 결과에 따르면 HDP 기반의 스파크 도커라이징 환경 적용 전과 후에는 성능에 대한 차이가 있는 반면 MapR 기반의 스파크 도커라이징 환경 적용 전과 후에는 성능에 대한 차이가 없다. (그림 1)은 도커라이징 환경 적용 전과 후의 실행시간 평균을 보여준다. (그림 1)에서 보듯이 HDP 는 도커라이징 환경 적용시 실행시간에 차이가 크지만 MapR 은 작음을 알 수 있다. (그림 2)는 도커라이징 환경 적용 전과 후 처리량의 평균을 보여준다. (그림 2)에서 보듯이 HDP 가 MapR 보다 처리량 평균 차이가 큼을 알 수 있다.



(그림 1) 도커라이징 전과 후의 실행시간 평균 (단위: 초)



(그림 2) 도커라이징 전과 후의 처리량 평균

4. 결론 및 향후 과제

본 논문은 컨테이너 가상화 기술을 이용한 실시간 데이터 분석의 효용성을 알아보고자 실제 HDP 와 MapR 기반 스파크 도커라이징 환경 적용 전과 후의 성능 데이터 분석을 통해 성능에 차이가 존재하는지 알아보았다. 분석 결과, HDP 는 오픈소스라는 강점이 있지만, 도커라이징 환경을 적용하였을 때 처리량은 감소하고, 실행 시간은 늘어나는 것을 알 수 있었다. 즉 HDP 는 도커라이징 환경을 적용했을 때 성능적인 부분에서 오버헤드가 존재함을 의미한다. MapR 은 하둡 배포판의 상용 버전으로 서비스 이용시에 비용이 발생하지만, 도커라이징 환경을 적용했을 때에 처리량과 실행 시간의 성능 저하가 발생하지 않는다. 따라서 도커라이징 환경을 적용하고자 한다면 MapR 이 HDP 보다 성능적인 면에서 더 효율적이라고 할 수 있다.

본 논문은 계속해서 성능 차이 여부에 대한 원인을 분석하고자 한다.

참고문헌

[1] Dirk Merkel. Docker: lightweight Linux containers for consistent development and deployment. Linux Journal,

2014.
[2] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. Spark: cluster computing with working sets. In Proc. HotCloud ’10, 2010.
[3] Hortonworks Data Platform (HDP), https://www.cloudera.com/downloads/hdp.html
[4] MapR, https://mapr.com/
[5] HiBench, https://github.com/Intel-bigdata/HiBench