〔별지 제5호서식〕

DCU DAEGU CATHOLIC UNIVERSITY

SW캡스톤디자인 과제 결과보고서

SW 대구가톨릭대학교 소프트웨어중심대학사업단

과제분야	■ 기업연계형 과제 (업체명 : 티쓰리큐(주))						
과제유형	□ 시제품 개발 및 제작 □ 분석, 연구, 실험, 논문 □ 디자인 개발 및 제작 □ 기타( )						
교과목명	운영체제(캡스톤디자인)						
과제명	컨테이너 환경에서 GPU 성능 측정에 관한 연구						
팀명	GPU						
팀장	배채정						
참 여 학 생 명 단							
연번	**소속학과(전공)**	**학번**	**학년**	**성별**	**성명**	**연락처**	**E-mail**
1	인공지능빅데이터공학과	17155636	3	남	김희규	01023382091	gmlrb2091@cu.ac.kr
2	인공지능빅데이터공학과	15106234	4	남	나동현	01037107732	nadonge@cu.ac.kr
3	인공지능빅데이터공학과	19126880	3	여	배채정	01053718720	bcj7231@cu.ac.kr
4	인공지능빅데이터공학과	19127029	3	남	이순주	01088725715	soonju0304@cu.ac.kr
5	인공지능빅데이터공학과	18118046	2	남	전효성	01087383369	gytjdwjs@cu.ac.kr
6							
7							
8							
9							
10							

집행금액(원)				
과제수행기간	2021년 03월 ~ 2021년 07월 (4개월)			
참여기업 멘토	**소속**	티쓰리큐(주)	**성명**	서민관
	연락처	02-6344-7660		
교과목 담당교수	**소속**	인공지능·빅데이터공학과	**성명**	이종혁
	연락처	053-850-2882		

상기의 내용과 같이 SW캡스톤디자인 과제 결과보고서를 제출합니다.

별첨 : 1. SW캡스톤디자인 과제 수행 결과보고서 1부

2. SW캡스톤디자인 과제 수행 결과물 1부. 끝.

2021년 월 일

과제수행팀 팀장 : 배채정 (인 또는 서명)

교과목 담당교수 : 이종혁 (인 또는 서명)

대구가톨릭대학교 SW중심대학사업단장 귀하

〔별지 제5호서식〕 - 〔별첨 1〕

SW캡스톤디자인 과제 수행 결과보고서

1. 과제 수행 결과보고

과 제 명	컨테이너 환경에서 GPU 성능 측정에 관한 연구

□ 과제 개요 및 필요성

. 티쓰리큐㈜는 인공지능 및 빅데이터 플랫폼을 컨테이너 환경에서 개발・배포・운영하기 위해 컨테이너 가상 자원과 애플리케이션에 대한 성능측정 방법의 조사가 필요하다.

. GPU는 PC 환경에서 게임 및 멀티미디어 성능을 가속화 하는 등 컴퓨터에서 중요한 자리에 위치해 있다.

. CPU보다 단순하지만, 더 많은 코어를 가지는 GPU는 비용 효율성이 좋고 다수의 코어로 인한 임시저장 데이터의 메모리 요구량이 증가하게 되므로 효율적인 사용을 목적으로 한다.

. 따라서 컨테이너 환경에서 그래픽 카드인 GPU의 성능 측정을 목적으로 연구한다.

□ 과제의 개발 방법 및 과제 수행 과정

▶ 개발 방법

. AWS EC2의 컨테이너 환경에서 Deep Learning AMI(Ubuntu 18.04) Version 43.0 선택, g3s.xlarge 인스턴스 생성

. 딥러닝 프레임 워크로 Tensorflow 채택

. Tensorflow 사용을 위해 Tensorflow tutorial 이용,
Tensorflow tutorial 속 딥러닝 모델인 CNN 기본모델 소스 코드를 불러옴

. Python 시간 모듈 time을 적용하여 CNN 모델 실행 전과 실행 후의 시간을 비교, 60회 반복 후 평균값 및 그래프를 도출

▶ 과제 수행 과정

(1) AWS EC2 인스턴스 생성

(2) tensorflow, tensorflow-gpu 및 GPU가속을 위한 NVIDIA GPU drive, NVIDIA CUDA, cuDNN 설치

(3) Mnist데이터와 CNN 모델을 포함한 소스 코드 (test.py) 실행 및 시간 측정
(모델의 시간을 측정하는 실험이기 때문에 CNN 모델의 정확도와 오버피팅 등 외적인 부분은 고려하지 않는다.)

(4) 도커 컨테이너 환경에서 1~3번 과정을 똑같이 진행 후 결과를 도출, 비교(60회 반복)

(5) 도출한 결과 데이터를 바탕으로 그래프 화
(단, 극단치 제외__40(시간) 이하의 데이터만 평균 도출 및 그래프 화)

□ 결과

▶ 예상

. 도커 컨테이너 환경일 때와 Host 환경일 때의 결과에 유의미한 차이가 있을 것이라 예상

. 또한 도커 컨테이너 환경에서는 클라우드 시스템을 사용하기에 좀 더 최적화되어있다고 예측

. 도커 컨테이너 환경에서 좀 더 유연한 연산, 빠른 속도를 요구하는 것으로 예상

. 도커 컨테이너 환경일 때 가상화 오버헤드가 발생하여 Host 환경일 때보다 속도가 느려질 것

▶ 결과

. Host 환경일 때와 도커 컨테이너 환경일 때 결과의 차이가 유의미하지 않음

. 가상화의 오버헤드가 별 영향을 주지 않음

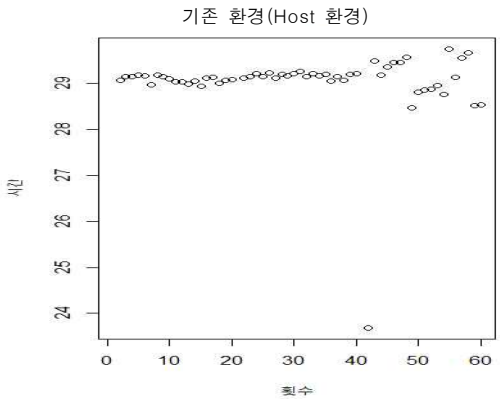
[결괏값 평균]

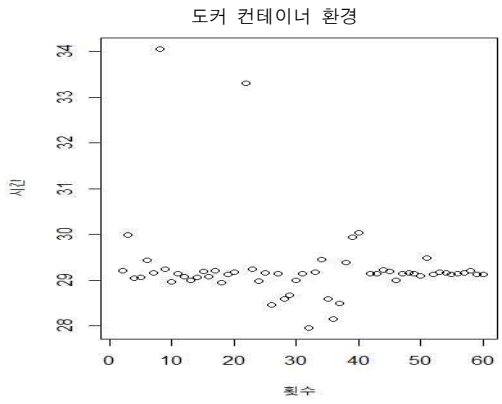
기존 환경(Host 환경)일 때 : 29.034

도커 컨테이너 환경일 때 : 29.267

. 도커 컨테이너 환경일 때가 Host 환경일 때에 비해 (0.233/29.034=0.008)
0.8% 느리게 나타남

[결과 그래프]


기존 환경(Host 환경)
시간
24
25
26
27
28
29
0
10
20
30
40
50
60
횟수


도커 컨테이너 환경
시간
28
29
30
31
32
33
34
0
10
20
30
40
50
60
횟수

2. 활용방안 및 기대효과

□ 활용방안

. 팀 [GPU]는 도커 컨테이너 환경과 도커 컨테이너 환경이 아닐 때 GPU 성능 차이에 대해 실험을 진행하였지만 유의미한 차이가 거의 없다는 결과를 도출

. 따라서 인공지능, 빅데이터 플랫폼을 컨테이너 환경에서 개발, 배포, 운영하기 위해 프로젝트의 규모, 예산 등 여러 부분을 고려하여 컨테이너 환경을 선택할 것을 권장

. 또한 인공지능, 딥러닝 개발(모델 생성 및 결괏값 도출)에서 데이터를 학습시키는 과정에서 GPU의 선택, 컨테이너 환경을 선택하는 것과 달리 데이터의 정제방법이나 신경망 모델의 선택, 모델의 구조(epoch, 파라미터) 등에도 집중을 하는 것이 좋은 방안이 될 것임

□ 기대효과

. 컨테이너 환경에서 GPU의 성능에 대한 차이가 없음을 확인

. GPU 시간 단축하기 위해서는 코어 수가 많은 GPU를 사용하거나 딥러닝의 모델 설정, 데이터 정제방법에 집중할 필요가 있음

. 인공지능, 빅데이터 플랫폼을 개발·배포·운영하기 위한 예산을 준비하여 컨테이너 환경을 사용하면 큰 문제점이 생기지 않을 것

. 그렇기에 활용방안에서 이야기한 데이터의 정제, 신경망의 모델 선택, 모델 구조를 유연하게 설정하여 시간 단축에 신경을 쓸 필요가 있음

. 개발과정과 배포 과정에서 드는 시간이 적어지기 때문에 프로젝트에 대해서는 목표와 이상향을, 소비자들에게는 빠르게 배포된 시스템을 사용 할 수 있음을 기대

□ 기타

[test.py 파일 내용]

```
import tensorflow as tf
import time
from tensorflow.keras import datasets, layers, models
```

```
(train_images, train_labels), (test_images, test_labels) = datasets.mnist.load_data()
train_images = train_images.reshape((60000, 28, 28, 1))
test_images = test_images.reshape((10000, 28, 28, 1))
# 픽셀 값을 0~1 사이로 정규화합니다.
train_images, test_images = train_images / 255.0, test_images / 255.0
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(10, activation='softmax'))
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
start = time.time()
model.fit(train_images, train_labels, epochs=5)
print(time.time() - start)
```

3. 예산 집행내역

구분	사용항목	구입물품	사용목적(상세기재)	금액(원)
과제 운영비	재료구입	AWS(Amazon Web Services)	과제수행을 위한 분석용 클라우드 서버 구입	400,000
자문료	전문가 활용비	기술지도비 및 자문료	과제수행을 위한 멘토 및 전문가 기술지도	200,000
합 계				**600,000**

※ 중간보고 시 제출한 집행내역을 포함한 전체 예산 집행내역 작성