GPU on kubernetes

2018. 06. 29

이종철 책임





Contents

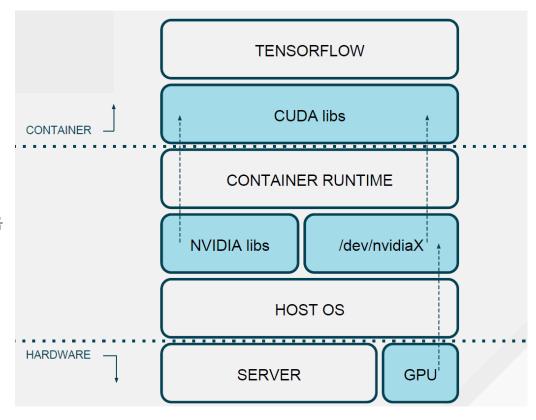
kubernetes≌ GPU

kubernetes GPU PoC

GPU Container

GPU Container 실행 요구사항

- kernel module
 GPU device driver
- user-level module
 CUDA Libs / NVML Libs ...
- kernel module은 container에 포함하지 않음
- user-level module은 kernel module에 의존도가 있음



출처 : Automating GPU Infrastructure for kubernetes & container linux , Lucas Serven Marin, 2018

kubernetes roadmap for GPU

Alpha GPU Support (--accelerators)

- ✓ Experimental support

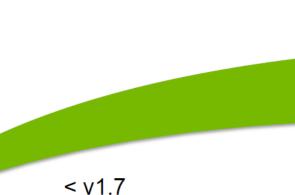
 (1.6 supports one GPU / node)
- Manually mount the volumes in your pod spec
- ✓ No GPU Monitoring or Health check

Device Plugin system

- ✓ NVIDIA Device Plugin (alpha)
- Uses new NVIDIA Container Runtime

GPU Enhancements

- ✓ GPU Health Checks and Monitoring
- ✓ Heterogeneous GPU Support
- ✓ Support for MPS
- ✓ GPU Topology aware
- Device Plugin support for GPU lifecycle management
- ✓ Representing resources in the cluster

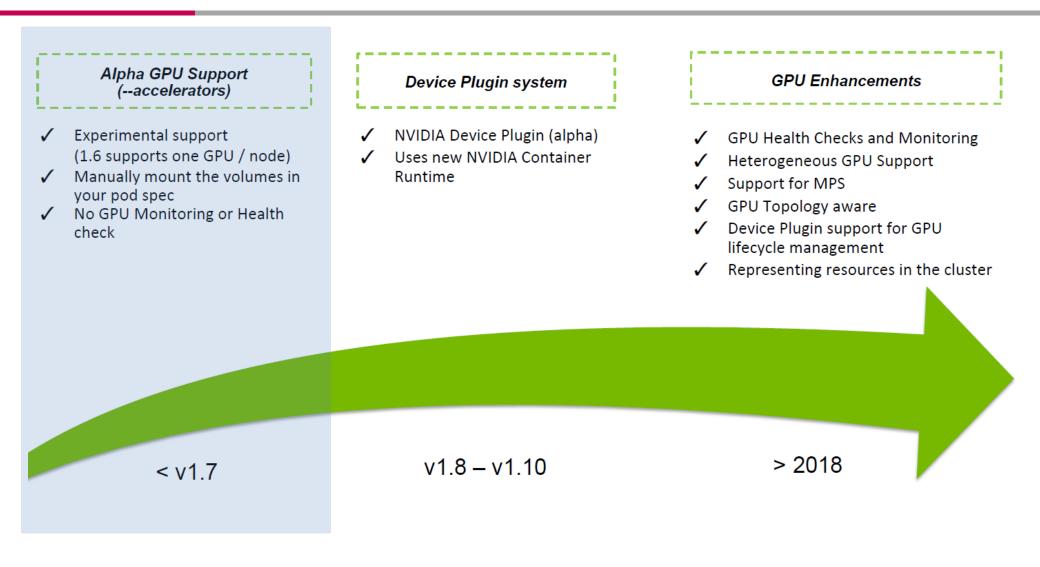


v1.8 - v1.10

> 2018

출처 : The path to GPU as a Service in Kubernetes , Renaud Gaubert, 2018

kubernetes roadmap for GPU



출처 : The path to GPU as a Service in Kubernetes , Renaud Gaubert, 2018

GPU Accelerator

● 제공기능

- GPU를 schedule 가능한 자원으로 등록
- kernel / user-level module을
 volume으로 mount

● 단점

- 1 노드당 1개 GPU만 할당 가능 (v1.6)
- Host 의존성 발생
- GPU health check 및 모니터링 기능 없음

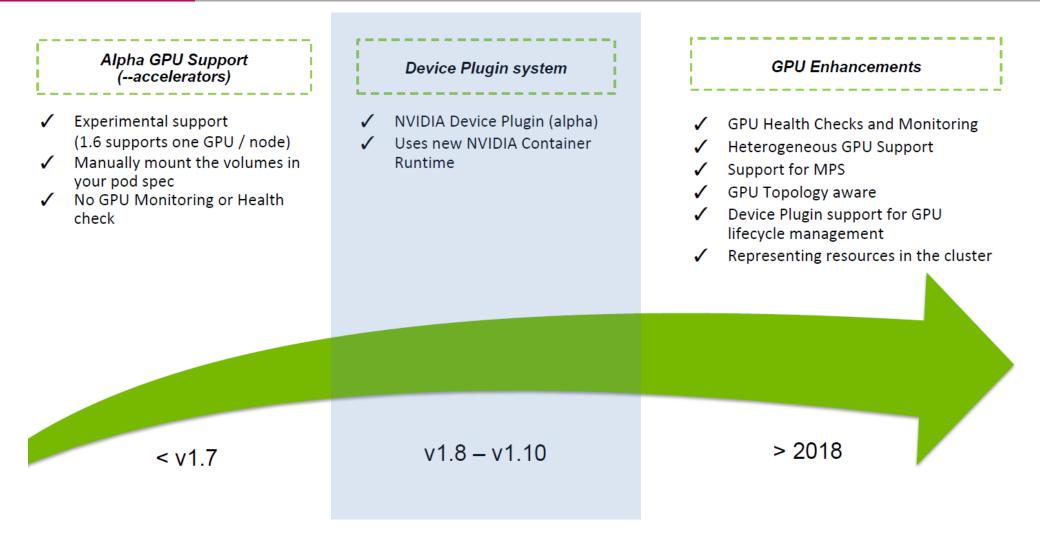
spec: volumes: - name: nvidia-libraries hostPath: path: /usr/lib/nvidia-390 containers: - name: accelerator-sample image: nvidia/cuda:latest resources: limits: alpha.kubernetes.io/nvidia-gpu: 1

volumeMounts:

- name: nvidia-libraries

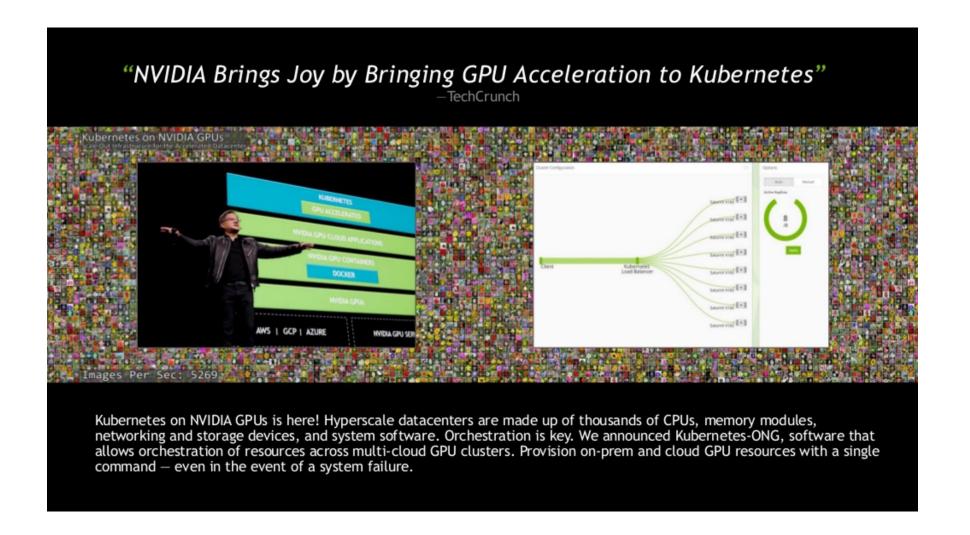
mountPath: /usr/local/nvidia/lib64

kubernetes roadmap for GPU



출처 : The path to GPU as a Service in Kubernetes , Renaud Gaubert, 2018

kubernetes on NVidia GPUs



출처 : GTC 2018 : a new AI era dawns , NVidia, 2018

Device Plugin

NVidia Device Plugin

• PodSpec의 Host 의존성을 제거

spec:

containers:

name: TENSORFLOW image: nvidia/cuda:latest resources: limits:

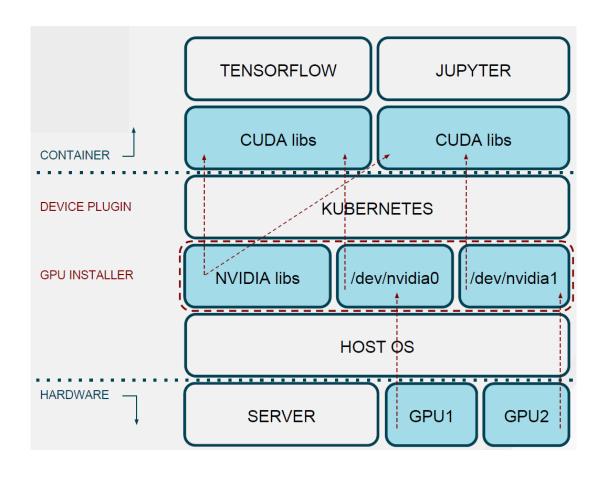
nvidia.com/gpu: 1

spec:

containers:

name: JUPYTER
 image: nvidia/cuda:latest
 resources:
 limits:

nvidia.com/gpu: 1

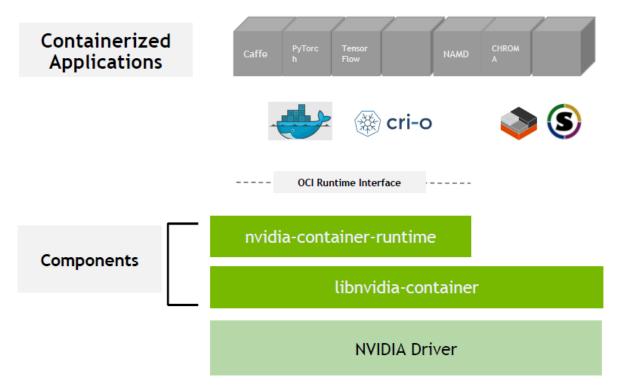


출처 : Automating GPU Infrastructure for kubernetes & container linux , Lucas Serven Marin, 2018

Container Runtime

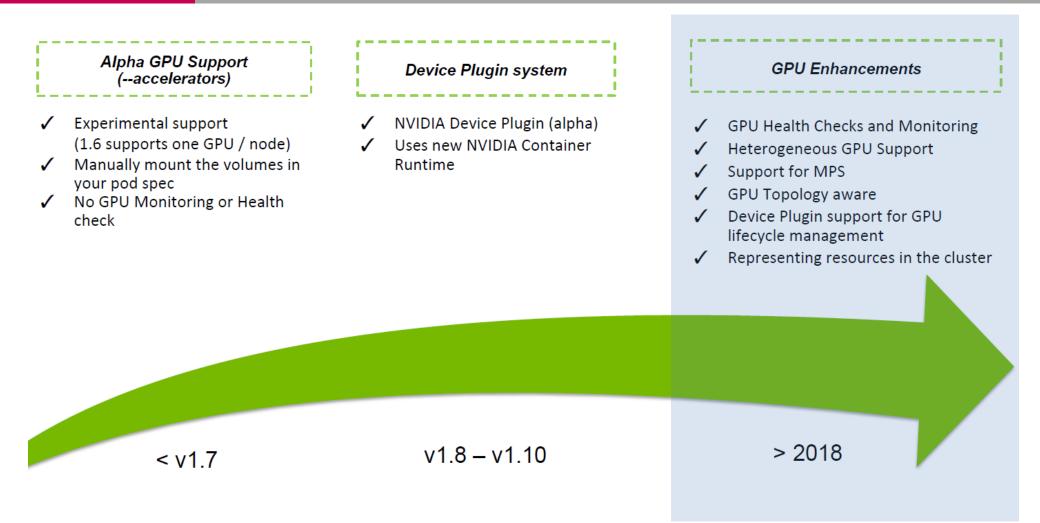
NVidia Container Runtime

- docker / cri-o / LXD / Singularity 와 같은 다양한 Container 에서 GPU 기능 사용 가능
- OCI Runtime Interface 사용



출처 : The path to GPU as a Service in Kubernetes , Renaud Gaubert, 2018

kubernetes roadmap for GPU



출처 : The path to GPU as a Service in Kubernetes , Renaud Gaubert, 2018

LG CNS에서 kubernetes GPU

Data Analytics & AI Platform

- Cloud상에서 Data 수집, 저장, 처리 및 분석 환경을 제공
- 기존 빅데이터 플랫폼에 AI를 추가하려는 요구 발생

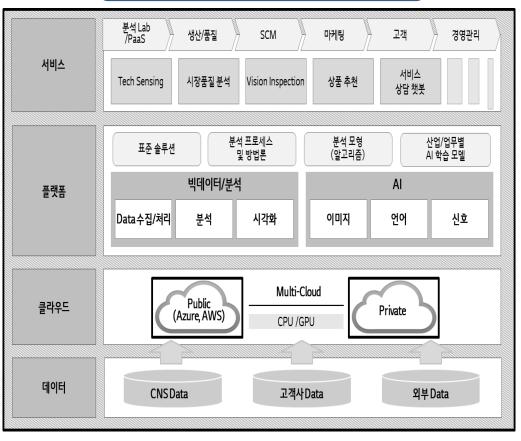
AI 기반으로 Kubernetes 적용

- Container Orchestration 제공 기능이 많아, 기존 Appliance 기반 구성보다 공수절감 예상
- kubernetes는 시장 점유율이 높고, 적용 사례도 많음
 Cloudera Data Science Workbench
- Cloud Big 3에 모두 서비스가 존재하여, 향후 멀티클라우드로 전환도 용이

● GPU 사용 검증 필요

• GPU 기능은 당시 alpha stage





kubernetes GPU PoC

● 목적

- alpha stage인 GPU 기능이 서비스에 문제가 없는지 검증
- NGC를 통해 로컬에서 수행하는 AI어플리케이션과 동일한 수준의 성능을 보장하는지 검증

● 기간

- 1차 테스트 (3/12~3/30) : 기본 구성 및 기능위주 테스트
- 2차 테스트 (4/16~4/27) : 장비 추가 및 비기능 위주 테스트

● 테스트 케이스

- 기능: unit test 케이스 중 GPU관련 필요검증항목 추출 (46/901) https://github.com/kubernetes/kubernetes/blob/release-1.9/test/test_owners.csv
- 안정성: GPU 주요 컴포넌트의 장애 테스트
- 어플리케이션 : 공개되어있는 benchmark 프로그램을 이용 (Tensorflow / Caffee...)

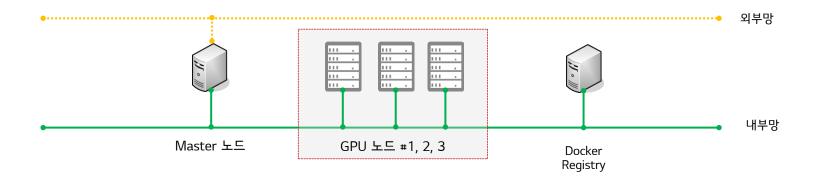
환경구성

● H/W 구성

- GPU 上드 #1 NVidia Tesla V100 * 8 /w NVLink 2.0 (DGX-1V)
- ・ GPU <u>노</u>드 #2 NVidia Tesla P100 * 2
- GPU 上三 #3 NVidia Tesla P4 * 4

● S/W 구성

- kubernetes v1.9.3
- nvidia-docker v2.0
- docker-ce v17.03.2
- nvidia driver v390.30
- Ubuntu 16.04



kubernetes GPU 설치 - v1.9

● device plugins 설정

• kubelet : --feature-gates="DevicePlugins=true"

● container-runtime 설치

- nvidia driver 설치 (>= v.361.93)
- nvidia-docker 2.0 설치 (docker 버전에 맞춤)
- daemon.json : default runtime 변경

● device plugin 설치

- device plugin 설치 (DaemonSet)
- node labeling (accelerator=[value])

docker image sample

Dockerfile

FROM nvcr.io/nvidia/tensorflow:18.02-py3

RUN mkdir -p /data/cifar10_estimator

ADD cifar10_estimator/* /data/cifar10_estimator/

WORKDIR /data/cifar10_estimator/

RUN python generate_cifar10_tfrecords.py --data-dir=\${PWD}/cifar-10-data

CMD exec /bin/bash -c "trap : TERM INT; sleep infinity & wait"

NVidia Container Registry

- NVidia에 의해 최적화된 검증된 Deep Learning Framework 이미지 제공
- NVidia GPU가 적용된 시스템이면 사용 가능

pod sample

cifar10-test.yaml

```
apiVersion: v1
kind: Pod
metadata:
 name: cifar10-test
spec:
 containers:
 - name: cifar10-test
  image: myregistrydomain.com:5000/cifar10-test
   resources:
    limits:
     nvidia.com/gpu: 2
   command: ["/bin/bash"]
  args: ["-c","python cifar10_main.py --data-dir=${PWD}/cifar-10-data --job-dir=/tmp/cifar10 --num-gpus=2
--train-steps=100000"]
 nodeSelector:
   accelerator: nvidia-tesla-p4
 imagePullSecrets:
 - name: myregistrydomain
 restartPolicy: OnFailure
```

docker options

NOTE: The SHMEM allocation limit is set to the default of 64MB. This may be insufficient for TensorFlow. NVIDIA recommends the use of the following flags: nvidia-docker run --shm-size=1g --ulimit memlock=-1 --ulimit stack=67108864 ...

--shm-size

- container 내부 IPC 통신시 사용할 Share Memory 크기
- 기본값: 64mb

--ulimit memlock

- 메모리 주소공간 최대크기
- 기본값: 64kb
- -1이면 swap을 사용하지 않음

--ulimit stack

- 스택 크기
- 기본값: 8192kb

docker options

kubernetes

```
433.5
  250
                   9.369 0.10000
  251
            432.8
                  9.367 0.10000
  252
            433.9 9.365 0.10000
  253
            434.6
                  9.364 0.10000
  254
            434.4 9.364 0.10000
  255
            433.8 9.367 0.10000
  256
            434.5 9.365 0.10000
...........
Images/sec: 433.8 +/- 0.0 (jitter = 0.7)
```

ri Apr 6 11:07:09 2018								
NVIDIA-SMI 390.30 Driver Version: 390.30								
PU an	Name Temp	Perf						
Θ /A			55W /				•	0 Default
1 /A	Tesla 67C							0 Default
2 /A								0 Default
3 /A							 100%	0 Default
roce GPU		PID	Туре	Process	name			GPU Memory Usage
0 1 2 3	43 43	3945 3945	C C C	python python python python				7455MiB 7455MiB 7455MiB 7455MiB 7455MiB
	April	Apr 6 17 VIDIA-SMI VIDIA-SMI PU Name an Temp 0 Tesla /A 63C 1 Tesla /A 67C 2 Tesla /A 62C 3 Tesla /A 73C	Apr 6 11:07:0 VIDIA-SMI 390.3 PU Name an Temp Perf 0 Tesla P4 /A 63C P0 1 Tesla P4 /A 67C P0 2 Tesla P4 /A 62C P0 3 Tesla P4 /A 73C P0	Apr 6 11:07:09 2018 VIDIA-SMI 390.30 PU Name Persist an Temp Perf Pwr:Usa 0 Tesla P4 /A 63C P0 55W / 1 Tesla P4 /A 67C P0 53W / 2 Tesla P4 /A 62C P0 54W / 3 Tesla P4 /A 73C P0 44W / rocesses: GPU PID Type 0 43945 C 1 43945 C	Apr 6 11:07:09 2018 VIDIA-SMI 390.30 PU Name	Apr 6 11:07:09 2018 VIDIA-SMI 390.30	Apr 6 11:07:09 2018 VIDIA-SMI 390.30	Apr 6 11:07:09 2018 VIDIA-SMI 390.30

docker options

nvidia-docker /w options

• python nvcnn.py -m resnet50 -fp16 -g 4 -num_batch=256

250 251 252 253 254 255 256	1	432.3 433.0 433.8 434.1 433.2	9.387 0.10000 9.384 0.10000 9.383 0.10000 9.382 0.10000 9.382 0.10000 9.384 0.10000 9.383 0.10000
			0.1 (jitter = 0.8)

Every 2.0s: nvidia-smi							
Fri Ap	r 6 1	9:57:2	9 2018				
NVID	IA-SMI	390.3	Θ		Driver Version: 39	0.30	
GPU Fan	Name Temp				Bus-Id Disp.A Memory-Usage		
Θ N/A			56W		00000000:04:00.0 Off 7465MiB / 7611MiB		0 Default
1 N/A			54W		00000000:05:00.0 Off 7465MiB / 7611MiB		0 Default
2 N/A		P4 P0	57W		00000000:06:00.0 Off 7465MiB / 7611MiB		0 Default
3 N/A			54W		00000000:82:00.0 Off 7465MiB / 7611MiB		0 Default
Proc GPU	esses:	PID	Туре	Process	name		GPU Memory Usage
===== 0 1 2	4	===== 1477 1477 1477	C	python python python			7455MiB 7455MiB 7455MiB 7455MiB
3	4	1477 	C	python			7455MiB

기능검증

구분	Test Case	Test Case 설명		점검 결과
	agentain on All Ad /Ull T	AI어플리케이션 이미지(tensorflow, caffe)를 GPU노드에 배포	Single	정상
	container 생성/배포	- 명령어 : kubectl create -f "yaml_file".yaml	Multi	정상
		container 로그를 출력하여 이상 유무 확인	Single	정상
7 7.1	container 로그 확인	- 명령어 : kubectl logs "pod_name"	Multi	정상
주기능		container 접속 후 정상 배포 여부 확인	Single	정상
	container 접속	- 명령어 : kubectl exec –it "pod_name"	Multi	정상
		container 삭제	Single	정상
	container 삭제	- 명령어 : kubectl delete –f "yaml_file".yaml	Multi	정상
	Master 노드 장애	Master 노드 셧다운 시 작업 수행중인 GPU container의 영향도 확인		정상
가용성	kubelet 서비스 장애	GPU 노드의 kubelet 서비스 장애시 작업 수행중인 GPU container의 영향도 확인		정상
	device plugin 장애	GPU 노드의 device plugin 프로세스 장애시 작업 수행중인 GPU container의 영향도 확인		정상
CPU/GPU CPU container 배포		GPU container가 배포된 동일 노드에 CPU container 배포 및 서비스 점검구동 - 서비스 : 데모 어플리케이션의 웹페이지 정상 접속 확인		정상

Tensorflow benchmark

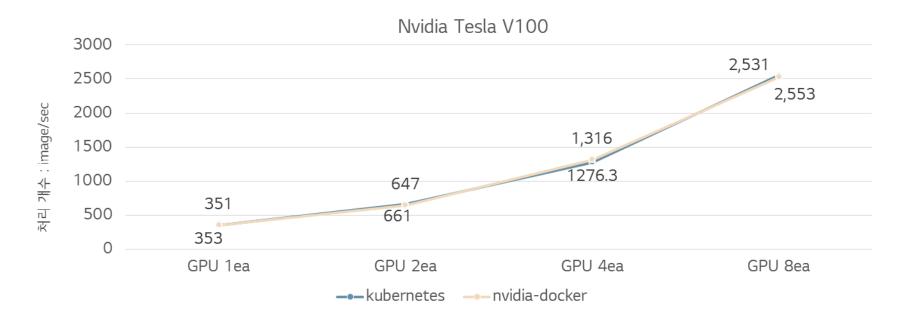
Program Spec

• Program: Tensorflow CNN benchmark

• Model: resnet50

• Dataset: ImageNet (2.7GB)

• Batch size: 254



Tensorflow benchmark

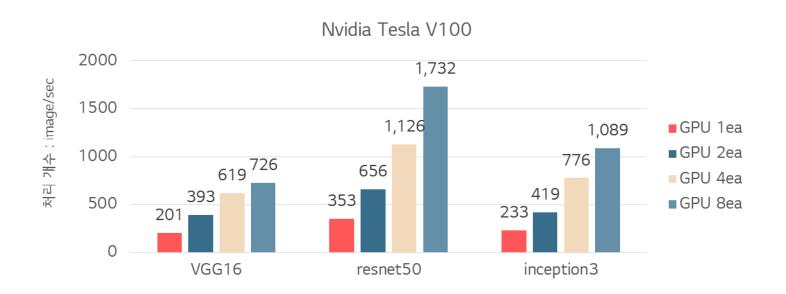
Program Spec

• Program: Tensorflow CNN benchmark

• Model: VGG16 / resnet50 / inception3

• DataSet: ImageNet (2.7GB)

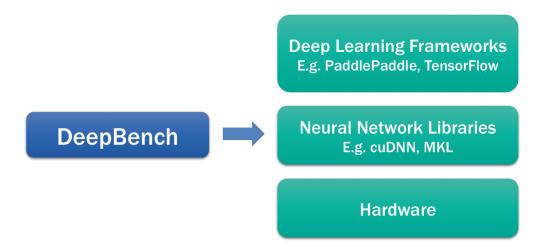
• Batch size: 64



Baidu DeepBench

DeepBench

- Deep Learning 운영하는데 어떤 하드웨어가 좋은 성능을 제공하는지 답변을 위해 작성
- Tensorflow와 같은 top level 프레임워크가 아닌 low level library를 이용하여 검증
- Baidu Research에서 자체 수행한 결과 데이터를 공유



출처: https://github.com/baidu-research/DeepBench

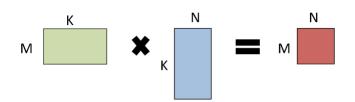
DeepBench benchmark

Program Spec

• Program: GEMM

• Model: Dense Matrix Multiplication

• NVidia P100로 비교



kubernetes							
М	N	K	Time(usec)				
7680	48000	2560	207228				
6144	48000	2048	132857				
4608	48000	1536	75045				
8448	48000	2816	250733				
3072	48000	1024	33758				

baidu			
М	N	K	Time(usec)
7680	48000	2560	206884
6144	48000	2048	135716
4608	48000	1536	74962
8448	48000	2816	250482
3072	48000	1024	33614

NVidia Tesla V100

● 백서

- TFLOPS: 초당 부동소수점 연산 횟수
- TFLOPS 기준 V100은 P100 대비
 약 1.5배 향상된 연산성능 보유

Tesla Product	Tesla P100	Tesla V100
GPU	GP100 (Pascal)	GV100 (Volta)
SMs	56	80
TPCs	28	40
FP32 Cores / SM	64	64
FP32 Cores / GPU	3584	5120
FP64 Cores / SM	32	32
FP64 Cores / GPU	1792	2560
Tensor Cores / SM	NA	8
Tensor Cores / GPU	NA	640
GPU Boost Clock	1480 MHz	1530 MHz
Peak FP32 TFLOPS ¹	10.6	15.7
Peak FP64 TFLOPS ¹	5.3	7.8
Peak Tensor TFLOPS ¹	NA	125
Texture Units	224	320
Memory Interface	4096-bit HBM2	4096-bit HBM2
Memory Size	16 GB	16 GB
L2 Cache Size	4096 KB	6144 KB
Shared Memory Size / SM	64 KB	Configurable up to 96 KB
Register File Size / SM	256 KB	256KB
Register File Size / GPU	14336 KB	20480 KB
TDP	300 Watts	300 Watts
Transistors	15.3 billion	21.1 billion
GPU Die Size	610 mm²	815 mm²
Manufacturing Process	16 nm FinFET+	12 nm FFN

¹ Peak TFLOPS rates are based on GPU Boost Clock

V100 vs P100

Program Spec

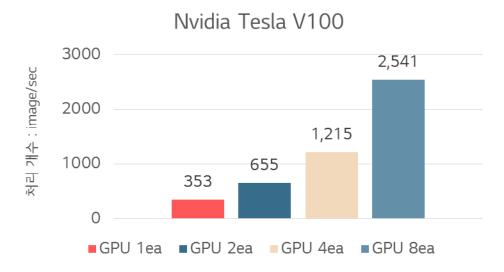
• Program: NVidia CNN benchmark

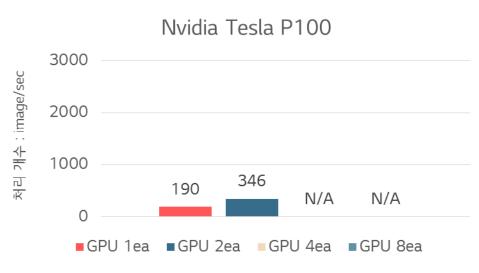
• F/W: Tensorflow

• Dataset: ImageNet (2.7GB)

• Batch size: 64

	P100	V100	
GPU 1ea	190	353	1.86x
GPU 2ea	346	655	1.89x





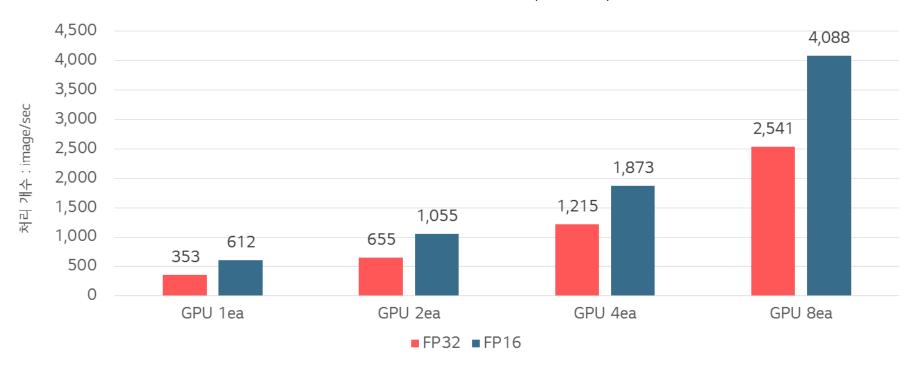
V100 신규 기능

tensor core

- 16비트 부동소수점 정밀도를 사용시 이용가능
- TFLOP 125로 매우 빠른 연산 가능

Program Spec

- program: NVidia CNN benchmark
- F/W: Tensorflow
- Dataset: ImageNet (2.7GB)
- Batch size: 64
- Option: --fp16



EOD

