DEVIEW 2020

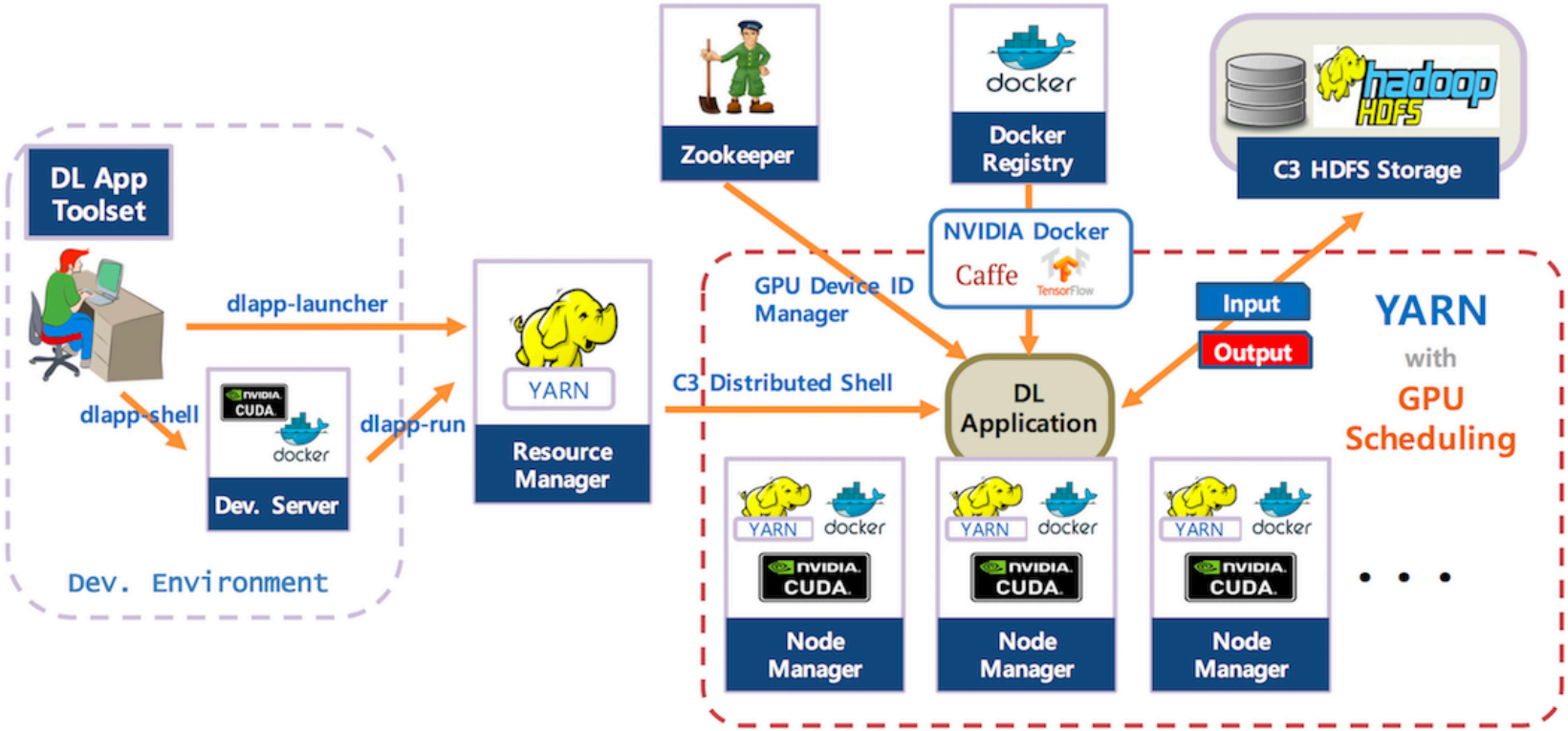
# 손쉽게 ML 라이프사이클을 다룰 수 있는 MLOps

유승현 AI Suite

# 하고 있는 일

DEVIEW 2020

## C3 Deep Learning Cluster



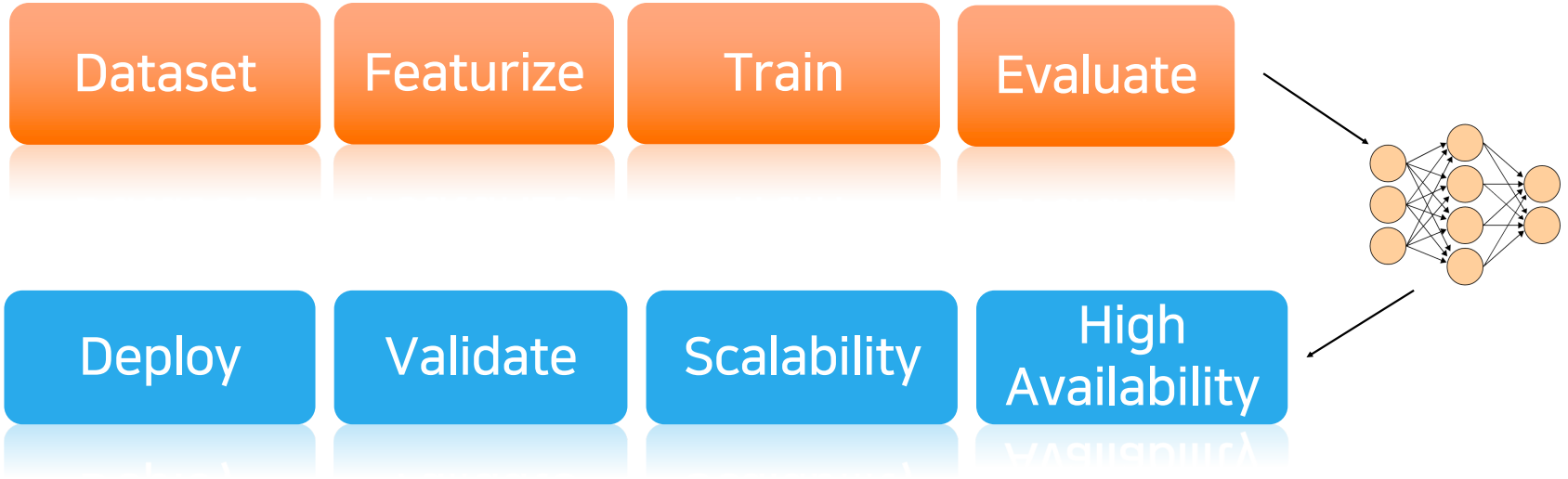
CONTENTS

DEVIEW 2020

# 1. 딥러닝 서빙의 어려운점

# 1.1 AI 연구자와 엔지니어 영역이 다름





# 1.2 자원 낭비

DEVIEW 2020

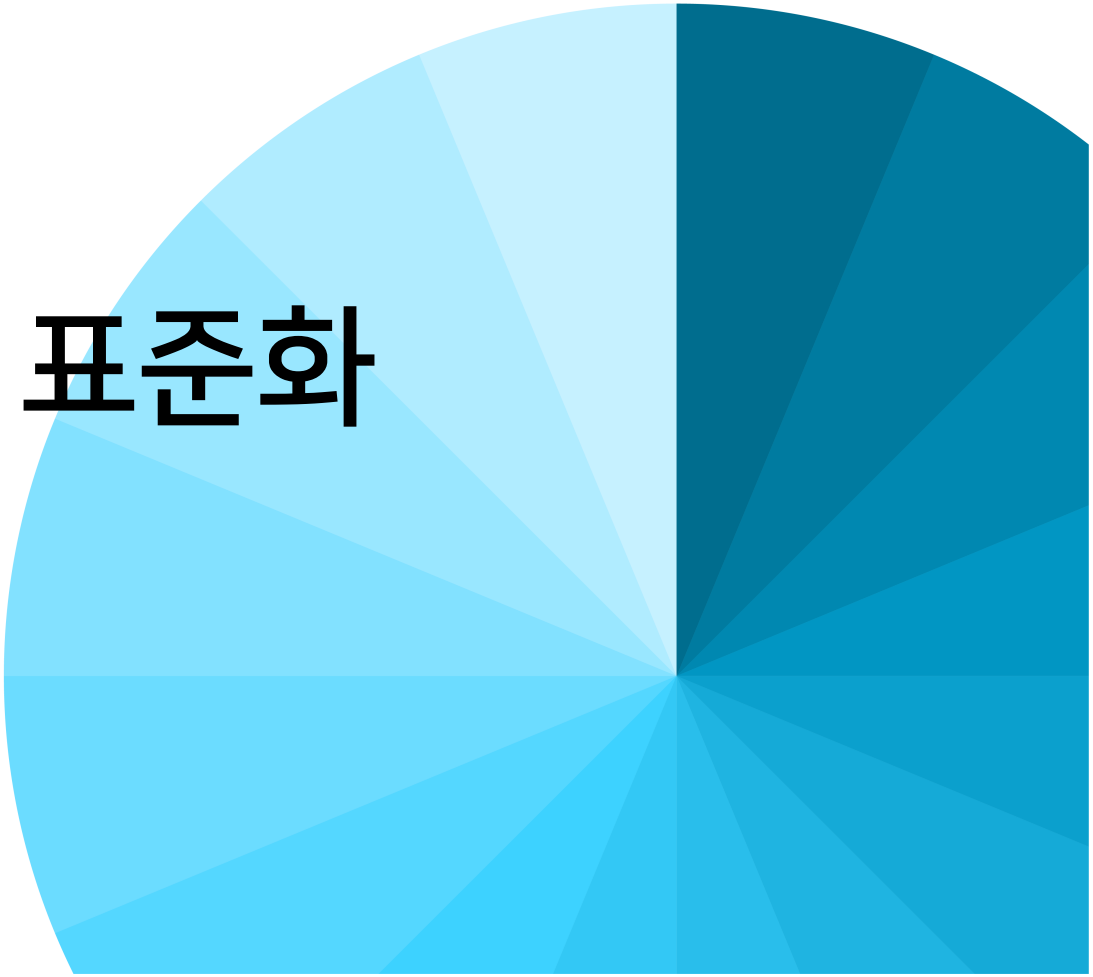
얼마나 트래픽이 들어올지 몰라서 많은 GPU 확보하고 시작

- “최대 트래픽을 계산해보니 GPU 50개가 필요해요”

인퍼런스를 하기위한 최소한 장비 사양을 모름

DEVIEW 2020

# 2.모델 표준화



# 2.1 모델 표준화의 장점

DEVIEW 2020

## AI 연구자와 엔지니어링 하는 사람 분리 가능

- AI 연구자는 모델만 전달
- 엔지니어는 모델을 받아서 서빙

# 2.2 바로 서빙 가능한 모델 형식

DEVIEW 2020

SavedModel
- Tensorflow 에서 사용, Tensorflow Serving 에서 실행가능

ONNX
- Pytorch로 사용해서 학습한 모델에서 사용 가능

TensorRT plan file
- NVIDIA의 TensorRT 기술을 사용해서 GPU 인퍼런스에 최적화
- Trition Inference Server 를 사용해서 서빙 가능

# 2.3 MLflow Models

DEVIEW 2020

An MLflow Model is a standard format for packaging machine learning models that can be used in a variety of downstream tools

https://www.mlflow.org/docs/latest/models.html#

# 2.4 어떤 형식의 모델 파일을 지원할까

DEVIEW 2020

## Savedmodel, ONNX, TensorRT 만 지원

- 네이버에서는 대부분 tensorflow, pytorch 를 사용
- MLflow models를 사용하기 위해서는 추가 코드 필요함


# 3. Model Registry

# 3.1 Model Registry

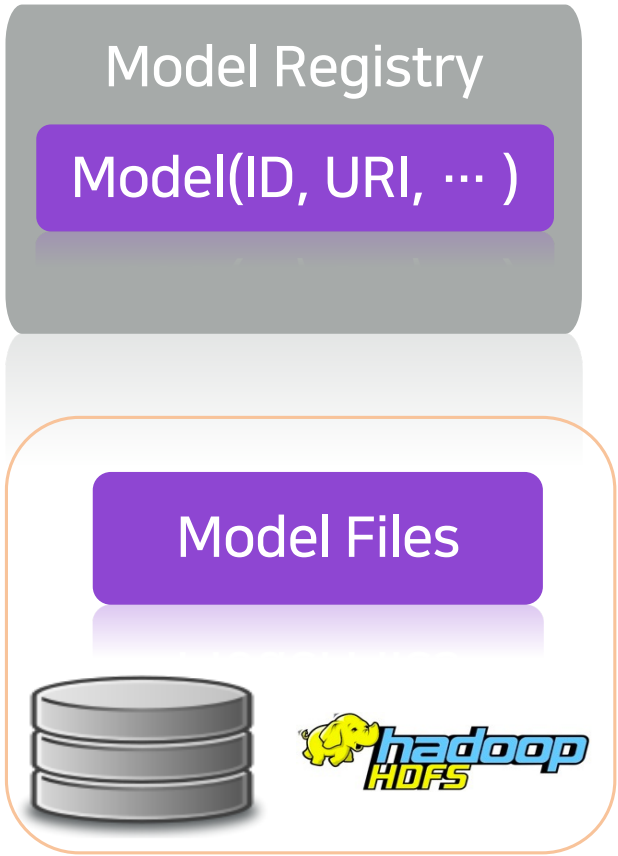


모델과 부가 정보를 저장

- model-id, model-URI, description, user, metrics, hyper-parameters…

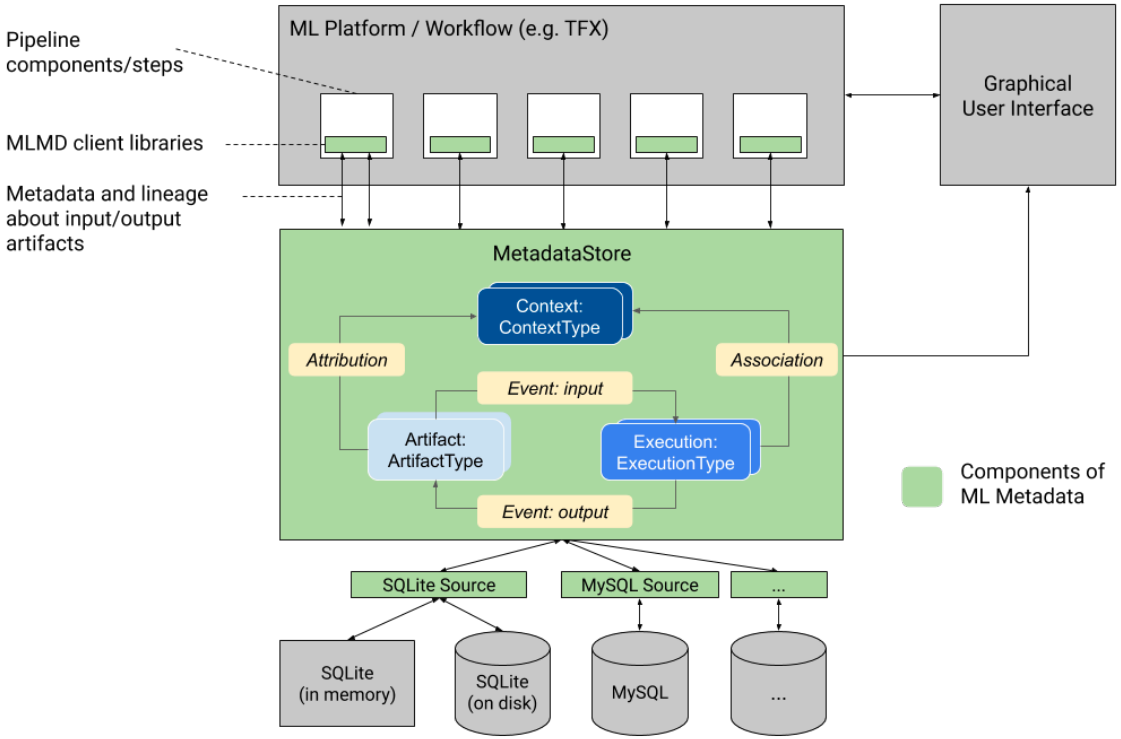
실제 모델 파일은 HDFS 에 저장

AI 연구자와 엔지니어를 이어주는 도구



# 3.2 ML-metadata 를 사용

DEVIEW 2020

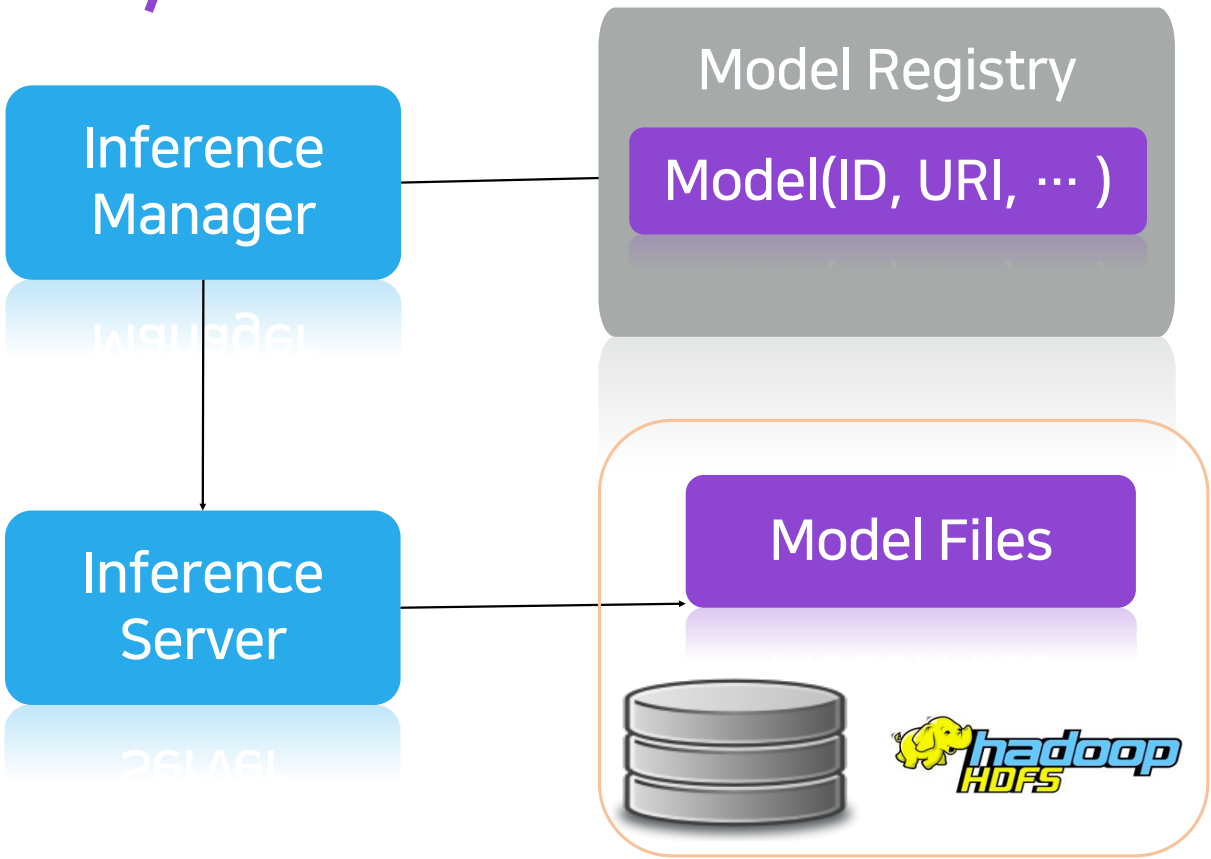



https://www.tensorflow.org/tfx/guide/mlmd

# 3.3 Model registry

DEVIEW 2020

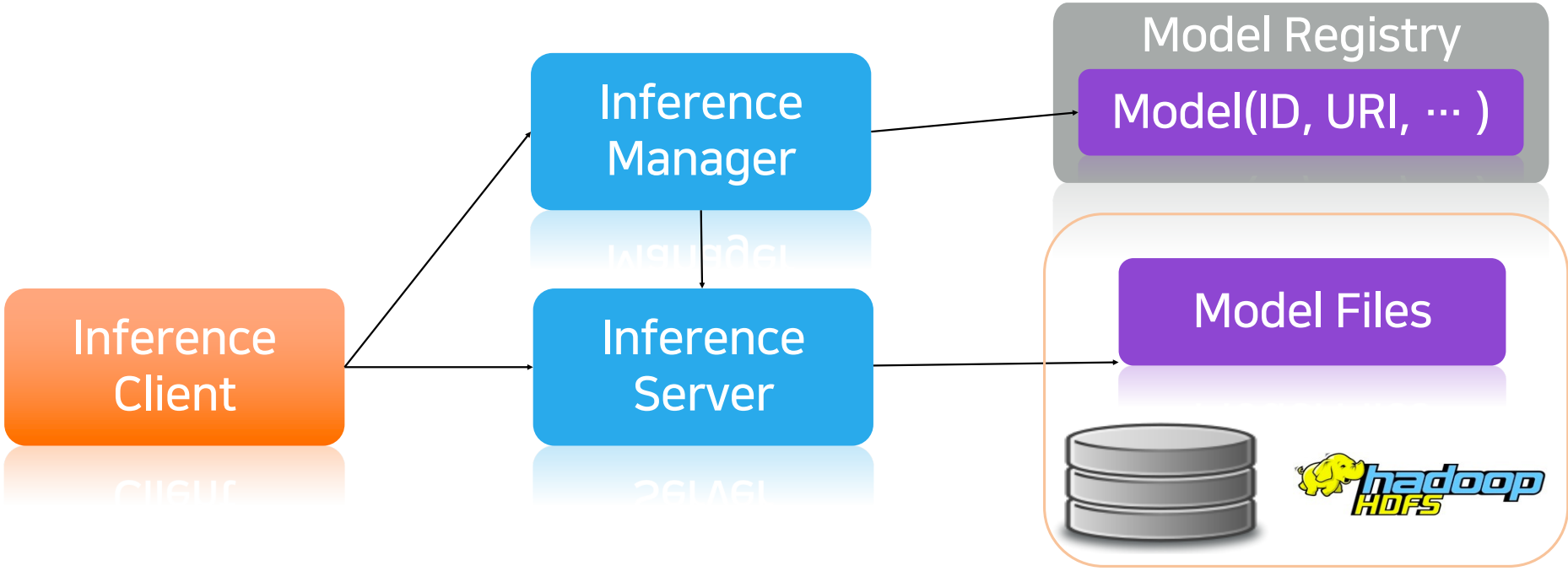
서빙 서버 실행



# 3.4 Inference client



output = inference(model-id, layer_name=input)



# 3.4 Inference client

DEVIEW 2020

inference(model-id, layer_name=input)

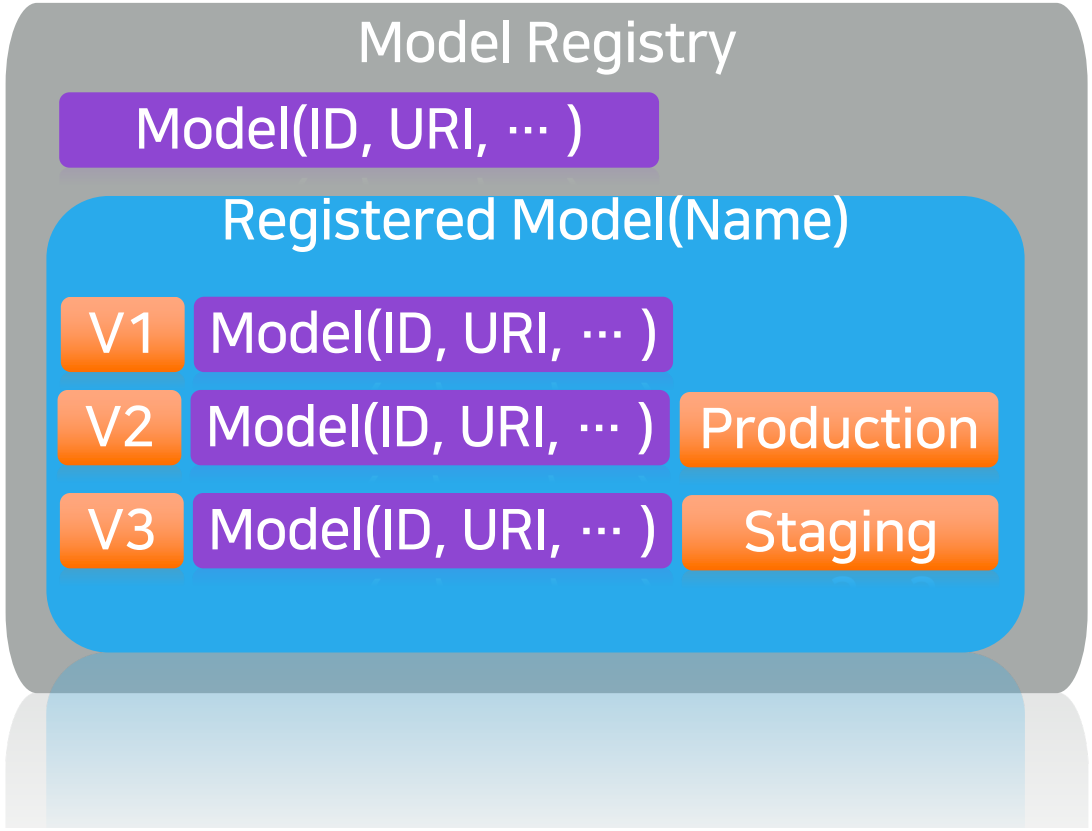
- Model-id 를 가지고 Inference Server가 어디에 실행되어 있는지를 찾기
- Load balance
- 에러 처리

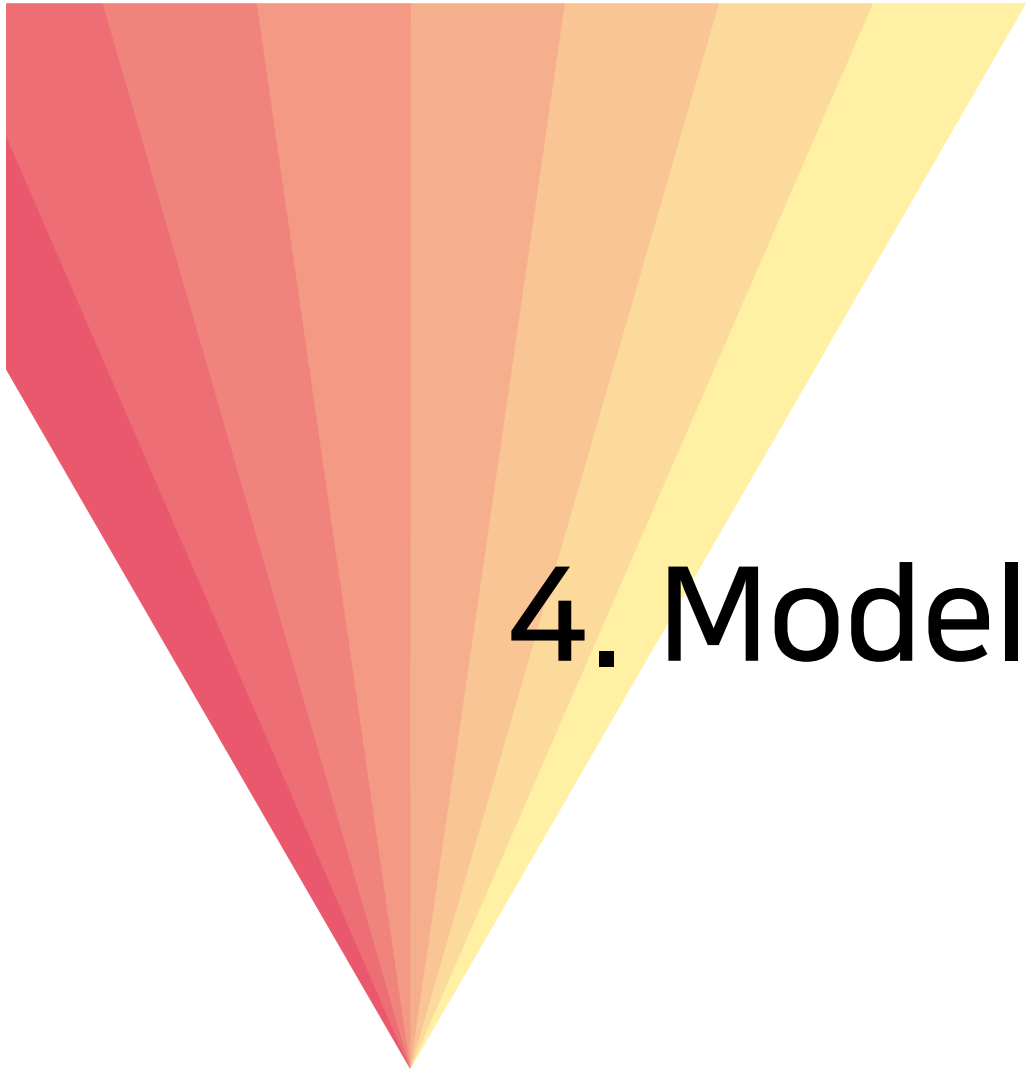
# 3.5 Registered Model

DEVIEW 2020

output = inference(mode-name, layer_name=input)

Model Name
Version 관리
배포 관리
 Staging, Production





DEVIEW 2020

# 4. Model Validation

# 4.1 validation 목적



모델을 변경하였을때 인퍼런스 결과가 이상하게 나오는것 방지

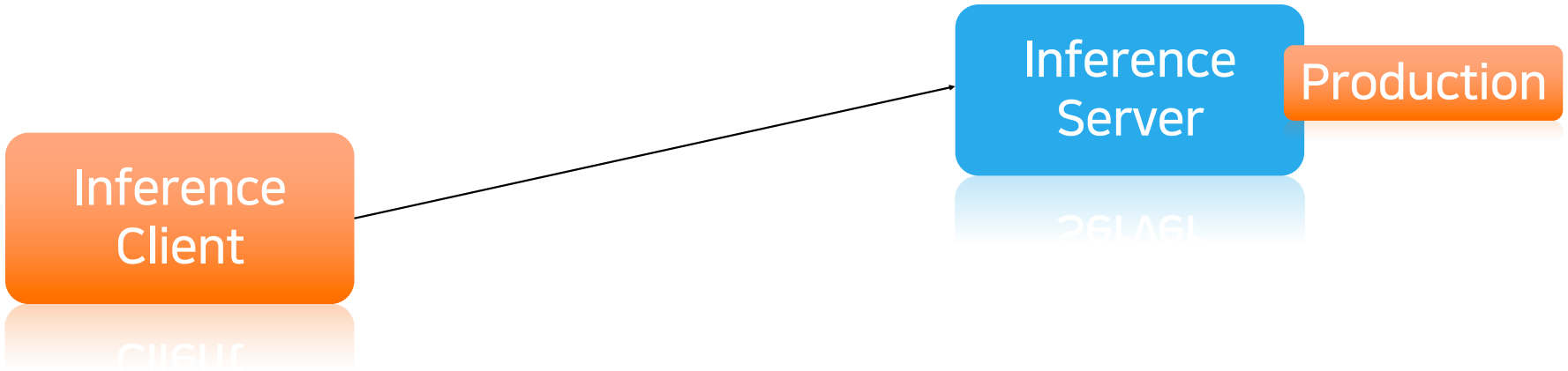
# 4.2 validation 구현 정책

DEVIEW 2020

AI 연구자나 엔지니어의 개입없이 해주자
Production 와 staging 를 비교하기

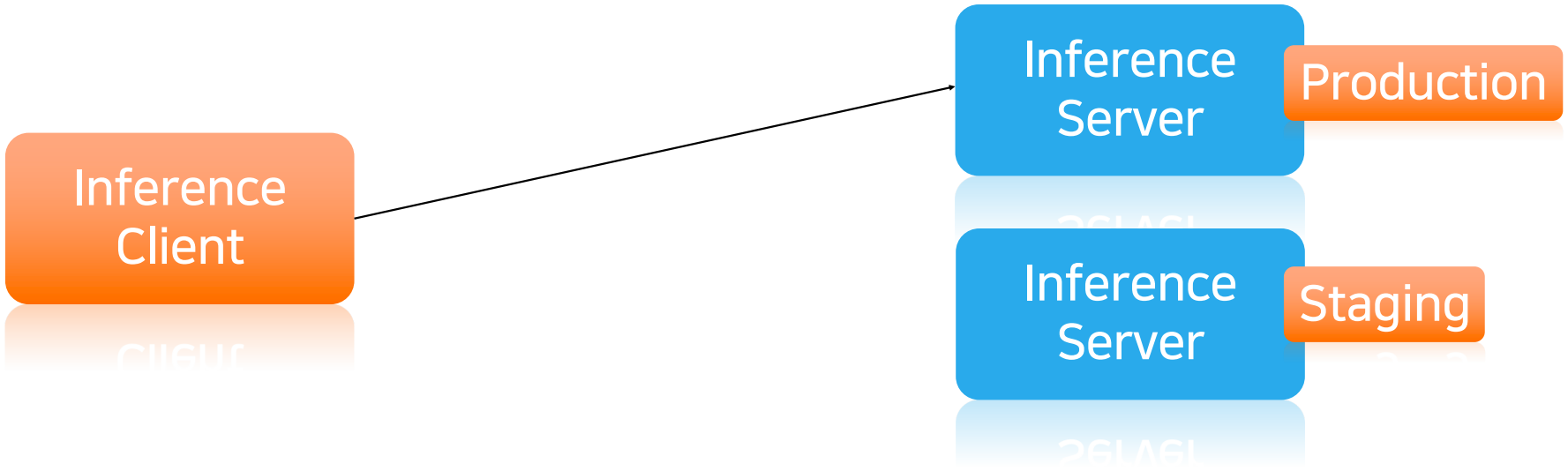
# 4.3 validation 구성





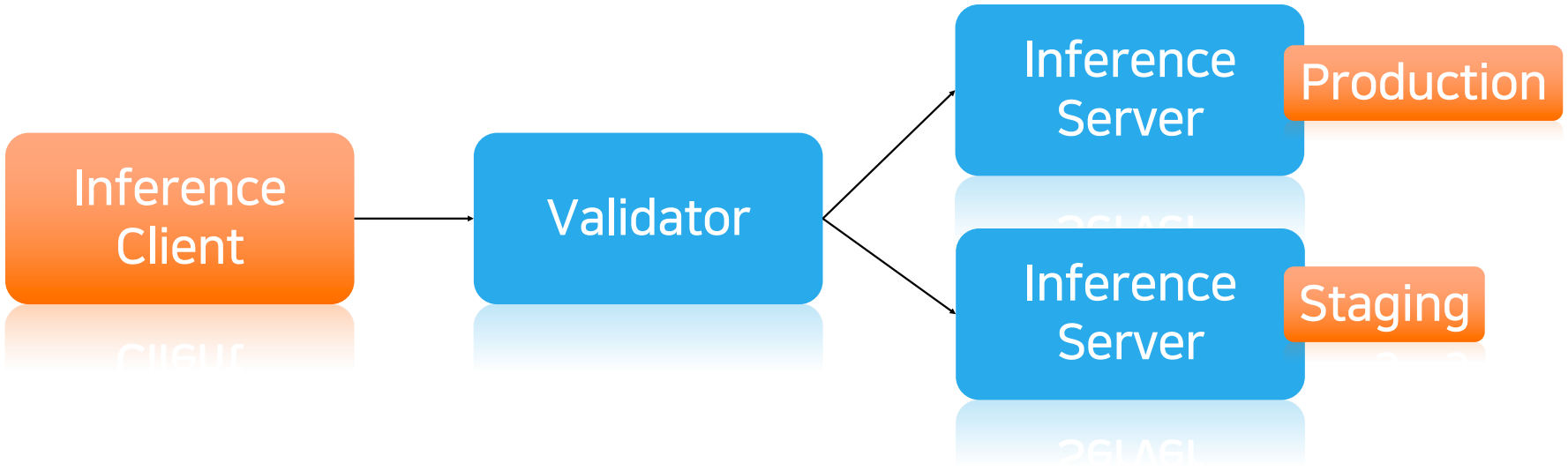
# 4.3 validation 구성

DEVIEW 2020



# 4.3 validation 구성





# 5. 자원 최적화

# 5.1 자원 낭비하는 이유

DEVIEW 2020

## 인퍼런스를 하기위한 최소한 장비 사양을 모름

- 트레이닝 할때 사용했던 사양을 그대로 인퍼런스에 사용
- 다양한 GPU 모델
- CPU 에서도 테스트 필요

# 5.2 성능 테스트 방법

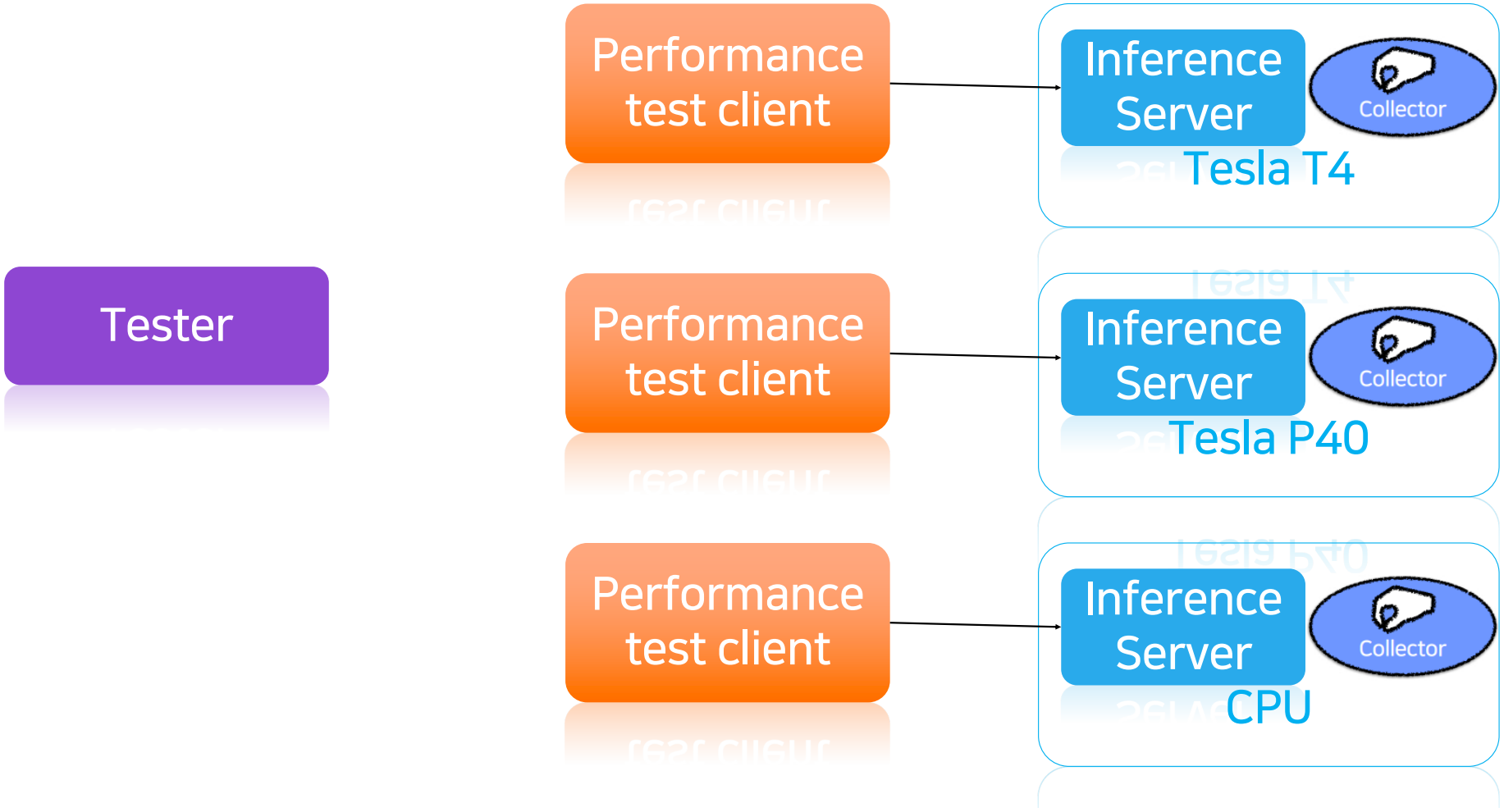
DEVIEW 2020

인퍼런스 서버를 실행하고, 입력 하나가 있다면, 이후 성능 테스트는 기계적인 과정

- 인퍼런스 서버 실행하고,
- CPU, GPU, Memory 등 시스템 자원 모니터링
- Client를 실행해서 부하를 서서히 증가시켜주기

# 5.2 성능 테스트 방법




Tester
Performance test client
Performance test client
Performance test client
Inference Server
Collector
Tesla T4
Inference Server
Collector
Tesla P40
Inference Server
Collector
CPU

# 5.3 MPS(Multi Process Service)

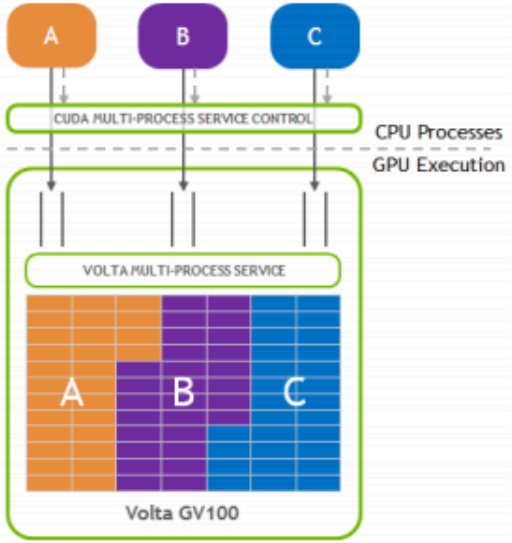
DEVIEW 2020

## MPS

- 한개의 GPU를 여러 프로세스가 사용하면서 성능 저하를 최소로 해주는 기술
- GPU 코어를 ½, ¼ 만 할당하는게 가능
- 메모리 사용은 제한할수 없음

## 성능

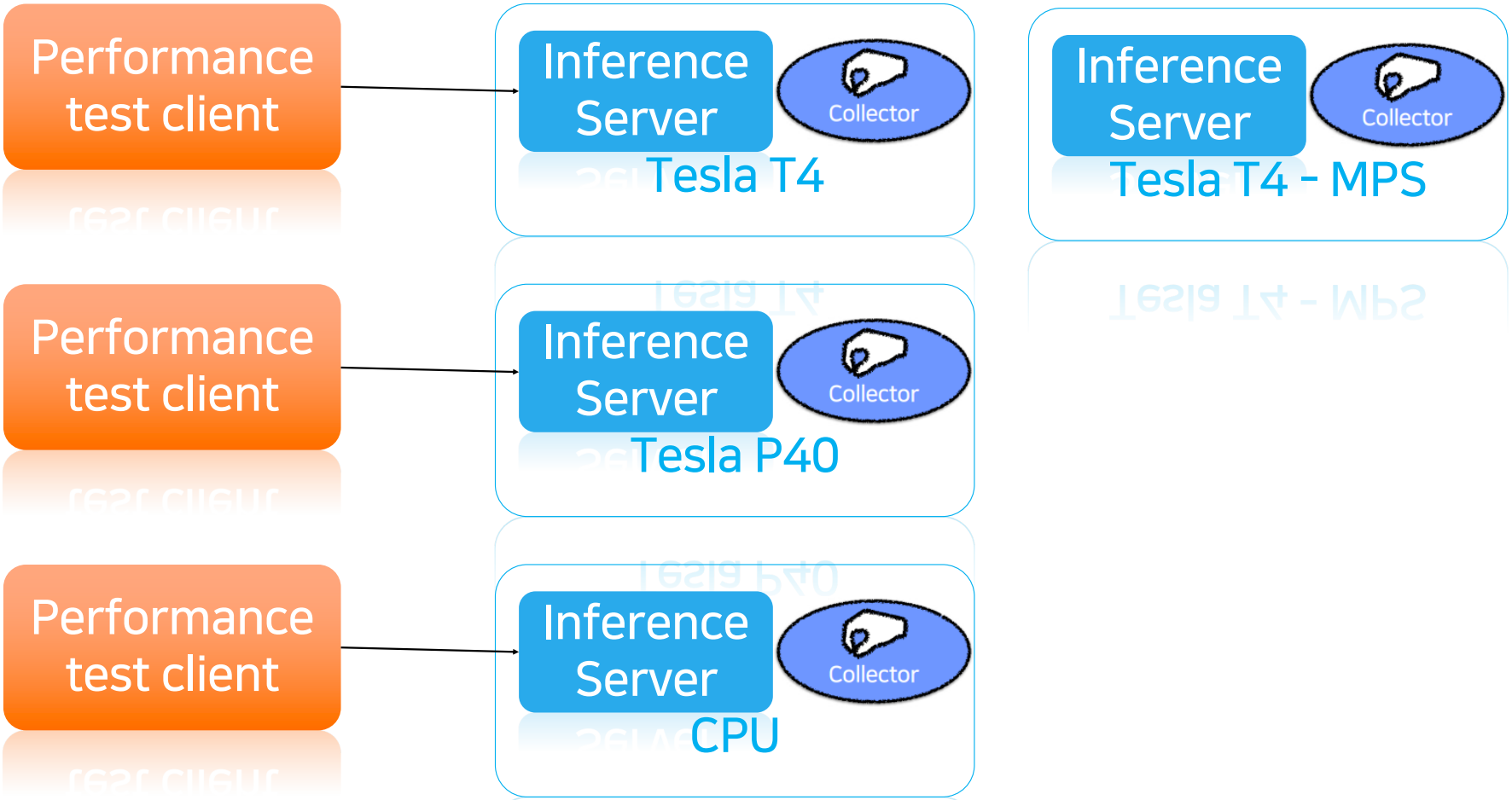
- GPU를 사용율이 낮은 작업은 성능 저하가 거의 없었음




https://docs.nvidia.com/deploy/pdf/CUDA_Multi_Process_Service_Overview.pdf

# 5.3 성능 테스트 – MPS 추가





# 5.4 CPU 성능 테스트



CPU도 다양한 모델이 있음

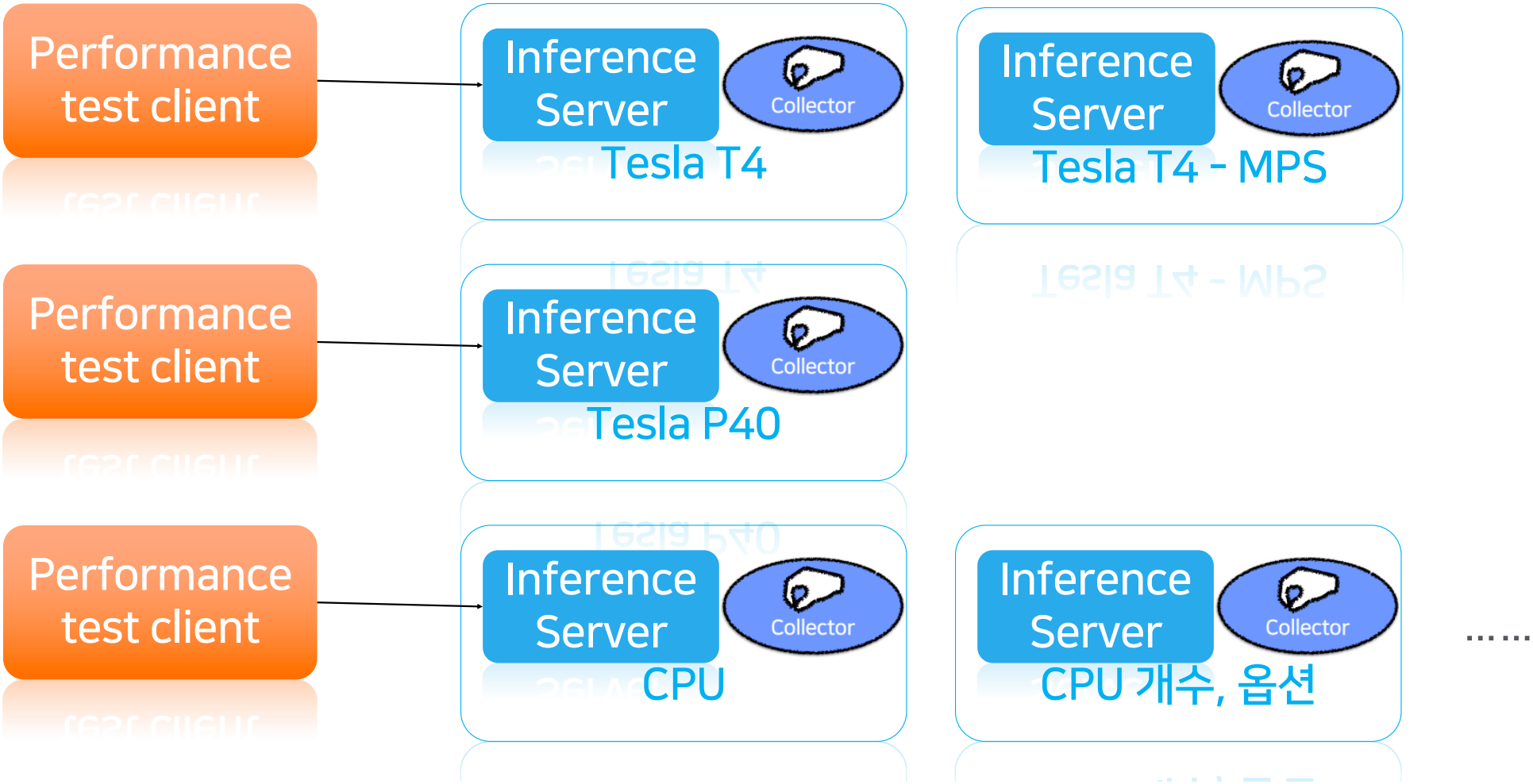
CPU 1개, 2개 …

환경 변수에 따라 성능이 달라짐

- OMP_NUM_THREADS

# 5.4 CPU 성능 테스트

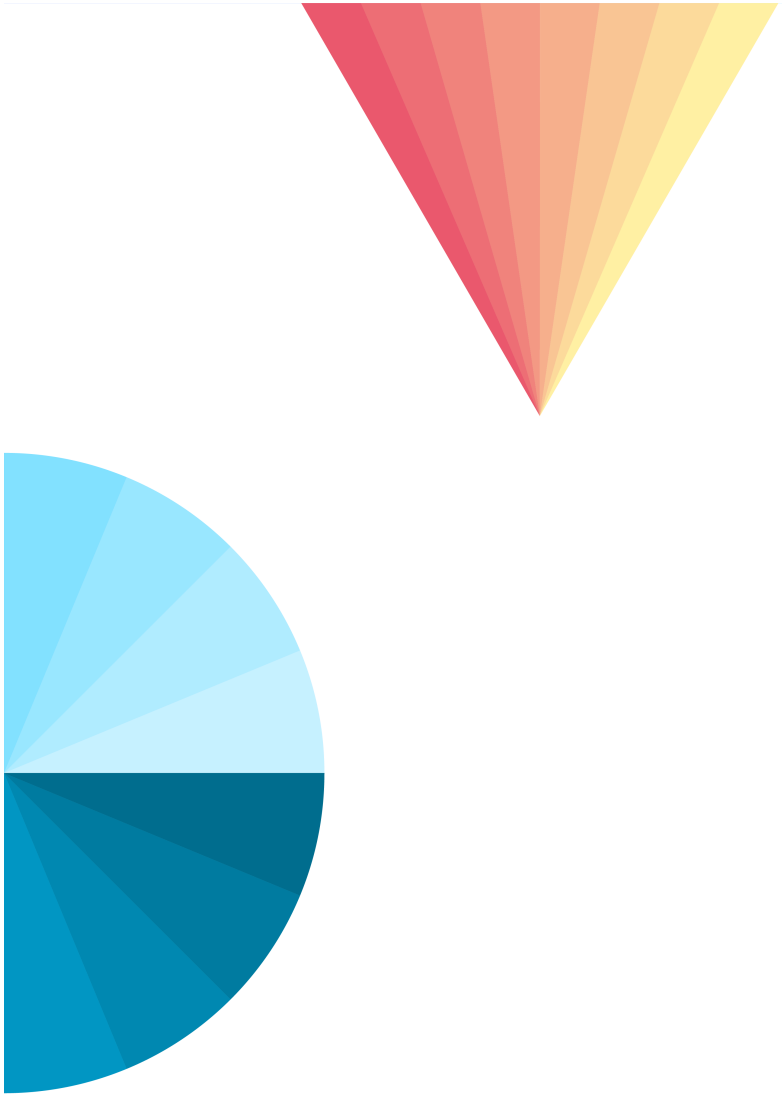



Performance test client
Inference Server
Collector
Tesla T4
Inference Server
Collector
Tesla T4 - MPS
Performance test client
Inference Server
Collector
Tesla P40
Performance test client
Inference Server
Collector
CPU
Inference Server
Collector
CPU 개수, 옵션
.....

# 5.5 autoscale



부하에 따라서 자동으로 컨테이너를 늘리거나 줄이는 기능
보통 CPU, MEMORY, GPU 사용율로 설정함
잘못 설정하면 동작 안함
  - ex) inference server가 최대 60%까지 GPU를 사용할수 있는데, GPU 사용율 70% 일때 늘리도록 설정하면 동작 안함.
성능 테스트 결과가 있으니, 최대 성능의 비율로 스케일 할 수 있음



# Q & A

# Thank You