AWS Machine Learning Stack

Overview of the services.
Closer look at the frameworks and infrastructure.

AI services (for Devs):
- Transcribe
- Personalise
- Rekognition
- Forecast

ML services SageMaker (for DS):
- Ground Truth (human labelling process)
- Notebook IDE or Jupyter Lab
- Algo (aws built-in models)
-Training
-Tune (automatic model tuning such as: random search, bayesian search)
- Neo (automatic optimise speed and memory consumption)
- Endpoint (automatic deployment)
- Automatically cloned on Git in Jupyter Lab
- Run RAPIDS on SageMaker

ML Framework & Infrastructure (Advanced DS):
Frameworks:
- TF, Pytorch, Mxnet
- Sklearn, SparkML

Infrastructure:
- EC2
- Deep Learning Container
- FPGAs (Field Programmable Data Arrays, NN training)
- GreenGrass (IoT)

AWS Deep Learning Containers

Docker images are pre-installed with deep learning frameworks
They are available from AWS marketplace or from Amazon Elastic Container ECS*

- run containers with ECS on EC2 on-demand instance
- * only support Mxnet and TF (because cost and complexity of optimizing)
- * we can run a container on EC2 with Deep Learning AMI

AWS Deep Learning Containers (Amazon marketplace)

- deploy on ECS
- deploy on Amazon Elastic Kubernetes Service EKS (used by RAPIDS)

AWS Deep Learning AMIs on EC2 - DLAMIs

AMI - Amazon Machine Image:
It’s is an instance to create a virtual machine in Amazon Elastic Compute Cloud EC2.
Includes OS and additional tools.
We can keep the data stored once we stop the instance.

All EC2 prices: https://ec2instances.info/

DLAMIs:

Conda AMI:

- Both Linux and Windows OS
- GPU drivers and Nvidia CUDA
- Pre-installed deep learning frameworks in a separate conda env (TF, Pytorch, Mxnet, Etc.)
- GPU monitoring
- Include Docker and Nvidia Docker

Base AMI:

- only Linux OS
- GPU drivers and Nvidia CUDA
- custom build of deep learning env

AWS Deep Learning AMIs on EC2 – DLAMIs
Possible appropriate solutions

1 - Amazon Linux 2 AMI with NVIDIA TESLA GPU Driver (Base AMI)
https://aws.amazon.com/marketplace/pp/Amazon-Web-Services-Amazon-Linux-2-AMI-with-NVIDIA/B07S5G9S1Z

2 - AWS Deep Learning AMI DLAMI (Conda AMI)
https://docs.aws.amazon.com/dlami/latest/devguide/what-is-dlami.html

- Deep Learning AMI with Conda options:
  https://docs.aws.amazon.com/dlami/latest/devguide/conda.html

  - AWS Deep Learning AMI (Ubuntu 18.04):
    https://aws.amazon.com/marketplace/pp/B07Y43P7X5

3 - BlazingDB
- GPU Data Science Cluster
- GPU Data Science Workstation
  - both use **G4** instances
  - low-cost

Amazon SageMaker

Model optimization
Model endpoint

Amazon Sagemaker makes use of pre-built Docker containers for building and runtime tasks.

Advantages:

- dedicated environment for training across one or many instances → **stored in S3**
- fully managed AWS cluster to run parallel **hyper-parameter** optimization
- aws neo:
  - neo complier: run the model in any frameworks (e.g. Pytorch, TF, XGBoost) and optimize it

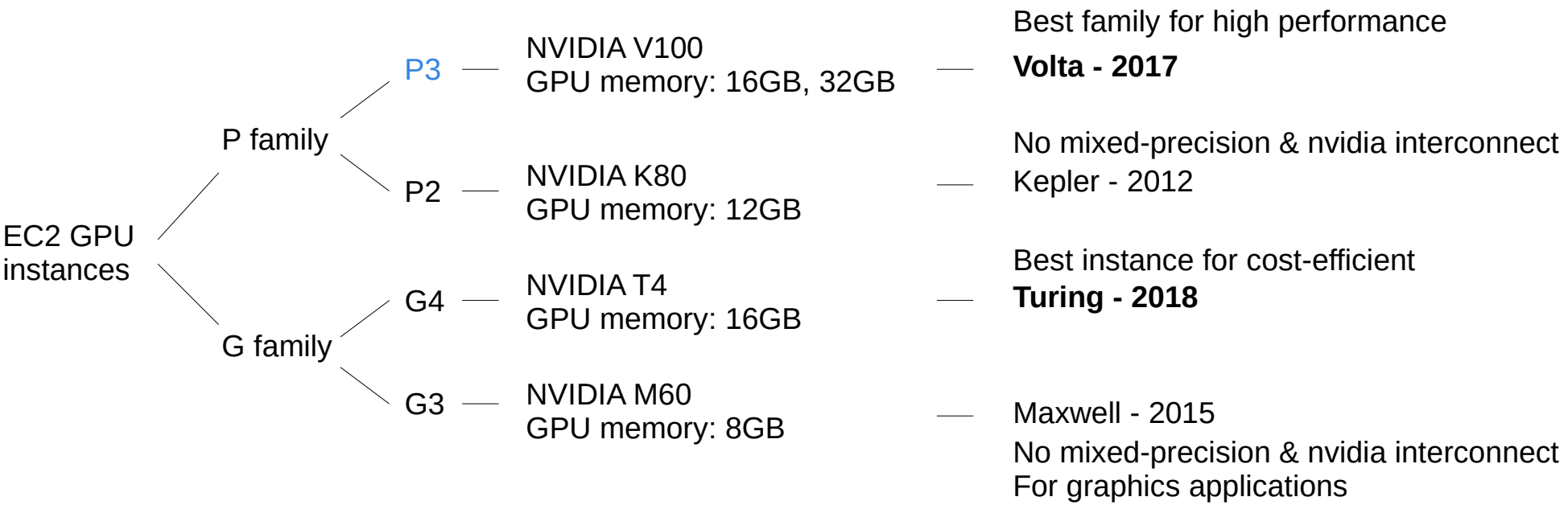
  - **avoid manual trial and error process during training:**
    - learning rate, number of layers, regularization, drop-out → **ANN**
    - number of trees, depth, boosting step size → **RF**
    - number of clusters, seed initialization, pre-processing → **Clustering**

- gaussian process regression model objective metric as function of hyper-parameter
  - works with low data (that is used in continuous training) → **hyperparameter_range(int or cont)**
    → **hyperparameter_tuner(max_jobs, max_parallel)**

  - **bayesian optimization** decides where to search next in grid search

Price

Nvidia GPUs


EC2 GPU instances
P family
G family
P3
P2
G4
G3
NVIDIA V100
GPU memory: 16GB, 32GB
NVIDIA K80
GPU memory: 12GB
NVIDIA T4
GPU memory: 16GB
NVIDIA M60
GPU memory: 8GB
Best family for high performance
Volta - 2017
No mixed-precision & nvidia interconnect
Kepler - 2012
Best instance for cost-efficient
Turing - 2018
Maxwell - 2015
No mixed-precision & nvidia interconnect
For graphics applications

Nvidia GPUs **Volta**

P3 — **NVIDIA V100**
GPU memory: 16GB, 32GB — Volta - 2017

Supported precision types: FP64, FP32, FP16, Mixed-precision

Local model training and prototyping — Single GPU:
p3.2xlarge (16GB / GPU) – 8 vGPUs – 61 mem

Distributed training — Multi-GPU:
Distributed training and large-scale experiments — **p3.8xlarge** (4 GPUs, 16GB / GPU) – 32 vGPU – 244 mem
Record setting performance on R/CNN and BERT — p3.16xlarge(8 GPUs, 16GB / GPU) – 64 vGPU – 488 mem
p3dn.24xlarge (8 GPUs, 32GB / GPU) – 96 vGPU – 768 mem

GPU interconnect:
NVLink high-bandwidth interconnect, 2nd generation (100 Gbps)

Nvidia GPUs **Turing**

G4 — **NVIDIA T4**
GPU memory: 16GB — Turing - 2018

Supported precision types: FP64, FP32, FP16, Mixed-precision

Model size
Number of models
Pre/Post-processing →

Single GPU:

g4nd.xlarge	4 vGPU – 16 mem
g4nd.2xlarge	8 vGPU – 32 mem
g4nd.4xlarge	16 vGPU – 64 mem
g4nd.8xlarge	32 vGPU – 128 mem
g4nd.16xlarge	64 vGPU – 256 mem

Target latency SLA
Real time vs. Batch predictions
Classical NN vs. custom code →

Multi-GPU:
g4nd.12xlarge (4 GPUs) – 4 vGPU – 16 mem
g4nd.metal (8 GPUs) – 8 vGPU – 32 mem

GPU interconnect: PCLe
GRID vGPU to increase the number of users
Best instance for cost-efficient deep learning training

The prepared dataset has to be copied into GPU memory and after training is done, results are copied back to system memory for post-processing and visualization. One downside of this approach is that moving data in and out of a GPU can affect overall processing times.

Nvidia
Storage
AMI backup

Object storage S3 — **Moderate and large dataset** — File mode (copy entire dataset to local volume)
Pipe mode (stream dataset from S3)
S3 pricing

Elastic Block Store EBS — **For DB** — Size (1GB to 16TB) - $0.10 per GB/month

Network drive (i.e. not a physical drive)
that allows our instance to persist data

Persist volumes for terminated instances

Detach / attach volume to a different EC2 instance

Storage is included as part of instance (EC2) pricing.

All EC2 prices: https://ec2instances.info/

EC2
Running Modes
Cost saving

On-Demand:
- auto scaling groups

Reserved Instances:
- discount (75% compared to On-Demand) when we use EC2 for long time
- **convertible RI:** we can change family (e.g. change from linux to win, change GPUs)
- **scheduler RI:** we can schedule specific launch period

Spot instances:
- cheaper instances we can lose at any time
- it uses the spare capacity in AWS with a discount up to 90%
- **at any time AWS can reclaim (terminate) our instance with 2 minutes of notification** → **Qubole**
- not for critical jobs, but we can use it for parallel computation

All EC2 prices: https://ec2instances.info/