AWS re:Invent

DECEMBER 2 – 6, 2024 | LAS VEGAS, NV

AIM380

High performance distributed model training with Amazon SageMaker

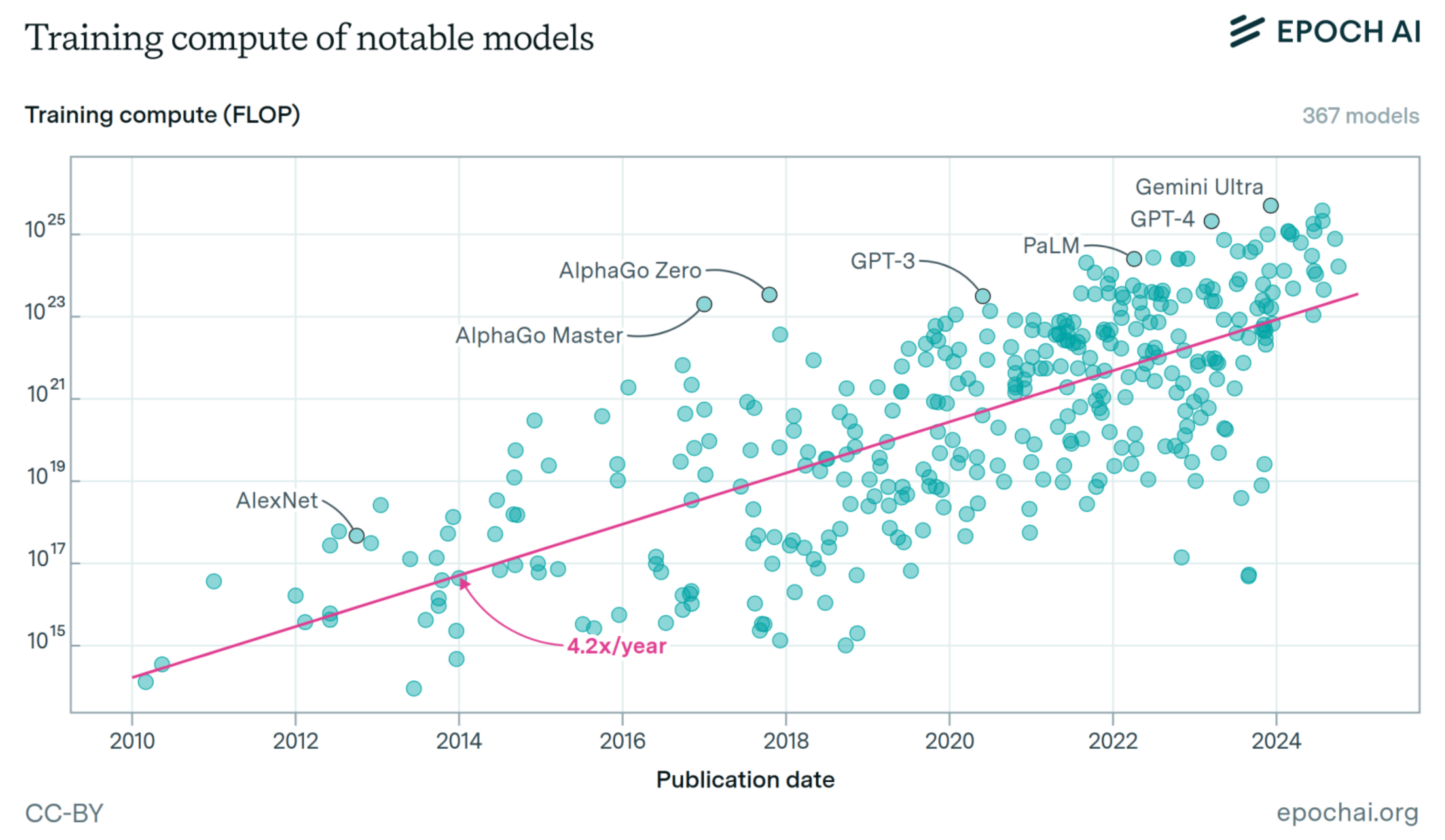
Anirudh Viswanathan
Sr. Product Manager, Technical,
Amazon SageMaker
AWS

Sanjay Dorairaj
Software Development Manager,
Amazon SageMaker
AWS

Antonio Ginart, Ph. D.
Lead Scientist,
AI Research
Salesforce

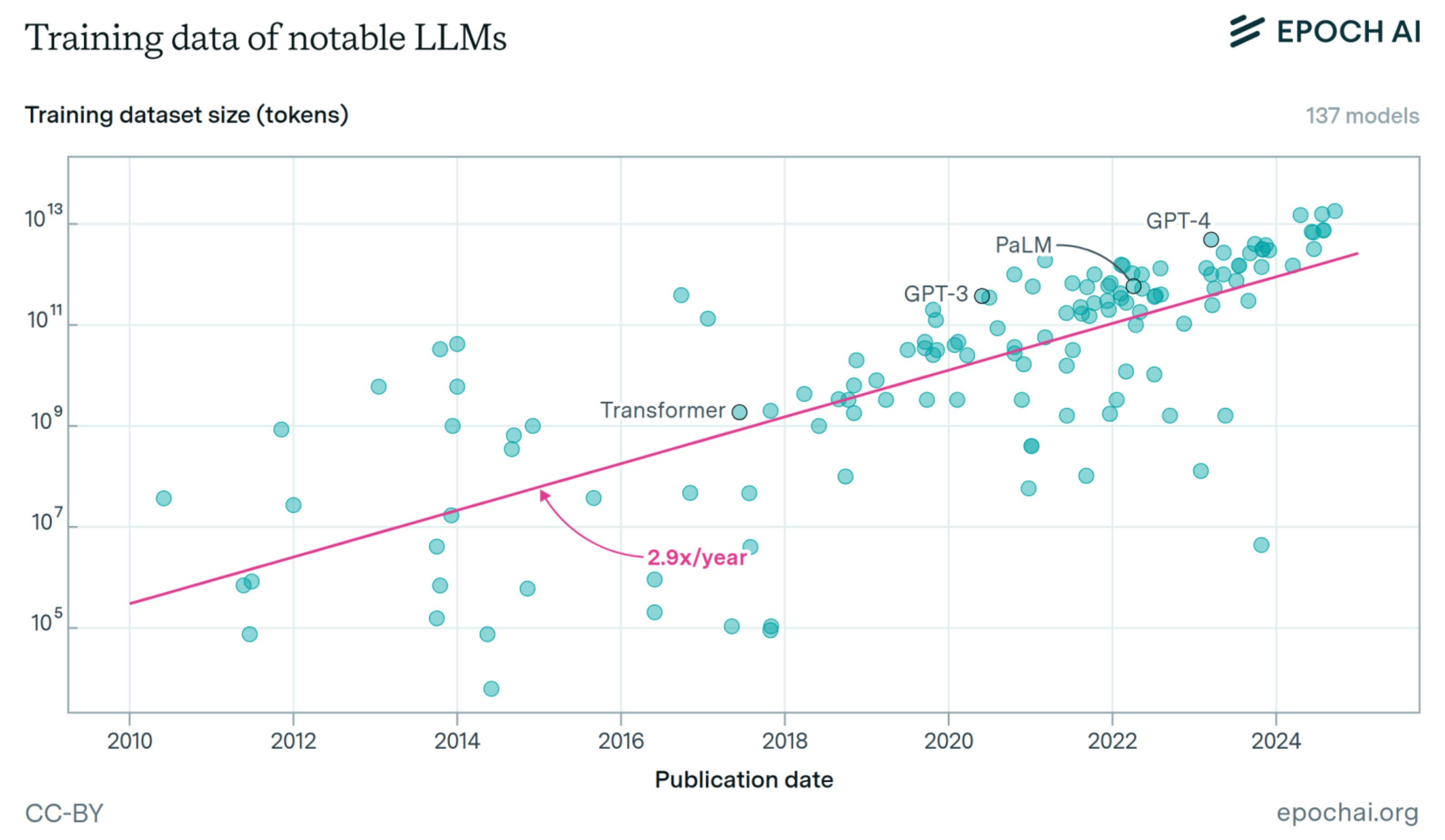
© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Training compute of foundational models is growing by ~5x per year, doubling ~6 months



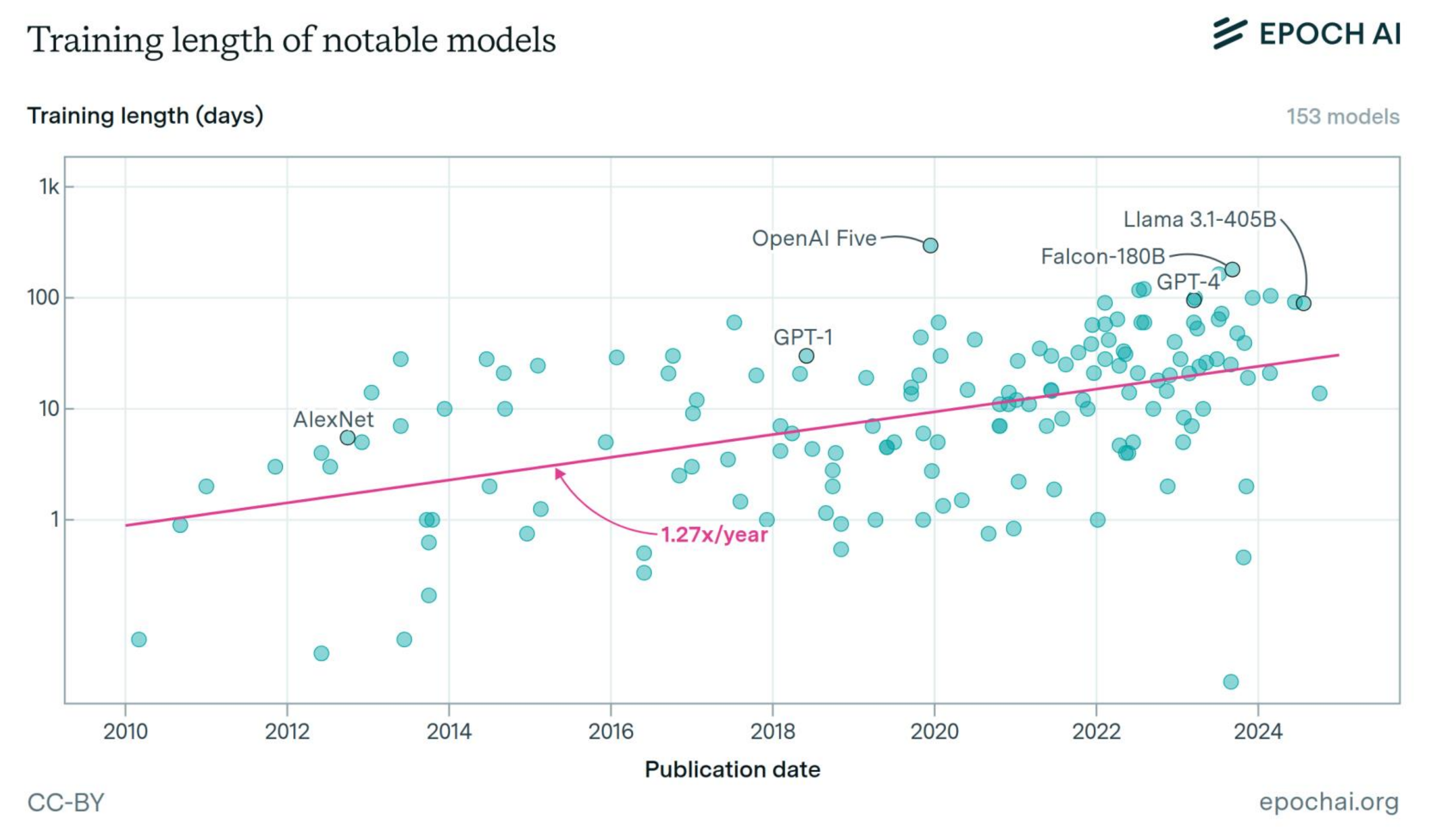
© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Dataset sizes are doubling every 8 months



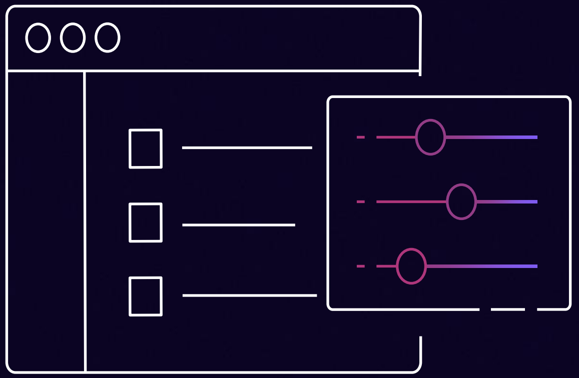
© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Time to market spans months of investment and continues to increase



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Unique challenges to manage hardware resources efficiently for large-scale FM training



Cluster provisioning and management

Infrastructure stability

Distributed training performance

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon SageMaker HyperPod

Scale and accelerate generative AI model development across thousands of AI accelerators

Designed for scale

Reduce model training time by 40%

Resilient environment

Advance observability and control

Top AI companies use HyperPod to train and deploy models

Articul8
Luma AI
BAYER
Coastal Carbon
datologyai
featherless.ai

Hippocratic AI
— Do No Harm —
HOPPR
Hugging Face
MOHAMED BIN ZAYED
UNIVERSITY OF
ARTIFICIAL INTELLIGENCE
OMAN
DATAPARK

perplexity
salesforce
Thomson Reuters
Twelve Labs
ubitus
H

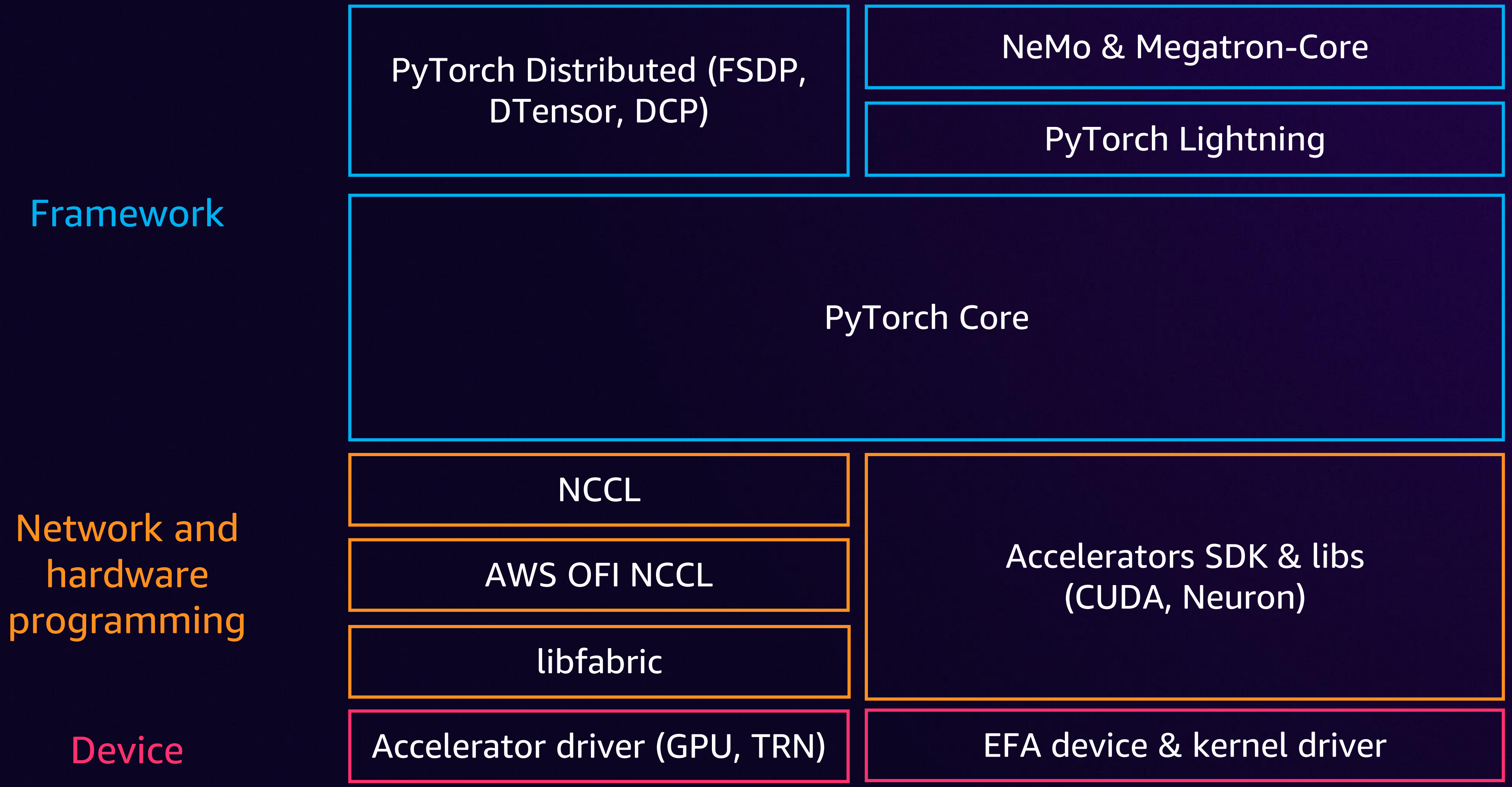
OpenBabylon
NOETIK
NinjaTech AI
Orbital
Stockmark
WRITER
arcee.ai

aws

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Optimizing the distributed training software stack

ILLUSTRATIVE OSS STACK



Framework

PyTorch Distributed (FSDP, DTensor, DCP)

NeMo & Megatron-Core

PyTorch Lightning

PyTorch Core

Network and hardware programming

NCCL

AWS OFI NCCL

libfabric

Accelerators SDK & libs (CUDA, Neuron)

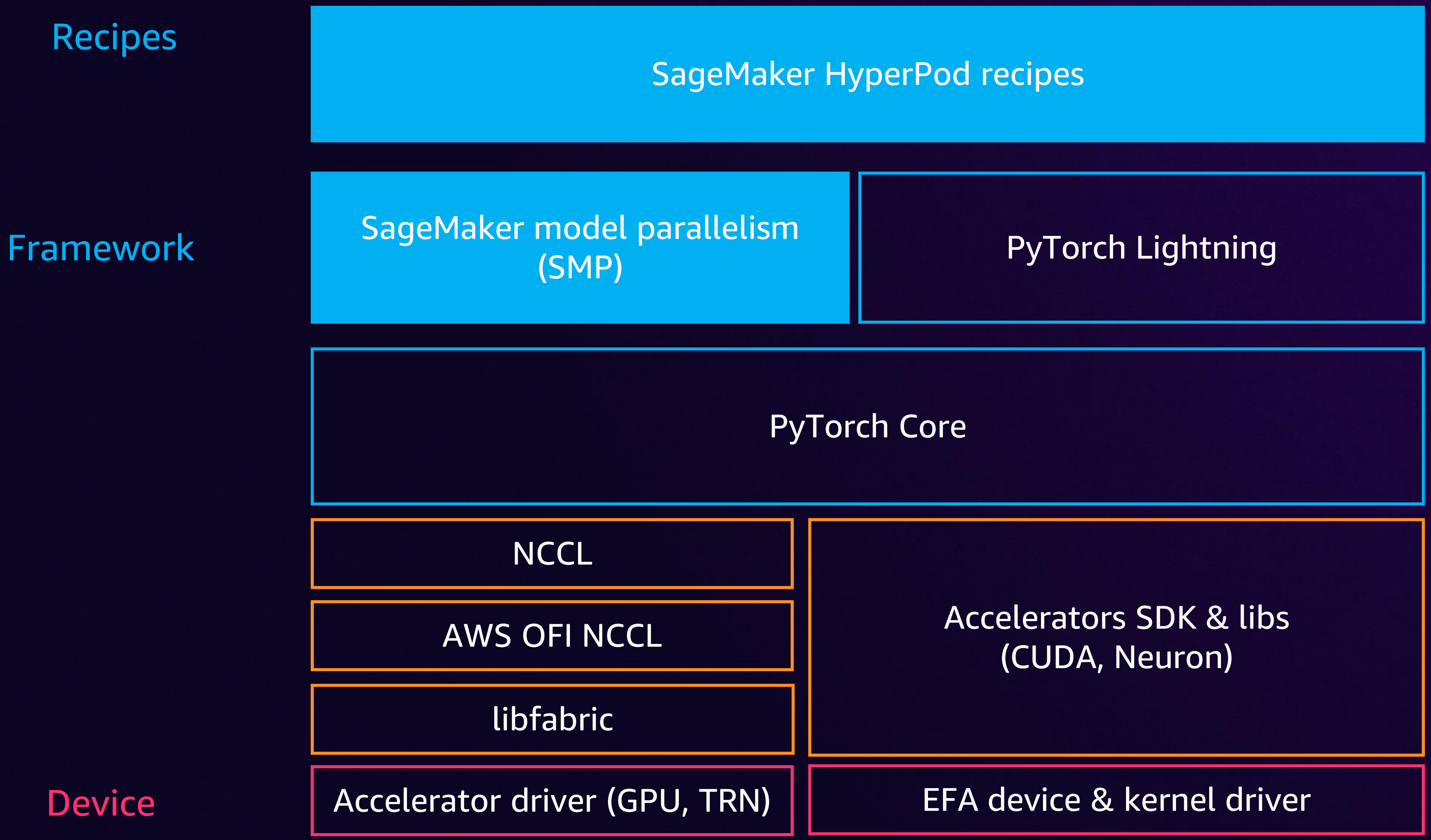
Device

Accelerator driver (GPU, TRN)

EFA device & kernel driver

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Optimizing the distributed training software stack



Recipes

SageMaker HyperPod recipes

Framework

SageMaker model parallelism (SMP)

PyTorch Lightning

PyTorch Core

NCCL

AWS OFI NCCL

libfabric

Accelerators SDK & libs (CUDA, Neuron)

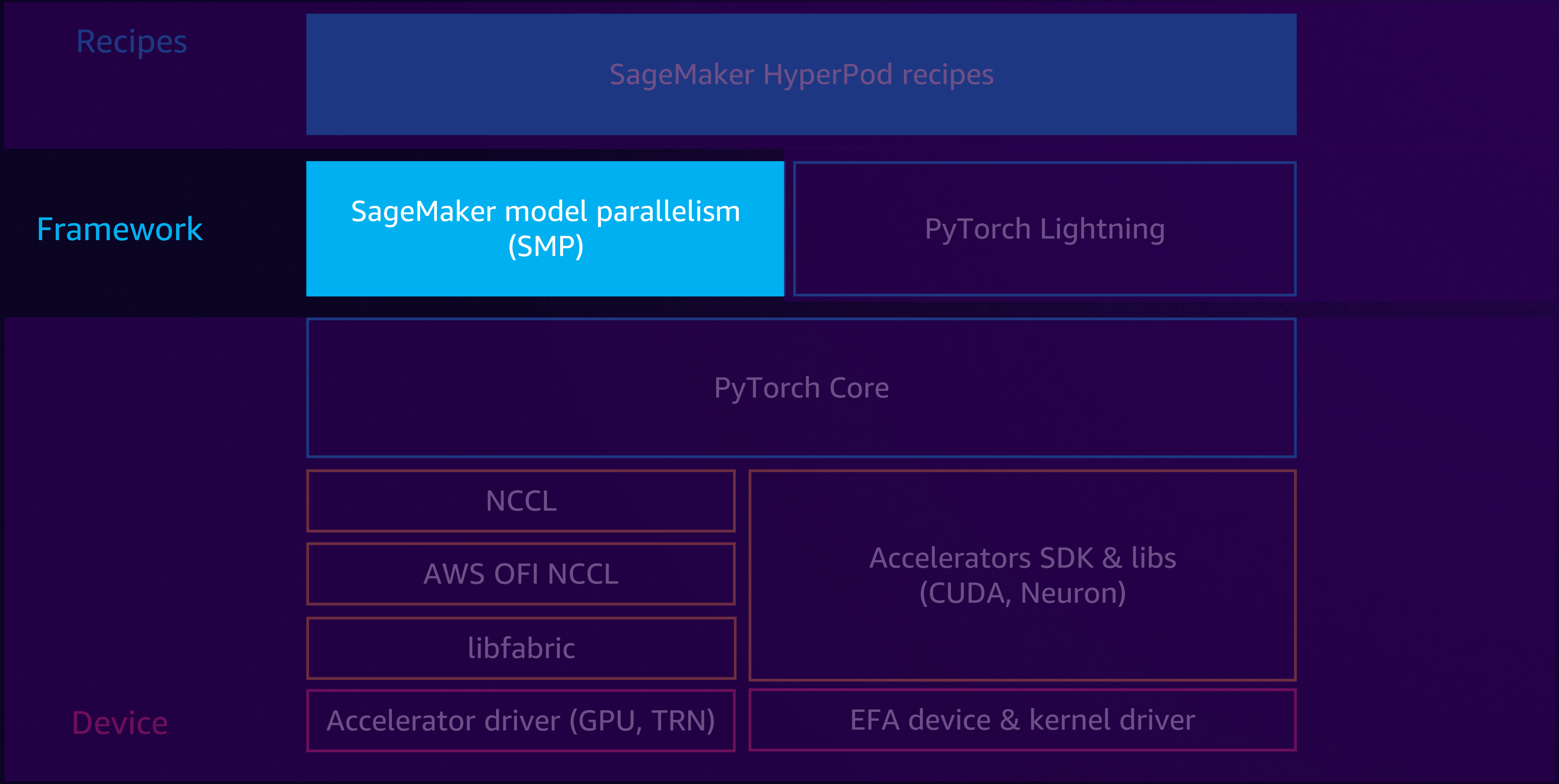
Device

Accelerator driver (GPU, TRN)

EFA device & kernel driver

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Optimizing the distributed training software stack



Recipes

SageMaker HyperPod recipes

Framework

SageMaker model parallelism (SMP)

PyTorch Lightning

PyTorch Core

NCCL

AWS OFI NCCL

libfabric

Accelerators SDK & libs (CUDA, Neuron)

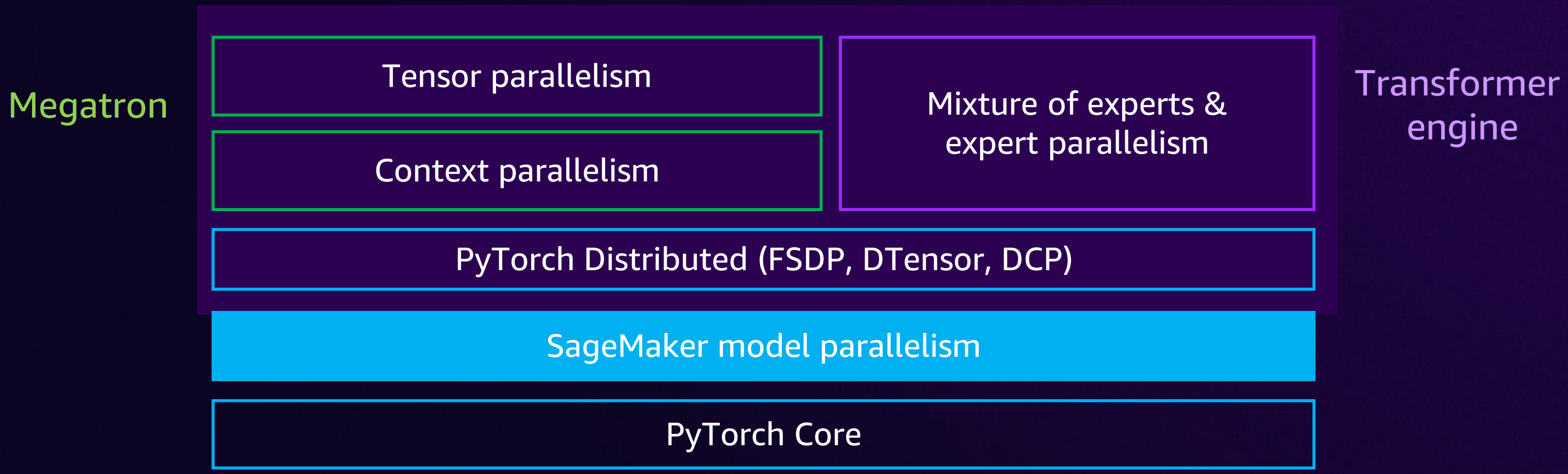
Device

Accelerator driver (GPU, TRN)

EFA device & kernel driver

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

PyTorch fork with best-in-class, composable training techniques that are mutually compatible



Megatron

Tensor parallelism

Context parallelism

Mixture of experts & expert parallelism

Transformer engine

PyTorch Distributed (FSDP, DTensor, DCP)

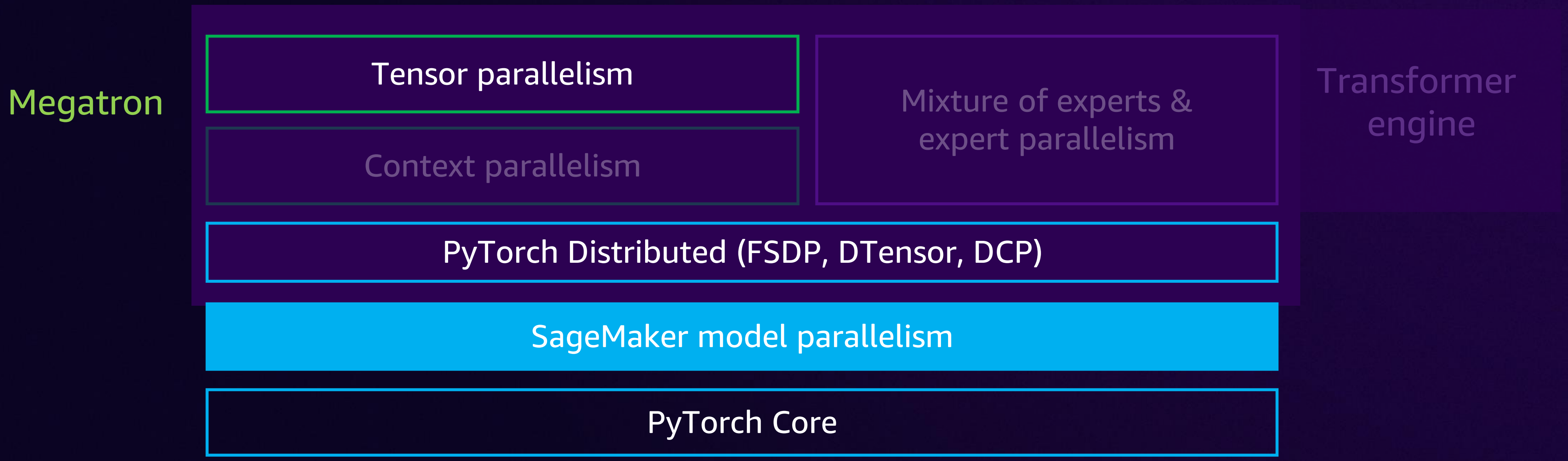
SageMaker model parallelism

PyTorch Core

Also includes: Delayed parameter initialization, activation checkpointing, activation offloading, etc.

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

PyTorch fork with best-in-class, composable training techniques that are mutually compatible



Megatron

Tensor parallelism

Context parallelism

Mixture of experts & expert parallelism

Transformer engine

PyTorch Distributed (FSDP, DTensor, DCP)

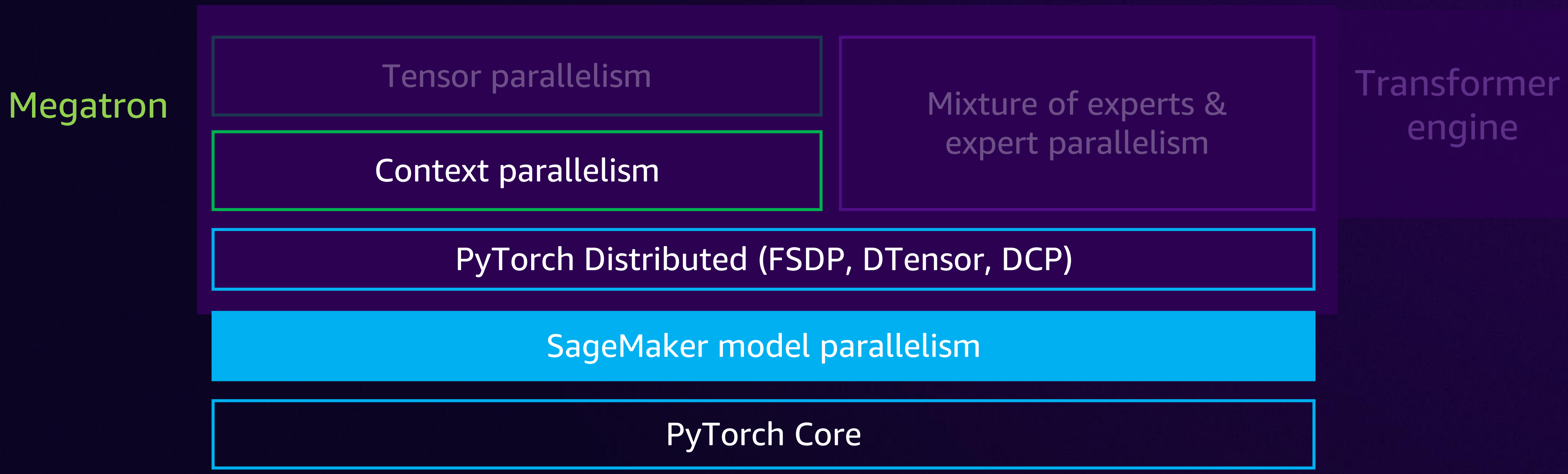
SageMaker model parallelism

PyTorch Core

Supports training with TP + FSDP

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

PyTorch fork with best-in-class, composable training techniques that are mutually compatible



Megatron

Tensor parallelism

Context parallelism

Mixture of experts & expert parallelism

Transformer engine

PyTorch Distributed (FSDP, DTensor, DCP)

SageMaker model parallelism

PyTorch Core

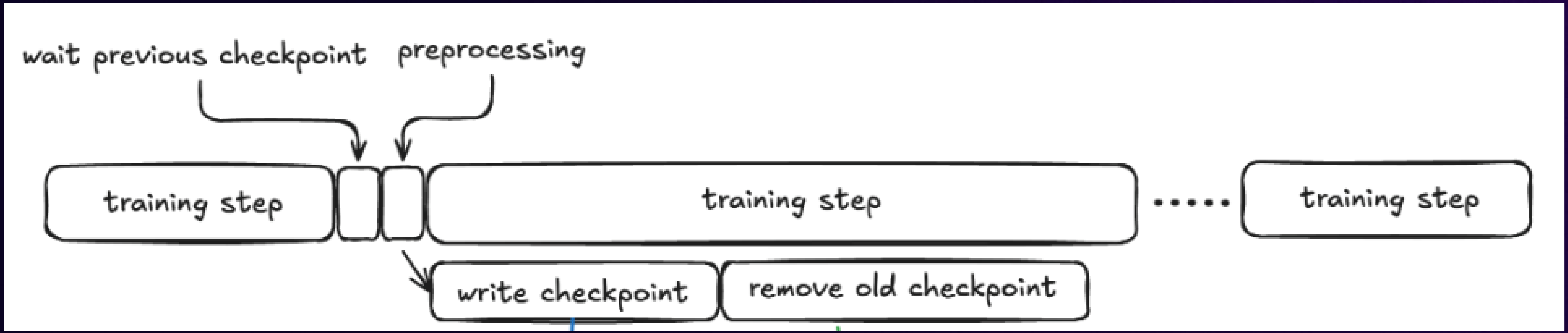
Supports training with CP + FSDP, FSDP + FP8 etc.

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Async checkpointing for accelerated training

ASYNC CHECKPOINT

- Compatible with native PyTorch Distributed Checkpoint (DCP)

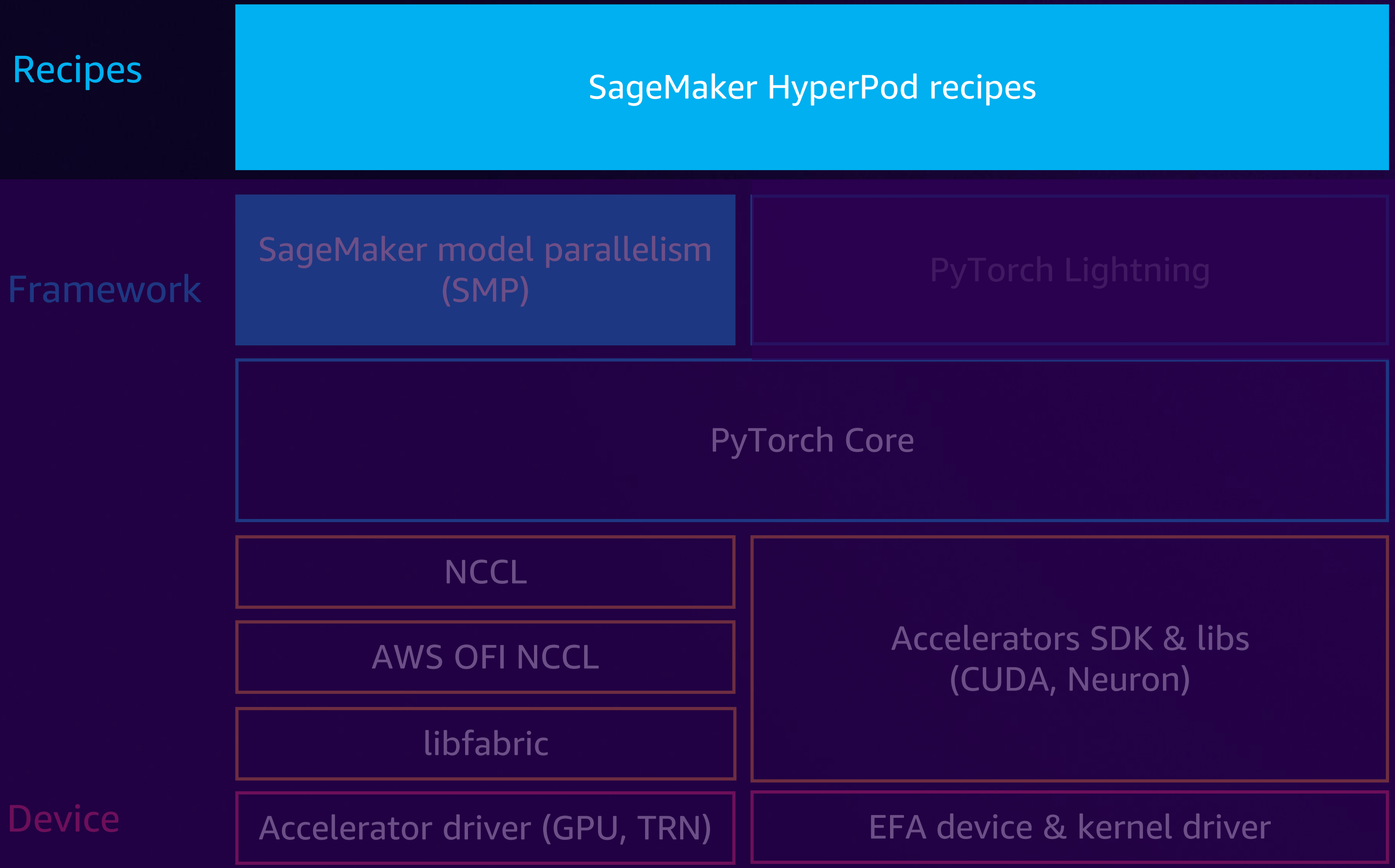


Optimizations

- Smart metadata caching system for faster checkpoint saving
- Native Amazon S3 support to use S3 links as the checkpoint saving/loading destination

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Optimizing the distributed training software stack



Recipes

SageMaker HyperPod recipes

Framework

SageMaker model parallelism (SMP)

PyTorch Lightning

PyTorch Core

NCCL

AWS OFI NCCL

libfabric

Accelerators SDK & libs (CUDA, Neuron)

Device

Accelerator driver (GPU, TRN)

EFA device & kernel driver

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Audience poll

How many of you are currently pre-training and fine-tuning LLMs?

How many of you are using SageMaker for LLM training?

How many of you plan to start LLM training in the next 6 months?

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Optimizing FM pre-training and fine-tuning can take weeks of effort

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Optimizing FM pre-training and fine-tuning can take weeks of effort

~minutes

Select a model

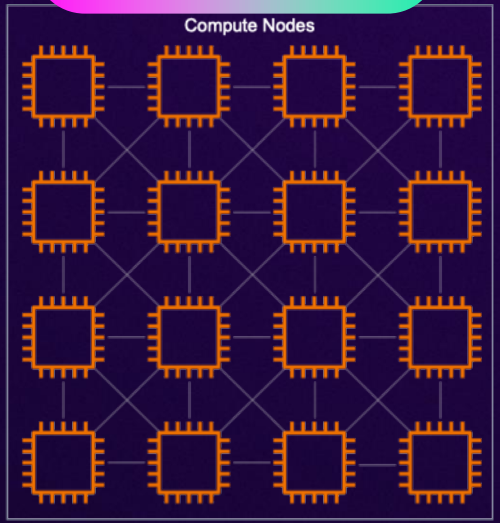
~hours

Configure framework

~weeks

Optimize model training

~months



Production pre-training & fine-tuning

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Why FM training falters: The technical bottleneck

CHANGING A SINGLE PARAMETER CAN CASCADE INTO OTHER TRAINING STACK CHANGES

- 70+ pre-training parameters and 100+ fine-tuning parameters
- Each parameter choice triggers cascading updates to other parameters
- Default choices may be suboptimal
- Leads to cost overruns, missed deadlines, and reduced productivity

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon SageMaker HyperPod recipes

Curated, ready-to-use recipes for pre-training and fine-tuning popular publicly available FMs

Tested and validated by AWS for foundational models such as Llama & Mistral

Automatic checkpoints for faster fault recovery and managed end-to-end training loop

Easily switch between GPU-based or Trainium-based instances

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Getting started in minutes

RUN FM PRE-TRAINING AND FINE-TUNING WITH A SINGLE LINE OF CODE

Select

a model training and fine-tuning recipe on GitHub

Set up prerequisites

Resource limits, AWS credentials, a training cluster

Run the recipe

on Amazon SageMaker HyperPod
or
SageMaker training jobs

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Getting started

HOW CAN YOU GET STARTED WITH HYPERPOD RECIPES?

https://github.com/aws/sagemaker-hyperpod-recipes

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How it works

AMAZON SAGEMAKER

HYPERPOD TRAINING RECIPES REPOSITORY

Open source implementation that includes launcher and recipes collection

Built on NeMo foundations (launcher, configuration hierarchy)

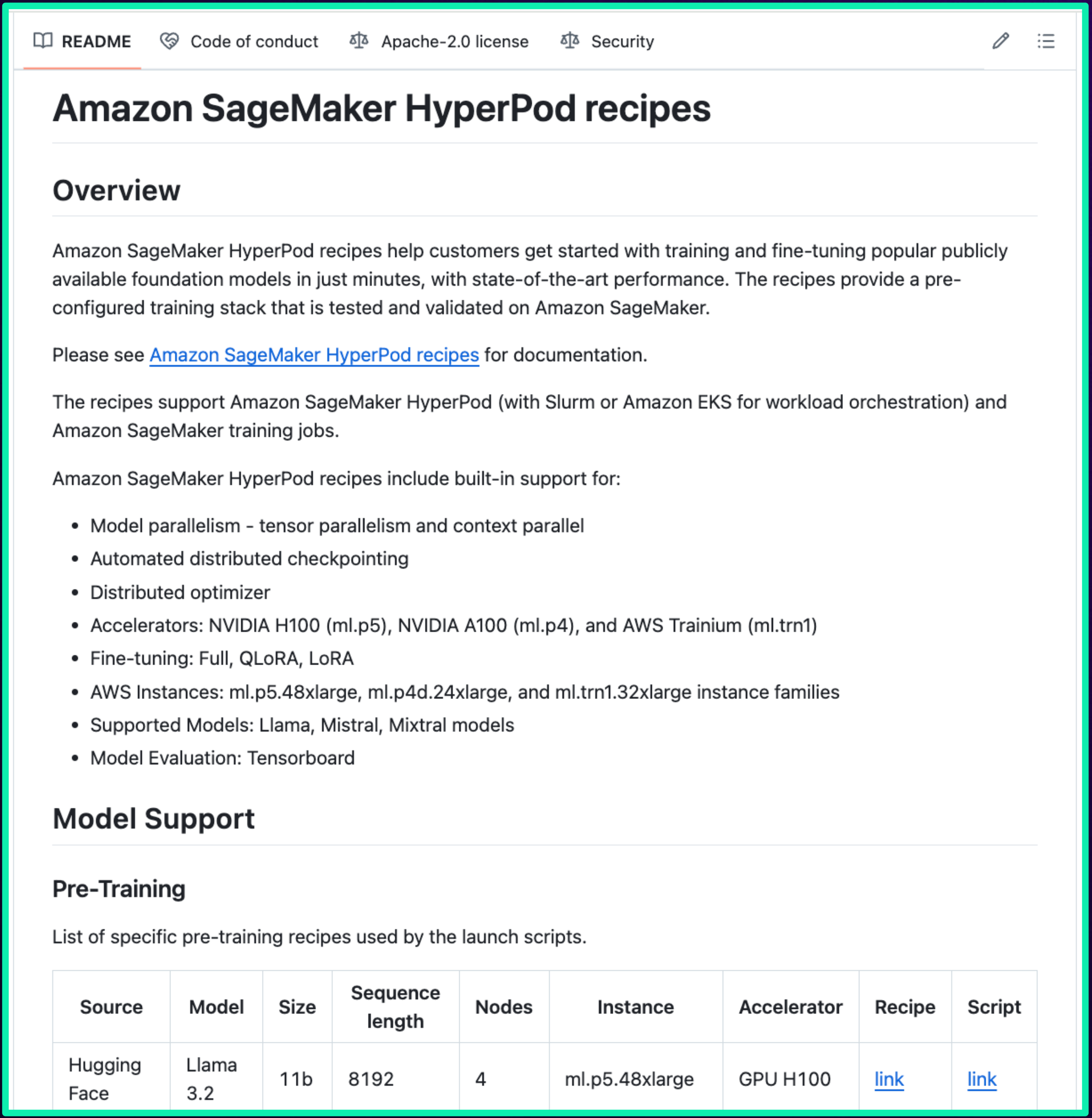
Over 30 recipes with different configurations

SageMaker optimized models (GPU)

AWS Neuron optimized models (Trainium)

Native NeMo models

Custom models



Amazon SageMaker HyperPod recipes

Overview

Amazon SageMaker HyperPod recipes help customers get started with training and fine-tuning popular publicly available foundation models in just minutes, with state-of-the-art performance. The recipes provide a pre-configured training stack that is tested and validated on Amazon SageMaker.

Please see Amazon SageMaker HyperPod recipes for documentation.

The recipes support Amazon SageMaker HyperPod (with Slurm or Amazon EKS for workload orchestration) and Amazon SageMaker training jobs.

Amazon SageMaker HyperPod recipes include built-in support for:

- Model parallelism - tensor parallelism and context parallel
- Automated distributed checkpointing
- Distributed optimizer
- Accelerators: NVIDIA H100 (ml.p5), NVIDIA A100 (ml.p4), and AWS Trainium (ml.trn1)
- Fine-tuning: Full, QLoRA, LoRA
- AWS Instances: ml.p5.48xlarge, ml.p4d.24xlarge, and ml.trn1.32xlarge instance families
- Supported Models: Llama, Mistral, Mixtral models
- Model Evaluation: Tensorboard

Model Support

Pre-Training

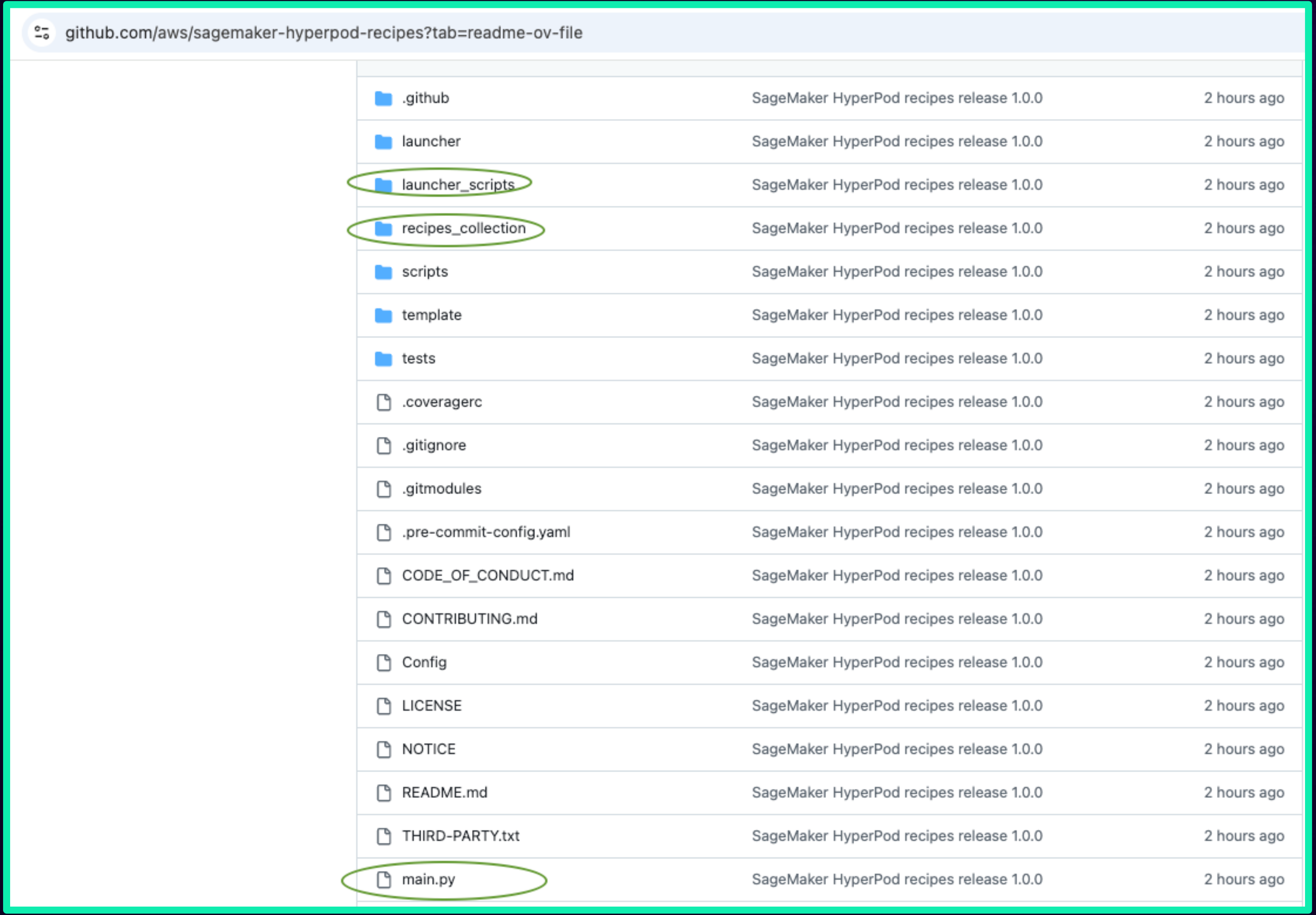
List of specific pre-training recipes used by the launch scripts.

Source	Model	Size	Sequence length	Nodes	Instance	Accelerator	Recipe	Script
Hugging Face	Llama 3.2	11b	8192	4	ml.p5.48xlarge	GPU H100	link	link

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Contents of the repository

SAGEMAKER HYPERPOD RECIPE REPOSITORY



github.com/aws/sagemaker-hyperpod-recipes?tab=readme-ov-file

.github	SageMaker HyperPod recipes release 1.0.0	2 hours ago
launcher	SageMaker HyperPod recipes release 1.0.0	2 hours ago
launcher_scripts	SageMaker HyperPod recipes release 1.0.0	2 hours ago
recipes_collection	SageMaker HyperPod recipes release 1.0.0	2 hours ago
scripts	SageMaker HyperPod recipes release 1.0.0	2 hours ago
template	SageMaker HyperPod recipes release 1.0.0	2 hours ago
tests	SageMaker HyperPod recipes release 1.0.0	2 hours ago
.coveragerc	SageMaker HyperPod recipes release 1.0.0	2 hours ago
.gitignore	SageMaker HyperPod recipes release 1.0.0	2 hours ago
.gitmodules	SageMaker HyperPod recipes release 1.0.0	2 hours ago
.pre-commit-config.yaml	SageMaker HyperPod recipes release 1.0.0	2 hours ago
CODE_OF_CONDUCT.md	SageMaker HyperPod recipes release 1.0.0	2 hours ago
CONTRIBUTING.md	SageMaker HyperPod recipes release 1.0.0	2 hours ago
Config	SageMaker HyperPod recipes release 1.0.0	2 hours ago
LICENSE	SageMaker HyperPod recipes release 1.0.0	2 hours ago
NOTICE	SageMaker HyperPod recipes release 1.0.0	2 hours ago
README.md	SageMaker HyperPod recipes release 1.0.0	2 hours ago
THIRD-PARTY.txt	SageMaker HyperPod recipes release 1.0.0	2 hours ago
main.py	SageMaker HyperPod recipes release 1.0.0	2 hours ago

launcher_scripts contains pre-configured bash scripts for model training

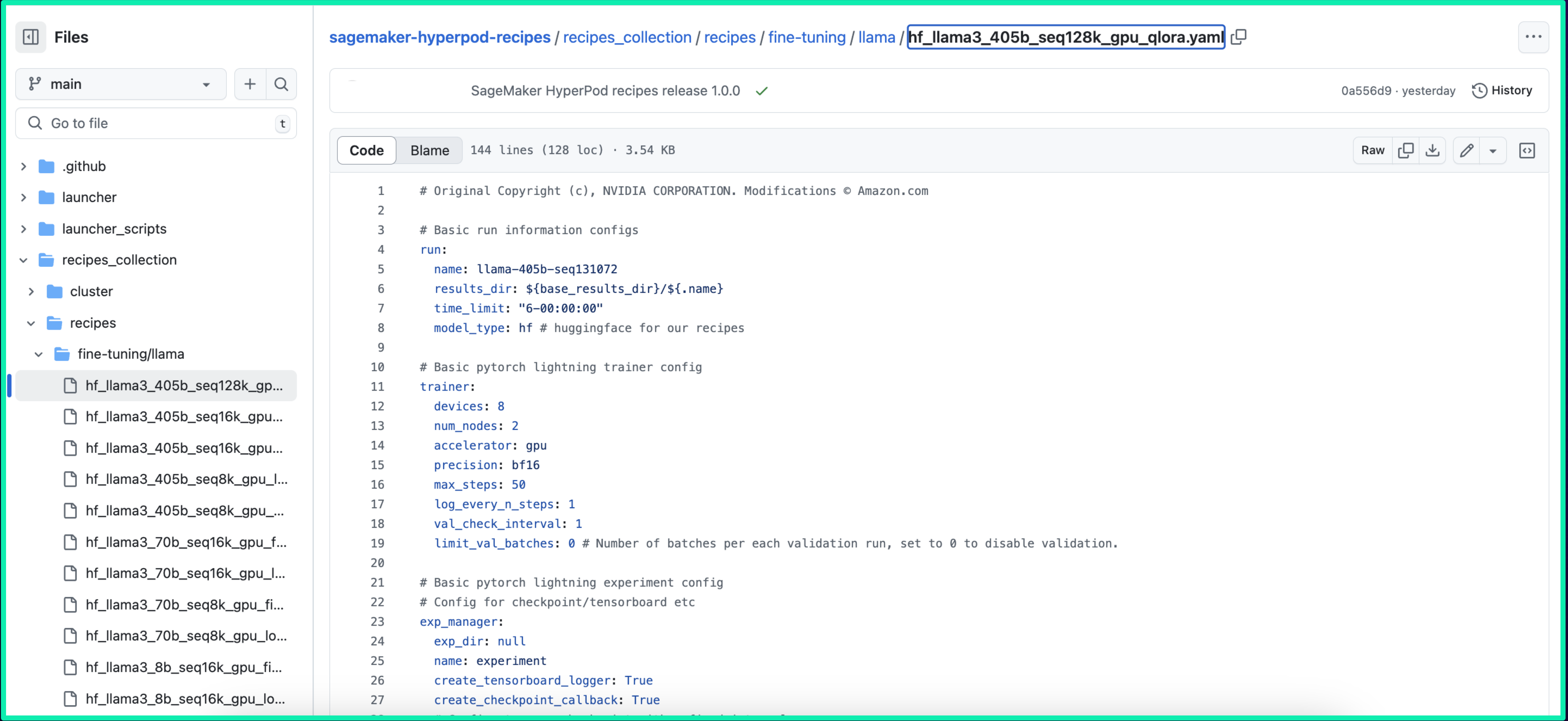
recipes_collection contains Hydra-based YAML training and fine-tuning recipes

main.py is a Nemo-style launcher

All dependencies for HyperPod recipes are present in a docker container and an enroot-based filesystem

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Getting started: Step 1 – Pick a recipe


Files
main
Go to file
t
.github
launcher
launcher_scripts
recipes_collection
cluster
recipes
fine-tuning/llama
hf_llama3_405b_seq128k_gp...
hf_llama3_405b_seq16k_gpu...
hf_llama3_405b_seq16k_gpu...
hf_llama3_405b_seq8k_gpu_l...
hf_llama3_405b_seq8k_gpu_...
hf_llama3_70b_seq16k_gpu_f...
hf_llama3_70b_seq16k_gpu_l...
hf_llama3_70b_seq8k_gpu_fi...
hf_llama3_70b_seq8k_gpu_lo...
hf_llama3_8b_seq16k_gpu_fi...
hf_llama3_8b_seq16k_gpu_lo...
History
Code
Blame
Raw

sagemaker-hyperpod-recipes / recipes_collection / recipes / fine-tuning / llama / hf_llama3_405b_seq128k_gpu_qlora.yaml

SageMaker HyperPod recipes release 1.0.0 ✓ 0a556d9 · yesterday

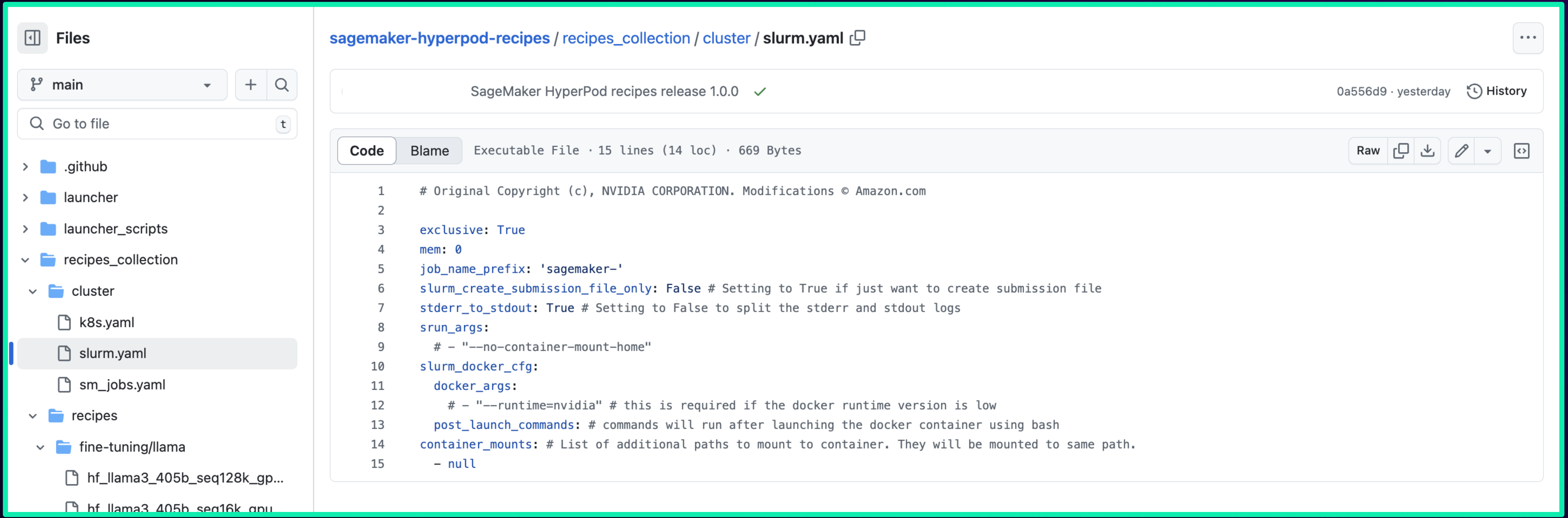
144 lines (128 loc) · 3.54 KB

```yaml
# Original Copyright (c), NVIDIA CORPORATION. Modifications © Amazon.com

# Basic run information configs
run:
  name: llama-405b-seq131072
  results_dir: ${base_results_dir}/${.name}
  time_limit: "6-00:00:00"
  model_type: hf # huggingface for our recipes

# Basic pytorch lightning trainer config
trainer:
  devices: 8
  num_nodes: 2
  accelerator: gpu
  precision: bf16
  max_steps: 50
  log_every_n_steps: 1
  val_check_interval: 1
  limit_val_batches: 0 # Number of batches per each validation run, set to 0 to disable validation.

# Basic pytorch lightning experiment config
# Config for checkpoint/tensorboard etc
exp_manager:
  exp_dir: null
  name: experiment
  create_tensorboard_logger: True
  create_checkpoint_callback: True
```

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

https://github.com/aws/sagemaker-hyperpod-recipes

Getting started: Step 2 – Specify the cluster config

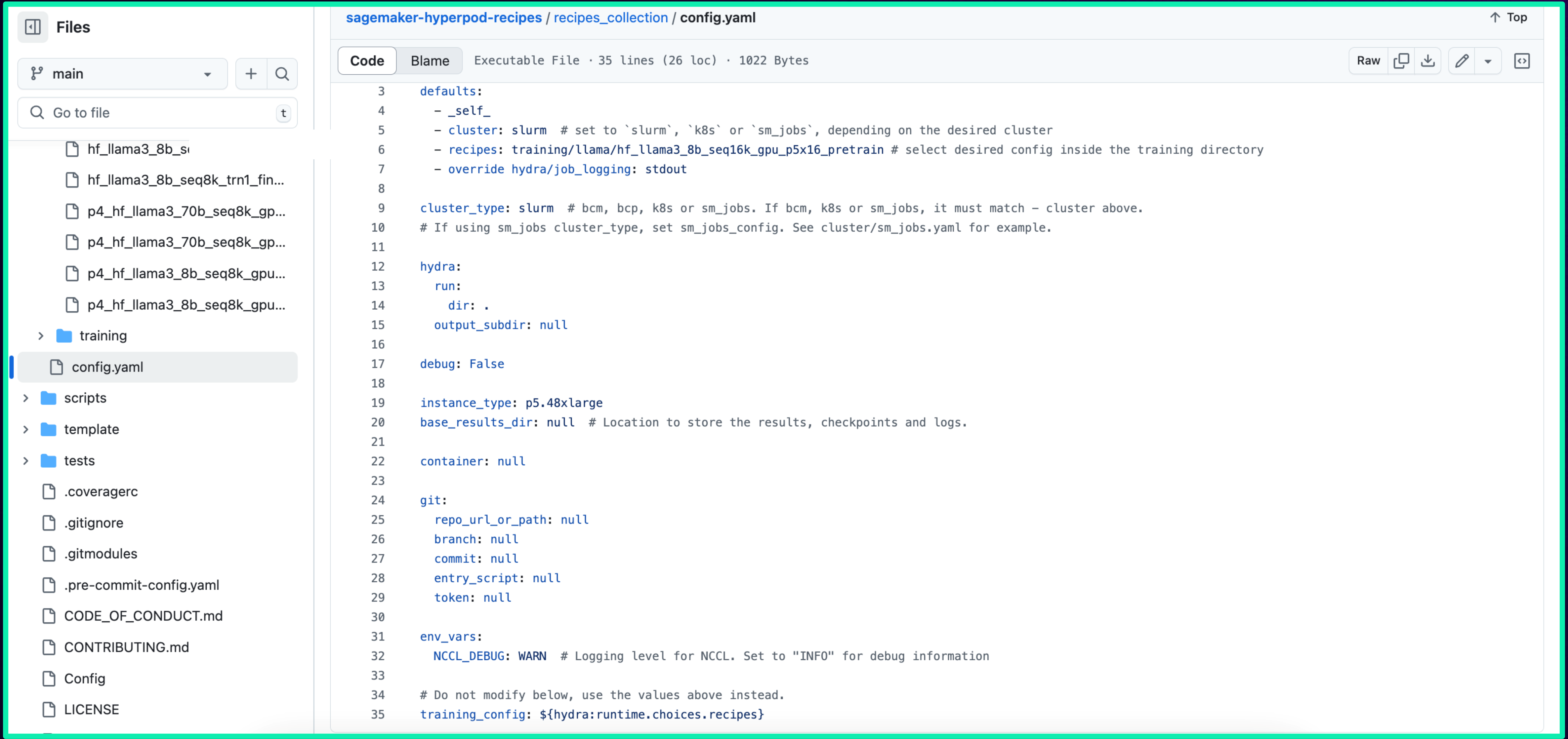

Files
main
Go to file
t
.github
launcher
launcher_scripts
recipes_collection
cluster
k8s.yaml
slurm.yaml
sm_jobs.yaml
recipes
fine-tuning/llama
hf_llama3_405b_seq128k_gp...
sagemaker-hyperpod-recipes / recipes_collection / cluster / slurm.yaml
SageMaker HyperPod recipes release 1.0.0 ✓
0a556d9 · yesterday
History
Code
Blame
Executable File · 15 lines (14 loc) · 669 Bytes
Raw

```yaml
# Original Copyright (c), NVIDIA CORPORATION. Modifications © Amazon.com

exclusive: True
mem: 0
job_name_prefix: 'sagemaker-'
slurm_create_submission_file_only: False # Setting to True if just want to create submission file
stderr_to_stdout: True # Setting to False to split the stderr and stdout logs
srun_args:
  # - "--no-container-mount-home"
slurm_docker_cfg:
  docker_args:
    # - "--runtime=nvidia" # this is required if the docker runtime version is low
  post_launch_commands: # commands will run after launching the docker container using bash
container_mounts: # List of additional paths to mount to container. They will be mounted to same path.
  - null
```

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

https://github.com/aws/sagemaker-hyperpod-recipes

Getting started: Step 3 – Update root-level config


Files
main
Go to file
t
hf_llama3_8b_s
hf_llama3_8b_seq8k_trn1_fin...
p4_hf_llama3_70b_seq8k_gp...
p4_hf_llama3_70b_seq8k_gp...
p4_hf_llama3_8b_seq8k_gpu...
p4_hf_llama3_8b_seq8k_gpu...
training
config.yaml
scripts
template
tests
.coveragerc
.gitignore
.gitmodules
.pre-commit-config.yaml
CODE_OF_CONDUCT.md
CONTRIBUTING.md
Config
LICENSE
sagemaker-hyperpod-recipes / recipes_collection / config.yaml
Top
Code
Blame
Executable File · 35 lines (26 loc) · 1022 Bytes
Raw

```yaml
defaults:
  - _self_
  - cluster: slurm  # set to `slurm`, `k8s` or `sm_jobs`, depending on the desired cluster
  - recipes: training/llama/hf_llama3_8b_seq16k_gpu_p5x16_pretrain # select desired config inside the training directory
  - override hydra/job_logging: stdout

cluster_type: slurm  # bcm, bcp, k8s or sm_jobs. If bcm, k8s or sm_jobs, it must match - cluster above.
# If using sm_jobs cluster_type, set sm_jobs_config. See cluster/sm_jobs.yaml for example.

hydra:
  run:
    dir: .
  output_subdir: null

debug: False

instance_type: p5.48xlarge
base_results_dir: null  # Location to store the results, checkpoints and logs.

container: null

git:
  repo_url_or_path: null
  branch: null
  commit: null
  entry_script: null
  token: null

env_vars:
  NCCL_DEBUG: WARN  # Logging level for NCCL. Set to "INFO" for debug information

# Do not modify below, use the values above instead.
training_config: ${hydra:runtime.choices.recipes}
```

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

https://github.com/aws/sagemaker-hyperpod-recipes

Getting started – Slurm/k8s

RUN FM PRE-TRAINING AND FINE-TUNING WITH A SINGLE LINE OF CODE

Launcher scripts

```
bash launcher_scripts/<model>/<launcher-script>
```

HyperPod Slurm/k8s

Recipes on SageMaker HyperPod (Amazon EKS)

```
hyperpod start-job --recipe recipe-name
```

HyperPod CLI

Recipes on SageMaker HyperPod (Slurm./k8s)

```
python3 main.py recipes=recipe-name
```

NeMo-style launcher

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Getting started – SageMaker training jobs

RUN FM PRE-TRAINING AND FINE-TUNING WITH A SINGLE LINE OF CODE

Recipes on SageMaker training jobs

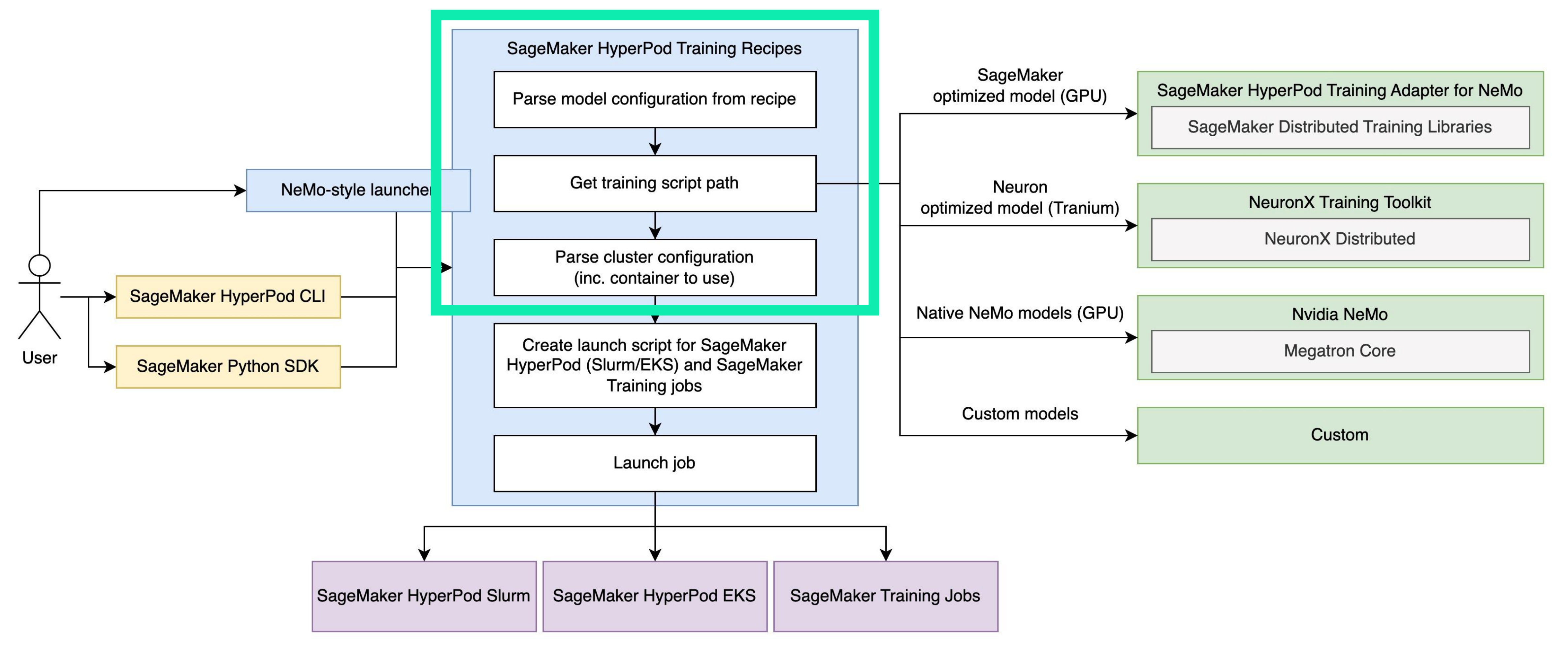
```
estimator = PyTorch(

   training_recipe=recipe-name
   ...,
)

estimator.fit(...)
```

SageMaker Python SDK

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Putting it all together

JOB SUBMISSION WORKFLOW

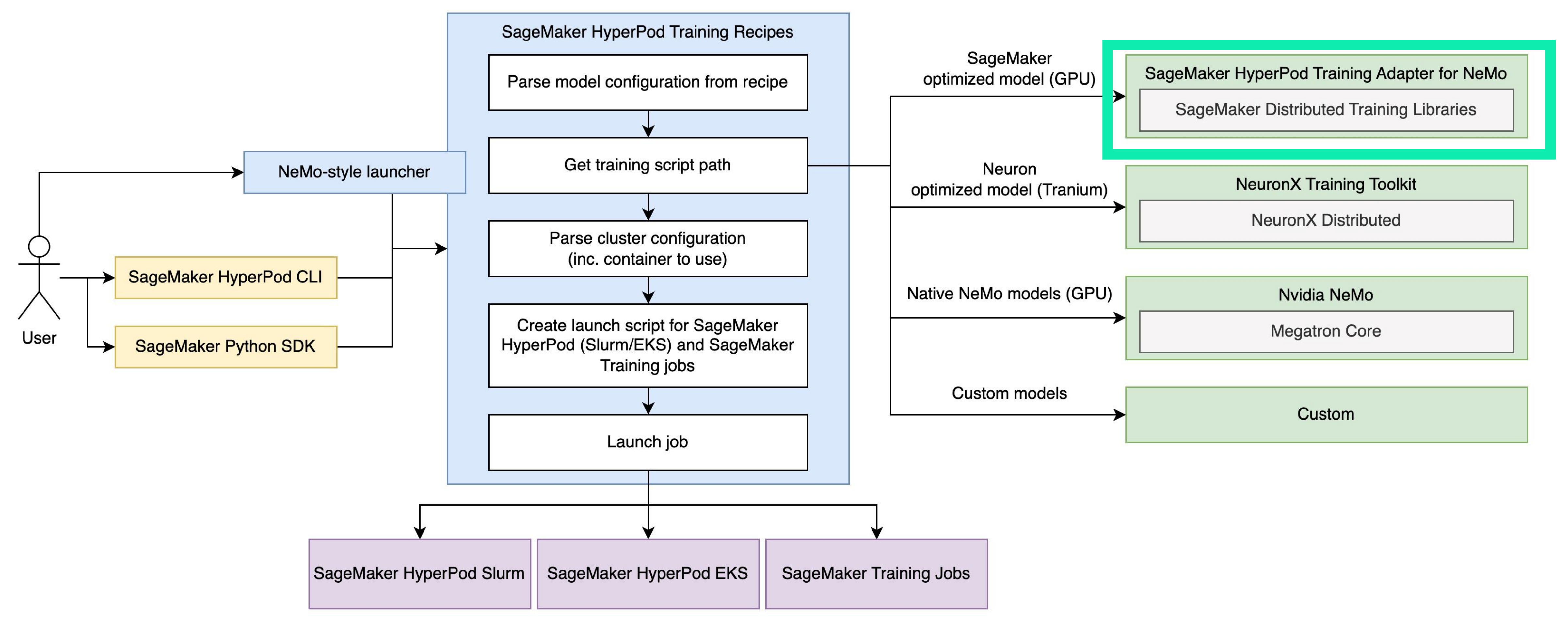

SageMaker HyperPod Training Recipes
Parse model configuration from recipe
Get training script path
Parse cluster configuration (inc. container to use)
Create launch script for SageMaker HyperPod (Slurm/EKS) and SageMaker Training jobs
Launch job
NeMo-style launcher
SageMaker HyperPod CLI
SageMaker Python SDK
User
SageMaker optimized model (GPU)
SageMaker HyperPod Training Adapter for NeMo
SageMaker Distributed Training Libraries
Neuron optimized model (Tranium)
NeuronX Training Toolkit
NeuronX Distributed
Native NeMo models (GPU)
Nvidia NeMo
Megatron Core
Custom models
Custom
SageMaker HyperPod Slurm
SageMaker HyperPod EKS
SageMaker Training Jobs

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

https://github.com/aws/sagemaker-hyperpod-recipes

Putting it all together

JOB SUBMISSION WORKFLOW

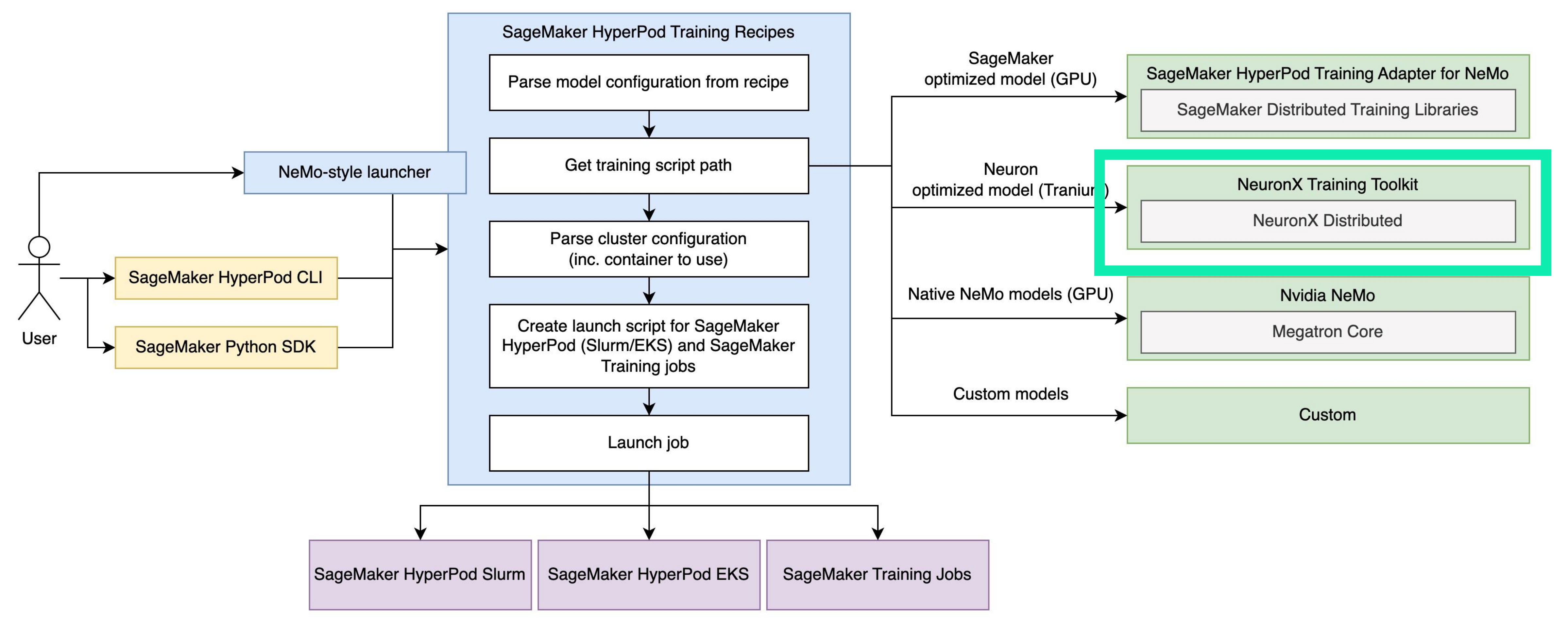

SageMaker HyperPod Training Recipes
Parse model configuration from recipe
Get training script path
Parse cluster configuration (inc. container to use)
Create launch script for SageMaker HyperPod (Slurm/EKS) and SageMaker Training jobs
Launch job
NeMo-style launcher
User
SageMaker HyperPod CLI
SageMaker Python SDK
SageMaker optimized model (GPU)
SageMaker HyperPod Training Adapter for NeMo
SageMaker Distributed Training Libraries
Neuron optimized model (Tranium)
NeuronX Training Toolkit
NeuronX Distributed
Native NeMo models (GPU)
Nvidia NeMo
Megatron Core
Custom models
Custom
SageMaker HyperPod Slurm
SageMaker HyperPod EKS
SageMaker Training Jobs

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

https://github.com/aws/sagemaker-hyperpod-recipes

Putting it all together

JOB SUBMISSION WORKFLOW



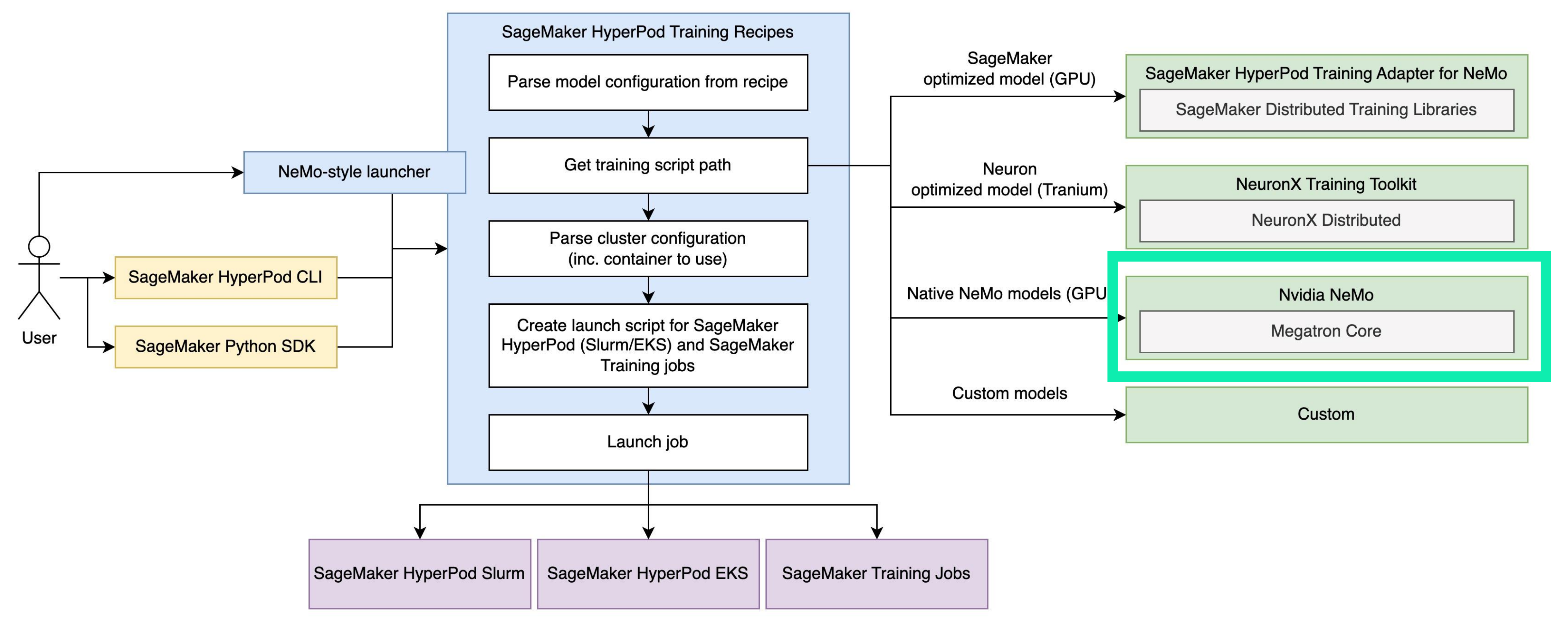
SageMaker HyperPod Training Recipes
Parse model configuration from recipe
Get training script path
Parse cluster configuration (inc. container to use)
Create launch script for SageMaker HyperPod (Slurm/EKS) and SageMaker Training jobs
Launch job
NeMo-style launcher
User
SageMaker HyperPod CLI
SageMaker Python SDK
SageMaker optimized model (GPU)
SageMaker HyperPod Training Adapter for NeMo
SageMaker Distributed Training Libraries
Neuron optimized model (Traniu)
NeuronX Training Toolkit
NeuronX Distributed
Native NeMo models (GPU)
Nvidia NeMo
Megatron Core
Custom models
Custom
SageMaker HyperPod Slurm
SageMaker HyperPod EKS
SageMaker Training Jobs

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

https://github.com/aws/sagemaker-hyperpod-recipes

Putting it all together

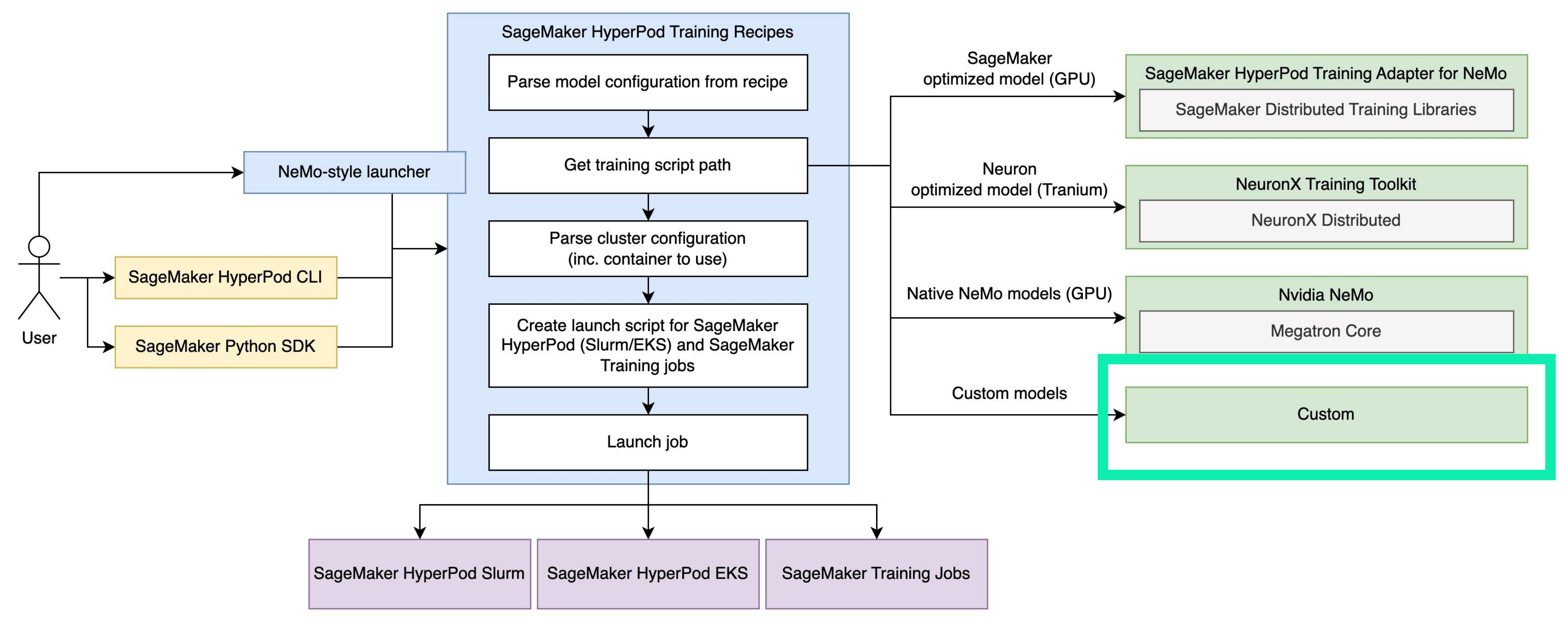
JOB SUBMISSION WORKFLOW


User
NeMo-style launcher
SageMaker HyperPod CLI
SageMaker Python SDK
SageMaker HyperPod Training Recipes
Parse model configuration from recipe
Get training script path
Parse cluster configuration (inc. container to use)
Create launch script for SageMaker HyperPod (Slurm/EKS) and SageMaker Training jobs
Launch job
SageMaker optimized model (GPU)
SageMaker HyperPod Training Adapter for NeMo
SageMaker Distributed Training Libraries
Neuron optimized model (Tranium)
NeuronX Training Toolkit
NeuronX Distributed
Native NeMo models (GPU
Nvidia NeMo
Megatron Core
Custom models
Custom
SageMaker HyperPod Slurm
SageMaker HyperPod EKS
SageMaker Training Jobs

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Putting it all together

JOB SUBMISSION WORKFLOW



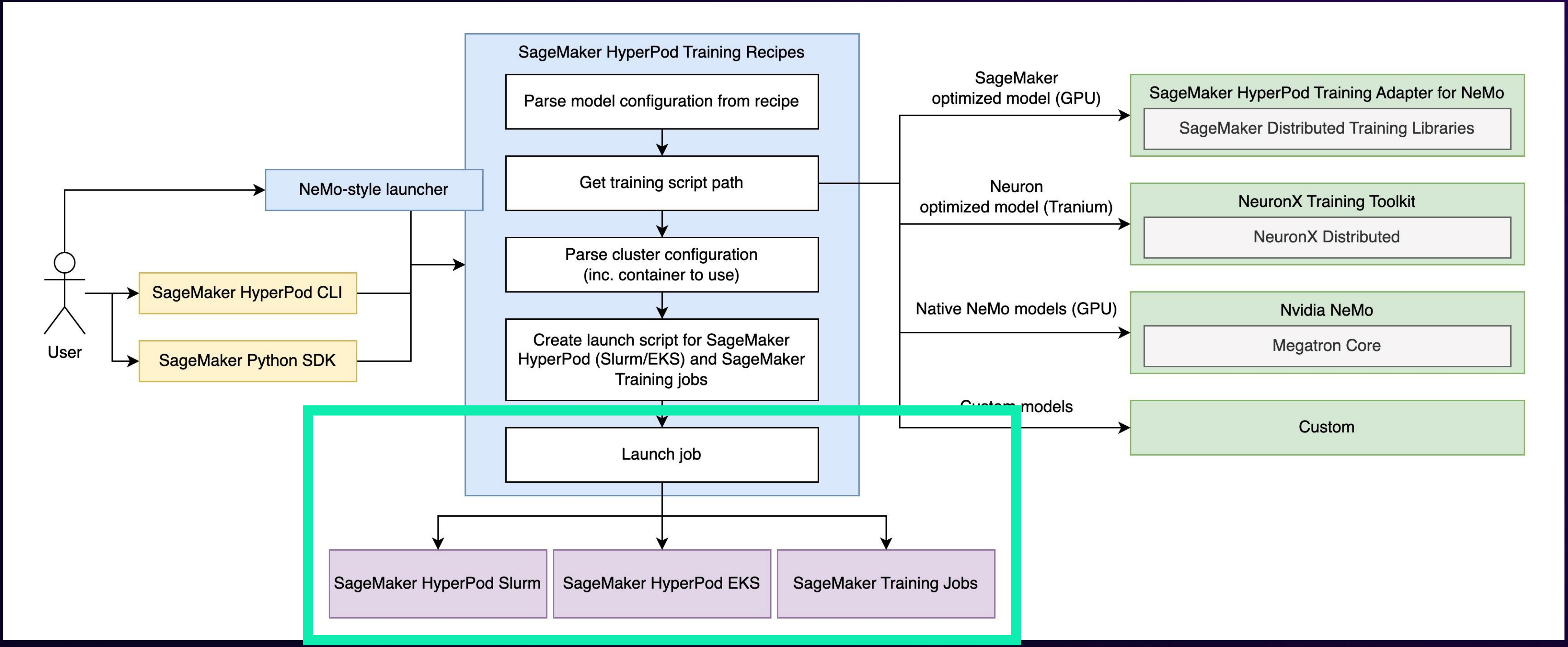
User
NeMo-style launcher
SageMaker HyperPod CLI
SageMaker Python SDK
SageMaker HyperPod Training Recipes
Parse model configuration from recipe
Get training script path
Parse cluster configuration (inc. container to use)
Create launch script for SageMaker HyperPod (Slurm/EKS) and SageMaker Training jobs
Launch job
SageMaker optimized model (GPU)
SageMaker HyperPod Training Adapter for NeMo
SageMaker Distributed Training Libraries
Neuron optimized model (Tranium)
NeuronX Training Toolkit
NeuronX Distributed
Native NeMo models (GPU)
Nvidia NeMo
Megatron Core
Custom models
Custom
SageMaker HyperPod Slurm
SageMaker HyperPod EKS
SageMaker Training Jobs

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

https://github.com/aws/sagemaker-hyperpod-recipes

Putting it all together

JOB SUBMISSION WORKFLOW



User

NeMo-style launcher

SageMaker HyperPod CLI

SageMaker Python SDK

SageMaker HyperPod Training Recipes

Parse model configuration from recipe

Get training script path

Parse cluster configuration
(inc. container to use)

Create launch script for SageMaker HyperPod (Slurm/EKS) and SageMaker Training jobs

Launch job

SageMaker HyperPod Slurm

SageMaker HyperPod EKS

SageMaker Training Jobs

SageMaker
optimized model (GPU)

SageMaker HyperPod Training Adapter for NeMo

SageMaker Distributed Training Libraries

Neuron
optimized model (Tranium)

NeuronX Training Toolkit

NeuronX Distributed

Native NeMo models (GPU)

Nvidia NeMo

Megatron Core

Custom models

Custom

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

https://github.com/aws/sagemaker-hyperpod-recipes

Recap

AMAZON SAGEMAKER HYPERPOD TRAINING RECIPES REPOSITORY

Over 30 recipes with different configurations

Run training with a single line of code

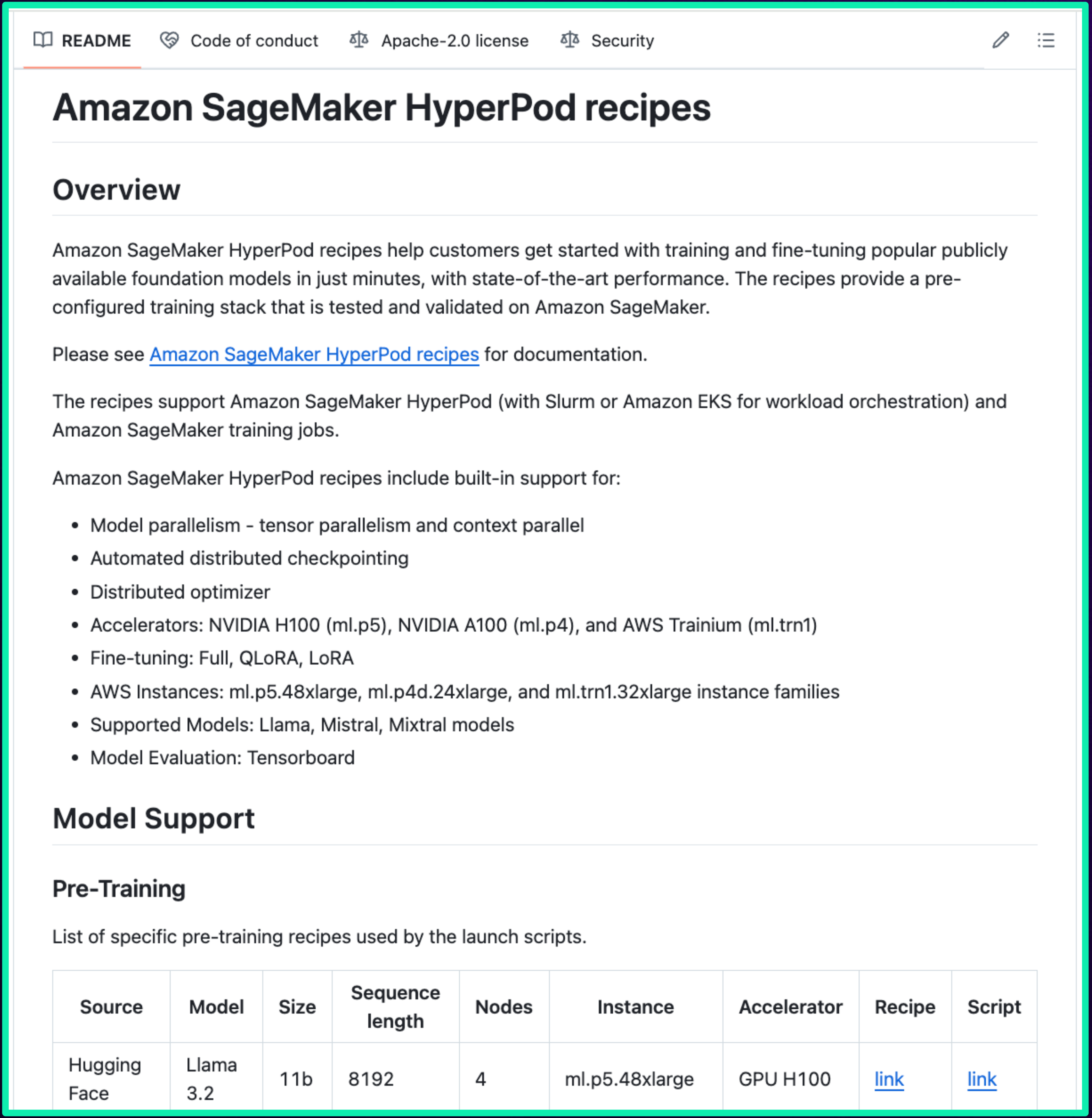
Recipes support SageMaker HyperPod and training jobs

SageMaker optimized models (GPU)

AWS Neuron optimized models (Trainium)

Native NeMo models

Custom models



Amazon SageMaker HyperPod recipes

Overview

Amazon SageMaker HyperPod recipes help customers get started with training and fine-tuning popular publicly available foundation models in just minutes, with state-of-the-art performance. The recipes provide a pre-configured training stack that is tested and validated on Amazon SageMaker.

Please see Amazon SageMaker HyperPod recipes for documentation.

The recipes support Amazon SageMaker HyperPod (with Slurm or Amazon EKS for workload orchestration) and Amazon SageMaker training jobs.

Amazon SageMaker HyperPod recipes include built-in support for:

- Model parallelism - tensor parallelism and context parallel
- Automated distributed checkpointing
- Distributed optimizer
- Accelerators: NVIDIA H100 (ml.p5), NVIDIA A100 (ml.p4), and AWS Trainium (ml.trn1)
- Fine-tuning: Full, QLoRA, LoRA
- AWS Instances: ml.p5.48xlarge, ml.p4d.24xlarge, and ml.trn1.32xlarge instance families
- Supported Models: Llama, Mistral, Mixtral models
- Model Evaluation: Tensorboard

Model Support

Pre-Training

List of specific pre-training recipes used by the launch scripts.

Source	Model	Size	Sequence length	Nodes	Instance	Accelerator	Recipe	Script
Hugging Face	Llama 3.2	11b	8192	4	ml.p5.48xlarge	GPU H100	link	link

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Salesforce AI research

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

A little bit about Salesforce AI Research …

We are Salesforce’s AI Research org

- **Foundational R&D**
  Pushing forward state-of-the-art models for enterprise AI
- **Customer incubation**
  Customer-centric pathfinding with cutting-edge AI for high-value use cases
- **Product innovation**
  Incorporating AI models and technologies into generally available features

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

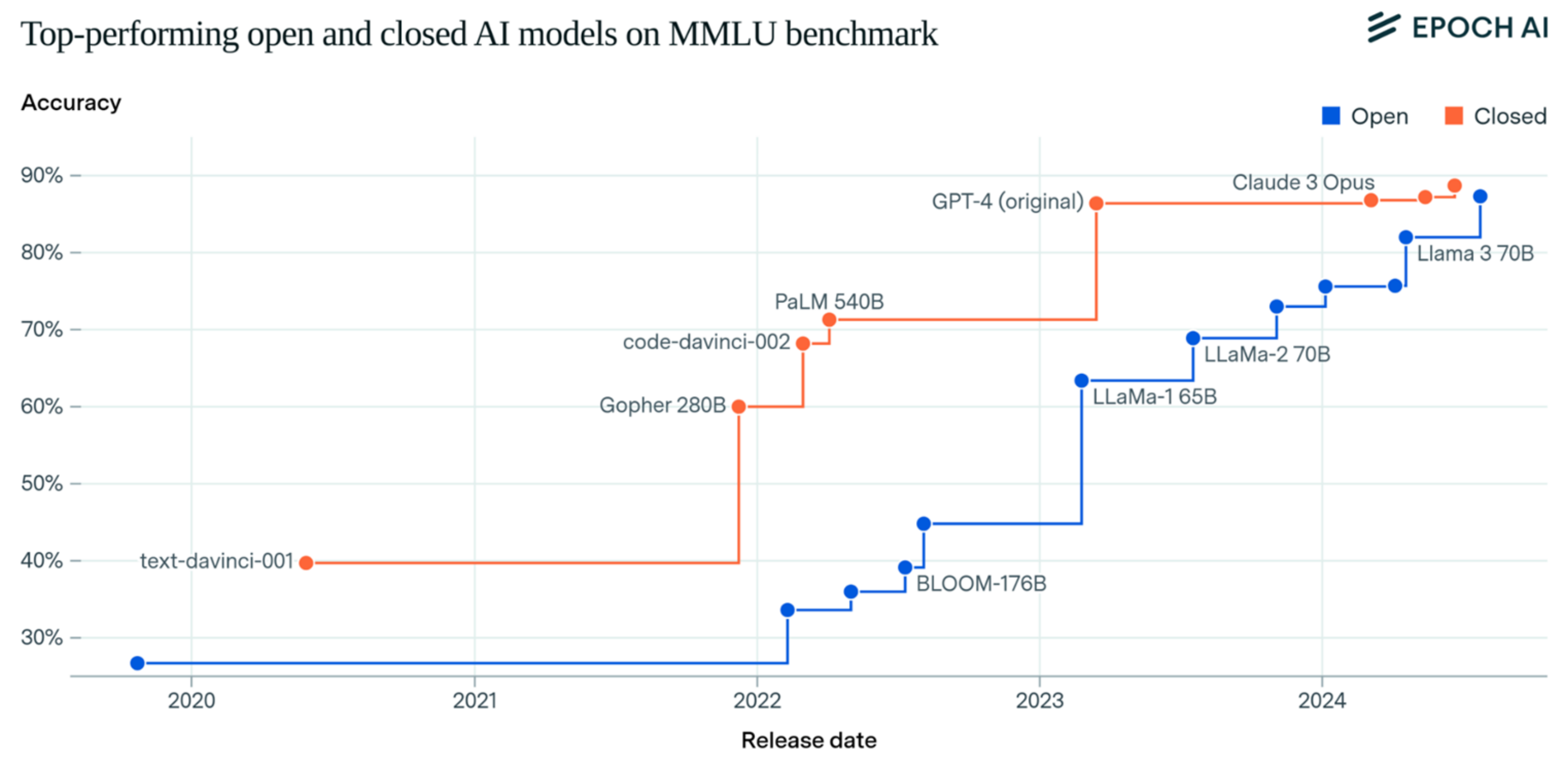
Salesforce AI & Amazon SageMaker HyperPod

We're big users of SageMaker for years

This year, we've leveraged HyperPod

"Supercomputer at our fingertips"

Why do model training and fine-tuning matter?



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Salesforce AI

Fine-tuning FMs with HyperPod recipes

SageMaker HyperPod recipes have great features that saved us ***significant*** infra work

FSDP + Context Parallelism + Low-Rank Adaptation (LoRA)
Unique Feature Set

Fine-tune Llama-3.1-405B at full 128K context with 4 nodes
1 node with Q-LoRA

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Production models at Salesforce trained on HyperPod

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

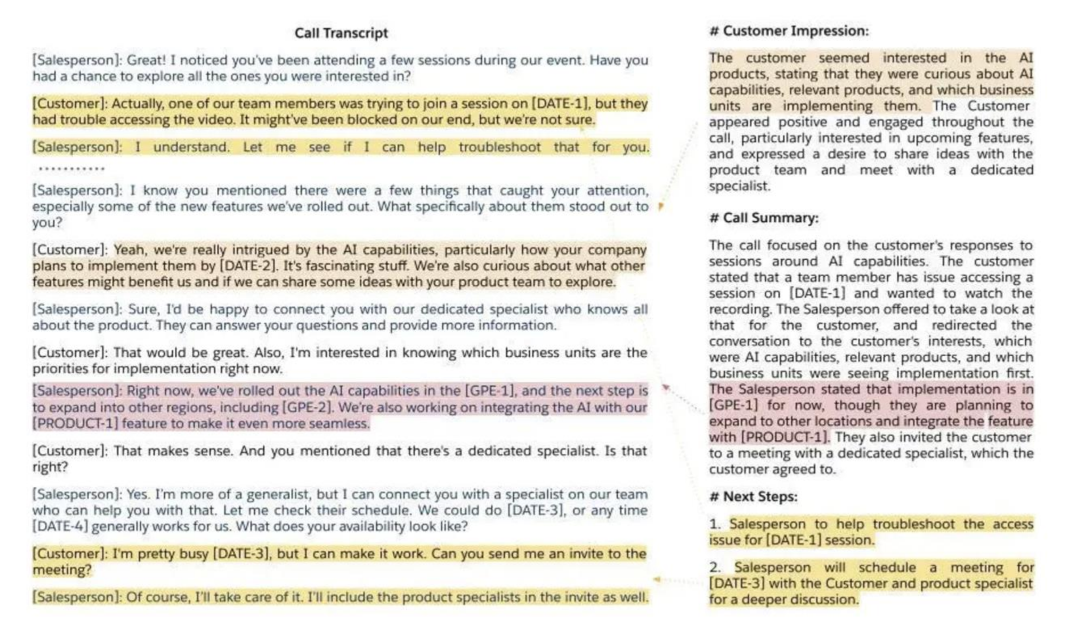
Agentforce: Powering next-gen AI for Sales with xGen-Sales LLM

First proprietary sales-focused large language model (LLM)

Key features: call summarization, customer profiling, contact enrichment, and pipeline tracking

Users can interactively engage with AI-generated call summary

AI-generated summaries favored >50% of the time against human-generated



Call Transcript

[Salesperson]: Great! I noticed you've been attending a few sessions during our event. Have you had a chance to explore all the ones you were interested in?

[Customer]: Actually, one of our team members was trying to join a session on [DATE-1], but they had trouble accessing the video. It might've been blocked on our end, but we're not sure.

[Salesperson]: I understand. Let me see if I can help troubleshoot that for you.

....

[Salesperson]: I know you mentioned there were a few things that caught your attention, especially some of the new features we've rolled out. What specifically about them stood out to you?

[Customer]: Yeah, we're really intrigued by the AI capabilities, particularly how your company plans to implement them by [DATE-2]. It's fascinating stuff. We're also curious about what other features might benefit us and if we can share some ideas with your product team to explore.

[Salesperson]: Sure, I'd be happy to connect you with our dedicated specialist who knows all about the product. They can answer your questions and provide more information.

[Customer]: That would be great. Also, I'm interested in knowing which business units are the priorities for implementation right now.

[Salesperson]: Right now, we've rolled out the AI capabilities in the [GPE-1], and the next step is to expand into other regions, including [GPE-2]. We're also working on integrating the AI with our [PRODUCT-1] feature to make it even more seamless.

[Customer]: That makes sense. And you mentioned that there's a dedicated specialist. Is that right?

[Salesperson]: Yes. I'm more of a generalist, but I can connect you with a specialist on our team who can help you with that. Let me check their schedule. We could do [DATE-3], or any time [DATE-4] generally works for us. What does your availability look like?

[Customer]: I'm pretty busy [DATE-3], but I can make it work. Can you send me an invite to the meeting?

[Salesperson]: Of course, I'll take care of it. I'll include the product specialists in the invite as well.

# Customer Impression:

The customer seemed interested in the AI products, stating that they were curious about AI capabilities, relevant products, and which business units are implementing them. The Customer appeared positive and engaged throughout the call, particularly interested in upcoming features, and expressed a desire to share ideas with the product team and meet with a dedicated specialist.

# Call Summary:

The call focused on the customer's responses to sessions around AI capabilities. The customer stated that a team member has issue accessing a session on [DATE-1] and wanted to watch the recording. The Salesperson offered to take a look at that for the customer, and redirected the conversation to the customer's interests, which were AI capabilities, relevant products, and which business units were seeing implementation first. The Salesperson stated that implementation is in [GPE-1] for now, though they are planning to expand to other locations and integrate the feature with [PRODUCT-1]. They also invited the customer to a meeting with a dedicated specialist, which the customer agreed to.

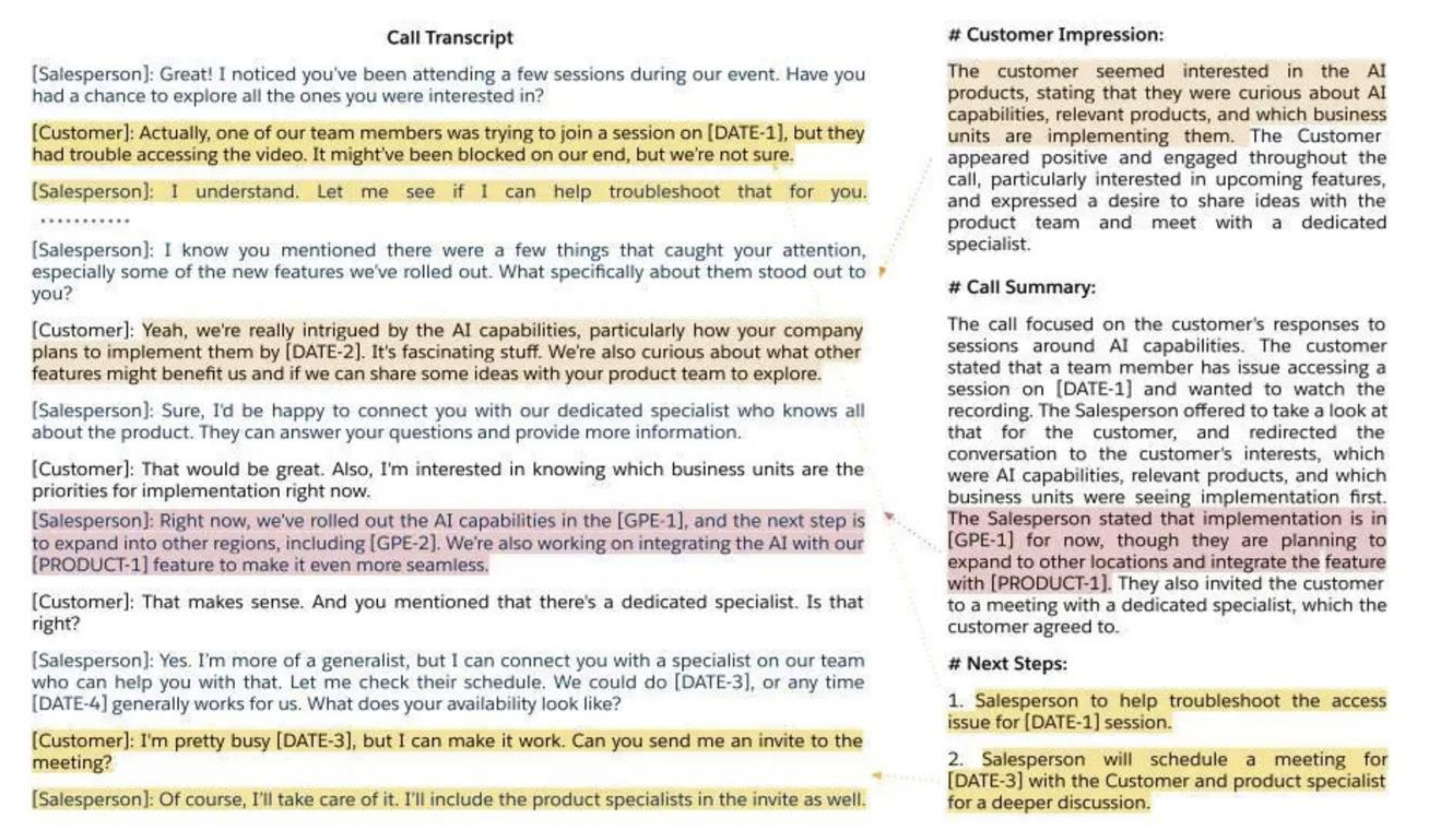
# Next Steps:

1. Salesperson to help troubleshoot the access issue for [DATE-1] session.
2. Salesperson will schedule a meeting for [DATE-3] with the Customer and product specialist for a deeper discussion.

Contributors:
Semih Yavuz, Xinyi Yang, Srijan Bansal, Donna Tran, John Emmons,
Jason Lee , Erik Nijkamp, Bo Pang, Egor Pakhomov,
Akash Gokul, Antonio Ginart, Yingbo Zho

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agentforce: Powering next-gen AI for Sales with xGen-Sales LLM



Call Transcript

[Salesperson]: Great! I noticed you've been attending a few sessions during our event. Have you had a chance to explore all the ones you were interested in?

[Customer]: Actually, one of our team members was trying to join a session on [DATE-1], but they had trouble accessing the video. It might've been blocked on our end, but we're not sure.

[Salesperson]: I understand. Let me see if I can help troubleshoot that for you.

....

[Salesperson]: I know you mentioned there were a few things that caught your attention, especially some of the new features we've rolled out. What specifically about them stood out to you?

[Customer]: Yeah, we're really intrigued by the AI capabilities, particularly how your company plans to implement them by [DATE-2]. It's fascinating stuff. We're also curious about what other features might benefit us and if we can share some ideas with your product team to explore.

[Salesperson]: Sure, I'd be happy to connect you with our dedicated specialist who knows all about the product. They can answer your questions and provide more information.

[Customer]: That would be great. Also, I'm interested in knowing which business units are the priorities for implementation right now.

[Salesperson]: Right now, we've rolled out the AI capabilities in the [GPE-1], and the next step is to expand into other regions, including [GPE-2]. We're also working on integrating the AI with our [PRODUCT-1] feature to make it even more seamless.

[Customer]: That makes sense. And you mentioned that there's a dedicated specialist. Is that right?

[Salesperson]: Yes. I'm more of a generalist, but I can connect you with a specialist on our team who can help you with that. Let me check their schedule. We could do [DATE-3], or any time [DATE-4] generally works for us. What does your availability look like?

[Customer]: I'm pretty busy [DATE-3], but I can make it work. Can you send me an invite to the meeting?

[Salesperson]: Of course, I'll take care of it. I'll include the product specialists in the invite as well.

Customer Impression:

The customer seemed interested in the AI products, stating that they were curious about AI capabilities, relevant products, and which business units are implementing them. The Customer appeared positive and engaged throughout the call, particularly interested in upcoming features, and expressed a desire to share ideas with the product team and meet with a dedicated specialist.

Call Summary:

The call focused on the customer's responses to sessions around AI capabilities. The customer stated that a team member has issue accessing a session on [DATE-1] and wanted to watch the recording. The Salesperson offered to take a look at that for the customer, and redirected the conversation to the customer's interests, which were AI capabilities, relevant products, and which business units were seeing implementation first. The Salesperson stated that implementation is in [GPE-1] for now, though they are planning to expand to other locations and integrate the feature with [PRODUCT-1]. They also invited the customer to a meeting with a dedicated specialist, which the customer agreed to.

Next Steps:

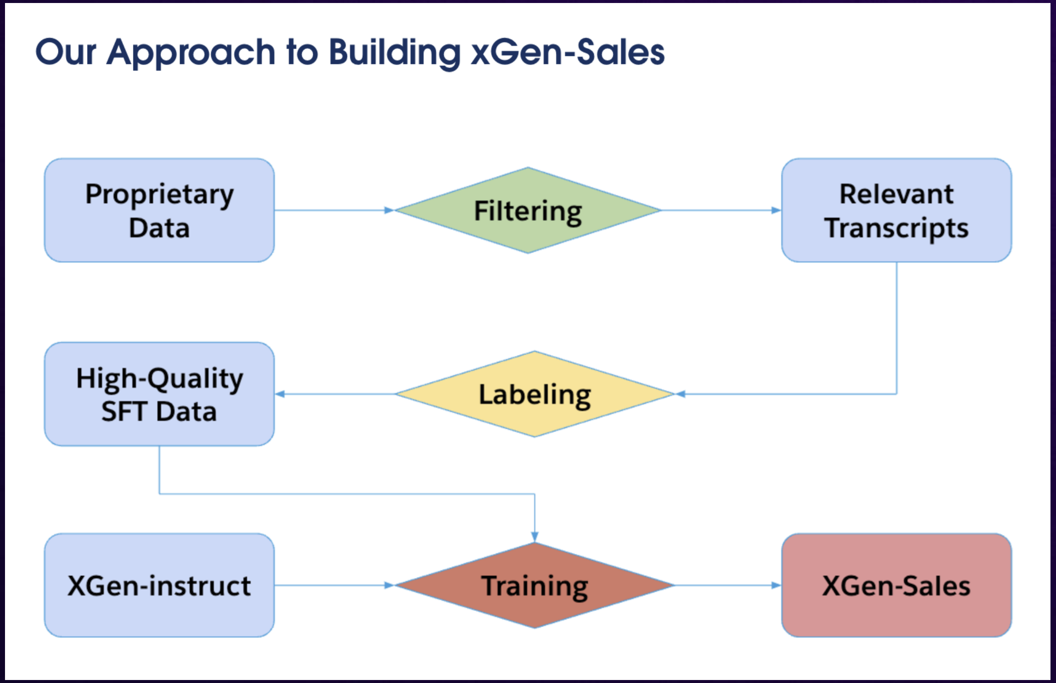
1. Salesperson to help troubleshoot the access issue for [DATE-1] session.
2. Salesperson will schedule a meeting for [DATE-3] with the Customer and product specialist for a deeper discussion.

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agentforce: Powering next-gen AI for Sales with xGen-Sales LLM

Developed using human-in-the-loop reinforcement learning and diverse sales datasets

Planned integration with Agentforce, live customer pilots are already underway!



Our Approach to Building xGen-Sales

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Ragforce: Building *contextually faithful* LLMs with SFR-RAG

Retrieval Augmented Generation: Combine semantic search + in-context learning + generative LLM

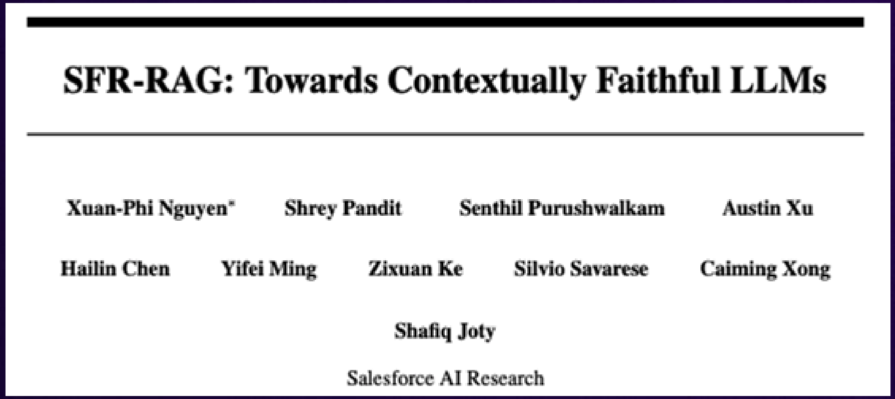
- Augments LLMs with enterprise data retrieval, but quality depends heavily on document hygiene
  - Conflicting versions, outdated info, and inconsistent formatting can confuse the model

- How should models handle prioritization of information given rich metadata?
  - Prioritization semantics based on sources, tags, and dates
  - “Official” tag takes priority over “Draft” tag unless date is <6 months old

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

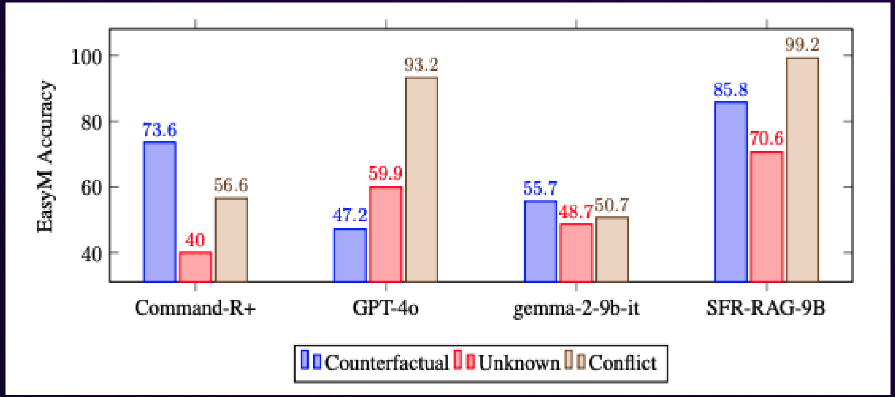
Ragforce: Building *contextually faithful* LLMs with SFR-RAG

- Trained a contextual LLM (9B) for RAG generation
  - Faithful: Trained to minimize hallucinations, even in adverse settings (conflicting info) & precisely follow complex prioritization rules
  - Lightweight and high-performing: State-of-the-art aggregate performance with 9B params, beating larger models (104B+)
  - Trained on HyperPod using 2x p5.48xlarge nodes



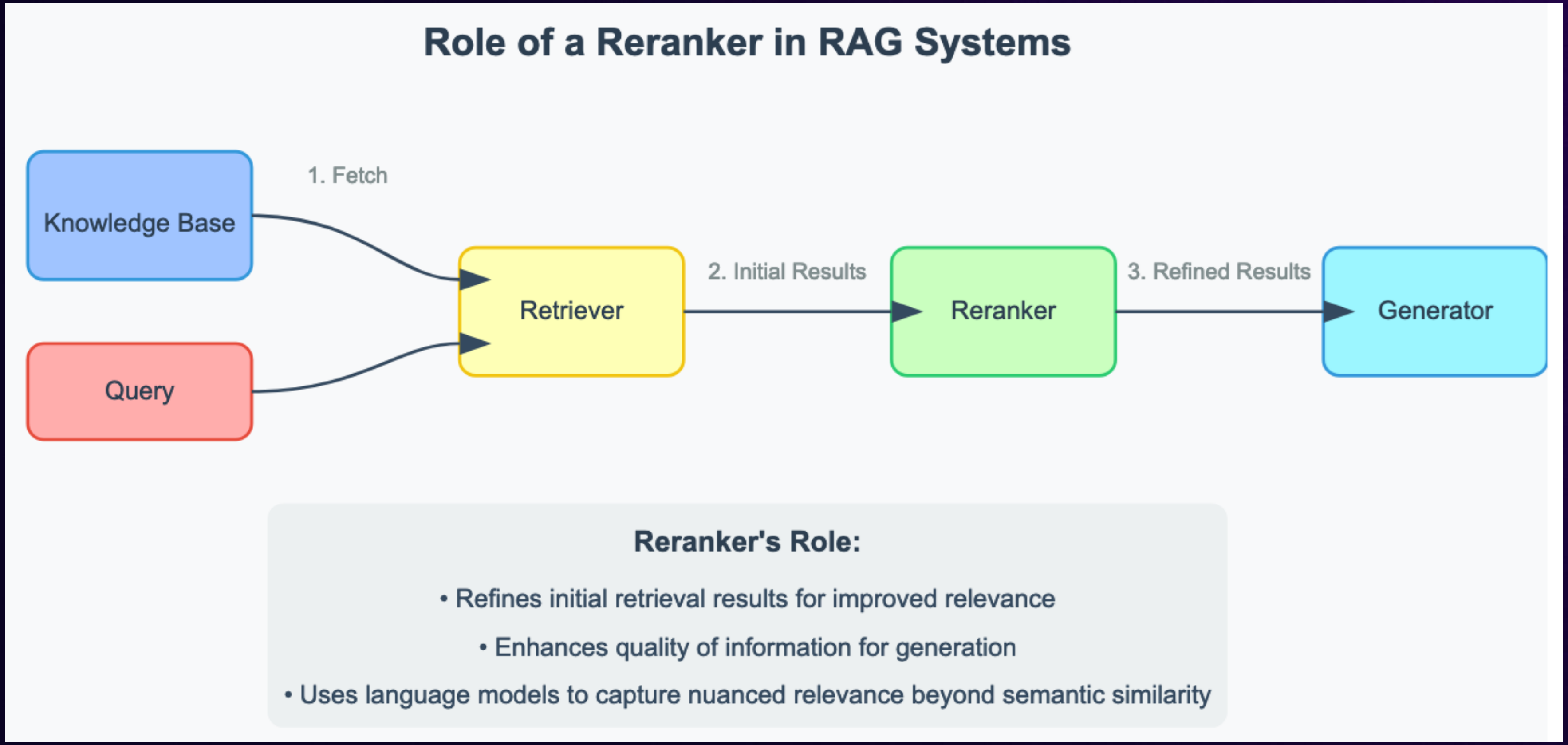
SFR-RAG: Towards Contextually Faithful LLMs

Xuan-Phi Nguyen* **Shrey Pandit** **Senthil Purushwalkam** **Austin Xu**
Hailin Chen **Yifei Ming** **Zixuan Ke** **Silvio Savarese** **Caiming Xong**
Shafiq Joty
Salesforce AI Research



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Rankforce: State-of-the-art enterprise reranking with SFR-LlamaRank



Reranker's Role:

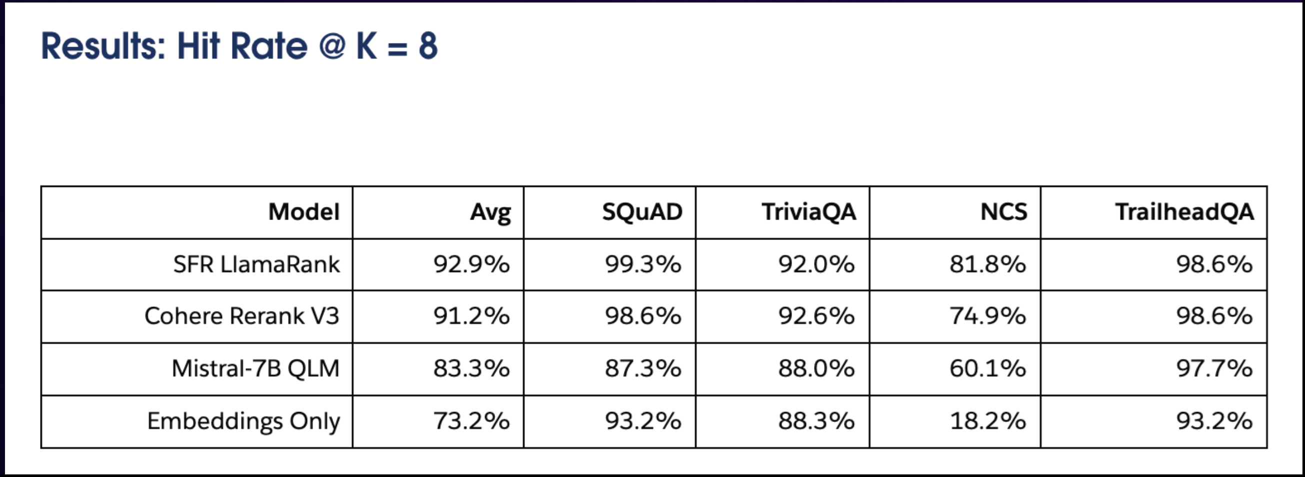
- Refines initial retrieval results for improved relevance
- Enhances quality of information for generation
- Uses language models to capture nuanced relevance beyond semantic similarity

Lead contributors: A. Ginart, N. Kodali, J. Emmons

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Rankforce: State-of-the-art enterprise reranking with SFR-LlamaRank

- **Core technical specs:**
  - Built on Llama3-8B-Instruct with human-guided training (4x p5.48xlarge)
  - Supports 8K document chunks with fast inference (<200ms/4 docs on H100)
- Superior code search performance
- Linear, calibrated scoring
  (0.9+ highly relevant to ~0.0 irrelevant)
- Horizontally scalable for enterprise deployment



Results: Hit Rate @ K = 8

Model	Avg	SQuAD	TriviaQA	NCS	TrailheadQA
SFR LlamaRank	92.9%	99.3%	92.0%	81.8%	98.6%
Cohere Rerank V3	91.2%	98.6%	92.6%	74.9%	98.6%
Mistral-7B QLM	83.3%	87.3%	88.0%	60.1%	97.7%
Embeddings Only	73.2%	93.2%	88.3%	18.2%	93.2%

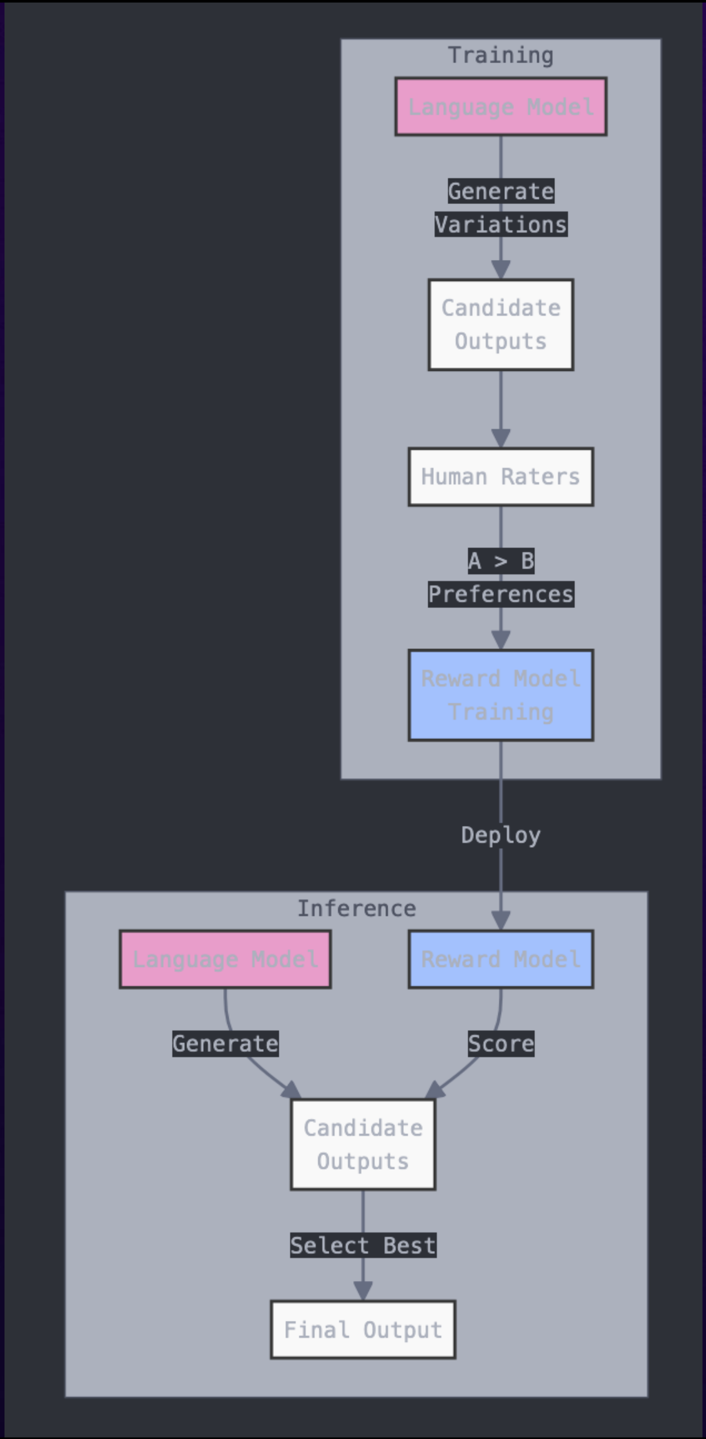
Lead contributors: A. Ginart, N. Kodali, J. Emmons

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Judgeforce

What is a reward model?

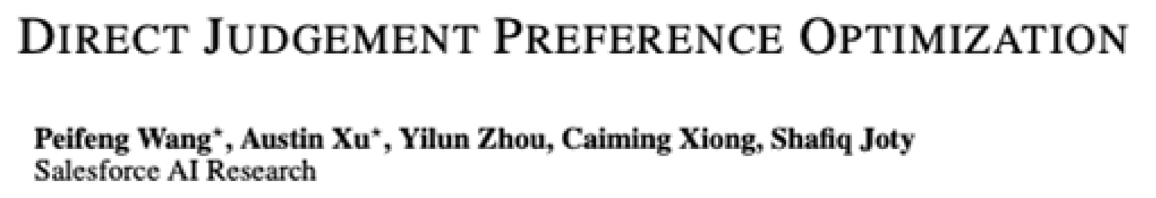
- Learn a model that predicts human preferences based on pairwise comparisons ("A is better than B")
- Used in training models (RLHF) as well as automatic evaluations and even inference



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

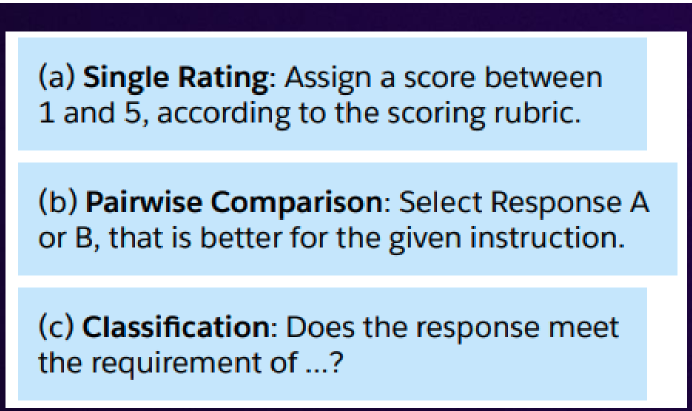
Judgeforce: Automating LLM eval with SFR-Judge

- Trained 3 LLM-as-judge models (8B, 12B, 70B) for *automatic evaluation*
  - Multifaceted: Trained to evaluate via pairwise comparisons, single rating, and classification
  - State-of-the-art aggregate performance across 13 evaluation benchmarks
  - Trained 70B on HyperPod using 4x p5.48xlarge nodes



DIRECT JUDGEMENT PREFERENCE OPTIMIZATION

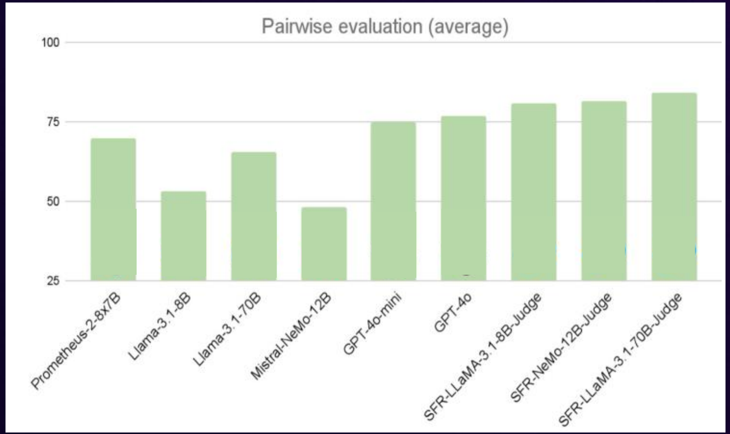
Peifeng Wang*, Austin Xu*, Yilun Zhou, Caiming Xiong, Shafiq Joty
Salesforce AI Research



(a) **Single Rating**: Assign a score between 1 and 5, according to the scoring rubric.

(b) **Pairwise Comparison**: Select Response A or B, that is better for the given instruction.

(c) **Classification**: Does the response meet the requirement of ...?



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What’s next at Salesforce AI?

Are we hitting entropy rate of internet corpus?
Can we scale up post-training dataset sizes?

What will inference-time compute scaling be?
2025 may be the year of inference-time scaling!

Multi-Agent Systems:

- **Components:** Specialized models for tasks – Re-rankers, reward models
  xGen-Sales
- **Assistants:** Dispatch directly with humans – real-time, multi-modal, low-latency
  SFR-RAG
- **Agents:** Long-running AI with tools - Minutes, hours, or even days
  - Powered by reasoning-style LLMs

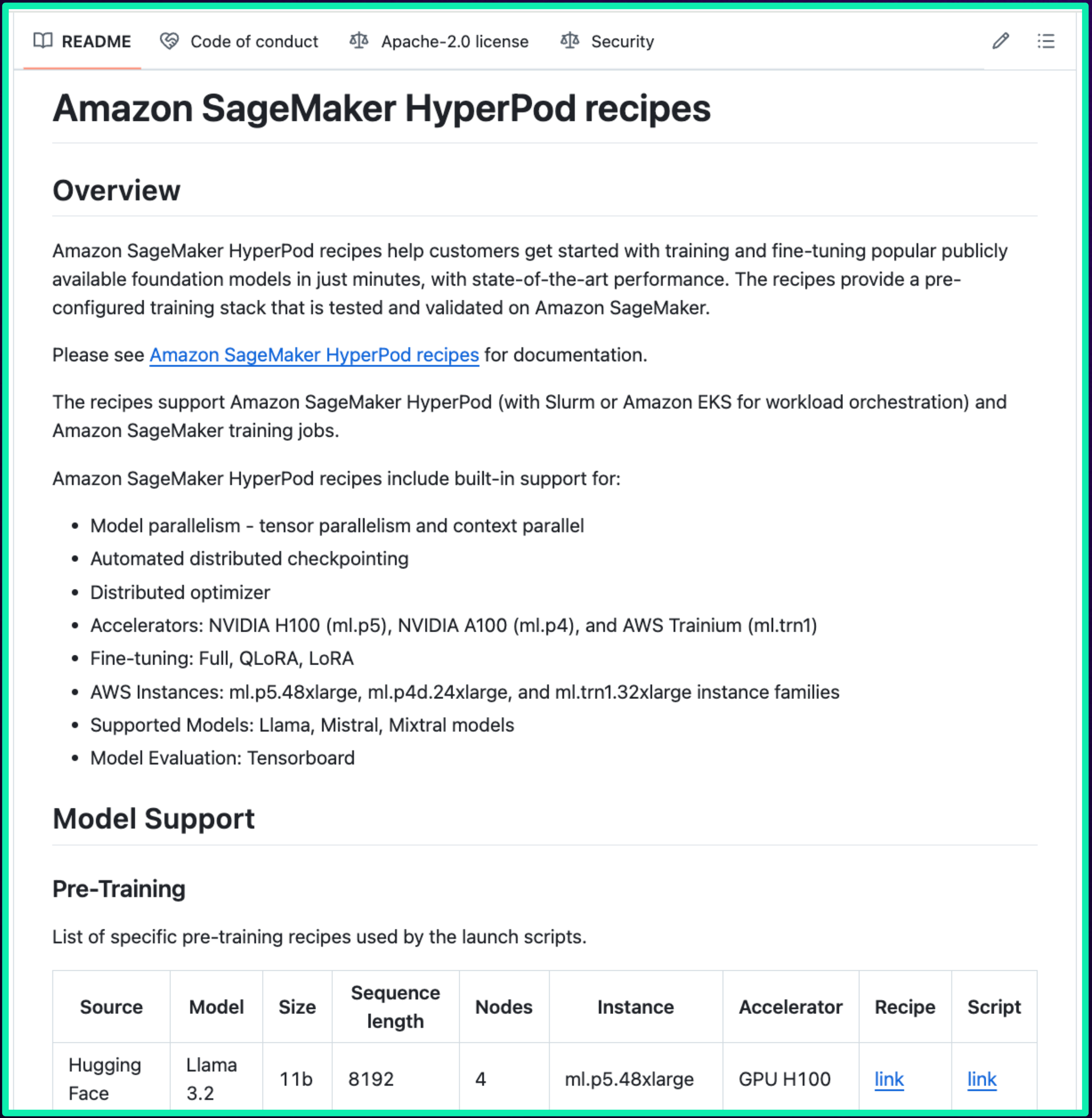
© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Get started with HyperPod recipes

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AMAZON SAGEMAKER HYPERPOD RECIPES

https://github.com/aws/sagemaker-hyperpod-recipes



Amazon SageMaker HyperPod recipes

Overview

Amazon SageMaker HyperPod recipes help customers get started with training and fine-tuning popular publicly available foundation models in just minutes, with state-of-the-art performance. The recipes provide a pre-configured training stack that is tested and validated on Amazon SageMaker.

Please see Amazon SageMaker HyperPod recipes for documentation.

The recipes support Amazon SageMaker HyperPod (with Slurm or Amazon EKS for workload orchestration) and Amazon SageMaker training jobs.

Amazon SageMaker HyperPod recipes include built-in support for:

- Model parallelism - tensor parallelism and context parallel
- Automated distributed checkpointing
- Distributed optimizer
- Accelerators: NVIDIA H100 (ml.p5), NVIDIA A100 (ml.p4), and AWS Trainium (ml.trn1)
- Fine-tuning: Full, QLoRA, LoRA
- AWS Instances: ml.p5.48xlarge, ml.p4d.24xlarge, and ml.trn1.32xlarge instance families
- Supported Models: Llama, Mistral, Mixtral models
- Model Evaluation: Tensorboard

Model Support

Pre-Training

List of specific pre-training recipes used by the launch scripts.

Source	Model	Size	Sequence length	Nodes	Instance	Accelerator	Recipe	Script
Hugging Face	Llama 3.2	11b	8192	4	ml.p5.48xlarge	GPU H100	link	link

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Please complete the session survey in the mobile app

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.