AWS re:Invent

DECEMBER 2 – 6, 2024 | LAS VEGAS, NV

AIM369

Optimize cost performance using Amazon SageMaker inference

Rishabh Ray Chaudhury
(he/him)
Senior Product Manager, AI
AWS

Raghu Ramesha
(he/him)
Senior GenAI/ML Specialist SA
AWS

Agenda

- Customer challenges with gen AI models for inference
- Amazon SageMaker inference optimization toolkit
- Performance improvements
- Inference optimization techniques on Amazon SageMaker
- How does it work?
- Demo

Customer challenges with gen AI models for inference

Gen AI Models

Billions of parameters

- Complexity with inference optimization
- Increased time-to-market
- Developer time and resource costs

Amazon SageMaker inference optimization toolkit

1. Fully managed toolkit

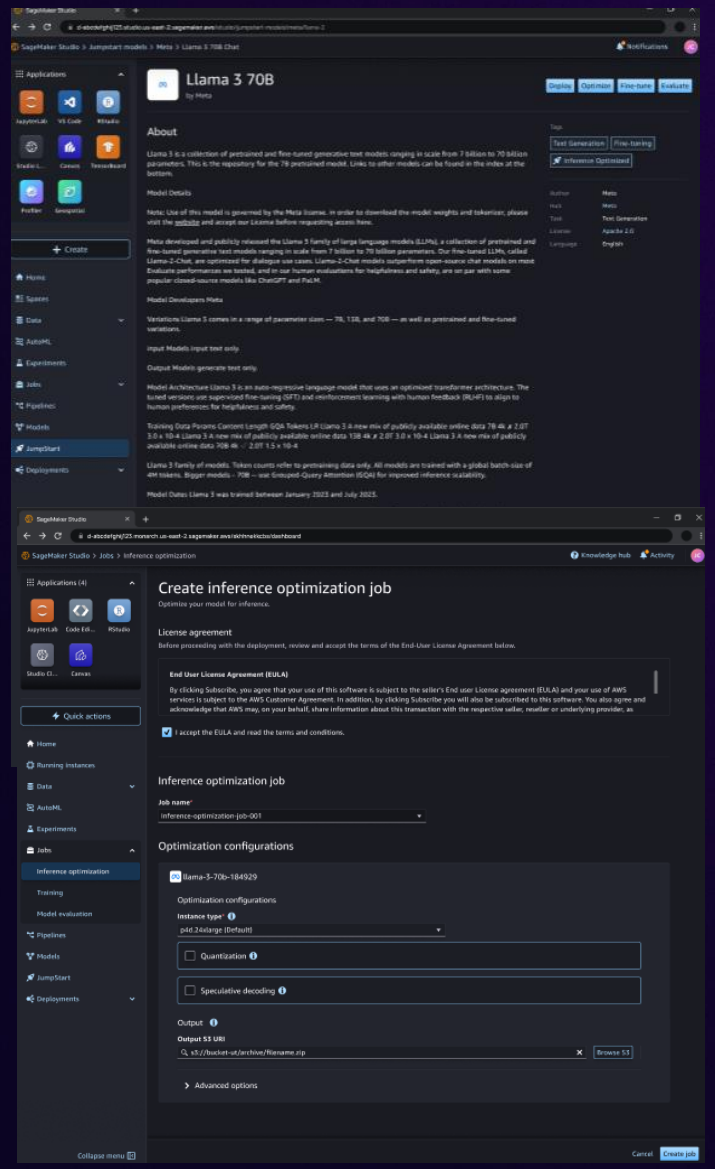
Out-of-the-box support for latest optimization techniques - speculative decoding, compilation, and quantization

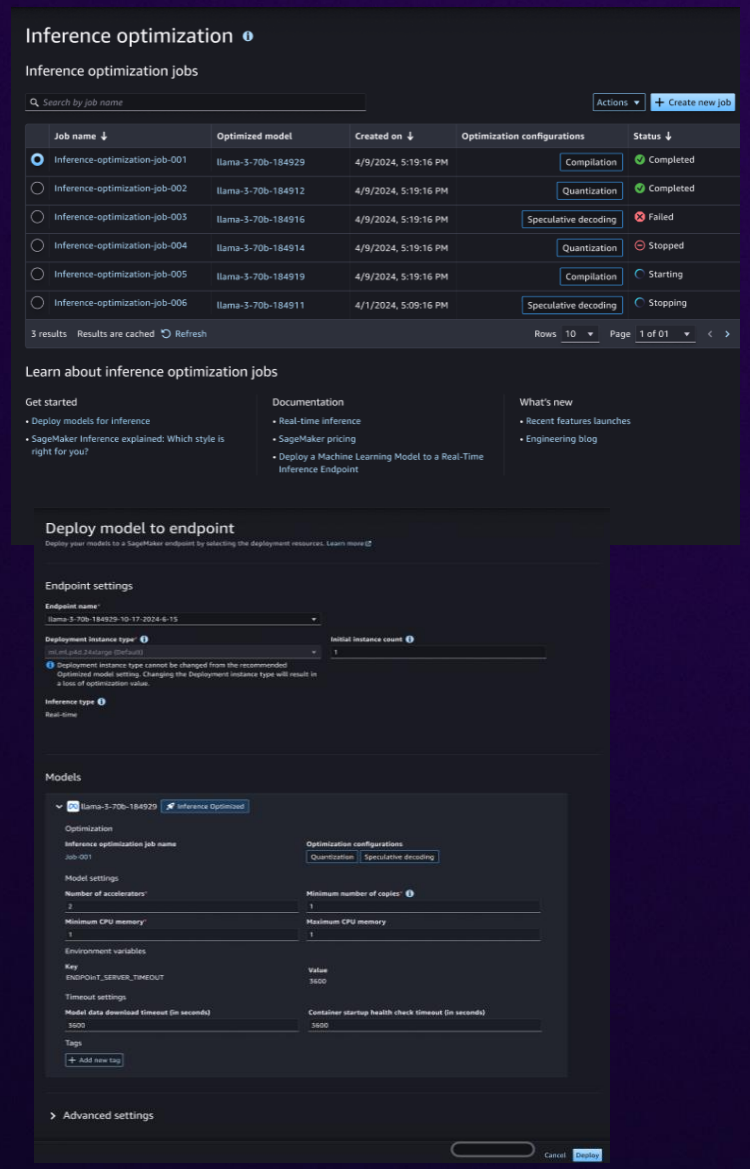
2. Reduced time-to-market

Optimize gen AI models for best price-performance in hours, compared to months earlier

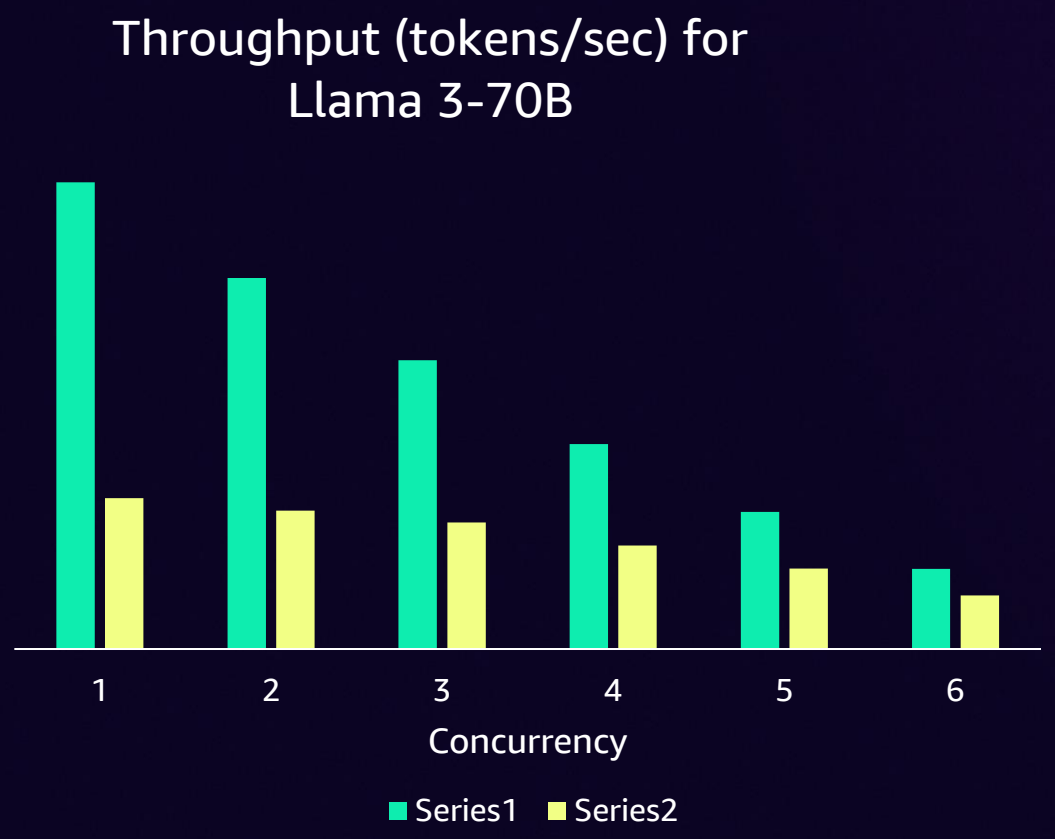
3. Increase throughput by up to 2x and reduce costs by up to 50%

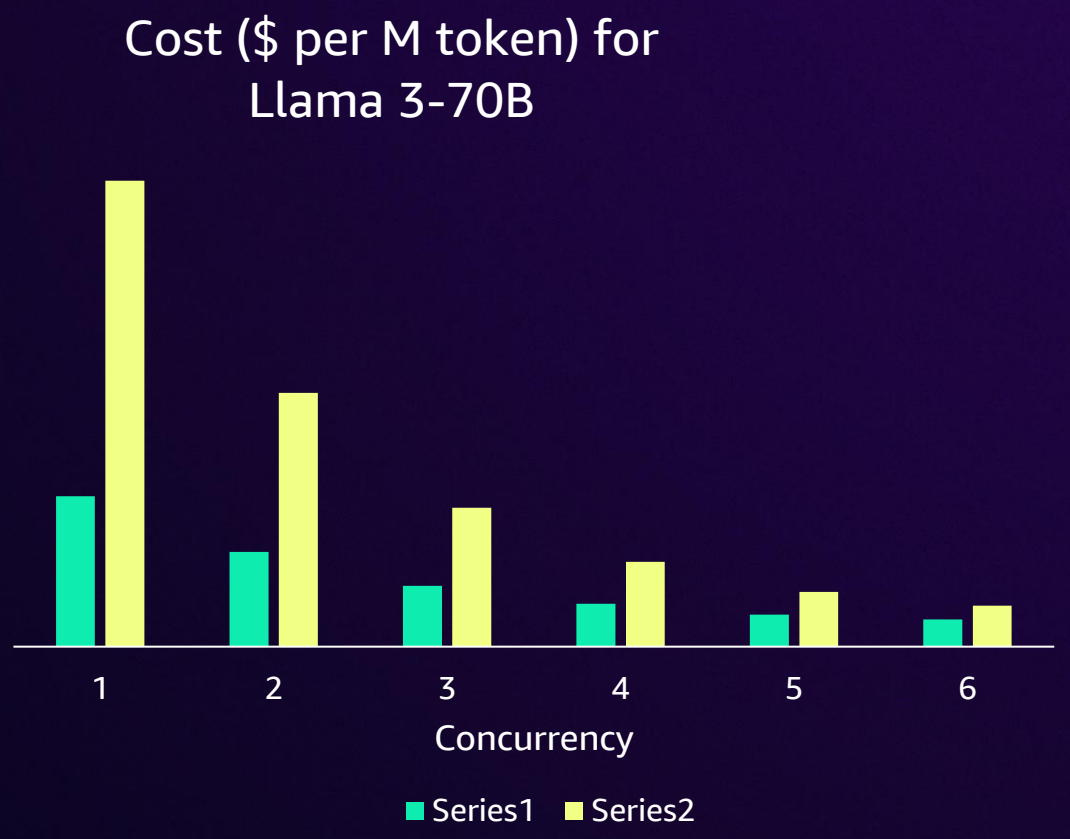
Improve cost/performance for a variety of gen AI models including Llama 3, Mistral, and Mixtral models





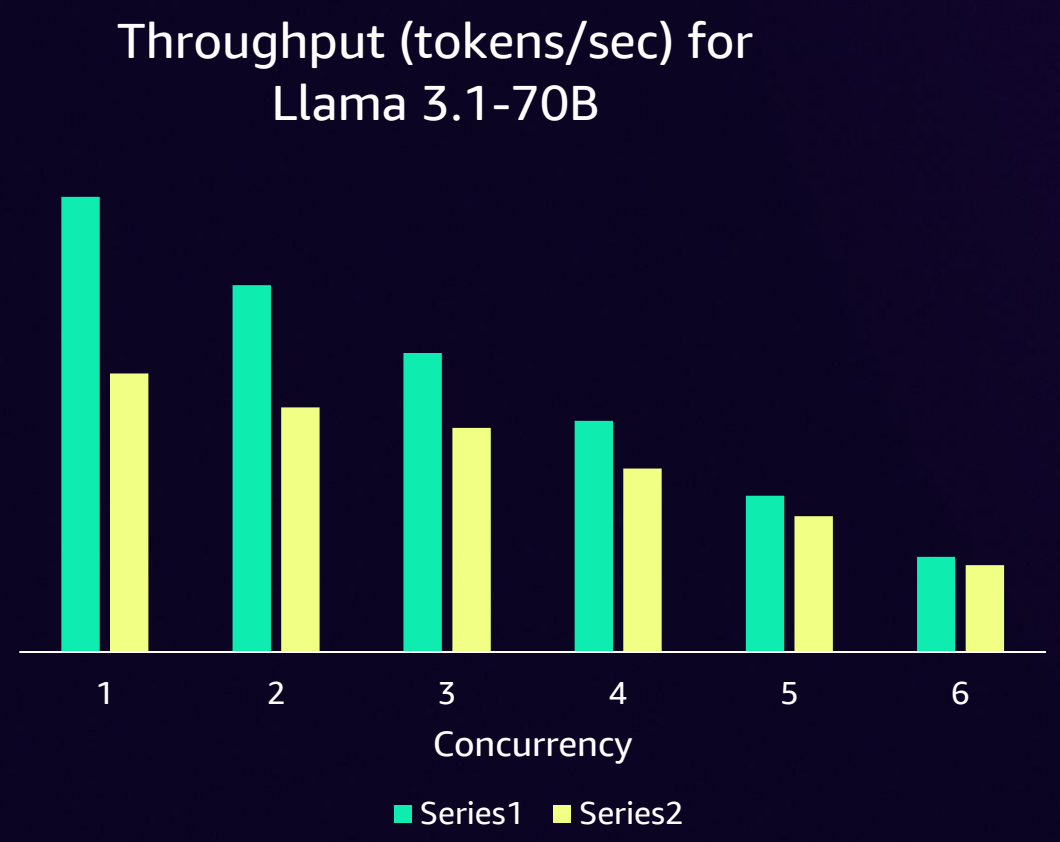
Up to 2x improvement in throughput and up to 50% lower costs for Llama 3 models

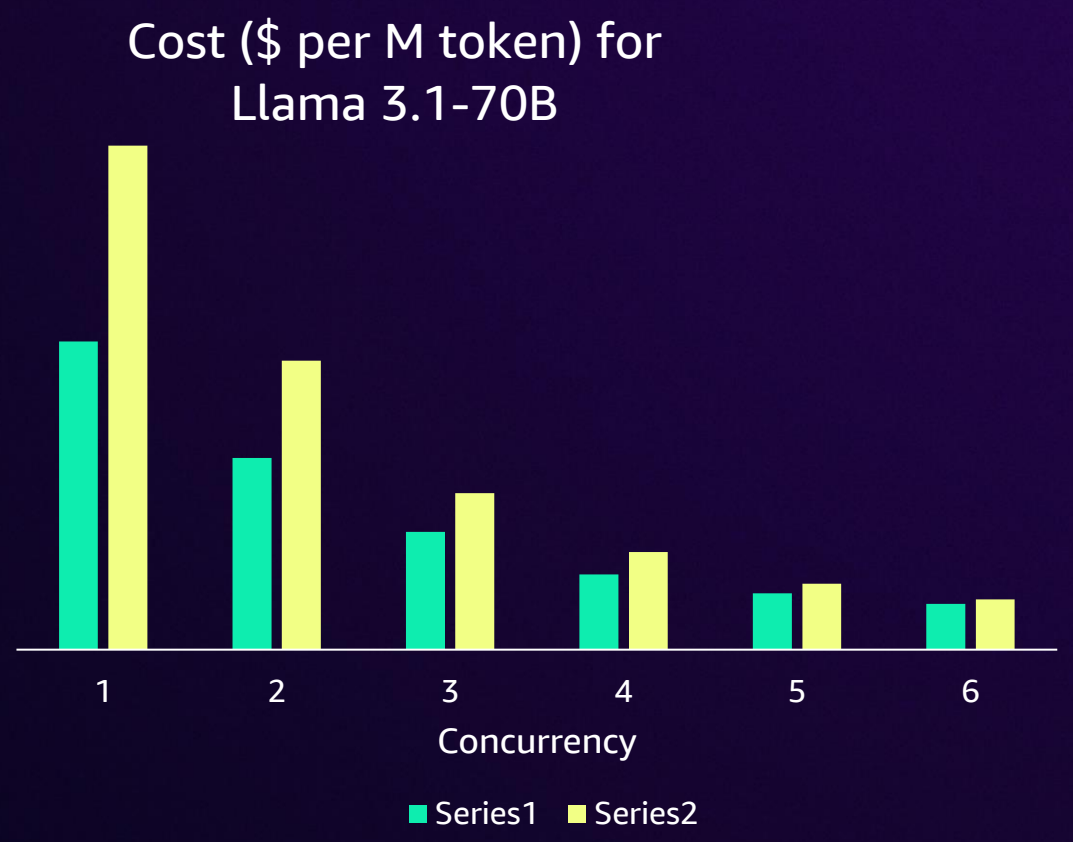




Internal benchmarks on ml.p4d.24xl using OpenOrca dataset

Significant speed-ups with the latest Llama 3.1 models, as well





Internal benchmarks on ml.p4d.24xl using OpenOrca dataset

Scaling improvements

Container caching:

- Cached Amazon SageMaker deep learning containers for scaling speed-up
- Supported frameworks - Large Model Inference (LMI), HuggingFace TGI, Nvidia Triton, PyTorch
- Reduce scaling latencies by up to **56%** for Llama3.1 70B

Fast model loading:

- Stream model weights directly in GPU memory
- Reduce scaling latency by up to 2**0%** for Llama3.1 70B

Scale down to zero

Scale your inference endpoints to zero capacity when there is no traffic

- Endpoint waits 15 minutes before releasing the capacity

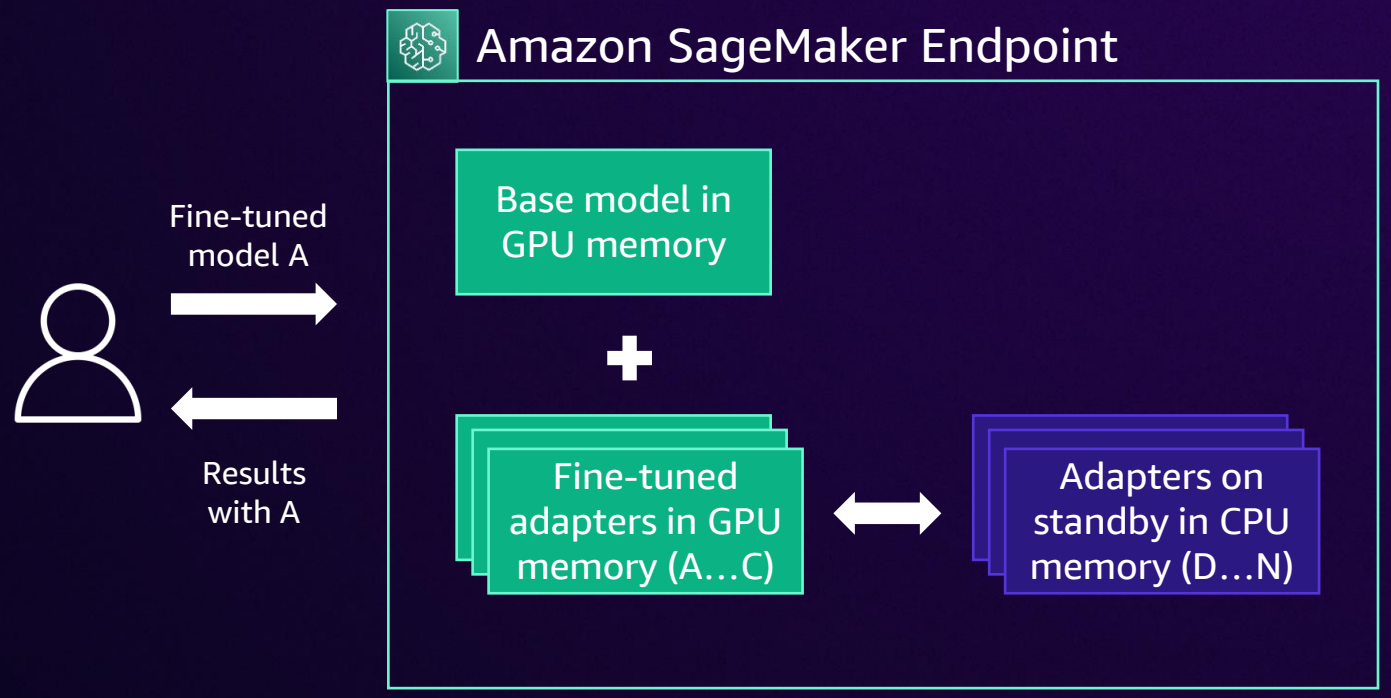
Scale back up with scaling improvements

- Scaling up begins in 1 minute
- Models like Llama3.1 70B scale up and are ready to serve traffic in 6 minutes

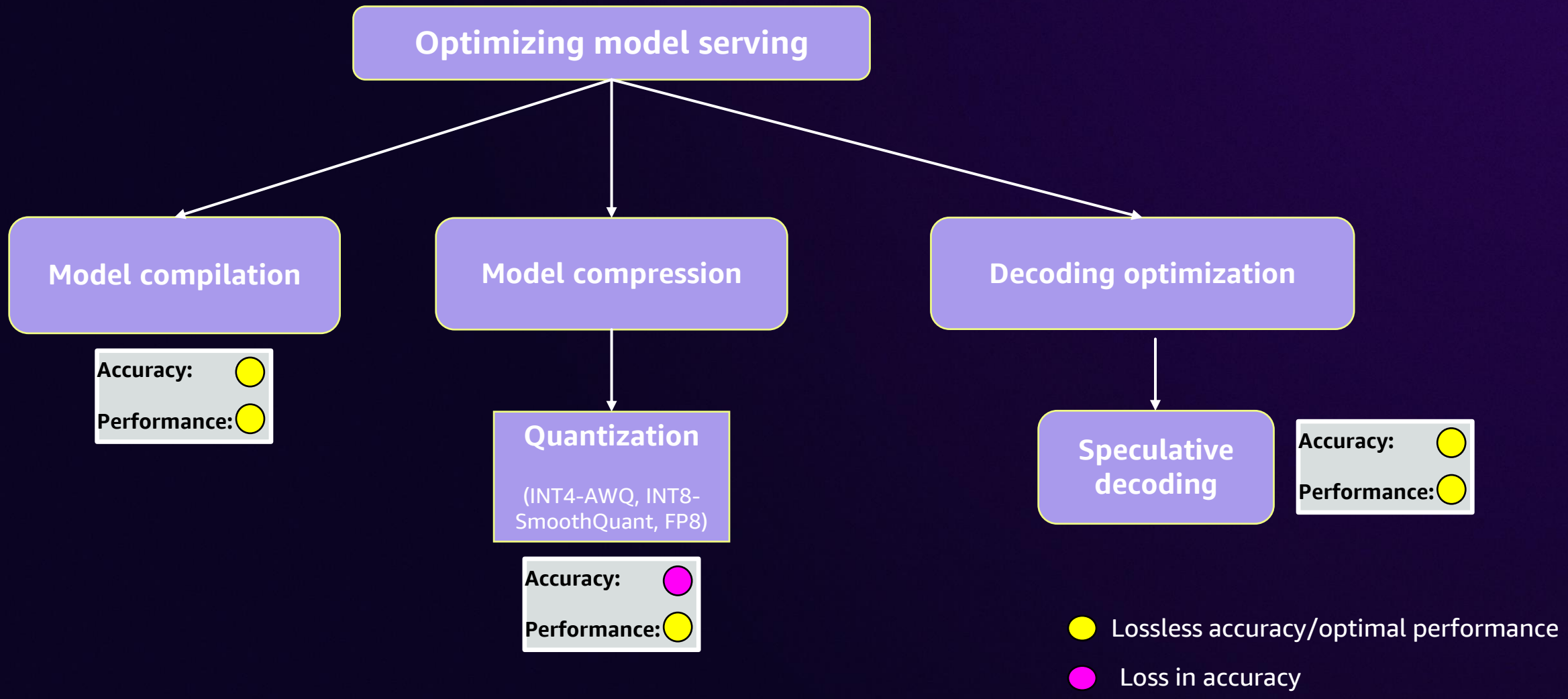
LLaMa3.1 70B			
Time to trigger step scaling policy (min)	Time to provision instance(s) (min)	Time to instantiate a new model copy (min)	Total time (min)
1	3.018	1.984	6.002

Host hundreds of fine-tuned adapters

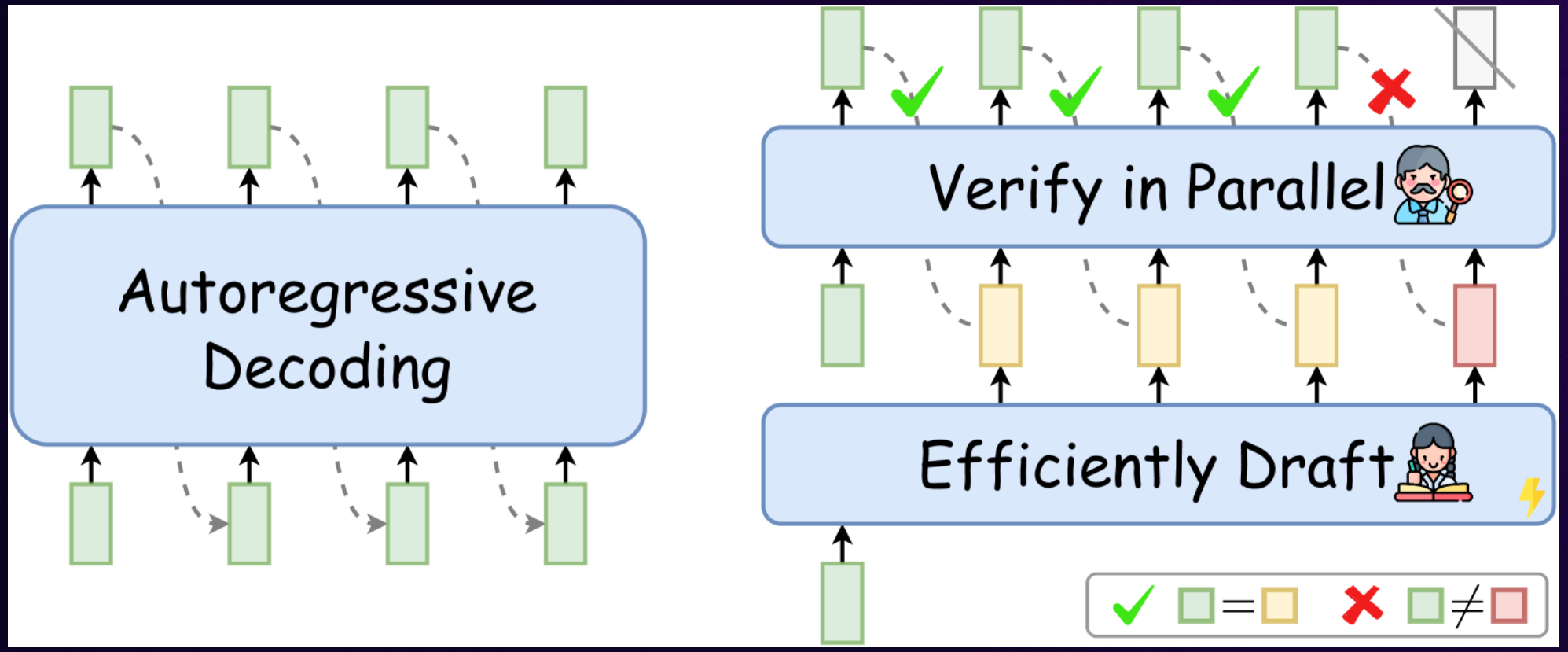
- **Save costs**
  - Host multiple base models with their own fine-tuned adapters on the same endpoint and instances
- **Ease of use**
  - Manage adapters on the endpoint with lifecycle APIs
  - Monitor usage of each fine-tuned adapter
  - Auto scale base model up and down in response to traffic
- **Low overhead latency**
  - <1ms overhead to use adapters in GPU memory
  - <10ms overhead to load adapters from CPU to GPU memory for inference



Inference optimization techniques on Amazon SageMaker



Speculative decoding



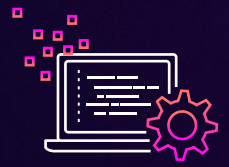
How does it work?

Choose techniques

From a menu of options in Studio/ SDKs

Run optimization jobs

Under-the-hood Amazon SageMaker selects the container, library, and hardware to run the job



Evaluate

Generate evaluation report with performance and accuracy metrics

Deploy

In a single click in Studio or single API call in SDK

Monitor

Validate cost-performance in production

What are customers saying?

Large language models require expensive GPU based instances for hosting, so achieving a substantial cost reduction is immensely valuable. With the new inference optimization toolkit from Amazon SageMaker, based on our experimentation, we expect to reduce deployment costs of our self-hosted large language models by roughly 30% and to reduce latency by up to 25% for up to 8 concurrent request

FNU Imran

Machine Learning Engineer, Qualtrics

What are customers saying?

The Scale to Zero feature for SageMaker Endpoints will be fundamental for iFood's Machine Learning Operations. Over the years, we've collaborated closely with the SageMaker team to enhance our inference capabilities. This feature represents a significant advancement, as it allows us to improve cost efficiency without compromising the performance and quality of our ML services, given that inference constitutes a substantial part of our infrastructure expenses.

Daniel Vieira

MLOps Engineer Manager at iFoods

Demo

Applications (7)

JupyterLab
RStudio
Canvas
Code Editor
Studio Cl...
Dataset ...

Partner AI Apps New

Home
Running instances
Compute
Data
Auto ML
Experiments
Jobs
Inference optimization New
Training
Model evaluation
Performance evaluation
Pipelines
Models
JumpStart
Deployments

Collapse Menu

What's New!

- **Bedrock ready models** You can now use JumpStart to deploy certain models directly into Bedrock and use its feature toolset like Playgrounds and Guardrails. Learn more
- Introducing speculative decoding for Llama-3.1 models.
- Introducing Fast Model Loading for large models.

All public models

Discover all popular pre-trained models offered by SageMaker

Providers 6

Search providers or models...

HuggingFace
Bedrock ready

Explore hundreds of popular and trending models from HuggingFace.

View 406 models

Meta

Explore popular and trending models from Meta including Llama, Code Llama, and more.

View 66 models

Stability AI

Explore popular and trending models from Stability.ai including Stable Diffusion and more.

View 10 models

TensorFlow

Explore popular and trending models from TensorFlow for computer vision and NLP tasks.

View 346 models

PyTorch

Explore popular and trending models from PyTorch for computer vision and NLP tasks.

View 34 models

Amazon

Explore popular and trending models from AWS for computer vision, NLP, and tabular tasks.

View 39 models

Applications (7)

JupyterLab
RStudio
Canvas
Code Editor
Studio Cl...
Dataset ...

Partner AI Apps New

Home
Running instances
Compute
Data
Auto ML
Experiments
Jobs
Inference optimization New
Training
Model evaluation
Performance evaluation
Pipelines
Models
JumpStart
Deployments
Endpoints
Manage deployed models
Projects
Automate model building & deployment

Collapse Menu

What's New!

- **Bedrock ready models** You can now use JumpStart to deploy certain models directly into Bedrock and use its feature toolset like Playgrounds and Guardrails. Learn more
- Introducing speculative decoding for Llama-3.1 models.
- Introducing Fast Model Loading for large models.

All public models

Discover all popular pre-trained models offered by SageMaker

Providers 6

Search providers or models...

HuggingFace Bedrock ready
Explore hundreds of popular and trending models from HuggingFace.
View 406 models

Meta
Explore popular and trending models from Meta including Llama, Code Llama, and more.
View 66 models

Stability AI
Explore popular and trending models from Stability.ai including Stable Diffusion and more.
View 10 models

TensorFlow
Explore popular and trending models from TensorFlow for computer vision and NLP tasks.
View 346 models

PyTorch
Explore popular and trending models from PyTorch for computer vision and NLP tasks.
View 34 models

Amazon
Explore popular and trending models from AWS for computer vision, NLP, and tabular tasks.
View 39 models

Amazon SageMaker launches | SageMakerPublicHub - SageM

studio-d-9ltxbezjspmx.devo.studio.us-west-2.asfiovnxocqpcry.com/jumpstart/SageMakerPublicHub

New Chrome available

Useful | Personal | Sagemaker | All Bookmarks

SageMaker Studio > Jumpstart > SageMakerPublicHub

Provide feedback

- Running instances
- Compute
- Data
- Auto ML
- Experiments
- Jobs
  - Inference optimization New
  - Training
  - Model evaluation
  - Performance evaluation
- Pipelines
- Models
- JumpStart
- Deployments
  - Endpoints
    Manage deployed models
  - Projects
    Automate model building & deployment

Collapse Menu

What's New!

- **Bedrock ready models** You can now use JumpStart to deploy certain models directly into Bedrock and use its feature toolset like Playgrounds and Guardrails. Learn more
- Introducing speculative decoding for Llama-3.1 models.
- Introducing Fast Model Loading for large models.

All public models

Discover all popular pre-trained models offered by SageMaker

Providers 6

Search providers or models...

HuggingFace Bedrock ready
Explore hundreds of popular and trending models from HuggingFace.
View 406 models >

Meta
Explore popular and trending models from Meta including Llama, Code Llama, and more.
View 66 models >

Stability AI
Explore popular and trending models from Stability.ai including Stable Diffusion and more.
View 10 models >

TensorFlow
Explore popular and trending models from TensorFlow for computer vision and NLP tasks.
View 346 models >

This provider contains open source models

PyTorch
Explore popular and trending models from PyTorch for computer vision and NLP tasks.

Amazon
Explore popular and trending models from AWS for computer vision, NLP, and tabular tasks.

JupyterLab
RStudio
Canvas
Code Editor
Studio Cl...
Dataset ...

Partner AI Apps New

Home
Running instances
Compute
Data
Auto ML
Experiments
Jobs
Inference optimization New
Training
Model evaluation
Performance evaluation
Pipelines
Models
JumpStart
Deployments
Endpoints
Manage deployed models
Projects
Automate model building & deployment

Collapse Menu

What's New!

- **Bedrock ready models** You can now use JumpStart to deploy certain models directly into Bedrock and use its feature toolset like Playgrounds and Guardrails. Learn more
- Introducing speculative decoding for Llama-3.1 models.
- Introducing Fast Model Loading for large models.

All public models

Discover all popular pre-trained models offered by SageMaker

Providers 6

Search providers or models...

HuggingFace
Bedrock ready

Explore hundreds of popular and trending models from HuggingFace.

View 406 models

Meta

Explore popular and trending models from Meta including Llama, Code Llama, and more.

View 66 models

Stability AI

Explore popular and trending models from Stability.ai including Stable Diffusion and more.

View 10 models

TensorFlow

Explore popular and trending models from TensorFlow for computer vision and NLP tasks.

View 346 models

PyTorch

Explore popular and trending models from PyTorch for computer vision and NLP tasks.

View 34 models

Amazon

Explore popular and trending models from AWS for computer vision, NLP, and tabular tasks.

View 39 models

llama3-70b-inference-op

Markdown | Notebook | Cluster | Python 3 (ipykernel)

How to optimize the Meta Llama-3.1 70B Amazon JumpStart model for inference using Amazon SageMaker model optimization jobs

Recommended kernel(s): This notebook can be run with any Amazon SageMaker Studio kernel.

In this notebook, you will learn how to apply state-of-the-art optimization techniques to an Amazon JumpStart model (JumpStart model ID: `meta-textgeneration-llama-3-1-70b-instruct`) using Amazon SageMaker ahead-of-time (AOT) model optimization capabilities. Each example includes the deployment of the optimized model to an Amazon SageMaker endpoint. In all cases, the inference image will be the SageMaker-managed LMI (Large Model Inference) Docker image. LMI images features a DJL serving stack powered by the Deep Java Library.

You will successively:

1. Deploy a pre-optimized variant of the Amazon JumpStart model with speculative decoding enabled (using SageMaker provided draft model). For popular models, the JumpStart team indeed selects and applies the best optimization configurations for you.
2. Customize the speculative decoding with open-source draft model.
3. Quantize the model weights using the FP8 algorithm.
4. Compile the model for a deployment TensorRT-LLM framework.

Notices:

- Make sure that the `ml.p4d.24xlarge` and `ml.g5.12xlarge` instance types required for this tutorial are available in your AWS Region.
- Make sure that the value of your "ml.p4d.24xlarge for endpoint usage" and "ml.g5.12xlarge for endpoint usage" Amazon SageMaker service quotas allow you to deploy at least one Amazon SageMaker endpoint using these instance types.

This notebook leverages the Model Builder Class within the `sagemaker` Python SDK to abstract out container and model server management/tuning. Via the Model Builder Class you can easily interact with JumpStart Models, HuggingFace Hub Models, and also custom models via pointing towards an S3 path with your Model Data. For this sample we will focus on the JumpStart Optimization path.

License agreement

- This model is under the Meta license, please refer to the original model card.
- This notebook is a sample notebook and not intended for production use.

Execution environment setup

This notebook requires the following third-party Python dependencies:

- AWS `sagemaker` with a version greater than or equal to 2.232.2

Let's install or upgrade these dependencies using the following command:

[ ]:
```
%pip install "sagemaker>=2.235.2" --upgrade --quiet --no-warn-conflicts
```

Setup

[ ]:
```
import sagemaker
from sagemaker.serve.builder.model_builder import ModelBuilder
from sagemaker.serve.builder.schema_builder import SchemaBuilder
from sagemaker.jumpstart.model import ModelAccessConfig
```

Resources

Inference optimization toolkit

Scale down to zero

Faster model loading

Container caching

Thank you!

Please complete the session survey in the mobile app

Rishabh Ray Chaudhury
rishrayc@amazon.com

Raghu Ramesha
ragmesh@amazon.com