Google Kubernetes Engine と TensorRT-LLM による LLM の大規模・高速推論環境の構築

Google Cloud Next Tokyo

新田 千尋

SpiralAI株式会社
LLMエンジニア

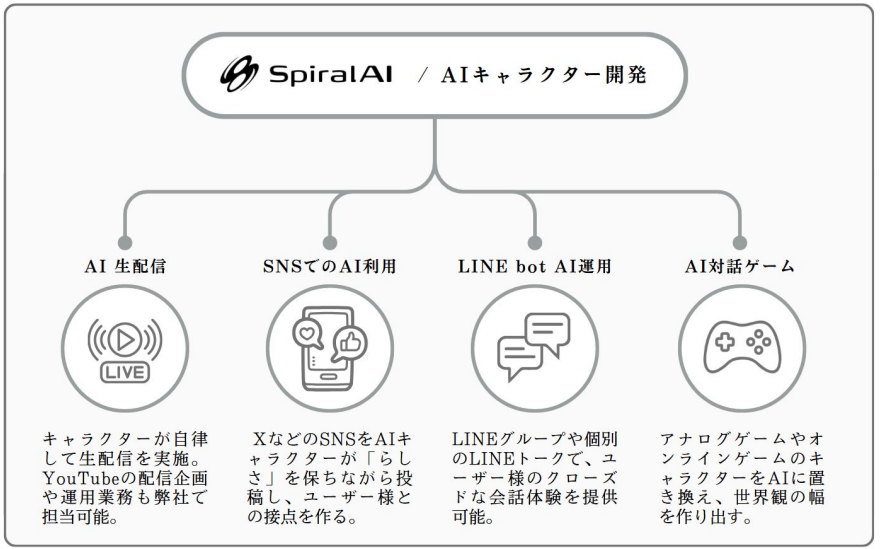
本日のアジェンダ

01 背景
02 課題と取り組み
03 推論サーバの構成
04 GKE 上でのデプロイ
05 まとめ

背景

SpiralAI について

私たちは、「AIキャラクター」を用いて、
対話を必要とするエンタメ事業を展開してます。



HAPPY RAT について

動画あり
アーカイブ動画をご視聴ください

課題と取り組み

LLM の Serving における課題

応答速度

応答速度が**ユーザ体験**に直結

大量のリクエストを**同時に**
高速に処理することが必要

コスト

LLM の推論には GPU が必須であり、**コストが高い**

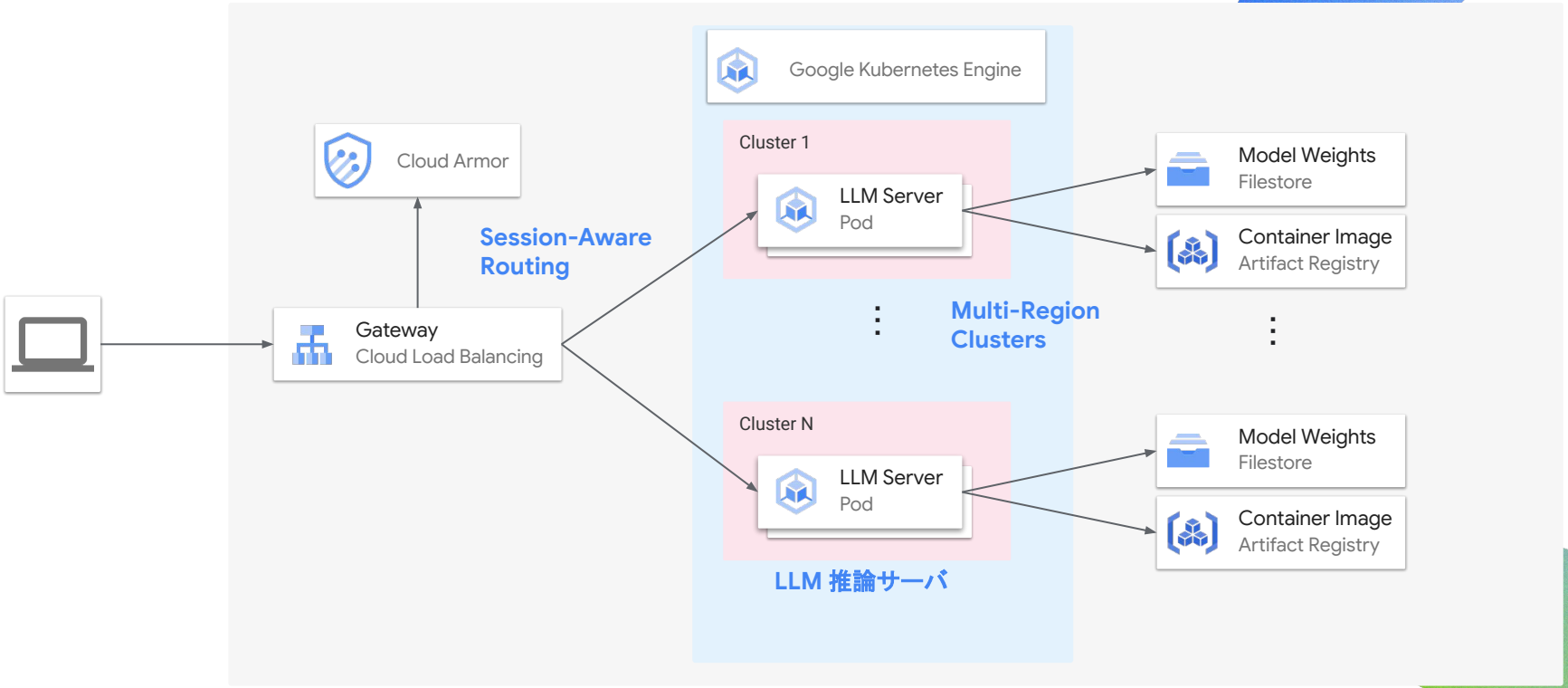
コスト最適化のためには
推論エンジン・サーバ側双方
で様々な工夫が必要

ドメイン知識

環境の構築や最適化のために
深いドメイン知識が必要

量子化のような**軽量化**や、
LLM の性質に基づいた
効率的なルーティングなどの理解が必要

全体のアーキテクチャ


Google Kubernetes Engine
Cloud Armor
Session-Aware Routing
Cluster 1
LLM Server
Pod
Model Weights
Filestore
Container Image
Artifact Registry
Gateway
Cloud Load Balancing
Multi-Region Clusters
Cluster N
LLM Server
Pod
Model Weights
Filestore
Container Image
Artifact Registry
LLM 推論サーバ

LLM 推論サーバの構成

LLM 推論サーバの構成

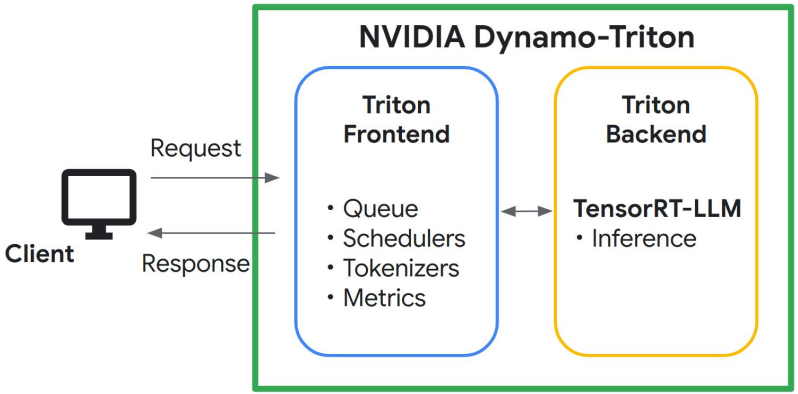
LLM 推論サーバの構成は以下の通り

フロントエンド : **NVIDIA Dynamo-Triton**

- リクエスト受付 / ルーティング
- スケジューリング / キューイング
- Tokenization / Detokenization
- リソース管理、メトリクス収集など

バックエンド : **TensorRT-LLM**

- GPU 上での高速な LLM 推論



NVIDIA Dynamo-Triton
（旧称：Triton Inference Server）

- NVIDIA 製の **AI モデル向けオープンソース推論サーバ フレームワーク**
- PyTorch、ONNX、TensorRT-LLM、vLLM など**多様なバックエンドに対応**
- **HTTP / gRPC エンドポイント** や **メトリクス / ヘルスチェック API** などを提供

TensorRT-LLM

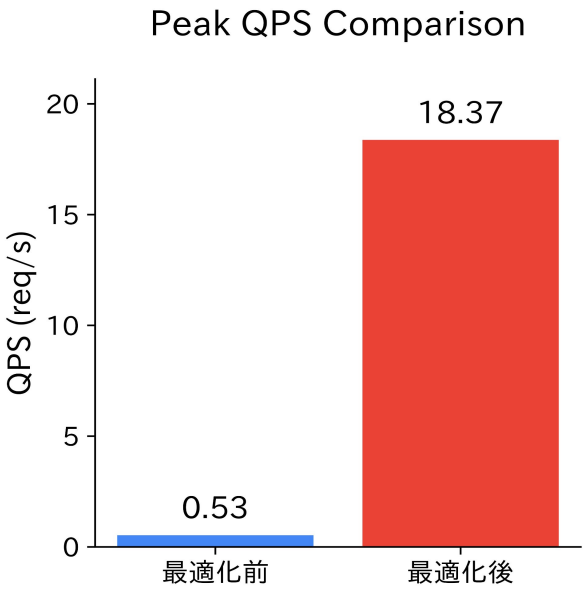
- NVIDIA 製の **LLM 高速推論エンジン**
- Llama、Gemma、Qwen など
  様々なモデルの推論に対応
- **Weight / Activation 量子化** や、
  In-Flight Batching など様々な効率化が可能

パフォーマンス最適化

- **量子化**
  - Weight / Activation / KV Cache
- **In-Flight Batching Scheduler の最適化**
  - max_batch_size / max_num_tokens
- Multiple Profiles / Paged Context Attention / GEMM Plugin / KV Cache Offloading / Speculative Decoding / ...

パフォーマンス最適化

各種最適化により QPS を **30 倍以上改善**



※ NVIDIA L40S 上で huggingface/inference-benchmarker を利用して計測

vLLM との比較

NVIDIA Dynamo-Triton ではバックエンドの
推論エンジンに **vLLM も選択可能**

TensorRT-LLM：速度重視

- 一般的に vLLM よりも推論が早いと言われる

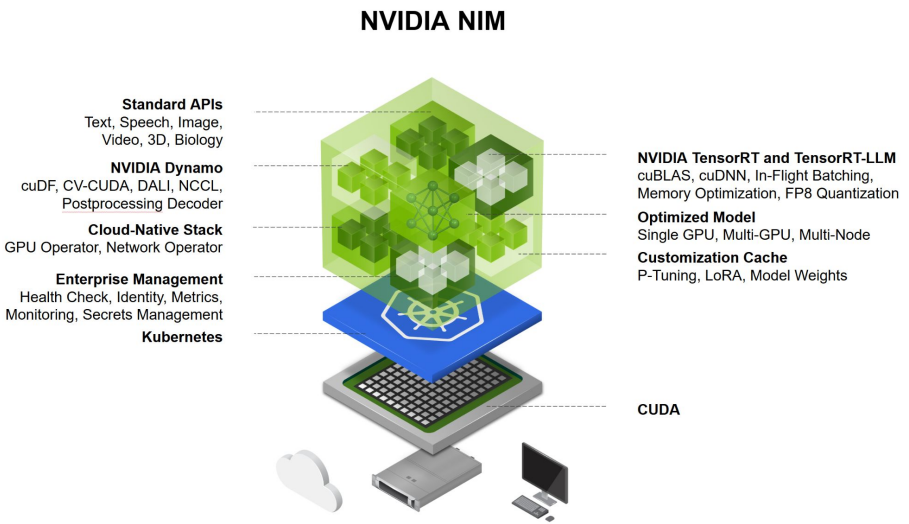
vLLM：扱いやすさ重視

- 導入が非常に簡単
- 新モデルへの対応が早い

NVIDIA NIM™

コンテナベースの Triton + TensorRT-LLM の推論サーバを、**コマンド 1 つ**で構築できるフレームワーク

NVIDIA AI Enterprise による様々なサポート・技術支援を受けられる



NVIDIA NIM™ image courtesy of NVIDIA

NVIDIA Dynamo

より**大規模な LLM の推論環境構築** のためのフレームワーク

Prefill / Decode の分散や KV Cache Offloading、KV Cache-Aware Routing など高度な機能を提供する

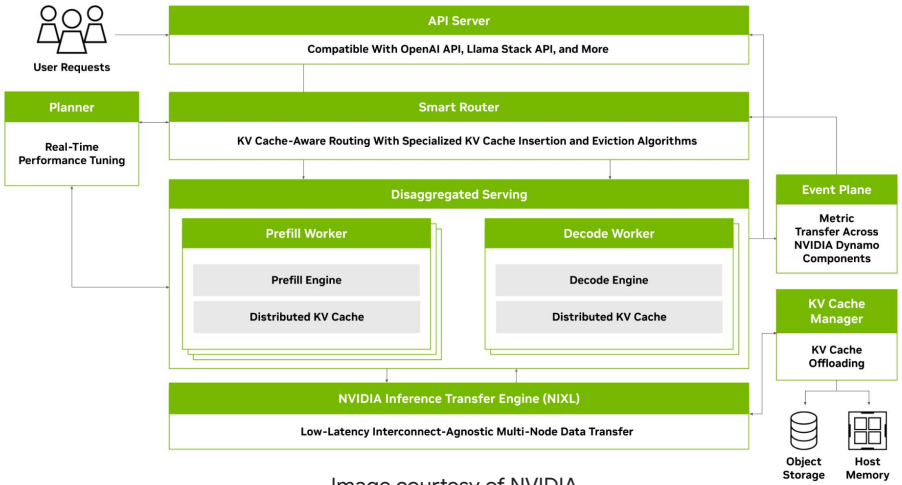
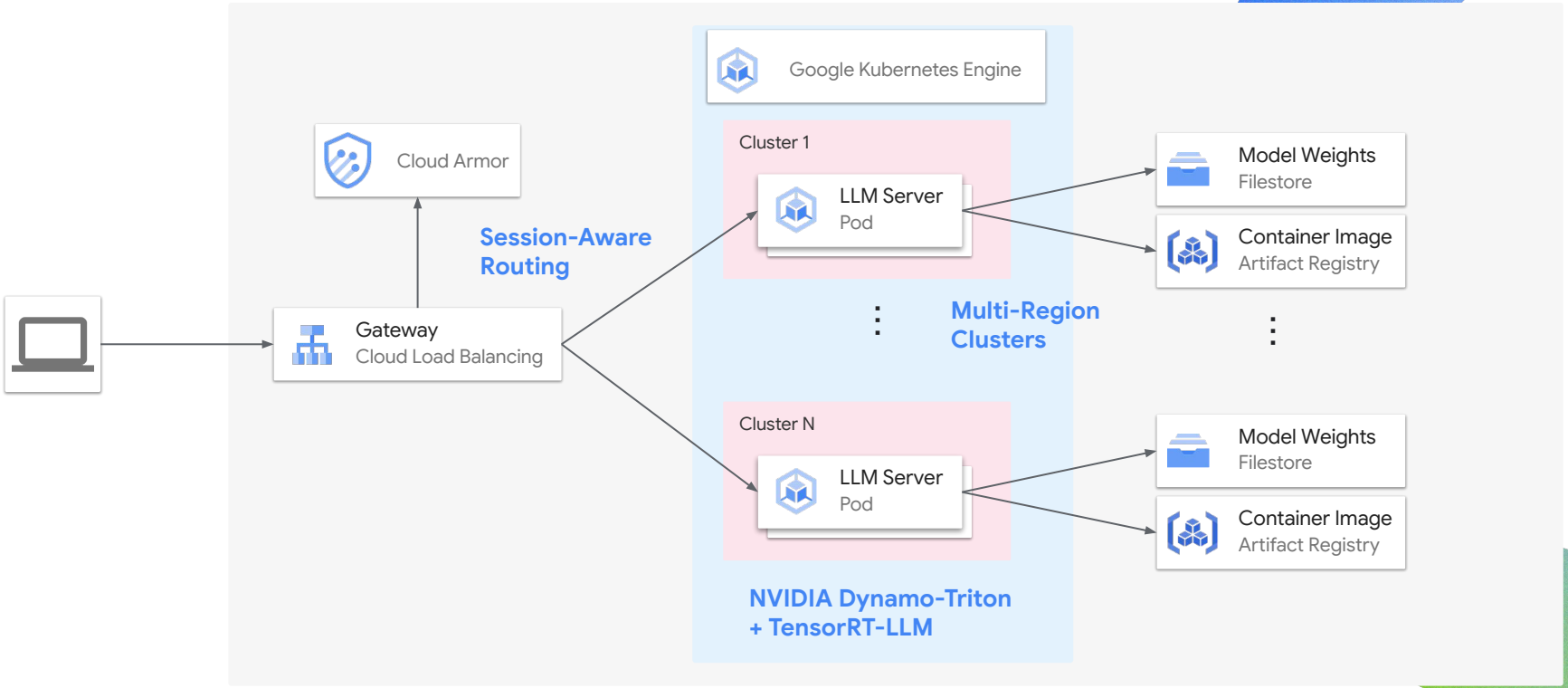


Image courtesy of NVIDIA

GKE 上でのデプロイ

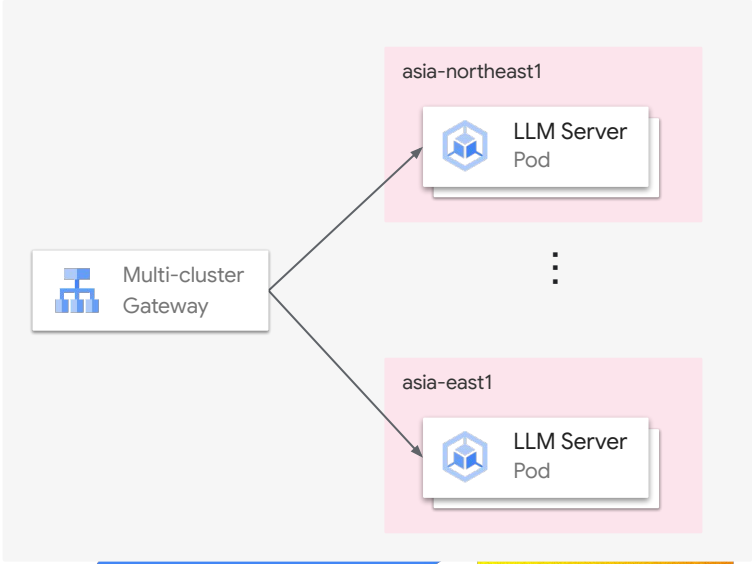
全体のアーキテクチャ



マルチリージョン分散

マルチクラスタ ゲートウェイを利用し、
複数リージョンにリクエストを分散

- **可用性向上**
- **GPU 枯渇対策**
- （リージョン近接性の確保）



Spot VM の有効活用

GKE の**カスタム コンピューティング クラス**を活用

- Spot VM を優先的に利用
- Spot VM が利用できない場合自動的に On-demand にフォールバック

マニフェストの例

```
apiVersion: cloud.google.com/v1
kind: ComputeClass
metadata:
  name: custom-gpu-class
spec:
  priorities:
  - machineType: g2-standard-12
    spot: true
    gpu:
      type: nvidia-l4
      count: 1
  - machineType: g2-standard-12
    spot: false
    gpu:
      type: nvidia-l4
      count: 1
  whenUnsatisfiable: DoNotScaleUp
  autoscalingPolicy:
    consolidationDelayMinutes: 5
    consolidationThreshold: 70
    gpuConsolidationThreshold: 100
  nodePoolAutoCreation:
    enabled: true
  activeMigration:
    optimizeRulePriority: false
```

Session-Aware Routing

KV Cache Reuse

既に計算済みの Prefix 部分の KV Cache を再利用
Prefill にかかる計算の大部分をスキップ・高速化

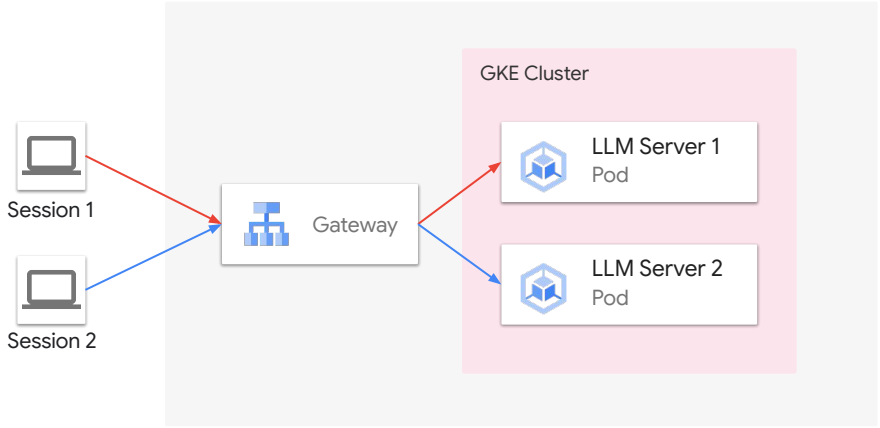
特に**マルチターン対話において有効** に働く

リクエスト1	System: あなたは親切なAIアシスタントです。 User: こんにちは！
リクエスト2	**System: あなたは親切な AI アシスタントです。** **User: こんにちは！** Assistant: こんにちは！なにか御用でしょうか？ User: あなたの名前は何ですか？

Session-Aware Routing

GKE Gateway の **Session Affinity** 機能により、
セッション単位で同じ Pod に Routing

これにより、**KV Cache Reuse の効果を最大化**



柔軟なスケーリング

Triton + TensorRT-LLM では様々な
Prometheus Metrics を収集・公開

これを利用し**柔軟にスケーリング** 可能

- リクエスト処理の成功 / 失敗数
- リクエストの処理に実際にかかった時間
- キューに溜まっている処理待ちリクエスト数

など

まとめ

課題の解決

応答速度

Triton + TensorRT-LLM という構成を採用し、**高速・大規模な推論** を実現

コスト

Spot VM の活用 や、**柔軟なスケーリング戦略** により、サーバコストを最適化

ドメイン知識

モデルの量子化などの **推論エンジン側の工夫** と、Session-Aware Routing などの **GKE 側の工夫** により、効率的な処理を実現