Technical Analysis: LLM Service & Observability Gaps

Overview

This document analyzes the current state of **metrics, monitoring coverage, and observability gaps** for the LLM Passthrough Service

The goal is to:

- Highlight **metrics** needs to be added in the monitoring dashboard and alerts

---

Metrics Categories

The metrics are grouped into the following categories:

1. Success Rate Metrics
2. Usage Metrics
3. Latency Metrics
4. Critical Business Metrics
5. Tenant-Level Error & Latency Metrics

Success Rate Metrics

HTTP Success Rate (2xx & 3xx)

Purpose
Measures the percentage of successful LLM API requests.

PromQL

```
rate(http_request_count_total{status=~"[23].."}[5m]) / rate(http_request_count_total[5m]) * 100
```

Analysis

- Treats only 2xx and 3xx responses as successful
- Suitable for user-perceived success and SLO measurement
- Uses a rolling 5-minute window

Usage Metrics

Service Request Count

Purpose
Measure the **request rate (RPS)** for the LLM service across supported API endpoints in order to understand traffic volume and usage patterns.

PromQL

```
rate(http_request_count-total{name='service-large-language-model',endpoint=~/api/v1/.*'}
```

- **Endpoints of Interest**
    - `/large-language-model/text/completions`
    - `/large-language-model/text/completions/model`
    - `/large-language-model/text/completions/model`/stream
    - `/large-language-model/chat/completions`
    - `/large-language-model/chat/completions/model`
    - `/large-language-model/chat/completions/model`/stream

Analysis

- We dont have any visibility at this point Grafana and Arize has data which can be used for this.

Latency Metrics

Average LLM Request Latency

Purpose
Measure average response time for LLM inference requests.

PromQL Query

```
sum by (name)( request_incoming_duration_seconds_sum{datacenter=~"dc", namespace=~"env", name="service-large-language-model" } ) / sum by (name)( request_incoming_duration_seconds_count{ datacenter=~"dc", namespace=~"env", name="service-large-language-model" } )
```

Analysis

- Add percentile latency metrics (P90/P95/P99)
- Break down by endpoints

Critical Business Metrics

Token Count, LLM Count, Cost of models

Purpose

- Monitor cost drivers
- Identify runaway usage
- Monitor Token Count & LLM Call count

Arize

- Dashboard from Token counts , LLM Calls , Cost of the models can be obtained from Arize
- Arize supports PagerDuty integration to route alerts

Tenant-Level Error Metrics

Endpoint Error Rates

Purpose
Track 4xx and 5xx errors per tenant and endpoint.

PromQL Query

rate(http_request_count_total{datacenter=~"dc",namespace=~"env",name=~"service-large-language-model",http_status=~"5.."}[5]/rate (http_request_count_total{datacenter=~"dc",namespace=~"env",name=~"service-large-language-model"}[5]) * 100

rate(http_request_count_total{datacenter=~"dc",namespace=~"env",name=~"service-large-language-model",http_status=~"4.."}[5]/rate (http_request_count_total{datacenter=~"dc",namespace=~"env",name=~"service-large-language-model"}[5]) * 100

Analysis

- This is the very important because the availability monitoring is currently set through kube_deployment_status_replicas_available which is infrastructure level monitoring for availability, this measures of Application Health monitoring with alerting is needed.
- Captures separate 4xx and 5xx error rate.
- Suitable for high-level tenant health monitoring.

Tenant-Level Response Latency

Tenant Latency

Purpose

Calculate latency from domain service (suitex-search-dispatcher, suitex-search-conversation-assistant) perspective

Arize

- Create dashboard in Arize which has all the datapoints for latency on the above endpoints usage
- Arize supports PagerDuty integration to route alerts

Alerting

Critical Alerts

- **Service Availability - Present**
- **High Error Rate -** Based on the error metric when the threshold is > 15% for 15 minutes

Warning Alerts

- **Response Time Degradation** - P95 latency > 10 seconds for 15 minutes

histogram_quantile(0.95, rate(`request_incoming_duration_seconds_sum{name="service-large-language-model"`}[15m])) > 5

Resource Trends

Resource trends are available, but needs tuning (filters not set to the service names in the panels) https://ukg.grafana.net/d/0VSiotDnk/k8s-cluster-health?var-interval=1m&orgId=1&from=now-6h&to=now&timezone=browser&var-dc=us-east4&var-environment=ds-dev&var-container=service-datascience-gateway&var-datasource=edi58rhsvq2v4b&refresh=5s

---

Story Breakdown

Jira	Story	Description	AC's	Estimate
	LLM Passthrough Service Observability Dashboard in Arize	Create Unified Observability for LLM Passthrough Service. The dashboard will provide a clear operational and business view of the LLM passthrough Service and serve as the primary troubleshooting and monitoring surface.	Panels created for Success rate, Usage and request patterns, Latency behavior , Tenant-level metrics and Business usage metrics Use visualization for trending as needed. Create filter which has Datacenter(all regions), namespace(ds-*), datasource(dev /prod), product_id	5
	LLM Passthrough Service Alerts	Implement Alert for LLM Passthrough for reliability and performance	Create critical alert for High Error Rate (5xx) **-** if error rate is 15% for 15 minutes (polling done 3 times in 5 min interval) Create warning alert for Response Time Degradation - if P95 > 10 sec for 15 minutes(polling done 3 times in 5 min interval) Setup Pagerduty/Slack integration in Arize	5
