Google

AIRS: Scaling Live Inference in Resource Constrained Environments

Search Evaluations, Google

Presenter: Nilesh Jagnik

Authors: Nilesh Jagnik, Xiaohao Yang, Chelsea Chen, Tuan Do, GM Harshvardhan

Evaluations: Guiding Product Excellence

TUNING

Iterative Refinement

Developers use evaluation insights to iterate on product changes before requesting launch approvals.

- Test vs. Base comparisons
- Qualitative & quantitative impact assessment

LAUNCHES

Decision Validation

Predict user impact to justify launching new features and AI-based offerings.

- Business metric forecasting
- Quality verification

GUARDRAILS

Risk Mitigation

Detect unexpected behaviors and regressions to ensure system stability over time.

- Continuous eval tracking
- Zero-diff validation checks

Google

The Search Quality Evaluation Platform

1. Query Stream & Search Stacks
Queries are processed by Test vs. Base stacks to produce results.

Our Team's Focus

2. Ratings Generation (AI & Human)
AIRS automates 100M+ ratings daily using LLMs to mimic human experts.

3. Metric Computation & Impact
Aggregating ratings into metrics, such as Page Quality (PQ) scores, to guide launch decisions.

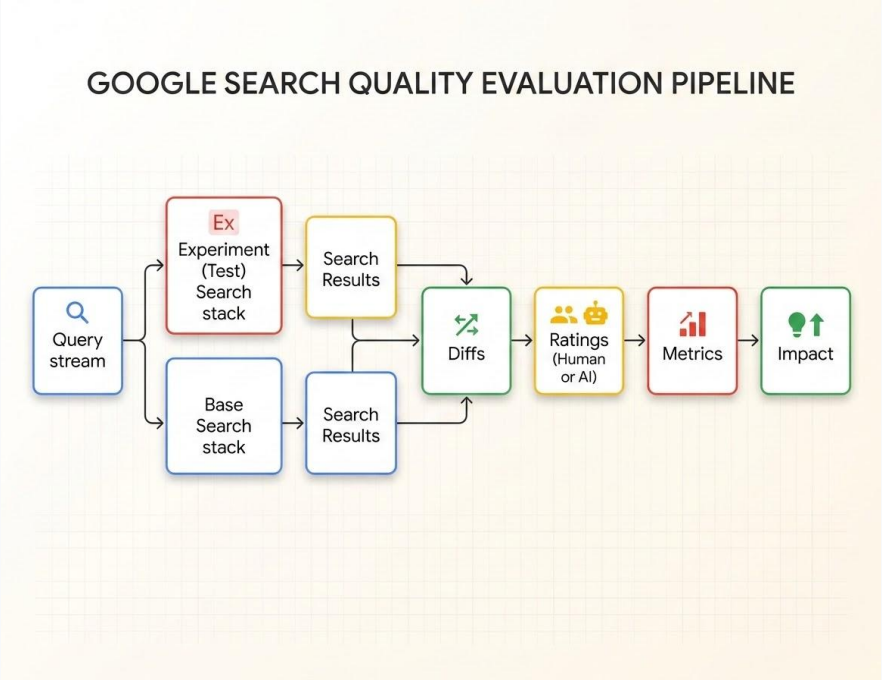

Figure 1. Google Search Quality Evaluation pipeline

Google

The Challenge: Scaling Search Quality Evaluation

Human Bottleneck

Traditional expert rating is **monetarily expensive** and slow, taking days to complete.

- Manual investigation of accuracy, credibility, trustworthiness, etc.
- 180+ page guideline mastery required

Exploding Demand

New AI-based products like **AI Overviews** and **AI Mode** have drastically increased evaluation needs.

- 100M+ rating requests daily
- Hundreds of unique autoraters and metrics

Resource Scarcity

TPU quotas are **heavily constrained** as resources are reserved for live user traffic.

- Allocation falls short of demand
- Disruptions to launch cycles

AI Rater Service (AIRS): Key Features

Intelligent Caching
Reduces TPU cost by reusing ratings within a 90-day freshness window.

Traffic Shaping
Employs back-pressure and retries to handle spiky QPS patterns.

Smart Prioritization
Fast-tracks business-critical auto ratings and evaluations.

Common API Surface
Standardized interface for diverse LLM-based autoraters and prompt logic.

Dynamic Batching
Aggregates requests (default size 12) to optimize inference throughput.

Lifecycle Management
Automated notifications once all ratings for a workflow are complete.

AIRS: Design Overview

Key Components

Rating Fulfillment
Receives tasks from workflows and manages overall fulfillment lifecycle.

Model Management
Provides hosting assistance, tracks TPU utilization, and scales resources.

Operational Modes

Isolated Mode
Components function independently; uses generic retry/back-off strategies.

Integrated Mode
Components work in tandem; Fulfillment peeks at TPU quota utilization for faster throughput.

AIRS Architecture

Figure 2. Unified Design for High Throughput

AIRS: Model Management

Design Principles

AIRS provides hosting assistance and tracks TPU resource utilization to optimize hosted models.

Automated Operations
Auto-restarting failed servers and adjusting TPU usage.

Resource Pooling
Dynamic fleet adjustment to increase TPU duty cycle.

Balanced Scaling
Redistribution based on utilization across the shared pool.

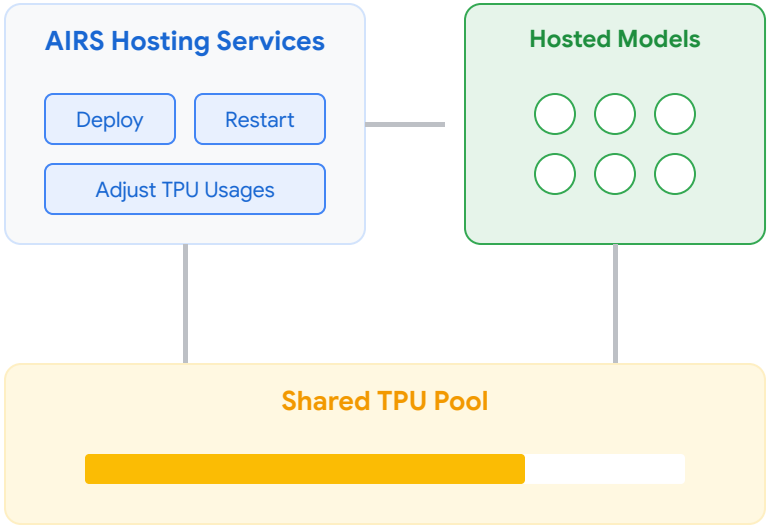

Figure 3. Model hosting and Resource management in AIRS

AIRS: Rating Fulfillment

Fulfillment Workflow

The Rating Fulfillment pipeline manages the reliable execution and tracking of AI-generated ratings.

Client-Side Caching: Reuse ratings within a 90-day window to save TPU costs.

Intelligent Queuing: Buffers tasks and adjusts delivery QPS based on resource health.

Dynamic Batching: Aggregates tasks (default 12) for optimal TPU inference.

Smart Prioritization: Fast-tracks business-critical and human-triggered evaluations.

Lifecycle Monitoring: Notifies workflows once all parallel ratings complete.

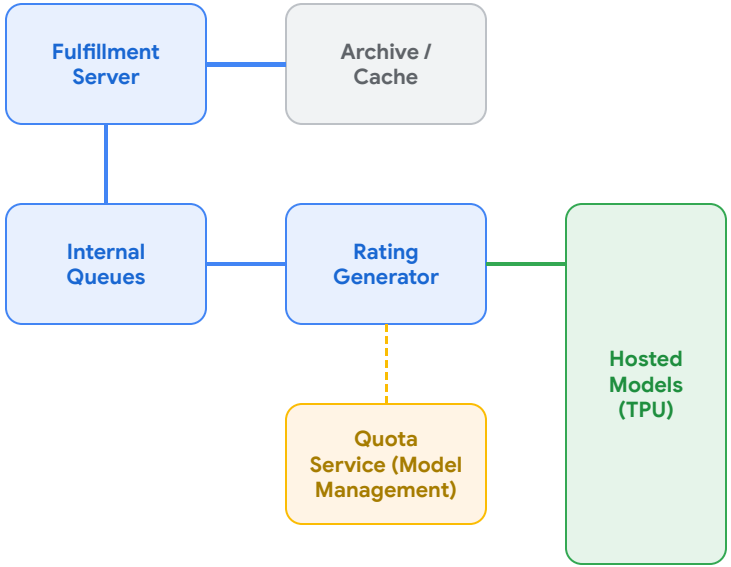

Figure 4. Rating Fulfillment Pipeline & Data Flow

AIRS: Performance Results

Key Efficiency Gains

AIRS optimizations significantly reduce the computational load and increase the reliability of AI-generated ratings.

40% Cache Hit Rate

Client-side caching reuses ratings within a 90-day window, cutting LLM requests drastically.

Sustained TPU Duty Cycle

Mean TPU utilization stays close to 1.0 during peak hours through traffic shaping.

97.8% Reliability

High success rate (80p) achieved despite resource constraints and spiky QPS patterns.

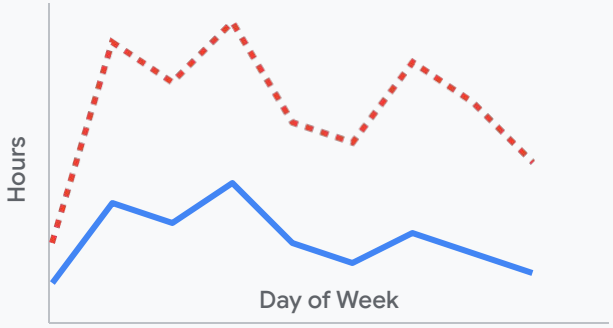

Figure 10. 75th percentile Autorater metric latency
— AR1 (Higher TPU) - - AR2 (Lower TPU)

Latency Impact: High-priority workflows (AR1) fast-tracked to ~30 min, while low-priority (AR2) scale based on shared pool availability.

Google

Thank You!