github.com/openxla/xprof

Google Cloud

research.google/pubs/pub1076558

XProf: An Open, Scalable, and Extensible Profiling System for the Modern ML Stack

Robert Hundt, Naveen Kumar, Jose Baiocchi Pardes, Scott Goodson, Clive Verghese, Prasanna Rengasamy, Kelvin Le, Jiya Zhang, Charles Alaras, Yin Zhang, Kan Cai, Jiten Thakkar, Sai Ganesh Bandiatmakuri, Yogesh SY, Aniruddha N Udipi, Vikas Aggarwal

Unified Visibility Across the ML Stack

Unified Host-Device Profiling: Correlates CPU and accelerator events to high level code.

Deep Compiler / Hardware Integration: XLA compiler connects traces to high level ops through instrumentation data and compiler annotations.

Multi Level Tooling Ecosystem: Specialized tooling across stack including model analysis, memory, performance, and thermals.

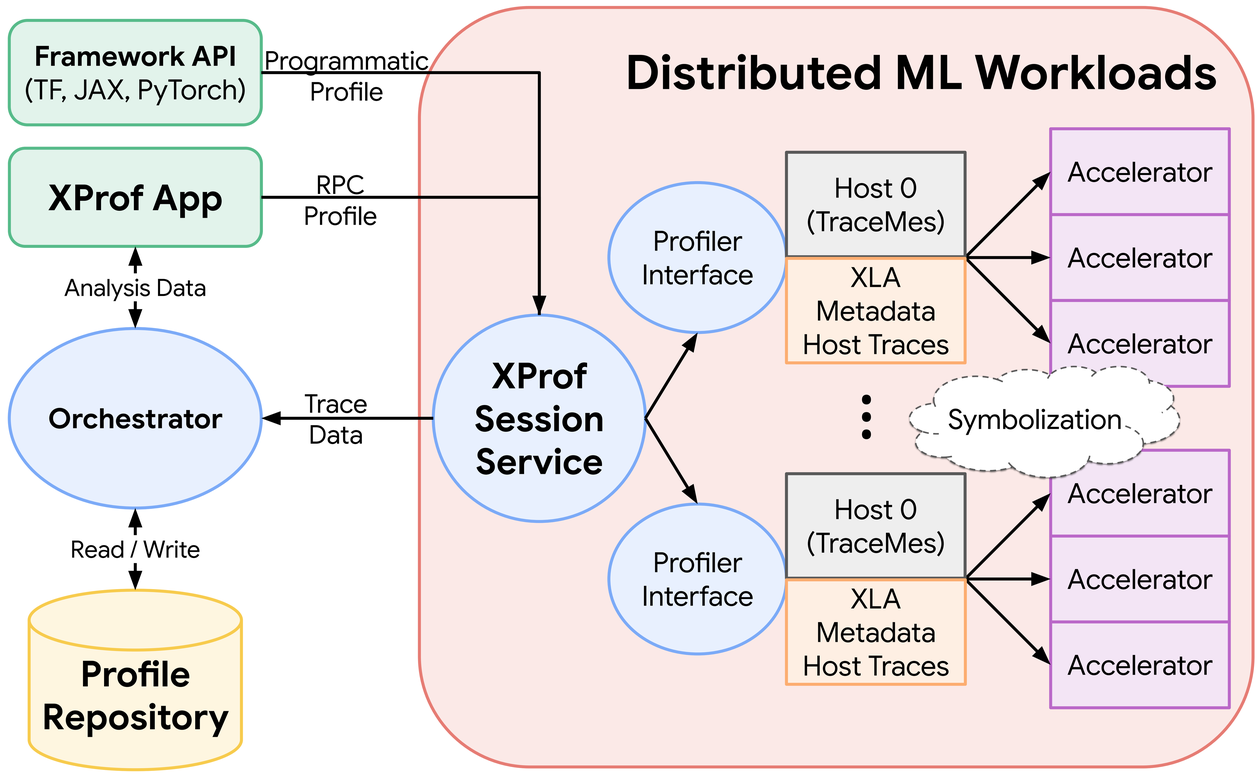
High Precision Distributed Computing: `XprofSessionService` connects to all hosts across a profile, synchronized through a specialized **Global Timestamp Counter** (GTC) on TPUs.

Open and Extensible Framework: XProf interfaces with OpenXLA **PJRT** to provide a Profiler C API for integration with high level frameworks (JAX, PyTorch, TensorFlow).

Advanced Observability: XProf is constantly pushing the boundaries with new profiling capabilities:

PyTorch TPU Continuous Kernel-Level

Join our team!
g.co/jobs/xprof



Parallel Processing Across 1000s of Accelerators

MapReduce Framework: A lead worker divides host collection and post processing across a fleet of workers. Each worker connects to their assigned hosts.

Fleet-wide Synchrony: Collects profiles across hosts at scale, synchronized with GTC leader node exchanging timestamps with leaf counter parents.

User Testimonials

"For our image inference performance challenge, XProf helped us collect and analyze profile traces through which we discovered our cache usage was the problem." - Mathpix

"XProf and Xprofiler tools are very helpful for us to optimize our models. This was one of the major areas of pain point when optimizing our workload during Trillium preview." - HubX

Framework Agnostic and Vendor-Ready Design

Configurable Profiling: TraceMe allows developer defined annotations to regions of interest. **TraceMe** is lightweight to process, and amortizes memory costs with efficient storage.

Vendor Agnostic: XProf introduces a **Profiler C API** as an extension to the PJRT, to allow any hardware vendor to implement profiling support.

Proven Production Efficiency

Low Overhead: Sustains **< 0.28% CPU** overhead on TPUs and **~2.3% on GPUs** during intensive LLM execution.

High-Speed Analysis: Processes massive **1 GB** profile sessions in **< 45 seconds**—enabling rapid iteration.

50% Power Stabilization: Identified synchronization bottlenecks, cutting datacenter power fluctuations in half.

Enhanced Reliability: Halved accelerator temperature swings (from 20°C to 10°C) with **< 1%** performance impact.

Rapid Ecosystem Growth: Driven by internal serving gains and a **49% increase** in open-source PyPI downloads.