Inference @ Scale:  
Accelerating AI models for over a billion users 

EMPOWERING COMMUNITIES WITH AI AT SCALE 

Michael Gschwind  
Facebook AI
AI @ Scale

Scale to complete model portfolio
AI models perform trillions of inference operations every day for the billions of people that use our technologies

Scale to billion-plus users
Run optimally at Facebook scale and for everyone

Scale to range of deployments
Regardless of what device, operating system, or quality of internet connection they may have
Facebook AI for a billion-strong community

Content Understanding

- Content Recommendation (Ads, News Feed)
- Translation (Global Communities)
- Content Review (Inappropriate Content)
AI server capacity growth

- Recommendation system models
- Other ML models

Server capacity (Normalized to Y1Q1)

Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3

Year 1 Year 2 Year 3
PyTorch: One Company, One Framework

On a daily average, there are over 4,000 models at Facebook running on PyTorch.

Over 1,700 inference models at Facebook are in full production on PyTorch.

93%

Over 93% of training models at Facebook are on PyTorch.
PyTorch: from Research to Production

**Experimentation**
PyTorch executes intuitively in Eager Mode familiar to Python developers.

**Development**
Pytorch makes model developers productive with efficient training stacks and domain libraries.

**Deployment**
Pytorch supports efficient model deployments with TorchScript computation graphs and JIT compilation.
PyTorch: a shared, open AI framework

Open Source Framework
State-of-the-art AI framework for global AI community, for researchers and developers

Best-of-breed Extensibility
Best-of-breed tooling and libraries enable modular re-use and focus on innovation.

Impact
Sharing of results in a common environment maximizes research impact, fostering reproducibility and speeding adoption
Machine Learning Execution Flow

Data → Features → Training → Model → Inference → Predictions

- FBLearner Feature Store
- FBLearner Flow
- FBLearner Predictor

CPU, CPU+GPU, CPU Accelerators
Machine Learning Execution Flow

Data → Features
- FB Learner Feature Store
- CPU

Features → Training
- FB Learner Flow
- CPU+GPU

Training → Model

Model → Inference
- FB Learner Predictor
- CPU Accelerators

Inference → Predictions
Model growth and CPU performance

![Graphs showing the growth of NLP model complexity and microprocessor performance over time.](image-url)
AI Accelerators at Facebook
GPU Acceleration

GPU support organically integrated in PyTorch
- Eager mode execution
- TorchScript JIT execution
- Graph-based execution

Widely used for training
- Familiar abstraction for model developers
- Share enablement between training and inference
Custom Inference Acceleration

Custom Inference acceleration

- Initial support with Graph-based execution
- Create multi-vendor shared ecosystem for inference acceleration
Use Case: Fighting hatespeech and misinformation

High quality multi-lingual models
Bring years of research into multi-lingual content understanding into production to understand posts in almost 100 languages and detect undesirable content, such as hate speech, bullying, and misinformation.

Fast model execution
Use accelerators to meet response time SLAs even with complex, compute-intensive large-scale models, for models with over 3 billion parameters and more.

Sustainability
Use accelerators to achieve efficiency gains that allow us to deploy large-scale models under the power constraints of today’s data centers and with the environmental goal of sustainability.
Compute Efficiency: Batching to Use The Whole GPU

Executing Requests one by one as they arrive leaves the GPU mostly idle (top diagram)

Batching improves things a lot!
  • Parallelism is increased 70x for highlighted kernel
Batching With No User Cost: Cross-Request Batching

Clients typically send 1 request at a time (online inference)

Dynamic cross-request batching

- Assemble batch from multiple singleton requests
- Use a watchdog timeout on batch creation to avoid waiting too long to build a batch
Accelerator-Optimized Custom Operators

Replace PyTorch operators with custom CUDA implementations

- e.g., PyTorch-implemented Transformer => Faster Transformers

(not to scale)
Pre- and Postprocessing

Every model requires some pre- and post processing

- Network representation
- Tokenization
- String to vocabulary expansion

Inference acceleration amplifies pre- and post-processing

Amdahls’ Law

\[ S_{\text{latency}}(s) = \frac{1}{(1 - p) + \frac{p}{s}} \]

Example:

\[ S(30) = \frac{1}{(1 - 0.99) + \frac{0.99}{30}} \]

\[ S(30) = \frac{1}{0.01 + 0.033} = \frac{1}{0.043} \]

\[ S(30) = 23 \]
Beyond PyTorch: Open Standards for Hardware and Performance

OCP Open Accelerator Infrastructure
MLCommons MLPerf Benchmarking

* ML Commons and ML Perf are trademarks of ML Commons
Open Compute Project

Industry initiative to standardize data center specifications

- Reduce development cost and enhance interoperability

Broad Industry participation covering

- Data Center Facility
- Hardware Management
- Networking
- Open System Firmware
- Rack & Power
- Security
- Server
- Storage
- Telcos
Open Accelerator Infrastructure

Open specification for integrating accelerators in servers

Subprojects cover all abstraction levels:

- Open Accelerator Infrastructure (OAI)
- OCP Accelerator Module (OAI-OAM)
- OAI Universal Baseboard (OAI-UBB)
- OAI Host Interface (OAI-HIB)
- OAI Power Distribution (OAI-PDB)
- OAI Expansion Beyond UBB (OAI-Expansion)
- OAI Security, Control, and Management (OAI-SCM)
- OAI Tray
- OAI Chassis
MLCommons

Consortium of AI leaders from academia, research labs, and industry

Mission

- Develop fair and consistent AI benchmarks (MLPerf)
- Create open, public data sets
- Share Best Practices for ML software

* ML Commons and ML Perf are trademarks of ML Commons
MLPerf

Benchmark suite to enable comparison between platforms

- CPU
- GPU
- Domain-specific AI accelerators

Benchmark classes

- Training, HPC Training
- Inference: Datacenter, Edge, Mobile, Tiny

* ML Commons and ML Perf are trademarks of ML Commons
Benchmarking AI

Each benchmark suite consists of AI models covering common AI application domains

Full stack evaluation covers

• Model & Dataset
• Framework
• Hardware

Domains

• Vision
• Speech
• Language
• Recommendation

* ML Commons and ML Perf are trademarks of ML Commons
MLPerf: Creating opportunity

Encourage competition for common problems
- "Closed Division": defined models and data sets
- "Open Division": innovative new solutions

Establish comparability and competition
- Reduce risk for technology adopters
- Increase opportunities for innovators

Drive growth and innovation
- AI optimization
- AI accelerators

* ML Commons and ML Perf are trademarks of ML Commons
Summary

Inference acceleration with PyTorch
- GPU acceleration
- AI Inference acceleration

Driving deployment of large-scale language models
- Improved outcomes in fighting hatespeech & misinformation
- Benefit many other content understanding models across modalities

PyTorch as ecosystem for collaboration on AI acceleration
- Big presence in submitted MLPerf benchmarks results
- Join us in enabling the next leap in model quality with accelerators