Tony Lee

STAFF FULL-STACK ENGINEER → ML INFERENCE

tonylee.bio
tungsheng@gmail.com
linkedin.com/in/tonyslee8

PROFESSIONAL SUMMARY

Staff-level full-stack software engineer with 10+ years shipping production frontend, backend, and APIs and leading architecture across cross-regional teams — Angular-to-React modernization, CI/CD and test automation, and measurable performance work. Now transitioning into ML inference engineering through self-directed, reproducible labs: vLLM GPU serving on AWS/EKS, autoscaling and admission control, KV-cache and quantization tradeoffs, and CUDA/Triton kernel benchmarking — measured honestly, negative results included.

SKILLS

Professional (Production Experience)

TypeScript • Go • Python • SQL • React • AngularJS → React Migration • Node.js • GraphQL / REST • Cypress E2E Testing • CI/CD Automation • Performance Optimization • Architecture Design • Cross-Functional Leadership

Self-Directed / Lab

AWS (VPC, IAM, EC2, ALB) • Terraform • Kubernetes (EKS) • Karpenter • Prometheus / Grafana • DCGM Exporter • GPU Capacity Planning • vLLM • LLM Serving • OpenAI-Compatible APIs • Autoscaling • Admission Control • KV Cache / Context Length • Quantization Evaluation • CUDA / Triton Benchmarking • Inference Load Testing (k6)

EDUCATION

M. E. Electrical Engineering
University of California, San Diego
SEP 2006 - JUN 2008

B. S. Electrical Engineering
University of Illinois at Urbana-Champaign
SEP 2003 - DEC 2005

WORK EXPERIENCE

Staff Software Engineer • DTEX Systems
SEP 2024 - PRESENT

- Designed distributed platform services across UI, API, data, and infrastructure boundaries, with emphasis on reliability, deployment safety, and cross-service coordination.
- Improved release stability by containerizing UI services, decoupling shared dependencies, and isolating high-risk deployment surfaces.
- Led a launch-critical OpenSearch Dashboards migration, resolving deployment and environment issues with AWS/OpenSearch engineers.
- Built local development and validation automation that cut dev/test iteration latency by 20% and improved onboarding.

Senior Technical Lead • Cisco Systems, Inc.
AUG 2019 - APR 2024

- Led architecture for the Onboarding Experience platform, helping enterprise customers complete onboarding in under 30 minutes.
- Drove system design across frontend, backend, APIs, and integrations, resolving tradeoffs across US, Europe, and India engineering groups.
- Led AngularJS-to-React modernization to remove security risk and improve maintainability; introduced Cypress end-to-end testing that cut test creation time by 50%.
- Built CI/CD and test automation improvements that reduced regression risk in high-visibility customer onboarding flows.

Senior Software Engineer • Tico Co., Ltd.
JAN 2019 - AUG 2019

- Improved message loading performance by 60% through frontend and API optimization and reduced codebase size by 30% through modular refactoring.

Co-founder / CTO • Popup Technology Co., Ltd.
APR 2016 - OCT 2018

- Built the core platform from scratch, led a 5-person engineering team, and shipped features that doubled revenue within 12 months.

ML INFERENCE — SELF-DIRECTED (2025–PRESENT)

Self-funded, reproducible labs built to learn ML inference in the open — GPU serving and kernel work documented with methodology and negative results, not just wins.

GPU Inference Decision Lab (AWS EKS + Karpenter + vLLM)

- Provisioned a GPU serving stack on AWS EKS with Terraform and Karpenter, running vLLM behind an OpenAI-compatible API and instrumented with Prometheus/Grafana — turning autoscaling, admission control, and capacity choices into reproducible evidence.
- Load-tested burst and long-context traffic: bounded queues held 100% delivery near 2s p95 under burst, and FP8 KV-cache was rejected after delivery dropped below ~70% — keeping the negative result as the decision.

CUDA/Triton GPU Kernel Lab (PyTorch baselines + A10G/H200)

- Built a reproducible CUDA/Triton benchmarking harness for LLM-shaped GPU kernels, validating correctness and measuring latency, bandwidth, and TFLOP/s against PyTorch/cuBLAS baselines.
- Reached ~90% of cuBLAS matmul throughput on H200 with a tuned Triton schedule, and recorded where a persistent-wave variant stayed below the baseline rather than reporting only the win.