Yong Wei

Github

Email : weiyong1024@gmail.com
Mobile : (+86) 173-7657-1024

Education

- **Master - EE - Beihang University** — Beijing, PRC
  Exempt Entry Exam; Finish courses and research publishing mission during the 1st semester. — *Sep. 2016 – Jan. 2019*
- **Bachelor - EE - Beihang University** — Beijing, PRC
  Honors College(top 50 among 3000+ freshmen.); GPA 3.7. — *Sept. 2012 – Jun. 2016*
- **High School - Dalian Yuming High School** — Dalian, PRC
  CPhO - Provincial First Prize, granted exemption from the college entrance examination. — *Sept. 2012 – Jun. 2016*

Competitive programming

- **Google Code Jam Kickstart 2017**
  Top 5%, rank 108th globally. ***scoreboard, id:WeiYong1024***

Career

- **Mindverse(Joined angel round, a pioneering startup for LLM app)** — Hangzhou, PRC
  Infra Tech Lead(4-person team), building AiInfra+DataInfra+DevOps for LLM applications. — *Jun. 2022 - Now*
  - **Company representative products**
    - **MeBot**: Inspiring personal assistant, Web version ranked second in the ProductHunt weekly list in August 2024, App version is now available on the Apple App Store.
    - **MindOS**: Early LLMAgentPlatform, launched in October 2022, ranked first in ProductHunt's weekly list in July 2023.
  - **Company technical design**
    - **LPM**: Large personal model. Based on Lora and user online data to train personalized LLM, which can be used in production environments.
    - **MindOS**: The earliest LLM Agent Platform in China, launched 17 months earlier than ByteDance's Coze platform.
    - **UMM**: Unified mind model. 16 months earlier than OpenAI first proposed the concept of AIAgent.
  - **Infra Technology Responsibilities**
    - **AiInfra - Best practices for building infra of LLM based application**
      - **MLOps Platform:** Design and implement a full-link automated MLOps platform that includes data collection, data augmentation, LLM training, and LLM deployment. Combine the model observation capabilities of the open-source tool ClearML to support real-time LPM training and production deployment..
      - **LLM Unified Access Layer:** In the early stage, Java was used to build basic services, achieving unified access, deployment management, multi-account pooling, health check, weight allocation, cost statistics, and visualization for models such as OpenAI, AzureOpenAI, GCPVertexAI, and Claude. In the later stage, it was migrated to LiteLLM as the best practice to provide the above functions.
      - **LLM Engineering Platform:** Use Langfuse's standardized algorithm for offline experiment management, prompt asset management, asset version management, and online LLM-Tracing functions.
    - **DevOps - Efficient and safe production**
      - **Production Environment Management:** Designed and built multiple production environments such as Prod/Pre/Test, including scalable GitServer with strict CodeReview processes, CI/CD processes, zero-trust device management, and other infrastructure. Technologies involved: Gitlab, Helm, Java, Jenkins, JumpServer, Kubernetes, Octant, Python, Rancher, Shell, etc.
      - **System Observability:** Selection, design, and construction of the distributed configuration, distributed rate limiting, distributed scheduling, application performance monitoring, and log monitoring internal production infrastructure used by the company's backend system. Technologies involved: Apollo, Datadog, Grafana, Loki, Nacos, Prometheus, Sentinel, Skywalking, xxl-job, Tencent Cloud CLS, etc.

      - **Cloud and third-party service management:** Internal user permission management, refined control of cloud usage and third-party service costs, design, production, and maintenance cost dashboard. Design technologies: Azure, GCP, CronJob, TencentCloud, etc.
    - **DataInfra - Data assets, experimental tools, data-driven decision making**
      - **Online and Offline Data System:** Design, build, and maintain online and offline data streams including OLTP->CDC->OLAP->DataVisualization for offline data development, and asynchronous data streams including message queue->buried point service for user behavior data analysis and other scenarios. Support various metrics dashboards, operational BI reports, and other requirements. Technologies involved: MySQL, Clickhouse, Debezium, Superset, RocketMQ, etc.
      - **A/B Testing Tools:**Selection, design, construction, and maintenance of a full-link experimental tool system that includes experimental configuration/result analysis platform (GrowthBook) + front-end tracking (Google Analytics) + back-end tracking (Springboot implementation, including engineering and algorithms). Implement feature toggles and A/B testing for front-end pages and back-end algorithms. Technologies involved: Google Analytics, GrowthBook, Java, NodeJS, Python, etc.
      - **Product Privacy Compliance:** Establish a series of privacy compliance standards, including but not limited to data anonymization, internal training, etc., to ensure that the company's products meet the privacy compliance standards of North America's USDP and the EU's GDPR.
    - **Business Layer Development**
      - **Middle Platform Services:** Designed and built mid-tier basic microservices, offering common foundational services to business systems like backend encrypted storage, user privacy database, web crawlers, offline data warehouse management, etc., and delivered them in forms of RestfulAPI, secondary libraries + RPC interfaces, etc.
  - **Business Support**
    - Led mobile app mini-programs and web product MindOS's subsystems & full-link load testing, and safeguarded the global launch processe; prepared and established SOPs for toB public cloud/private cloud/private deployment; supported service architecture migration across countries/cloud regions as per business needs.
    - Promoted a company-wide blameless postmortem culture, organized safety production weekly meetings.

- **Alibaba Cloud(China's leading cloud service provider)** Hangzhou, PRC
  Senior Engineer, Backend, Java, Springboot, Alibaba Cloud *Aug. 2020 - May. 2022*
  - **Traffic Scheduling Middleware:** : In the group's high-availability team, responsible for Alibaba Cloud's internal traffic scheduling middleware. Utilizing Alibaba's internal PandoraBoot (an enhanced version of Springboot), provided container granularity second-level health monitoring and3σoutlier detection on the cluster. Leveraged the weight table mechanism of Alibaba's internal RPC framework HSF for minute-level traffic scheduling and recovery linkage of microservices. This system prevented cluster avalanches caused by single-point failures and was also used for JIT preheating. During my tenure, the product was integrated into all core applications of Alibaba Group, supporting over 400,000 containers, and served as a core security component safeguarding the peak traffic of 600,000 QPS during the 2020 Double Eleven shopping festival.
  - **Private Cloud Management Platform**: As part of the P8 team, the goal was to build a private cloud management system for government scenarios. Developed using the Java framework Springboot, the system is divided into three subsystems: application operations, resource operations, and general management. I was responsible for backend development of the application operations center subsystem. The subsystem's base layer included cloud resource listing, monitoring and inspection, usage statistics, custom interface inspection, SQL inspection, etc. On top of this, we built expert reports such as slow SQL statistical analysis, business dashboards, and health check reports. This system is currently commercially deployed in over 6 locations nationwide, supporting projects like elementary school admissions in a provincial capital city in 2021 and nucleic acid testing during the 2022 Spring Festival.
  - **Lightweight Delivery for Government Cloud Management**: As a new P9 team member, the goal was to deliver private clouds for government use in 7 regions. Initially, developers used the Rainbond interface to deploy microservices from different teams onto customer K8S clusters, but the complicated configuration led to long delivery cycles. To address this, I developed a lightweight deployment delivery solution using docker-compose, based on generic environment initialization scripts and delivery step documents. By maintaining environment variables for different customer sites, we centralized control and integrated over 20 microservices from various older teams. I edited the WBS and SOP, organized training for outsourced staff on the delivery process, and trained outsourced individuals could deliver or upgrade a site in 2 days. With this solution, the team's monthly commercial output capacity increased from single to double digits.

- **Pony.ai(L4 autonomous driving solution)** Beijing, PRC
  Engineer, Infra, C++, Python, Linux, Bazel, Strict Code Quality Standards *Feb. 2019 - Mar. 2020*
  - **Driving Recording Toolchain**: Autonomous vehicle road tests required an accompanying engineer to record issues via an external keyboard. To eliminate the need for this, a driving recording toolchain was developed using microphones and physical buttons as the hardware solution. Firstly, an input device interface library evdev, licensed under BSD, was integrated into the Bazel main project. Based on this library and Linux's ALSA audio driver, a daemon was written responsible for hardware detection, listening for signals from external buttons and the vehicle system process's pipeline, triggering and stopping microphone recording, and storing issue information as audio files on the industrial computer during the drive. In the data processing stage, a speech-to-text service was set up on GCP. Python scripts were used to extract the contents of the audio files, and Google's Protobuf tool was used to serialize the original information into the QA data stream.
  - **Onboard Car Voice System**: The original onboard voice module used Google's Pico TTS library to play hardcoded voice content text in real-time during driving, leading to issues of language monotony and repetitive consumption of computing resources for TTS processes. The new onboard voice system uses an audio file-based workflow, aimed at reducing the computational load on the vehicle's industrial computer and supporting voice I18N. For this, 1. A Python-based internal CLI tool, `car_sound_utils` (referred to as CLI), was developed for engineers to create and modify existing corpora and manage voice pack versions. First, an entire TTS service built on AWS using CloudFormation and Lambda functions was encapsulated in the CLI for creating audio commands. Then, commands for uploading and downloading voice files and uploading voice packs to the internal storage server were added to the CLI, along with a caching mechanism to speed up uploads and downloads. 2. In the onboard system, the main process initializes the voice module using Linux's ALSA driver to load the audio stream into RAM, and plays it when the voice module receives a playback request message.

Internship

- **Airbnb**: In the summer of 2018, Airbnb expanded into the Chinese market, and as one of the first 8 interns in the Mainland China office, developed the domestic version of the annual host review page with full-stack tech.

- **Megvii(Face++)**: During the cooperation between Megvii and VIVO in early 2018, participated in the development of the facial recognition module for the X21 model and conducted mobile model search optimization based on InceptionV3(a type of CNN) for mobile devices.

System experience & Tech stack

- **System experience**: MLOps, LLM applications, Prompt engineering, large-scale distributed systems, high availability, observability, autonomous driving, etc.

- **Tech stack**: Java, Python, Kubernetes, Shell, C++, mainstream clouds services, mainstream DevOps tools, etc.