Project InfiniEdge AI

Presentation

https://docs.google.com/presentation/d/1mnMpxQvUofmwayh0Gixfne7pWKrq27nRNM_kTyazT4c/edit?usp=sharing

Meetings

Date: every Monday Pacific, every Tuesday Beijing

Time:

7pm Pacific, 10am Beijing,

Location: https://zoom.us/j/92366176176?pwd=cjNacytlcG1iVGNjMGN3RGJ2OEU2Zz09

Required Information	**Responses (Please list N/A if not applicable)**
Name of Project	InfiniEdge AI

Project Description (what it does, why it is valuable, origin and history)	InfiniEdge AI is a project that brings AI to the edge, enabling real-time AI inference. This technology extends the advantages of AI and Machine Learning to edge devices, thus aligning with the LF Edge Mission Statement. This technology can enhance applications in various sectors including manufacturing, telecommunications, healthcare, automotive (autonomous driving and smart cockpit), retail and etc. InfiniEdge AI originated as part of the AI Edge Blueprint Family (https://wiki.akraino.org/display/AK/The+AI+Edge+Blueprint+Family) and is based on the Shifu Framework (https://github.com/Edgenesis/shifu) and YoMo (https://github.com/yomorun/yomo) • Objective of the InfiniEdge AI Project ○ Compared running on centralized mega data centers, InfiniEdge AI Project aims to bring AI inference closer to end users, achieving low-latency response, cost-optimized inference and enhanced privacy protection. • Importance of integrating AI at the edge ○ Real-time decision making: By Integrating AI and IoT at the edge, we can analyze data right at the edge and make immediate decisions. This eliminates the delay caused by transmitting data back to the cloud or a centralized server for processing. In cases like autonomous driving and industrial 4.0 this can be critical. ○ Data/Traffic management: Edge computing units generate large amount of data. Instead of sending them to the centralized cloud, edge AI can process data locally, deciding what to send for further analysis and what to discard, leading to more efficient data management. Processing data at the edge reduces the amount of data that needs to be sent over the network, saving on bandwidth costs and reducing network traffic. ○ Security: With AI at edge. Data can be processed without ever leaving the edge. Reducing the risk of privacy violation and data breaches. • **Problem Statement**: ○ Data can not be transferred to AI model in time (e.g. real-time speech recognition by OpenAI whisper, demo: https://edge-ai.yomo.run/). ○ The cost of building and maintaining production-grade AI inference services is high (the infra of running AI inference is different from AI training). ○ More sensitive data will be involved when using AI (e.g. biological data when using speech recognition). ○ Our project aims to enable efficient inference on resource-constrained edge devices by utilizing large models to train and generate smaller models. Edge devices often face limitations in computing power, memory, and energy resources, while modern deep learning models tend to be large and computationally intensive. Directly deploying complex models on edge devices leads to poor performance and high energy consumption, negatively impacting user experience and device practicality. To address this challenge, we employ knowledge distillation to assist in training and generating compact small models. Knowledge distillation transfers the knowledge of the large model to the smaller one, enabling the small model to perform similarly to the large model while using fewer computational resources. This approach not only facilitates efficient inference on edge devices but also allows complex tasks to be offloaded to cloud-based large models, fully leveraging the advantages of both edge and cloud computing. • **Goal of the project**: ○ To create a unifying platform for running AI inference on the edge. • **Expected benefits**: ○ Low-latency processing capabilities. ○ Cost Efficiency. ■ Pre-process / AI Predict / Post-process running on different systems. ■ Optimize AI Model for heterogeneous systems. ○ Improved privacy protections.
Statement on alignment with Foundation Mission Statement	InfiniEdge AI aligns with the LF Edge Mission Statement by creating an open, scalable, and interoperable framework for edge computing. This project embodies LF Edge's vision for edge applications by extending AI and Machine Learning benefits to edge devices.
High level assessment of project synergy with existing projects under LF Edge, including how the project compliments/overlaps with existing projects, and potential ways to harmonize over time. Responses may be included both here and /or in accompanying documentation.	InfiniEdge AI enhances the overall LF Edge ecosystem by providing an AI/ML interface for edge devices. It does not overlap significantly with existing projects but brings unique capabilities to the table. Harmonization potential exists with IoT and edge computing-focused projects.
Link to *current* Code of Conduct	N/A
2 TAC Sponsors, if identified (Sponsors help mentor projects) - See full definition on Project Stages: Definitions and Expectations	Toshimichi Fukuda , Fujitsu; Tina Tsou , Arm
Project license	Apache 2.0
Source control (GitHub by default)	GitHub
Issue tracker (GitHub by default)	GitHub

<table>
<tr><td>External dependencies (including licenses)
Tom Qin</td><td>
<table>
<tr><td>https://github.com/apache/plc4x</td><td>Apache-2.0 license</td></tr>
<tr><td>https://github.com/gopcua/opcua</td><td>MIT license</td></tr>
<tr><td>https://github.com/eclipse/paho.mqtt.golang</td><td>EPL-2.0</td></tr>
<tr><td>https://github.com/kubernetes/client-go</td><td>Apache-2.0 license</td></tr>
<tr><td>https://github.com/DATA-DOG/go-sqlmock</td><td>BSD license</td></tr>
<tr><td>https://github.com/briandowns/spinner</td><td>Apache-2.0 license</td></tr>
<tr><td>https://github.com/go-sql-driver/mysql</td><td>MPL-2.0 license</td></tr>
<tr><td>https://github.com/microsoft/go-mssqldb</td><td>BSD-3-Clause license</td></tr>
<tr><td>https://github.com/minio/minio-go/</td><td>Apache-2.0 license</td></tr>
<tr><td>https://github.com/mochi-mqtt/server</td><td>MIT license</td></tr>
<tr><td>https://github.com/onsi/ginkgo/</td><td>MIT license</td></tr>
<tr><td>https://github.com/spf13/cobra</td><td>Apache-2.0 license</td></tr>
<tr><td>https://github.com/stretchr/testify</td><td>MIT license</td></tr>
<tr><td>https://github.com/taosdata/driver-go</td><td>MIT license</td></tr>
<tr><td>https://github.com/knative/pkg</td><td>Apache-2.0 license</td></tr>
<tr><td>https://github.com/kubernetes-sigs/controller-runtime</td><td>Apache-2.0 license</td></tr>
<tr><td>https://github.com/yomorun/yomo</td><td>Apache-2.0 license</td></tr>
<tr><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Release methodology and mechanics</td><td></td></tr>
<tr><td>Names of initial committers, if different from those submitting proposal</td><td>Liya Yu, Baidu Yu, Liya
C.C., Allegro fanweixiao
Jun Chen, Baidu Jun Chen
Tom Qin, Edgenesis Tom Qin
Yongli Chen, Edgenesis
Kevin Zheng, Edgenesis
Wenhui Zhang, Bytedance/TikTok Wenhui Zhang
Joe Speed, Ampere
Ray Chi, Advantech
Roger Chen, SuperMicro
Rick Cao, Meta
Reo Inoue, Fujitsu Inoue Reo
Ashok Bhat, Arm
Milos Puzovic, Arm
Tina Tsou , InfiniEdge AI
Qi Wang, Google
Caleb Jiang, Applied Concept Inc.
Vijay Chintha , Comcast Vijay Chintha</td></tr>
<tr><td>Current number of code contributors to proposed project</td><td>6</td></tr>
<tr><td>Current number of organizations contributing to proposed project</td><td>4 (Baidu, Allegro, Edgenesis, TikTok)</td></tr>
</table>

Briefly describe the project's leadership team and decision-making process	Ye Wang / Architect, Baidu C.C. / CEO, Allegro fanweixiao Yongli Chen / CEO, Edgenesis
Advisors	Ranny Haiby Tina Tsou
List of project's official communication channels (slack, irc, mailing lists)	N/A
Link to project's website	N/A
Links to social media accounts	N/A
Existing financial sponsorship	N/A
Infrastructure needs or requests (to include GitHub/Gerrit, CI/CD, Jenkins, Nexus, JIRA, other ...)	GitHub
Currently Supported Architecture	x86-64, AArch64
Planned Architecture Support	N/A
Project logo in svg format (see https://github.com/lf-edge/lfedge-landscape#logos for guidelines)	N/A
Trademark status	N/A
Does the project have a Core Infrastructure Initiative security best practices badge? (See: https://bestpractices.coreinfrastructure.org)	No
Any additional information the TAC and Board should take into consideration when reviewing your proposal?	N/A

Project Proposal - Mapping Criteria and Data:

Stage 1: At Large Projects (formerly 'Sandbox')

2 TAC Sponsors, if identified (Sponsors help mentor projects) - See full definition on Project Stages: Definitions and Expectations	N/A
A presentation at an upcoming meeting of the TAC, in accordance with the project proposal requirements	N/A
The typical IP Policy for Projects under the LF Edge Foundation is Apache 2.0 for Code Contributions, Developer Certificate of Origin (DCO) for new inbound contributions, and Creative Commons Attribution 4.0 International License for Documentation. Projects under outside licenses may still submit for consideration, subject to review/approval of the TAC and Board.	Yes
Upon acceptance, At Large projects must list their status prominently on website/readme	Yes

Project Proposal - Taxonomy Data:

Functions (Provide, Consume, Facilitate, or N/A; Add context as needed)

APIs	Provide
Cloud Connectivity	Provide
Container Runtime & Orchestration	Consume
Data Governance	Provide, Consume
Data Models	Provide
Device Connectivity	Consume
Filters/Pre-processing	N/A
Logging	Consume

Management UI	Consume
Messaging & Events	N/A
Notifications & Alerts	N/A
Security	N/A
Storage	Provide, Consume, Facilitate

Deployment & Industry Verticals (Support, Possible, N/A; Add context as needed)

Customer Devices (Edge Nodes)	N/A
Customer Premises (DC and Edge Gateways)	Support
Telco Network Edge (MEC and Far-MEC)	Support
Telco CO & Regional	Possible
Cloud Edge & CDNs	Cloud Edge – Support; CDNs: Possible
Public Cloud	Support
Private Cloud	Support

Deployment & Industry Verticals (or X; Add context as needed)

Automotive / Connected Car	
Chemicals	
Facilities / Building automation	
Consumer	
Manufacturing	
Metal & Mining	X
Oil & Gas	
Pharma	X
Health Care	
Power & Utilities	
Pulp & Paper	X
Telco Operators	
Telco/Communications Service Provider (Network Equipment Provider)	
Transportation (asset tracking)	
Supply Chain	
Preventative Maintenance	
Water Utilities	X
Security / Surveillance	
Retail / Commerce (physical point of sale with customers)	
Other - Please add if not listed above (please notify TAC-subgroup@lists.lfedge.org when you add one)	No

Deployments (static v dynamic, connectivity, physical placement) - (or X; Add context as needed)

Gateways (to Cloud, to other placements)	
NFV Infrastructure	X
Stationary during their entire usable life / Fixed placement edge constellations / Assume you always have connectivity and you don't need to store & forward.	

Stationary during active periods, but nomadic between activations (e.g., fixed access) / Not always assumed to have connectivity. Don't expect to store & forward.	
Mobile within a constrained and well-defined space (e.g., in a factory) / Expect to have intermittent connectivity and store & forward.	X
Fully mobile (To include: Wearables and Connected Vehicles) / Bursts of connectivity and always store & forward.	X

Compute Stack Layers (architecture classification) - (Provide, Require, or N/A; Add context as needed)

APIs	Provide
Applications	Provide
Firmware	Required
Hardware	Required
Orchestration	Required
OS	Required
VM/Containers	Required

Cloud Stack Layers (architecture classification) - (Provide, Require, or N/A; Add context as needed)

Applications	Provide
Configuration (drive)	N/A
Content (management system)	N/A
IaaS	N/A
PaaS	Required
Physical Infrastructure	N/A
SaaS	N/A

Engineering Plan

Key Components of the Project AI Edge:

- **Project Description and Goals**: AI Edge is designed to integrate AI and IoT at the edge, enabling low latency machine learning inference. This is crucial for real-time, edge-based decision-making and analytics in various sectors, such as manufacturing and healthcare.
- **Project Synergy**: AI Edge complements existing LF Edge projects by providing an AI/ML interface for edge devices, enriching the ecosystem without significant overlap.
- **Contributors and Leadership**: The project has contributions from multiple organizations, including Meta, TikTok, Fujitsu, Arm, Baidu, Allegro, and Edgenesis, with a leadership team consisting of members from these organizations.
- **Technical Requirements**: The project supports architectures like x86-64 and AArch64 and is licensed under Apache 2.0. In the future, we may consider supporting far-edge 32bits architectures.

Steps for Integrating AI Edge into LF Edge:

- **Asset Transfers**: This includes the transfer of trademarks, domain names, social media handles, and GitHub repositories. This step is crucial for aligning the project with LF Edge branding and infrastructure.
- **Integration into LF Edge Materials**: The project should be added to the LF Edge Overview Deck and other relevant resources.
- **Project Logo and Wiki**: Developing a project logo and setting up a comprehensive wiki page are vital for public visibility and documentation.
- **Communication Channels**: Establishing project-specific mailing lists and Slack channels for effective communication.
- **Meeting Calendars and Governance**: Organizing project meetings and setting up a governance model, including the formation of a Technical Steering Committee (TSC).
- **GitHub / Hugging Face Practices**: Implementing recommended practices for GitHub / HuggingFace usage and contribution processes.
- **Technical Charter Review**: Understanding and documenting the mission, scope, and policies as per the project's technical charter.
- **Community and Policy Compliance**: Ensuring open participation, transparent operations, and adherence to LF Edge's code of conduct and policies.

Phase 1: Initiation and Planning (1-2 Months)

- Weeks 1-2: Finalize project description, goals, and initial documentation.
- Weeks 3-4: Start the process of asset transfers (trademarks, domain names, etc.).

Phase 2: Integration and Setup (2-3 Months)

- Months 1-2:
    - Integrate the project into LF Edge materials.
    - Develop and finalize the project logo.
    - Establish communication channels (mailing lists, Slack).
    - Set up the project wiki with initial content.
- Month 3:
    - Organize initial project meetings.
    - Begin forming the Technical Steering Committee (TSC).
    - Implement recommended GitHub practices.

Phase 3: Governance and Community Building (1-2 Months)

- Weeks 1-4:
    - Finalize the Technical Charter.
    - Document the project's governance model.
    - Establish clear roles and responsibilities within the TSC.
    - Kick-off regular TSC meetings.
- Weeks 5-8:
    - Enhance community engagement through active communication channels.
    - Begin compliance with LF Edge policies and code of conduct.

Phase 4: Ongoing Development and Iteration

- Ongoing:
    - Continuous development of the project.
    - Regular updates to the project wiki and documentation.
    - Consistent engagement with the community through meetings, mailing lists, and Slack channels.
    - Regular assessment and iteration of governance and technical strategies.

Key Milestones:

- End of Month 2: Completion of asset transfers and basic project setup.
- End of Month 3: Initial project integration into LF Edge and establishment of communication channels.
- End of Month 5: Formation and activation of the Technical Steering Committee.
- End of Month 6: Establishment of a stable governance framework and active community engagement.

Benchmarking Methodology

1. **Introduction**: A brief overview of the importance of benchmarking in the context of AI at the edge. This section will set the stage by explaining why benchmarking is crucial for assessing the efficiency, performance, and scalability of AI edge technologies.
2. **Benchmarking Criteria**:

    - **Performance Metrics**: Description of the key performance indicators (KPIs) used in benchmarking, such as latency, throughput, power consumption, and accuracy.
    - **Hardware and Software Considerations**: Outline of the hardware platforms (e.g., CPUs, GPUs, Edge TPUs) and software frameworks (e.g., TensorFlow, PyTorch) included in the benchmarks.
    - **Test Scenarios and Use Cases**: Elaboration on various scenarios and use cases for which the benchmarks are applicable, providing context and relevance.
3. **Methodology**:

    - **Test Environment Setup**: Detailed description of the test environment, including hardware specifications, software versions, and network configurations.
    - **Data Collection and Analysis**: Explanation of the process for collecting and analyzing data during benchmark tests.
    - **Reproducibility and Transparency**: Emphasis on the importance of reproducibility in benchmarking, including guidelines for documenting and sharing test procedures and results.
4. **Results and Reporting**:

    - **Presentation of Results**: Guidelines on how to effectively present benchmarking results, including visualization techniques and comparative analysis.
    - **Interpreting Results**: Tips on interpreting results, understanding limitations, and drawing meaningful conclusions.
5. **Continuous Improvement**:

    - **Feedback Loop**: Encouraging community feedback and contributions to refine and enhance the benchmarking methodology.
    - **Updates and Versioning**: Details on how the benchmarking methodology will be updated over time, including version control and change logs.

Work steams

Work stream 1: Geo-distributed Cloud

- **Leaders:** Yona Cao C.C. Fan
- **Objective:** This work stream focuses on optimizing the performance and scalability of cloud infrastructure across multiple geographical locations. It aims to address challenges related to latency, data sovereignty, and efficiency in distributed computing environments.
- **Approach:** The team will develop strategies for seamless data synchronization and application performance across dispersed networks, ensuring robust, secure, and compliant operations globally.

Work stream 2: Edge Database

- **Leader:** Rick Cao Qi Wang
- **Objective:** The Edge Database work stream is dedicated to advancing database solutions tailored for edge computing environments. It targets improvements in data handling and storage capabilities on edge devices, enhancing local data processing and decision-making.
- **Approach:** Efforts will include the development of lightweight, scalable database systems that support real-time data processing and analytics, pivotal for Edge AI Virtual Agents.

Contributor	