Software Engineering for AI/AS

Yangyang Wen

July 19, 2024

Contents

1. My research area

This chapter introduces my research topic and its relevance to Software Engineering principles in AI/AS.

1.1 Abstract and introduction

My research addresses the urgent need for sustainable and energy-efficient computing solutions within Cloud-Edge environments. It introduces a formal model that integrates energy sustainability and consumption considerations into the cloud-edge continuum and workload, primarily focusing on optimizing energy efficiency and resource utilization.

Cloud computing involves the on-demand delivery of the IT resources over the Internet with pay-as-you-go pricing, relying on centralized servers in large-scale data centers. It offers services such as storage, processing, and analysis. However, traditional cloud computing struggles with delay-sensitive real-time applications due to inherent latency issues, especially with the rise of the Internet of Things (IoT).

Edge computing addresses this by placing resources closer to devices and users, for example, putting data to the edge data centers, reducing transmission latency, and improving response time. It processes time-sensitive data locally and supports services like data storage and analysis. Despite its advantages, edge computing cannot handle complex applications and analytics because of the limited quantity in data processing and storage resources.

Fog computing acts as an intermediary layer between cloud and edge, extending cloud capabilities closer to data generation points. It supports AI-driven decision-making and enhances IoT device functionality through local processing. This layered approach, known as the cloud-edge continuum, integrates cloud, edge, and fog computing, enabling distributed data processing, hierarchical storage solutions, and improved application performance.

1.2 Research overview

My research aims to make the cloud-edge continuum more sustainable by reducing carbon emissions while maintaining performance. It focuses on four layers: applications, orchestration, infrastructure, and network.

My research's initial phase concentrates on the network layer within the cloud-edge continuum. By categorizing various energy, network, hardware, and software components within the cloud-edge continuum, the research aims to understand their dynamics and interactions comprehensively. It includes leveraging metrics from the Ericsson Research Data Center and prior efforts of the WCIB project.

In the network layer of the cloud-edge continuum, I have explored measuring and reducing transmission energy costs and using network virtual functions (NVF) and software-defined networking (SDN) technologies. SDN enhances network management by separating the control plane from the data plane. The data plane is responsible for actual data transmission and processing, including packet forwarding, traffic management, and security handling, while the control plane manages network device configuration and administration, performing routing decisions, policy management, and network monitoring. In Software-Defined Networking (SDN), these two are separated, with the control plane managed by a centralized controller and the data plane executed by programmable switches.

In the orchestration layer of the cloud-edge continuum, I aim to improve resource utilization through virtual machine (VM) selection and allocation. I use machine learning to predict workload patterns and classify workloads before resource allocation and migration.

As defined by Wikipedia, *"In software engineering, a microservice architecture is an architectural pattern that arranges an application as a collection of loosely coupled, fine-grained services, communi-*

cating through lightweight protocols" [1]. There are various system architectures existing in the cloud-edge continuum according to practical requirements, such as monolithic architecture and microservice architecture. Monolithic architecture treats the entire application as a whole, suitable for a small application or team. Microservices disparate an application into multiple independent sections, and each section is responsible for a specific function designed for large and sophisticated projects. Generally, one microservice runs in one container to isolate itself from other ones. Container virtualize applications' runtime environment on a single operating system to provide portability and isolation. Containerized microservices can help reduce overhead, faster development and easier adoption of a microservice architecture.

Another direction of my research is optimizing the service mesh layer, which manages communication between micro-services in containerized applications. Despite challenges in deploying and integrating open-source platforms like Istio and Linkerd, the service mesh can significantly enhance resource management and performance.

1.3 Objectives

Through analysis in the early stages of my research, I am exploring various approaches to enhance energy efficiency and reduce carbon emissions within the cloud-edge continuum. The goal is to develop algorithms and tools that integrate into the existing cloud-edge continuum to monitor and optimize energy usage in real time. The results of my research could significantly benefit AI-intensive applications by reducing their energy consumption and improving sustainability, thereby contributing to a greener and more efficient computing environment. The methodology includes creating a model, identifying workload types, and estimating energy consumption within data centers.

2. Principles and techniques from Robert's lectures applied to cloud-edge continuum

In Robert's lectures, I found several valuable and interesting points. These concepts include *"orthogonality," "reversibility,"* and *"prototype learning"* in practical software engineering, as well as *"testing methods"* in software engineering, which include both static and dynamic techniques.

2.1 Orthogonality

First, "orthogonality" refers to the independence and non-interference of components or functions within a system. It emphasizes modular design, low coupling, and high cohesion. In my research of data transmission in microservice applications in a low-carbon way by utilizing mesh service in a microservice architecture, I plan to use the orthogonality principle to design a modular, low-coupled cloud-edge system. It will ensure that each edge server can independently handle its tasks, thereby reducing cross-service data transmission and improving overall system efficiency and response speed.

2.2 Reversibility

Second, "reversibility" enhances the adaptability and maintainability of a system by allowing it to return to a previous state quickly. In my research on creating an energy-aware green cloud-edge continuum, I can achieve reversibility through version control, snapshot technology, and local backup techniques. These ways can avoid the sudden concurrent requests and expense of downloading and transmitting big files remotely to some extent. It will reduce data transmission during fault handling, optimize energy utilization, keep the system coherent and stable, and improve its maintainability and adaptability.

2.3 Prototype learning

Another key concept, the "prototype learning" concept, is applied in machine learning to simplify complex datasets widely. In my research, I will use prototype learning methods, such as the centroids in the K-means clustering method, to make simple data processing on edge devices, transmitting only the necessary data prototypes to the cloud. It will reduce data transmission, save energy, and ensure the efficiency and accuracy of classification and clustering tasks, especially in federated learning.

2.4 Testing methods and techniques

Lastly, testing methods and techniques are pivotal in software engineering and are closely related to reducing data transmission and reception between services in the cloud-edge continuum. Excellent testing processes can make quality assurance improve the quality of the software and design the software efficiently according to timely and comprehensive feedback. In my research on reducing carbon emissions between communications within the cloud-edge continuum, I will adopt static techniques such as code review, static analysis, and formal verification to optimize data transmission logic. Additionally, I will use white-box testing methods such as unit testing and path testing to ensure the internal logic efficiency of the code, reducing unnecessary data transmission and energy consumption. Black-box testing will be employed to test software functionality, ensuring it meets the requirements specifications. Effective testing can enhance the system's efficiency and stability, reducing unnecessary data transmission and energy consumption.

2.5 Summary

In summary, the concepts and methods reviewed in this section will help the cloud-edge continuum minimize data transmission, enhance interaction efficiency and stability, and reduce data transmission during fault recovery. They can further improve performance and maintainability while effectively lowering energy consumption. By adhering to these principles and testing methods to my research on transmitting data within the cloud-edge continuum in a low-carbon way, I aim to build an efficient cloud-edge system.

3. Concepts from guest lectures

In the guest presentation, many points caught my interest. The concepts I chose to focus on from Behavioral Software Engineering are *"Psychological Safety"* and *"Norm Clarity"*.

3.1 Psychological safety and norm clarity in behavioral software Engineering

Psychological safety is crucial for academic research and collaboration. It ensures I can freely ask questions and share new ideas with my supervisor and other researchers without fear of criticism or retaliation. This sense of safety fosters open communication and innovation, helping to solve complex research problems and advance projects.

Norm clarity is also essential for the smooth functioning of a team. Clearly defining roles and responsibilities, research goals and expectations, and communication methods can prevent misunderstandings and conflicts, improve collaboration efficiency, and create a pleasant working atmosphere. Transparent norms help all team members understand their tasks and how they contribute to the project, enhancing cooperation.

3.2 Fostering collaboration in my research

To incorporate these two concepts into my Ph.D. research, I will describe my current practices and plans. I will explain how I apply psychological safety and norm clarity through one-on-one communication and group interactions.

Establishing and maintaining psychological safety is vital for successful collaboration. In one-on-one communication with each team member, I express respect and gratitude, openly share my views and questions, seek feedback, and demonstrate my willingness to learn and improve. In group interactions, I plan to share progress, issues, and opinions proactively, encourage discussions, and create an open dialogue atmosphere. Promptly responding to group messages and appreciating contributions from supervisors and researchers will enhance the team's sense of psychological safety.

Ensuring norm clarity helps define responsibilities and capabilities clearly, fostering mutual understanding and message synchronization. In conversations with each member, which is "one-to-one communication," I discuss roles, expectations, and how I can contribute to our collaboration while understanding their schedules and arrangements. Through mutual communication online and offline, we ensure everyone understands each other's tasks and expectations. Additionally, I will regularly update the team on research progress, challenges, and support needed.

In group interactions, I will set clear communication rules, understand each other's work and schedules, arrange meetings appropriately, synchronize updates, and discuss emergency handling. I will also consider various tools to enhance team efficiency, such as shared documents (e.g., Google Docs), meeting schedules, personal availability timelines, strategic discussion points, and next steps. Additionally, I will consider using git to share my codes and make it available to other researchers. These tools will ensure all members can access historical discussions, the latest progress, and each other's availability, facilitating optimal meeting arrangements.

Until now, my collaboration team has had an initial offline meeting to discuss research goals, roles, and responsibilities, and we've established contact information and group chats. However, efforts to establish regular communication channels are ongoing, and we haven't agreed on communication frequency and methods due to unstable schedules. It has led me to consider incorporating psychological safety and norm clarity into our communication. These concepts can enhance team atmosphere and cohesion, improve collaboration efficiency, and reduce time waste.

Firstly, I plan to strengthen communication with each member before moving on to the future steps, helping them fully understand my work content, research progress, learning goals, schedule, concerns, and plans. Secondly, I will promote information synchronization and consensus among group members, especially on vital timelines and plans. Third, establish convenient communication channels, such as shared project management tools, shared documents, or document management platforms to facilitate easy access and updates. Finally, update shared documents after each meeting, add member feedback, and encourage feedback to ensure nothing is missed or misunderstood.

4. Analysis of two papers from CAIN conference

This chapter analyzes two CAIN conference papers. I picked up two papers. The first one is *"Identifying architectural design decisions for achieving green ML serving"* [2], the second one is *"Green Runner: A tool for efficient deep learning component selection of deep learning components to improve efficiency."*[3].

The AI-intensive software I decided to pick up is the intelligent transportation system. As an intelligent transportation project, it can improve safety, optimize the existing infrastructure, make more connections, and save cost for time and money. The technologies it has been involved in include closed circuit television, vehicle detection, high-speed network, crosswalk detection, Bluetooth, signal coordination, roadside and onboard units, and drone assistance.

Then I will discuss how to apply two papers' core ideas and my research results to assist its improvement and reflect the usability of my research and two papers' core ideas for the reality work.

The first paper, titled "Identifying Architectural Design Decisions for Achieving Green ML Serving," explores how architectural decisions can enhance the energy efficiency of machine learning serving systems. Architectural choices are crucial for the sustainability and efficiency of AI systems. This paper is highly relevant as it addresses the intersection of architecture and energy efficiency, one of the core components of my research focus.

The *"Green Runner"* paper introduces a tool designed to optimize the selection of deep learning components to improve efficiency. The importance of optimizing component selection lies in its ability to create more efficient and effective AI systems, thereby saving resources and energy. The paper is directly relevant to my research as it deals with efficiency improvements, which could be essential for green computing initiatives.

4.1 Identifying architectural design decisions for achieving green ML serving

4.1.1 Core ideas and their importance

"Identifying Architectural Design Decisions for Achieving Green ML Serving" is a preliminary investigation and research. Their ultimate goal is to improve energy efficiency when making decisions related to architecture decisions affecting energy efficiency in machine learning models. Specifically, the study systematically summarizes the current designs of architectures that serve machine learning models. It describes in detail the considerations of architectural quality characteristics in existing contributions from literature. It raises the question of whether energy efficiency receives sufficient attention as one of the quality characteristics in the design decision process. Through an extensive literature review of related literature, the paper identifies the vital design decisions and strategies for creating green ML service architecture designs. These design decisions not only lay a theoretical foundation for their further research but also provide a theoretical basis and direction for other practitioners in exploring ML service architecture design decisions.

The researchers addressed two primary research questions in their literature review: *"(1) What are the architectural design decisions of ML serving? (2) What are the quality factors that literature has studied?"* Regarding the first question, they identified that the main architectural decision in ML serving is the serving infrastructure, categorized into options like 'No runtime engine,' 'Runtime engine,' 'DL-specific software,' and 'End-to-end ML cloud service.' They also noted additional cross-cutting decisions related to the serving infrastructure, such as 'Containerization,' 'Model format,' 'Request processing,' and 'Communication protocol.' In response to the second question, their findings focus on performance efficiency characteristics. They observed limited attention given to the energy consump-

tion aspects of ML serving components. Moreover, insufficient research investigates how choices within serving infrastructures and cross-cutting decisions impact overall quality characteristics.

4.1.2 Relationship to personal research

This paper on green machine learning service architecture design is consistent with my research goals and has complementary technical approaches. My research focuses on reducing carbon emissions during data transmission by optimizing network routing algorithms and leveraging service mesh technology. This is consistent with the goal of achieving green machine learning services through architectural design decisions in the paper. Both are committed to achieving energy conservation and emission reduction goals by optimizing technical means. The paper explores how to reduce energy consumption in machine learning services by optimizing model deployment, selecting energy-saving hardware, and resource management strategies. My research can be used to optimize data transmission paths in ML service architectures, thereby reducing energy consumption and improving the green performance of the system. This paper can help me provide beneficial and effective theoretical support when building the architecture design of cloud edge systems to speed up system design and construction.

Similar to this article, another challenge of my work is to consider good metrics for measuring the sustainability of microservice applications. As a example, I plan to consider carbon emission in addition to energy consumption.

4.1.3 Application in AI-intensive projects

This paper discusses how to achieve green (energy efficient) goals through architectural design decisions when delivering machine learning services. This includes optimizing model deployment, selecting energy-saving hardware, and efficient resource management strategies. Through these design decisions, the energy consumption of machine learning services can be significantly reduced and the overall efficiency of the system improved.

In intelligent transportation systems, the architectural design decisions in this article can be used to optimize the deployment of machine learning services such as CCTV video stream analysis and vehicle detection. Among specific measures, optimized model deployment refers to deploying optimized models on edge devices (such as roadside units and vehicle-mounted units) to reduce bandwidth requirements for data transmission. Choosing energy-efficient hardware means using low-power GPUs or dedicated ML accelerators to process traffic data. Resource management strategies refer to dynamically adjusting resource allocation to optimize the use of computing resources based on changes in traffic flow and events.

My research includes network routing algorithms and service meshes. In transportation systems, it can be used to optimize transmission paths and methods, reduce energy consumption and carbon emissions, improve data transmission efficiency, and enhance system sustainability. Architectural design decisions for green machine learning services can integrate my network routing algorithms to optimize data transmission paths and achieve dynamic resource management more energy-efficiently. Specifically, the optimized network routing algorithm is applied to the deployment of ML services in intelligent transportation systems to reduce carbon emissions when data is transmitted between cloud and edge devices. My algorithm could be used to optimize the transmission path of CCTV video streams and vehicle detection data to reduce unnecessary energy consumption. Combined with the resource management strategy in the paper, the data transmission path is dynamically adjusted, and the most energy-saving transmission method is selected according to changes in traffic flow and events.

4.1.4 Research adjustment suggestions

There are several key areas where adjustments can be made to make my research more consistent with the green machine-learning service principles advocated in the paper. First, improving service mesh algorithms to prioritize energy-efficient data routing paths can significantly reduce the carbon footprint

associated with data processing in intelligent transportation systems. Simplifying the deployment and management of AI models within the system using optimized service mesh configurations will simplify AI engineering tasks. For example, enhancing service discovery mechanisms and load-balancing algorithms can ensure efficient utilization of AI resources, thereby improving system responsiveness and reducing overall energy consumption. By adapting my research to incorporate green machine learning principles into AI-intensive transportation projects, we can not only improve operational efficiency but also make positive contributions to environmental sustainability. These adjustments are designed to make AI more efficient and environmentally friendly, in line with the development trend of modern green technologies.

4.2 Green Runner: A tool for efficient deep learning component selection

4.2.1 Core ideas and their importance

"Green Runner: An Efficient Deep Learning Component Selection Tool" introduces a green and energy-saving deep learning component selection tool developed by the author. The tool uses the reasoning power of large language models. Through the user input of the problem description and the expected trade-offs, the tool selects a series of deep learning components to balance resource efficiency, so the problem is solved while meeting expectations. The core ideas of the tool include energy efficiency evaluation, optimization selection, and deployment recommendations, aiming to help developers select models and components with the best energy efficiency, thereby reducing energy consumption. The tool has been preliminarily proven to reduce the amount of computation compared to brute force methods and provide better results than ad-hoc methods.

4.2.2 Relationship to personal research

This article on green deep learning component selection tool has a common energy-saving goal with my research and has complementary technologies. The paper introduces a tool for selecting efficient deep learning components to improve overall efficiency and reduce energy consumption. My research is also committed to reducing carbon emissions by optimizing network transmission paths and microservice communication methods. My research results can be combined with the component selection function of the Green Runner tool to provide a more comprehensive system optimization solution. By working together on both the component selection and data transmission path optimization levels, the overall efficiency of the system can be significantly improved and carbon emissions can be reduced. The Green Runner tool can be used to evaluate and select the most energy-efficient deep learning components, while my service mesh technology can ensure that data transmission between these components is carried out in a way that minimizes carbon emissions.

By adopting Green Runner's approach, the optimized selection and deployment process of deep learning models can aid in improving the development efficiency of deep learning model design and energy efficiency.

4.2.3 Application in AI-intensive projects

This paper introduces a tool to help select efficient deep learning components to improve overall efficiency. The tool analyzes and evaluates the performance and energy consumption of different components and provides optimization suggestions. By using this tool, more efficient deep learning models and hardware can be selected to reduce energy consumption and improve computational efficiency.

In various detection tasks (such as vehicle detection and pedestrian crossing detection) of intelligent transportation systems, the Green Runner tool can be used to select efficient deep learning models and components. The Green Runner tool could be used to analyze the performance and energy consumption of different deep learning models and select the most suitable model for real-time traffic monitoring.

According to the tool's recommendations, the most energy-efficient hardware configuration is selected to achieve a balance between the best performance and the lowest energy consumption.

My research can be integrated into intelligent transportation systems. The Green Deep Learning Component Selection Tool can combine my research results to evaluate transmission efficiency and optimize component selection. Specifically, we use efficient deep learning models for real-time traffic monitoring and incident detection, reducing unnecessary energy consumption during calculation and transmission. We also select efficient deep learning components in conjunction with the Green Runner tool to improve the efficiency and accuracy of drone-assisted traffic monitoring.

4.2.4 Research adjustment suggestions

There are several key areas where adjustments can be made to better align my research with the principles encompassed in Green Runner. First, improving the service mesh algorithm to incorporate energy-efficient deep learning component selection criteria can significantly reduce the overall energy consumption of the system. Simplifying the integration and management of energy-efficient deep learning components by optimizing the service mesh configuration will simplify AI engineering tasks in the project. Enhancing the orchestration of AI model deployment and workload distribution across the network can improve computational efficiency and system responsiveness while reducing environmental impact. Incorporating the energy-efficient deep learning component selection principles from Green Runner into AI-intensive transportation projects is expected to improve operational efficiency and sustainability. These adjustments are designed to make the AI engineering process more efficient and environmentally friendly, reflecting the commitment to using cutting-edge technologies to achieve greener solutions.

4.2.5 Summary

These two papers provide practical energy efficiency improvement methods and tools in the field of machine learning and deep learning, which are of great reference significance to my research. Applying these results to the intelligent transportation system can greatly improve the energy efficiency and overall operation efficiency of the system. My research focuses on optimizing energy consumption in the cloud-edge continuum. In terms of architecture design and resource management for paper 1, my research can further optimize data transmission paths and reduce energy consumption. In terms of component selection and performance evaluation for paper 2, my research can provide a low-carbon data transmission solution and improve the green performance of the overall system. Currently, I focus on network routing algorithms to reduce carbon emissions during data transmission, further supporting the goal of green AI engineering. By combining the core ideas of these two papers, my research can not only create more efficient AI systems in the cloud-edge continuum but also actively promote the development of environmental sustainability.

References

[1] Wikipedia contributors. Microservices, 2024. [Online; accessed 18-July-2024].

[2] Francisco Durán, Silverio Martínez-Fernández, Matias Martinez, and Patricia Lago. Identifying architectural design decisions for achieving green ml serving. In *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering-Software Engineering for AI*, pages 18–23, 2024.

[3] Jai Kannan, Scott Barnett, Anj Simmons, Taylan Selvi, and Luis Cruz. Green runner: A tool for efficient deep learning component selection. In *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering-Software Engineering for AI*, pages 112–117, 2024.