Scope
Volume 14 Number 04 December 2024

An Operating System for the Rise of AI Technology

Dr. Surekha Lanka[1], Latifa Amar Al-Nadhdy[2] , Nan Su Yee Htike[3]
[1]Bangkok, Thailand, [2]Tanzania, [3]Yangon, Myanmar

Abstract: This research paper explains the importance of operating systems for Artificial Intelligence based systems and as we are aware that AI technology growing fast and many of them trying to adapt the new technology to make their tasks easier. So, in this paper we will be explaining how operating systems could support and make AI computing better in terms of performance, security, and efficiency. Any system collaborating with AI to enhance users' experience and optimize the performance of system. It specifies the software system pledged for management of software and hardware resources. This paper aims to understand exactly what AIOS is and what are the required features that make AIOS, not only that but also get to know what it means for the future of artificial Intelligence based operating systems in various fields. Eventually more clearly understanding on what the current performance of AIOS and its framework and concentrates deeper into why AI-enhanced operating systems are essential for bringing new ideas and better performance to computer processes.

Keywords: User-Interface, AIOS- Artificial Intelligence operating systems, resource management, framework, Federated Learning (FL), Multi-agent system architecture (MAS)

Research Questions

In the section we are introducing the research questions, which our paper aims to explore AIOS.

1. What AI-driven developments impact the performance of operating systems in latency, speed and throughput?
2. What strategies can be used to encourage cooperation and knowledge sharing among AI-based systems functioning together?
3. What strategies can be employed to dynamically allocate resources based on changing workloads and priorities?

Background

Operating Systems (OS) essentially determine how users interact with computers, impacting performance and multitasking abilities and operating systems have been around since way back in 1956. Therefore, Artificial Intelligence has been one of the most astonishing subjects in the past recent years, with many research and projects including how it has an impact on enhancing Operating systems with Artificial intelligence. Although many are reluctant and dispute this AI-based innovation due to several concerns about instability and automation bias, humans cannot deny that it helps people in daily life.[1]. Therefore, securing and affirming that Operating systems with Artificial intelligence will have efficiency to reduce some risks regarding the current issues. In this review paper, we will contribute knowledge-based information according to references of efficiency and user performance with A.I.

Initially, In the world of Artificial Intelligence, tasks usually require more computing power than everyday operations like opening or reading files. Imagine from the computer's perspective, running a machine learning task is like transferring a high-resolution image from the disk to memory using the graphics card. This comparison makes sense because both tasks need similar amounts of computer resources. Having more context switches doesn't necessarily mean better user experience or improved CPU performance. In fact, frequent context switches can make individual processes wait longer. Each context switch comes with its own overhead, using up CPU time for each switch, which means valuable processor time is lost. [2]. It could slow down the process.

Moreover, AI technologies can greatly expand the possibilities for robotics autopilot systems, boosting their accuracy, efficiency, and safety. By using AI, these systems can process data from sensors and cameras in real-time, helping them make smarter decisions and navigate changing conditions more effectively. [4]. At the same time, Machine Learning (ML) has been rapidly evolving. It started with mathematical formulas and now we have practical software assistants, which is quite a leap from the early days of operating systems. Combining these two fast-moving fields in Computer Science could bring big benefits to both.

According to author Kulkarni & Kamble [2], Another problem with modern operating systems is that, deep down, they treat all programming languages the same. This means that, for example, in a Linux-based system, the kernel doesn't distinguish between C, C++, or Python. In simpler terms, operating systems treating high-level and low-level languages alike is another issue. Furthermore, the selection of a model is a critical decision with significant long-term implications. Opting for an appropriate model has the potential to enhance system performance and efficiency, whereas selecting an inappropriate one may result in decreased performance and efficiency.

Furthermore, the fact that our existing platforms have not progressed much beyond the computer architectures, operating systems, and programming languages of the 1960s and 1970s. During that era, the computer landscape differed significantly from today's environment in several critical aspects: Firstly, computers were notably constrained in terms of speed and memory capacity. Secondly, they were predominantly utilized by a relatively limited community of technically proficient and trustworthy users. Thirdly, connections to networks or controlled objects were rare occurrences. Nevertheless, contemporary progressions in computer technology, spanning architectures, operating systems, and programming languages, have not undergone sufficient transformation to activate fundamental shifts in computer usage. Furthermore, the urgency of the situation is also growing due to the rising number of implemented artificial intelligence applications. [3].

Lastly, this paper also suggests a simple solution for managing resources while Artificial Intelligence (AI) runs in the background: allocating dedicated hardware for this purpose. The idea is that this hardware won't use up the resources meant for users because only the AI and the operating system kernel can access it.

Introduction

Artificial intelligence (AI) integration into operating systems is a revolution in user experience and resource management. Through Artificial intelligence, OS can adapt to user behavior, anticipate needs, and streamline tasks, enhancing efficiency and productivity. AI also can optimize resource allocation, dynamically managing system resources like CPU, memory, and storage, leading to better performance and responsiveness. Overall, AI in operating systems holds the potential to deliver personalized experiences and efficient resource utilization, transforming the way users interact with their devices. At the core of AI-powered operating systems lies the ability to analyze vast amounts of data to gain insights into user behavior, preferences, and habits. By leveraging the machine learning process, operating systems can continuously learn and adapt to individual users, offering personalized experiences that cater to their specific needs and requirements as well as preferences. For example, an AI-powered OS can learn a user's most frequently used applications and prioritize them on the desktop or taskbar for quick access, thereby reducing the time and effort required to locate essential tools.

According to author [5], understanding how users perceive and expect from AI-based systems is essential for designing interfaces that are effective and user-friendly.

Users' perceptions and expectations are shaped by their past experiences and mental models, which can vary significantly across different user groups. Therefore, conducting thorough user experienceresearch is imperative to gain insights into these diverse perspectives. Various interaction principles have been developed over time to facilitate smooth interactions between humans and computers. These principles serve as guidelines for designing interfaces that are intuitive and easy to navigate, ensuring a positive user experience. However, with the integration of AI into systems, traditional user roles may undergo significant changes.

In traditional computing systems, users typically act as controllers, making decisions based on the information presented to them. However, in AI-enabled systems, users may transition into a more passive role, becoming beneficiaries of the system's automated decisions. This shift in roles necessitates a reevaluation of the user's role within the system. Defining this new role for users in AI-driven systems is essential and must be tailored to the specific context and requirements of each system. Depending on the nature of the tasks performed and the complexity of the AI rolesinvolved, users may need to adapt to new ways of interacting with the system. For example, users may need to trust the system's decision-making capabilities and rely on its recommendations, rather than actively controlling every aspect of the process. [6]

Moreover, the integration of AI introduces new challenges and considerations for user experience design. Designers must ensure that AI-driven systems are transparent and explainable which allows users to understand the reasoning behind the system's decisions. Additionally, measures must be taken to reduce potential biases and ensure fairness in the system's outcomes. Moreover, the AIOS (Artificial Intelligence Operating System) represents a fusion of an Intelligent Agent with an Operating System. This integration harnesses the power of cloud computing and high-speed internet connectivity to operate seamlessly across all platforms belonging to an individual user. By leveraging these technologies, AIOS facilitates uninterrupted connectivity with the user's devices. The core concept behind AIOS lies in its ability to unify various computing environments under a single umbrella, allowing users to access their applications and data seamlessly across different devices and platforms. This means that users can transition seamlessly from their desktop computer to their smartphone or tablet without experiencing any interruptions in their workflow. [7]

One of the key factorsof AIOS is its ability to provide consistent user experience across all devices. Regardless of the device being used, users can expect the same level of functionality and performance from their applications and services. This not only enhances productivity but also simplifies the user experience by eliminating the need to learn different interfaces or workflows for different devices. Additionally, AIOS enables users to take advantage of the processing power and storage capacity available in the cloud. This allows for more efficient use of resources and enables users to access their data and applications from anywhere with an internet connection. AIOS represents a significant leap forward in the realm of operating systems, offering users a seamless and unified computing experience across all their devices. By leveraging cloud computing and high-speed internet connectivity, AIOS empowers users to stay connected and productive in an increasingly interconnected world. [8]

Furthermore, in resource management such as handling communications between machines, decision trees can be employed to organize data, facilitating its distribution across the network. Similarly, reinforcement learning can serve similar purposes, especially when the system's status can be categorized into discrete states. Another approach involves the distributed operating system constructing models of the system dynamically using model learning, enabling it to detect and address resource or communication failures. Not only that, but a neural network can also be integrated into a distributed operating system and trained to recognize known security threats. Due to its ability to generalize, a neural network may effectively handle both known and previously unknown attacks. [7].

Dynamical allocation and rouse management in AIOS

The integration of Artificial Intelligence (AI) into operating systems represents a significant advancement in user experience and resource management. By leveraging AI capabilities, operating systems can adapt to user behavior, anticipate needs, and streamline tasks as well as leading to enhanced efficiency and productivity. Moreover, AI-driven resource management optimizes system resources dynamically, resulting in better performance and responsiveness. Integrating AI into operating systems brings new challenges for user experience design. Designers must ensure that AI-driven systems are transparent, explainable, and fair. Yet, AI integration has the potential to transform how users interact with their devices, empowering them to stay connected and productive in our highly interconnected world.

Toenhance user experience, AI plays a crucial role in optimizing resource management within operating systems. Traditional resource allocation methods often rely on static configurations that may not fully utilize available hardware resources or adapt to changing workload demands. However, AI-driven resource management enables operating systems to dynamically allocate system resources, such as CPU, memory, and storage, based on real-time. Additionally, AI enables operating systems to anticipate user needs and practively assist with routine tasks. By analyzing patterns in user behavior and contextual data, such as time of day and location. Theseintelligent systemscan predict upcoming actions and provide relevant suggestions or automated actions. For instance, if a user frequently sends emails to a particular contact during the morning hours, the AI-powered OS can suggest composing an email to that contact as soon as the user logs in, saving time and streamlining the workflow [6].

Usage patterns and workload characteristics. For example, during periods of high demand, AI processes can intelligently allocate additional CPU resources to critical tasks, ensuring smooth performance and responsiveness. Conversely, during idle periods or low activity, AI can optimize resource usage by reallocating surplus resources to other tasks or entering low-power states to conserve energy. This dynamic resource allocation not only maximizes system performance but also improves energy efficiency, extending battery life in mobile devices and reducing operational costs in data centers.[5].

Current versus AI based Operating's systems.

Let's observe another Integration of big data system with AI system, which big are very popular in current era. The illustration below shows how AI uses large amounts of data and strong infrastructure to perform complicated analysis to identify useful information [9]. The data infrastructure and acquisition channels provide the required input, which is big data with lots of different characteristics, which is then processed using AI models and statistical approaches to generate intelligent systems capable of making predictions, optimizing processes, and improving decision-making. In other words, Big Data provides the large datasets required for AI, while AI provides the advanced analytics to extract value from this data, enabling improvements in a variety of sectors and enhancing system intelligence and responsiveness.

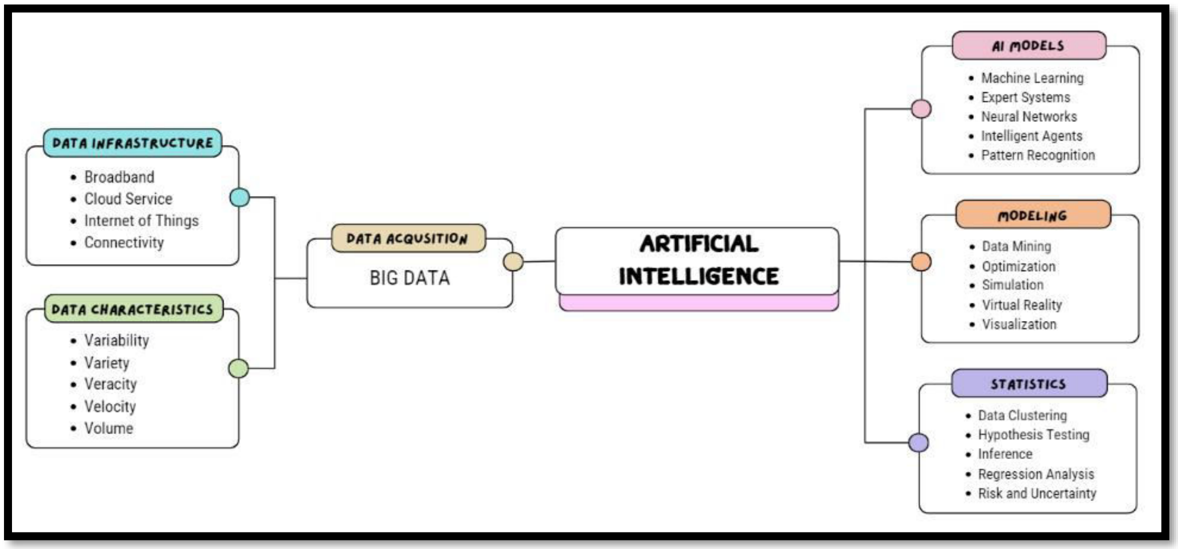

Figure 1: Integrated elements and processes involved in big data and artificial intelligence.

Current operating systems like Windows, macOS, iOS, and Android, operate on fixed logic patterns and are designed for a wide range of users, which can make them less personalized and flexible [9]. These systems have problems like app crashes, bad design, and not being able to change in real-time. AI Operating Systems (AIOS), on the other hand, implement and integrate artificial intelligence into their architecture. This implementation has made it possible for them to offer more dynamic and personalized experiences that change based on the way users interact with it. Therefore, it can be said that AIOS systems are different from other systems because they are always learning and improving based on the needs of each user. Not only that, but they also have features such as understanding context, predicting the future, and managing resources based on individual user needs.

AI Driven Developments on Operating systems

Current operating systems like Windows, macOS, iOS, and Android, operate on fixed logic patterns and are designed for a wide range of users, which can make them less personalized and flexible [9]. These systems have problems like app crashes, bad design, and not being able to change in real-time. AI Operating Systems (AIOS), on the other hand, implement and integrate artificial intelligence into their architecture. This implementation has made it possible for them to offer more dynamic and personalized experiences that change based on the way users interact with it. Therefore, it can be said that AIOS systems are different from other systems because they are always learning and improving based on the needs of each user. Not only that, but they also have features such as understanding context, predicting the future, and managing resources based on

individual user needs. Apart from the AIOS, it is also crucial to take note of the involvement of Artificial Intelligence in various fields and industries. One of the most noticeable advancements could be seen in the Software Development field. The use of artificial intelligence (AI) in the software development process is commonly known as "AI-driven development." This method uses machine learning algorithms and natural language processing to help developers understand and even generate code. This makes their jobs easier and improves the quality of the software that they create [10].

Additionally, AI-driven tools can produce components of code or entire.programs automatically from high-level descriptions provided by the developers. For example, a developer may define a function in plain English, and the AI tool will convert that description into executable code. The development of this speed minimizes manual coding errors and enables developers to concentrate on more complicated elements of program design.

However, it is also important to note how the development of operating systems (OS) significantly impacts the key performance metrics like latency, task scheduling, speed, and throughput. These measurements are critical to the efficiency and responsiveness of both general-purpose and real-time systems.

Latency: Refers to the duration between a user's input and the corresponding response from the system. It is measured in time units, such as milliseconds, microseconds, or nanoseconds [11]. Low latency is very important for ensuring that the user experience is smooth and fast. In other words, lower latency means faster response times, which is critical for applications like video conferences, online gaming, and real-time data processing [12].

Task scheduling:_Furthermore, AI-driven has an impact on predictive task scheduling. Predictive analytics enables operating systems to anticipate user needs and preload applications and data before they are needed. This decreases the waiting time for consumers and reduces latency. For example, by predicting which programs a user is likely to utilize based on the previous data, the OS may ensure these applications are available to use immediately [13].

Speed: The key impact on the operating system's performance is speed. Speed relates to how rapidly a system can complete tasks and operations. This depends on several factors like CPU utilization, memory access speed, and disk I/O performance. In other words, it relates to how the operating system handles CPU cycles, memory access, and other essential resources. AI can increase operating system speed by optimizing resource allocation and ensuring that high-priority tasks receive the required computational power. AI can also improve operating system performance by dynamically managing system resources. Not only that, but Machine learning algorithms can also forecast the resources required to perform incoming tasks and distribute CPU, memory, and storage accordingly. This proactive resource management reduces the overhead and latency associated with resource contention, enhancing the overall speed of the system [14].

Throughput: As the number of operations or amount of data processed per unit of time [11]. The efficiency of the operating system throughput can be increased using AI-driven developments by optimizing resource management and parallel processing. Throughput also can be improved overall by using machine learning to assess workload trends and allocate tasks among resources in an effective way. A high throughput refers to the operating system is efficient and can handle many jobs at once. If the throughput is high and the latency is low, the operating system is fast and efficient. If the throughput is low and the latency is high, the operating system is slow and inefficient [15].

Throughput also can be improved overall by using machine learning to assess workload trends and allocate tasks among resources in an effective way. A high throughput refers to the operating system is efficient and can handle many jobs at once. If the throughput is high and the latency is low, the operating system is fast and efficient. If the throughput is low and the latency is high, the operating system is slow and inefficient.

AI engine simulation for Latency and Throughput

Launching and thread synchronization is overhead in the environment of multi-threaded process. Normally, the threads will be oversight during the simulation of AI engine. So thoroughly the calculation on individual threads must be consequential when correlated to the overhead. In the simulator command at first constructs the specific compiler arising from work/config/scsim_config.json file. Which includes the loading block of IP and establish their connections and configure I/O drivers and DDR memory to execute the application and quit the simulator.

Normally in the simulator average throughput undoubtedly calculated over the duration of simulator environment and produce output at the end of AI engine simulation. AI

Engine simulation generates Latency, Continuous Latency and Continuous Throughput estimates.

```
aiesimulator --pkg-dir=. /Work --dump-vcd foo --options-file=aiesim-options.txt
```

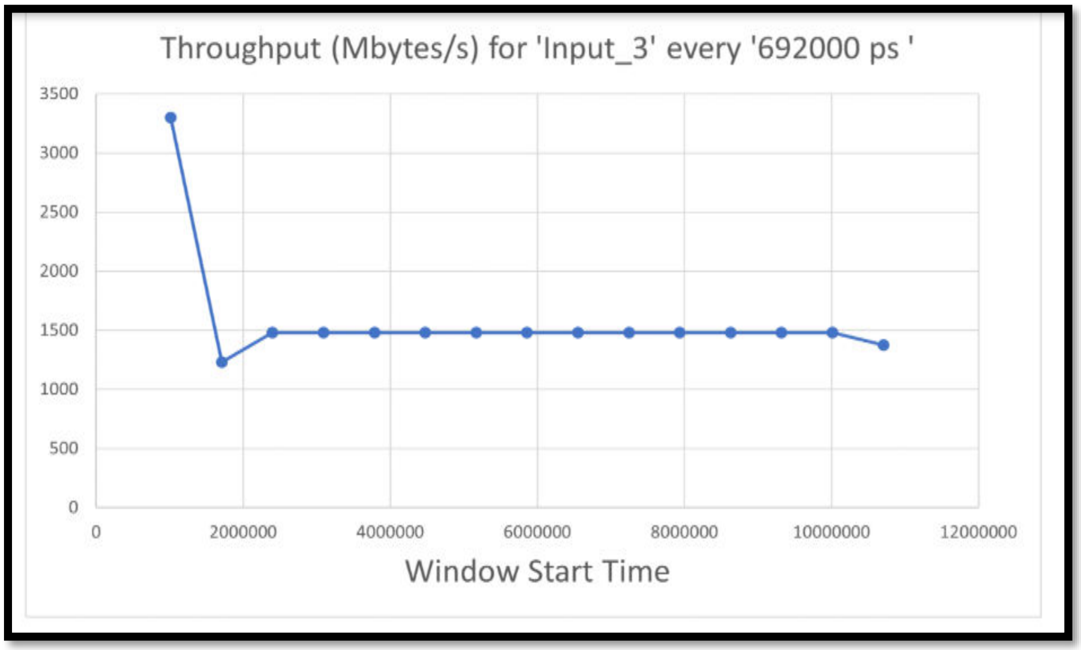

Fig 2: Graph of Throughput from AI engine simulator

Another thing with lengthened latency is the dynamic expansion of the Latency between ports must be specified with Input and output. IDE enables us to perform analysis of latency and throughput. Beginning default.run_summary file using vitis_analyzer will produce the measurement of latency for the Array of AI Engine Array in the Trace section within the Latency tab.

```
vitis_analyzeraie/aiesimulator_output/default.aierun_summary
```

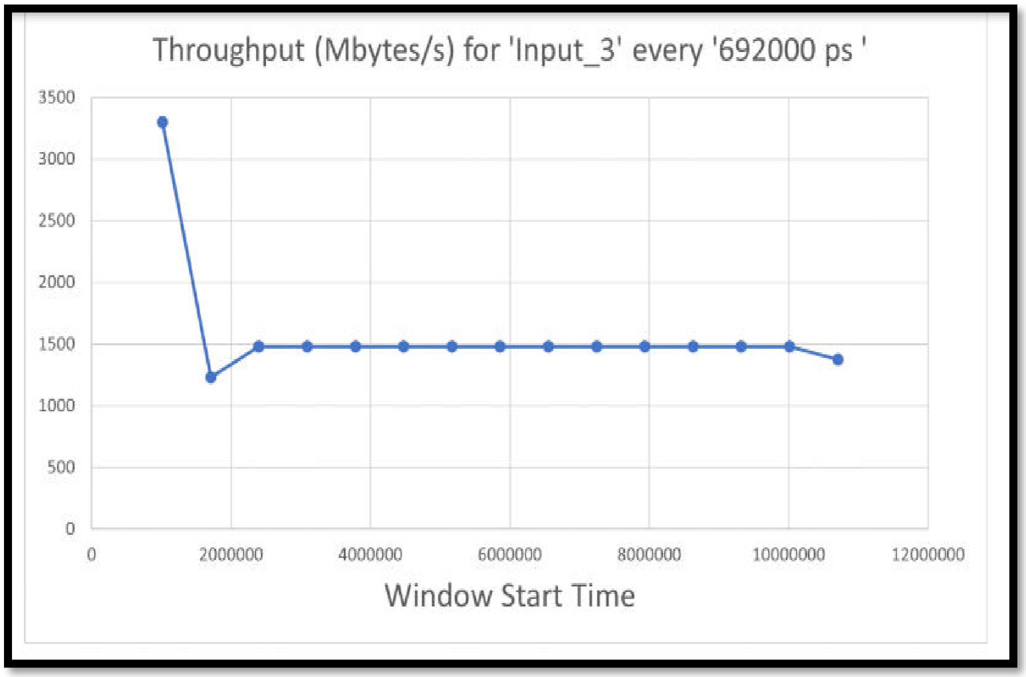

Fig 3: Graph of Latency from AI engine simulator

Strategies tocooperate and knowledge sharing among AI-based systems
AI-based systems are the essential tool used across multiple domains to solve complex problems and enhance decision-making processes. However, the true potential of AI-based systems can be unleashed by using various strategies approaches.

Strategy 1- establishinteroperability standards: AI-based systems are crucial as they often rely on complex and huge data sources as well astechnologies. According to researcher [16], implementing the interoperability for AI systemsnot only to perform seamless data exchange, information and resources to support decision-making, but also work automation and collaboration across several departments and functions among an organization. Therefore, it is important to take note of how establishing interoperability is crucial for leveraging the cooperation and knowledge sharing among the AI-based systems functioning together.

Strategy 2- Federated Learning (FL): Thisis another key strategy to encourage cooperation and knowledge sharing among AI-based systems functioning together. Federated learning refers to building machine learning models using data sets that are distributed across several devices while preventing data leakage [17]. In other words, it allows multiple entities to cooperate in training a machine-learning model without sharing their local data. This technique ensures the confidentiality of data and encourages the participation of entities without the risk of data breaches or violating privacy regulations

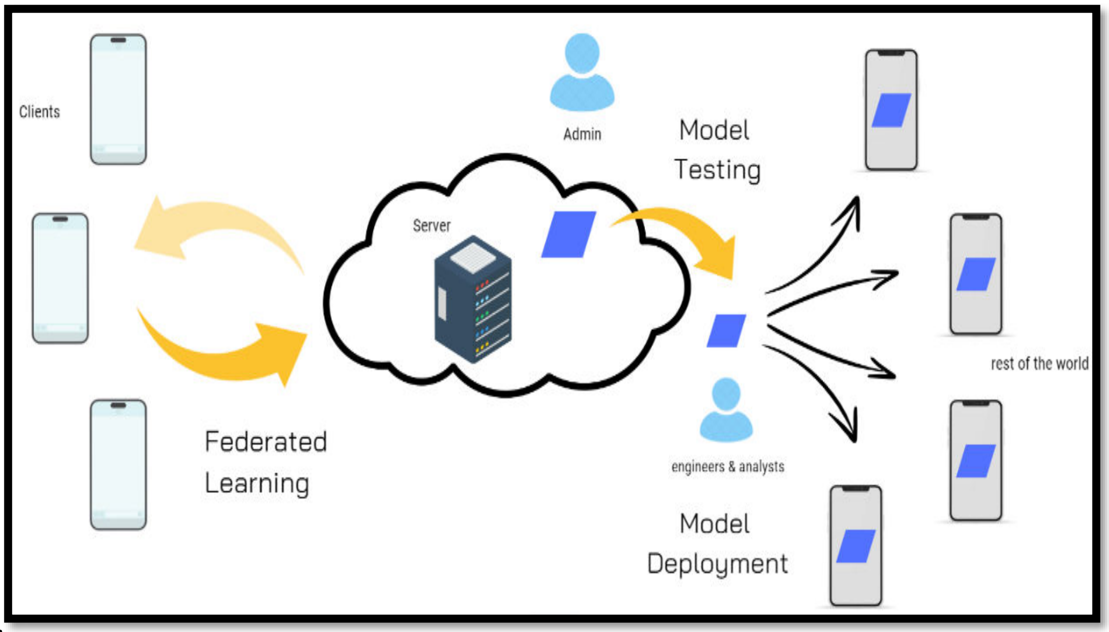

Figure 4: The lifecycle of the Federated Learning trained model and the various actors in a federated learning system.

Functionality of Strategy 2:

Figure 4 indicates flow of FL performance by clients in the lifecycle of a Federated Learning (FL) trained model and various actor roles in federated learning system [18]. How will the central server handle model testing, how administrators, engineers, and analysts test the models, and how end users deploy the models. The process begins with several clients (e.g., mobile devices or edge devices) each storing local datasets. Each of these clients trains a local model using their own data. This server collects updates from all participating clients to form a global model. This aggregation process ensures that the global model benefits from the different data distributed among clients without requiring access to the raw data itself. Many actors, including an administrator, engineers, and analysts then test the aggregated global model to ensure its robustness and performance. Normally Latency is ideally the server would be stored in the same region as the individuals performing the project however even if the server was physically located in other region, the API calls would at the most take a few seconds maximum[27]. Once the model has passed the testing phase, it is returned to clients and maybe to other end users globally, so they can use the collaboratively trained model.

Strategy 3 multi-agent system architecture: This strategy for enhancing interoperability among AI-based systems. In multi-agent system exists with multiple agents of decision makers, who can interact in a shared environment to achieve a common or conflicting goal. Multi-agent system architecture [MAS] can increase flexibility and efficiency among the AI-based systems as it can help adapt to various changing environments and requirements as well as dividing tasks among multiple agents, allowing it to find solutions quicker than a single agent [19]. Nevertheless, designing incentives mechanisms within the MAS framework can also motivate the individual AI agents to collaborate and share knowledge through offering rewards and privileges [20]. Thus, helping the AI-based systems to work together seamlessly.

Architecture and workflow of MAS

As we discussed in the previous section implementing a multi-agent system architecture also serves as a strategy for enhancing interoperability among AI-based systems. Multi-agent system architecture [MAS] can increase flexibility and efficiency among the AI-based systems as it can help adapt to various changing environments and requirements as well as dividing tasks among multiple agents, allowing it to find solutions quicker than a single agent [19]. Not only that but designing incentives mechanisms within the MAS framework can also motivate the individual AI agents to collaborate and share knowledge through offering rewards and privileges [20]. Thus, helping the AI-based systems to work together seamlessly.

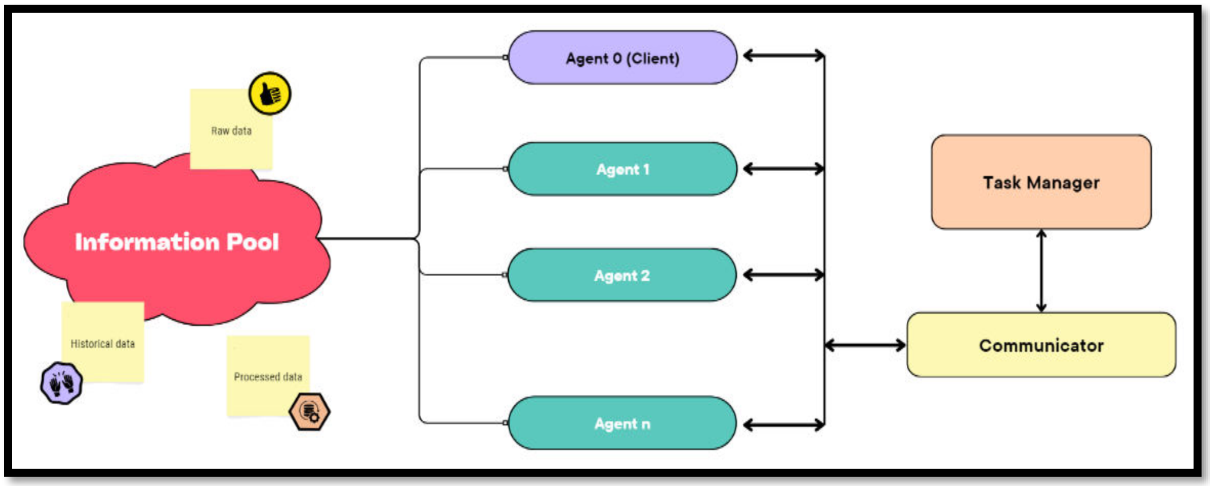

Figure 5: Multi-agent system for decision support

Figure 5 shows an intelligence agent system that is run by a task manager and communicator that can gather information and take actions on its own. The task manager breaks down problems into smaller jobs, arranges data, and communicates to the user while the communicator's responsibility is to give agents input in a way that they can understand and to get useful output from agents in the system. Agent can be either software, hardware, or a human. Each agent has specific features such as behavior, data, goals, and motivation. Agents monitor and respond to specific data using several different ways [21].

Dynamic Resource Allocation in AIOS

An AI-based operating system can use many methodologies that use AI and machine learning techniques to dynamically allocate resources as well as to optimize the distribution of system resources like CPU, memory, and storage among applications. One of those is using machine learning for predictive analytics to predict the resource requirements of applications using historical data and current usage patterns [22]. This allows the AIOS to pre-allocate resources more efficiently, reducing latency and improving performance during peak usage periods. Reinforcement learning (RL) is also another effective technique that can be used to adapt resource allocation in real-time. It is a technique that trains the system to make decisions based on the outcomes that would lead to the best results [23]. In other words, it is about learning the appropriate behavior in an environment to gain the highest reward. Reinforcement Learning is an efficient method to dynamic resource allocation in AIOS that uses trial-and-error learning to generate adaptive solutions [24]. RL can handle uncertainty and learn from interactions which makes it an effective tool for improving resource allocation without explicit job modeling.

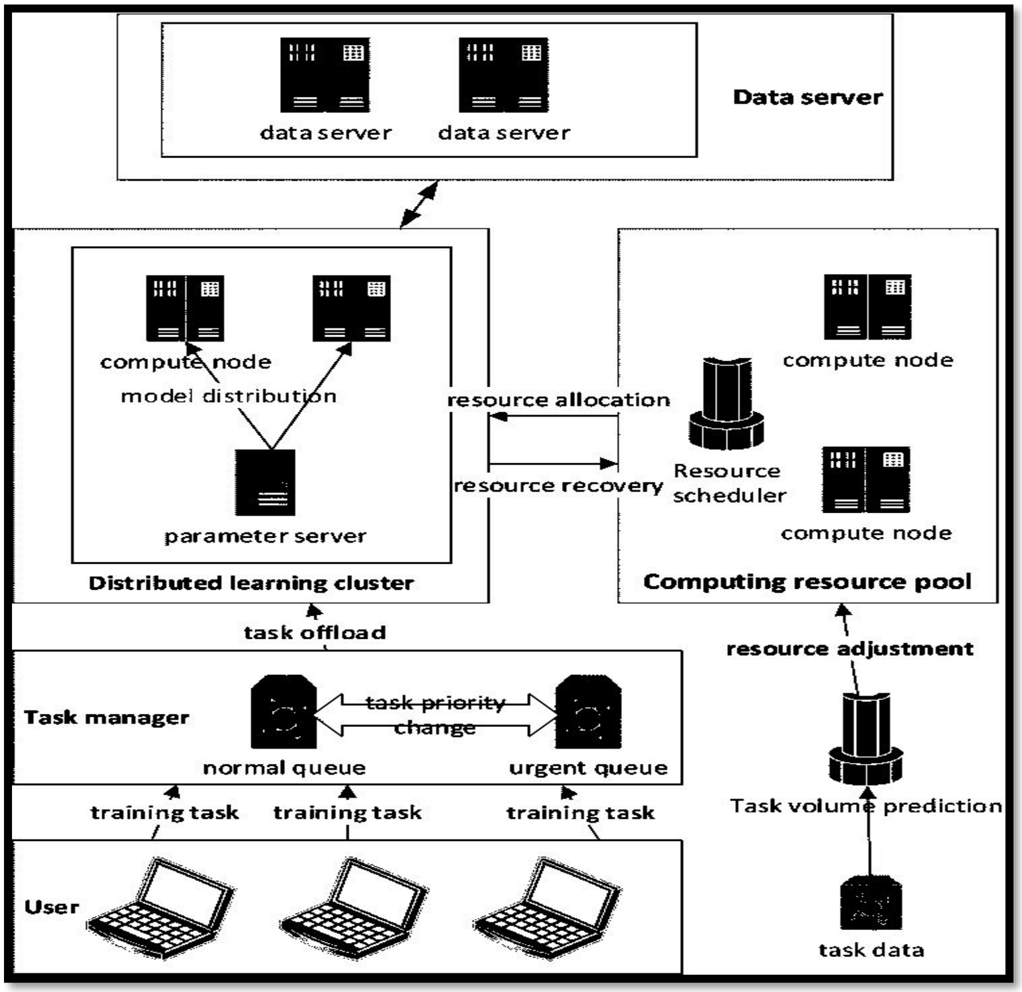

Fig 6: Dynamic resource allocation in the application of AI

Resource Allocation in Terms of C hanging Workloads and Priorities

When it comes to resource allocation within the AI-based operating system, work allocation and scheduling use AI to adapt and be flexible to any demands through continuously monitoring system performance and workloads. By doing so, it ensures that the resources are distributed efficiently as well as effectively so that it meets the needs of intensive tasks within the systems. Moreover, the AI algorithms integrated within the operating systems are made to predict future resource requirements based on both the historical data and real-time usage patterns so that the system can anticipate those changes beforehand and adjust the distribution of the resources among the tasks. Additionally, the AI algorithms prioritize the queue of tasks according to the urgency and importance of the task so that necessary and important resources can be distributed promptly to the more critical tasks while less critical tasks are allocated resources during the lower demand periods within the system [25]. Not only that but also unlike non-AI-based operating systems, where the resource allocation is managed through static and rule-based approaches, AI-based OS integrated reinforcement learning to adapt to new patterns of resource demands within the system, thus maintaining the optimal operational performance. Moreover, the reinforcement learning technique even allow the system to learn from its past decisions and changes its strategy to adapt its task prioritization as well as resource allocation over time which makes the performance and responsiveness of the system to improve continuously over time [26].To conclude, in AI-based operating systems, they used the AI algorithms mostly integrated reinforcement learning to carry out real-time workload monitoring so that the system can analyzes the current workload patterns as well as historical data to predict what the future resource demands and task priorities will be and flexibly adapt to any fluctuating changes.

Conclusion

In our opinion AI based operating systems are not only for the better user experience but also through use of cloud technologies,which holds significant potential in our modern world. to that enables synchronization across multiple devices, learning from interactions and personalization from us (users), and managing resources to make the system performance better. All in all, AIOS makes our interactions with technology smarter, faster and seamless.Therefore, it is beneficial to continue improving and developing AI-based operating systems to fully harness their capabilities and drive progress for our future world. So as per this paper we understand that compared with the normal operating system, AI based systems are better, especially in resource allocation and deciding priorities for tasks since it integrates AI algorithms like reinforcement learning in which the system's performance can learn and grow continuously on its own and adapt to changes. Where the Interoperability standards, shared learning, and multi-agent systems are all strategies that help these AI systems work together and share information, which makes them more useful overall.

References

1. Kulkarni, Mayuresh & Kamble, Torana. (2020). Integration of Machine Learning into Operating Systems: A Survey. 1270. Retrieved from (PDF) Integration of Machine Learning into Operating Systems: A Survey
2. Soori, Mohsen & Arezoo, Behrooz &Dastres, Roza. (2023). Artificial Intelligence, Machine Learning and Deep Learning in Advanced Robotics, A Review. 10.
3. Brand, Lisa & Humm, Bernhard & Krajewski, Andrea & Zender, Alexander. (2023). Towards Improved User Experience for Artificial Intelligence Systems.

4. Fomin, Vladislav. (2021). The shift from traditional computing systems to Artificial intelligence and the implications for bias.
5. Banafa, Prof. A. (2024, April 21). Objective-driven AI: Optimizing for specific goals in a complex world.
6. Nadun, C. (2023, December 21). AI Operating Systems - Unleashing the Future of Technology. Chanuka Nadun Perera.
7. Takyar, A., &Takyar, A. (2023, November 29). AI-driven development. LeewayHertz - AI Development Company.
8. National Instruments Corp. (2024, April 23). More Throughput vs. Less Latency: Understand the Difference. NI.
9. Brand, L., Humm, B. G., Krajewski, A., & Zender, A. (2023). Towards Improved User Experience for Artificial Intelligence Systems. Engineering Applications of Neural Networks, 33–44.
10. Kulkarni, M., & Kamble, T. (2020, April 1). Integration of Machine Learning into Operating Systems: A Survey.
11. Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated Machine Learning: Concept and Applications.
12. Kairouz, E. B. P., & McMahan, H. B. (2021). Advances and Open Problems in Federated Learning.
13. Guo, Q., & Zhang, M. (2009). A novel approach for multi-agent-based Intelligent Manufacturing System. Information Sciences, 179(18), 3079–3090.
14. Shakshuki, E., & Reid, M. (2015). Multi-Agent System Applications in Healthcare: Current Technology and Future Roadmap. Procedia Computer Science, 52, 252–261.
15. Wang, J., Cao, J., Wang, S., Yao, Z., & Li, W. (2022). IRDA: Incremental Reinforcement Learning for Dynamic Resource Allocation. IEEE Transactions on Big Data, 8(3), 770–783.
16. Balasubramanian, S., Fawad, A., Zahoor, M. S., &Muniandi, B. (2023). Efficient Workload Allocation and Scheduling Strategies for AI- Intensive Tasks in Cloud Infrastructures. ResearchGate.
17. Vengerov, D. (2007). A reinforcement learning approach to dynamic resource allocation. Engineering Applications of Artificial Intelligence, 20(3), 383–390.
18. S. Tripura and S. Lanka, "Architecture of Data Recognition System Using Machine Learning and TensorFlow," 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, 2024, pp. 1694-1699