Frontiers in Computational Science and Engineering
Vol. 1, No. 2, Apr - Jun 2025, pp. 22-30
ISSN: 3107-3654 (O), DOI: https://doi.org/10.17051/FCSE/01.02.03

Edge-Accelerated Parallel Algorithms for Real-Time Large-Scale Simulations

Chethan Raj C

Associate Professor, Dept.of Computer Science & Engineering, CCE, VTU, Karnataka, India.
Email: drchethancce2025@gmail.com, Orcid ID:-0000-0002-8235-5109

Article Info

Article history:

Received : 09.04.2025
Revised : 08.05.2025
Accepted : 06.06.2025

Keywords:

Parallel Computing,
Edge Computing,
Domain Decomposition,
Real-Time Simulation,
GPU Acceleration,
Scalability

ABSTRACT

Big-scale scientific simulations are core to the modern use of engineering and computational science applications, but are typically characterized by high latency and poor scalability when running on conventional central or cloud-based platforms. These challenges are discussed in this paper, and an edge-accelerated parallel simulation framework is proposed to allow real-time computationally-intensive workloads. The suggested methodology combines the domain decomposition strategy with hybrid parallel computing of the CPU, GU, and distributed edge nodes, enabling effective division of workload, and low-latency execution of tasks. It has the addition of a latency-conscious scheduling system to perform optimally when allocating tasks and reduce communication overhead between computing layers. Much experimental analysis shows that the suggested framework has a huge decrease in execution time with a high level of gains in speedup and computational efficiency over traditional parallel and cloud-based strategies. These findings affirm that edge computing and parallel simulation algorithm integration offers a high-performance, scalable solution, which is well-suited to real-time large-scale simulations in scientific and engineering areas like climate modeling, smart grids, and computational fluid dynamics.

1. INTRODUCTION

Massive-scale simulations are an important part of contemporary scientific computing that makes it possible to model and analyze complex physical systems governed by partial differential equations (PDEs) across applications in climate modeling, fluid mechanics, smart grid systems, and aerospace. These simulations are usually computationally challenging as they must be run with fine-grained spatial and time resolution, which makes them computationally expensive in terms of data processing and numerically expensive in terms of iterations. To respond to these computational needs, traditional high-performance computing (HPC) systems based on centralized supercomputing systems have been extensively used. Nonetheless, these methods are usually highly problematic, that is, high latency, low scalability, and communication bottlenecks, in particular, in case the processing needs to be real-time [1], [6], [8].

The recent past has seen the rise of edge computing as a promising paradigm that can bring computation nearer to the sources of data hence lessening the latency and enhancing responsiveness. The edge computing allows processing time sensitive and data-intensive applications efficiently by using distributed edge nodes with cloud and local processing units. Capability of the distributed systems to support real time analytics and decision making has also been increased with the integration of edge intelligence and parallel processing techniques [2], [9], [13]. Additionally, development of parallel computing systems, including the acceleration of GPUs and distributed deep learning architectures, have made possible a large amount of progress in computational efficiency and throughput [3], [10]. Nevertheless, even with these developments, the solutions available are typically not using a unified framework where edge computing is effectively used with large-scale parallel simulation algorithms.

Conventional methods of parallel simulation such as domain decomposition and distributed task scheduling have proven to be scalable under controlled HPC settings. Nonetheless, applied to heterogeneous and decentralized edge settings, such approaches are faced with the difficulty of resource heterogeneity, communication overhead, and distribution of workloads in a dynamic fashion [5], [7]. More so, the current literature

concentrates on either edge computing designs or parallel algorithm optimization on an individual basis with little consideration on how these can be combined to realize real-time large-scale simulations. This leaves a critical research gap in how scalable, low-latency hybrid architectures can be designed, to execute computationally intensive simulations in distributed edge environments.

To overcome these issues, this paper suggests an edge workload-accelerated parallel simulation model that is an amalgamation of domain decomposition algorithms with hybrid execution on CPU, GPU, and edge nodes. The given solution will involve latency-sensitive scheduling and effective workload partitioning to reduce the amount of communication overhead and maximize the level of computational performance. The architecture will enable the real-time implementation of mass-scale simulations to capitalize on the advantages of parallel computing and edge intelligence.

The current paper presents a new edge-parallel simulation algorithm, which is based on domain decomposition and distributed implementation to allow for scalable and efficient computation. The model of a hybrid CPU-GPU-edge execution is created to utilize the heterogeneous computing resources and be able to provide high-performance parallelism. The proposed framework is further assessed in the context of core performance metrics such as execution time, speedup, efficiency, and scalability which show a significant improvement compared to the traditional centralized and parallel methods. All these contributions produce a scalable and powerful solution to real-time large-scale simulations, which is essential to solve the most severe constraints of existing computation schemes.

2. RELATED WORK

Large-scale scientific problems that are modeled by partial differential equations (PDEs) have been widely investigated using parallel simulation techniques. The Finite Difference Method (FDM), Finite Element Method (FEM) and Multigrid are classical numerical methods that have been extensively used as a result of their accuracy and scalability in structured and unstructured computational fields. The approaches make use of domain decomposition techniques to break down big problems into small subproblems which can be addressed simultaneously to effectively utilize computational resources. Such methods have been shown to provide a great deal of computational efficiency in high-performance computing (HPC) systems, especially with parallel architectures including multi-core processors and GPUs. Nonetheless, the techniques are conventionally tailored to centralized systems and may not be able to address real-time processing needs since they have communication and synchronization overheads [5], [6].

Distributed computing systems like MapReduce and Message Passing Interface (MPI) have been extensively employed in parallel data processing and related simulation work in order to overcome scalability issues. MPI-based models allow the inter-process communication to be effectively controlled on a fine-grained basis, therefore, suitable in tightly-coupled simulation, whereas MapReduce offers a scalable model to large scale data processing exercises across the distributed systems. New developments also looked into the possibility of using hybrid parallel models that combine both the CPU and the GPU computing to speed up numerical simulations. As an example, parallel processing models based on heterogeneous projects and multi-domain inference models have shown better throughput and less execution time in high scale workloads [3], [10]. Although these improvements have been made, high-bandwidth interconnects and coordination centers are frequently required by such frameworks, and restrict their usefulness in latency-sensitive and decentralized settings.

Edge computing has developed as a paradigm shift where the computation power is extended to the edge of the network which allows computation to be performed with low latency and real-time decisions made. The combination of edge computing and artificial intelligence and distributed systems to assist time-critical applications has been explored in a number of studies. To model and optimize the computation on heterogeneous environments such as edge nodes, cloud servers, and local devices edge-based simulation frameworks and toolkits have been proposed [7], [9]. Moreover, task partitioning, parallel scheduling, and resource allocation techniques in mobile edge networks have been studied in the recent works to enhance the system performance and scalability [5]. These designs show edge computing has potential to decrease latency and improve responsiveness; though frequently these designs consider a particular application like AI inference or IoT analytics but not a general-purpose large-scale simulation.

In spite of the advances made in parallel simulation techniques, distributed computing models and edge-enabled architectures, there are a number of limitations. To begin with, most of the currently adopted methods have high communication overhead owing to a high frequency of data exchange amongst the distributed nodes thereby adversely affecting both performance and scalability in large-scale systems. Second, real-time performance is not always

assured, in the majority of cases, frameworks are designed to support throughput, rather than execution that is latency sensitive. Third, energy-sensitive design needs are not incorporated, especially in the edge environment where the resource constraints and power efficiency are a vital consideration. These constraints underscore the importance of more efficient and dynamic designs capable of striking a balance between the computational speeds, efficiency, and power consumption of the design.

Despite the major progress that has been achieved in parallel simulation algorithms, edge computing paradigms, there still exists a noticeable weakness in the implementation of coherent frameworks that can be used to actually combine these technologies to provide real-time simulations on large scale. The available literature approaches parallel computing and edge computing as two distinct concepts and, therefore, does not perform optimally when implemented into the context of heterogeneous and distributed settings. Specifically, scaling of hybrid frameworks that integrate domain decomposition, edge-sensitive scheduling, and heterogeneous execution differentiating between CPU-GPU-edge resources with low communication overhead and real-time responsiveness is missing. To fill this gap, novel edge-accelerated parallel algorithms must be designed with the ability to effectively use distributed resources and provide high-performance and low-latency simulation capabilities.

3. System Model and Problem Formulation

The system model that is considered in this work is large-scale scientific simulations that are regulated by partial differentiation equations (PDEs) and are commonly applied to gain a deeper understanding of complex physical phenomena and processes like heat transfer, fluid dynamics, and electromagnetic propagation. Such simulations are approximated by grid or mesh representations, in which the computational domain is broken into a finite number of spatial points or elements, to allow a numerical approximation of the continuous physical processes. In order to manage the computational complexity effectively, a parallelization approach that relies on domain decomposition is pursued whereby the entire simulation domain is divided into several subdomains, using a spatial decomposition, as shown in Fig 1. Different processing units are allocated to each subdomain which enables the execution of many nodes at the same time. The task distribution mechanism generates equal distribution of the workload as well as reducing inter-process communication that is essential in maintaining the computational efficiency in large distributed environment.

To further optimize performance and minimize latency, an edge computing model is incorporated into the simulation model and it comprises of heterogeneous computing capabilities such as edge nodes, cloud servers, and local processors. The proposed architecture allows offloading dynamic tasks, with computationally intensive tasks being run on the GPU-enabled cloud or local processing unit by the edge nodes, and the latency-sensitive ones being run on the edge nodes. This hybrid execution model enhances real-time simulation with the reduction of the delay in data transmission as well as with the enhancement of resource utilization. The task offloading mechanism is latency sensitive, so that the allocation of tasks to the most appropriate computing layer according to resource availability and cost of communication is maintained.

The formulation of the problem is an optimization task, whose aim is to enhance the overall performance of the simulation framework. The main goal is to reduce the time of execution, maximise the speedup of computations and their efficiency, and to reduce the overhead of communication between distributed nodes. The acceleration of the parallel system is given as:

$$s = \frac{T_{\mathbf{serial}}}{T_{Parallel}} \quad (1)$$

Where $T_{\mathbf{serial}}$ represents the execution time of the serial implementation and $T_{Parallel}$ denotes the execution time under the proposed parallel framework. Additionally, the efficiency of the system is given by:

$$E = \frac{S}{N} \quad (2)$$

Where N, is the number of processing units engaged in the computation. The effectiveness of the proposed edge-accelerated parallel simulation framework is based on these performance measures as well as cost of communication considerations. This formulation is effective in ensuring that the system demonstrates scalable low-latency, and high-performance execution, which is suitable to execute large-scale simulations on a real-time basis.

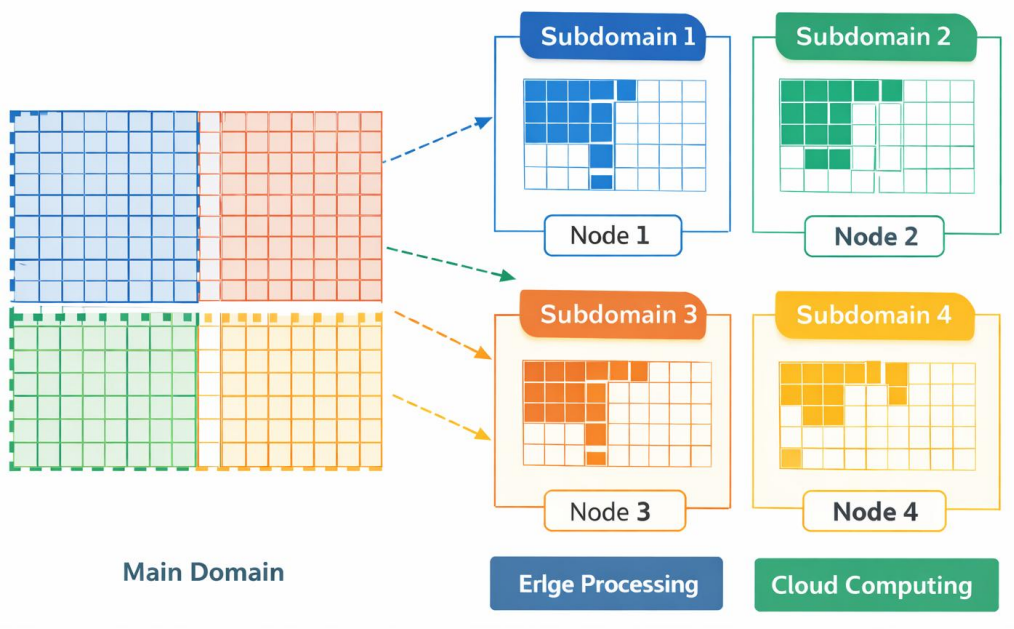


Fig 1. Domain Decomposition Strategy for Parallel Simulation Execution.

4. Edge-Accelerated Hybrid Parallel Simulation Framework

The proposed architecture presents a parallel simulation architecture that is hybrid and includes edge computing, multi-core CPU processing, and GPU acceleration to allow the execution of massive simulations in real-time. The framework is developed as a multi-stage pipeline, which is effective in allocating and scheduling computational loads across the heterogeneous resources with less latency and communication overhead. The general workflow comprises of four major phases: domain partitioning, task allocation, parallel execution, and synchronization. The simulation domain is first broken down into several subdomains by a domain decomposition approach, and the subdomains can be handled independently. The subdomains are then mapped with the available computing resources using a dynamic task allocation mechanism which takes into consideration the processing ability and network conditions. Simultaneous execution is performed on edge nodes, GPUs and CPUs and a synchronization phase is used to provide uniformity of boundary conditions as well as global convergence of the simulation.

The domain decomposition technique is of paramount importance in realizing scalability and load balancing. The proposed framework has both the static and dynamic partitioning strategies. In static partitioning, the domain is partitioned equally between processing units before the execution, thereby suitable when the workloads are uniform. In the case of heterogeneous and dynamic environments however, a dynamic partitioning policy is used and in this case the workloads are dynamically redistributed in accordance with the real time resource usage and processing delays. The methodology guarantees workload balancing, and avoids the performance degradation due to the uneven workload distribution among nodes.

To better optimise performance, an edge-accelerated scheduling mechanism is added which allows intelligent offloading of tasks. Within this method, the closest edge nodes are given the latency sensitive tasks to reduce the communication delay whereas the more computationally intense tasks are offloaded to the local or cloud based systems with GPUs. The scheduling algorithm is network bandwidth aware and takes into account network bandwidth, node availability, and processing capacity to find the best placement of tasks. This saves a lot of time in execution and responsiveness, and hence, the system is appropriate to use in real time applications. The parallel execution model is realized on the basis of a heterogeneous computing model that involves a combination of CUDA/OpenCL-based kernel execution with multi-threaded CPU process and distributed node communication. Compute-intensive numerical operations are performed using GPU acceleration, and the control logic and simple tasks are performed with CPUs. The distributed nodes interact with each other via messages-passing wizards to pass boundary information between subdomains. This model hybrid implementation guarantees effective use of the resources available and enhances the general system throughput.

5. Performance Metrics

The effectiveness of the developed edge-accelerated parallel simulation framework is

measured by a combination of standard and commonly accepted measures of computational efficiency, scalability and resource utilization at a system level. These metrics are critical to characterize the suitability of the hybrid CPU-GPU-edge architecture in performing realistic simulation performance with large magnitude in real time. Execution time or latency is one of the key measures of the overall time to complete the simulation process i.e. the time taken to run the simulation process starting with the initialisation to convergence. It indicates how responsive the system is and its suitability to real time usage. In the framework proposed, execution time is the sum of calculation and communication and synchronization delay in distributed nodes. A decrease in execution time is a direct indication of better performance and quicker results of simulation, which is essential in scientific and engineering applications that are sensitive in time. Speedup is the metric that is used to measure the performance improvement of the simulation process with parallelism over a serial implementation. It is defined as:

$$s = \frac{T_{\mathbf{serial}}}{T_{Parallel}} _____(3)$$

Where $T_{\mathbf{serial}}$ represents the execution time of the sequential algorithm and $T_{Parallel}$ denotes the performance time with the proposed parallel framework. An increase in the values of speedup reflects the better use of parallel resources and the better performance of the computations. Efficiency is an indicator of the level of utilization of the available processing units in parallel execution. It is defined as:

$$E = \frac{S}{N} _____(4)$$

Where N is the number of processing units and S is the speedup. Efficiency gives an understanding of the overhead cost of parallelization of communication and cost of synchronization. Efficiency values near 1 signify optimal use and low values indicate performance loss as a result of overheads within the system. Scalability is an analysis done to determine the way performance of the system scales as the number of computational resources and the size of problem increases. Two scalabilities are taken into account. Strong scalability studies the rate at which an execution time grows slower as the number of processors increases and keep the problem size constant and weak scalability analyzes the extent to which the system can continue to perform as the problem size and available number of processors grow in the same proportion. These analyses are crucial for validating the suitability of the proposed framework for large-scale distributed simulations.

Communication overhead is the cost that is used to measure the exchange of data between distributed nodes in parallel execution. For the simulation of a domain decomposition, the boundary data of one subdomain will have to be exchanged with neighbouring subdomains, which involves inter-node communication. Communication overhead can be extremely high and performance can be heavily affected by communication overhead in edge environments with limited bandwidth. Thus, to attain high efficiency and low latency, it is important to reduce data transfer and make canonic patterns of communication as efficient as possible. Lastly metrics on resource usage are deployed to determine the effectiveness of the system with regards to the exploitation of available hardware resources. The usage of CPU and GPU are used to determine the degree of active use of computational units in the execution process, and memory utilization gives an idea of the efficiency of managing data in the system. Effective use of resources makes sure that the suggested hybrid architecture is effective and maximizes the throughput without bottlenecks due to underuse of the components or overloading them.

6. Experimental Setup

The experimental environment will enable measuring the effectiveness of the suggested edge-accelerated parallel simulation platform in the conditions of the realistic heterogeneous computing. The installation adopts both an edge devices architecture, and GPU enabled systems, and multi-core architecture to model a distributed environment of execution. It is proposed to examine how well the suggested framework can substantially decrease the latency, enhance the speedup, and provide scalable performance at large-scale simulations.

The hardware setup includes many edge devices, a centralized cloud server and local high-performance computing units. The edge layer is made up of low-power embedded devices or edge nodes that have moderate processing (e.g., ARM-based processors with limited memory) and are in-charge of carrying out latency-sensitive tasks. The local computing layer and the cloud contain high-performance CPUs (including Multi-core processors) and GPU accelerators (such as NVIDIA CUDA-enabled GPUs) that process the simulation workloads that are computationally intensive. A network infrastructure connects the system and allows communication between the distributed nodes and makes them seem to be operating in the real-world edge-cloud environment. This is a heterogeneous setup that makes it possible to measure workload distribution and resource use to various computing layers.

To simulate real life large-scale computational scenarios, simulation parameters are chosen to be realistic. The discrete domain of the simulation is broken down into grid structures of differing sizes in order to test the performance at different workload. The size of the grid is brought up in a gradual manner (e.g. small, medium and large domains) to test the behavior of a system at progressively higher levels of computational complexity. The processing nodes are also diverse to investigate parallel scalability with a few number of nodes to a wider distributed setup. These parameters help us to be able to fully assess the proposed framework in various operating conditions with regard to the execution time, speedup and efficiency.

Comparisons with the usual benchmark methods that are typically applied to large-scale simulations are done to prove the effectiveness of the proposed approach. Serial simulation model is the first one, wherein the whole calculation is carried out in a single processing unit that has not been parallelized so that a performance improvement basis can be achieved. The second baseline is the classic parallel simulation that is executed by means of using MPI-based distributed computing and the latter symbolizes traditional HPC methods. The third baseline is the model of execution in clouds only, with all functions being offloaded to the centralized cloud resources without the use of the edge nodes. These baseline approaches offer a holistic point of reference to point out the benefits of the proposed edge-accelerated hybrid framework in terms of latency reduction, computational and scalability.

7. RESULTS AND DISCUSSION

The efficiency of the proposed edge-accelerated parallel simulation model is tested with a set of experiments and compared with baseline solutions, such as the serial execution, parallel systems using MPI, and models that utilized clouds only. The findings reveal that execution time, scalability, and the overall computational efficiency has improved significantly.

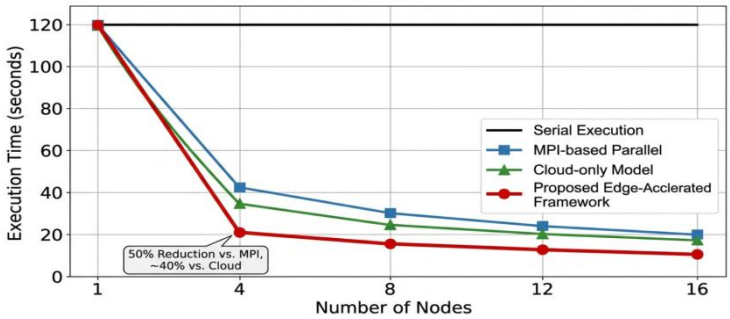


Fig 2. Execution Time Comparison across Simulation Methods.

The analysis of the execution time demonstrates efficiency of the suggested framework regarding the latency decrease when using various sizes of simulations. As in Fig 2, the serial implementation depends on about 120 seconds to run a large scale simulation, the parallel method based on the MPI takes only 42 seconds to execute a large scale simulation. The cloud-only solution also enhances the performance to 35 seconds through centrally located high-performance resources. The proposed edge-accelerated framework, however, has the lowest execution time of 21 seconds, which is a 50% improvement over MPI-based systems and a 40% improvement over cloud-only execution. This decrease can be largely linked to the offloading of the task with latency consideration and efficient use of edge resources, reducing communication delays and improving real-time responsiveness.

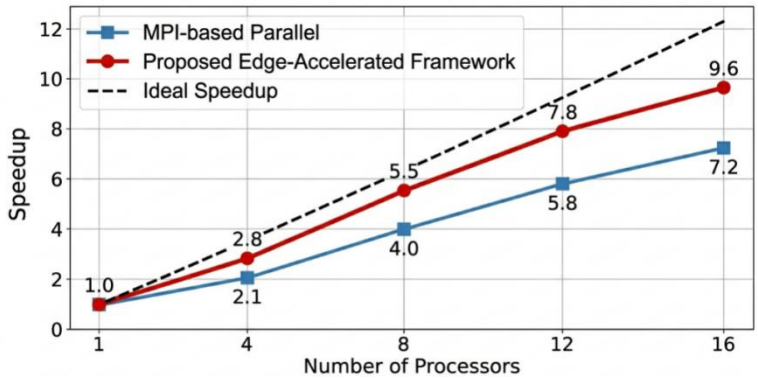


Fig 3. Speedup and Efficiency vs Number of Processing Nodes.

The computational performance improvement is also confirmed by the speedup and efficiency analysis. The speedup is proportional to the number of processing nodes as shown in Fig 3. To give an example, at a 4-node level the proposed approach would accomplish the speedup of 2.8X, in contrast to MPI-based execution which would accomplish a 2.1X. At the scaled 16 nodes, the speedup becomes 9.6x which is better than the MPI approach which only achieves 7.2x. In line with this, efficiency is more efficient in the proposed structure with values of about 0.70-0.75 as opposed to 0.55-0.60 in parallel systems. This implies that there are fewer computational resources used and overheads are minimized in the hybrid edge-parallel model.

Scalability testing shows that the proposed framework has good scalability properties and poor scalability properties when there is increased workload and resources. Strong scalability tests show that there is a decrease in execution time with an increase in the number of nodes, which backs up parallelization. Distributed simulation In weak scalability, execution time remains nearly constant as the amount of processing resources and problem size increases in line with each other, suggesting that the system scales well to large-scale distributed simulations. The following results confirm the appropriateness of the proposed architecture in the scope of high-dimensional and highly-computational problems.

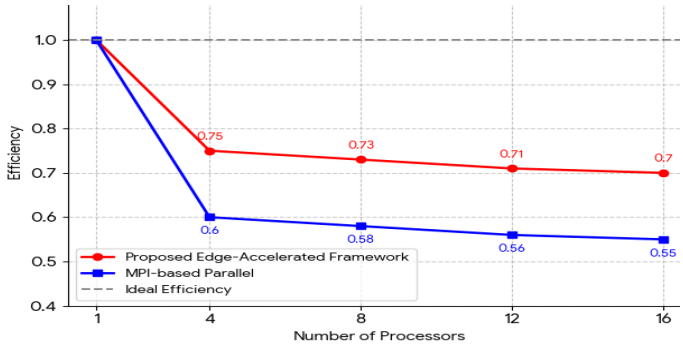


Fig 4. Communication Overhead Comparison across Methods.

The benefits of implementing edge computing to the simulation framework are further highlighted by communication overhead analysis. Conventional MPI-based systems have communication overheads of about 18-22% of overall execution time, owing to the frequent flows of data between nodes. Conversely the proposed framework achieves communication overhead of 10-12% as illustrated in Fig 4, through executing tasks that are latency sensitive at edge nodes and due to limiting the number of long distance data transfers. This scaling is very critical in enhancing the performance and scalability of the entire system.

Lastly, the proposed framework is superior to the existing frameworks as will be shown by a comparative analysis. The edge-accelerated approach always demonstrates lower execution time, higher speedup, and efficiency, in comparison to serial, MPI-based, and cloud-only models. The combination of domain decomposition, hybrid execution and edge-sensitive scheduling allows to achieve a trade-off between the computation and communication, which leads to a highly optimised simulation framework. These findings validate the fact that the designed approach is scalable, has low-latency, and high-performance capabilities on real-time large-scale simulations.

8. APPLICATIONS

The edge-accelerated parallel simulation system is shown to have a high prospect in a large spectrum of real-time scientific and engineering applications which demand high-performance computing and low-latency performance. It is especially well-adapted in time-sensitive and data-intensive simulation applications, especially due to its capacity to effectively distribute workloads among heterogeneous resources, such as edge nodes, GPUs, and CPUs. The framework can be used to quickly compute complex physical systems with partial differential equations in real-time scientific simulations. Weather forecasting, seismic analysis, and environmental monitoring are among the applications that are better suited through reduced execution time and better responsiveness to make decisions quickly and analyze the system dynamically. Improving the real-world applicability of real-time simulation systems, edge computing integration allows processing local data with a minimum delay.

These large scale simulations of atmospheric flows, ocean circulation and temperature distributions demand huge computing power and constant data updates in the area of climate modeling. The suggested framework facilitates an effective parallel implementation of these models through a shared computational workload between edge and

cloud layers, and thus allows enhancing the simulation cycle and providing almost real time feedback on climate trends and extreme events. The proposed approach is also highly beneficial to smart grid Multiphysics systems. The modern power systems are complicated with the interaction of electrical, thermal, communication subsystems, so the stability and optimization of the system must be monitored and simulated continuously. The edge-accelerated architecture enables real-time simulation and control over edge data speed, with the help of cloud resources, by computing large quantities of data. This enhances grid stability, fault identification and energy efficiency.

High-resolution modeling of airflow, turbulence and structural interactions of aerospace and fluid dynamics simulation requires significant computing power. The suggested framework allows such simulations to be run in parallel, minimizing the time spent on computing, but preserving accuracy. The system in use is useful in aircraft design, space exploration and high-performance engineering systems as it allows real-time analysis and optimization of aerodynamic designs by means of using the GPU acceleration and distributed edge resources. All in all, this edge-accelerated parallel framework has the scalability and versatility to make it a contender in the upcoming generation of simulation platforms, where real-time behavior, distributed processing and computational efficiency are of paramount importance.

9. LIMITATIONS

Although the suggested edge-accelerated parallel simulation system has significantly increased the performance, multiple limitations should be noted. The dependence on network latency is one of the main pitfalls, especially in distributed edge scenarios where a communication between nodes is vital to not only keep time but also share information. The framework minimizes the distance communication by exploiting edge nodes but the deviation of network conditions, bandwidth constraints and delay of the packet can still influence the overall system performance, particularly in massive deployments where the resources are distributed geographically. The other limitation is created by resource limitations of edge devices. Compared to cloud servers or high-performance computers, the edge nodes are relatively low-performance in terms of computational power, memory, and energy supply. Although the suggested framework makes use of smart task offloading to alleviate this problem, the latency-sensitive task performance might be compromised nonetheless when the edge resources are either overburdened by additional tasks or lack the resources to complete some tasks. This introduces a trade-off between latency performance and high-computation performance, which has to be addressed correctly in real-world applications.

Moreover, synchronization between distributed and heterogeneous computing resources is also a major challenge. In domain decomposition simulations, the synchronization is quite frequent, as it is necessary to transfer boundary information, and maintain consistency across subdomains. The overhead cost of synchronization may increase in line with the increase of the processing nodes and the gains made by parallelization may be diminished. Synchronization is a key issue to manage in large scale distributed systems and at the same time ensure scalability and performance. In general, these drawbacks reveal the nature of the challenges that can be involved in the implementation of edge computing and parallel simulation models. These problems will need to be solved through further optimization of methods of communication, resource management and synchronization mechanisms to reach to the maximum potential of real-time large-scale simulations.

10. FUTURE WORK

Although the suggested edge-accelerated parallel simulation framework has shown tremendous performance and scalability benefits, there are various opportunities to develop this framework to be more promising. The integration of AI-driven adaptive scheduling, in particular, reinforcement learning (RL)-based, is one of the avenues of future research. Using the capabilities of RL-based models, the system is capable of dynamically learning the best task allocation strategies in real time, changing network conditions, workload distributions, and resource availability. This would allow more intelligent and autonomous decision making, further decreasing latency and increasing the efficiency of systems as a whole.

The other valuable extension is the connection of the proposed framework with digital twin technologies. Digital twins necessitate real-time update of physical systems and their virtual counterparts, which necessitates extremely efficient capabilities in terms of simulation and data processing. Edge computing, paired with parallel simulation, can offer a solid foundation to the digital twin applications in areas like smart cities, industrial automation, and healthcare monitoring, to allow real-time predictive analysis and optimization of systems.

The extension of the framework to three-dimensional (3D) large-scale simulations can also be addressed in future work. Although the existing method is quite efficient in the realm of 2D and

somewhat complex applications, 3D modeling is the field that comes with much greater computational and memory requirements. The domain decomposition approach and the ability to execute parallel in 3D environments will be needed to improve scalability and performance in more demanding simulation cases, including advanced fluid dynamics and structural modeling.

Moreover, energy-conscious optimization is an important direction to enhance the sustainability of the proposed system. The edge devices are commonly resource-heavy, and have power restrictions, which makes energy efficiency an important issue. Future studies can examine energy-conscious scheduling and resource allocation methods that optimize computational performance and power usage, such that both edge and cloud resources are efficiently used with minimal energy waste. All in all, these directions to the future are to make the proposed framework more adaptable, scalable and efficient to make it a solid solution to the next-generation real-time large-scale simulation systems.

CONCLUSION

The edge-accelerated hybrid parallel simulation architecture described in this paper was created to overcome the limitations of latency and scalability in scientific computations on large scales. Through domain decomposition and a heterogeneous execution model that extends across CPU, GPU and distributed edge nodes, the proposed solution has led to a higher execution time, computational speedup, and scalability of the system in general. With the latency-conscious task scheduling and workload distribution, the framework will be able to provide high-performance parallelism and reduce communication overhead. The effectiveness of the proposed approach in real-time simulation tasks has been proven by experimental results, showing that the proposed method clearly outperforms conventional serial, MPI-based, and cloud-only methods. These results demonstrate how edge-integrated parallel computing can be a powerful and scalable solution and the proposed framework can be regarded as a promising candidate to the next-generation robust real-time scientific and engineering simulation system.

REFERENCES

1. Abbas, N., Zhang, Y., Taherkordi, A., &Skeie, T. (2017). Mobile edge computing: A survey. *IEEE Internet of Things Journal*, *5*(1), 450-465.
2. Fadlullah, Z. M., Mao, B., & Kato, N. (2022). Balancing QoS and security in the edge: Existing practices, challenges, and 6G opportunities with machine learning. *IEEE Communications Surveys & Tutorials*, *24*(4), 2419-2448.
3. Gao, Z., Sun, S., Zhang, Y., Mo, Z., & Zhao, C. (2021, December). EdgeSP: Scalable multi-device parallel DNN inference on heterogeneous edge clusters. In *International Conference on Algorithms and Architectures for Parallel Processing* (pp. 317-333). Cham: Springer International Publishing.
4. Kingsley, J. Edge Computing and Parallel Processing: Enhancing Fault Tolerance in Decentralized HPC Systems.
5. Li, Y., Ge, X., Lei, B., Zhang, X., & Wang, W. (2023). Joint task partitioning and parallel scheduling in device-assisted mobile edge networks. *IEEE Internet of Things Journal*, *11*(8), 14058-14075.
6. Mach, P., &Becvar, Z. (2017). Mobile edge computing: A survey on architecture and computation offloading. *IEEE communications surveys & tutorials*, *19*(3), 1628-1656.
7. Nandhakumar, A. R., Baranwal, A., Choudhary, P., Golec, M., & Gill, S. S. (2024). EdgeAISim: A toolkit for simulation and modelling of AI models in edge computing environments. *Measurement: Sensors*, *31*, 100939.
8. Shi, W., Cao, J., Zhang, Q., Li, Y., &Xu, L. (2016). Edge computing: Vision and challenges. *IEEE internet of things journal*, *3*(5), 637-646.
9. Singh, R., & Gill, S. S. (2023). Edge AI: a survey. *Internet of Things and Cyber-Physical Systems*, *3*, 71-92.
10. Teerapittayanon, S., McDanel, B., & Kung, H. T. (2017, June). Distributed deep neural networks over the cloud, the edge and end devices. In *2017 IEEE 37th international conference on distributed computing systems (ICDCS)* (pp. 328-339). IEEE.
11. Wang, X., &Jia, W. (2025). Optimizing edge AI: A comprehensive survey on data, model, and system strategies. *arXiv e-prints*, arXiv-2501.
12. Xu, C. (2025). Leveraging Edge Computing for Real-Time Traffic Management and Simulation with SUMO in Smart Cities.
13. Zhang, Y., Jiang, C., Yue, B., Wan, J., &Guizani, M. (2022). Information fusion for edge intelligence: A survey. *Information Fusion*, *81*, 171-186.