Scholar Publishing Group
International Journal of Neural Network
https://doi.org/10.38007/NN.2025.040106
ISSN 2790-2692 Vol. 4, Issue 1: 48-56

Discussion on Low-Latency Computing Strategies in Real-Time Hardware Generation

Huijie Pan

Identity Department, PayPal Inc., San Jose, California, 95131, United States

Keywords: Low latency computing; Real-time hardware generation; Hardware resource scheduling

Abstract: The rapid development of real-time hardware generation has led to low latency computing becoming one of the factors to improve system performance. This paper discusses the delay challenges in real-time hardware generation, including low efficiency of hardware resource scheduling, bandwidth limitation of data transmission, memory access bottleneck and insufficient collaborative optimization of hardware and software, and proposes to use dynamic scheduling to improve the efficiency of hardware resource use, upgrade the data transmission interface to improve bandwidth, and realize intelligent cache management to improve memory access speed. And deepen the hardware and software co-design to improve the running degree, effectively reduce the delay time, improve the comprehensive performance of real-time hardware generation.

1. Introduction

In the context of the rapid development of information technology, the development of real-time hardware generation systems has brought an important impact on the development of various industries, including autonomous vehicles, financial exchange and industrial control. Low latency computing is a key factor to ensure system performance. However, due to the limited hardware resources, the bandwidth limit of data transmission and the bottleneck of memory access, low latency computing in real-time hardware generation faces many challenges. This paper will deeply discuss the current status of low latency computing technology, analyze the main problems and optimization paths, in order to provide reference for the innovation and development of related technologies.

2. Key technologies of low latency computing

The core goal of low latency computing is to achieve real-time task completion by reducing system reaction time. This involves many technical levels, including hardware architecture optimization, data transmission acceleration, memory and cache management, parallel computing and task scheduling. Hardware acceleration plays a decisive role, such as GPU, FPGA, ASIC and other hardware accelerators have greatly improved the performance of computing. GPU can process a large amount of information at the same time, which is suitable for large traffic data processing.

Fpgas can greatly increase the computation time through the execution of custom accelerators on the hardware. Asics provide specialized computing power, especially when small latency is required. Other improvements in memory and data exchange can also help reduce latency. By improving the hierarchical structure of memory, intelligent buffer management and memory prefetch strategy, the access speed of memory can be improved effectively, and the access delay of memory can be reduced. For data transmission, high-speed interconnects (such as NVLink) and efficient communication methods can speed up data exchange, thereby further reducing latency. Parallel computing and task scheduling use various types of processors and distributed computing platforms to make full use of computer resources, thereby improving work efficiency. In addition, there is a real-time operating system (rtos) and advanced scheduling algorithm to avoid scheduling delays. Figure 1 below summarizes the multi-layered technical support for low latency computing:

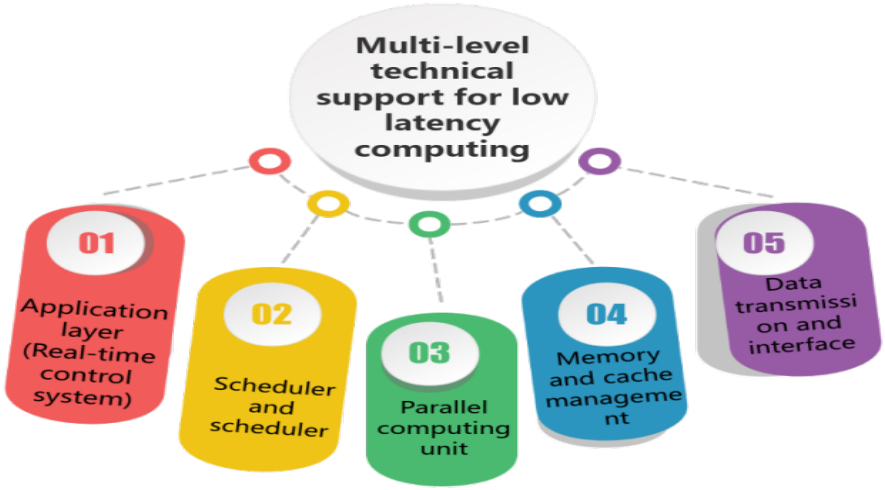

Figure 1.Multi-level technical support for low-latency computing

3. Status of low latency computing technology in real-time hardware generation

3.1 Low hardware resource scheduling efficiency

The low efficiency of hardware resource scheduling in real-time hardware generation is mainly reflected in the aspects of resource contention, load imbalance, scheduling delay and resource waste. Due to the extensive use of GPU, FPGA and other hardware accelerators, multiple computing tasks need to be scheduled to different hardware components. Under the above conditions, traditional manual scheduling methods cannot adapt to task changes, resulting in the system cannot effectively schedule limited hardware devices. Resource contention is an important cause of inefficient hardware resource scheduling. Multiple tasks competing for the same type of hardware resources will reduce computing performance and delay system reaction time. The performance of load imbalance is significant. Some hardware components delay the execution of tasks due to overload, while some hardware components are idle due to improper scheduling, resulting in a waste of computing resources. Scheduling delay hinders the fast response to high-priority tasks and affects the real-time performance of the system. Hardware resource waste exists in manual scheduling. The static allocation of some hardware resources does not follow the requirements of tasks and remains idle for a long time. Table 1 below shows the possible effects of inefficient hardware resource scheduling:

Table 1.Impacts of low hardware resource scheduling efficiency

influence	expression	reason
Resource contention	The parallelism of computing tasks decreases and the overall latency increases	Static scheduling cannot handle the resource requirements of multiple tasks
Load unbalance	Some hardware units are overloaded and the task execution efficiency is low	Scheduling algorithm can not allocate computing tasks reasonably
Scheduling delay	The response of high-priority tasks is delayed, affecting system performance	Lack of flexible priority scheduling mechanism
Hardware resource waste	Some hardware units are idle and cannot fully utilize computing resources	Resource scheduling cannot be dynamically adjusted based on task requirements

Table 1 shows that the inefficiency of hardware resource scheduling is mainly reflected in resource contention, load imbalance, scheduling delay, and resource waste. These problems seriously affect system performance.

3.2 Limitation of data transmission bandwidth

The limitation of data transmission bandwidth is an important factor affecting system performance and latency. With the increasing complexity of computing tasks, the amount of data in computing tasks increases gradually, which leads to the bottleneck of data transmission and increases the system delay time. The main reason is the bandwidth limit of the hardware interface itself. For example, traditional bus and network interfaces (such as PCIe and Ethereal) cannot meet the high-speed communication of large amounts of data, especially in multi-core cpus and multi-task parallel computing. At the same time, the design of the transmission protocol does not effectively improve the data transmission path, resulting in a lot of unnecessary delays and a lot of ineffective bandwidth consumption in the process of data transmission, and then affect the overall performance of the system. In addition, the limitation of data transmission bandwidth will directly affect the computing efficiency. In real-time tasks that require a large amount of data to interact frequently, bandwidth bottlenecks may also cause data transmission delays, resulting in slower computing processes. Bandwidth limitations also cause low utilization of hardware devices, causing some hardware modules to wait for data without reason, and thus cannot operate efficiently. Table 2 below summarizes the possible effects of data transmission bandwidth limitations:

Table 2. Impact of data transmission bandwidth limitation

influence	expression	reason
Data transmission delay	The system response time is extended and the real-time performance is reduced	The bandwidth of the hardware interface is insufficient to meet the transmission requirements of large amounts of data
Computational efficiency decline	The data processing speed is reduced and the overall task execution time is increased	The transmission protocol is not optimized, and the bandwidth resources cannot be used efficiently
Uneven utilization of hardware resources	Some hardware units are idle and do not fully participate in computing tasks	Data transmission bottlenecks prevent the hardware from working efficiently
Uneven utilization of hardware resources	Efficient parallel task processing is limited by bandwidth	Due to insufficient bandwidth, data transmission cannot be completed in time, affecting the parallelism

As can be seen from Table 2, the limitation of data transmission bandwidth directly affects the computing efficiency and real-time performance of the system, resulting in transmission delay, hardware resource waste and limited parallel computing capability.

3.3 Memory Access Bottlenecks

Memory access bottleneck is one of the key factors affecting low latency computing performance. With the complexity of computing workload, the continuous data interaction between CPU and memory makes the memory access rate a bottleneck that affects the response time and execution efficiency of the system. The deepest root of the memory access bottleneck lies in the storage access hierarchy and its memory access methods. In general, memory is divided into different layers (such as L1, L2 cache, and main memory, etc.), and different layers have different access rates, and if the CPU needs to obtain data from a lower memory layer, the data transmission latency will increase significantly, thereby reducing the performance of the entire system. Especially for parallel computing environment, when multi-core or compute nodes request memory data at the same time, memory access conflict and memory access bandwidth competition further aggravate the bottleneck. Memory bandwidth bottlenecks and memory access delays are the main factors causing bottlenecks. Although the memory bandwidth is improved, the memory read and write rate cannot meet the requirements of real-time computing in the case of high concurrency and big data processing, and the unreasonable memory access mode and memory layout will increase the latency, especially in the case of frequent access to a large amount of data, memory access bottlenecks are more prominent. Table 3 below analyzes the impact of memory access bottlenecks on system performance:

Table 3. Analysis of the impact of memory access bottleneck on system performance

influence	expression	reason
Data processing delay	System response time increases, affecting real-time performance	Memory access is slow and the processor is idle while waiting for data
Insufficient utilization of computing resources	The computing unit is idle and hardware resources cannot be used efficiently	The memory bandwidth is insufficient, and the data transmission speed limits the usage of computing resources
Parallel computing performance deteriorates	Multi-core or multi-processing unit collaboration is inefficient and task latency increases	Multiple threads or cells compete for memory bandwidth, causing delays
System performance degradation	Overall system efficiency decreases and task processing time increases	The memory level access is improper, and the memory level that is slow is frequently read

As can be seen from Table 3, memory bandwidth limitation and access delay are key factors of the bottleneck, which leads to data processing delay, insufficient utilization of computing resources, and degraded performance of parallel computing.

3.4 Insufficient collaborative optimization of hardware and software

The lack of hardware and software co-optimization is the key factor affecting the performance of low latency computing. In the process of hardware optimization and software optimization, the lack of close connection between hardware and software will cause the function of hardware not to be fully utilized, resulting in the decline of the ability of the whole system. It is mainly caused by the division of hardware structure and software design, and does not take into account the

characteristics of hardware in the level of software implementation, resulting in software can not give full play to the hardware equipment, resulting in the delay of the system. At the same time, the software algorithm design often does not consider the hardware platform and is specially designed, resulting in the parallel computing function of the hardware can not be played, and the performance of a lot of parallel work is more prominent. The lack of coordination between hardware resource management and task scheduling also exacerbates this bottleneck. Lack of effective collaboration between hardware and software, resulting in hardware devices are not well used, there are some hardware devices waiting for data, no action, or tasks can not be evenly distributed between the calculator, will lead to increased latency of the system. Table 4 below summarizes the impact of insufficient hardware and software collaborative optimization on system performance:

Table 4.Effects of insufficient collaborative optimization of hardware and software on system performance

influence	expression	reason
Data transmission delay	The computing power of the hardware accelerator is not fully utilized	Software is not optimized and resources are not scheduled according to hardware characteristics
Computational efficiency decline	The task execution time is long, and the overall performance deteriorates	Software algorithms are not optimized for hardware platforms
Uneven utilization of hardware resources	The real-time task response time is prolonged, which affects the real-time performance of the system	Improper scheduling and coordination between hardware and software and uneven resource allocation
Parallel computing capability is limited	The processing efficiency of multi-core or parallel computing tasks is low	The hardware and software do not cooperate efficiently, and the task scheduling is unreasonable

As can be seen from Table 4, insufficient collaborative optimization of hardware and software leads to waste of hardware resources, decrease of computing efficiency and increase of system delay. The main reason lies in the lack of effective coordination between hardware architecture and software design and improper resource scheduling.

4. Low latency calculation optimization strategy in real-time hardware generation

4.1 Dynamic scheduling improves hardware resource utilization

Dynamic scheduling is a relatively scientific and effective optimization strategy, that is, to maximize the use of hardware equipment by properly adjusting the configuration of tasks and hardware equipment. Dynamic scheduling allocates computing resources based on the task importance level, current hardware device status, and task requirement ratio. Computing resources are allocated using scheduling technologies, such as load balancing and priority scheduling, to prevent too many hardware devices from being idle or underutilized and enhance the practical application benefits of hardware devices. Suppose the system has N tasks and M hardware resources, where the computing requirements of each task and the processing power of each hardware resource can be represented as a two-dimensional matrix.Quest i The computing requirements are C_i ,Hardware resource j The processing capacity is P_j ,The goal of dynamic scheduling is to maximize hardware resource utilization while ensuring that each task is completed within a specified time. The formula for hardware resource utilization can be expressed as:

$$U = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} x_{ij} C_i}{\sum_{j=1}^{M} P_j} \tag{1}$$

Among them, x_{ij} Presentation task i Whether to allocate hardware resources j , If task i Allocate resources j , the $x_{ij} = 1$ Or else $x_{ij} = 0$ 。Through the dynamic scheduling algorithm, the x_{ij} value can be adjusted according to the real-time load changes, so that the hardware resources can be optimally allocated, and the computing efficiency of the system can be improved and the delay can be reduced.

4.2 Upgrading a Data Transfer Interface to increase bandwidth

The bandwidth of the data transmission interface is one of the key factors that determine the performance of the system. Data transmission bandwidth can be optimized by using high-speed interconnection technologies, such as Peripheral Component Interconnect Express (PCIe) and NVLink. This kind of high-speed transmission data interface can significantly increase the rate of data transmission and reduce the delay of data transmission. Data compression and Efficient Data Transfer Protocol (RDMA) can also further improve bandwidth utilization to reduce network load. Assume that there are N data blocks in the system that need to be transferred between different hardware resources, and the size of each data block is D_i , the transmission speed is V_j ,Then the total data transmission bandwidth B_{total} It can be expressed as:

$$Btotal = \sum_{i=1}^{N} \frac{Di}{Ti} \tag{2}$$

Among them, Ti Is the transfer time of data block i ,The calculation formula is $T_i = \frac{D_i}{V_j}$, That is, the size of the data block divided by the transfer speed. By adopting higher speed interface technologies such asPCIe 4.0Or later versions, or the introduction of dedicated data transfer channels (such as using NVLink), can be significantly improved V_j ,This reduces the transfer time of each data block T_i , Thereby increasing the total transmission bandwidth B_{total}

3.3 Implementing Intelligent Cache Management to improve memory access efficiency

In low latency computing, it is very important to improve the efficiency of memory access. Enhancing computing power by optimizing memory access performance has become one of the key measures in low-latency computing. The core of it is to predict the required data size by observing and understanding the use of memory, and put frequently used information into the cache in advance, thereby reducing the delay caused by repeated memory reads and writes. The ways to improve cache include the selection of cache replacement, the prefetch technology and the management of cache consistency. Reasonable prefetch and replacement policies can effectively improve the cache hit ratio and reduce the memory access delay. Set the cache hit ratio to H and cache access time to T_{cache} , the memory access time is T_{memory} , the overall access delay T_{total} It can be expressed by the following formula:

$$T_{total} = H \times T_{cache} + (1-H) \times T_{memory} \tag{3}$$

Improve cache hit ratio by optimizing cache management H, can significantly reduce memory access time T_{total} To improve the overall system access efficiency. Intelligent cache management can dynamically adjust cache policies for different computing tasks, effectively reducing memory access bottlenecks, and improving low latency performance of the system.

3.4 Deepen hardware and software co-design to improve execution efficiency

The co-design of hardware and software is the key to reduce system delay. Through hardware and software co-design, the system's non-response time and task delay can be reduced by controlling and optimizing the running flow of hardware devices and software algorithms on the basis of ensuring the normal operation of the system. Hardware and software co-design includes hardware architecture design and software algorithm design. For example, the use of hardware accelerators (such as FPGA, GPU) for specific computing tasks to customize the hardware design can effectively improve performance. In addition, software can also take full advantage of related hardware devices by using relevant compilation modes or libraries (CUDA, OpenCL, etc.) and other tools. Therefore, it is necessary to maintain a high degree of consistency between hardware resource control and scheduling and allocation software algorithms to avoid the computation delay caused by resource waste and conflict. Set the total number of tasks to be executed in the system to N and the execution time of each task to T_i ,The parallelism of hardware resources is P, then the total execution time of the system T_{total} It can be expressed as:

$$T_{total} = \frac{1}{P} \sum_{i=1}^{N} T_i \tag{4}$$

By optimizing the parallelism P of hardware resources and task scheduling, it can be significantly reduced T_{total} To improve the execution efficiency of the system. The deepening of hardware and software co-design enables the system to utilize hardware resources more efficiently, reduce computing latency, and improve the overall performance of low-latency computing.

5. Conclusion

The real-time hardware production technology with low latency computing plays an important role in the context of the operation of high-intensity tasks and the increasing demand for fast response. This paper analyzes the factors that affect the efficiency of low-delay computing, and by introducing dynamic scheduling algorithm, improving the bandwidth of data transmission interface, optimizing cache management strategy and deepening the co-design of hardware and software, the performance and low delay of the system can be significantly improved, and lays a foundation for the further development and practical application of low-delay computing technology.

References

[1] *Zhou F, Ding H. Application of cloud and fog networks and QoS routing optimisation strategies for low delay. International Journal of Embedded Systems, 2023(3):16.*

[2] Fekete G, T. Kovácsházy. Execution of Resource Intensive Tasks on a Heterogeneous SoC for Low-Latency Embedded Compute. 2023 24th International Carpathian Control Conference (ICCC), 2023:124-129.

[3] Zhou Y, Ren Z Y, Shao E, et al. FILL: a heterogeneous resource scheduling system addressing the low throughput problem in GROMACS. CCF Transactions on High Performance Computing, 2024(1):6.

[4] Mo H, Zhu L. HetSev: Exploiting Heterogeneity-Aware Autoscaling and Resource-Efficient Scheduling for Cost-Effective Machine-Learning Model Serving. Electronics, 2023, 12(1):240-.

[5] K. Zhang, "Optimization and Performance Analysis of Personalized Sequence Recommendation Algorithm Based on Knowledge Graph and Long Short Term Memory Network, " 2025 2nd International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS), Hassan, India, 2025, pp. 1-6.

[6] Y. Zhao, "Design and Financial Risk Control Application of Credit Scoring Card Model Based on XGBoost and CatBoost, " 2025 International Conference on Intelligent Communication Networks and Computational Techniques (ICICNCT), Bidar, India, 2025, pp. 1-5.

[7] B. Li, "Research on the Spatial Durbin Model Based on Big Data and Machine Learning for Predicting and Evaluating the Carbon Reduction Potential of Clean Energy, " 2025 International Conference on Intelligent Communication Networks and Computational Techniques (ICICNCT), Bidar, India, 2025, pp. 1-5.

[8] Q. Xu, "Implementation of Intelligent Chatbot Model for Social Media Based on the Combination of Retrieval and Generation," 2025 2nd International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS), Hassan, India, 2025, pp. 1-7.

[9] Y. Zou, "Research on the Construction and Optimization Algorithm of Cybersecurity Knowledge Graphs Combining Open Information Extraction with Graph Convolutional Networks, " 2025 2nd International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS), Hassan, India, 2025, pp. 1-5.

[10] M. Zhang, "Research on Joint Optimization Algorithm for Image Enhancement and Denoising Based on the Combination of Deep Learning and Variational Models, " 2025 International Conference on Intelligent Communication Networks and Computational Techniques (ICICNCT), Bidar, India, 2025, pp. 1-5.

[11] W. Han, "Using Spark Streaming Technology to Drive the Real-Time Construction and Improvement of the Credit Rating System for Financial Customers, " 2025 International Conference on Intelligent Communication Networks and Computational Techniques (ICICNCT), Bidar, India, 2025, pp. 1-6.

[12] J. Huang, "Research on Multi-Model Fusion Machine Learning Demand Intelligent Forecasting System in Cloud Computing Environment, " 2025 2nd International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS), Hassan, India, 2025, pp. 1-7.

[13] J. Huang, "Performance Evaluation Index System and Engineering Best Practice of Production-Level Time Series Machine Learning System, " 2025 International Conference on Intelligent Communication Networks and Computational Techniques (ICICNCT), Bidar, India,

[14] X. Liu, "Research on User Preference Modeling and Dynamic Evolution Based on Multimodal Sequence Data," 2025 2nd International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS), Hassan, India, 2025, pp. 1-7.

[15] D. Shen, "Complex Pattern Recognition and Clinical Application of Artificial Intelligence in Medical Imaging Diagnosis, " 2025 International Conference on Intelligent Communication Networks and Computational Techniques (ICICNCT), Bidar, India, 2025, pp. 1-8.

[16] Wang Y. Application of Data Completion and Full Lifecycle Cost Optimization Integrating Artificial Intelligence in Supply Chain. 2025.

[17] Chen M. Research on Automated Risk Detection Methods in Machine Learning Integrating Privacy Computing. 2025.

[18] Shen, D. (2025). Construction And Optimization Of AI-Based Real-Time Clinical Decision Support System. Journal of Computer, Signal, and System Research, 2(7), 7-13.

[19] Hu, Q. (2025). The Practice and Challenges of Tax Technology Optimization in the Government Tax System. Financial Economics Insights, 2(1), 118-124.

[20] Sheng, C. (2025). Analysis of the Application of Fintech in Corporate Financial Decision-Making and Its Development Prospects. Financial Economics Insights, 2(1), 125-130.