Deadline-Aware Predictability Scheduling in Distributed Computing with GPU Clusters: A Survey

www.surveyx.cn

Abstract

This survey paper explores the intricacies of deadline-aware, predictability scheduling in distributed computing environments utilizing GPU clusters. It highlights the critical role of advanced scheduling techniques in optimizing system performance and resource utilization. Key findings emphasize the effectiveness of heuristic and algorithmic approaches, such as the Preferential Queueing Algorithm and Deadline-aware Power Management, in meeting stringent deadlines while optimizing energy efficiency. Predictive models and machine learning techniques, exemplified by the FIRM framework and Runtime Variation Prediction Method, significantly enhance scheduling predictability, allowing dynamic adjustments to changing conditions and maintaining high efficiency and reliability. Dynamic and adaptive scheduling approaches, such as Meta-Reinforcement Learning and Deadline-aware Two-Timescale Resource Allocation, are pivotal in managing fluctuating workloads and ensuring timely task execution. Specialized scheduling policies, tailored to use cases like industrial cyber-physical systems and VR video streaming, underscore the importance of customized strategies in optimizing resource allocation and enhancing performance. The survey identifies challenges in deadline-aware scheduling, including memory access interference and workload dynamics, complicating predictability and resource management. Addressing these challenges necessitates research into adaptive algorithms, machine learning integration, and optimizations for GPU platforms, alongside advancements in hardware virtualization. In conclusion, improved scheduling techniques hold the potential to significantly impact distributed computing environments by enhancing predictability, optimizing resource utilization, and ensuring deadline compliance, driving innovation across diverse applications.

1 Introduction

1.1 Significance of Deadline-Aware Scheduling

Deadline-aware scheduling is essential in distributed computing, ensuring timely task completion and enhancing system performance across diverse applications. This paradigm is particularly crucial in real-time systems using neural networks, where runtime variability can significantly impact outcomes [1]. By prioritizing deadline adherence, these systems optimize resource utilization and mitigate runtime fluctuations, which is vital for stable resource allocation.

In data centers, deadline-aware scheduling is pivotal for efficient flow management, directly influencing user experience through effective handling of both deadline-sensitive and regular flows [2]. This approach is similarly indispensable in multiplexing compute resources across microservices, where timely task completion enhances overall system efficiency [3].

The importance of deadline-aware scheduling extends to high-bandwidth applications, such as VR video streaming, where low end-to-end delay is critical [4]. In environments with concurrent AI

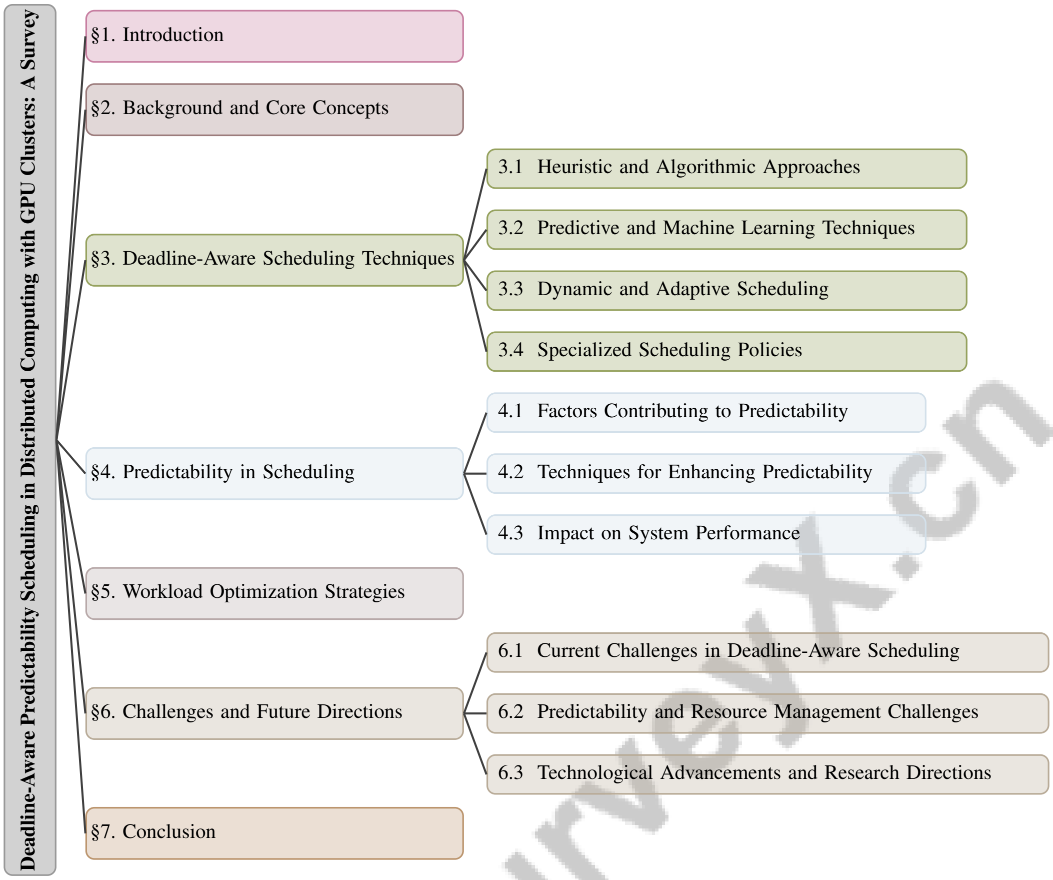
Deadline-Aware Predictability Scheduling in Distributed Computing with GPU Clusters: A Survey
§1. Introduction
§2. Background and Core Concepts
§3. Deadline-Aware Scheduling Techniques
3.1 Heuristic and Algorithmic Approaches
3.2 Predictive and Machine Learning Techniques
3.3 Dynamic and Adaptive Scheduling
3.4 Specialized Scheduling Policies
§4. Predictability in Scheduling
4.1 Factors Contributing to Predictability
4.2 Techniques for Enhancing Predictability
4.3 Impact on System Performance
§5. Workload Optimization Strategies
§6. Challenges and Future Directions
6.1 Current Challenges in Deadline-Aware Scheduling
6.2 Predictability and Resource Management Challenges
6.3 Technological Advancements and Research Directions
§7. Conclusion

Figure 1: chapter structure

applications, including Deep Neural Networks (DNNs) and transformers, this scheduling is vital for maintaining performance standards and ensuring timely execution.

Moreover, in processing large medical images, deadline-aware scheduling is essential for adhering to privacy, budget, and deadline constraints [5], underscoring its significance in contexts with high computational demands and sensitive data handling.

1.2 Challenges and Benefits of Predictability

Achieving predictability in distributed computing systems, especially those utilizing GPU clusters, presents several challenges that complicate effective scheduling and resource management. Interference within the memory subsystems of contemporary hardware disrupts predictability by impacting task execution timelines [1]. Additionally, privacy concerns hinder medical data custodians from exporting sensitive data to public cloud services, limiting resource allocation flexibility [5].

Dynamic network conditions further complicate predictability, as varying frame importance in video streaming can degrade quality [4]. Existing methods often inadequately manage low-level resources, leading to latency spikes that violate service-level objectives (SLOs) [3]. The need to balance job deadlines with energy costs in data centers adds another layer of complexity, as activating additional servers to meet deadlines incurs significant energy expenses not sufficiently addressed by current approaches [2].

Despite these challenges, the benefits of achieving predictability are substantial, including enhanced software dependability, improved performance forecasting in communication networks, and optimized resource allocation for job execution. Focusing on predictability mitigates risks associated with unpredictable behaviors, facilitates proactive network adaptation for time-critical applications, and employs analytical models like OptEx for accurate job completion time estimation and cost optimization [6, 7, 8]. Enhanced predictability leads to improved system reliability and performance, ensuring

AI-generated, for reference only.

efficient resource utilization and task scheduling, thereby meeting stringent deadline requirements and optimizing user experience.

1.3 Role of Workload Optimization

Workload optimization is crucial for enhancing system efficiency and meeting deadlines in distributed computing environments, particularly those employing GPU clusters. This process involves strategic resource management to maximize performance while adhering to deadline constraints. The Preferential Queueing Algorithm (PQA) exemplifies this by prioritizing requests based on deadlines, thus optimizing workloads for improved SLA compliance and resource utilization [9]. In real-time systems executing neural networks, workload optimization is vital for ensuring predictable timing and adequate computational resources [1].

In the context of large medical image processing, workload optimization is essential, as demonstrated by a privacy-preserving data splitting algorithm that optimizes service allocation while respecting privacy, budget, and deadline constraints [5]. The FIRM framework further illustrates the importance of workload optimization by employing a multilevel machine learning approach to detect SLO violations, localize their causes, and dynamically adjust resource allocations based on online telemetry data, thereby enhancing overall system efficiency [3].

Advanced optimization techniques, such as the two-timescale resource allocation scheme in VR video streaming, prioritize frame importance and deadlines, improving workload optimization in environments requiring low end-to-end delays [4]. Additionally, the integration of job deadlines with server activation energy constraints in the Deadline-aware Power Management method demonstrates how workload optimization can achieve significant energy savings while maintaining system performance [2].

1.4 Structure of the Survey

This survey is structured to provide a thorough exploration of deadline-aware scheduling and predictability in distributed computing environments with GPU clusters. It begins with an introduction highlighting the significance of deadline-aware scheduling and its impact on system performance, followed by a discussion of the challenges and benefits of achieving predictability. The introduction also emphasizes the critical role of workload optimization in meeting deadlines and maximizing GPU utilization.

The survey then presents a detailed background section elucidating core concepts of distributed computing and GPU clusters, along with key terminology essential for understanding subsequent discussions. Following this, the paper examines various deadline-aware scheduling techniques, including heuristic and algorithmic approaches, predictive and machine learning techniques, dynamic and adaptive scheduling methods, and specialized scheduling policies tailored for specific use cases.

Subsequent sections analyze factors contributing to predictability and techniques employed to enhance it, assessing their impact on overall system performance. The survey further explores workload optimization strategies, detailing effective resource allocation, task scheduling methods, dynamic resource management, and strategies for partitioning and load balancing to optimize workloads.

The paper concludes by addressing current challenges in deadline-aware scheduling and predictability, discussing potential technological advancements and research directions to enhance scheduling efficiency. This organized approach provides a comprehensive framework for understanding the intricate dynamics of deadline-aware scheduling and predictability in distributed computing environments, particularly those utilizing GPU clusters. It incorporates advanced modeling techniques, such as the OptEx model for job execution time estimation on Apache Spark, and leverages data-driven frequency scaling methods to optimize energy efficiency while meeting performance deadlines. Additionally, it addresses challenges posed by virtualization in high-performance computing and applies minimal-variance distributed scheduling algorithms to enhance predictability and stability, significantly improving resource utilization and reducing the likelihood of deadline violations in cloud-based environments [10, 11, 12, 13, 8].The following sections are organized as shown in Figure 1.

AI-generated, for reference only.

2 Background and Core Concepts

2.1 Overview of Distributed Computing and GPU Clusters

Distributed computing systems utilize interconnected nodes to enhance performance, reliability, and scalability for large-scale computations and data-intensive tasks. The architecture of these systems is crucial for managing complexity and facilitating efficient task execution, particularly in inter-datacenter networks where data replication strategies address capacity limitations [14]. Efficient scheduling of coflows within datacenter networks is essential for optimal data flow and resource allocation [14]. In cloud environments, resource architecture is pivotal for executing data-intensive jobs on parallel engines like Apache Spark, underscoring the importance of cost optimization and deadline-aware scheduling [8]. Effective resource management is imperative in dynamic cloud settings, where fluctuating workloads are driven by diverse traffic types [15]. This is further exemplified in industrial cyber-physical systems (ICPS), where efficient scheduling is necessary for real-time updates [16].

GPU clusters enhance distributed computing by providing computational power for intensive workloads. Systems like Clockwork, utilizing NVIDIA Tesla v100 GPUs, exemplify the architecture and functionality of GPU clusters in distributed settings [1]. These clusters are critical for applications requiring high parallelism and computational throughput, especially in healthcare diagnostics where AI and ML integration is essential [17]. The architecture must accommodate dynamic scheduling algorithms that adjust service rates to meet job demands and deadlines. For instance, the Generalized Exact Scheduling algorithm minimizes service capacity variance while fulfilling job requirements [18]. VR video streaming architectures, involving multiple streams sharing a bottleneck link, require effective resource allocation to meet quality demands [4]. Furthermore, dynamic power optimization challenges in data centers highlight the operational characteristics influencing energy efficiency, vital for maintaining energy-efficient operations while meeting performance requirements [2].

2.2 Key Terminology and Concepts

Understanding deadline-aware scheduling and predictability in distributed computing with GPU clusters requires familiarity with several key terminologies. In high-performance computing (HPC) environments, where applications operate under stringent deadlines, deadline-aware virtual machine schedulers are critical. These utilize real-time algorithms to dynamically optimize job execution, drawing on concepts from signal processing and statistical methods. Models like OptEx offer analytical frameworks for estimating job completion times and optimizing resource allocation in environments such as Apache Spark, achieving high accuracy in predicting execution times and cost-effective cluster configurations while adhering to service level objectives (SLOs) [8, 13].

’Predictability’ refers to a system’s ability to consistently meet predefined performance metrics, ensuring task execution aligns with expected timelines, crucial in environments where variability impacts performance. ’Scheduling’ involves the allocation and management of resources to execute tasks within specified deadlines, employing strategies and algorithms to optimize task execution across distributed systems. A significant aspect of scheduling is the ’SLO deadline’, indicating the performance target that tasks must meet [8].

’Workload optimization’ focuses on enhancing system efficiency by managing resources and tasks effectively. Techniques aim to optimize resource allocation, minimize latency, and maximize throughput, ensuring deadlines are met without compromising performance. The ’energy-efficient deadline-aware algorithm’ (E2DA) uses a contextual multi-armed bandit approach to balance energy consumption with deadline adherence [19]. Additionally, ’cost optimal cluster composition’ describes the optimal arrangement of cluster resources to minimize costs while meeting deadline requirements [8]. This is crucial for economic efficiency in distributed computing, especially in cloud environments where resource usage incurs financial implications.

These terminologies are essential for understanding the mechanisms and strategies of deadline-aware scheduling in distributed computing. They highlight the need to balance performance, efficiency, and cost-effectiveness, particularly in contexts such as dynamic service capacity adjustment, virtual machine scheduling, and big data stream analysis. Optimal distributed algorithms can minimize service capacity variance to meet strict demand and deadline requirements, while deadline-aware algorithms can adapt job execution in virtualized environments to ensure compliance with tight

AI-generated, for reference only.

deadlines. Frameworks like BCframework enhance scheduling efficiency by considering public cloud utilization costs and performance variations, minimizing latency and deadline violations [10, 14, 12, 13, 8].

3 Deadline-Aware Scheduling Techniques

Category	Feature	Method
Heuristic and Algorithmic Approaches	Time-Constrained Strategies	QC[20], DAVMS[13], PQA[9]
Predictive and Machine Learning Techniques	Scheduling and Adaptation	ONSOCMAX[21], OptEx[8], FIRM[3]
	Predictive Resource Management	BF[22], D-DVFS[11], RVPM[23]
	Cost and Performance Optimization	LAOF[24], BCF[10]
Dynamic and Adaptive Scheduling	Time-Conscious Strategies	TOSC[18]
	Predictive Decision-Making	MMPA[15]
Specialized Scheduling Policies	Time-Constrained Strategies	HLF-D[16], OGWP[17], PBD-SO[5], DTRM[4], DPM[2]

Table 1: This table provides a comprehensive overview of various deadline-aware scheduling techniques categorized into heuristic and algorithmic approaches, predictive and machine learning techniques, dynamic and adaptive scheduling, and specialized scheduling policies. Each category is detailed with specific methods, illustrating their contributions to optimizing resource management and ensuring compliance with strict deadlines in distributed computing environments.

This section explores methodologies within deadline-aware scheduling, emphasizing strategies that enhance resource management in distributed computing environments. Table 5 presents a detailed classification of deadline-aware scheduling techniques, emphasizing their roles in optimizing resource management and meeting deadlines in distributed computing systems. As illustrated in Figure **??**, the hierarchical structure of deadline-aware scheduling techniques categorizes them into heuristic and algorithmic approaches, predictive and machine learning techniques, dynamic and adaptive scheduling, and specialized scheduling policies. Each category is further detailed with specific methods and strategies, highlighting their contributions to optimizing resource management and meeting strict deadlines. Heuristic and algorithmic approaches are central to optimizing workload distribution and ensuring efficiency in these complex environments.

Figure 2: This figure illustrates the hierarchical structure of deadline-aware scheduling techniques, categorizing them into heuristic and algorithmic approaches, predictive and machine learning techniques, dynamic and adaptive scheduling, and specialized scheduling policies. Each category is further detailed with specific methods and strategies, highlighting their contributions to optimizing resource management and meeting deadlines in distributed computing environments.

3.1 Heuristic and Algorithmic Approaches

Method Name	**Optimization Techniques**	**Application Domains**	**Performance Metrics**
OptEx[8]	Constrained Optimization	Cloud Environments	Mean Relative Error
PQA[9]	Preferential Queueing Strategy	Mec Environments	Deadline Adherence
OGWP[17]	Cuda Mps	Medical AI Systems	Latency Determinism Metrics
DTRM[4]	Heuristic Algorithms	VR Video Streaming	Quality Loss
HLF-D[16]	Transmission Scheduling Policy	Industrial Cyber-physical Systems	Average Latency
PBD-SO[5]	Greedy Pareto Front	Hybrid Cloud Environment	Output Utility
DAVMS[13]	Deadline-aware Algorithm	Virtual Machines	Job Success Rates
MESC[25]	Instruction-level Preemption	Dnn Accelerators	Real-time Performance
QC[20]	Packet Clustering	Data Centers	Flow Completion Times
DPM[2]	Hungarian Algorithm	Data Centers	Energy Consumption

Table 2: Overview of various heuristic and algorithmic methods applied in distributed systems for deadline-aware scheduling, highlighting their optimization techniques, application domains, and performance metrics. This table provides a comparative analysis of methods like OptEx, PQA, OGWP, and others, emphasizing their contributions to enhancing efficiency and adherence to deadlines in diverse environments.

Table 2 presents a comprehensive comparison of heuristic and algorithmic approaches used in distributed systems to improve deadline-aware scheduling, detailing their optimization techniques, application domains, and performance metrics. Heuristic and algorithmic methods are vital in improving deadline-aware scheduling in distributed systems, especially in GPU clusters. The OptEx model, for example, achieves 98% accuracy in estimating job completion times on Apache Spark, optimizing cluster composition under Service Level Objectives (SLOs) [8]. Algorithms for virtualized environments dynamically adapt to execution delays, ensuring high-performance jobs meet deadlines. The BCframework enhances scheduling for Big Data applications by minimizing cost and latency [13, 10].

Heuristic methods like the Preferential Queueing Algorithm (PQA) prioritize requests by deadlines [9], while Optimized GPU Workload Partitioning (OGWP) leverages CUDA MPS for effective workload distribution [17]. The Deadline-aware Two-Timescale Resource Allocation combines queuing theory with heuristics to optimize VR video stream resources [4]. The HLF-D method prioritizes packet transmission by age and deadlines [16], and the Privacy-, Budget-, and Deadline-Aware Service Optimization (PBD-SO) balances privacy with efficient service allocation [5].

Algorithmic strategies, such as the Deadline-Aware Virtual Machine Scheduling (DAVMS), optimize job execution by adjusting to delays [13]. The MESC framework minimizes priority inversions through context switching [25], and QCluster adapts flow scheduling for limited queues [20]. In real-time neural networks, partitioning networks into subtasks ensures timely execution [1]. The Deadline-aware Power Management method addresses power management as a dynamic assignment problem [2]. Collectively, these strategies advance deadline-aware scheduling by optimizing resource use and deadline compliance.

3.2 Predictive and Machine Learning Techniques

Method Name	Predictability Enhancement	Resource Management	Machine Learning Integration
OptEx[8]	Accurate Estimations	Optimize Resource Allocation	-
BCF[10]	Deadline-aware Scheduling	Public Cloud Utilization	-
ONSOCMAX[21]	Deadline-aware Jobs	Optimize Resource Allocation	-
BF[22]	Proactive Scheduling Decisions	Better Resource Management	Predictive Analysis
LAOF[24]	Scheduling Accuracy	Resource Scheduling	Deep Learning
D-DVFS[11]	Deadline-aware Scheduling	Frequency Scaling Decisions	Machine Learning Models
RVPM[23]	Over 96FIRM[3]	Performance Predictability Improvements	Resource Management Framework
Machine Learning Framework			

Table 3: This table presents a comparative analysis of various predictive and machine learning techniques employed to enhance scheduling predictability and resource management in distributed systems. Each method is evaluated based on its ability to improve predictability, manage resources effectively, and integrate machine learning for performance optimization. The table highlights the diverse strategies adopted by different frameworks to meet deadlines and optimize system operations.

Predictive models and machine learning significantly enhance scheduling predictability in distributed systems with GPU clusters. Table 3 provides a comprehensive overview of methods that utilize predictive models and machine learning to enhance scheduling and resource management in distributed systems with GPU clusters. These approaches improve resource allocation and task scheduling, ensuring deadlines are met with high accuracy. The OptEx model achieves 98% accuracy in job completion time estimation under SLOs [8]. The BCframework addresses streaming Big Data challenges by integrating cloud costs and performance variations [10].

The ONSOCMAX method enhances predictability by adjusting to system conditions [21], while the Beacons Framework uses machine learning to predict resource usage [22]. Learning-assisted optimization combines deep learning with iterative methods for efficiency [24]. A data-driven frequency scaling method predicts energy use and execution time, aiding resource management [11]. The Runtime Variation Prediction Method (RVPM) accurately predicts job runtimes with machine learning [23].

OptEx refines heuristics for deadline-aware scheduling, enhancing resource allocation and reliability [8]. The FIRM framework uses machine learning to dynamically adjust resources and prevent SLO violations [3]. These techniques improve scheduling predictability by adjusting service capacity to workloads and ensuring deadline compliance through innovative algorithms [8, 22, 13, 12].

Method Name	Scheduling Techniques	Performance Optimization	Resource Adaptation
MMPA[15]	Markov Decision Process	Resource Efficiency	Dynamic Resource Allocation
HLF-D[16]	Age-based Scheduling	Maximize Expected Utility	Dynamic Network Conditions
TOSC[18]	Dual Fitting Technique	Maximize Job Completion	Preemption Rule Applied
MESC[25]	Context Switching	Instruction-level Preemption	Resource Allocation
DTRM[4]	Two-timescale Scheme	Frame Importance Model	Short-timescale Adjustments
DPM[2]	Hungarian Algorithm	Energy Consumption	Adaptive Algorithms

Table 4: Summary of various dynamic and adaptive scheduling methods employed in distributed GPU cluster environments, highlighting their scheduling techniques, performance optimization strategies, and resource adaptation mechanisms. The table provides a comparative analysis of methods such as MMPA, HLF-D, TOSC, MESC, DTRM, and DPM, each leveraging distinct approaches to enhance system responsiveness and efficiency.

3.3 Dynamic and Adaptive Scheduling

Dynamic and adaptive scheduling is crucial in distributed environments using GPU clusters, improving system responsiveness to workload changes. Techniques like data-driven frequency scaling and deadline-aware algorithms optimize energy use while meeting performance needs, enhancing job completion in cloud and HPC applications [11, 21, 12, 13, 8]. Table 4 presents a comparative overview of dynamic and adaptive scheduling methods, illustrating their respective techniques and strategies for optimizing performance and resource allocation in distributed GPU cluster environments.

The Meta-Reinforcement Learning Approach adapts resource policies based on system states [15]. In ICPS, Deadline-Aware Scheduling adjusts service rates to maximize information freshness [16]. The Generalized Exact Scheduling algorithm minimizes service capacity variance [18], and the MESC framework allows flexible resource allocation [25].

In video streaming, dynamic scheduling manages streams sharing a bottleneck, with the Deadline-aware Two-Timescale Resource Allocation ensuring low latency and quality [4]. The Deadline-aware Power Management method optimizes energy efficiency while meeting deadlines [2]. These approaches enhance resource allocation and task execution efficiency, leveraging advanced techniques to optimize performance by predicting application behavior under various workloads [21, 13, 12, 11].

3.4 Specialized Scheduling Policies

Specialized scheduling policies address specific needs in distributed computing, enhancing performance and resource use. These policies manage diverse workloads within deadline and resource constraints using advanced techniques like online dispatching and intelligent resource management [14, 21, 3, 12].

The Deadline-aware Service Optimization (DASO) framework balances privacy with efficient service allocation, optimizing workloads while respecting constraints [5]. In ICPS, policies ensure real-time updates and freshness through Deadline-Aware Scheduling [16].

For video streaming, the Deadline-aware Two-Timescale Resource Allocation prioritizes frame importance and deadlines [4]. In medical applications, AI and ML technologies require policies ensuring deterministic latency for diagnostics [17].

In data centers, policies address power optimization, balancing deadlines with energy costs [2]. These policies enhance resource use and task execution in distributed environments, addressing challenges like service locality and resource contention, ultimately improving performance metrics [14, 21, 13, 12]. By addressing constraints, these policies improve system performance, reliability, and efficiency.

4 Predictability in Scheduling

Predictability in scheduling is critical for enhancing efficiency in distributed computing environments, especially those utilizing GPU clusters, by reducing costs associated with unpredictability and service capacity instability. This is essential for applications with strict deadlines and varying workloads, where optimal scheduling algorithms minimize both the mean and variance of service capacity. Techniques such as data-driven frequency scaling and deadline-aware scheduling employ predictive

Feature	Heuristic and Algorithmic Approaches	Predictive and Machine Learning Techniques	Dynamic and Adaptive Scheduling
Optimization Focus	Workload Distribution	Scheduling Predictability	System Responsiveness
Application Domain	Gpu Clusters	Distributed Systems	Cloud Applications
Performance Metric	98		

Table 5: This table provides a comparative analysis of various deadline-aware scheduling techniques, categorized into heuristic and algorithmic approaches, predictive and machine learning techniques, and dynamic and adaptive scheduling. It highlights the optimization focus, application domains, and performance metrics associated with each category, offering insights into their effectiveness in distributed computing environments. The table underscores the distinct methodologies employed to enhance workload distribution, scheduling predictability, and system responsiveness.

models to optimize energy consumption while ensuring real-time performance requirements, thus improving system reliability and resource utilization [13, 12, 11]. Understanding the factors contributing to predictability is crucial for analyzing their interactions, enhancing scheduling performance, and meeting system demands.

4.1 Factors Contributing to Predictability

Several factors are pivotal in shaping predictability within distributed computing environments, particularly GPU clusters, ensuring timely task completion and optimized performance. Prioritization of requests based on deadlines, as managed by algorithms like the Preferential Queueing Algorithm, enhances scheduling predictability [9]. Static scheduling of memory access and centralized orchestration reduce variability in task execution timelines [1]. Privacy-preserving data splitting algorithms ensure efficient service allocation under privacy constraints, stabilizing system operations [5]. The FIRM framework enhances predictability by adaptively learning from workloads and resource contention, adjusting resource allocations based on real-time data [3]. In VR video streaming, modeling frame importance ensures high-priority frames meet deadlines [4]. Accurate predictions of job arrival times and service demands, as demonstrated by the Deadline-aware Power Management method, are crucial for effective power management [2]. Challenges such as subsystem unpredictability, performance forecasting complexities in communication networks, and transaction latency variances in database systems necessitate robust analytical and adaptive strategies to optimize scheduling performance [26, 6, 7].

4.2 Techniques for Enhancing Predictability

Enhancing predictability in task execution within distributed computing environments, particularly those utilizing GPU clusters, requires optimizing resource allocation and ensuring consistent performance. The mathematical framework by Mostafavi et al. defines predictability through information theory, providing a quantitative basis for optimizing scheduling strategies [6]. Predictive modeling techniques forecast task execution times and resource demands, enabling proactive management of performance and quality of service across applications like communication networks and machine learning systems [6, 7, 8, 27]. Robust load balancing algorithms, such as the Preferential Queueing Algorithm, prioritize tasks based on deadlines and importance, ensuring critical tasks receive necessary resources [9]. Techniques like static memory access scheduling and centralized resource management minimize execution timeline variability [1]. Adaptive frameworks like FIRM enhance predictability by learning from real-time telemetry data to dynamically adjust resource allocations [3]. Integrated approaches underscore the importance of predictive modeling, robust scheduling algorithms, and adaptive resource management in enhancing predictability. Frameworks like OptEx achieve a mean relative error of just 6

4.3 Impact on System Performance

Enhanced predictability significantly impacts the performance and efficiency of distributed computing systems, particularly GPU clusters. It is crucial for managing the complex relationship between application performance, energy consumption, and execution time, especially in dynamic workloads such as AI and deep learning. Data-driven frequency scaling optimizes GPU clock settings based on predictive models, leading to energy-efficient scheduling that meets application deadlines. Novel deadline-aware scheduling algorithms dynamically adjust to execution delays, ensuring high-performance computing jobs adhere to strict deadlines while maximizing hardware utilization

[13, 23, 11]. The DDCCast method illustrates how improved predictability can reduce total bandwidth usage by up to 45

5 Workload Optimization Strategies

5.1 Resource Allocation and Task Scheduling

Efficient resource allocation and task scheduling are pivotal in optimizing workloads within distributed computing environments, particularly GPU clusters. These strategies ensure task execution meets performance requirements while minimizing resource wastage. The SLAMIG method enhances workload management by grouping migration tasks based on resource independence [28]. Meanwhile, the MMPA method predicts future workloads to improve CPU utilization, showcasing a proactive approach to resource management [15]. In flow scheduling, QCluster improves system efficiency by clustering packets according to their properties [20]. FIRM leverages telemetry data and machine learning for real-time resource management, dynamically adjusting resources to meet fluctuating demands [3]. Dynamic server assignment based on energy consumption and deadlines further balances job requirements with energy efficiency, as demonstrated by the Deadline-aware Power Management approach [2]. These techniques underscore the importance of integrating predictive insights and dynamic management to enhance workload efficiency, as seen in frameworks like BCframework and OptEx, which improve performance in processing streaming Big Data [8, 10].

5.2 Dynamic Resource Management

Dynamic resource management is crucial for adapting to fluctuating workload demands in distributed computing, especially GPU clusters. This adaptability is essential for optimizing performance and energy efficiency, particularly in heterogeneous scenarios involving AI and machine learning workloads. Techniques like Dynamic Voltage Frequency Scaling (DVFS) adjust clock frequencies based on real-time performance to manage energy consumption effectively. Predictive models forecast power usage and execution time, enabling scheduling algorithms that meet deadlines while minimizing energy costs. Frameworks like FIRM use machine learning to reduce resource contention among microservices, enhancing predictability and reducing SLO violations [3, 23, 11]. The Meta-Reinforcement Learning Approach adapts resource policies based on system states, allowing real-time adjustments [15]. In industrial cyber-physical systems, dynamic management maintains real-time updates and maximizes information freshness through methods like Deadline-Aware Scheduling [16]. Dynamic power optimization in data centers, as seen in the Deadline-aware Power Management method, balances job deadlines with energy efficiency [2]. In video streaming, dynamic resource management ensures quality requirements are met through methods like Deadline-aware Two-Timescale Resource Allocation [4]. These strategies highlight the importance of dynamic management in optimizing resource utilization and maintaining service quality.

5.3 Partitioning and Load Balancing

Partitioning and load balancing are critical for optimal workload distribution and resource utilization in distributed computing environments, particularly GPU clusters. These strategies manage computational demands during peak times and ensure timely job completion to meet SLOs. The OptEx model for Apache Spark demonstrates how analytical approaches can optimize cluster composition, achieving high accuracy in resource allocation [8]. In multi-access edge computing, advanced load orchestration techniques prioritize requests to enhance system performance and deadline adherence [9]. The Optimized GPU Workload Partitioning (OGWP) method utilizes CUDA Multi-Process Service for spatial partitioning, distributing workloads efficiently across GPU resources [17]. Load balancing strategies, such as the Preferential Queueing Algorithm, prioritize tasks based on deadlines to ensure critical workloads receive necessary resources [9]. In cloud environments, the Meta-Reinforcement Learning Approach adapts resource allocation policies in real-time for effective workload distribution [15]. The FIRM framework enhances load balancing by using machine learning to manage resources across microservices, adjusting distributions based on telemetry data [3]. In video streaming, load balancing manages multiple streams over a bottleneck link to ensure low latency and high-quality service [4]. These strategies are vital for optimizing resource utilization, addressing challenges like resource contention and service locality, and ensuring applications meet execution deadlines while minimizing power consumption [21, 8, 9, 11].

6 Challenges and Future Directions

6.1 Current Challenges in Deadline-Aware Scheduling

Effective deadline-aware scheduling in distributed GPU cluster computing faces significant challenges impacting performance and resource utilization. Hardware architectural limitations, such as memory access interference, create unpredictable behavior hindering scheduling [1]. Oversimplified communication overhead models, especially when processing large medical images, exacerbate computational demands and jeopardize deadline adherence [5]. Fluctuating workloads, particularly during peak times in distributed load orchestration [9] and VR video streaming (due to dynamic network conditions) [4], further complicate deadline compliance. Traditional methods struggle with resource contention and workload fluctuations, failing to adapt to dynamic demands [3]. Accurate prediction of job arrivals and service demands also remains problematic, impacting real-time scheduling and deadline adherence [2]. These challenges necessitate innovative, adaptive scheduling methods to mitigate service capacity unpredictability and reduce operational inefficiencies and infrastructure costs. Research into optimal distributed algorithms, such as Exact Scheduling, and novel deadline-aware virtual machine algorithms, incorporating signal processing and statistical techniques, aims to minimize service capacity variance while meeting stringent demand and deadline requirements [13, 12].

6.2 Predictability and Resource Management Challenges

Predictability and effective resource management in distributed GPU cluster computing are hampered by several key challenges. Variability in task execution times, often exacerbated by memory subsystem interference in current hardware architectures [1], complicates scheduling and deadline compliance. Dynamic workloads, especially in VR video streaming, require balancing frame importance and network conditions to minimize quality loss while maintaining low latency [4]. Resource contention significantly impacts predictability, with traditional methods often failing to effectively manage low-level resources, leading to latency spikes and service-level objective violations [3]. This is compounded by the need for dynamic resource allocation in response to fluctuating workloads, demanding sophisticated real-time telemetry and adaptive scheduling strategies. Privacy concerns, stemming from data custodians' reluctance to transfer sensitive data to public clouds [5], further limit resource allocation flexibility and scheduling predictability. Modeling communication overhead in large medical image processing contributes to high computational demands, further complicating resource management. In data centers, balancing job deadlines with energy costs remains a challenge, as activating additional servers to meet deadlines incurs significant energy expenses [2]. These challenges highlight the need for advanced scheduling algorithms to enhance predictability and stability, adaptive resource management strategies considering power and activation constraints, and innovative solutions to balance competing demands while minimizing operational costs. This includes minimal-variance distributed scheduling techniques meeting hard and soft deadlines and deadline-aware power management strategies optimizing server assignments and power consumption [2, 12].

6.3 Technological Advancements and Research Directions

Advancements in deadline-aware scheduling and predictability are crucial for optimizing distributed GPU cluster computing. Future research should integrate deadline-aware scheduling with adaptive transmission architectures for improved VR video streaming performance, addressing dynamic network conditions and frame importance [4]. Developing adaptive algorithms leveraging historical data for improved job arrival predictions is promising, enhancing scheduling in dynamic environments, particularly in data centers balancing job deadlines and energy costs [2]. Extending scheduling capabilities from single nodes to clusters, incorporating live virtual machine migration to enhance job throughput and deadline adherence, is also important. This improves resource management efficiency and scheduling outcomes in complex distributed systems through advanced workload dispatching policies, minimizing service capacity variance, and using predictive analytics for runtime consistency [21, 23, 12, 3, 8]. Integrating machine learning techniques into scheduling algorithms can provide predictive insights enhancing efficiency and predictability through data-driven adaptive resource allocation and task prioritization based on real-time analytics [21, 23, 12]. Optimizations for single GPU platforms, such as data-driven frequency scaling and advanced hardware virtualization technologies, are vital for high-demand applications like medical diagnostics [5, 17, 11, 23, 13]. Enhancing framework adaptability to various DNN accelerators and OS kernels could advance distributed scheduling techniques. These research directions highlight the potential of technological advancements to address challenges in deadline-aware scheduling and predictability. Innovative scheduling algorithms, such as the deadline-aware highest latency first policy for industrial cyber-physical systems [16], minimal-variance distributed algorithms for enhanced predictability and stability [12], and novel deadline-aware virtualization approaches for scientific grids and cloud computing [13], along with frameworks like BCframework for streaming Big Data [10], demonstrate a multifaceted approach to improving scheduling efficiency and predictability.

7 Conclusion

This survey underscores the critical role of advanced scheduling techniques in enhancing performance and resource utilization in distributed computing environments with GPU clusters. The exploration of heuristic and algorithmic methods, such as the Preferential Queueing Algorithm and Deadline-aware Power Management, illustrates their effectiveness in meeting deadlines while optimizing energy consumption. Predictive models and machine learning frameworks, including the FIRM framework and Runtime Variation Prediction Method, are pivotal in refining scheduling predictability, allowing systems to adapt to dynamic conditions efficiently and reliably.

Dynamic and adaptive scheduling approaches, exemplified by the Meta-Reinforcement Learning Approach and Deadline-aware Two-Timescale Resource Allocation, are essential for managing fluctuating workloads and ensuring prompt task execution. Tailored scheduling policies for specific applications, such as industrial cyber-physical systems and VR video streaming, highlight the importance of customized strategies in resource allocation and performance enhancement.

The survey also identifies persistent challenges in deadline-aware scheduling, such as memory access interference and workload variability, which impede predictability and resource management. Addressing these challenges requires continued research into adaptive algorithms, the integration of machine learning, optimizations tailored to GPU architectures, and advancements in hardware virtualization technologies.

AI-generated, for reference only.

References

[1] Maximilian Kirschner, Konstantin Dudzik, and Jürgen Becker. Work-in-progress: Real-time neural network inference on a custom risc-v multicore vector processor, 2024.

[2] Cengis Hasan and Zygmunt J. Haas. Deadline-aware power management in data centers, 2015.

[3] Haoran Qiu, Subho S. Banerjee, Saurabh Jha, Zbigniew T. Kalbarczyk, and Ravishankar K. Iyer. Firm: An intelligent fine-grained resource management framework for slo-oriented microservices, 2020.

[4] Qingxuan Feng, Peng Yang, Zhixuan Huang, Jiayin Chen, and Ning Zhang. Deadline aware two-timescale resource allocation for vr video streaming, 2023.

[5] Yuandou Wang, Neel Kanwal, Kjersti Engan, Chunming Rong, Paola Grosso, and Zhiming Zhao. Towards privacy-, budget-, and deadline-aware service optimization for large medical image processing across hybrid clouds, 2024.

[6] Samie Mostafavi, Simon Egger, György Dán, and James Gross. Predictability of performance in communication networks under markovian dynamics, 2024.

[7] George Candea. Predictable software – a shortcut to dependable computing ?, 2004.

[8] Subhajit Sidhanta, Wojciech Golab, and Supratik Mukhopadhyay. Optex: A deadline-aware cost optimization model for spark, 2016.

[9] Ricardo N. Boing, Hugo Vaz Sampaio, Fernando Koch, Rene N. S. Cruz, and Carlos B. Westphall. Distributed load orchestration for vision computing in multi-access edge computing, 2022.

[10] Mahmood Mortazavi-Dehkordi and Kamran Zamanifar. Efficient deadline-aware scheduling for the analysis of big data streams in public cloud. *Cluster Computing*, 23(1):241–263, 2020.

[11] Shashikant Ilager, Rajeev Muralidhar, Kotagiri Rammohanrao, and Rajkumar Buyya. A data-driven frequency scaling approach for deadline-aware energy efficient scheduling on graphics processing units (gpus), 2020.

[12] Yorie Nakahira, Andres Ferragut, and Adam Wierman. Minimal-variance distributed deadline scheduling, 2020.

[13] Omer Khalid, Ivo Maljevic, Richard Anthony, Miltos Petridis, Kevin Parrot, and Markus Schulz. Deadline aware virtual machine scheduler for scientific grids and cloud computing, 2010.

[14] Mohammad Noormohammadpour and Cauligi S. Raghavendra. Comparison of flow scheduling policies for mix of regular and deadline traffic in datacenter environments, 2017.

[15] Siqiao Xue, Chao Qu, Xiaoming Shi, Cong Liao, Shiyi Zhu, Xiaoyu Tan, Lintao Ma, Shiyu Wang, Shijun Wang, Yun Hu, Lei Lei, Yangfei Zheng, Jianguo Li, and James Zhang. A meta reinforcement learning approach for predictive autoscaling in the cloud, 2022.

[16] Devarpita Sinha and Rajarshi Roy. Deadline-aware scheduling for maximizing information freshness in industrial cyber-physical system, 2020.

[17] Soham Sinha, Shekhar Dwivedi, and Mahdi Azizian. Towards deterministic end-to-end latency for medical ai systems in nvidia holoscan, 2024.

[18] Yossi Azar, Inna Kalp-Shaltiel, Brendan Lucier, Ishai Menache, Joseph, Naor, and Jonathan Yaniv. Truthful online scheduling with commitments, 2015.

[19] Babak Badnava, Keenan Roach, Kenny Cheung, Morteza Hashemi, and Ness B Shroff. Energy-efficient deadline-aware edge computing: Bandit learning with partial observations in multi-channel systems, 2023.

[20] Tong Yang, Jizhou Li, Yikai Zhao, Kaicheng Yang, Hao Wang, Jie Jiang, Yinda Zhang, and Nicholas Zhang. Qcluster: Clustering packets for flow scheduling, 2022.

AI-generated, for reference only.

[21] Hailiang Zhao, Shuiguang Deng, Jianwei Yin, Schahram Dustdar, and Albert Y. Zomaya. Theoretically guaranteed online workload dispatching for deadline-aware multi-server jobs, 2022.

[22] Girish Mururu, Sharjeel Khan, Bodhisatwa Chatterjee, Chao Chen, Chris Porter, Ada Gavrilovska, and Santosh Pande. Compiler-guided throughput scheduling for many-core machines, 2021.

[23] Yiwen Zhu, Rathijit Sen, Robert Horton, John Mark, and Agosta. Runtime variation in big data analytics, 2023.

[24] Lei Lei, Lei You, Qing He, Thang X Vu, Symeon Chatzinotas, Di Yuan, and Björn Ottersten. Learning-assisted optimization for energy-efficient scheduling in deadline-aware noma systems. *IEEE Transactions on Green Communications and Networking*, 3(3):615–627, 2019.

[25] Jiapeng Guan, Ran Wei, Dean You, Yingquan Wang, Ruizhe Yang, Hui Wang, and Zhe Jiang. Mesc: Re-thinking algorithmic priority and/or criticality inversions for heterogeneous mcss, 2024.

[26] Jiamin Huang, Barzan Mozafari, Grant Schoenebeck, and Thomas Wenisch. Identifying the major sources of variance in transaction latencies: Towards more predictable databases, 2016.

[27] Arpan Gujarati, Reza Karimi, Safya Alzayat, Wei Hao, Antoine Kaufmann, Ymir Vigfusson, and Jonathan Mace. Serving {DNNs} like clockwork: Performance predictability from the bottom up. In *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*, pages 443–462, 2020.

[28] TianZhang He, Adel N. Toosi, and Rajkumar Buyya. Sla-aware multiple migration planning and scheduling in sdn-nfv-enabled clouds, 2021.

AI-generated, for reference only.

Disclaimer:

SurveyX is an AI-powered system designed to automate the generation of surveys. While it aims to produce high-quality, coherent, and comprehensive surveys with accurate citations, the final output is derived from the AI's synthesis of pre-processed materials, which may contain limitations or inaccuracies. As such, the generated content should not be used for academic publication or formal submissions and must be independently reviewed and verified. The developers of SurveyX do not assume responsibility for any errors or consequences arising from the use of the generated surveys.