Analysis and Application of Computer Queueing Theory

Siyang Ding [1, *]

[1] Southwest Jiaotong University, Chengdu, 610031, China

*el22sd2@leeds.ac.uk

Abstract. Computer queueing theory is a vital mathematical discipline that focuses on the study of queue behavior in computer systems. Its significance has grown substantially in recent years due to the increasing demand for high-performance computer systems. This paper aims to provide a comprehensive analysis of computer queueing theory, encompassing its historical background, fundamental concepts, and mathematical models. The paper begins by delving into the history of queueing theory in computer systems, highlighting key milestones and influential contributors. It outlines how the discipline has evolved over time to address the specific challenges and requirements of computer systems. The fundamental concepts of computer queueing theory are then explored in detail. This includes examining the characteristics of queues, such as arrival rates, service rates, queue length, and waiting times. The paper elucidates various queueing models, including single-server queues, multi-server queues, and network queues, emphasizing their relevance and applicability in computer systems. Moreover, the paper investigates the practical applications of queueing theory in computer systems. Performance analysis is one significant application, as queueing models enable the evaluation of system performance metrics such as throughput, response time, and utilization.

Keywords: Computer Queueing Theory, Computer Systems, Comprehensive analysis

1 Introduction

Computer queueing theory is a branch of mathematics that deals with the analysis of queues in computer systems [1]. It was first introduced by Agner Krarup Erlang in 1909 to study the behavior of telephone networks. Since then, it has been widely used in various fields, including computer science, engineering, and operations research. In computer systems, queueing theory is used to analyze the behavior of queues, such as the length of queues, waiting times, and service times. This information is critical for system designers and administrators to optimize system performance and resource utilization [2].

P. Kar et al. (eds.), *Proceedings of the 2023 International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2023)*, Advances in Computer Science Research 108,
https://doi.org/10.2991/978-94-6463-300-9_45

2 Queueing Theory Fundamentals

The fundamental concepts of computer queueing theory include arrival process, service process, queue discipline, and system capacity. The arrival process refers to the pattern of arrivals of tasks or requests to the system. The service process refers to the time required to complete a task or request. The queue discipline refers to the rules that determine the order in which tasks or requests are serviced [3]. The system capacity refers to the maximum number of tasks or requests that the system can handle at a given time. Queueing theory is a mathematical tool used to analyze and optimize system performance, and it has wide applications in computer systems. In computer systems, queueing theory can be used to analyze and optimize various systems, including operating systems, networks, databases, distributed systems, and more. By using queueing theory, system performance and throughput can be predicted, and system designers can optimize resource utilization, improve system reliability and scalability [4].

2.1 Basic Concepts and Definitions

Queueing theory is a branch of mathematics that deals with the analysis and modeling of waiting lines or queues. It provides a mathematical framework to study the behavior and performance of queueing systems. Here are some basic concepts and definitions: Queue: A queue represents a waiting line where entities, referred to as customers or jobs, arrive and wait for service. Arrival Process: The arrival process determines the pattern and timing of customer arrivals. It can be modeled using statistical distributions, such as Poisson or exponential distribution.

Service Process: The service process represents the time required to serve a customer [4]. It can also be modeled using statistical distributions, such as exponential or Erlang distribution. Queue Length: The queue length refers to the number of customers waiting in the queue at a given point in time. Queue Discipline: The queue discipline specifies the rules for customer service allocation. Common disciplines include first-come-first-served (FCFS), last-come-first-served (LCFS), and priority-based disciplines. Service Time: The service time is the time required to complete the service for a customer. Arrival Rate: The arrival rate represents the average number of customers arriving per unit of time. Service Rate: The service rate represents the average number of customers served per unit of time. Utilization: The utilization of a queueing system refers to the fraction of time that the server is busy serving customers [5]. It is calculated as the ratio of the service rate to the arrival rate.

2.2 Characteristics of a Queueing System

A queueing system is characterized by several key elements and performance measures: Arrival Process: The arrival process determines how customers arrive at the queueing system. It can be modeled by the arrival rate and the distribution of interarrival times. Service Process: The service process determines how customers are served. It can be modeled by the service rate and the distribution of service times.

Queue Length: The queue length indicates the number of customers waiting in the queue at a particular time. Waiting Time: The waiting time is the amount of time a customer spends in the queue before being served [6]. Service Time: The service time is the amount of time required to complete the service for a customer. Utilization: The utilization is the ratio of the service rate to the arrival rate, indicating the percentage of time the server is busy. Performance Measures: Performance measures of interest include average queue length, average waiting time, system throughput, probability of waiting, and server utilization.

2.3 Queueing Models and Their Properties

Queueing theory offers various models to analyze different types of queueing systems. Some commonly used models include: M/M/1 Queue: The M/M/1 queue represents a single-server queueing system with Poisson arrivals, exponentially distributed service times, and a first-come-first-served discipline. M/M/c Queue: The M/M/c queue extends the M/M/1 model to include multiple servers (c) serving customers in parallel. M/G/1 Queue: The M/G/1 queue represents a queueing system with Poisson arrivals, general (G) service time distribution, and a single server. M/D/1 Queue: The M/D/1 queue assumes deterministic (D) service times, where each customer receives the same fixed service time. M/M/∞ Queue: The M/M/∞ queue represents a queueing system with Poisson arrivals, exponentially distributed service times, and an infinite number of servers [7].

3 Applications of Queueing Theory in Computer Systems

In operating systems, queueing theory can be used to analyze and optimize process scheduling algorithms, memory management algorithms, file systems, and more. In networks, queueing theory can be used to analyze and optimize routing algorithms, congestion control algorithms, and more. In databases, queueing theory can be used to analyze and optimize query processing algorithms, transaction processing algorithms, and more [8]. In distributed systems, queueing theory can be used to analyze and optimize distributed algorithms, data consistency algorithms, and more. Queueing theory is an important tool for computer system design and optimization. It can help system designers predict and optimize system performance, improve system reliability and scalability.

3.1 Traffic Modeling and Performance Evaluation of Computer Networks

Queueing theory plays a crucial role in modeling and analyzing the performance of computer networks. By representing the network as a series of interconnected queues, various aspects of network behavior can be studied, including: Traffic Modeling: Queueing theory helps in modeling and characterizing network traffic patterns, such as arrival rates, packet sizes, and interarrival times. This information is vital for designing efficient network protocols and determining network capacity requirements.

Congestion Analysis: Queueing models can be used to evaluate congestion in computer networks, identifying bottlenecks and estimating the queuing delays experienced by packets. This information helps in optimizing network configurations and implementing congestion control mechanisms. Quality of Service : Queueing theory enables the analysis of different QoS metrics, such as delay, packet loss, and throughput. By considering different service disciplines and prioritization schemes, it helps in designing and optimizing QoS-aware network architectures [9].

Traffic Engineering: Queueing models aid in traffic engineering by studying the impact of routing strategies, load balancing techniques, and resource allocation policies. This allows network administrators to optimize network performance and achieve efficient resource utilization.

3.2 Resource Allocation and Scheduling in Cloud Computing Systems

Queueing theory provides valuable insights into resource allocation and scheduling problems in cloud computing systems, where multiple tasks or jobs compete for shared resources. Some specific applications include: Virtual Machine Placement: Queueing models help in determining the optimal placement of VMs on physical servers. By considering factors such as resource requirements, workload characteristics, and performance objectives, queueing models aid in efficient resource allocation. Task Scheduling: Queueing theory can be used to analyze different task scheduling policies in cloud environments. By considering the arrival patterns of tasks, their resource requirements, and service times, queueing models help in designing scheduling algorithms that optimize resource utilization and minimize task completion times. Load Balancing: Queueing models assist in load balancing by studying the distribution of tasks among servers.

3.3 Performance Analysis and Optimization of Computer Systems

Queueing theory provides a powerful framework for analyzing the performance of computer systems, ranging from single servers to complex distributed architectures. Some areas of application include: CPU Scheduling: Queueing models are used to analyze different CPU scheduling algorithms, evaluating metrics such as response time, throughput, and fairness. This helps in optimizing scheduling policies and improving overall system performance. Disk and I/O Systems: Queueing theory aids in modeling and analyzing disk access and I/O operations. By considering disk service times, arrival patterns of requests, and device characteristics, queueing models help in optimizing disk scheduling algorithms and improving system throughput. Web Server Performance: Queueing models can be applied to analyze the performance of web servers, considering factors such as request arrival rates, service times, and server capacities. This helps in optimizing web server configurations, load balancing strategies, and resource allocation policies. Distributed Systems: Queueing theory is utilized to analyze the performance of distributed systems, including client-server architectures, distributed databases, and content delivery networks.

4 Simulation Techniques for Queueing Systems

Simulation techniques for queueing systems are essential for analyzing and optimizing system performance. They help simulate and evaluate various system designs and algorithms for operating systems, networks, databases, and distributed systems. These techniques can be applied to process scheduling, memory management, routing, congestion control, query processing, transaction processing, distributed algorithms, and data consistency. By using simulation techniques, system designers can understand system performance and optimize it accordingly, without the cost and risks of experimentation and testing in actual systems. Overall, simulation techniques for queueing systems are a valuable tool for understanding and evaluating different system designs and algorithms [10].

4.1 Discrete Event Simulation and its Properties

Discrete event simulation (DES) is a widely used technique for analyzing and modeling queueing systems. In DES, the system's behavior is simulated over time by modeling the occurrence of discrete events that affect the system's state. The simulation progresses in discrete time steps, and events are processed sequentially based on their timestamps. Properties of discrete event simulation include: Event-based modeling: DES focuses on modeling the occurrence of events that impact the system. These events can represent arrivals, departures, service completions, and other system-specific actions. Time advancement: The simulation progresses in discrete time steps, with events occurring at specific timestamps. After processing an event, the simulation time is advanced to the timestamp of the next event. Event scheduling: Events are scheduled in a priority queue, sorted by their timestamps. The simulation engine retrieves the next event from the queue and processes it.

4.2 Simulation Models for Queueing Systems

Simulation models for queueing systems are constructed using various components and parameters to capture the behavior of the system. Key elements of a simulation model for queueing systems include: Arrival Process: The arrival process determines how customers enter the system. It can be modeled using statistical distributions, such as Poisson or exponential, to represent interarrival times. Service Mechanism: The service mechanism defines how customers are served by servers in the system. It can be modeled using different distributions, such as exponential or Erlang, to represent service times. Queue Discipline: The queue discipline specifies the rules for customer waiting and service allocation. Examples include first-come-first-served, last-come-first-served, or priority-based disciplines. Number of Servers: The number of servers in the system can significantly impact its performance. Models can vary from single-server systems to complex multi-server setups. System Capacity: The system capacity determines the maximum number of customers that can be accommodated at a given time. It can be finite or infinite, depending on the specific scenario. Performance Measures: Simulation models can be designed to capture various performance

measures, such as average waiting time, queue lengths, server utilization, or system throughput.

4.3 Applications of Simulation Techniques

Simulation techniques find applications in various domains where queueing systems are prevalent. Some common applications include: Transportation Systems: Simulating traffic flows, tollbooth operations, or airport security checkpoints can help optimize queue lengths, waiting times, and resource allocation. Call Centers: Simulation models can aid in analyzing call arrival patterns, staffing levels, and agent performance to improve customer service and reduce waiting times. Manufacturing and Supply Chains: Simulating production lines, inventory systems, or order fulfillment processes can optimize resource allocation, minimize bottlenecks, and enhance overall efficiency. Healthcare Systems: Simulation techniques can assist in analyzing patient flows, hospital bed capacities, emergency department operations, or appointment scheduling to improve service quality and reduce waiting times. Computer Networks: Simulating network traffic, packet routing, or server loads helps evaluate network performance, identify congestion points, and optimize resource allocation. Retail and Service Operations: Queueing simulation can be utilized to analyze checkout lines, waiting times, and customer flow in retail stores, banks, or service centers, enabling process optimization and resource allocation improvements.

5 Conclusion

Queueing theory in computer science is an invaluable tool that provides the means to analyze the performance and efficiency of computer systems. It revolves around the study of waiting lines, or queues, through mathematical modeling. This study has important implications for a vast range of applications, including performance analysis, capacity planning, and resource allocation. Performance analysis, for instance, is critical in evaluating how a system would respond under various loads and usage conditions. Through queueing theory, we can predict how efficiently tasks will be processed and how long they will be queued, thereby enabling us to optimize the system performance. Another vital application lies in capacity planning. As organizations increasingly rely on complex computer systems to operate, it is critical to ensure that these systems have the capacity to handle expected workloads. Queueing theory can guide capacity planning by helping us understand the relationships between system demand, capacity, and performance. Resource allocation, an additional important application, is crucial for optimizing system performance by ensuring resources are efficiently distributed among competing tasks or processes. Through queueing theory, we can determine optimal strategies for allocating resources to minimize queue lengths and waiting times. As computer systems continue to evolve, growing in scale and complexity, queueing theory's relevance intensifies. The need to process tasks quickly and efficiently has never been more critical, given the data-rich era we live in. Thus, it is essential for computer

scientists and engineers to deepen their understanding of queueing theory and its real-world applications. Their expertise in this domain will be instrumental in designing, managing, and optimizing computer systems to meet the ever-increasing demands of the digital age.

References

1. Liu, J., & Zhao, X. (2019). Analysis of computer systems performance based on queueing theory. Journal of Computing and Information Science in Engineering, 19(3), 031006.
2. Bolch, G., Greiner, S., & de Meer, H. (2017). Queueing networks and Markov chains: modeling and performance evaluation with computer science applications. John Wiley & Sons.
3. Zhang, D., & Wang, L. (2018). The study of computer network performance based on queueing theory. Security and Communication Networks, 2018.
4. Kleinrock, L. (2010). Queueing systems, volume 1: theory. John Wiley & Sons.
5. Song, H., Shu, L., Liu, H., Li, H., & Liu, R. (2017). Queueing analysis of virtualized computer systems. International Journal of Communication Systems, 30(12), e3355.
6. Paul, R., & Viswanath, P. (2018). Quantitative System Performance: Computer System Analysis Using Queueing Models. Springer.
7. Ghahramani, M., & Schwan, K. (1997). Analyzing computer system performance with queueing network models. IEEE Transactions on Software Engineering, 23(12), 766-779.
8. Dou, C., Zhang, Y., & Zhang, X. (2017, August). A study on queuing theory application in computer network. In 2017 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS) (pp. 558-561). IEEE.
9. Chen, X., & He, F. (2016, June). Performance evaluation and optimization of computer systems using queueing theory and monte carlo simulation. In 2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech) (pp. 421-426). IEEE.
10. Zhang, L., Zhang, Y., & Li, J. (2020). A study of cloud computer systems performance based on queueing theory and neural network. Journal of Supercomputing, 1-23.