УДК 004.42: 004.65 DOI: 10.32626/2307-4507.2024-30.168-172

Halyna TKACHUK[1], Nadia STECENKO[2], Tetiana BONDARENKO[3], Rostislav MOTSYK[4]

[1,2,3]Pavlo Tychyna Uman State Pedagogical University
[4]Kamianets-Podilskyi Ivan Ohiienko National University

e-mail: [1]tkachuk.g.v@udpu.edu.ua, [2]stecenkonm@gmail.com, [3]tanyabond2006@gmail.com, [4]motsyk@kpnu.edu.ua;
ORCID: [1]0000-0002-6926-1589, [2]0000-0002-9802-6529, [3]0000-0001-9330-9661, [4]0000-0003-0947-3579

THE STUDY OF NETWORK RESOURCE REQUIREMENTS FOR DIFFERENT TYPES OF CLOUD APPLICATIONS

Abstract. This article explores how coding techniques can optimize the allocation and utilization of network resources in various cloud applications. It describes the challenges associated with diverse application requirements and presents effective coding strategies for managing bandwidth, reducing latency, and improving performance. The research examines different types of cloud applications, from data-intensive to real-time streaming, and highlights specific encoding techniques that are appropriate for each. By implementing these techniques, cloud service providers can achieve more efficient resource management, resulting in cost savings and improved service quality. Particular attention is paid to the specifications that should be taken into account to ensure high availability, data transfer speed, security and optimal resource utilization in cloud environments. The importance of dynamic network management to ensure the stable operation of cloud applications is emphasized, and the prospects for the development of networking technologies in cloud computing are considered.

Key words: network resource requirements, cloud applications, bandwidth optimization, latency management, data processing efficiency, real-time interaction, IOT devices optimization.

INTRODUCTION

The relevance of studying the network resource requirements for different types of cloud applications is driven by the rapid growth in data volume and the widespread adoption of cloud technologies across all areas of life. Today, cloud services are becoming critically important for various sectors of human life–business, education, entertainment, healthcare, and other fields where high availability, reliability, and network performance are essential. The increasing need for real-time data processing, continuous media streaming, fast access to large volumes of information, and support for the Internet of Things (IoT) creates significant demands on network infrastructure.

The issue of efficient use of cloud network resources has been the subject of research for many scientists, who have described the current state of cloud computing, the challenges faced by traditional approaches, and the application of effective methods and ways to optimize cloud systems in terms of their latency and energy efficiency [1-3].

With the increasing demands for Quality of Service (QoS), the study of network resource requirements is highly relevant, as it provides a better understanding of how different types of cloud applications impact the network environment, the resources needed for their stable operation, and the resource allocation strategies that should be applied to reduce costs [3]. Such studies also contribute to improved network management, flexibility, and scalability of cloud platforms, which are critical for maintaining high-quality service.

MAIN PART

Let’s identify the network resource requirements using examples such as streaming applications, big data processing applications, web applications, real-time interactive applications, backup and disaster recovery applications, and the Internet of Things (IoT).

Streaming Applications are applications that transmit media content (video, audio, or data) in real-time, allowing users to consume content immediately without full downloading. This approach is extremely popular in entertainment, communication, and education. Streaming applications require high bandwidth, minimal latency, and a stable connection to ensure smooth content delivery without buffering. They generate a prolonged, continuous traffic flow with minimal allowable delays. To achieve this, cloud providers need distributed servers across various geographic locations (CDNs – Content Delivery Networks) to deliver content to users with minimal latency. The network should provide low latency and high bandwidth, which can create peak loads, especially during high user activity (e.g., live sports events or premiere shows). Cloud systems must be able to quickly scale resources to meet demand and prevent server overload.

Thus, we see that critically important parameters for streaming applications are bandwidth (Bandwidth Calculation) and latency (Latency Calculation). Another important parameter is jitter (Jitter Calculation), which measures the variation in time between the arrival of consecutive data packets in a network stream. An ideal network would have equal time intervals between each packet's arrival, but in the real world, this is rarely the case due to factors such as network congestion, packet routing, and more.

Let's consider the formula for bandwidth calculation:

$$B = \frac{D}{T}, \quad (1)$$

where, B – bandwidth (Mbps); D – data volume (Mb); T – data transmission time (seconds). Bandwidth defines the maximum amount of data that can be transmitted through a network in a unit of time. The formula shows how quickly data can be transferred given the data volume and the time taken to transmit it.

To use the formula, it is necessary to know the data volume D to be transmitted and the time T it takes for this transmission. For example, if a 100 Mb file is transmitted in 10 seconds, we will obtain the following calculations:

$$B = \frac{100\,Mb}{10\,s} = 10\,Mb/s\,.$$

Of course, network bandwidth depends on the physical characteristics of the data transmission channel, the quality of equipment, network load, and other factors.

We can use this formula in a script to calculate bandwidth for analyzing network traffic, measuring data transmission efficiency, or optimizing data streaming. Here is an example of a simple Python script (*fig. 1*) that calculates bandwidth based on data volume and transmission time.

```
def calculate_bandwidth(data_size, time_seconds):
    if time_seconds <= 0:
        raise ValueError("The transmission time must be > 0.")
    bandwidth = data_size / time_seconds
    return bandwidth
data_size = float(input("Amount of data transferred in MB: "))
time_seconds = float(input("Transmission time in seconds: "))
try:
    bandwidth = calculate_bandwidth(data_size, time_seconds)
    print(f"Network bandwidth: {bandwidth:.2f} M6/c")
except ValueError as e:
    print(e)
```

Fig. 1. *Example of a simple Python script.*

In this example, the function *calculate_bandwidth()* takes the data size (in megabits) and the transmission time (in seconds). The formula is defined as the ratio of the data size to the transmission time: *bandwidth=data_size/time_seconds*. If the user enters zero or negative time, the function raises an error.

The proposed script example can be useful in scenarios for analyzing network performance, calculating bandwidth requirements, or for educational purposes to explain the principles of data transmission in networks.

Another important parameter for streaming applications is latency. The following formula is used to calculate it:

$$L = \frac{P}{R}, \tag{2}$$

where, L – latency, typically measured in milliseconds (ms); P – size of the transmitted data packet, in bits or bytes; R – data transmission rate in bits per second (bps). The formula calculates the time required to transmit a packet of a certain size through a channel with a defined transmission speed. Latency is an important parameter for evaluating network performance, as it determines how quickly data can travel from the sender to the receiver. High latency can negatively impact the quality of service, especially for real-time applications such as video conferencing or online gaming.

And the third important parameter for streaming applications is jitter. It is calculated using the following formula:

$$J = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(d_i - \bar{d}\right)^2}, \tag{3}$$

where J – jitter, which indicates the variation in the arrival time between packets; N – total number of measurements (received packets); d_i – delay for each individual packet (packet arrival delay); $\bar{d}$ – average delay for all packets.

The formula calculates the root mean square deviation between the delays for each individual packet (d_i) and the average delay ($\bar{d}$). The higher the jitter value, the greater the variations in packet arrival times, indicating instability in data transmission. This parameter is critically important for real-time applications, as the stability and predictability of packet delivery times are key for maintaining high-quality performance.

Big Data Processing Applications are programs and systems designed for collecting, storing, processing, and analyzing large data volumes that cannot be efficiently handled with traditional tools or databases. These applications are used to gain valuable insights from data, aiding in informed decision-making, understanding user behavior, predicting trends, and optimizing business processes. Big data applications generate bursty traffic when data is uploaded or processed in large volumes. They are less sensitive to latency but require significant computing resources, creating high storage demands and requiring efficient bandwidth management. Optimization is necessary for processing this data without overwhelming the network. Examples include real-time data analysis, machine learning, and large-scale data processing for scientific research (e.g., Apache Hadoop, Apache Spark, Google BigQuery, Amazon Redshift).

Thus, for processing big data, critically important parameters include processing time and data processing rate. Let's consider the formulas for calculating these parameters.

For processing time, the following formula is used:

$$T_{process} = \frac{D_{total}}{R_{process}}, \tag{4}$$

where T_{process} – processing time; D_{total} – total data volume to be processed; $R_{process}$ – data processing rate.

The parameter D_{total} indicates the total volume of data that needs to be processed. This can refer to the volume of data transmitted over a network, processed by a server, or analyzed by an algorithm. The parameter $R_{process}$ specifies how many units of data can be processed per unit of time (e.g., megabytes per second). The data processing rate depends on the processor's capacity, available resources, and system characteristics. If you have a large data volume and a low processing rate, the processing time will be long. Conversely, increasing the processing rate will reduce the processing time.

To determine the data processing rate, the following formula is used:

$$R_{process} = \frac{D}{T_{process}}, \tag{5}$$

where $R_{process}$ – data processing rate: D – data volume to be processed; $T_{process}$ – time required for data processing.

The formula for calculating processing time and the formula for determining the processing rate are closely related, as they are inverses of each other and describe the interdependence between three key parameters: data volume, processing rate, and processing time. These formulas allow for evaluating system performance, balancing workloads, and optimizing data processing by adjusting one of the parameters.

Web Applications are applications that run on servers and are accessible to users via a browser without the need for installation on a computer or mobile device. Web applications interact with the server over the Internet, processing user requests and responding with web pages or interactive interfaces. They typically generate variable traffic depending on the time of day and user activity. These applications require short request and response times, with high request volumes during peak hours. Web applications need effective management of peak loads and rapid server response. Bandwidth should be sufficient to handle requests from a large number of users simultaneously. Examples of web applications include e-commerce sites, social media platforms, email services, and more.

For web applications, critically important parameters include response time and the number of simultaneous connections. Let's consider the formulas for their calculation.

Response time is calculated using the following formula:

$$R_{response} = T_{server} + T_{network}, \quad (6)$$

where $T_{response}$ – the total response time that a user waits when making a request to a web application; T_{server} – the server processing time, meaning the time required for the server to execute the request, generate a response, and, if necessary, perform complex computations or access a database; $T_{network}$ – the data transmission time over the network, which refers to the time it takes to send a request from the client to the server and deliver the response back to the client.

The server processing time (T_{server}) includes executing business logic, database queries, running scripts, data processing, and forming the response. High latency at the server level can be caused by high load, inefficient code, or complex database operations.

The network data transmission time ($T_{network}$) encompasses delays related to the network, such as routing delays, jitter, bandwidth limitations, and other factors affecting the speed of data transfer between the client and server.

The total response time ($T_{response}$) is a crucial metric for measuring the performance of a web application. Reducing server processing time and network transmission time can improve overall speed and user experience. Optimization may include caching, reducing the volume of transmitted data, optimizing database queries, reducing network delays, and other methods to enhance the web application's performance.

The next important metric for web applications is the number of simultaneous connections, which is calculated using the following formula:

$$U_{max} = \frac{C}{T_{avg}}, \quad (7)$$

where $U_{\max}$ – the maximum number of simultaneous connections; C – the capacity or throughput of the system (for example, the number of requests that can be processed within a certain time); T_{avg} – the average processing time for a single request or connection. The formula indicates that the smaller the average processing time for each request, the more connections the system can handle simultaneously. This is an important metric for evaluating the performance of servers, networks, or other data processing systems.

Real-Time Interactive Applications are programs that require instant interaction between the user and the system, ensuring data transmission and processing in real time. They are highly sensitive to delays, as data must be transmitted in real-time. These applications demand a stable, low-latency network to support seamless user interaction. Quality connectivity is essential, especially for applications with high audio or video quality requirements. Network load may fluctuate based on the number of users. Examples include video conferencing (e.g., Zoom, Microsoft Teams), online gaming, and chat applications.

For real-time interactive applications, latency is important, just as it is for streaming applications, as well as the packet loss ratio.

The packet loss ratio defines the percentage of lost packets, which is critical for interactivity and is calculated using the following formula:

$$PLR = \frac{N_{lost}}{N_{total}} \times 100\%, \quad (8)$$

where PLR – the packet loss ratio percentage; N_{lost} – the number of lost packets; N_{total} – the total number of sent or received packets.

The formula shows what portion of packets was lost as a percentage of the total number of transmitted packets. Packets that do not reach their destination may be lost due to various reasons, including network congestion, transmission errors, or hardware failures. This is an important metric for measuring data transmission quality in a network, where a high packet loss percentage indicates potential network or communication issues.

Backup and Disaster Recovery Applications are programs designed for creating backups of important data and ensuring its recovery in case of loss, damage, or disruption to main systems. These applications help organizations and individuals protect their data from unexpected failures, system outages, cyber-attacks, or other risks. They perform large data transfers during backups or recovery. Traffic may be generated periodically (e.g., at night or on weekends). These applications require significant bandwidth for data transfer, which can burden the network. Optimizing the transfer process is essential to avoid impacting the performance of other applications. Examples include cloud backups (e.g., Google Drive, Dropbox, Microsoft Azure Backup).

For backup and data recovery applications, critically important parameters include data transfer time and bandwidth utilization ratio.

The formula for calculating data transfer time is:

$$T_{transfer} = \frac{D_{backup}}{B}, \quad (9)$$

where $T_{transfer}$ – the data transfer time (in units such as seconds, minutes, or hours) needed to transfer a certain volume of data; D_{backup} – the amount of data to be transferred (e.g., in bytes, kilobytes, megabytes, or gigabytes). This could refer to the volume of data for backup or any other data quantity to be transferred; B – data transfer speed (bandwidth of the communication channel) in relevant units (e.g., bytes/second, kilobytes/second, etc.).

The formula calculates the time required to transfer a specific data volume by dividing the data size by the transfer speed. It is widely used to estimate the time needed for performing backups, copying files, network file transfers, and similar tasks.

To measure the bandwidth utilization ratio, the following formula is used:

$$U = \frac{B_{used}}{B_{total}} \times 100\%, \quad (10)$$

where U – utilization ratio, expressed as a percentage; B_{used} – the amount of resource or bandwidth used; B_{total} – the total available amount of the resource or bandwidth.

The formula is often used to measure resource utilization in various systems, such as network bandwidth, memory capacity, disk space, or other resources. It shows the portion of the total resource volume that has already been used.

Internet of Things (IoT) is a concept of connecting physical devices to the Internet for data exchange, allowing them to interact with each other and users. These devices may include sensors, household appliances, vehicles, medical devices, security systems, and other "smart" objects that collect, transmit, and analyze information in real-time. Through IoT, devices can operate autonomously, perform tasks, and provide users with information, enhancing convenience, efficiency, and security. IoT creates a constant low stream of data from numerous devices or a large volume when an event is triggered (e.g., an alarm signal). They require low latency and continuous data transmission. Network scalability is critical due to the high number of connections, and reliable connectivity and efficient transmission for all connected devices are necessary. Examples include smart homes, monitoring systems, and real-time sensors.

For IoT devices, critically important parameters include data transmission time from the device and average load.

Data transmission time from the device is calculated using the following formula:

$$T_{transmit} = \frac{D}{B}, \quad (11)$$

where $T_{transmit}$ – data transmission time (e.g., in seconds); D – the volume of data to be transmitted (e.g., in bytes, kilobytes, megabytes, etc.); B – the bandwidth of the transmission channel or data transmission speed (e.g., bytes/second).

The formula reflects the time needed to transmit a specific volume of data at a given channel speed. It is important for evaluating data transmission efficiency and performance in networks, including IoT devices. IoT devices often have limited resources, such as bandwidth, memory, and power consumption. This formula helps to estimate the time required for data transmission, influencing the choice of optimal solutions for energy consumption and performance.

Moreover, for many IoT devices, it is crucial for data to be transmitted quickly and efficiently, especially in real-time systems (e.g., monitoring or automation systems). This formula enables the prediction of data transmission time and the appropriate configuration of the network.

To calculate the average load, the following formula is used:

$$L_{avg} = \frac{\sum_{i=1}^{N} L_i}{N}, \quad (12)$$

where L_{avg} – the average load or resource utilization intensity over a selected period of time. This can refer to, for example, average CPU load, average network load, or average energy consumption; $\sum_{i=1}^{N} L_i$ – is the sum of all measured values of load or resource utilization intensity over a certain period of time. The values L_i are specific measurements of load at different moments in time, where i ranges from 1 to N; N – the number of measurements or observations over the selected period of time. The more measurements are taken, the more accurate the average value.

Thus, it is clear that this parameter is critical, as it helps evaluate how heavily loaded an IoT device or system is during data transmission, processing, and reception.

Calculating the average load can show how effectively an IoT device operates. If the load is high for an extended period, it may indicate an overload, which could lead to delays in data transmission, reduced processing speed, or even device failure.

Calculating the average load is important for a device's energy consumption, as many IoT devices operate on limited power sources (e.g., batteries), so the average load directly affects the operating time from a power source. The greater the load, the more energy is consumed. Additionally, in cases where an IoT device transmits data, the average load can affect data transmission quality and reliability. High average network load may cause delays, data loss, or reduced system performance.

CONCLUSION

The study of network resource requirements for different types of cloud applications is important due to the rapid growth of data volumes and the widespread adoption of cloud technologies in many areas of life. Different applications, such as streaming media, big data processing, web applications, real-time interactive programs, backup, and IoT, have specific network resource requirements that impact their performance. Optimizing network infrastructure to support these applications ensures quality service and high reliability. This promotes efficient resource usage, stable network performance, and improved user experience. Calculating parameters such as bandwidth, latency, jitter, data processing time, and average load helps better understand and optimize network processes. The importance of optimization increases due to the limited resources of IoT devices, such as energy and bandwidth. Improving data transmission and minimizing delays is critical for real-time applications. Optimized solutions enable load balancing and enhance network performance.

References:

1. Jeyaraj R., Balasubramaniam A., MA A.K., Guizani N., Paul A. Resource management in cloud and cloud-influenced technologies for internet of things applications. *ACM Computing Surveys*. 2023. Vol. 55. No. 12. P. 1-37.
2. Soyata T., Ba H., Heinzelman W., Kwon M., Shi J. Accelerating mobile-cloud computing: A survey. In *Cloud Technology: Concepts, Methodologies, Tools, and Applications, IGI Global.* 2015. P. 1933-1955.
3. Zhang Y., Liu B., Gong Y., Huang J., Xu J., Wan W. Application of machine learning optimization in cloud computing resource scheduling and management. In *Proceedings of the 5th International Conference on Computer Information and Big Data Applications*. Apr. 2024. P. 171-175.

Галина ТКАЧУК[1], Надія СТЕЦЕНКО[2], Тетяна БОНДАРЕНКО[3], Ростислав МОЦИК[4]

[1,2,3]Уманський державний педагогічний університет імені Павла Тичини

[4]Кам'янець-Подільський національний університет імені Івана Огієнка

ДОСЛІДЖЕННЯ ВИМОГ ДО МЕРЕЖЕВИХ РЕСУРСІВ ДЛЯ РІЗНИХ ТИПІВ ХМАРНИХ ПРОГРАМ

Анотація. У статті досліджується, як методи кодування можуть оптимізувати розподіл і використання мережевих ресурсів у різних хмарних програмах.

У ньому описано проблеми, пов'язані з різноманітними вимогами додатків, і представлено ефективні стратегії кодування для керування пропускною здатністю, зменшення затримки та підвищення продуктивності. Дослідження розглядає різні типи хмарних додатків, від інтенсивної обробки даних до потокової передачі в реальному часі, і висвітлює спеціальні методи кодування, які підходять для кожного. Впроваджуючи ці методи, постачальники хмарних послуг можуть досягти більш ефективного управління ресурсами, що призведе до економії коштів і підвищення якості обслуговування. Окрему увагу приділено специфікаціям, які повинні враховуватися для забезпечення високої доступності, швидкості передачі даних, безпеки та оптимального використання ресурсів у хмарних середовищах. Підкреслено важливість динамічного управління мережею для забезпечення стабільної роботи хмарних додатків, а також розглянуто перспективи розвитку технологій мережевої взаємодії у хмарних обчисленнях.

Ключові слова: вимоги до мережевих ресурсів, хмарні програми, оптимізація пропускної здатності, керування затримкою, ефективність обробки даних, взаємодія в реальному часі, оптимізація пристроїв IoT.

Отримано: 20.09.2024