Optimization of File Distribution in Cloud-Based Data Storages

Ivan Muzyka[1], Dennis Kuznetsov[1], Yurii Kumchenko[1] and Anton Senko[1]

[1] *Kryvyi Rih National University, 11 Vitaliy Matusevych Street, Kryvyi Rih, 50027, Ukraine*

Abstract

The article presents one of the approaches to optimizing the distribution of files across servers. Distributed information system operating in a global computer network with a mixed topology is considered. The authors prove that there is a certain correlation between traffic speed and network delay. This made it possible to use ping as a parameter that indicates the overall speed of the information system. The article proposes a mathematical model that uses two optimization criteria: the total cost of server maintenance and the weighted average ping. The cost criterion takes into account the total load on the server, the amount of additional storage, and the ability of the software to be parallelized. The authors showed the possibility of using genetic algorithms to find a suboptimal distribution. This approach works acceptably for small amounts of files and servers. However, for large volumes of data, the proposed heuristic algorithm gives an adequate result in a reasonable number of steps. The proposed model can be used for load balancing on servers that store huge amounts of audio and video files or provide online streaming services.

Keywords

File distribution, cloud storage, network latency, heuristic optimization, load balancing.

1. Introduction

Information technology has undergone huge changes over the past 5 years. The global challenges posed by pandemic COVID-19 have led to a significant increase of data in computer networks. Remote work and studying have led to increasing in video and audio content. Platforms such as Google Meet, Zoom, Microsoft Teams, YouTube as well as Viber, Telegram, WhatsApp messengers have become everyday tools not only for IT professionals but also for ordinary people. Cisco's Visual Networking Index (VNI) report demonstrated a significant increase in global IP traffic during 2018-2023, including video, online education, and entertainment [1]. Zoom has accumulated more than 3.3 trillion annual meeting minutes. 50 billion minutes of webinars are hosted on Zoom every year. Nowadays, the rapid development of cloud storage is driving up costs. Globalization and increase in traffic volumes require upgrading of network equipment. Every year, Internet Service Providers (ISPs) try to grow their bandwidth capacity. Despite reducing the cost of high-speed SSD drives to $30-60/TB, the problems of cloud storage optimization remain relevant [2]. Software developers and database architects should take into account the problems that arise when there is no balancing between individual servers. Microservice architecture helps to divide a complex application into separate components. Thus, finding the optimal location of files in distributed information systems can significantly affect the cost of their support.

2. Related work

Optimizing distributed databases and content in cloud-based storages involves various strategies aimed at improving performance, scalability, reliability, and cost-effectiveness. Professor at University of Waterloo M. T. Özsu defined a distributed database as a collection of multiple, logically interrelated databases located at the nodes of distributed system [3]. A distributed

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

musicvano@knu.edu.ua (I. Muzyka); kuznetsov.dennis.1706@knu.edu.ua (D. Kuznetsov); kumchenko@knu.edu.ua (Y. Kumchenko); antonsenko@knu.edu.ua (A. Senko)

0000-0002-9202-2973 (I. Muzyka); 0000-0002-2021-5207 (D. Kuznetsov); 0000-0001-9940-4854 (Y. Kumchenko); 0000-0002-4104-8372 (A. Senko)

database management system (DBMS) is defined as the software system that permits the management of the distributed database and makes the distribution transparent to the users. It is important to note that a distributed DBMS is not just a collection of files that can be stored individually on servers. However, today, as a rule, unique identifiers of resources such as videos, sound files, image files, etc. are stored in the database. Unique keys are used to identify files. For example, Universally Unique Identifiers (UUID) are best suited for generating unique keys in distributed systems. At the same time, UUIDs provide an extremely low probability of collisions.

Often, to optimize data storage in distributed systems, researchers can use the following approaches [4] such as sharding and partitioning, load balancing, dynamic resource allocation etc. Dividing the dataset into smaller partitions (shards) distributed across nodes enhances parallel processing and reduces contention, leading to improved performance.

Some authors in their scientific works [5, 6] justify the feasibility of using heuristic or genetic algorithms to query optimization, etc. Professor G. G. Tsehelik of Lviv University described in detail a series of mathematical models that can be used to optimize file distribution in information systems [7]. The main optimization criteria include the following:

- minimizing the time for processing queries in a distributed database;
- minimizing the traffic transmitted in the computer network and generated by the corresponding database queries;
- minimization of the cost of the distributed system operation, taking into account the hardware requirements for the nodes.

Other criteria may include the minimizing of electricity cost [8]. Overall, this research offers a valuable approach for Internet data centers (IDC) operators to optimize their energy costs while maintaining QoS in dynamic electricity markets.

There are a lot of modern researches concerning problems of effective data placement in cloud storages [9-11]. The article offers a comprehensive overview of data deduplication techniques and their applications in cloud storage, focusing on challenges and future directions. Also, authors tackle the problem of choosing the optimal location for storing data across geographically distributed cloud storage systems, considering cost and access latency. It's worth to note that there are methods based on using tiered cloud storage with dynamic data migration based on access patterns to optimize costs. However, insufficient attention has been paid to algorithms for redistributing content in a distributed system depending on changes in the intensity of user requests.

Thus, the scientific problems of researching distributed systems are quite relevant today and require further research, in particular with methods of parallelizing computing on the basis of high-performance computers.

3. Problem statement

Suppose we have a distributed video content storage system like YouTube, Facebook or TikTok. To ensure the operation of such an information system, many servers are needed around the world. Since data centers are located on different continents and serve requests from different countries, it is necessary to propose a model for optimal file distribution. Some videos may have different popularity in different regions. The number of views over a certain period of time will determine the intensity of requests.

However, the question arises whether placing files on a particular server can really affect the user experience. Obviously, a single server cannot serve hundreds of thousands of users streaming millions of videos. Some estimates suggest Google, which owns YouTube, might have over 1 million servers in its data centers worldwide [12]. However, this figure likely includes servers for all Google services, not just YouTube. Experts agree YouTube's infrastructure is highly scalable and continuously adapts to handle the ever-growing demand for video content. It likely utilizes a distributed network of data centers spread across various locations globally.

Let's look at the example of the cloud provider DigitalOcean to see how the physical distance from the user to the data center affects the network latency. DigitalOcean has datacenters across 9 regions [13]. Developers strongly suggest to pick a data center location geographically close to the end users who most frequently access your applications [14]. Figure 1 shows the locations of data centers around the world. The ping command demonstrates a clear correlation. The closer

the data center is located, the lower the delay. It is assumed that the communication line is of high quality.

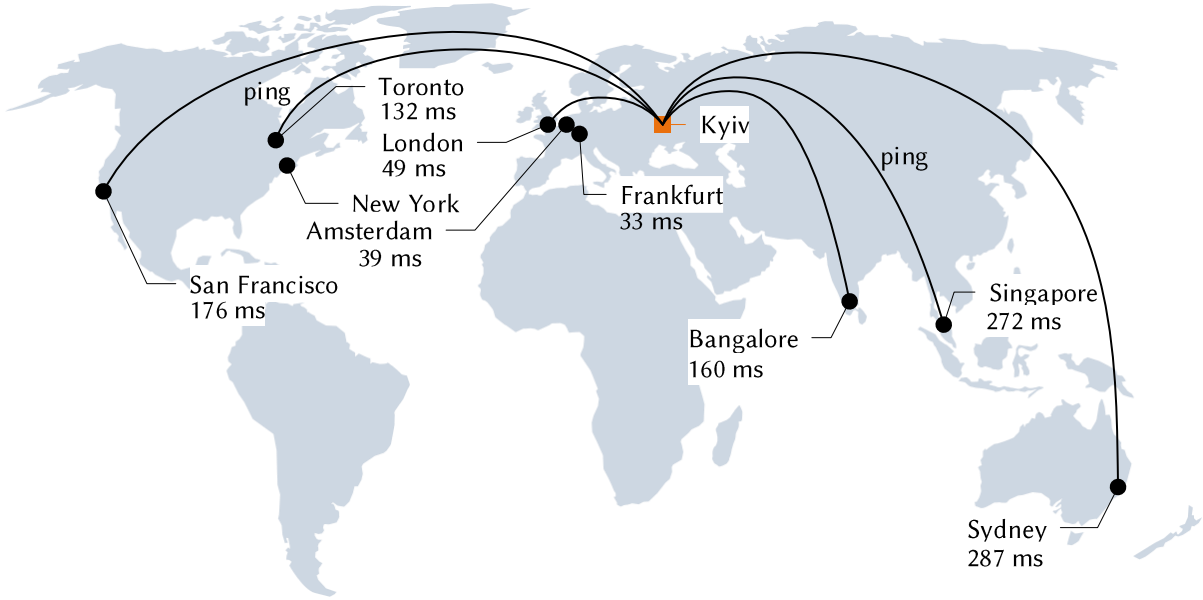

Figure 1: DigitalOcean data centers with network ping latency (tested from Ukraine)

Download speed and ping latency are related but not directly dependent on each other. They measure different aspects of network performance. Download speed refers to the rate at which data is transferred from the internet to your device. Download speed depends on various factors such as your internet service plan, network congestion, and the capabilities of your internet service provider. Ping, or latency, is the time it takes for a data packet to travel from your device to a server and back. Lower ping values indicate faster response times. Ping is influenced by the physical distance between your device and the server, as well as the efficiency of the network infrastructure. Let's try to consider this dependence in more detail.

According to the information collected by independent service Meter.net [15] there is some correlation between network latency and download speed. We used different data servers of Vultr cloud provider for testing (Table 1).

Table 1
Testing of data servers (Vultr.com)

Data center location	Country	Ping, ms	Download, Mb/s	Upload, Mb/s
Chicago	USA	82.44	29.15	14.61
Delhi	India	29.58	66.82	65.29
Frankfurt	Germany	20.83	58.99	74.93
Hawaii	USA	223	4.85	0.3
Johannesburg	South Africa	72	17	12.6
Melbourne	Australia	50.22	30.35	9.69
Mexico	Mexico	62.25	21.31	10.2
Osaka	Japan	24.4	103.3	252.6
Paris	France	16.75	103.8	154.5
Seoul	South Korea	16.3	175.1	417.9
Sydney	Australia	35.63	22.93	21.95

The graph in Figure 2 shows that as the ping delay increases, the data download speed decreases.

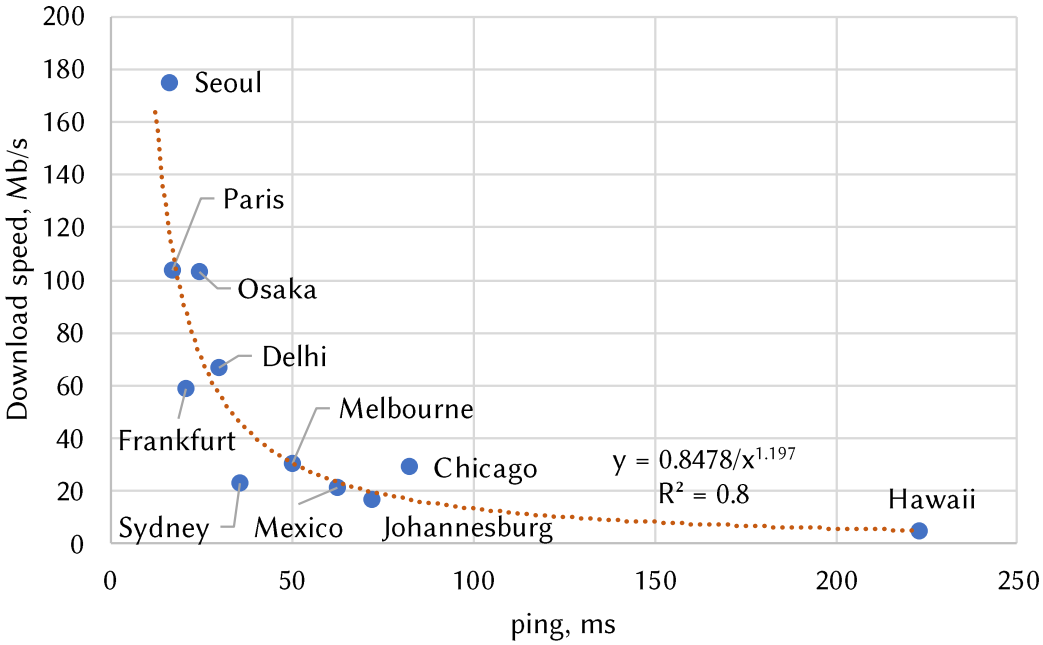

Figure 2: Dependency of download speed on ping network latency

Dependency of average download speed on ping latency can be described such empirical equation

$$v = \frac{0.8478}{\tau^{1.197}}, \quad (1)$$

where v – download speed, Mb/s, τ – ping latency, s. The coefficient of determination is R^2=0.8.

Thus, when creating a mathematical model to find the optimal file distribution, it is reasonable to use network latency as one of the parameters that directly affects the quality and performance of the system.

4. Proposed method

Let's have a look at a typical structure of a segment of a Wide Area Network (WAN), as shown in Figure 3. WANs support various applications such as data sharing, cloud access, file transfers, video conferencing, and more. Personal computers (PC1, PC2, …, PC10, PC11, …, PC30, PC31 etc.) are connected to the global network through switches (Sw1, Sw2, Sw4) and routers (R1, R4) of Internet service providers. In the vast and interconnected world of the global network, routers play the critical role of traffic directors, ensuring data reaches its intended destination efficiently and seamlessly. The servers (S1, S2, S3) are located in data centers that provide cloud services to users. Large facilities hosting major cloud providers or enterprise systems can have hundreds of thousands of servers, while smaller ones might have just a few hundred.

Many cloud providers (AWS, Google, Microsoft Azure) use different types of load balancers (LB1): hardware, software and cloud load balancers. Hardware load balancers (HLBs) are dedicated physical appliances designed to efficiently distribute traffic across multiple servers. Their robust hardware and specialized software offer high performance, security, and scalability, making them suitable for mission-critical applications and high-traffic websites. But load balancers make decisions based on CPU, RAM or network usage and don't guarantee the best user experience for sophisticated scenarios.

Databases (Db1, Db2, Db3) located on servers (S1, S2, S2) form a single distributed storage for the information system. Databases must support replication, transactions, multi-threaded operation, and handle high loads. Millions of files (F1, F2, …, F20, …, F120 etc.) can be stored in cloud storage on servers (S1, S2, S3). Users make requests to these files with different intensity. In the diagram, files from different servers are marked with different colors for better clarity.

It should be noted that the global computer network is not permanent and its characteristics are constantly changing throughout the day. Routing protocols use various metrics to ensure the speed and quality of information transmission. This means that network latency can vary widely over the period of a month. The ping value sometimes increases by 2-4 times when network problems occur.

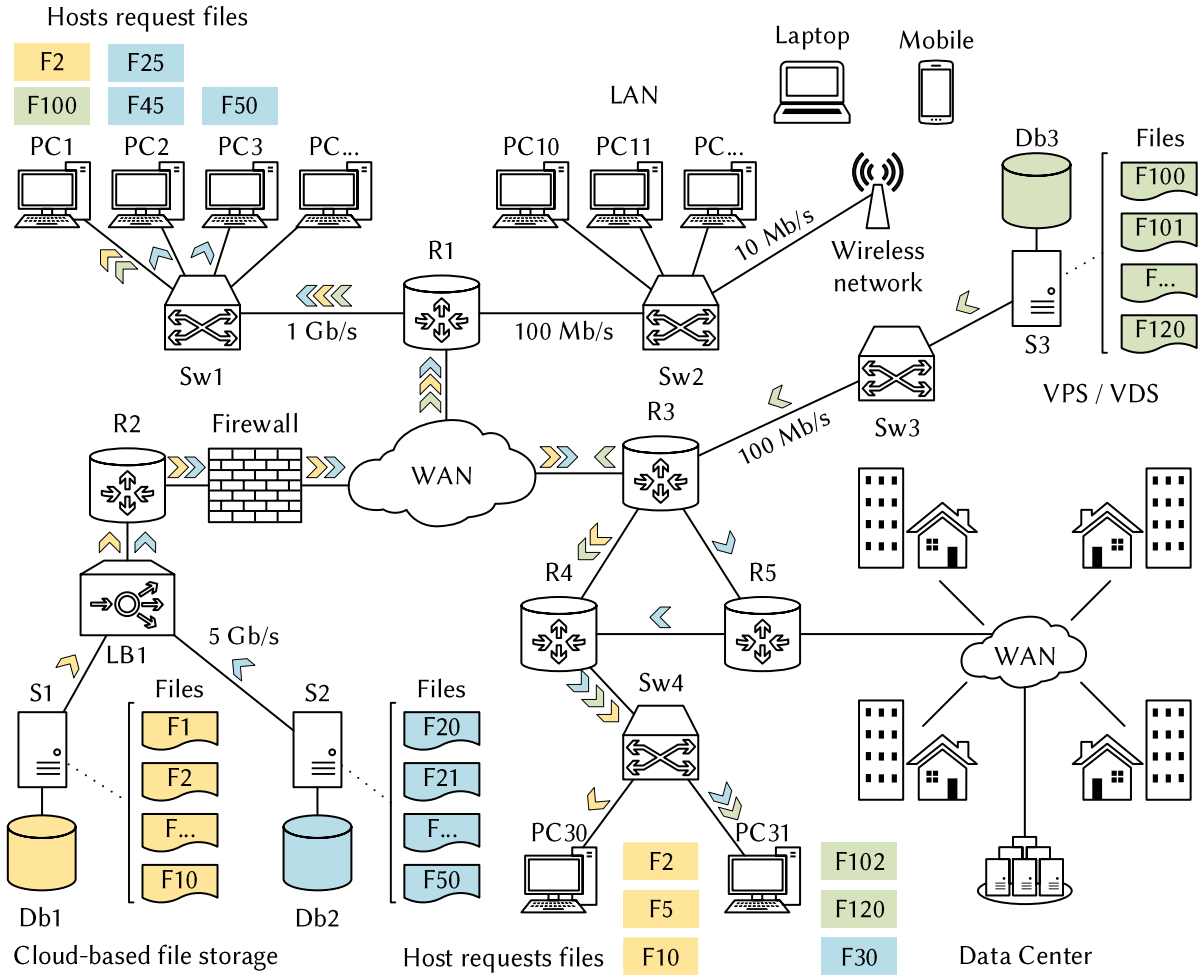

Figure 3: Typical structure of computer network with distributed file storage

Let's introduce the following notation:
n – the number of available working servers;
m – the total number of files (video, audio, images) to be stored by the distributed system;
p – the total number of users (personal computers / hosts);
S_i – the server with index i, $i = 1, 2, \ldots, n$;
V_i – the size in GB of the server's data storage S_i, which is required to store files;
F_j – the file with index j, $j = 1, 2, \ldots, m$;
L_j – size in GB of file with index j;
H_k – the host with index k, $k = 1, 2, \ldots, p$;
τ_{ki} –network latency (ping) from the host H_k to the server S_i;
λ_{kj} – intensity of requests per unit of time from the host H_k to the file F_j;
x_{ji} – Boolean value that indicates that the file F_j is stored on the server S_i;

$$x_{ji} = \begin{cases} 1, \text{if file } F_j \text{ is stored on the server } S_i \\ 0, \text{otherwise} \end{cases}, \quad (2)$$

Q_i – cost of server maintenance with index i.

Obviously, the optimal file distribution is the one that minimizes the cost of all working servers and maximizes the speed of request processing

$$Q = \sum_{i=1}^{n} Q_i \rightarrow min, \quad (3)$$

However, determining the cost of running one server is not as easy as it might seem at first glance. Most cloud providers build separate web applications that operate as a specialized calculator for estimating the cost depending on the hardware parameters of the server. Based on the information provided by the official website of Vultr company [16], we can conclude that the cost of a virtual server depends linearly on the number of processors (Figure 4). This approach allows us to conveniently scale the server performance when the number of requests changes.

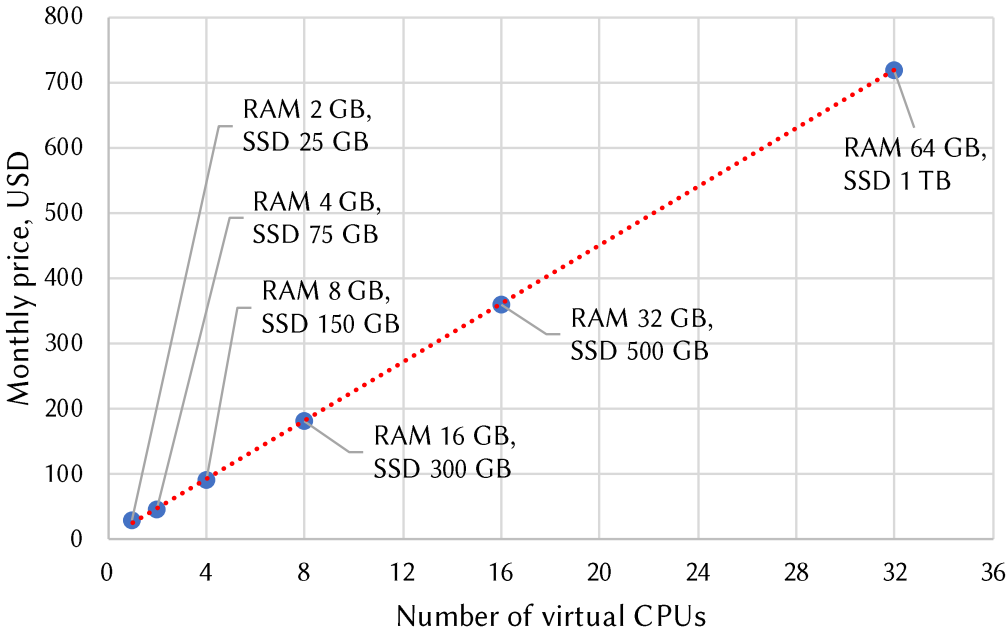

Figure 4: Dependency of VPS monthly cost on the server configuration

According to Amdahl's law, the speedup factor of the algorithm *s*, when the number of processors N_{CPU} for parallel query processing increases, is limited by the percentage of the program code α that is executed only in the sequential mode [17]

$$s = \frac{1}{\alpha + \frac{1-\alpha}{N_{CPU}}}. \quad (4)$$

Increasing the number of virtual processors in a server is more efficient the higher percentage of code that can be fully parallelized (Figure 5).

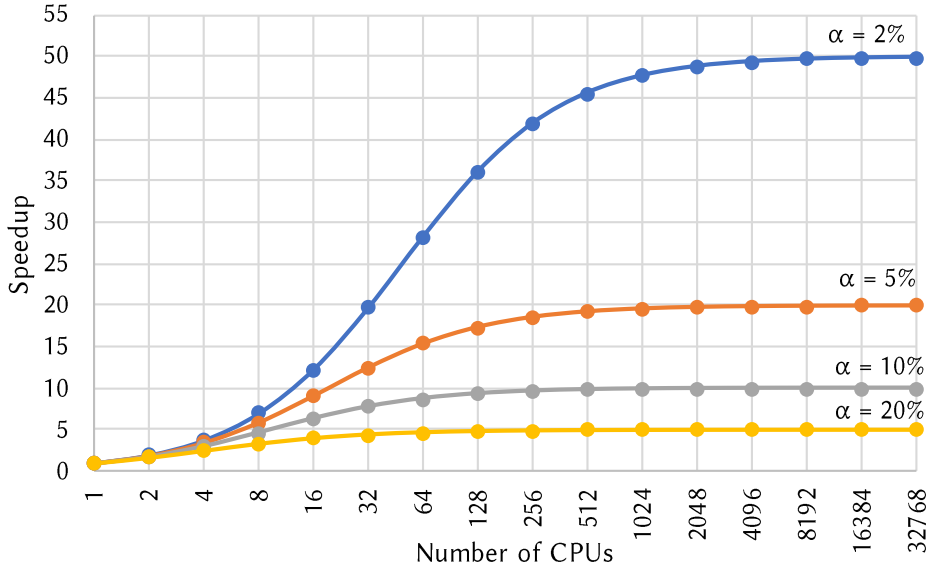

Figure 5: Dependency of program speedup on the number of virtual processors (cores)

If we assume that the speedup from using a more powerful server is the ratio of a certain desired request intensity λ_i to the base one λ_0, then expression (4) can be written as follows

$$\frac{\lambda_i}{\lambda_0} \cong \frac{1}{\alpha + \frac{1-\alpha}{N_{CPUi}}} \Rightarrow N_{CPUi} \cong \frac{1-\alpha}{\frac{\lambda_0}{\lambda_i} - \alpha}. \quad (5)$$

The running cost of a server consists of many components, but for simplicity's sake, it is advisable to consider the most important components: the actual cost of a cloud server with a certain performance and additional data storage that allows you to store a large amount of files.

In addition to the standard server that processes the requests, additional storage is also required, which typically costs about \$100/TB. Therefore, it is possible to estimate the cost using the formula

$$Q_i \cong q_{CPU} \cdot N_{CPUi} + q_{SSD} \cdot V_i, \quad (6)$$

where q_{CPU} = \$22/core – a coefficient that depends on cloud pricing and may be defined from a plot (Figure 4); q_{SSD} = \$0.1/GB – typically cost of additional SSD blocks.

The total intensity of requests from all hosts to a particular server *Si* can be calculated as following

$$\lambda_i = \sum_{k=1}^{p} \sum_{j=1}^{m} \lambda_{kj} x_{ji}. \quad (7)$$

With a given distribution of files x_{ji} and their sizes L_j, multiplying the vector L by the matrix x, we can get the total volume of certain server storage

$$V_i = \sum_{j=1}^{m} L_j x_{ji}. \quad (8)$$

Thus, the optimization criterion in its simplest form may look like this

$$Q \cong \sum_{i=1}^{n} \left(q_{CPU} \frac{(1-\alpha) \sum_{k=1}^{p} \sum_{j=1}^{m} \lambda_{kj} x_{ji}}{\lambda_0 - \alpha \sum_{k=1}^{p} \sum_{j=1}^{m} \lambda_{kj} x_{ji}} + q_{ssd} \sum_{j=1}^{m} L_j x_{ji} \right) \rightarrow min, \quad (9)$$

The formula (9) shows that if the total intensity of requests to the server increases, an error and service denial may occur

$$\sum_{k=1}^{p} \sum_{j=1}^{m} \lambda_{kj} x_{ji} \geq \frac{1}{\alpha} \lambda_0. \quad (10)$$

Since each file F_j should be stored on one of the working servers only S_i, therefore

$$\sum_{i=1}^{n} x_{ji} = 1, j = 1..m. \quad (11)$$

Consequently, the problem of optimal file distribution on servers in a computer network with a mixed topology is to determine the following variables x_{ji}, where

$$x_{ji} = \{0 \cup 1\}, j = 1..m, \mathrm{i} = 1..n. \quad (12)$$

However, the research conducted by the authors shows that it is not enough to have only one criterion for the operation of an information system that serves millions of users. It is also necessary to take into account the quality of service (QoS), speed of service, reliability indicators, etc.

Understanding QoS is crucial for designing and managing robust and efficient distributed information systems. By considering various aspects like performance, reliability, security, and dynamic environments, developers and administrators can implement effective QoS mechanisms and ensure optimal service delivery for users and applications. As Equation 1 shows, there is a certain correlation between the geographical distance from the client to the server, the speed of data transmission on the global network, and network latency. Therefore, the ping value can be used to estimate the data transfer rate. In the user's opinion the lower value of the average network delay, the higher quality of the information system. For example, when we talk about web browsing higher ping can lead to slower page loading times, causing delays and frustration. Even small increases in ping can be noticeable, especially on websites with a lot of content or interactive elements. High ping during the watching media can cause buffering and stuttering in

video or audio streaming, ruining the smooth playback experience. This is especially frustrating for high-resolution content or live streams.

Based on the above, it is necessary to introduce an additional parameter that will determine the weighted average delay in the system. This indicator should be minimal. Obviously, this parameter has a greater weight for large files and for servers with high request rates

$$T = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{p}\tau_{ki}\lambda_{kj}L_j x_{ji}}{\sum_{j=1}^{m}\sum_{k=1}^{p}\lambda_{kj}L_j} \to min. \quad (13)$$

The formulated challenge is similar to the linear programming problem described in paper [7]. However, it is still difficult to offer a clear algorithm for finding the optimal solution. Criterion (9) becomes minimal when the files are distributed evenly across all available servers, as this results in the lowest maintenance cost. Moving all the files to one super server is technically not possible, because there are channel bandwidth and memory capacity constraints. Besides, this solution is less reliable in terms of data transfer issues. Criterion (13) can be minimized by applying a simple principle: large files with a high intensity of requests should be placed on the server with minimal network latency (ping).

5. Modeling and results

A C# program was developed to simulate the distribution process. Using of .NET 8.0 and parallel LINQ technology made it possible to implement a stochastic method of searching for optimal values according to criteria (9) and (13). Authors have used genetic algorithm [18]. The initial distribution was random. The simulation was performed on a computer with AMD Ryzen processor (6 cores) and 8 GB of RAM.

Figure 6 shows the results of stochastic simulation. 20 files were distributed across 3 servers. The file sizes were in the range of 1.5-10 GB, and the ping from hosts to servers was in the range of 25-220 ms. The intensity of requests to the files did not exceed 40 requests per second. Each point on the graph represents a certain distribution of files and is described by two parameters: the total cost of maintaining all servers and the average network latency. Only 1000 points are shown on the graph for simplicity of presentation. However, if parallelization is used, the above computing system is capable of processing up to 1 million combinations per second. Given that the total number of all possible file arrangements on the servers is $n^m = 3^{20} \approx 3.48*10^9$, this problem can be solved even by a full brute-force search and will take about 1 hour.

As shown in the graph, there are a large number of suboptimal file allocations that have approximately the same cost (within 5-10%), but the average latency of the system can vary within a significant range of 105-145 ms (almost 40%). This fact proves the need to apply the criterion (13) when making a decision.

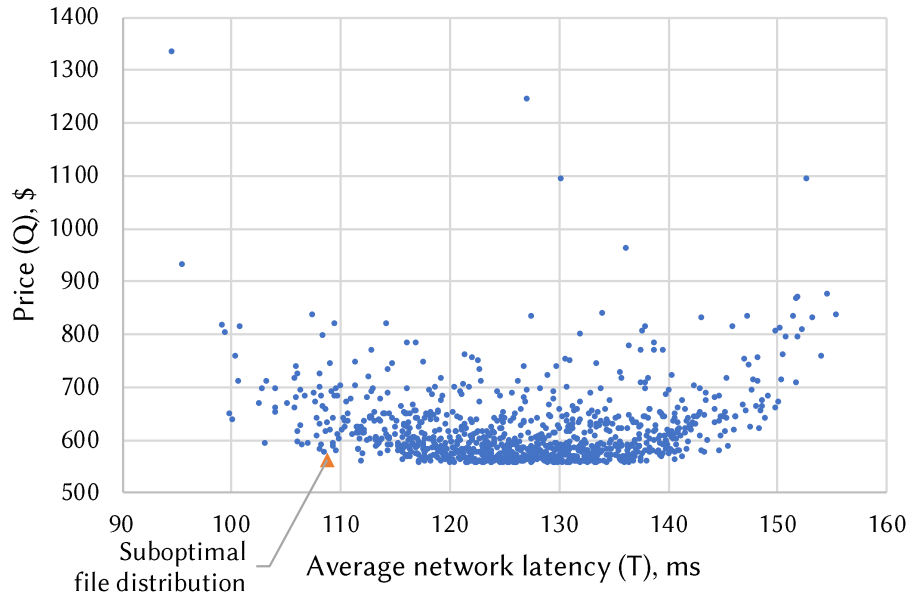

Figure 6: The set of random file distributions (1000 attempts)

It should be noted that a genetic algorithm can effectively find a suboptimal solution if it is applied in two stages. First, the minimum cost is sought, and then the algorithm starts looking for the minimum latency in the network. The cost can vary within 10% in this case. However, if the number of servers is increased to 10 and the number of files exceeds 10-20 thousand, the genetic algorithm may have poor convergence. The number of all possible combinations is too large to process in a reasonable time even 1% of all variations. Genetic algorithms are stochastic, meaning they rely on randomness. While they often find good solutions, there is no guarantee they will reach the absolute best one. They require evaluating many potential solutions across multiple generations, making them computationally expensive for complex problems. This can be a significant issue if resources are limited or real-time solutions are necessary. Therefore, it is worth considering an alternative approach to optimization.

At the initial stage, all files are placed on those servers that have the lowest ping. Let the weighted average network latency when placing a file F_j on the server S_i be determined by the formula

$$\bar{\tau}_{ji} = \frac{\sum_{k=1}^{p} \tau_{ki} \lambda_{kj} L_j}{\sum_{k=1}^{p} \lambda_{kj} L_j}. \quad (14)$$

Then we can calculate a matrix $\bar{\tau}$ from which it is easy to find the initial distribution X_0. The ones ("1") should be in the positions where the latency value is minimal in each row of the matrix $\bar{\tau}$.

$$\bar{\tau} = \begin{bmatrix} 140 & 100 & 125 & 50 \\ 80 & 95 & 130 & 90 \\ 55 & 200 & 115 & 70 \\ \dots & \dots & \dots & \dots \\ 70 & 125 & 65 & 90 \end{bmatrix} \overset{min}{\Longrightarrow} X_0 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (15)$$

For the initial allocation, we need to determine the initial cost, which is usually not optimal. Given the assumption that all servers in different parts of the world scale equally, a balanced load should be achieved. When the load is even, the cost is minimized.

In each line, we analyze the possibility of moving the file to another server. With such a rearrangement, it is necessary to keep the minimum possible increase in the value of $\Delta\tau$. It is worth moving the file that leads to a minimal increase in network latency, but the cost is reduced. This process of redistribution should be continued as long as the total cost is reduced. In general, this can be described by the following flowchart (Figure 8).

Figure 7 shows the result of finding the optimum. The input parameters remain the same as in the previous example. As you can see from the graph, the heuristic algorithm took no more than 20 steps to find an acceptable solution. The difference between the results does not exceed 5%.

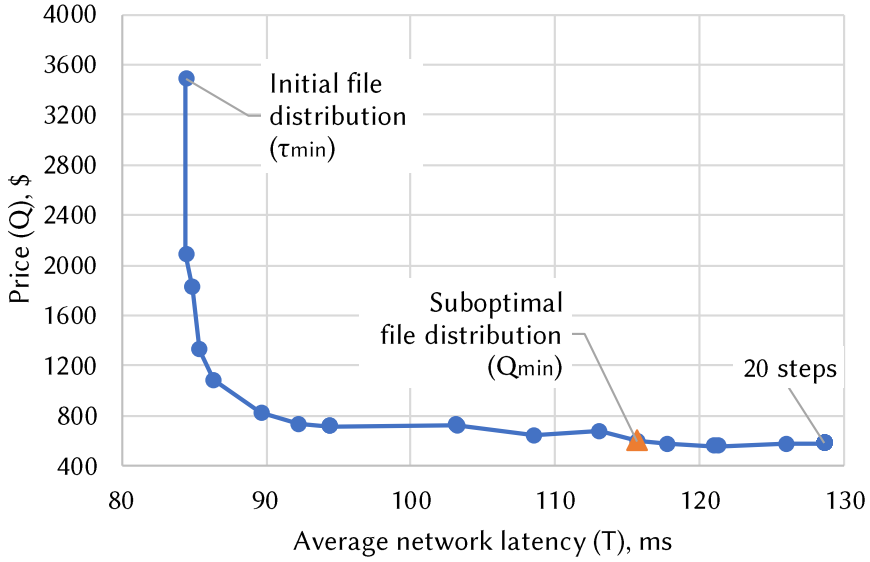

Figure 7: How the heuristic optimization algorithm works

It is important to note that by adjusting the parameters of the algorithm, we can get a compromise solution (τ ≈ 90 ms, Q ≈ 800 $). This option may be necessary to ensure a certain level of service quality.

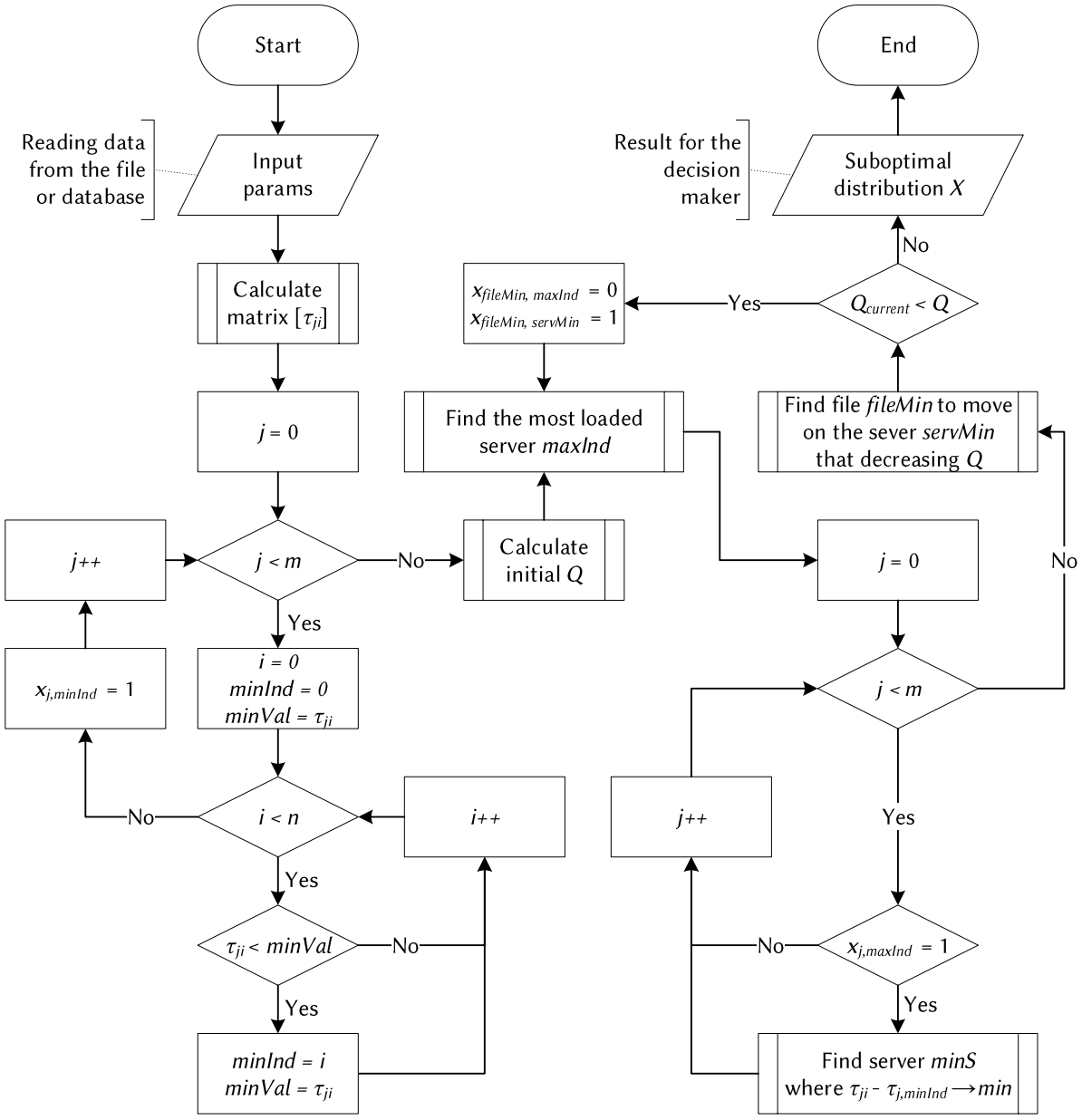

Figure 8: Flowchart of a heuristic algorithm for finding suboptimal file distribution

6. Conclusions

Thus, optimizing the placement of files in information systems is important. If there are several criteria, it is not always possible to combine them into one, because they have different units of measurement. The proposed heuristic algorithm has an advantage over the genetic optimization algorithm. It has a much higher speed, although it does not guarantee an absolute minimum. As shown in the graphs above, the error of the result does not exceed 5%.

Conducted research shows a significant variation in the weighted average network delay of more than 40%. At the same time, the cost can vary between 5-10%. This means that by moving files, you can improve the responsiveness of graphic user interface because high ping values significantly degrade the user experience.

An important aspect of optimization is reducing the load on the global computer network, since traffic passes through fewer hops and servers are loaded evenly.

Acknowledgements

The team of authors conducts research in various areas of information technology. The authors are sincerely grateful to the head of the department of Computer Systems and Networks, professor Andrey Kupin, the first vice-rector of Kryvyi Rih National University, associate professor Vladyslav Chubarov, and other colleagues. Valuable advice, organizational assistance, and support for research helped to obtain scientific results and publish this article. The authors are open for further fruitful cooperation.

References

[1] Cisco.com, Cisco Annual Internet Report (2018–2023) White Paper, 2024. URL: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.
[2] A. Piltch, SSD Prices Have Dropped 25% Since March, Now Average $0.06 per GB, 2024. URL: https://www.tomshardware.com/news/ssd-prices-sink-june-2023.
[3] M. T. Özsu, P. Valduriez, Principles of Distributed Database Systems, 4th ed., Springer, 2020.
[4] A. Petrov, Database Internals: A Deep Dive into How Distributed Data Systems Work, 1st ed., O'Reilly Media, 2019.
[5] J. Konečný, B. McMahan, D. Ramage, Federated Optimization: Distributed Optimization Beyond the Datacenter, 2015. ArXiv. doi:10.48550/arXiv.1511.03575.
[6] E. Sevinç, A. Coşar, An Evolutionary Genetic Algorithm for Optimization of Distributed Database Queries. The Computer Journal, 54(5), 2011, pp. 717-725. https://doi.org/10.1093/comjnl/bxp130.
[7] G. G. Tsehelyk, Distributed Database Systems, Lviv, Svit, 1990. 167 p.
[8] Lei Rao, Xue Liu, Le Xie, Wenyu Liu, Minimizing Electricity Cost: Optimization of Distributed Internet Data Centers in a Multi-Electricity-Market Environment, San Diego, CA, USA, 2010. doi: 10.1109/infcom.2010.5461933.
[9] X. Ma, W. Yang, Y. Zhu and Z. Bai, A Secure and Efficient Data Deduplication Scheme with Dynamic Ownership Management in Cloud Computing, IEEE International Performance, Computing, and Communications Conference (IPCCC), Austin, TX, USA, 2022, pp. 194-201, doi:10.1109/IPCCC55026.2022.9894331.
[10] P. Austria, C. H. Park, A. Hoffman and Y. Kim, Performance and Cost Analysis of Sia, a Blockchain-Based Storage Platform, 2021 IEEE/ACIS 6th International Conference on Big Data, Cloud Computing, and Data Science (BCD), Zhuhai, China, 2021, pp. 98-103, doi:10.1109/BCD51206.2021.9581866.
[11] K. Lee, J. Kim, J. Kwak and Y. Kim, Dynamic Multi-Resource Optimization for Storage Acceleration in Cloud Storage Systems, in IEEE Transactions on Services Computing, vol. 16, no. 2, pp. 1079-1092, 1 March-April 2023, doi:10.1109/TSC.2022.3173333.
[12] Data Center Knowledge, How Many Servers Does Google Have? 2017. URL: https://www.datacenterknowledge.com/data-center-faqs/google-data-center-faq.
[13] Digitalocean.com, Regional Availability Matrix, DigitalOcean Documentation, 2024. URL: https://docs.digitalocean.com/products/platform/availability-matrix.
[14] K. K. Devleker, How to Choose a Data Center Location for Your Business, 2023 URL: https://www.digitalocean.com/blog/choosing-a-data-center-location.
[15] Meter.net, VULTR.net – statistics, 2024. URL: https://www.meter.net/test-server/102-vultr.
[16] Vultr.com, High Performance, High Frequency, Bare Metal, Affordable Cloud Computing, 2024. URL: https://www.vultr.com/pricing/#optimized-cloud-compute.
[17] M. A. Noaman Al-hayanni, F. Xia, A. Rafiev, A. Romanovsky, R. Shafik, A. Yakovlev, Amdahl's law in the context of heterogeneous many-core systems – a survey, IET Comput. Digit. Tech., 2020, Vol. 14: pp. 133-148. https://doi.org/10.1049/iet-cdt.2018.5220.
[18] A. Kupin, I. Muzyka, D. Kuznetsov, Y. Kumchenko, Stochastic Optimization Method in Computer Decision Support System, International Conference on Theory and Applications of Fuzzy Systems and Soft Computing. Advances in Intelligent Systems and Computing, 2018, Vol. 754. pp. 349-358.