Cloud Computing Task Scheduling Algorithms and Advances

Yifan Zhang*

Department of computer, North China Electric Power University (Baoding), Baoding, China.

*Corresponding author e-mail: zhangyifan@ncepu.edu.cn

Abstract. In today's big data era, cloud computing has become an indispensable part of people's production and life. Cloud computing is a pay-as-you-go service approach, which is responsible for integrating various resources for users in need. However, with the exponential growth of users and data volume in recent years, the cloud platform is facing great challenges in terms of storage, computing power and network I/O. The problem of reasonable scheduling and allocation of tasks has an important impact on the performance of cloud computing. This paper describes the cloud platform and its scheduling problem in detail, lists its development of task scheduling problem in recent years, explains its advantages and disadvantages, and summarizes and outlooks its future development.

Keywords: Cloud computing, Task scheduling strategy, Heuristic algorithm.

1. Introduction

In this digital era, massive amounts of information are generated every day and data is carried in various forms, which makes the scale of data processing and storage unprecedented. Cloud computing, as a distributed virtualized resource, has become a common solution for processing data in the Internet due to its powerful computing power, storage capacity and convenience. The core idea of cloud computing is to integrate multiple idle physical computer resources (e.g., computing resources CPU, storage resources DRAM, network resources I/O, etc.) around the world through heterogeneous technologies and they are managed by the cloud. Users do not need to consider the underlying connectivity details of the cloud platform when using cloud resources, nor do they need to purchase hardware and software and other infrastructure, but only pay for them according to their purchase needs. However, there is a potential problem that subscribers are not guaranteed to use cloud resources efficiently for the duration of their ownership, which may result in a significant amount of idle or wasted cloud resources. Therefore, with the development of cloud technology and the widespread deployment of cloud platforms, the problem of scheduling resources and tasks in cloud computing, for example, how to rationally allocate real computer resources mapped by virtual machines to many tasks and dynamically coordinate them according to the changes of task demands, has become one of the hot topics of research in cloud computing these years [1].

2. Cloud Platform and Its Scheduling Issues

2.1 Cloud Platform

Cloud computing, also known as grid computing, is actually a pay-as-you-go business service model in essence. The most distinctive feature of cloud computing is its virtualization technology, which breaks through the boundaries of time and space and enables the separation of physical resources from the virtual deployment environment, so that there is no need to worry about the impact on physical resources when backing up, migrating and expanding data. It can be regarded as a network of resource provisioning, and users can continuously obtain resources from the "cloud" and pay for them according to their needs, which is the third major change in the information age after computers and the Internet. In addition, compared with the traditional network application model, it also has the characteristics of on-demand deployment, high flexibility, high reliability, and cost-effectiveness.

The convenience of information technology makes many enterprises are using or planning to use cloud computing. These cloud resources can be divided into three categories from the perspective of

security of use [2]: public, private, and hybrid clouds. Public clouds are less costly and can be accessed by users through a common network, and with computer infrastructure and resources. However, the disadvantages of public cloud are obvious, as a large number of users performing tasks online at the same time is likely to cause network congestion, and the security cannot be guaranteed. The private cloud is customized for each user, and it can solve the security and network congestion problems of the public cloud well. However, it can be expensive and requires a more demanding environment for access, and sometimes ultra-remote access is not available. To combine the best of two, more and more enterprises are choosing to use hybrid clouds. While this approach solves old problems, it also creates some new ones, such as potential compatibility issues when interconnecting infrastructure due to the use of different types of cloud platforms, as well as significant challenges in data integration.

There are many cloud computing providers developing their own cloud platforms for customers to use, and there is no unified standard system among them. There are many differences in the underlying architecture design, but this does not affect users' use. According to the service model of cloud platforms, they can be divided into three categories [3], which are infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), and software-as-a-service (SaaS) models. Common IaaS [4] are Amazon AWS EC 2 [5], AL iCloud ECS [6], which only provides a most basic operating system, and other software is installed by users themselves according to their needs. Common PaaS [7] are Google's GAE [8], and domestic SAE [9], which provide a complete runnable environment that does not allow users to install software and provides the infrastructure and computing resources needed to develop, test, run, and manage SaaS [10] applications in an integrated cloud environment. We will be more familiar with SaaS [10], and it is both convenient and inexpensive. Commonly used are Apple iCloud [11], Google Doc [12] and Baidu Cloud, which provides an off-the-shelf software that allows users to access software applications over the Internet.

2.2 Scheduling Problem in Cloud Computing

The scheduling problem in cloud computing is an NP-Hard problem, which can be subdivided into two categories: task scheduling and resource scheduling. Task scheduling is to dynamically assign tasks based on the workload of the server so as to achieve scheduling goals such as load balancing and cost-effectiveness. Resource scheduling is to dynamically allocate resources to users by responding to their needs and following certain rules. Resource allocation is not necessarily fair, and it needs to set scheduling goals based on specific application scenarios. In general, both scheduling issues exist in a cloud service, but task scheduling is more important. Because task scheduling can actively select nodes with suitable resources to execute tasks, while resource scheduling is reactive and is forced to start when a node is found to be under-resourced during task execution. Therefore, a reasonable task scheduling strategy not only improves the user experience, but also keeps the cloud resources at a high utilization rate in the long run, avoiding idle and wasteful resources.

The following metrics are often used as scheduling objectives for a task scheduling strategy [13]: load balancing, economic principles, time horizon minimization, and quality of service (QoS). Achieving load balancing is a more challenging goal due to the heterogeneity among multiple cloud platforms of large scale. Lang et al. [14] pointed out the importance of load balancing in big data centers and show a web case of load balancing based on Nginx. Li [15] et al. argued that unbalanced workload between nodes not only affects cluster performance, but also affects user experience. They considered resources such as CPU, I/O, DRAM, and used an improved genetic algorithm to demonstrate better performance when the cluster achieves load balancing. Due to the pay-as-you-go usage rule of the cloud platform, users and enterprises have to consider economic metrics and seek an acceptable price/performance ratio when using the cloud platform. Deelan [16] et al. argued that the start-up and shutdown of numerous servers also incur significant cost loss. In order to scale the cluster more economically, Ge et al. [17] proposed an architecture for scaling the cluster and a scheduling strategy based on computational intensity, which could reduce the cost by selecting the appropriate cloud resources according to different needs. Time span minimization is a common

optimization metric in cloud computing scheduling problems, where both users and service providers want tasks to be completed in the shortest time span. IBM's cloud platform is targeting the optimal span for task scheduling [18]. Sun et al [19] proposed a new scheduling strategy that combines dynamic multilevel resource pooling with tenancy theory, which can reduce the time of resource constraints, avoid resource waste and thus improve system utilization. In cloud services, it is also an important indicator to ensure the quality of service and optimize the user experience, which is an important guarantee to maintain the longevity of the cloud platform. Chanhan [20] et al. used network I/O as a QoS indicator and divide the QoS priority according to the execution time, and the one with short execution time is dispatched first. Sun [21] et al. proposed a new scheduling mechanism by establishing a multi-for QoS optimization objective function and combining it with the immune Cologne algorithm, which has significantly improved the problems such as load imbalance and low user experience.

3. Task Scheduling Methods in Cloud Computing

The essence of task scheduling in cloud computing can be viewed as a planning problem with constraints, where the system obtains an approximate optimal solution by satisfying the optimal value of the objective function as much as possible, and the scheduling algorithms can be classified into different categories according to different division bases. From the viewpoint of the number of objectives, it can be divided into single-objective scheduling and multi-objective scheduling. The number of "clouds" can be divided into single-cloud scheduling and cross-cloud scheduling. This section will focus on the target-based scheduling strategy.

3.1 Single-Target Scheduling

Single-target scheduling is a traditional scheduling strategy that evolved from common algorithms based on operating systems. Such as first-in-first-out scheduling (FIFO), short job first scheduling (SJF), minimum execution time scheduling (MET), minimum completion time (MCT), and round robin queue scheduling (RR). Specifically, in FIFO, the priority of a task is determined by the time of its arrival, with the first task to come first and the later task to be executed later. This policy has simple rules but does not consider other constraints. The SJF policy prioritizes the arriving tasks in the task queue by the estimated execution time, so that short jobs are executed first. If there are always short jobs arriving, it will cause long jobs to always fail to execute. The MET mechanism assigns tasks to the resources with the smallest time span, which can cause the problem of unbalanced liabilities when there are significant performance differences between physical devices. MCT assigns tasks to different nodes in any order and completes that task with the shortest execution time. Although this maintains a balanced load between nodes, some tasks do not have the shortest execution time on that node, so it will make the overall time cycle too long. RR allocates resources for each task with a quantitative time according to fairness. Once some tasks spend too long time, the problem of long overall execution time still exist. A common feature of these scheduling strategies is that they discard multiple constraints to achieve an optimal solution in one aspect, which will create many problems. If the least overall time is pursued, it will lead to unbalanced liabilities between nodes. If the load fairness between nodes is pursued, it may cause the problem of long execution period due to uneven task length. In order to take a compromise between many constraints and to better meet the practical needs of task scheduling, here comes following multi-objective scheduling strategies.

3.2 Multi-objective Scheduling

In practical application scenarios, it is often necessary to consider the optimization of multiple objectives in order to obtain a relatively optimal solution, hence multiple heuristic algorithms are brought out. The commonly used heuristic algorithms are classified into two categories [21].

Biologically based heuristic algorithms. For example, the genetic algorithm (GA) in 1975, the taboo search algorithm (TSA) in 1986, the differential evolution algorithm (DE) in 1995, and the imperial competition algorithm (ICA) in 2007.

Group intelligence based heuristic algorithms. For example, simulated annealing algorithm (SA) in 1953, particle swarm optimization (PSO) in 1995, artificial bee colony algorithm (ABC) in 2002, bat algorithm (BAT) in 2010, symbiotic organisms search (SOS) in 2014, moth to flame optimization (MFO) in 2015.

Genetic algorithm. Genetic algorithm was first proposed by John [22] in 1975, who borrowed the law of biological evolution of nature and applied Darwin's evolutionary theory of “survival of the fittest” to practice. The algorithm iterates through selection, crossover and variation to produce optimal solutions. Although this method is globally optimized and highly parallel, the experimental results are directly affected by the population size, crossover coefficient and variation coefficient, and are highly uncertain.

Taboo search algorithm. Taboo search is an iterative search algorithm first proposed by Glover, an American, in 1986. Its most important feature is to use memory priming to guide the search process of the algorithm. In other words, to build a taboo table during the search process to store the local optimal solutions that have been obtained to prevent repeating the previous actions and falling into the local optimal solutions, so as to gradually achieve the global optimum. However, it also relies heavily on the information in the taboo table, especially the initial solution.

Differential evolution algorithm. In 1995, differential evolution was first proposed by Torn and Price, which is an improvement on the idea of genetic algorithm. The variance vector is generated from the father's differential vector and crossed with it to generate a new individual vector, which is selected together with its father's individual. This results in better convergence of optimal results and reduced hyperparameters, but the convergence slows down significantly as the number of iterations increases.

Imperial competition algorithm. First proposed by Atashpaz-Gargari and Lucas in 2007, the imperial competition algorithm is an evolutionary algorithm that models the mechanisms of imperialist colonial competition [23], which can be seen as a product of genetic algorithms in sociology. Although this algorithm is also a global search and converges quickly and with high accuracy, it may lead to a premature end of convergence and "premature maturity" as many countries die out (population diversity decreases).

Simulated annealing algorithm. The simulated annealing algorithm was first proposed in 1953 by N. Metropolis [24], who drew on the cooling process of smelting metals. The core idea of the algorithm is a serially structured greedy strategy that jumps out of the current locally optimal solution with a certain probability each time and stops iterating when the temperature is low enough. The iteration speed is affected because the probability and temperature are difficult to control.

Particle swarm optimization. This algorithm, proposed by Eberhart and Kennedy, is a typical local optimum strategy. It simulates the behavior of birds foraging for food, comparing each bird searching for food to a "particle", and each iteration has to follow the optimal "particle" for random search. The single-college mobility of the update information leads to faster convergence, but also leads to falling into local optimal solutions.

Artificial bee colony algorithm. This algorithm simulates the behavior of bees collecting honey, either serially or in parallel. The division of labor between the bees is different, with the leader and follower bees responsible for collecting honey and accelerating the convergence of results, while the scout bees providing a safeguard against the algorithm falling into local optimum solutions. However, the choice of parameters, such as the population size, the number of iterations, and the ability of scout bees, relies entirely on experience.

Bat algorithm. This algorithm simulates the echolocation of bats in nature, and iterates to continuously approach the optimal solution while having a set of random initial solutions. Although it does not have too many parameters to adjust compared to other heuristic algorithms, it is prone to

fall into local optimal solutions and has disadvantages such as slow convergence and low accuracy in later stages.

Symbiotic organisms search. This algorithm is based on the symbiotic phenomenon in biology, and the whole process is divided into three phases: mutual benefit phase, symbiosis phase and parasitic phase. Compared with most intelligent optimization algorithms, it does not have too many hyperparameters and is suitable for large-scale tasks and has high stability, but it also has the same common problems of most algorithms, like "premature", slow convergence, low precision and other characteristics.

Moth to flame optimization. This algorithm was first proposed by Seyedali Mirjalili [25] et al. in 2015, and it has the advantages of strong parallelism and global optimality compared to many previous algorithms. This is due to the fact that the moth is a greed-free strategy in its flight, which uses a special navigation mechanism for lateral positioning with the light source, always flying at a fixed angle to the light source. Therefore, when the ce is closer to the moth, it leads to navigation failure and generates a fatal spiral flight path. light sour.

4. Conclusion

In present study, a detailed description of cloud computing from multiple perspectives has been given, pointing out the salient features of cloud computing and the specific uses of different types of cloud platforms. Then, the common objectives in cloud computing scheduling problems have been presented, and the current status of task scheduling strategies in cloud computing is analyzed from scheduling objectives perspective. Strategies in terms of scheduling ideas, scheduling characteristics, and specific implementations have been analyzed and research shows that single-objective task scheduling strategies are computationally simple and can obtain the most intuitive answers, but they are not universally applicable in practical scenarios. Multi-objective task scheduling is an NP-Hard problem, and usually there is no optimal solution, but only a compromise of multiple constraints to produce a relative optimal one. Therefore, in multi-objective optimization, it is important to choose a suitable scenario, otherwise the modeling will be meaningless. At present, there are also some artificial intelligence-based scheduling algorithms, which will be a new trend to be widely used in the cloud computing field in the future.

Besides, there are other problems existing in the cloud computing field at present. With the development of digital society, “cloud” is playing an irreplaceable and increasingly important role in our daily life. As a huge amount of data is uploaded to and downloaded from the cloud every moment, and the cloud servers are thousands of miles away from the users. This not only poses a great challenge to the I/O of network resources, but also makes the physical devices (disks, routers, sensors, etc.) that transmit the resources under huge load pressure. Keeping peak values and high throughput for a long time can reduce the life of these devices. Therefore, low latency, low power consumption, and low cost have also become one of the challenges for future cloud computing performance optimization.

References

[1] WU F, WU Q, TAN Y. Workflow scheduling in cloud: a survey [J]. The Journal of Super Computing, 2015, 71(9): 3373-3418.

[2] TIAN C, HUANG Z, ZHANG Y. A review of research on task scheduling methods for cloud computing environments [J]. Computer Engineering and Applications, 2021, 57(02):1-11.

[3] VOUK M A. Cloud computing: issues, research and implementations [J]. Journal of Computing and Information Technology, 2008, 42(4): 235-246.

[4] Zhang L. A review of research on cloud computing IaaS instances [J]. Software Guide, 2017, 16(08):208-210.

[5] Todd R Weiss. New AWS EC2 Instances Target Data-Intensive Cloud Workloads [J]. SQL Server Pro, 2017:

[6] Huang H, Han J,Hu 乂,Han F. Ali cloud-based data storage[J]. Science and technology innovation and application, 2021, 11(23):50-52.

[7] Qi M. Research on cloud service supply chain of PaaS model [D]. Nanjing University, 2014.

[8] Wang J. Design of news publishing system based on google app engine [J]. Fujian computer, 2017, 33(11):107.

[9] Chu Y, Chen D. Exploration of cloud computing application based on Sina App Engine platform[J]. Computer Knowledge and Technology, 2014, 10(23):5441-5444.

[10] Lu L. Research on trustworthiness evaluation of SaaS services in cloud computing environment [D]. Beijing University of Posts and Telecommunications, 2021.

[11] Wang F, Fang Y. Another milestone in international cyberspace cooperation - Cloud on Guizhou operates US Apple's iCloud China data [J]. Contemporary Guizhou, 2018(05):74-75.

[12] MacIsaac Dan. Dolores Gende's "Resources for Physics Remote Learning" Google Doc [J]. The Physics Teacher, 2020, 58(7):

[13] Zuo 乚, Cao 乙. A review of research on scheduling problems in cloud computing [J]. Computer Application Research, 2012, 29(11):4023-4027.

[14] Lang W, Yao J, Zhao Y. Research on load balancing technology for large data centers [J]. Telecommunications Express, 2018(04): 1-4.

[15] Li C, Xie Y. Simulation of resource load balancing scheduling optimization in cloud computing environment[J]. Computer Simulation, 2017, 34(12):420-425.

[16] DEELMAN E, SINGH G, LIVNY M, et al. The cost of doing science on the cloud: the Montage example [C]/ /Proc of ACM/IEEE Confe- rence on Supercomputing. Piscataway: IEEE Press, 2008: 1-12.

[17] Ge X, Chen H, Du B, et al. Research on scheduling strategies in cloud-based computing cluster scaling [J]. Computer Application Research, 2011, 28(3) : 995-997.

[18] Tian W Zhao Y. Cloud Computing: Resource Scheduling Management [M]. Beijing: Defense Industry Press, 2011.

[19] Sun R, Zhao Z. Resource Scheduling Strategy Based on Cloud Computing [J]. Aviation Computing Technology, 2010, 40(3): 103-105.

[20] CHANHAN S S, JOSHI R C. A heuristic for QoS based independent task scheduling in grid environment [C]/ /Proc of International Con- ference on Industrial and Information Systems. 2010: 102-106.

[21] Singh P, Dutta M, Aggarwal N. A review of task scheduling based on meta-heuristics approach in cloud computing. knowledge and information systems, 2017, 52(1): 1-51.

[22] Holland J H. Adaptation in Natural and Artificial Systems [J]. University of Michigan Press Ann Arbor Mich, 1975.

[23] Guo W, Ye D. Evolutionary optimization of imperial competition algorithms [J]. Computer Science and Exploration, 2014, 8(4): 473-482.

[24] Steinbrunn M, Moerkotte G, Kemper A. Heuristic and Randomized Optimization for the Join Ordering Problem[J]. The VLDB Journal, 1997, 6 (3):8 - 17.

[25] Li 乙, Mo 丫. Mothballing optimization algorithm based on Lévy flight [J]. Computer Engineering and Design, 2017, 38(03):807-813.