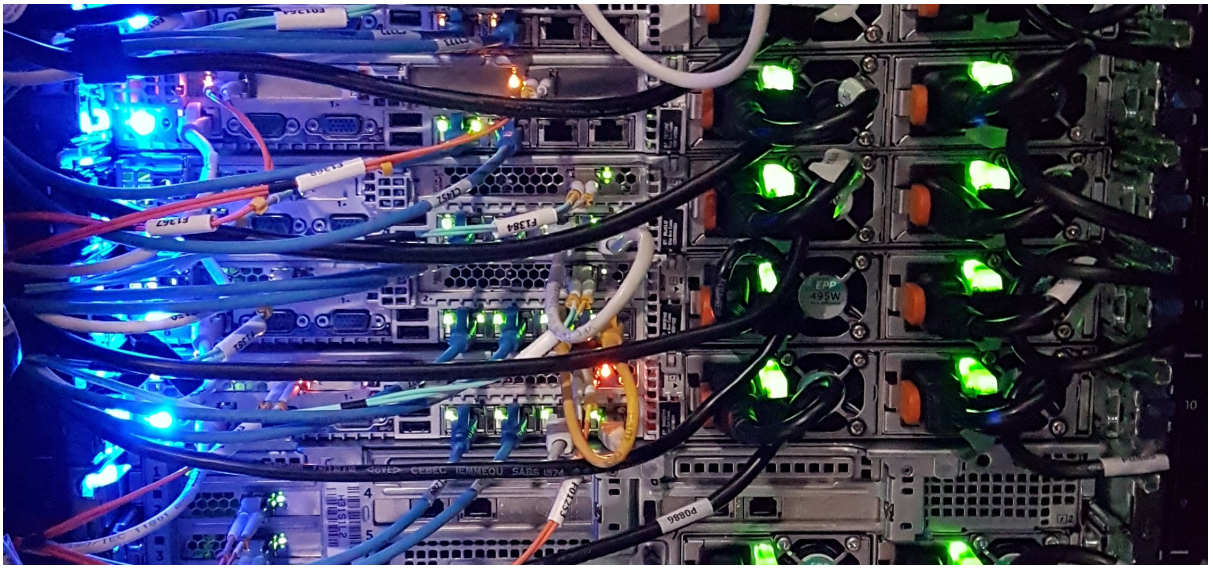

Load Balancing /Scheduling algorithms for cloud computing

Srijan Dutta | 23MCA0131 | PMCA506L

References

Introduction

Cloud Computing can be defined as the on-demand delivery of IT resources over the internet as with a pay-as-you-go pricing scheme [1]. Instead of owning a headache of maintaining and setting up a physical server or data centers, organizations can hire various Cloud Service Providers for the fraction of the actual cost of the servers.

Cloud Computing has revolutionized the way of managing the digital infrastructures and service delivery through its flexibility, scalability, cost efficiency. However, as we slowly slide into the information era where literally everything is documented and analyzed, there is always an unprecedented growth of data and processing or generation demands. Therefore as the demand increases, there is a significant chance of latency related issues which is really not a desirable quality to have where execution and delivery speed is of the utmost importance.

In order to fix this problem we need efficient resource allocation and management. Load balancers and Scheduling Algorithms play a pivotal role in optimizing the utilization of cloud resources, ensuring equitable distribution of workloads, and enhancing system performance and reliability. These load balancing/scheduling algorithms works as orchestrators that intelligently handles incoming requests and performs computational tasks among multiple servers, virtual machines or containers. They always ensure that cloud resources are fully utilized and maintained at all times so that services are available to customers on demand.

This literature study moves through the uses and trends of various load balancing and scheduling algorithms in order to provide a better understanding of the ways of a cloud server and how it handles all the incoming requests. Moreover we learn about the various technologies associated with it, platforms they are deployed on and on what pace and how research is being conducted on this ever evolving field.

Research Works

Search Strategy

In conducting this literature study, an extensive search strategy was used in order to identify all the relevant research topics, articles and conference papers and scholarly publications. Databases such as IEEE Xplore, and Google Scholar was used. To get

efficient results, a combination of ‘keywords’ and phrases were used like `load balancing`, `scheduling`, `cloud computing`, `resource allocation` and `server optimization`. Boolean operators like AND and OR were used to refine the results. In order to focus on the most recent trends, publications from the last 5 years were considered.

Inclusion and Exclusion Criteria

In order to maintain the relevance and quality of the study strict inclusion and exclusion criteria were applied. The inclusion criteria includes articles published between the years 2018 and 2023. Each one of the 10 selected papers went through a thorough review with a special emphasis on the research methodology, key findings and relevance to the topic of load balancing and scheduling algorithms in cloud computing.

Data Extraction

Information extracted includes publication year, number of citations, author names, objective of the study, key findings and unique insights related to load balancing and scheduling algorithms. This structured approach allowed for better organization and analysis of the literature.

Recent Trends

Load Balancing in Fog and Edge Computing

Fog Computing is a recent research trend to bring cloud computing services to network edges. Edge Datacenters (EDCs) are deployed in order to reduce the latency and network congestion by processing user requests in real time [2]. In some cases a few edges might be using little to no resource while a few others will be overwhelmed. This is where load balancing kicks in as it helps to divide the workload equally among all the edge servers thus avoiding network congestion.

A notable challenge in edge computing is the secure authentication of the EDCs. As the EDCs are often deployed in unattended or open-access environment, ensuring the authenticity and trustworthiness is of utmost importance. The recent trends now incorporate authentication of EDCs using load balancing. This can be achieved through the following ways:

- Adaptive EDC Authentication: The development of adaptable EDC authentication methods is a current trend, frequently started by a centralized cloud datacenter. Each EDC will be checked and provided access to join the network through cloud-initiated authentication.
- Sharing Load Information: During the authentication process, the EDCs will share important information that helps the algorithm to make better decisions regarding the task allocation and communication overhead.
- Sustainable Load Balancing: Here not only the current EDC is evaluated but also the destined EDC is checked for its current load status thus providing a dynamic approach for load utilization.

Thus incorporation of Load Balancing in Fog and Edge computing brings :

- enhanced security that mitigates the risk of unauthorized nodes,
- efficiency that improves resource utilization and reduces response time,
- scalability that makes easier to handle complex environments,
- performance validation that ensures that they meet the necessary requirements.

Machine Learning Powered Load Balancing

Cloud workloads are typically managed in a distributed environment and processed across geographically distributed data centers [3]. Such large scale data management and orchestration is a difficult problem to solve. Often mathematical formulae are used to maintain and address all the issues but with the different types and complexity of all the cloud architecture and the ever changing environments make it difficult to incorporate these mathematical optimization techniques. So in order to make things easy, researchers now introduce machine learning algorithms that adapt themselves in ways to adjust to the needs of the cloud architecture.

The authors in [3] has described a few machine learning algorithms and how they work. They go on to explain how the main problem in the geo-distributed data center is the handling of data and workload. They provide effective solution through:

- Workload to Resource Mapping Optimization: It refers to the assigning of incoming workloads to the best hardware resources available across the data centers. It incorporates an adaptive resource management technique based on regression learning with custom random action selection. The proposed

RL based technique aims to balance the quality of service and power consumption of the data center.

- VNF Placement Optimization: Here the objective is to use service function chaining (SFC) to establish virtual network functions (VNFs) in distributed data centers and facilitate routing between them. This method uses Deep Reinforcement Learning (DRL). This proposed method is quite beneficial as it reduces DRL action space, combines model based algorithms with model free DRL for improved efficiency also the model based information, such as, gradients, leads to significant performance enhancement in the algorithm.

Algorithm

So far we have been studying why we need and where we need to incorporate the load balance and scheduling algorithms. In this section we dive into the comprehensive study of a few load balancing and scheduling algorithms. These algorithms are fundamental to the optimization of resource allocation, enhancing system performance and ensuring a responsive cloud computing infrastructure.

Meta-Heuristic Algorithm

A meta-heuristic algorithm is a versatile and high level problem solving approach used in optimization and search problems. Unlike traditional algorithms that mostly follow deterministic and specific procedures, meta-heuristic algorithms usually use algorithms that usually draw inspiration from nature and human-specified behavior. Some key characteristics of this kind of algorithms are:

- Heuristic Nature: They do not guarantee finding the best solution but aim to explore and improve solutions efficiently.
- Versatility: These algorithms provide solution for a vast range of optimization problems, including scheduling , routing, clustering, and more.
- Iterative Improvement: These algorithms iteratively refine solutions by making small adjustments and evaluating their quantity.
- Inspiration from nature: Most of these algorithms are inspired from how nature or natural phenomena works.
- Parameter Tuning: Meta-heuristic algorithms may have several parameters that need to be tuned for specific problems. Proper parameter setting is

crucial for their effectiveness.

- Convergence: Over iterations, meta-heuristics tend to converge towards a solution. The quality of the final solution depends on the algorithm's design, parameter settings, and the nature of the problem.

There are many Meta-Heuristic Algorithm approaches. A two of them are described below:

Bat Algorithm

This is a nature inspired optimization algorithm based on the echolocation behavior of bats. This algorithm mimics how the bat preys for its food through the use of echolocation, adjusting the frequency and loudness of the sound emitted based on the proximity of the prey.

Here's how it works:

```
Step 1:
It starts with an initial population of virtual machines(VMs) and sets their attributes,
such as position, velocity, loudness, pulse rate, and frequency.

Step 2:
Each of these bats emits a pulse (ultrasonic call) representing a potential solution.
These pulses represents the VMs current position in the solution space.

Step 3:
Bats adjusts the loudness and pulse rate based on the current situation.

Step 4:
Bats update their position based on the loudness and pulse rate adjustments.

Step 5:
Bats perform a local search around their current positions to refine their solutions
further.

Step 6:
The algorithm keeps track of the best solution found by any bat.

Step 7:
The algorithm iterates until a termination criterion is met, such as the max number of
iterations.
```

In cloud computing, this algorithm is used to distribute tasks or workloads among virtual machines(VMs) in the most optimized way.

The main advantage of this algorithm is its ability to handle complex organization problems, making it suitable for addressing load balancing challenges in cloud

environments. It balances the trade-off between exploration and exploitation to converge towards an optimal or near-optimal solution [4].

[4] suggests that the Bat algorithm is ideal for for load balancing because of its ability to generate uniform load distributions, optimize traffic circulation, and reduce overloading of servers and cloud nodes. It enhances the quality of service (QoS) and improves resource utilization in cloud computing systems.

Ant Colony Optimization Algorithm (ACO)

Ant Colony Optimization is an intelligent optimization algorithm that is applied to solve large-scale task scheduling problems in the cloud computing environment. [6] fights the decrease in distribution efficiency with increasing problem size, which is a prevalent issue when utilizing fundamental ACO for large-scale job scheduling. The authors propose a new ACO algorithm with a concentration on load balancing to get over this limitation and increases its usefulness in large-scale scheduling settings.

The primary objective of the ACO is to optimize the scheduling tasks in a cloud computing environment. It works in such a way that it balances load evenly while reduces the execution time of the scheduling process.

The key components of this algorithm includes:

- Task Scheduling Model: It consist of the user given tasks, the virtual machines and the processing time of each task in a given VM.
- Effective Ants: These are those that considers load balancing constraints when selecting paths for task scheduling. This addition helps to ensure that tasks are evenly distributed among the systems.
- VM Selection: Here the approach is probability based, which means ants select the VMs based on a combination of pheromone levels and visibility which are influenced by factors like pheromone heuristic factor (α) and visibility heuristic factor (β).
- Objective function: The main aim here is to minimize the execution of the task after they are allocated to each VM so that no VM is heavily loaded.
- Pheromone Update: It is responsible for the adjustment of the pheromone levels for the paths taken by each ant. This ensures that paths that leads to better load-balanced solutions receive higher pheromone levels so that future ant behavior is influenced to take the best possible solutions.

This is how the following algorithm works:

```
Initialize:
  Set parameters (α, β, ρ, Q0, iterations, num_ants, utilization_threshold)
  Create pheromone matrix τ
  Initialize VM runtime and utilization arrays
  Create ant colony with num_ants ants

While iterations > 0:
  For each ant in the colony:
    Initialize ant-specific variables
    Generate a random task scheduling sequence Qt
    Create VM selection sequence Qv

    For each task in Qt:
      Calculate probabilities for VM selection
      Select a VM for the task based on probabilities
      Update Qv and VM runtime

    Calculate the makespan (maximum VM runtime) for the ant's solution

    If makespan < best_makespan:
      Update best_makespan and best solution
      Apply pheromone update rule to selected paths by the ant

  Update pheromone levels globally for all paths

  For each VM:
    Calculate CPU utilization
    If utilization < utilization_threshold:
      Evaporate pheromone on paths to that VM

  iterations -= 1

Return best solution found
```

Explanation:

- `α` and `β` are pheromone and visibility heuristic factors, respectively.
- `ρ` is the pheromone evaporation rate.
- `Q0` is a parameter controlling the exploration-exploitation balance.
- `iterations` is the number of iterations or generations.
- `num_ants` represents the number of ants in the colony.
- `utilization_threshold` is a threshold for VM CPU utilization, below which pheromone evaporation is applied.

Technologies

Load Balancer and Scheduling Algorithms have a wide range of applications. These are a few technologies that needs these algorithms in order to provide seamless services.

Content Delivery Networks (CDNs)

A CDN is a sophisticated network infrastructure designed to enhance the delivery of the web content to the end users [8]. It improves the QoS when we access content online. CDNs achieve this by replicating the contents of the main server to the global network of cache servers. This replication allows users to deliver content quickly and reliably to end-users from the optimal local server.

This technology combines computing technologies, high performance networking infrastructures and distributed content management techniques. The main aim of CDN is to address the problems that occur due to large number of incoming requests like network congestion and high response time.

As said in [8] CDN comprises of several key components like content providers, CDN providers and end users. Content providers deliver their content through CDNs so that end users can access it seamlessly.

CDNs employ caching and replica servers located in various geographic locations. These servers, often referred to as edge servers or surrogates, store copies of the content. When a user requests content, they are redirected to the nearest surrogate server, which delivers the requested content. This architecture provides transparency to users and significantly improves content delivery speed.

In summary, CDNs play a crucial role in improving internet content delivery by strategically replicating content and efficiently redirecting user requests to nearby servers. Their architecture and services have evolved to address the challenges of network congestion and performance, making them an integral part of modern web infrastructure.

Serverless Computing

As mentioned in [9], Serverless computing is a cloud computing paradigm hat allows developers to focus solely on writing code functions without worrying about managing servers or infrastructure. With serverless computing, developers upload their code functions to a cloud platform, and these functions are executed in

response to specific triggers, such as events or HTTP requests. The key features of serverless computing include:

- The pay-as-you-go model
- Automatic scaling
- Stateless functions
- Fine grained billing

Serverless computing has gained popularity because of how simple it is, cost effective, and scalable. It provides a wide range of functionalities and can thus has many applications. Major cloud providers like Amazon, Google, Microsoft provides serverless computing platforms thus making it one of the most popular forms of cloud computing technologies.

Platforms

Cloud Service Providers (CSPs)

Computing services are often provided by the CSPs like Amazon, Microsoft, Google, etc.

Virtualization

Usually CSPs uses a hypervisor to virtualize hardware or software. Some examples include: VMWare, Microsoft Hyper - V, and KVM

Containerization Platforms

Platforms like docker and Kubernetes are used to deploy easily scalable cloud applications and manage the environment. These platforms offer features for load balancing.

Serverless Computing Platforms

They include AWS Lambda, Azure Functions and Google Cloud Functions making them relevant for load balancing discussions

Edge and Fog Computing Platforms

These could include platforms provided by IoT companies, networking equipment manufacturers, and cloud providers with edge offerings.

References

[1] aws.amazon.com/what-is-cloud-computing

[2] Puthal D, Obaidat MS, Nanda P, Prasad M, Mohanty SP, Zomaya AY. Secure and Sustainable Load Balancing of Edge Data Centers in Fog Computing. IEEE Communications Magazine. 2018 May;56(5):60-65. doi: 10.1109/MCOM.2018.1700795.

[3] Hogade N, Pasricha S. A Survey on Machine Learning for Geo-Distributed Cloud Data Center Management. IEEE Transactions on Sustainable Computing. 2023;8(1):15-31. doi:10.1109/TSUSC.2022.3208781.

[4] Hamidi A, Goal MK, Astya R. Load Balancing in Cloud Computing Using Meta-Heuristic Algorithm: A Review. In: Proceedings of the 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom); 2022. New Delhi, India: pp. 639-643. doi: 10.23919/INDIACom54597.2022.9763131.

[5] Senthil GA, Somasundaram K, Arun M, Naga Saranya NN, Prabha R, Vijendra Babu DV. A Novel Hybrid GAACO Algorithm for Cloud Computing using Energy Aware Load Balance Scheduling. In: Proceedings of the 2022 International Conference on Computer Communication and Informatics (ICCCI); 2022. Coimbatore, India: pp. 1-5. doi: 10.1109/ICCCI54379.2022.9740795.

[6] Ye J, Zhang L. The Load Balancing Ant Colony Optimization Based on Cloud Computing. In: Proceedings of the International Conference on Network, Communication, Computer Engineering (NCCE 2018); 2018: pp. 953-956.

[7] Joshi V. Load Balancing Algorithms in Cloud Computing. Int J Res Eng Innov.

[8] Pathan AMK, Buyya R. A Taxonomy and Survey of Content Delivery Networks. IEEE Trans Network Serv Manag. 2008;5(4):202-214.

[9] Castro P, Ishakian V, Muthusamy V, Slominski A. The Rise of Serverless Computing. Commun ACM. 2019;62(12).

[10] Magesh Kumar S, et al. Innovative Task Scheduling Algorithm in Cloud Computing. IOP Conf. Ser.: Mater. Sci. Eng. 2020;981:022023.