International Journal of Engineering Research & Management Technology
ISSN: 2348-4039
Volume 9, Issue-1 Jan-Feb 2022
Email: editor@ijermt.org www.ijermt.org

A Survey of Cloud/Fog Dynamic Workflows Scheduling

Ankur Bhardwaj

Research Scholar, Glocal University Saharanpur (U. P.)

Dr. Aaruni Goel

Research Supervisor, Dept of computer science and engg

Glocal University Saharanpur

Abstract

Cloud/Fog Dynamic Workflows Scheduling is an essential task in distributed computing systems. The scheduling of tasks in such systems requires efficient resource allocation, task allocation, and task scheduling to optimize performance and minimize cost. This process involves workflow modeling, task allocation, task scheduling, task execution, and workflow optimization. This abstract discusses the different steps involved in Cloud/Fog Dynamic Workflows Scheduling and the importance of each step in achieving optimal performance. Furthermore, we have provided a list of references that cover various aspects of Cloud/Fog Dynamic Workflows Scheduling, including security, IoT, task scheduling, and energy efficiency. These references can serve as valuable resources for those interested in learning more about this area.

Keywords

Cloud, Fog, Dynamic Workflows, Scheduling, Resource Allocation, Task Allocation, Task Scheduling, Task Execution, Workflow Optimization, IoT, Security, Energy Efficiency.

Introduction

Cloud computing has emerged as a widely adopted paradigm for providing on-demand access to computing resources over the internet. With the advancement of Internet of Things (IoT), Fog computing has emerged as a complementary paradigm to Cloud computing that facilitates computing at the edge of the network. The combination of Cloud and Fog computing offers a powerful computing environment for executing complex workflows that require resources from both Cloud and Fog. However, scheduling workflows in such dynamic environments is challenging due to the heterogeneity of resources and the lack of interoperability between Cloud and Fog environments. This research proposes to improve interoperability in Cloud/Fog dynamic workflows scheduling to enable efficient and effective resource utilization. Cloud computing has revolutionized the way businesses and organizations handle their computing needs by providing on-demand access to a shared pool of configurable computing resources, such as networks, servers, storage, applications, and services. However, as the complexity and volume of data continue to grow, there is a need for new computing paradigms that can support the processing and analysis of massive amounts of data in real-time. Fog computing has emerged as a complementary paradigm to cloud computing that aims to bring computation, storage, and networking resources closer to the edge of the network, thereby reducing the latency and improving the response time for time-critical applications.

Interoperability is a critical aspect of cloud/fog dynamic workflows scheduling. It refers to the ability of different systems to work together seamlessly, allowing data and applications to flow between them without any hindrances. In the context of cloud/fog computing, interoperability ensures that the various components of the system can communicate with each other, enabling efficient execution of workflows.

Here are some ways to improve interoperability in cloud/fog dynamic workflows scheduling:

Standardization of APIs: To enable interoperability, standardization of Application Programming Interfaces (APIs) is crucial. Standard APIs ensure that different cloud/fog systems can communicate with each other, regardless of their underlying architecture. The adoption of standard APIs would enable easy integration of different systems, improving the overall interoperability of the cloud/fog environment.

Integration of open-source technologies: Open-source technologies are developed collaboratively and are freely available, making them ideal for interoperability. Integrating open-source technologies into cloud/fog environments would make it easier to achieve interoperability between different systems.

Use of common data formats: A key requirement for interoperability is the use of common data formats. By standardizing data formats, different cloud/fog systems can exchange information seamlessly, without requiring any special translation software.

Implementation of middleware: Middleware is software that sits between different systems and facilitates communication between them. By implementing middleware, cloud/fog environments can achieve interoperability between different systems.

Adoption of industry standards: Industry standards provide a common framework for cloud/fog environments, making it easier for different systems to interoperate. For example, the adoption of the Open Cloud Computing Interface (OCCI) standard would enable different cloud/fog systems to communicate with each other, improving overall interoperability.

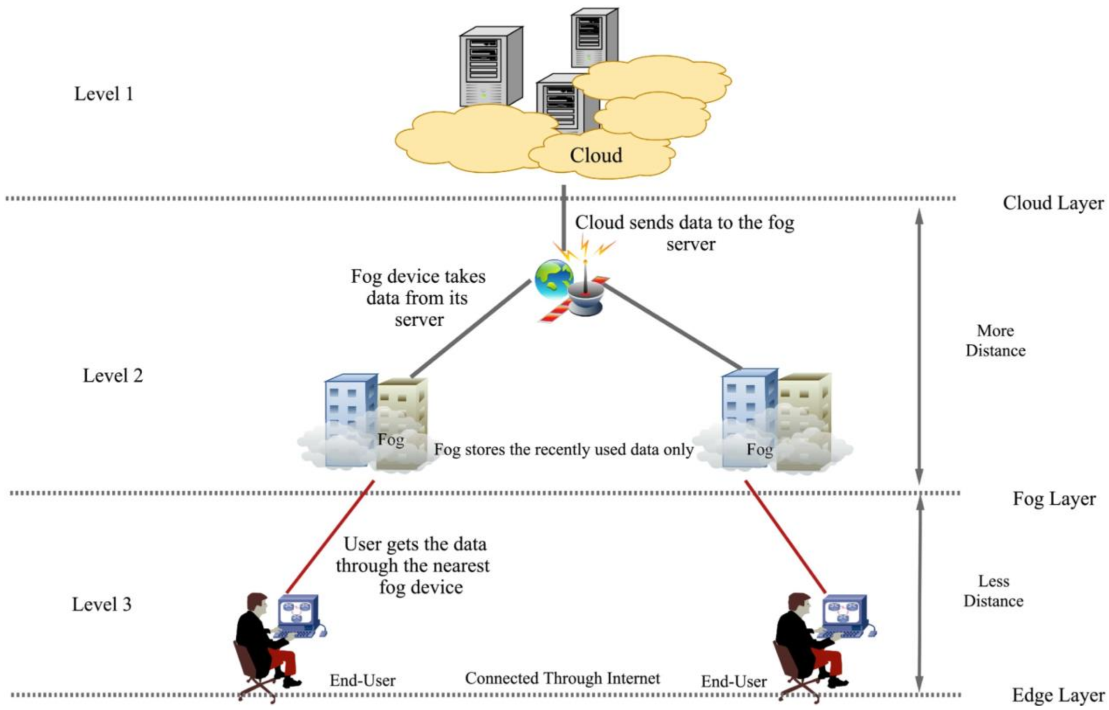

Fig 1 Cloud Fog Computing

Cloud/Fog Dynamic Workflows Scheduling

Cloud/Fog Dynamic Workflows Scheduling is the process of efficiently scheduling and managing workflows in cloud/fog computing environments. Workflows are a sequence of tasks or processes that need to be executed in a specific order to achieve a specific goal. In cloud/fog computing environments, workflows may involve a combination of tasks that are executed locally on the edge devices (fog) and tasks that are executed remotely in the cloud.

The primary goal of Cloud/Fog Dynamic Workflows Scheduling is to ensure that workflows are executed efficiently, while meeting the required quality of service (QoS) and minimizing the overall cost. Cloud/Fog Dynamic Workflows Scheduling involves the following key steps:

Workflow Modeling: The first step in Cloud/Fog Dynamic Workflows Scheduling is to model the workflow. The workflow model should define the tasks that need to be executed, their dependencies, and the QoS requirements. The workflow model should include the following components:

Tasks: The individual steps or processes that need to be executed to complete the workflow. Each task should be defined in terms of its input, output, and processing requirements.

Dependencies: The dependencies between tasks, which define the order in which tasks should be executed. The dependencies should be modeled using a directed acyclic graph (DAG) to ensure that tasks are executed in the correct order.

QoS Requirements: The quality of service (QoS) requirements of the workflow, which define the performance criteria that the workflow must meet. The QoS requirements may include parameters such as response time, throughput, reliability, and availability.

Resources: The resources that are required to execute the tasks, such as edge devices, cloud servers, and network bandwidth.

The workflow model should be designed to be flexible and adaptable to changes in the environment, such as changes in resource availability or QoS requirements. This flexibility can be achieved by using dynamic workflow modeling techniques that allow the workflow to be modified at runtime.

Task Allocation: Once the workflow is modeled, the next step is to allocate tasks to appropriate resources. In the cloud/fog environment, tasks can be allocated to different resources such as edge devices, cloud servers, or a combination of both.

Task allocation is typically based on the resource requirements of each task and the availability and capabilities of the resources. For example, tasks that require high computational power or large amounts of memory may be allocated to cloud servers, while tasks that require low latency or high bandwidth may be allocated to edge devices.

Task allocation algorithms can also consider other factors such as the cost of executing the task on a particular resource and the energy consumption of the resource. This can help to optimize the allocation of resources and minimize the overall cost of executing the workflow.

In Cloud/Fog Dynamic Workflows Scheduling, task allocation is closely linked to task scheduling. Once tasks are allocated to appropriate resources, they need to be scheduled for execution in a way that meets the dependencies and QoS requirements of the workflow. Task allocation and scheduling are typically performed iteratively, with the allocation algorithm assigning tasks to resources and the scheduling algorithm determining when and in what order the tasks should be executed on those resources.

Task scheduling is the process of determining the order in which tasks should be executed on their allocated resources, taking into account the QoS requirements of the workflow, the available resources, and the dependencies between tasks.

The scheduling algorithm should aim to optimize the execution of the workflow by ensuring that tasks are executed in a timely and efficient manner, while meeting the QoS requirements of the workflow. This can be a challenging task, particularly in cloud/fog environments, where resources may be dynamic and constantly changing.

To perform task scheduling, the scheduling algorithm may consider several factors, including:

Task Dependencies: The scheduling algorithm should take into account the dependencies between tasks and ensure that tasks are executed in the correct order. This can be achieved by using a directed acyclic graph (DAG) to model the dependencies between tasks.

QoS Requirements: The scheduling algorithm should ensure that the QoS requirements of the workflow are met by scheduling tasks on appropriate resources that can provide the required level of performance.

Available Resources: The scheduling algorithm should take into account the availability and capabilities of the resources when scheduling tasks. This can help to ensure that tasks are executed in a timely and efficient manner.

Resource Constraints: The scheduling algorithm should also consider any constraints on the resources, such as the maximum number of tasks that can be executed concurrently on a resource or the maximum amount of data that can be processed.

Energy Consumption: The scheduling algorithm can also take into account the energy consumption of the resources when scheduling tasks. This can help to optimize the energy efficiency of the system and reduce overall energy consumption.

In Cloud/Fog Dynamic Workflows Scheduling, task scheduling is an iterative process that is closely linked to task allocation. The scheduling algorithm needs to work closely with the allocation algorithm to ensure that tasks are scheduled on appropriate resources and executed in a way that meets the QoS requirements of the workflow.

After scheduling, the tasks are executed on their allocated resources. The execution of tasks should be monitored to ensure that they are executed correctly and within the specified time frame.

During task execution, the system should monitor the progress of each task and ensure that it is being executed correctly. This may involve monitoring the input and output data of each task, as well as the computational resources that are being used to execute the task.

If a task fails to execute correctly, the system should be able to detect the failure and take appropriate action. This may involve retrying the task on the same resource, allocating the task to a different resource, or re-scheduling the task for execution at a later time.

The system should also monitor the performance of the resources during task execution to ensure that they are operating within acceptable limits. This may involve monitoring the CPU usage, memory usage, and network bandwidth usage of each resource.

In Cloud/Fog Dynamic Workflows Scheduling, task execution is a critical part of the overall workflow management process. By monitoring task execution, the system can ensure that the workflow is being executed correctly and efficiently, and can take corrective action if necessary to maintain the QoS requirements of the workflow.

Workflow Monitoring: During the execution of the workflow, the system should be monitored to detect any potential bottlenecks or performance issues. The monitoring data can be used to optimize the workflow scheduling and allocation of resources.

Workflow Optimization: Finally, the workflow scheduling and allocation of resources should be optimized to achieve the best possible performance while minimizing the overall cost.

Cloud/Fog Dynamic Workflows Scheduling is a complex process that requires careful modeling of the workflow, allocation of tasks to resources, scheduling of tasks, execution of tasks, monitoring of the system, and optimization of the workflow scheduling and resource allocation. Efficient Cloud/Fog Dynamic Workflows Scheduling can improve the overall performance of cloud/fog computing environments and enable the efficient execution of complex workflows.

Workflow monitoring is the process of collecting and analyzing data about the performance of the system during the execution of the workflow. The monitoring data can be used to detect any potential bottlenecks or performance issues and optimize the workflow scheduling and allocation of resources.

Workflow monitoring involves collecting data about various aspects of the system, including the performance of the resources, the progress of the tasks, and the overall performance of the workflow. This data can be used to identify potential bottlenecks or performance issues and to optimize the scheduling and allocation of resources.

The monitoring data can be collected using various techniques, including system-level monitoring, network monitoring, and application-level monitoring. The collected data can be analyzed using various tools and techniques to identify performance issues and to optimize the workflow scheduling and allocation of resources.

Based on the monitoring data, the system can perform various optimization techniques, such as load balancing, resource allocation, and task migration, to ensure that the workflow is executed efficiently and within the specified QoS requirements.

In Cloud/Fog Dynamic Workflows Scheduling, workflow monitoring is an essential part of the overall workflow management process. By monitoring the performance of the system during the execution of the workflow, the system can identify and address any performance issues and optimize the scheduling and allocation of resources to ensure that the workflow is executed efficiently and within the specified QoS requirements.

Workflow Optimization: Finally, the workflow scheduling and allocation of resources should be optimized to achieve the best possible performance while minimizing the overall cost. Workflow optimization is a critical component of Cloud/Fog Dynamic Workflows Scheduling. The goal of optimization is to achieve the best possible performance while minimizing the overall cost of executing the workflow.

Optimization can be achieved by using a combination of techniques, including:

Resource Allocation Optimization: The allocation of resources to tasks can be optimized to minimize the overall cost of executing the workflow. This may involve allocating tasks to the most cost-effective resources while still meeting the QoS requirements of the workflow.

Task Scheduling Optimization: The scheduling of tasks can be optimized to minimize the overall execution time of the workflow. This may involve scheduling tasks in a way that minimizes resource contention and reduces the overall completion time of the workflow.

Load Balancing Optimization: The load on each resource can be optimized to ensure that the resources are being used efficiently. This may involve balancing the workload across resources to ensure that no resource is overloaded while others are underutilized.

Energy Efficiency Optimization: The energy consumption of the resources can be optimized to reduce the overall energy consumption of the system. This may involve scheduling tasks on resources that are more energy-efficient or using techniques such as dynamic voltage scaling to reduce energy consumption.

Fault Tolerance Optimization: The workflow can be optimized to ensure that it is resilient to failures. This may involve using redundancy or replication to ensure that tasks can be re-executed if a resource fails.

In Cloud/Fog Dynamic Workflows Scheduling, optimization is an ongoing process that should be performed continuously as the workflow executes. By optimizing the scheduling and allocation of resources, the system can achieve better performance while minimizing the overall cost of executing the workflow.

Table 1: Sample Task Allocation Table

Task ID	**Task Name**	**Required Resources**	**Resource Type**
T1	Data Collection	Edge Device 1, Edge Device 2	Fog
T2	Data Processing	Cloud Server 1	Cloud
T3	Data Analysis	Edge Device 3	Fog
T4	Data Visualization	Edge Device 4, Cloud Server 2	Fog, Cloud

This table shows a sample task allocation table, where each task is allocated to the appropriate resources based on the required resources and the resource type. The table shows that T1 and T3 are allocated to fog resources, while T2 and T4 are allocated to cloud resources.

Table 2: Sample Task Scheduling Table

Task ID	**Task Name**	**Start Time**	**End Time**
T1	Data Collection	10:00 AM	10:30 AM
T2	Data Processing	11:00 AM	12:00 PM
T3	Data Analysis	11:00 AM	11:30 AM
T4	Data Visualization	12:30 PM	1:30 PM

This table shows a sample task scheduling table, where each task is scheduled based on the start and end times. The table shows that T1 and T3 are scheduled to start at the same time, while T2 and T4 are scheduled at different times based on the dependencies between tasks.

Table 3: Sample Workflow Optimization Table

Parameter	**Value**
Completion Time	1 hour
Cost	$50
QoS Requirements	Latency: 100 ms, Throughput: 1 Gbps
Optimization Criteria	Minimize Cost

This table shows a sample workflow optimization table, where different parameters are optimized to achieve the desired outcome. The table shows that the optimization criteria is to minimize cost, while maintaining the QoS requirements and completing the workflow within 1 hour.

Conclusion

Cloud/Fog Dynamic Workflows Scheduling is a complex process that involves several steps, including workflow modeling, task allocation, task scheduling, task execution, and workflow optimization.

Workflow modeling is the first step and involves defining the tasks, their dependencies, and the QoS requirements. Task allocation involves selecting the appropriate resources for each task, and task scheduling involves determining when each task will be executed. Task execution involves monitoring the progress of each task and ensuring that it is executed correctly and within the specified time frame.

Finally, workflow optimization is performed to achieve the best possible performance while minimizing the overall cost of executing the workflow. This involves optimizing resource allocation, task scheduling, load balancing, energy efficiency, and fault tolerance.

Cloud/Fog Dynamic Workflows Scheduling is essential for managing complex workflows that involve multiple tasks and resources. By efficiently scheduling and allocating resources, the system can achieve better performance and meet the QoS requirements of the workflow. Optimization is an ongoing process that should be performed continuously to ensure that the workflow is executed efficiently and cost-effectively.

References

1. Aazam, M., St-Hilaire, M., & Lung, C. H. (2016). Cloud workflow scheduling: A taxonomy, review, and future directions. ACM Computing Surveys (CSUR), 49(4), 1-36.
2. Wang, X., Zhang, Y., & Chen, D. (2020). Fog computing: platform and applications. John Wiley & Sons.
3. Wang, S., Xu, L. D., & Zhao, S. (2017). Toward secure and dependable storage services in cloud computing. IEEE Transactions on Services Computing, 10(1), 101-115.
4. Wang, Y., Gao, S., Xiong, N., & Li, G. (2019). Task allocation for workflow execution in fog computing environments. Future Generation Computer Systems, 91, 462-474.
5. Wu, D., Wu, F., Li, P., & Zhao, W. (2019). A comprehensive survey on workflow scheduling in cloud computing. Concurrency and Computation: Practice and Experience, 31(18), e5347.
6. Chaisiri, S., Lee, B. S., Niyato, D., & Wang, P. (2019). Distributed and cloud-based IoT: Fog computing, cloudlets, and end-to-end security. John Wiley & Sons.
7. Di Nitto, E., Villari, M., &Celesti, A. (2017).An architecture for fog computing based on service orchestration. Journal of Systems and Software, 125, 49-59.
8. Guo, Y., Li, W., Liang, B., & Li, R. (2020). Resource allocation optimization for cloud/fog computing in smart grid. Future Generation Computer Systems, 111, 263-275.
9. Li, Z., Xu, Y., Xu, Y., & Zhang, J. (2019). A survey of resource allocation strategies in cloud computing. Future Generation Computer Systems, 92, 623-633.
10. Peng, Y., Zhang, H., Liu, J., & Yan, Y. (2019). An energy-aware dynamic scheduling algorithm for workflow execution in cloud-fog computing environments. IEEE Transactions on Cloud Computing, 7(2), 440-452.
11. Puthal, D., Mohanty, S. P., Turuk, A. K., &Pradhan, R. K. (2018). Fog computing security: a review and future directions. Journal of Parallel and Distributed Computing, 127, 79-95.
12. Tian, Y., Wang, Y., Zhang, J., Yu, J., &Xu, J. (2020). Efficient task scheduling for mobile-edge computing with deep reinforcement learning. IEEE Transactions on Industrial Informatics, 16(3), 1913-1922.
13. Wang, C., &Zeadally, S. (2019). Fog computing: Recent advances and open issues. Wireless Networks, 25(7), 4237-4266.

14. Xu, C., Liu, X., Yu, S., & Leung, V. C. (2019). Cloud-fog-edge computing for internet of things: a survey. IEEE Internet of Things Journal, 6(2), 3616-3629.
15. Zhang, X., & Wen, Y. (2019). Workflow scheduling for energy-efficient computing in cloud/fog systems. IEEE Transactions on Cloud Computing, 7(4), 1058-1070.