CHAPTER 1
INTRODUCTION

1.1. BACKGROUND

Cloud computing is the rescue of computing and storage capacity as a support to a group of people or end-users. The term is derived from the exploits of a cloud-shaped figure as a concept for the different communications that each node holds in structural figures. Cloud computing guarantees service support with the user's data, software and computation over a network.

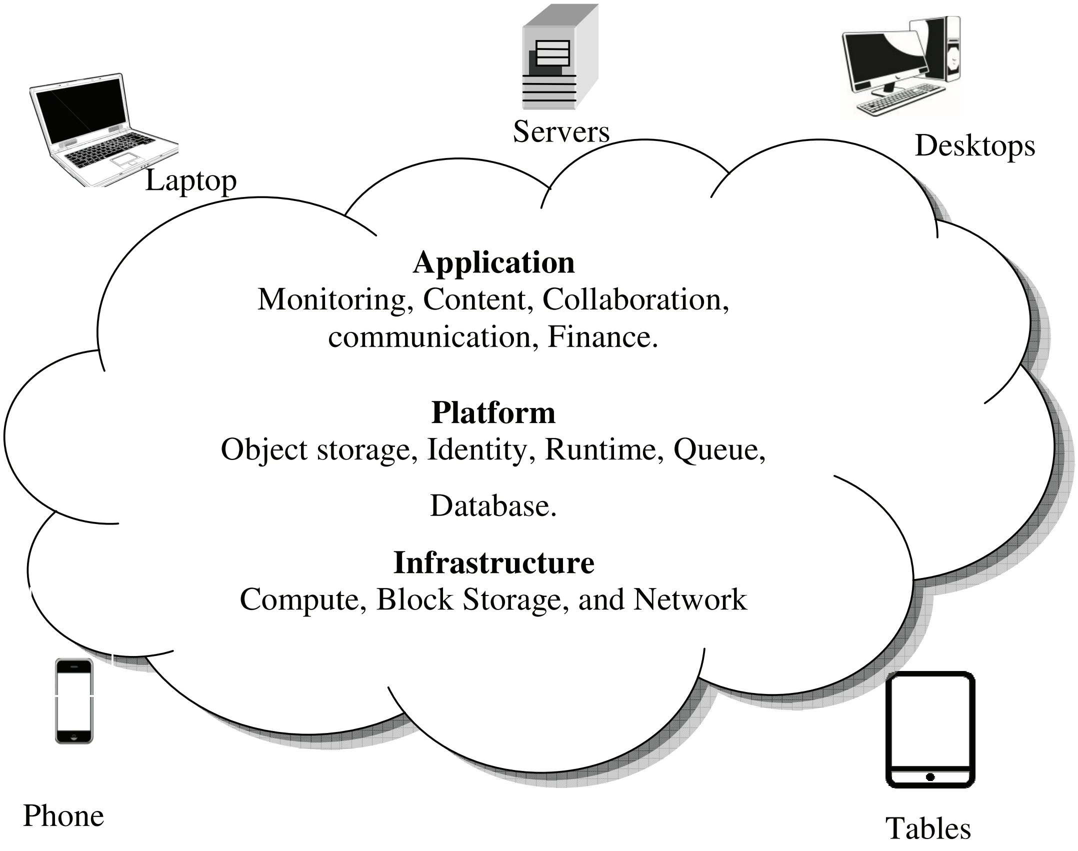
Servers
Desktops
Laptop
Application
Monitoring, Content, Collaboration, communication, Finance.
Platform
Object storage, Identity, Runtime, Queue, Database.
Infrastructure
Compute, Block Storage, and Network
Phone
Tables

Fig. 1.1 Cloud Computing Paradigm

The standard model in cloud computing infrastructure is depicted in Fig. 1.1 in which, the layers, namely application, platform and infrastructure act as a significant fact to bond the equipment, like laptop, desktop, phone and tablets. Mainly, there are three main types of cloud computing, namely Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).

Through Software as a Service, clients also hire application software and databases for rent. The cloud service providers handle the infrastructure and platforms, on which the application is executed. Proponents exploit cloud-based applications by means of a web browser or a light-weight desktop or mobile application, whereas the sensitive data storage software and users data are loaded on to the servers at an isolated place. End users maintain that cloud computing permits organizations to get their applications up and management faster, with enhanced manageability and less preservation. Moreover, cloud system allows IT to quicken the fine-tune resources so as to convene unpredictable and changeable enterprise demand.

Cloud computing depends on sharing of resources in order to accomplish consistency and finance of amount alike to a service, like either the electricity grid over a network or the Internet. At the basis of cloud computing is the wider thoughts of united infrastructure and shared services.

Most of the proficient equivalent data processing is achieved by using Nephele's agenda or by dynamic resource allocation, provided by current IaaS clouds for both, job scheduling and implementation. More specific tasks of a processing job can be allocated to various kinds of virtual machines or servers, which are often instantiated and triggered during task execution. Based on the processing job, researchers do investigations on a wider analysis of motivated processing jobs or on an IaaS cloud system. However, most of the existing systems face limits in terms of expenses, complexity and increase in data based organization.

1.2. RESOURCE ALLOCATION IN CLOUD ENVIRONMENT

Resource allocation is an essential and constantly developing aspect of many cloud computing and data center management troubles. For instance, a cloud service offers frequently allocated servers to boarder Virtual Machines (VM) based on CPU, memory space and disk availability and according to the needs of the VMs. On later stages, the service provider improves the standard and allocates the network bandwidth resources also to the already assigned boarder VMs. Even later, the service offers to establish a fresh fault-tolerant reproduction strategy, placing the VMS and data replicas cleverly across the fault domains. At this stage, the VM allocation plan relies on the status that involves a unique server capacity, network bandwidth capacity in the data center, as well as fault-domain characterizations.

Such emerging and developing resource allocation requirements are inbuilt in addition to the multi-tenant data centers. Facility development for cloud services, VM assignment in confidential data centers, network virtualization and virtual network embedding, multi-path routing, and data copy handling are the major use of resource allocation components. Mainly, resource allocation engaged partitioning and allocating resources which focus on definite constraints, such as ensured server performance, network performance, and fault tolerance needs. Most of the resource allocation issues are NP-hard variants of the recognizable bin-packing crisis. The main aim of NP-hard variants is to robust a set of balls into a given set of bins, while fulfilling the constraints which are definitely based upon the special features of the balls and bins.

1.2.1. Dynamic Resource Allocation in Cloud Infrastructure

In recent times, ad-hoc parallel data processing has developed to be one of the executioner applications of IaaS cloud. The important Cloud Computing enterprises are in progress to incorporate standards for parallel data processing in their artifact collection and to make it trouble-free for clients so as to access these cloud services and to arrange their programmes. But, the existing processing standards are designed for stationary cloud nodes, as well as for consistent cluster setups and they ignore the specific feature of a cloud. As a result, the assigned resources may be insufficient for large parts of the requested job, without cause increase, processing time and computational cost. Nephele is the foremost data processing standard to openly use the dynamic resource allocation, supported by recent IaaS cloud for, both job allocation and execution.

At present, an emerging amount of enterprises have to process large amounts of data in a cost-efficient way. Typical agents for these enterprises are workers of Internet search engines, like Google, Yahoo, or Microsoft. The large range of data they have to manage every day have made fixed database solutions which are excessively cost. As an alternative, these enterprises have a more famous architectural design, based on a large range of service VMs. Issues like processing edged data or redeveloping a web file are divided into a few self-sufficient subtasks, shared among the available nodes, and computed in equivalence.

The VMs are characteristically provided in various kinds; each type has its own features, like number of CPU cores, amount of main memory and cost. The VM idea of IaaS cloud robusts the design structure assumed by the data processing standards, like Hadoop in the cloud. Only in recent times, Amazon has fused Hadoop as one of its main infrastructure services. But, as an alternative of assumption, its dynamic resource allocation and current data processing standard relatively to the cloud to copy the static feature of the cluster environments have actually been designed. For instance, at an instant the kinds and numbers of the VMs allocated at the start of a forecast task cannot be modified during the period of processing, as the tasks include the whole special demands of the environment. Consecutively, the charged resources may be insufficient for large parts of the processing task, which may worse the complete processing performance and increase the cost.

A dynamic data processing framework for cloud environment has been elaborated. An improved Nephele takes up many thoughts of the existing processing ideas but redefines them into a better competition of dynamic and opaque characteristics of a cloud. The design paradigm of the dynamic data processing in the structural overview of Nephele running in an IaaS cloud is shown in fig. 1.2.

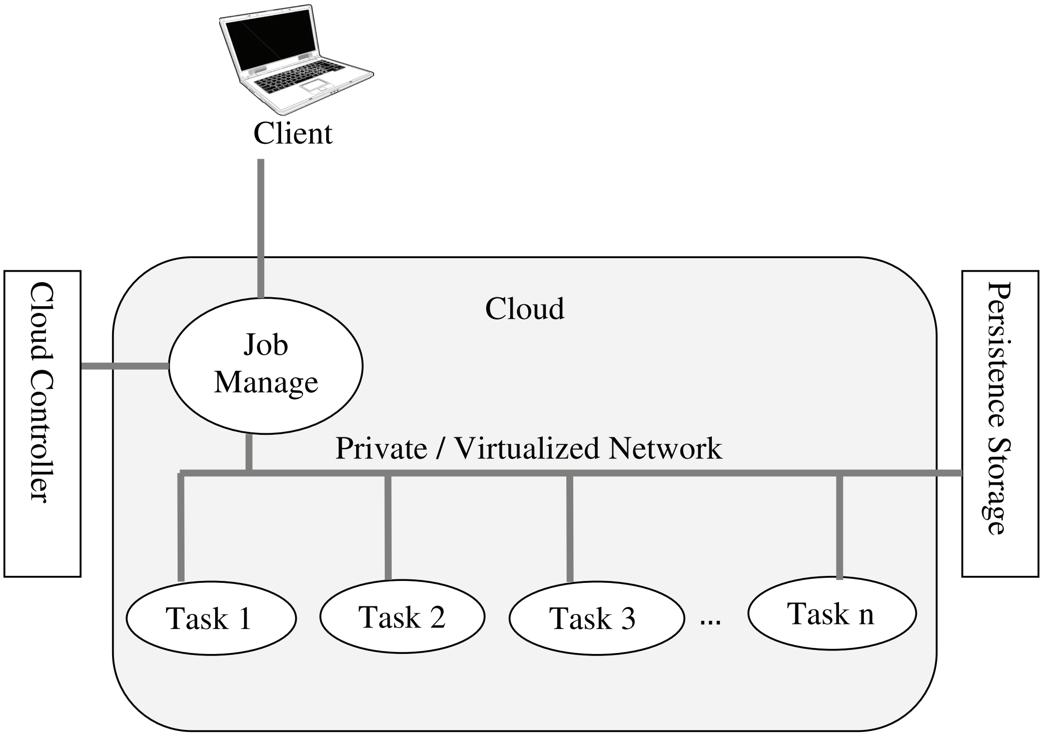

Fig. 1.2 Design Paradigm of Dynamic Data Processing in Structural Overview of Nephele

Facing a Nephele compute task, a user must begin a VM in the cloud which executes the post of the Job Manager. Mainly, the Job Manager collects the client's task which is dependable for scheduling jobs and manages their implementation. Moreover, the job Manager is competent in communicating with his intermediates as the cloud workers offer to handle the instantiation of the VMs. The management of the instantiation of the VMs is termed as the interface of the Cloud Controller. Through the Cloud Controller, the Job Manager can allocate or deallocate the VMs, depending on the modern job execution phase.

The original run of jobs i.e., a Nephele job includes a group of instances. Each instance is executed by a Task Manager. The Task Manager accepts one or more tasks from the Job Manager at a time, runs them, and after that reports to the Job Manager about their conclusion or possible faults. But when a task is requested to the Job Manager, the system anticipates the group of instances and hence the set of Task Managers is to be unfilled. On receiving the task response, the Job Manager then chooses, based on the job's specific tasks, how many and what type of instances the job should be executed on, and when the particular instances must be allocated/deallocated to guarantee an uninterrupted, but cost-efficient processing.

The freshly allocated instances are initiated with an earlier compiled VM image. The image is arranged to routinely establish a Task Manager and schedules it with the Job Manager. After all the essential Task Managers have effectively linked to the Job Manager, it stops the execution of the scheduled job.

Originally, the VM images are used to initiate the Task Managers who are empty and do not hold any of the data on which the Nephele task is believed to function on. Consequently, the systems expect the cloud to support the persistent storage, like e.g. Amazon S3. This persistent storage is supposed to load the job's input data and ultimately accept its output data. The operator must be reachable for both the Job Manager as well as for the set of Task Managers, even if they are linked by a private or virtual network.

Finally, the dynamic job execution and scheduling provide high ability to allocate the specific VM types to explicit the tasks of a processing job, as well as the prospect to routinely allocate/ deallocate the virtual machines in the course of a job implementation. Moreover, they facilitate to enhance the complete resource utilization and there by reduce the processing cost.

1.2.2. Congestion Control Method in Resource allocation at Cloud

As cloud computing services speedily increase their customer support, it has become significant to allocate cloud resources, so as to afford the cost effectively. In cloud computing services, numerous kinds of resources, such as the processing ability, bandwidth and storage, need to be shared simultaneously. If there is a course of requests, a fight will occur between these requests for the use of the cloud resources. The fight between the requests, results in the interference of the service and it is essential to evaluate, to discard or to lighten the blocking of cloud computing environments. A congestion control method for cloud computing environments is developed with the intention of reducing the amount of the required resources for congested resource type as a substitute of restricting all the service requests as in the existing networks. Moreover, the congestion control emerges from the client service notations for blocking the interference happenings in-between the communication. And also verifies the algorithm to choose the optimal amount of the required resources to be minimized, depending on the load supported by the system. Congestion control method is expressed by imitation evaluations so that the proposed method can manage more requests compared to the conventional methods and there by alleviate the congestion.

1.2.2.1. Cloud Resource Allocation Model

The resource allocation model for a cloud computing environment is such that several resources taken from a general resource group are allocated, concurrently to each request for a convinced period of time. In congestion control method, two resource types, namely processing ability and bandwidth are mainly considered for the beginning evaluation. In addition, congestion control method considers the objective services for supporting the cloud computing services that are shared over multiple centers in order to make it trouble-free, to boost the number of the services when order increases, to allocate load balancing, and to improve its reliability. The cloud resource allocation model, that adopts these assumptions, is illustrated in Fig. 1.3.

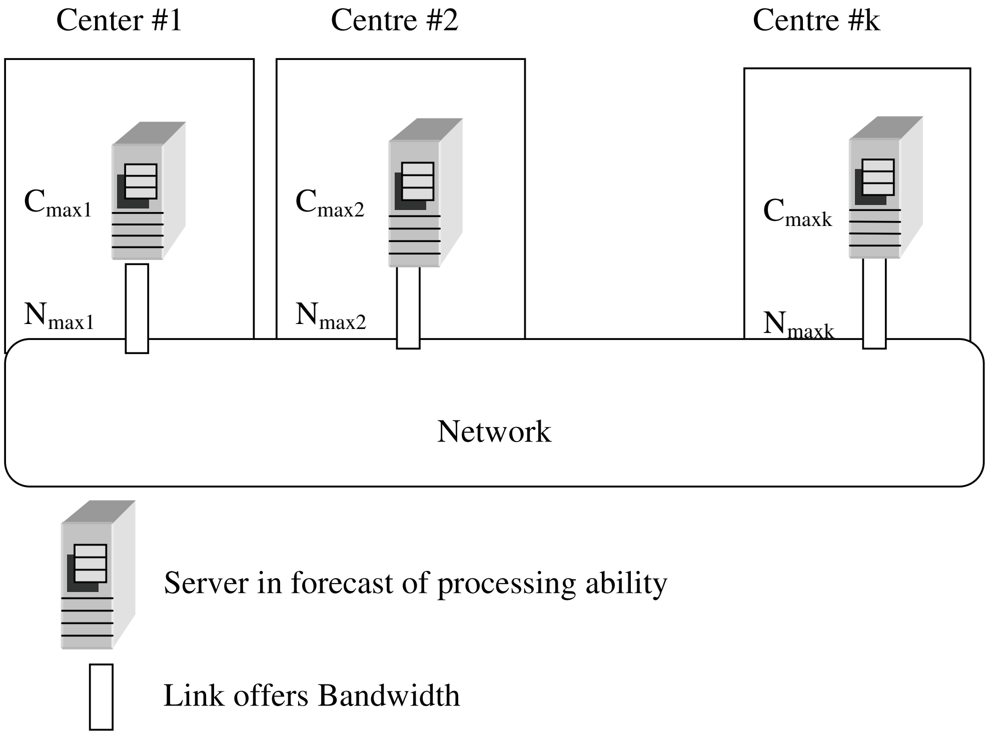

Fig. 1.3 System Model for Cloud Computing Services

Each center has servers, including virtual servers, which forecasts the processing ability. Further, the centre includes network devices, which offer the bandwidth to access the servers. The maximum size of the processing ability and the bandwidth at the center are measured and denoted as C_{maxj} and N_{maxj} respectively. When a service request is created, one of the best centers is chosen from the k centers, and the processing ability and the bandwidth in that center are assigned concurrently to the appeal for a specific period of time. If no center has adequate resources for a new request, the request will be discarded.

Finally, in the congestion control method for cloud computing infrastructure, both the processing ability and the bandwidth are simultaneously allocated. Allocation of the processing ability and the bandwidth minimize the range of the required resources for the congested resource type instead of restricting all the service requests. Congestion control method emerges as a user service specification, that verifies the algorithm to choose the best range of the required resources to be reduced, based on the load provided to the system. However, congestion control method is risky when it is executed by a large number of users, resources and centers.

1.3. ENERGY CONSUMPTION IN CLOUD ENVIRONMENT

Wide extent of data centers becomes common in the computing industry with the benefits of cloud computing. Moreover, a significant increase in energy consumption at these data centers causes a major problem, which is to be addressed. For most of the period, a data center takes rest. Therefore, a large range of energy is consumed by the shifting virtual VMs from unused machines to other active running machines and covering such unused machines. Researchers investigate the issues related to energy utilization, based on which a design is made such that it provides an energy-efficient cloud data centers. The design exploits chronological traffic data from the data centers and extracts the needs from the service request prediction model which allows the prediction of the number of active servers required at a given moment. Hence, the design leads to the possible covering which is left unused by the VMs. A deep analysis justifies a better design functioning of the system that brings out a significant amount of energy consumption.

The design model includes service request identification and also it manages VM relocation. The relocation of VMs is based on a principle which aspires to guarantee that the SLA destruction rate is minimal. In order to achieve the maximum energy consumption, insignificant changes are made in the present cloud infrastructure. The proposed cloud infrastructure, as shown in Fig 1.4, involves of a Cluster Manager, Storage Manager, Cloud Manager, and a large amount of Node Managers/Servers which are competent of executing the VMs.

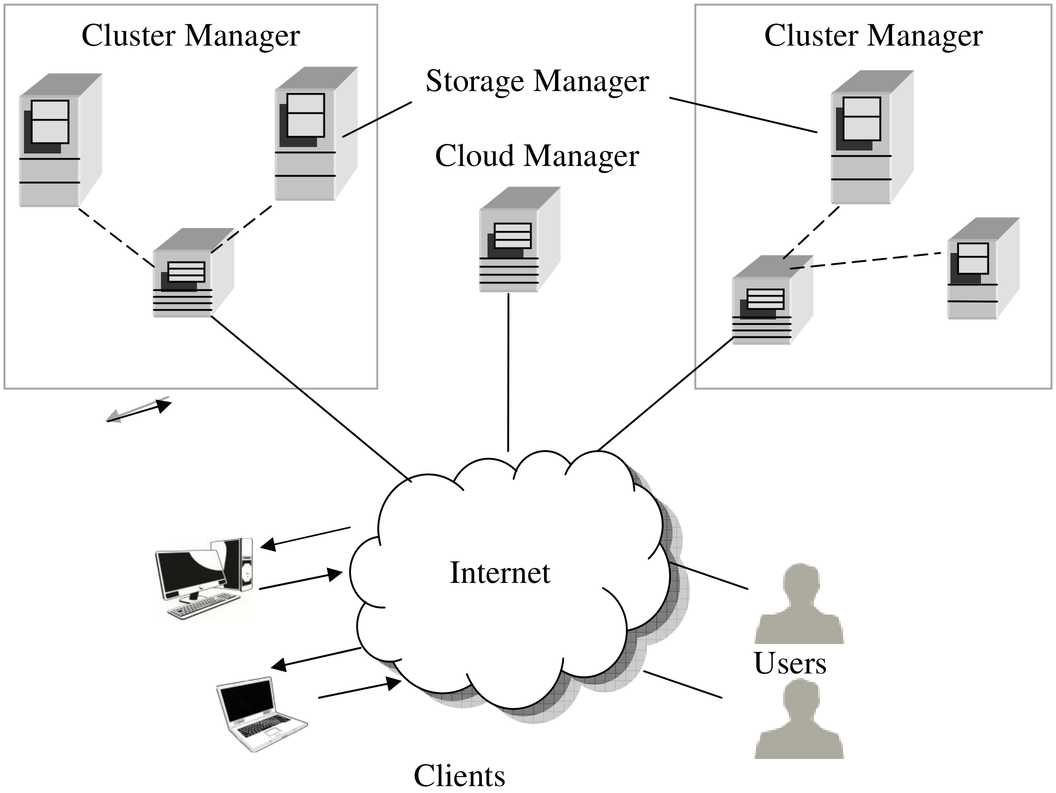

Fig. 1.4 System Architecture of Cloud with Cluster Manager

The Cluster Manager handles the executions of the VMs, operating on the cloud nodes and controls the virtual networking between the VMs and the external users. The operation of the Storage Manager is to support the central storage in a cluster that can be actively attached to the VMs. The Node Manager handles the VM functions, including the implementation, examination and execution of VM instances. A vital statement of this design is that the Cloud Manager, Cluster Manager, Storage Manager and Node Manager are placed at the same geographic locality.

The service request identification model operates on the chronological request data. In a general state, the complete cloud platform is widened across the world so that the VMs can be relocated across the clusters, even in the different continents or within a cluster, based on the SLAs and relocation issues. The relocation problems hold the expense of migration operation, changeable tariff ideas, real-time/non-real-time services, and SLA metrics. Moreover, use of a service request identification model and relocation of the VMs from one system to another system requires guarantee placed SLAs in order to reach a very high chance in energy consumption.

A service request identification model is designed to decide the pre-mentioned period of time when the server cluster will possibly be unused. The workload forecast is based on the chronological usage data. The service request identification should be carried out to determine the potential periodic or extremely serious periodic workloads that may modify the hidden decisions as in some cases the unhidden decisions are better. As the system needs to consider the time needed to re-activate a cover server when needed, which may cause a delay or need of accessibility some SLAs are being overlooked. The service request operation functions with a changeable number of servers, and can change when the servers are appended to or removed from the system. The Cluster Manager identifies the workload too, by discovering the patterns from the chronological service request data.

This energy consumption model tends to offer better energy conservation with less SLA breach, by making the service request identification model accurate and relocating the VMs across the cluster. The design offers the same energy consumption compared to the other existing works, but with only 2% SLA breach rate and with the same percent of extra servers. The benefit of energy consumption model leads to increase in the existence of infrastructure and critical down outfitted costs, of which energy is conceivably the primary element of cloud systems.

1.3.1. Secure Collaborative Integrity Verification

A hybrid cloud is a cloud computing environment in which a group offers and handles some interior resources and has others supporting them superficially. But, further enhancement to cloud environment can fetch irreversible errors to the users, due to the deficience of integrity proof method for shared data outsourcing. In order to provide a scalable service and data relocation, the design phase includes a shared integrity proof model in hybrid clouds. The collaborative shared integrity model considers the subsistence of multiple cloud service providers to collaboratively load and preserve the user data. The establishment of collaborative provable data possession scheme employs the concepts of homomorphic certifiable responses and hash index hierarchy. In addition, the communicative action optimization designs the collaborative shared integrity model and justifies the protection based on multi-prover zero-knowledge proof system. The multi-prover zero-knowledge proof assures features of unity, knowledge reliability, and zero-knowledge.

Although the provable data possession schemes evolve around open clouds, they facilitate an openly available remote interface to ensure and run a large amount of data, as the common practices of recent provable data possession schemes are unable to fulfill, such an intrinsic requirement of hybrid clouds in the features of protection, bandwidth and usability. In order to solve this problem, the researchers have developed a hybrid cloud storage service with collaborative provable data possession as illustrated in Fig. 1.5.

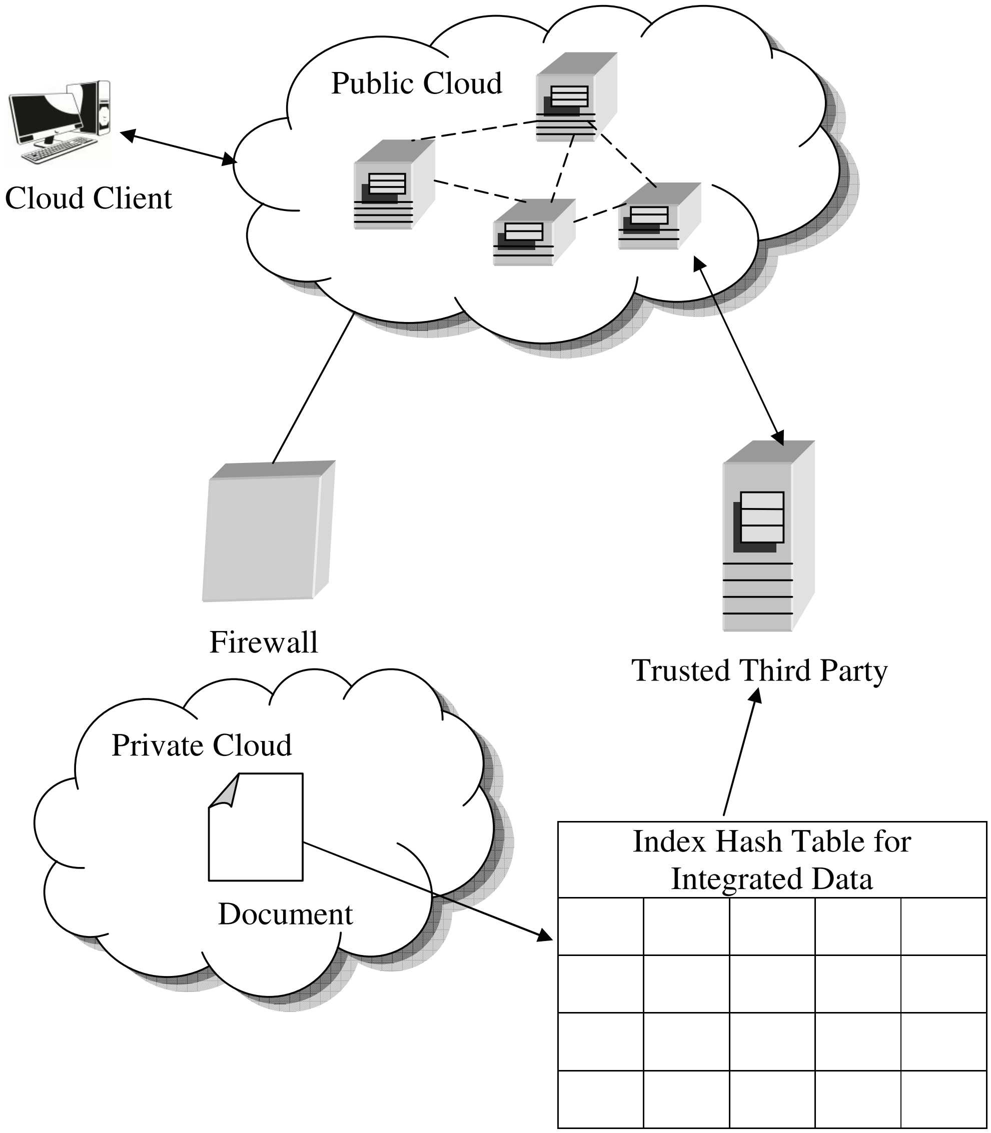

Fig. 1.5 Verification Architecture for Data Integrity in Hybrid Clouds

In Fig. 1.5, an index hash table is presented to run the application data, which is integrated with the multiple cloud service providers in a hybrid cloud. In addition, some basic data items, such as data block position, access domain and hash value should be included to this hash table. In CPDPS scheme, one of the most significant aspects is a cryptographic hash value, is used to reduce the proof itself and provides data integrity confirmation in collaborative PDP services. More prominently, this hash table is also used to resolve the heterogeneous storage difficulty.

In order to support this CPDP framework, the cloud storage provider also requires, appending equivalent modules to execute the collaborative services. For instance, Open Nebula is an open source, practical communications manager that is combined with multiple virtual machine controller, transfer managers, and outside cloud providers. In a cloud computing platform, based on Open- Nebula architecture, a service module of collaborative PDP is added included in the cloud computing management platform. Finally, the CPDP is proved to provide all the security features needed by the zero-knowledge interactive confirmation system, so that it can oppose the various errors, even if it is deployed as a public proof service.

1.4. MULTI-TASKING IN CLOUD PLATFORM

The scheduling of a multiple-task workflow in a shared computing platform is a familiar NP-hard problem. The problem is even more complicated and problematic when the virtualized clusters are used to implement a huge range of jobs in a cloud computing infrastructure. Complexity is faced in fulfilling the multiple objectives that may be of inconsistent character. For example, it is a complex task, to reduce the make distance of many jobs, while minimizing the resource charge and maintaining the fault tolerance and/or the quality of service (QoS) all together. These inconsistent needs and objectives are hard to optimize because of the unidentified runtime criteria, such as the accessibility of the resources and arbitrary workload sharing. As an alternative, to consider a method is developed to create the suboptimal or adequately best schedules for even multitask workflows in a cloud infrastructure.

Multi-objective Scheduling (MOS) Scheme is particularly designed for clouds and based on Ordinal Optimization (OO) Method that has been formerly emerged by the mechanization society for the model optimization of very composite active systems. In addition, it broadens the OO Scheme so as to congregate the particular demands from cloud infrastructure, which performs the essential clusters of the servers from the various data centers. The system intends in justifying the sub-optimality, through arithmetic evaluations. The most important benefit of MOS Method depends on the considerably reduced scheduling operating cost and it needs to increase its security towards optimal performance. Wide-ranging quantitative analyses have been performed on virtual clusters, with 16 to 128 virtual machines. The multi-tasking workflow is achieved from an actual scientific LIGO workload for gravel gravitational wave evaluations.

1.4.1. Multi-objective QoS Optimization Based on Multiple Workflow Scheduling

Cloud computing is an expansion of parallel computing, shared computing and grid computing. Cloud computing offers protected, rapid, suitable data storage and measuring power with the aid of internet. Cloud service provides the necessary services, based on user needs. Whenever we gather the diverse users with various QoS needs scheduling, the services will be a demanding one with various existing works for scheduling the focus on cost or time or both. Multi-objective QoS Optimization based workflow schedules the services, based on more than three QoS needs, like time cost, reliability and availability. Further, it evaluates the performance for the different test cases, with different number of workflows and different sets of QoS metrics for each workflow. The Multi-objective QoS Optimization results, which have reduced the time effect, reduces the cost effect and increases the dependability and availability in a single objective way, revealing the enhanced results than the existing methods.

In Multi-objective QoS Optimization Based Workflow Scheduling Algorithm, reliability and availability are included as the additional metrics to satisfy the customer's various needs. In addition, the work is used to schedule the workflow actively and it is used to reduce the execution time, cost as well as it increased availability and reliability. The cloud servers are preserved at two levels, namely storage server and computational server. The storage server offers the role of storage and modification. The computational server supports the work of mapping the cloud services to the user based needs. Fig 1.6 shows the architecture for the overall function of Multi-objective QoS Optimization Based Workflow Scheduling.

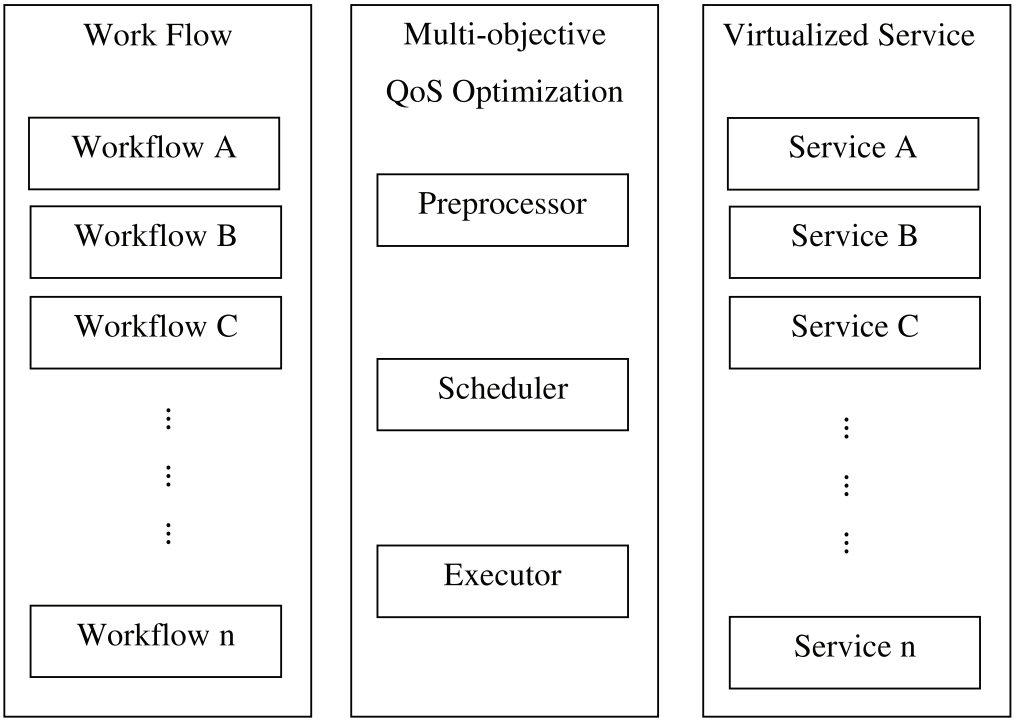

Fig. 1.6 Overview of Multi-objective QoS Optimization Based Workflow Scheduling

The Multi-objective QoS Optimization Based Workflow Scheduling strategy to arrange all the completed tasks have been based on the following considerations.

(i) The number of services in a cloud is finite, and in most cases, the number of the tasks waiting to be executed is larger than the service number. So, a task with minimum available service number should be scheduled first. The reason is that the task will not have available services, if the other tasks are scheduled first.

(ii) The tasks which belong to the workflow, with minimum time surplus and cost surplus as well as reliability surplus should be scheduled first. The reason is similar to the above.

(iii) The task with minimum covariance should be scheduled first.

(iv) The task with reliability is executed by the sorting algorithm. It is considered by the reliability of the task. The minimum reliability will be scheduled first.

Cloud Computing leads to the possibility to provide anything as a service over the Internet. The service for the different clients at the same time with different QoS needs, proves that the scheduling strategy should be established for many workflows with different QoS needs. The Multiple QoS Optimization Scheduling principle of multi-workflows achieves the users multiple QOS condition, such as execution time, implementation cost as well as the schedule of the workflows. In most of the present algorithms, the focus is only on either the cost or time, but not much attention is given to reliability and availability. However, the Multi-objective QoS Optimization Based Workflow Scheduling assures the multiple QoS time, cost as well as reliability and availability. As a result, the scheduling is carried out by the users multiple QOS requirements in a single purpose way. But, more concentration is required to fulfill the other metrics, like computational complexity and load balancing factor.

1.5. CHALLENGES AND OPPORTUNITIES

In progress data, processing standards like Google's MapReduce or Microsoft's Dryad engine are developed for cluster infrastructure. This is exposed in a range of consideration they make, which is not unavoidably suitable in cloud environments. These challenges and opportunities reveal that discarding these considerations not only gain new opportunities but also issues professional parallel data processing in clouds.

1.5.1. Opportunities

Recent, processing standards imagine that the resources they run include a standing group of homogeneous work out nodes. Even though they are designed to handle unique nodes error, they assume that the number of offered machines to be invariable, particularly when scheduling the processing job's implementation. While IaaS clouds can definitely be used to create such cluster-like formations, much of their elasticity remains underutilized.

One of the IaaS cloud's main aspects is the provisioning of the division of resources on requirements. The novel VMs can be assigned at any period or at the time of a well-defined interface and they become obtainable with in a few seconds. If the machines, are underutilized or exploited to trigger instantaneously, the cloud customer will not pay for them are more. In addition, cloud operators like, Amazon let their customers charge the VMs of various categories, the VMs with unusual computational power, diverse sizes of main memory, and storage. Therefore, the measure resources which are accessible in a cloud, are extremely active and probably heterogeneous.

With regard to parallel data processing, this elasticity results in a chance for new possibilities, specifically for scheduling data processing jobs. This scenario permits to allocate the compute resources actively and just for the interval, they are needed in the processing workflow. For instance, a structure using the possibilities of a cloud can begin with a single VM which examines an external job and then guides the cloud to openly begin the needed VMs, according to the job's dealing out stages. After each completion, the machines can be unconfined and they no longer favour the complete cost for the processing tasks.

Supporting such use cases reveals some needs on the plan of a processing framework and the way its jobs are elaborated. The foremost thing is that, the scheduler of such a structure must be converted to be responsive of the cloud environment in which a task should be implemented. The system must recognize the various categorizations of the available VMs as well as their complexities and be able to allocate or demolish them, with respect to the cloud customer.

Next, a scenario that is used to describe the tasks must be dominant, sufficient to state dependencies between the different tasks the jobs comprise. The system must be conscious of the fact that which job's output is needed as another jobs input. Or else, the scheduler of the processing structure cannot make a decision at each position in the time interval, a particular VM is no longer required and deallocated. The Map Reduce pattern is an optimal instance of an inappropriate model. Even though at the conclusion of a job, only few reducer jobs may still be working, it is hard to close the underutilized VMs, as they will because indistinct if they hold middle results, which are still necessary.

Lastly, the scheduler of such a processing structure must be able to decide, which task of a job should be run on which kind of VM and perhaps, how many of those are there. This information can either be offered superficially, for instance as a footnote to the job explanation, or deduced inside, e.g. from gathered statistics; likewise this way the database systems attempt to optimize their execution schedule over time.

1.5.2. Challenges

The cloud's virtualized features aid to facilitate for ensuring the fresh use cases for well-organized parallel data processing. But, it also exposes new issues compared to the traditional cluster formations. The major issue widely seen in cloud's environment with prospect to exploiting data locality:

In a cluster, the working nodes are normally interconnected through a substantial high-performance network. The structure of the network, i.e. the manner in which the compute nodes are actually connected to each other, is generally eminent and what is more significant is the fact that they do not vary over time. Present data processing structures supports to control this awareness about the network priority and challenge the schedule tasks on active nodes so that the data sent from one node to the other has to cross few network modes as possible. In way the network problem can be discarded and the complete throughput of the cluster can be enhanced.

In a cloud, this infrastructure formation is generally not revealed to the users, as the nodes involved in the dispensation of a data concentrated work often have to move large amounts of data through the network. This limitation is mainly harsh ie, fractions of the network may become blocked, while others will be basically idle, even though there have been studies on possible network frameworks exclusively from end-to- end forecasts. Moreover, an indistinguishable fact arises if these techniques are related to IaaS clouds. For safety causes, clouds frequently include network virtualization techniques which can obstruct the conclusion process, especially when based on latency capacity.

Although it is possible to determine the idle network priority in a cloud, use of cloud for structural aware scheduling is inappropriate, resulting in insufficient information needs leading to invalid processing time. VMs may be relocated for administrative needs between the different places inside the data center without any announcement and thereby completing any preceding information of the applicable network infrastructure outdated.

Consequently, the only manner to guarantee the position between the jobs of a processing task is to implement these jobs on the same VM in the cloud. This may engage allocating a smaller amount, but with more controlling

VMs having multiple CPU cores. For example, an aggregation job getting data from various generator jobs should be assumed. The data locality can be guarantee by scheduling these tasks to be executed on a VM, with elevation cores as an alternative of different distinct single-core machines. Nevertheless, currently no data processing framework includes such strategies in its scheduling algorithms. Therefore, the challenge lies in the design of an appropriate model to fulfill proper resource allocation, load balancing with energy consumption and handling many tasks.

1.6. PURPOSE OF THE STUDY

The main objective of the work is,

(i) To enhance the resource scheduling process, with the proposed works of multitasking based resource scheduler

(ii) To allocate the resources with optimal energy and bandwidth consumption by using Interference Aware Resource Allocation (IARA) Technique and thereby resolving the sub-optimization problems

(iii) To further enhance the energy consumption in cloud infrastructures, using Adaptive Load Balancing (ALB) Approach

(iv) To balance the load in resources through cluster formation by adopting ALB Algorithm

(v) To handle the workload management on multi-tasking with the introduction of novel Genetic Clustering with Workload Multi-task (GCWM) Scheduler Scheme

(vi) To minimize the computational cost and complexities involved during computation, using genetic concepts in GCWM Scheduler.

1.7. ORGANIZATION OF THE THESIS

Chapter 2 describes the review of literature and discusses the techniques of resource allocation in cloud infrastructure, dealing with the different datasets, limitations faced in the other existing resource allocation and load balancing methods, explanation of various drawbacks and explaining the related methods with their pros and cons in detail.

Chapter 3 initially presents a novel resource scheduling algorithm in the cloud computing environment so as to allocate the resources, based on the processing capability, electric power and network bandwidth. In addition, the method removes the surrounding interferences happening in between the transmission.

Chapter 4 explores the issues of load imbalance factor scenarios. This chapter addresses the high energy utilization problems as well as the imbalance load issues, with the development of Adaptive Load Balancing approach (ALB). ALB Approach balances the load from every cluster group by minimizing the bandwidth and energy consumption.

Chapter 5 performs multi-tasking so as to increase the performance of workload management with Genetic Clustering with Workload Multi-task

(GCWM) Scheduler Scheme. This chapter elaborates GCWM Scheduler on performing clustering with similar workload, using genetic principles.

Chapter 6 discusses the result analysis of IARA Technique, ALB Approach and GCWM Scheduler, in terms of clustering efficiency, load balance factor, energy consumption, multi-task clustering effect, execution time and computational cost.

Chapter 7 includes the key results of this research study, conclusions arrived and future research directions.