

Efficient Framework For Resource Management In Cloud Using Hybrid Fuzzy Clustering

k.Sindhuja, P.Anand Raj

Abstract— Cloud-based hosting is claimed to possess many advantages over traditional in-house hosting such as better scalability, ease of management, and cost savings. It is not difficult to understand how cloud-based hosting can be used to address some of the existing limitations and extend the capabilities of many types of applications. There are several algorithms already used for resource allocation in cloud computing. Based on analyzing these existing cloud resource allocation schemes, the new resource allocation algorithm based on hybrid fuzzy clustering will be introduced. The proposed algorithm cleverly assigns appropriate resource to the tasks that exactly satisfy its need for resource, while reserving the resources whose power greatly exceed the needs of current tasks for future use. By this way, the proposed algorithm can effectively avoid assigning powerful resources to simple and medium scale tasks or assigning poor resources to complex large-scale tasks, for they may lead to misuse of resources and failure in scheduling of tasks. The proposed algorithm is expected to have high efficiency and good robustness and will work better than other similar algorithms. The highlight of a new many-to-many stable matching algorithm that efficiently matches virtual machines with heterogeneous resource needs to servers. This work will provide a new approach for cloud based computing.

Index Terms— Cloud computing, Resource Scheduling, Hybrid Fuzzy Clustering, Virtual Machine Placement, Stable Matching.



1 INTRODUCTION

Cloud computing is the delivery of computing services over the Internet. Cloudservices allow individuals and businesses to use software and hardware that are managed by third parties at remote locations. The cloud computing model allows access to information and computer resources from anywhere that a network connection is available. Cloud computing provides a shared pool of resources, including data storage space, networks, computer processing power, and specialized corporate and user applications.

Cloud computing has a variety of characteristics, with the main ones being:

Shared Infrastructure — Uses a virtualized software model, enabling the sharing of physical services, storage, and networking capabilities. The cloud infrastructure, regardless of deployment model, seeks to make the most of the available infrastructure across a number of users.

Dynamic Provisioning — Allows for the provision of services based on current demand requirements. This is done automatically using software automation, enabling the expansion and

contraction of service capability, as needed. This dynamic scaling needs to be done while maintaining high levels of reliability and security.

Network Access — Needs to be accessed across the internet from a broad range of devices such as PCs, laptops, and mobile devices, using standards-based APIs (for example, ones based on HTTP). Deployments of services in the cloud include everything from using business applications to the latest application on the newest smartphones.

Managed Metering — Uses metering for managing and optimizing the service and to provide reporting and billing information. In this way, consumers are billed for services according to how much they have actually used during the billing period. In short, cloud computing allows for the sharing and scalable deployment of services, as needed, from almost any location, and for which the customer can be billed based on actual usage.

Once a cloud is established, how its cloud computing services are deployed in terms of business models can differ depending on requirements. The primary service models being deployed.

Software as a Service (SaaS)

Consumers purchase the ability to access and use an application or service that is hosted in the cloud. A benchmark example of this is Salesforce.com, as discussed previously, where necessary information for the interaction between the consumer and the service is hosted as part of the service in the cloud.

-
- Sindhuja is currently pursuing masters degree program in computer science and engineering in E.G.S.Pillay Engineering college, Nagapattinam. E-mail: sindhu7@gmail.com
 - Anand Raj Assistant Professor, E.G.S.Pillay Engineering college, Nagapattinam. E-mail: anand.happy123@gmail.com

Also, Microsoft is expanding its involvement in this area, and as part of the cloud computing option for Microsoft® Office 2010, its Office Web Apps are available to Office volume licensing customers and Office Web App subscriptions through its cloud-based Online Services.

Platform as a Service (PaaS)

Consumers purchase access to the platforms, enabling them to deploy their own software and applications in the cloud. The operating systems and network access are not managed by the consumer, and there might be constraints as to which applications can be deployed.

Infrastructure as a Service (IaaS)

Consumers control and manage the systems in terms of the operating systems, applications, storage, and network connectivity, but do not themselves control the cloud infrastructure. Also known are the various subsets of these models that may be related to a particular industry or market. Communications as a Service (CaaS) is one such subset model used to describe hosted IP telephony services. Along with the move to CaaS is a shift to more IP-centric communications and more SIP trunking deployments. With IP and SIP in place, it can be as easy to have the PBX in the cloud as it is to have it on the premise. In this context, CaaS could be seen as a subset of SaaS.

Deploying cloud computing can differ depending on requirements, and the following four deployment models have been identified, each with specific characteristics that support the needs of the services and users of the clouds in particular ways.

Private Cloud - The cloud infrastructure has been deployed, and is maintained and operated for a specific organization. The operation may be in-house or with a third party on the premises.

Community Cloud - The cloud infrastructure is shared among a number of organizations with similar interests and requirements. This may help limit the capital expenditure costs for its establishment as the costs are shared among the organizations. The operation may be in-house or with a third party on the premises.

Public Cloud - The cloud infrastructure is available to the public on a commercial basis by a cloud service provider. This enables a consumer to develop and deploy a service in the cloud with very little financial outlay compared to the capital expenditure requirements normally associated with other deployment options.

Hybrid Cloud - The cloud infrastructure consists of a number of clouds of any type, but the clouds have the ability through their interfaces to allow data and/or applications to be moved from one cloud to another. This can be a combination of private and public clouds that support the requirement to retain

some data in an organization, and also the need to offer services in the cloud.

The following are some of the possible benefits for those who offer cloud computing-based services and applications:

Cost Savings - Companies can reduce their capital expenditures and use operational expenditures for increasing their computing capabilities. This is a lower barrier to entry and also requires fewer in-house IT resources to provide system support.

Scalability/Flexibility - Companies can start with a small deployment and grow to a large deployment fairly rapidly, and then scale back if necessary. Also, the flexibility of cloud computing allows companies to use extra resources at peak times, enabling them to satisfy consumer demands.

Reliability - Services using multiple redundant sites can support business continuity and disaster recovery.

Maintenance - Cloud service providers do the system maintenance, and access is through APIs that do not require application installations onto PCs, thus further reducing maintenance requirements.

Mobile Accessible - Mobile workers have increased productivity due to systems accessible in an infrastructure available from anywhere.

The following are some of the notable challenges associated with cloud computing, and although some of these may cause a slowdown when delivering more services in the cloud, most also can provide opportunities, if resolved with due care and attention in the planning stages.

Security and Privacy - Perhaps two of the more "hot button" issues surrounding cloud computing relate to storing and securing data, and monitoring the use of the cloud by the service providers. These issues are generally attributed to slowing the deployment of cloud services. These challenges can be addressed, for example, by storing the information internal to the organization, but allowing it to be used in the cloud. For this to occur, though, the security mechanisms between organization and the cloud need to be robust and a Hybrid cloud could support such a deployment.

Lack of Standards - Clouds have documented interfaces; however, no standards are associated with these, and thus it is unlikely that most clouds will be interoperable. The Open Cloud Forum is developing an Open Cloud Computing Interface to resolve this issue and the Open Cloud Consortium is working on cloud computing standards and practices. The findings of these groups will need to mature, but it is not known whether they will address the needs of the people deploying the services and the specific interfaces these services need. However, keeping up to date on the latest standards as they evolve will allow them to be leveraged, if applicable.

Continuously Evolving -User requirements are continuously evolving, as are the requirements for interfaces, networking, and storage. This means that a “cloud,” especially a public one, does not remain static and is also continuously evolving.

Compliance Concerns - The Sarbanes-Oxley Act (SOX) in the US and Data Protection directives in the EU are just two among many compliance issues affecting cloud computing, based on the type of data and application for which the cloud is being used. The EU has a legislative backing for data protection across all member states, but in the US data protection is different and can vary from state to state. As with security and privacy mentioned previously, these typically result in Hybrid cloud deployment with one cloud storing the data internal to the organization.

There has been various types of scheduling algorithm exist in distributed computing system. Most of them can be applied in the cloud environment with suitable verifications. The main advantage of job scheduling algorithm is to achieve a high performance computing and the best system throughput. Traditional job scheduling algorithms are not able to provide scheduling in the cloud environments. According to a simple classification, job scheduling algorithms in cloud computing can be categorized into two main groups; Batch mode heuristic scheduling algorithms (BMHA) and online mode heuristic algorithms. In BMHA, Jobs are queued and collected into a set when they arrive in the system. The scheduling algorithm will start after a fixed period of time. The main examples of BMHA based algorithms are; First Come First Served scheduling algorithm (FCFS), Round Robin scheduling algorithm (RR), Min-Min algorithm and Max-Min algorithm by On-line mode heuristic scheduling algorithm, Jobs are scheduled when they arrive in the system. Since the cloud environment is a heterogeneous system and the speed of each processor varies quickly, the on-line mode heuristic scheduling algorithms are more appropriate for a cloud environment. Most fit task scheduling algorithm (MFTF) is suitable example of On-line mode heuristic scheduling algorithm.

First Come First Serve Algorithm: Job in the queue which come first is served. This algorithm is simple and fast.

Round Robin algorithm: In the round robin scheduling, processes are dispatched in a FIFO manner but are given a limited amount of CPU time called a time-slice or a quantum. If a process does not complete before its CPU-time expires, the CPU is preempted and given to the next process waiting in a queue. The preempted process is then placed at the back of the ready list.

Min-Min algorithm: This algorithm chooses small tasks to be executed firstly, which in turn large task delays for long time.

Max – Min algorithm: This algorithm chooses large tasks to be executed firstly, which in turn small task delays for long time.

Most fit task scheduling algorithm: In this algorithm task which fit best in queue are executed first. This algorithm has high failure ratio.

Priority scheduling algorithm: The basic idea is straightforward: each process is assigned a priority, and priority is allowed to run. Equal-Priority processes are scheduled in FCFS order. The shortest-Job-First (SJF) algorithm is a special case of general priority scheduling algorithm. An SJF algorithm is simply a priority algorithm where the priority is the inverse of the (predicted) next CPU burst. That is, the longer the CPU burst, the lower the priority and vice versa. Priority can be defined either internally or externally. Internally defined priorities use some measurable quantities or qualities to compute priority of a process.

2 Resource Scheduling Method

2.1 Anchor Architecture

Anchor is a general resource management architecture that uses the stable matching framework to decouple policies [1] from mechanisms when mapping virtual machines to physical servers.

In Anchor, clients and operators are able to express a variety of distinct resource management policies as they deem fit and these policies are captured as preferences in the stable matching framework.

The highlight of Anchor is a new many-to-one stable matching theory that efficiently matches VMs with heterogeneous resource needs to servers, using both offline and online algorithms. Our theoretical analysis shows the convergence and optimality of the algorithm.

Anchor Architecture

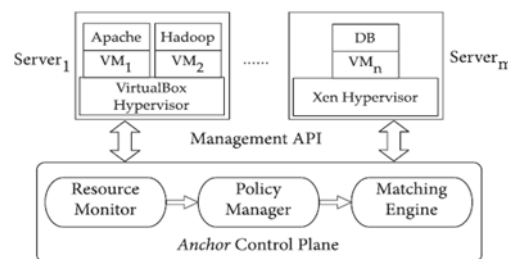


Fig 1. Anchor Architecture

Anchor consists of three components

1. A Resource Monitor
2. A Policy Manager
3. A Matching Engine

Resource Monitor

- Both the operator and its clients are able to configure their resource management policies, based on performance, cost, etc., as they deem fit via the policy manager.

Manager Policy

- When VM placement requests arrive, the policy manager polls information from the resource monitor and feeds it with the policies to the matching engine.

Matching Engine

- The matching mechanism resolves conflicts of interest among stakeholders, and outputs a matching between VMs and servers.

2.2 Novel Matching Mechanism

- It is based on the stable matching framework from economics, which elegantly achieves all the design objectives.
- Specifically, the concept of preferences is used to enable stakeholders to express various policies with simple **rank-ordered lists**, fulfilling the requirement of generality and expressiveness.
- Rather than optimality, stability is used as the central solution concept to address the conflicts of interest among stakeholders, fulfilling the fairness requirement.
- Finally, its algorithmic implementations based on the classical Deferred Acceptance algorithm (DA) have been demonstrated to be practical in many real-world applications fulfilling the efficiency requirement.

2.3 Deferred Acceptance Algorithm (DA)

- Jobs propose to machines following the order in their preferences.
- We randomly pick any free job that has not proposed to every machine on its preference to propose to its favourite machine that has not yet rejected it.
- That machine accepts the most favourable offers made so far up to the capacity, and rejects the rest.
- Individual rationality needs to be respected for the matching to be acceptable to all participants.
- Optimization solvers are computationally expensive due to their combinatorial nature, and do not scale well.

2.4 Job-Machine Stable Matching

- Different size VMS.
- Size heterogeneity make easy .
- **Job-optimal weakly stable matching**, in the sense that every job is assigned its best machine possible in all stable matching's.

- **A Revised DA Algorithm** guaranteed to find a weakly stable matching for a given problem.
- **Multistage DA algorithm** to iteratively find a better weakly stable matching with respect to jobs and it produces a correctness, convergence, and job optimality.

2.5 Problem Definition

- Many-to-one stable matching theory that efficiently matches VMs with heterogeneous resource needs to servers.
- **Many-to-one problem** where one server can enroll multiple VMs but one VM can only be assigned to one server. Preferences are used as an abstraction of policies no matter how they are defined.
- Uniform VMs so virtual machine stable to work in demand of resource so speed of the resource management decreases.
- Efficiency also decreases.
- Two problems arise when applying the classical DA algorithm here.
- First, the execution sequence is no longer immaterial to the outcome.
- Second, it may even produce an unstable matching. This creates considerable difficulties since we cannot determine which proposing sequence yields a weakly stable matching for an arbitrary problem.
- The DA algorithm fails precisely due to the size heterogeneity of jobs.

3 Proposed Work

The proposed algorithm assigns appropriate resource to the very task that exactly satisfy its needs for resource, while reserving the resources whose power greatly exceed the needs of current tasks for future use. By this way, our algorithm can effectively avoid assigning powerful resources to simple and medium scale tasks or assigning poor resources to complex large-scale tasks, for they may lead to misuse of resources and failure scheduling of tasks. The presented algorithm has high efficiency and good robustness and works better than other similar algorithms. Our work has provided a new approach for Cloud resource allocation.

In this proposed work,

By using Hybrid Fuzzy clustering algorithm[13] approach. A new way to schedule appropriate resources to relevant tasks. It exactly satisfy the needs of tasks. Our algorithm avoid misuse of resources or failure scheduling of tasks when assigning powerful resources to simple or medium scale tasks or assigning poor resources to complex large scale tasks.

3.1 Proposed Algorithm Idea (HFC)

Before allocating Cloud resources to Cloud tasks, firstly use a set of parameters.

Parameters are,

- CPU main frequency (f_{cpu})
- CPU free rate (CPU_{free})
- Total memory size (MEM_{all})
- Available memory size (MEM_{avl})
- Available network bandwidth (NB_{avl})
- Available disk space ($DISK_{avl}$)

So, The components of resource vector are

- $R = (f_{cpu}, CPU_{free}, MEM_{all}, MEM_{avl}, NB_{avl}, DISK_{avl})$

The same way, the task vectors are

- $T = (f_{cpu_by_task}, CPU_{free_by_task}, MEM_{all_by_task}, MEM_{avl_by_task}, NB_{avl_by_task}, DISK_{avl_by_task})$

3.2 Proposed Multilayer Workout

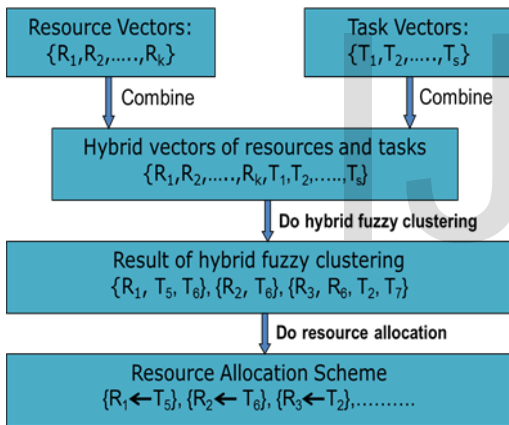


Fig 2. Hybrid Fuzzy Clustering

4.2 Analyzing The Task

- Find the suitable resource for the task.
- Combine the similar task in to the Group.
- Name the Groups and arrange in order.

4.3 Mapping The Task & Resources

- Using the Hybrid Fuzzy Clustering Algorithm map the resource with task.
- Our algorithm avoid misuse of resources or failure scheduling of tasks when assigning powerful resources to simple or medium scale tasks or assigning poor resources to complex large scale tasks.

4.4 System Architecture

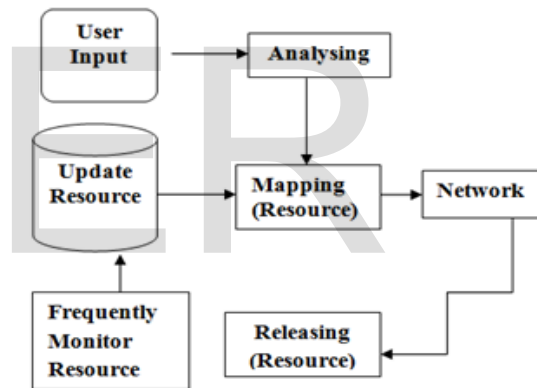


Fig 3. System Model

4 Modules

- Resource table updation
- Analyzing the Task
- Mapping the task & Resource

4.1 Resource Table Updation

- Adding the resource details in the database.
- Number of Resources of available in the Network-Cloud for the Application.
- Address the Resources.

5 Implementation

We investigate the performance of Hybrid Fuzzy Clustering with implementation. cloud sim is a web application that runs in a Virtual Machine on the Amazon webservice cloud. Fuzzy logic is a form of many valued logic. fuzzy logic variable may have truth value that ranges in degree between 0&1. Hybrid fuzzy clustering used to schedule the partially freed resources. The goal of fuzzy clustering is to determine the best grouping in a set of unlabeled data. The group or cluster n no of objects in to m number of groups depends on defined criteria. Hybrid fuzzy clustering is a new way to schedule appropriate resources to relevant tasks.

6 Related works

We achieve the good resource scheduling by using the Hybrid fuzzy clustering techniques. Basically it depends upon the concept of resource scheduling.

This work is related to research in the following fields.

An Optimal Model for Priority based Service Scheduling Policy for Cloud Computing Environment. Another planning calculation dependent upon necessity and affirmation control plan [2]. In this calculation necessity is doled out to each one conceded queue. Concession of each one queue is chosen by computing middle of the road postpone and administration cost.

Enhanced Cost-Based Algorithm for Task Scheduling. An enhanced expense based planning calculation for making productive mapping of assignments to accessible assets in cloud [8]. The act of spontaneity of accepted movement based setting back the ol' finances is proposed by new errand planning procedure for cloud environment where there may be no connection between the overhead provision base and the way that distinctive undertakings cause overhead cost of assets in cloud. This planning calculation separates all client undertakings relying upon necessity of each one assignment into three separate records. This planning calculation measures both asset expense and reckoning execution, it additionally improves computation/communication degree.

Stable matching [1]. Calculation to fathom the coupled position of Vms in server farms. Our late work backer stable matching as a general structure to tackle organizing issues. As far as anyone is concerned, all these works accept an accepted unsize employment model, while we concentrate on a more general size-heterogeneous model.

Our work is likewise identified with the written works on asset booking.

Conclusion

The proposed technique of resource scheduling using hybrid fuzzy clustering is a new way to schedule appropriate resources to relevant tasks and it exactly satisfies the need of tasks. The presented algorithm has high efficiency and good robustness and work better than other similar algorithms. Our work has provided a new approach for cloud resource allocation.

Acknowledgment

I would like to sincere thanks to Mr. Anand Raj, my project guide who gave me valuable advice and were good enough to find time for fruitful discussions.

References

- [1] Hong Xu, Student Member, IEEE, and Baochun Li, "Anchor: A Versatile and Efficient Framework for Resource Management in the Cloud", *IEEE transactions on parallel and distributed systems*, 2013, vol. 24, no. 6
- [2] Dakshayini.M and Guruprasad H.S, "An Optimal Model for Priority based Service Scheduling Policy for Cloud Computing Environment", *IJCSEA (0975 – 8887) 2011Volume 32–No.9*
- [3] El-Sayed El-kenawy.T, Ali Ibraheem El-Desoky, Mohamed Al-rahamawy.F, "Extended Max-Min Scheduling Using Petri Net and Load Balancing", 2012, (IJSCE) ISSN: 2231-2307, Volume-2.
- [4] Saeed Parsa and Reza Entezari-Maleki "RASA: A New Task Scheduling Algorithm in Grid Environment", in *World Applied Sciences Journal 7 (Special Issue of Computer & IT)*, 1995 pp. 573-595.
- [5] Shalmali Ambike, Dipti Bhansali, Jaee Kshirsagrand, Juhi Bansiwl "An Optimistic Differentiated Job Scheduling System for Cloud Computing", (IJERA) ISSN: 2248-9622 www.ijera.com, 2012, Vol. 2, Issue 2, pp.1212-1214 .
- [6] Shamsollah Ghanbari and Mohamed Othman "A Priority based Job Scheduling Algorithm in Cloud Computing", 2012, *International Conference on Advances Science and Contemporary Engineering (ICASCE)*.
- [7] Selvarani.S and sudha Sadhasivam.G "Improved cost-based algorithm for task scheduling in Cloud computing", 2010, *IEEE* .
- [8] A.E. Roth "Deferred Acceptance Algorithms: History, Theory, Practice, and Open Questions," 2008, *Int'l J. Game Theory*, vol. 36, pp. 537-569.
- [9] B. Sotomayor, R. Montero, I. Llorente, and I. Foster "Virtual Infrastructure Management in Private and Hybrid Clouds," 2009 *IEEE Internet Computing*, vol. 13, no. 5, pp. 14-22.
- [10] A. Kalbasi, D. Krishnamurthy, J. Rolia, and M. Richter "MODE: Mix Driven On-Line Resource Demand Estimation," 2011 *Proc. IEEE Seventh Int'l Conf. Network and Service Management (CNSM)*.
- [11] Z. Shen, S. Subbiah, X. Gu, and J. Wilkes, "CloudScale: Elastic Resource Scaling for Multi-Tenant Cloud Systems," 2011, *Proc. ACM Second Symp. cloud Computing (SoCC)* .
- [12] S. Nascimento, B. Mirkin and F. Moura-Pires, "Multiple prototypes model for fuzzy clustering", 1999 in J. Kok, D. Hand and M. Berthold, (eds.), *Advances in Intelligent Data Analysis Third International Symposium, (IDA'99)*, Lecture Notes in Computer Science, 1642, Springer-Verlag, 1999, pp. 269–279.
- [13] R. Dave, "Fuzzy shell-clustering and applications to circle detection of digital images", *International Journal of General Systems*, 16(4), 1990 ,pp. 343–355..