Efficient Cloud Computing for Scientific Communities: Design and Implementation

Seema Verma
PG Fellow (Computer Network)
Smt. Kashibai Navale College of Engineering
Pune-411041, India

Rachana A. Satao
Asst. Prof. (Depts. Of Computer Engineering)
Smt. Kashibai Navale College of Engineering
Pune-411041, India

ABSTRACT
The main purpose of cloud system is to provide the unlimited computing and storage resources. This paper describes that existing DawningCloud is an efficient cloud system for scientific communities. DawningCloud provides efficient resource management and provisioning policy. DawningCloud gave the concepts of Enhanced Scientific Public cloud and also included the concepts of runtime environment of PhoenixCloud. However, this system didn’t investigate the effect of different scheduling policy. DawningCloud simply used First Come First Served (FCFS) scheduling Policy and didn’t investigate the effect of different scheduling techniques on them.

Therefore, we proposed a new cloud system for scientific communities. This proposed science cloud included the concept of DawningCloud and novel Earlier Account Expire Prioritized with Round Robin (EAEP-RR) scheduling technique.

This paper describes the design and implementation of proposed Science Cloud. However, this scheduling method provides benefits only to the earlier account expiration user. Remaining users served as a FCFS basis.

General Terms
Cloud Computing, Scientific Community, Runtime environment (RTE), Earlier Account Expire Prioritized with Round Robin (EAEP-RR) scheduling

1. INTRODUCTION
Traditionally, many scientific communities used dedicated cluster systems (DCS) to provide resources to research group. So, we consider this cluster system as a dedicated system model. For dedicated system model, scientific community owns a dedicated cluster system and then deploys the system according to specific scientific workflow or scientific workload. The drawback of a dedicated system is that for peak load, it is not able to provide sufficient resources, while lots of resources are idle for light loads. The cost of a dedicated cluster system is very high.

Recently, more and more research groups showing great interests in private cloud or proposing hybrid cloud models to argument their local computing resources with external public clouds. However, L. Wang et al. [1] have proved that private cloud needs separate skilled staff to manage the whole systems and only research communities belonging to same institution shared the same cloud resources. Paper [1] concludes that hybrid cloud is not good for some parallel applications due to complexity and network delay. According to L. Wang et al. [1] public cloud can provide solutions of all the issues of private and hybrid cloud.

S. Verma et al [2] classified scientific cloud computing in three categories: first, cloud, according to scientific workflows; second, cloud, according to different workloads; and third, cloud for scientific communities which uses runtime environment (RTE) as an important entity.Paper [2], also classified that DawningCloud[1] comes under the category where RTE is a important entity.

However, classification and evaluation of scientific cloud computing, conclude that paper [1] able to provide an economic cloud to research communities. Paper [1] provides resources to each individual end user and also to a group of many organizations. Existing DawningCloud proposed an automatic resource management and provisioning policy and able to reduce the total resource consumption of the resource provider. However, this system simply uses “First Come First Served (FCFS)” model for scheduling the requests. Though this method is economic for scientific communities, the system faces the problem in scheduling.

S. Verma et al [3] conclude that scheduling is an NP-hard problem where there is no optimal solution. Therefore, we should investigate that which scheduling technique will be better for DawningCloud.Paper [3] proposed a new scheduling technique for DawningCloud called as Earlier Account Expire Prioritized with Round Robin (EAEP-RR) scheduling. EAEP-RR is a better algorithm than FCFS. However this scheduling method provides benefit only to the earlier account expiration user. Other users served as a FCFS basis.

The organization of this paper is as follows: Section 3 describes the software requirement specification of scientific communities cloud; Section 4 represents the design of scientific communities cloud; Section 5 represents the literature survey; Section 6 draws a conclusion and future work.

2. MOTIVATION
Motivation of this paper is to understand the software requirement specification and software design specification of proposed science cloud for scientific communities [3].

3. SOFTWARE REQUIREMENT SPECIFICATION

3.1 Activity Diagram
The objective of this section is to understand the sequence of activity. Fig. 1 shows the Activity Diagram. The sequence of activities explains as follows:

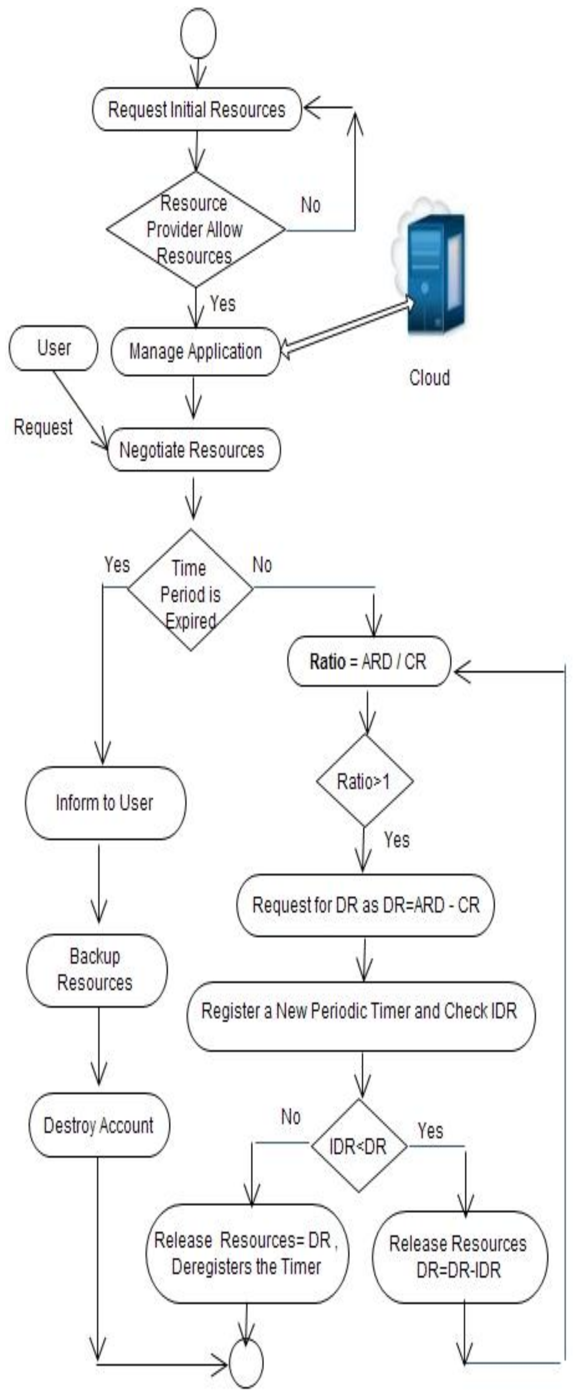
Request Initial Resources
Resource Provider Allow Resources
No
Yes
User
Manage Application
Cloud
Request
Negotiate Resources
Yes
Time Period is Expired
No
Ratio = ARD / CR
Inform to User
Ratio>1
Yes
Request for DR as DR=ARD - CR
Backup Resources
Register a New Periodic Timer and Check IDR
Destroy Account
No
IDR<DR
Yes
Release Resources= DR , Deregisters the Timer
Release Resources DR=DR-IDR

Fig 1: Activity Diagram

Service Provider defines its requirement and requests for Initial Resources. Resource Provider allows Initial Resources according to requirement of the service provider.

Now the service provider has authority to manage resources. If end user request for resources service provider can creates accounts for end users.

Resource provider negotiates resources to release idle resources according to requests of users. If the time period is expired then, the service provider informs to the user. End user takes bakeup of data. Therefore, resource provider destroys the account and withdraws the corresponding resources.

Server scans jobs in queue per checking resource cycle. If the ratio of accumulated resource demand to current resources owned (Initial Resources) is exceeds by one then server requests for dynamic resources.

Here value of dynamic resources DR=Accumulated resource demand (ARD) – Current resources owned (CR).After receiving dynamic resources the server register a new periodic timer and checks available idle dynamic resources (IDR).

Server releases the resources of size DR if the value of IDR is equal or more than requested DR and deregisters the timer.If value of available IDR is less than requested DR releases resources of size DR= (DR – Idle dynamic resources).

3.2 Usage Scenario

3.2.1 Users Profile

- Resource Provider: It is a cloud resource provider which provides resources to service providers.
- Service provider: It is a web application by which end users can request for resources.
- End User: End users are also belongs to scientific communities which are affiliated by service providers

3.2.2 Use Case View

Fig. 2 shows the use case view.

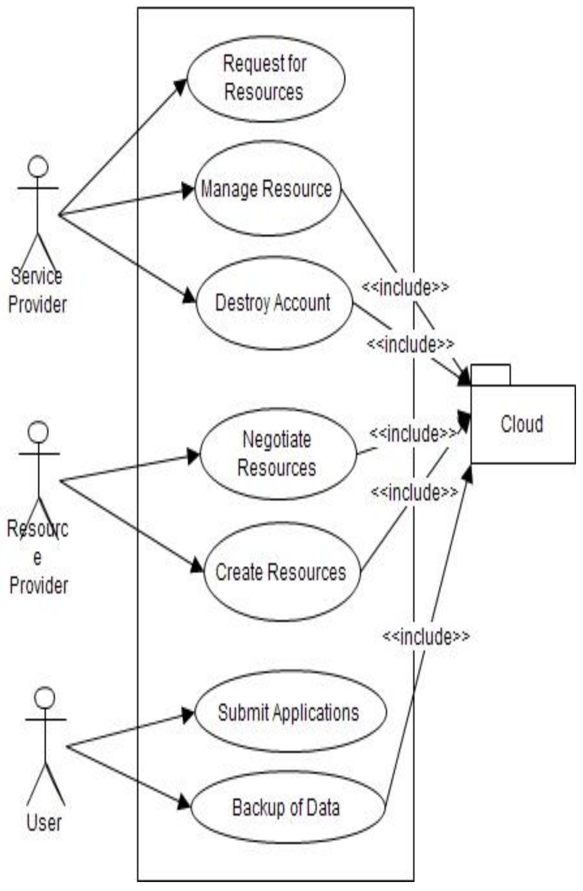

Fig 2: Use- Case View

- Request for resources: Service provider requests for resources from cloud system. Cloud resource provider provides resources to service provider according to their requirement.

- Manage resources: Cloud provided resources to service provider according to size of resources and time duration. After resource allocation by cloud provider, service provider manages its resources with full right of creating account for users.

- Destroy Account: Service provider destroys account if time period is expired.

- Create Resources: Resource Provider creates resources according to runtime environment of service provider. Run time environment includes types of resources, size of resources and time period.

- Negotiate Resources: Resources provider negotiates resources if end user requesting for resources.

- Submit Applications: After authentication by service provider end user can submit application or data.

- Backup Data: Before time period expiration end user takes backup of data.

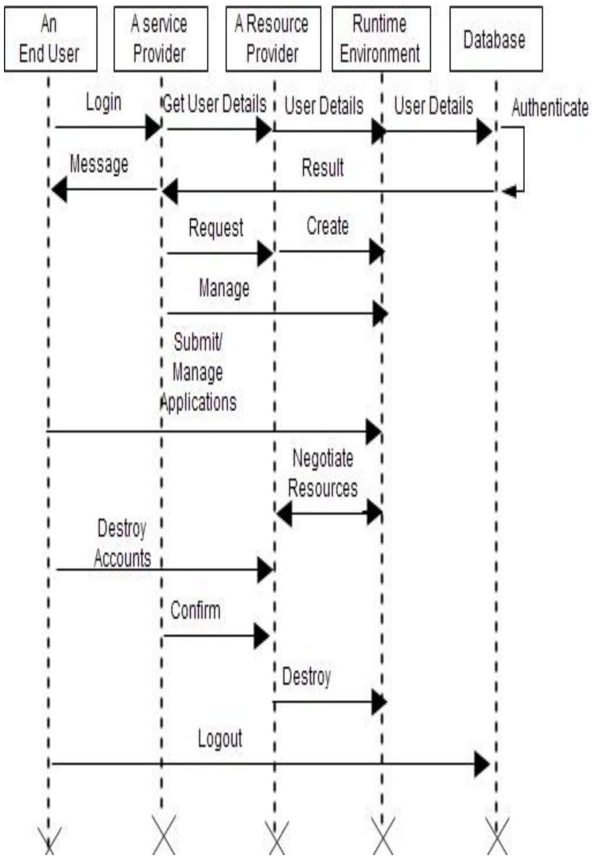

Fig 3: Sequence Diagram

3.3 Sequence Diagram

Fig. 3 shows the sequence diagram.

- A service provider specifies its runtime environment requirements, including size of resources, time duration and then requests to a resource provider.

- A resource provider creates a runtime environment for a service provider according to its requirement.

- After a runtime environment is created, a service provider manages its runtime environment with full control, e.g. creating accounts for end users.

- Each end user uses its accounts to submit and manage applications in a runtime environment.

- When a runtime environment is being providing services, a runtime environment can automatically negotiate resources with the proxy of a resource provider to resize resources by releasing idle resources according to current workload status.

- If a service provider wants to stop its service, it will inform its affiliated end users to backup data.
- Each end user can backup its data to storage servers provided by a resource provider. And then a service provider will destroy accounts of each end user in a runtime environment.
- A service provider confirms a resource provider that the runtime environment is ready for destroying.
- A resource provider destroys the specified runtime environment and withdraws the corresponding resources.

3.4 Data Flow Diagram

Fig. 4 shows the sequence diagram.

3.4.1 Level 0 DFD

The following fig. 4 shows the DFD level-0 of system. Users of the system requests for resources according to their own Run time environments. Our system provides resources as a result of requests.

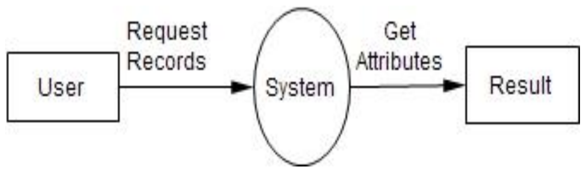

Fig 4: Level 0 DFD

3.4.2 Level 1 DFD

The following fig. 5 shows the DFD level-1 of system. Service providers will request for resources for their own scientific application.

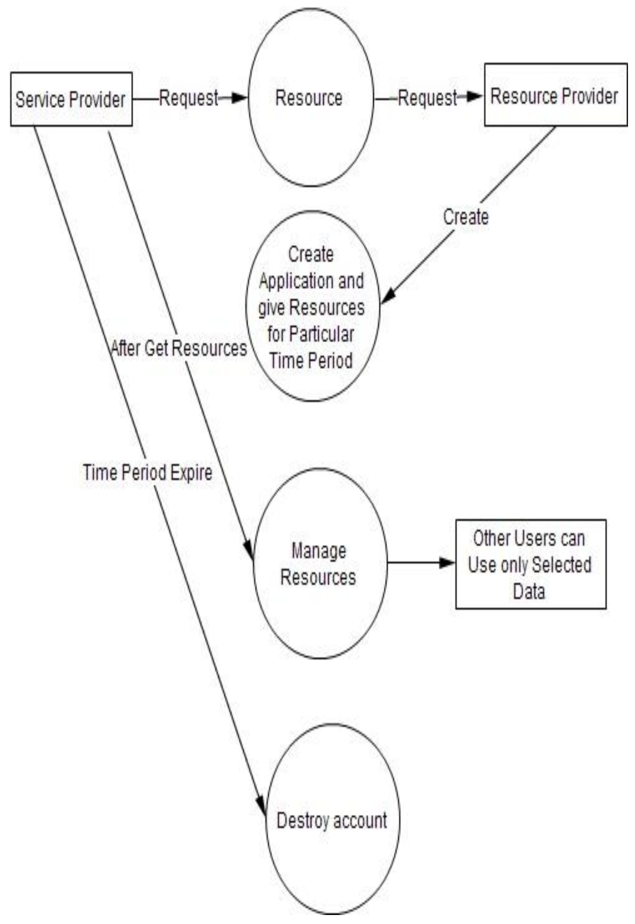

Fig 5: Level 1 DFD

3.4.3 Level 2 DFD

The following fig. 6 shows the DFD level-2 of system. If service provider requested for new resources then resource provider checks Ratio (R). If accumulated resource demands of all jobs in the queue is more than the current resources owned by a thin run time environment (TRE) than resource provider creates the new dynamic resources to the existing service provider.

4. DESIGN

4.1 Deployment Diagram

Fig. 7 shows the deployment diagram.

- User requests to Web Server by SOAP (Simple object access object protocol) and HTTP (Hypertext Transfer Protoco.l).
- Web server [11], IIS (Internet Information Services, also known as Internet Information Server) provides services to users.
- The .NET Framework's [12] Base Class Library provides user interface, data access, database connectivity, cryptography, web application development. Programmers produce software by combining their own source code with the .NET Framework and other libraries. Microsoft also produces an integrated development environment largely for .NET software called Visual Studio.

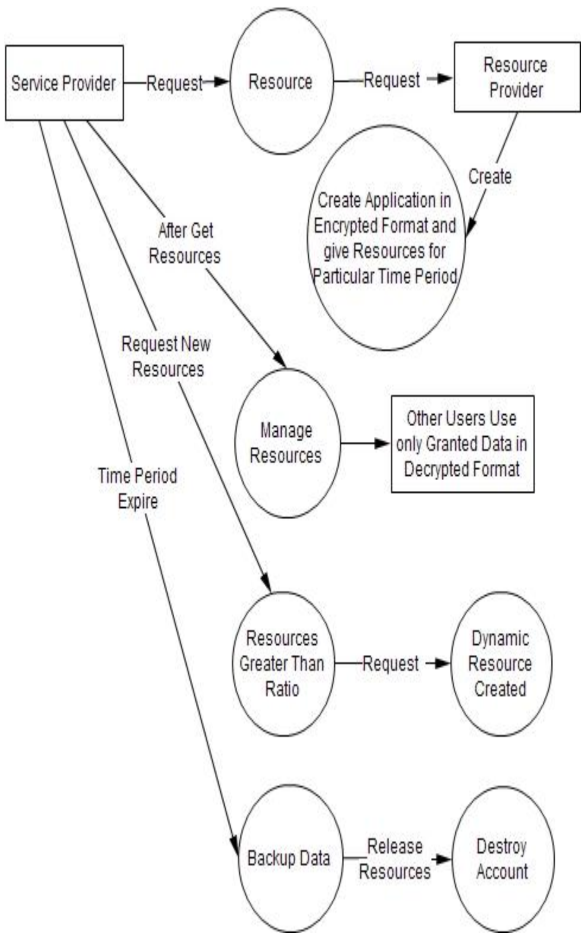

Fig 6: Level 2 DFD

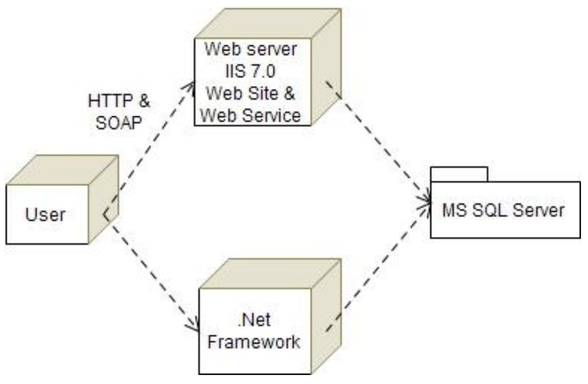

Fig 7: Deployment Diagram

- Microsoft SQL Server [13] is a relational database management system developed by Microsoft. As a database, it is a software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet).

4.2 Class Diagram
Fig. 7 shows the deployment diagram. Main classes of this Web Application are as follows:

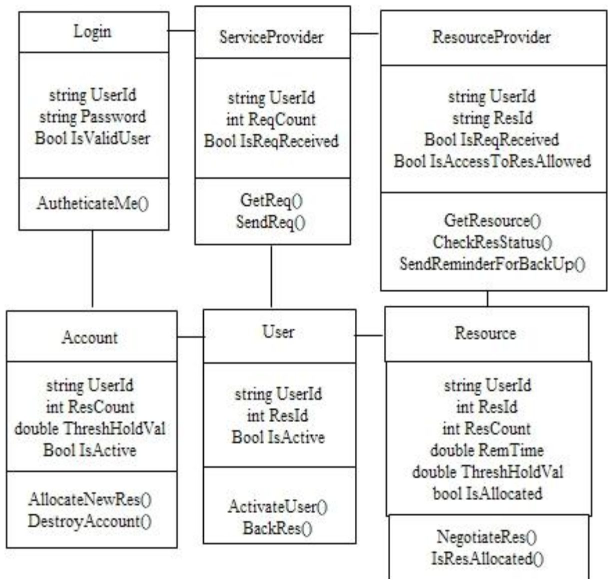

Fig 8: Class Diagram

- Login: This class defines the function for the authentication of user.
- Account: If any user time period expired then this class destroys account.
- ServiceProvider: This class accepts request from end users and also sends request to resource provider.
- User: This class contains information of active users. If a service provider wants to stop its service then it sends information to end users, so that they can take backup of data.
- ResourceProvider: This class gives resources to service provider and checks resource status. If time period expires then sends reminder to service provider to backup data.
- Resources: This class negotiates resources if service provider or end user requests for resources. .

5. LITERATURE SURVEY
5.1 Existing System
Wang et al [1] proposed a system that provides an Enhanced Scientific Public cloud Model (ESP). ESP allows small and medium scale research organizations to share resources from the other communities for a period of time. According to ESP model and PhoenixCloud, Wang et al [1] designed a DawningCloud system.

This proposed DawningCloud system able to handle the heterogeneous requests from the clients. DawningCloud proposed an automatic resource management and provisioning policy and able to reduce the total resource consumption of the resource provider. However, this system simply uses "First Come First Served" model for scheduling the requests. Though this method works far better than the previously developed systems, the system faces the problem in scheduling.

The FCFS [14] is the most basic type of scheduling in the computer field. The concept of this scheduling is very simple. The request that comes earlier will be given the resources. The tricky part is, when a long time required to execute a current job remaining jobs has to wait for a long time. Now, the remaining jobs, no matter how small they are (according to burst time), or even their importance or priority is not at all concerned in this method.

5.2 Resource Management Issues
Resource management issues are generally researched in the situation of cloud computing and grid computing.

In the setting of private cloud B. Sotomayor et al [4] implement Haizea which represent architecture of lease management building block. The OpenNebula is a platform which can manage the virtual infrastructure to construct private, public and hybrid implementations of infrastructure as a service. Authors [4] conclude that integration of OpenNebula and Haizea lease manager can provide a VM management solution helping a variety of lease types. However integration of OpenNebula and Haizea is yet in an early stage. So, integration of OpenNebula and Haizea further required many improvement and enhancement.

L. Grit et al [5] implemented the Winks scheduler to help a weighted fair sharing model for a virtual Cloud computing utility. WINKS provided efficient sharing between research groups and also provided incentive if the shared servers are not utilized by others groups.

In the situation of hybrid cloud, M. D. de Assuncao et al [6] uses scheduler, which schedule or redirect the requests according to different provisioning policies.

Paper [6] investigate the performance of six scheduling techniques utilized by an association that provides a cluster handle by virtual machine technology and looks to use resources from a cloud provider to minimize the response time of its client request. Experiment results conclude that if the local site's cluster is underutilized, then the cost of increasing the application scheduling performance will be high.

To overcome these issues, we are providing our proposed model with a novel scheduling technique.

5.3 Scheduling Algorithm
Job scheduling is a vast research topic of computer science and no any particular scheduling can be an optimal solution for all types of applications.

S. Ghambari et al [7] proposed a new priority based job scheduling algorithm (PJSC), based on multiple criteria decision making model. PJSC is a view of Analytical Hierarchy Process (AHP). AHP is a representation of multi-criteria decision-making (MCDM) and multi-attribute decision-making (MADM) model. PJSC has reasonable complexity. However, target to achieve less completion time is considered as future work.

H. A. Abba et al [8] proposed modified prioritized deadline based scheduling algorithm (MPDSA) for the grid system. This scheduling algorithm is utilizing a project management technique for effective job execution with a due date constraint of the user's jobs. MPDSA executes jobs by closest due time, postpone in a cyclic way utilizing dynamic time quantum. Simulation result shows that MPDSA gives short average waiting time, and short turnaround time.

Li Yang et al [9] proposed a new weighted fair scheduling algorithm. It is taking into account strict rob priority class which includes a priority queue. Priority queue taking into account the establishment of based class weighted fair scheduling (CBWFQ) algorithm. Therefore, this algorithm called as a strict rob priority class (SRPQ-CBWFQ) algorithm. This algorithm covers the weakness of customary weighted fair scheduling algorithm. Weighted Fair Scheduling algorithm separates the administrations of all dynamic queues on the premise of weight of every business stream. At the point when the job arrives, the classifier creates different groups according to job classifications. The buffer is checked for every classified job. If the buffer is not over-burden then the job is stored in the buffer otherwise job is dropped. Each one job is entered into an alternate virtual queue. The fundamental point of interest of this algorithm is that it has presented the rob rule together with dropping rule. Various evaluations are carried out on a NS-2 product to reproduce SRPQ-CBWFQ algorithm. This new algorithm consolidated buffer administration and queue scheduling and just ensures minimal delay of ongoing applications. It additionally offered thought to reasonableness and better use of buffers. This algorithm has two incredible favorable circumstances of bandwidth allotment and some delay without throughput reduction.

In the paper [10] authors focused on giving solution for online real time administrations utilizing non-preemptive scheduling algorithm to reduce execution time of the migrated assignment. Prior, a non-preemptive scheduling with migration algorithm is utilized to get the lower penalty. At whatever point an undertaking misses its due date, it will relocate the task to an alternate virtual machine and begins its execution from the earliest starting point. Hence it builds the execution time of the migrated assignment. With a specific end goal to conquer this issue, a non-preemptive ongoing scheduling utilizing check pointing algorithm is proposed to reduce the execution time of the relocated assignments and minimizes the penalty better by prior finish of the task. This enhances the general framework execution.

Table 1. Analysis of scheduling algorithms

Paper	Description	Conclusion
S. Ghambari et al [7]	1. Used PJSC Algorithm. 2.Based on MCDM and MADM model	Target to achieve less completion time is considered as future work
H. A. Abba et al [8]	1. Used MPDS Algorithm. 2.Based on effective job execution with a due date constraint of the user's jobs	Gives short average waiting time, and short turnaround time
Li Yang et al [9]	1 Used new weighted fair scheduling algorithm. 2. Called SRPQ-CBWFQ algorithm.	1.Ensures minimal delay 3. Better use of buffers.
R. Santosh et al [10]	1.Non-preemptive scheduling 2. Used check pointing algorithm.	Provided better execution time of the migrated assignment.
S Verma et al [3]	EAEP-RR scheduling	1. Provides benefit only to the earlier account expiration user. 2. Other users served as a FCFS basis.

In the round robin scheduling, courses of action are dispatched in a First in first out (FIFO) way, however, are given a restricted measure of CPU time called a quantum. On the off chance that a methodology does not finish before its CPU-time lapses, the CPU is appropriated furthermore given to the following methodology holding up in a queue. The preempted methodology is then set at the once again of the prepared list.

S. Verma et al [3] proposed Earlier Account Expire Prioritized with Round Robin (EAEP-RR) scheduling which can provide benefits only to the earlier account expiration user. Other users served as a FCFS basis.

5.4 Contribution

Fig. 9 shows proposed architecture diagram.

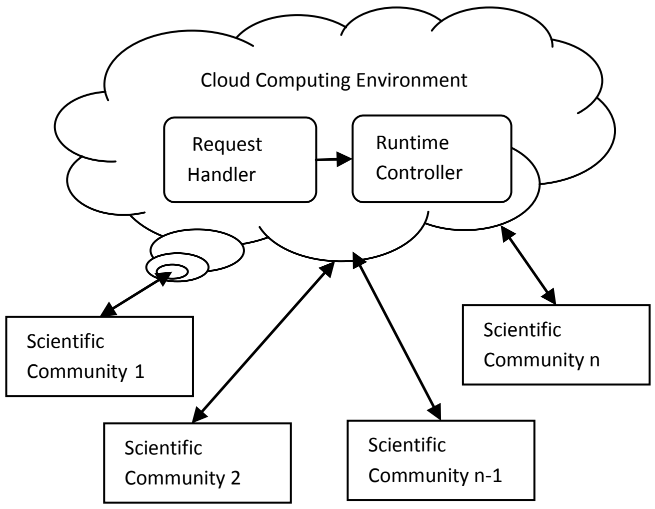

Fig 9: Proposed Architecture [3]

Paper [1] concludes that proposed system provides smaller management overhead and peak resource consumption is also greater than dedicated cluster system and Evangelinos's system. However, it is able to minimize the total resource consumption than all other systems. Small management overhead of cloud computing is acceptable because it minimizes total resource consumption than all other systems.

Paper [1] has proposed an efficient resource management and provisioning policy for a single public resource provider so that scientific communities can get the benefit of elasticity.

After the large amount of study in this field, we have finally reached the system that we are proposing in this paper. As we have studied, it is very costly for small and medium scale scientific communities to buy all the necessary resources for the research. Thus, it is easy to share resources for the scientific communities. Hence, we are providing extended the idea of [1] in our paper.

Fig. 9 shows an overview our [3] proposed system. Initially, we are presenting a novel model for the scientific communities. Using this model, server (request handler) is providing the resources according to research community's requirement. After resource allocation each community can control their own runtime environment, and can share the resources with other communities. This allows the communities to share the needed resources from other communities, which in turn will lower the necessary cost for the execution. Paper [1] used very efficient resource management and provisioning policy which reduced the total resource consumption of cloud provider.

Though, this model is enough for the resource sharing process, we need to think about the better scheduling algorithm than paper [1] used.

To overcome these issues of scheduling, we have studied a number of scheduling techniques. We have studied that, different scheduling techniques provide different advantages for the scheduling. Some saves the turnaround time, while some saves the waiting time for requests. But the problem with them is that, they lacks at some point. Thus, to overcome these issues, we have proposed a novel scheduling method, called as "Earlier Account Expire Prioritized with Round Robin (EAEP-RR) scheduling".

With this EAEP-RR scheduling, we can easily handle the requests on the basis of priorities, without making the queued requests wait longer. Also, the inclusion of round robin makes sure that the multiple requests can be handled simultaneously.

6. CONCLUSION AND FUTURE WORK

In this paper, we have described the software requirement specification and the software design specification of proposed cloud for scientific communities.

Proposed Earlier Account Expire Prioritized with Round Robin (EAEP-RR) scheduling provides benefits only to the earlier account expiration user. Other users served as a FCFS basis.

Proposed science cloud also provides benefits only to the small and medium scale research institution. In the near future, we are looking ahead to propose this system for the larger scale scientific communities. Thus, a large amount of study is required.

7. ACKNOWLEDGMENTS

The proposed system is based on the paper under the title "In Cloud, Can Scientific Communities Benefit from the Economies of Scale?", IEEE Transaction on Parallel and Distributed Systems, Vol. 23, Feb 2012.

The authors would like to thank the researchers as well as publishers for making their resources available and professors for their guidance. We are also thankful to all anonymous reviewers for their valuable suggestions.

8. REFERENCES

[1] Lei Wang, Jianfeng Zhan, Weisong Shi, Senior Member IEEE, and Yi Liang, "In Cloud, Can Scientific Communities Benefit from the Economies of Scale?", IEEE Trans Parallel & Distributed Computing , 2012 , pp. 296-303.

[2] Seema Verma and Rachana A. Satao, "A Classification of Cloud Computing for Scientific Communities", Published in Third Post Graduate Conference on Computer Engineering cPGCON 2014, Elsevier.

[3] Seema Verma and Rachana A. Satao, "A Survey on the impact of Economies of Scale" Accepted for IEEE Conference, May 2015 edition.

[4] W. Kornfeld, C.E. Hewitt, "The Scientific Community Metaphor", IEEE Trans. Sys, Man, and Cyber, SMC-11 (1): 24–33, 1981.

[5] L. Grit, J. Chase, "Weighted fair sharing for dynamic virtual clusters", In Proceedings of SIGMETRICS 2008, pp. 461-462.

[6] M. D. de Assuncao, A. di Costanzo, and R. Buyya, "Evaluating the cost-benefit of using cloud computing to extend the capacity of clusters", In Proceedings of HPDC '09.

[7] Shamsollah Ghanbari and Mohamed Othman. "A Priority based Job Scheduling Algorithm in Cloud Computing." Elsevier, Procedia Engineering 50 (2012), 778-785.

[8] H. A. Abba, N Zakaria, S.N.M. Shah and A.J. Pal "Design, Development and Performance Analysis of Deadline Based Priority Heuristic for Job Scheduling on a Grid" Procedia Engineering, Vol- 50(2012), 397-405.

[9] Li Yang, ChengSheng Pan, ErHan Zhang, HaiYan Liu , "A new Class of Priority-based Weighted Fair Scheduling Algorithm", Elsevier,Physics Procedia, 33 (2012) 942 – 948.

[10] R. Santhosh, T. Ravichandran, "Non-Preemptive Real Time Scheduling using Checkpointing Algorithm for Cloud Computing", IJCA Journal Volume 80 - Number 9 Year of Publication: 2013.

[11] Internet Information Services: http://en.wikipedia.org/wiki/Internet_Information_Services, Accessed on 18th May, 2015.

[12] .NET Framework: http://en.wikipedia.org/wiki/.NET_Framework, Accessed on 18th May, 2015.

[13] Microsoft SQL Server: http://en.wikipedia.org/wiki/Microsoft_SQL_Server, Accessed on 18th May, 2015.

[14] Operating System Design/Scheduling Processes/FCFS, http://en.wikibooks.org/wiki/Operating_System_Design/Scheduling_Processes/FCFS, Accessed on 9th January, 2015.