International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Advance Reservation Policy for Inter-Member Workflow Scheduling in Community Cloud

Ajeet Kumar Bhartee[1], Akshay Saxena[2], Anmol Anand Srivastava[3], Ashish Kumar Patel [4]

[1]Assistant Professor at Galgotias College of Engineering and Technology, Greater Noida, Uttar Pradesh , India

[2, 3, 4] Student at Galgotias College of Engineering and Technology, Greater Noida, Uttar Pradesh , India

Abstract: ***The problem of solving fast collaboration among many related enterprises has been an important research issue in the Web Society research area. The recent proposed Cloud Computing application model called the Community Cloud Computing has provided a good solution for this problem. The problem with the traditional allocation of resources in inter-member workflow scheduling in community cloud is that the resource may be not available when needed and the task/job will have to wait for a long time. This can lead to a relatively poor performance of the system resource utilization rate. The reason of the low utilization rate is because it is prone to pick up an overloaded service node and cause a waiting delay. To overcome this problem, an advance reservation technique has been proposed.***

Keywords: Advance Reservation, Workflow, Grid, PCSA, Community Cloud

1. Introduction

The rapid development and widely use of Internet is attracting more and more people every day. Without question, Web has created a big virtual society for many people and hence the web society has become an important research field in computer science. Among the researches on web society, the problem of how to provide fast collaboration for many related organizations has always been an important research hotpot. In the current background of network development, various business activities become more dependent on online operation than ever, the approach of artificial agreement, coding and deployment to achieve inter-organizational networking and collaborative has been completely unable to meet the needs of internet applications, the demand of a novel network structure providing unified coordination mechanism for fast collaboration of organizations is becoming more and more significant. In recent years, the concept of cloud computing and related frameworks has provided a good solution for this issue. As an application model of cloud computing, the Community Cloud Computing [4] is especially prominent in the aspect of fast internet collaboration provision. Community cloud is a resources community formed by services, hardware resources and other network resources from organizations such as enterprises and institutions. Community cloud users can easily use the services of resources from different members, without knowing the specific location where the service is running. Community members are the basic composition units of community cloud, each member is a complete web application system and has a independent network topology structure. Through the fast collaboration mechanism provided by community cloud, resources within the community members can be flexibly integrated and shared, to achieve a variety of collaboration applications. But how does the fast collaboration mechanism within a community member be achieved so that resources within the community member can be flexibly integrated and shared with minimum waiting delay? In addition to hardware support, supporting software system, especially the workflow management system is also needed in order to provide creation of collaborative service process, formation of collaboration services network and management of process execution. We have used the existing workflow framework design as shown in figure 1. .But, we have made a slight change. We have used two management nodes this time.

2. Workflow Framework Design

In this section the previously proposed workflow system framework and its architecture has been described.

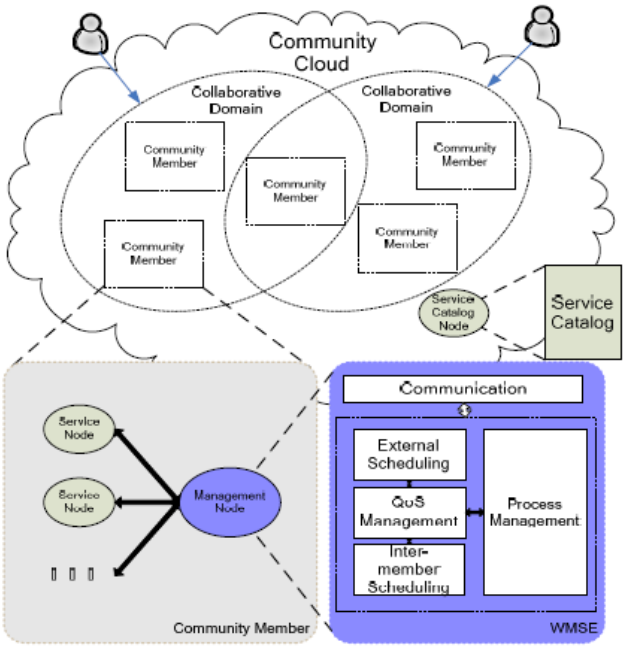

Figure 1: The Workflow system framework architecture[4]

Figure 1 shows the overview of the workflow system framework in community cloud, it includes all levels of the system architecture. First, from the global point of view, the framework contains many virtual collaborative domains, which is composed by the community members related to the corresponding service and built when the corresponding collaboration service is created. Second, in each community

member, the system contains a number of service nodes and **at least one management node**:

1) The service node is an abstraction symbol of the services or other web resources in community members. The service nodes are maintained and managed independently by each community member, and it's the final executor of the collaboration tasks.
2) The management node is set up for the task allocation and other management functions. Through this node the computing and storage resources can be connected to the upper layer of the system in order to participate in the execution of collaboration services.

Furthermore, in each management node a Workflow Management and Scheduling Engine is deployed, accomplishing all the functions of the management node. As shown in the figure, the engine contains four main modules:

1) Process Management Module: This module in charge of the creation of the collaboration services, the creation of the process instances, and the execution management. What is more important, this module is also responsible for the establishment of community member collaborative network.
2) External Scheduling Module: There may be many services with similar functions belong to different community members, the external scheduling is the procedure allocating collaboration tasks onto the appropriate community members, so this module take charge of the scheduling support among the community members, allocating process instances and task partitions onto the community members.
3) Inter-member Scheduling Module: This module takes charge of the internal scheduling procedure inside of community members. Although each community member uses an independent manner to manage its resources, a proper scheduling mechanism is still needed in the management node to assign tasks onto service nodes with similar function. This module can establish a mapping between workflow tasks and service nodes in the community member, to ensure the outcome can meet the user's specific needs.
4) QoS Management Module: This module provide support for the QoS division and the monitoring mechanism. As only global QoS constraint is given while the creation of the collaboration services, the QoS management module need to divide it into pieces of local QoS constraint to meet the needs of execution. At the same time, the monitoring mechanism is also needed in this module in order to detect the execution faults or errors. In addition to the function modules of the structure described above, the community cloud workflow system framework should also need some basic structures such as the communication middleware and service catalogue, which is also included in Figure 1, in order to provide communication support among community members and service discovery support for the creation of the virtual collaborative domains.

For our advance reservation based policy we have used the existing workflow system framework architecture[4]. As per the existing workflow framework system architecture [4], there must be atleast one management node. For our proposed inter-member scheduling policy in community cloud, we have used two management nodes as shown in figure 2.

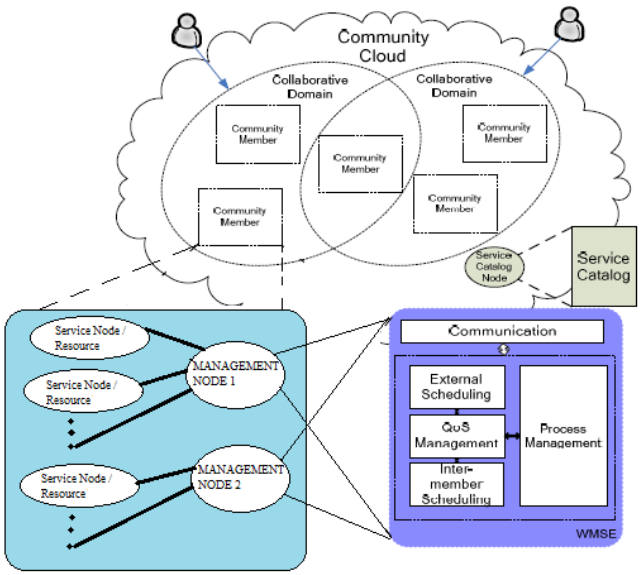

Figure 2: Modified System framework architecture for our proposed policy

3. Proposed Advanced Reservation Based Policy For Inter-Member Workflow Scheduling In Community Cloud

In the original PCSA[1], tasks were scheduled according to the vacant service nodes which had highest simulation period visits but the result indicate that it has a relatively poor performance of the system resource utilization rate. The reason of the low utilization rate is because it is prone to pick up an overloaded service node and cause a waiting delay. So we propose advance reservation based scheduling rule to avoid the tasks to be scheduled to the overloaded service nodes.

In our proposed policy, we have used two management nodes and three resources/service node as shown in figure 3.

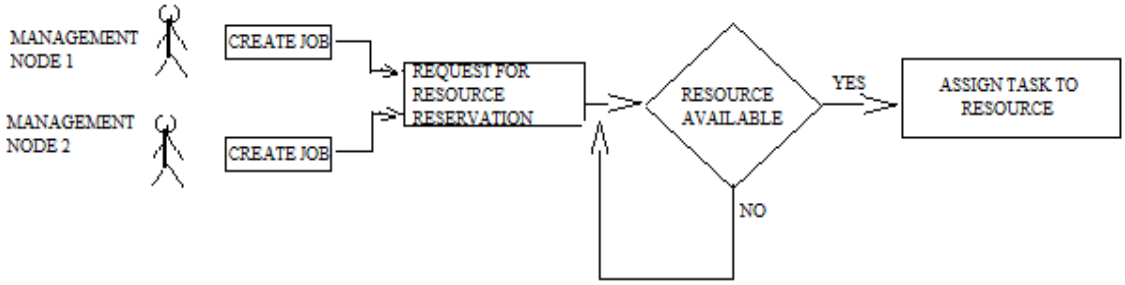

Figure 3: Flow of Algorithm

Management nodes create jobs/tasks and request a reservation in future time. Now if at that time any of the resource is free/ available for execution of jobs/ tasks then the management node submits the jobs to that resource . The two management nodes are assigned different time zones. This means that they request for reservation at different times. Whichever resource is available at the requested reservation time that job/ task is assigned to that resource.

In advance reservation based policy, the resources are space shared.Thus using the advanced reservation policy, resources are never overloaded as they have to process the request from a single management node at a time. Also if the job/ task submitted are greater than the capacity of the resource that is available at the desired reservation time, then the remaining jobs/ tasks are assigned to either the other resource if it is free or else management node has to wait for the current resource executing its job to get free for processing the remaining job requests of that management node.

Steps of Algorithm
Step 1:- Take the two management nodes as input
Step 2:-Assign different time zones to the two management nodes say GMT+8 and GMT-3 respectively.
Step 3:- Now, the two management nodes will request for reservation in future time as per their respective different time zones.
Step 4:-Exit when no more requests for reservation are there from the management nodes.

4. Simulation and Result

All the algorithms have been implemented in java, due to space restriction, the details are not given in this paper. The coding of the comparison experiment is based on the GridSim toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing [2]. Although it is not designed for the simulation in Cloud Computing environment, we can also use it to simulate the working states of computing resources and the scheduling of instances and tasks in Cloud Computing environments.

In Gridsim package, creation of users corresponds to creation of management nodes. Jobs/tasks are represented as Gridlet objects. These Gridlets/jobs are created by the user node/ management node. Each gridlet/ job has a size/ gridlet length of 5000 in terms of integer datatype.

The resources are space shared and each having cost of $3 per second. Each resource and each gridlet possess unique id.

The simulated community member has three resources, each having one machine and each machine having three Processing Element (PE). The parameters of these resources are given in table 1.

Table 1: Resource/ Service Node Configuration

Resource ID	*No. Of Machines*	*No. Of PE*
5	1	3
9	1	3
13	1	3

During simulation, we created three jobs at one management node and five jobs at the other one. We focused on the total cost in terms of $ associated with the execution of a gridlet/job by the resource.

The results have been shown in table 2 for Management Node 1 and in table 3 for management node 2.

Table 2 : Management Node 1

Gridlet ID	Resource ID	Cost
1	13	258.0
2	9	341.99
0	5	600.0

Table 3: Management Node 2

Gridlet ID	Resource ID	Cost
0	5	600.0
1	13	11010.0
2	9	11094.0
3	9	1218.792
4	9	10217.20799

5. Conclusions and Future Work

Community Cloud Computing is a new application model of Cloud Computing and is especially prominent in the aspect of fast internet collaboration provision. This paper has discussed about the advance reservation based policy for inter-member workflow scheduling in community cloud.

In our proposed advance reservation based policy for inter-member workflow scheduling in community cloud, the management nodes are assigned different time zones. Also the resources are space shared. Thus, using the advanced reservation policy, resources are never overloaded as they have to process the request of a single management node at a time. Also if the job/ task submitted are greater than the capacity of the resource that is available at the desired reservation time, then the remaining jobs/ tasks are assigned to either the other resource if it is free or else management node has to wait for the current resource executing its job to get free for processing the remaining job requests of that management node. Thus it also solves the problem of PCSA[1] which is prone to pick up an overloaded service node and cause a waiting delay which results in a poor system utilization rate. In the near future, we will further enhance our scheduling method for energy efficiency. There is also a tradeoff between makespan time and energy[3].

References

[1] Wenhao LI, Haiyang WANG. A Scheduling Algorithm for the Distributed Student Registration System in Transaction-Intensive Environment International Journal of Distance Education Technologies (IJDET), volume 8, issue 5, 2010.
[2] R. Buyya and M. Murshed. GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing, J. of Concurrency and Computation: Practice and Experience, 14(13–15), 2002, 1175–1220.

[3] Multi-Objective Workflow Scheduling: An Analysis of the energy Efficiency and Makespan Tradeoff , Juan J. Durillo and Vlad Nae and Radu Prodan , University of Innsbruck, 6020-Innsbruck, Austria
[4] A Community Cloud Oriented Workflow System Framework and its Scheduling Strategy, LI Wenhao, Faculty of Information and Communication Technologies, Swinburne University of Technology, ayumin_5420467@hotmail.com