3. PROPOSED ARCHITECTURE AND MECHANISM FOR RECOMMENDING CLOUD RENDERFARM SERVICES

The detailed survey of the literature about cloud services, selection and recommendation as discussed in the previous chapter indicated the need for a cloud renderfarm service domain specific cloud broker that could act as an aggregator of these services to overcome various practical problems encountered in developing the recommendation engine for cloud renderfarm services. Hence Cloud Broker architecture was designed with the proposed Recommendation Engine in the topmost layers. This chapter explains in detail about the proposed cloud services broker model, the recommendation engine framework and the proposed recommendation methodology. The details of the experimental test bed that was set up to collect the real time QoS and QoE data on the cloud renderfarm services is also discussed in detail at the end of this chapter.

3.1 ROLE OF CLOUD BROKER IN CLOUD SERVICES RECOMMENDATION

The main goal of the recommender systems is to help the users to identify the right cloud renderfarm services that could meet their multiple criteria requirements that include both the QoS and QoE requirements. However, collecting and maintaining these details about all the cloud renderfarm services is a challenge and requires a third party like the cloud broker.

A cloud broker service is a third-party entity or business that acts as an intermediary between the user and the cloud service providers. A cloud broker usually aggregates, integrates and customizes the cloud services to match or satisfy the user requirements. It aggregates the knowledge about the various cloud services offered by all providers and provides a unified interface / API to enable the users to compare the characteristics of different cloud services [123, 124]. It may play the role of a negotiator between the service provider and the user [125-127].
Many research works have explored the challenges involved in developing the cloud broker model for general cloud services like the IaaS services that are not domain specific. However, developing a cloud broker service model for domain specific cloud renderfarm services involves many challenges like collecting and updating the domain specific knowledge, QoS and QoE data, etc. and not has been explored by many. The proposed cloud broker model acts as a middle man between the cloud renderfarm service providers and the users or animators. The proposed cloud broker architecture and the recommendation methodology are explained in detail in the following section.

3.2 PROPOSED CLOUD BROKER SERVICE ARCHITECTURE

The proposed cloud broker framework for cloud renderfarm services consists of four layers. The topmost layer is the Service Discovery and Recommendation layer. This is the first layer that the user encounters when he accesses the Cloud Broker Service (CSB) portal and consists of the identity and access management module that verifies the identity of the users and provides access to the broker services. The proposed recommendation engine is also built in this topmost layer. The proposed recommendation engine consists of the two main modules, ie) the service discovery agent and the Multi Criteria Recommendation module and is explained in detail as a separate section below.

The second layer is the SLA Manager, which is responsible for the Service Level Agreements (SLA) management. It has the sub modules like the SLA negotiator and the third party monitoring service manager. The SLA negotiator is responsible for negotiating the Service level agreements between the users and the service providers. The third party monitoring service manager manages the collection of QoS data about the services from the third party monitoring systems.

The third layer contains the Information Manager that is responsible for managing all the related information about the services and the users. It contains three sub modules namely the SLA manager, QoS manager and the QoE manager, that are responsible for collecting information about the SLAs signed, QoS and QoE information about the services respectively from all the available sources.
The Data Layer is the fourth layer and contains the database about various types of information stored related to the cloud renderfarm services and the users. The proposed Cloud Broker Architecture is given below in Figure 3.1.
3.3 PROPOSED RECOMMENDATION ENGINE ARCHITECTURE

The proposed recommendation engine architecture of the recommendation mechanism is given below in Figure 3.2. It consists of two main modules, namely the Service Discovery Engine module and the Multi Criteria Recommendation module.

3.3.1 Service Discovery Engine Module

The Service Discovery Engine module is responsible for matching the user functional requirements with the services that could satisfy them. It consists of the Service Profile Collector (SPC). The Service Profile Collector is responsible for collecting the promised SLA details about the cloud renderfarm services from the service providers, when they register their services. The requirements collector (RC) collects the functional requirements details from the user. The semantic layer is responsible for collecting the details about the cloud renderfarm services that has not registered with the cloud broker from their web pages using the domain specific ontology.

The matchmaker module uses three similarity reasoning methods of the knowledge based filtering techniques to identify the services that could satisfy the functional requirements of the animators. The three similarity reasoning methods used are the Similarity reasoning, Equivalent reasoning and the Numerical reasoning methods. At this stage a first level of recommended services list that match only the functional requirements of the animators or users is generated.

3.3.2 Multi-Criteria Recommendation Module

The Multi Criteria Decision Making Methods were integrated into the recommender engine in the Multi Criteria Recommendation Module. The Multi Criteria Recommendation Module comprises of two other sub modules, namely the QoS based Ranking module and the QoE based Re- Ranking module.
The QoS based Ranking module enables the users or animators to specify the weights for the domain specific QoS attributes identified and ranks the services using the Multi Criteria Decision Making (MCDM) Methods. The data required about the QoS attributes are fetched from the QoS repository of the broker service.

If the real time QoS data about the services are not available, then the promised QoS values provided by the service provider is used for ranking services and the actual QoS data are collected, when the animators use the service. This helps to collect more data about the services and about the users. A second level of recommendations is generated at this stage based on the MCDM method ranking order of the services. Thus, at this second level of recommendation, the animators are recommended a list of services that not only satisfies their functional requirements, but also satisfies the multi Criteria QoS requirements of the users.

3.3.3 QoE Based Re-Ranking Module

At the third level of recommendation process, the recommended services at level 2 are further re-ranked based on the content based filtering methods. The content based filtering method, filters the services that are similar to the ones that the user has
liked in the past. In order to identify the services that the user has liked in the past, the QoE attributes ratings provided by the users are considered.

The QoE attributes that are specific to the animation file rendering domain were identified for this specific purpose and were rated by the animators. Based on the QoE ratings of the services provided by the user, the level two recommended services are further re-ranked and a third level of recommendations is provided to the users. Hence, the third level of recommended services satisfies all the three requirements, namely the functional requirements, multi criteria QoS requirements and also the QoE expectations of the user. Since the services are recommended in three different stages the cold start problem could be overcome effectively.

3.4 RECOMMENDATION METHODOLOGY

The recommendation methodology is explained using a data flow diagram given in Figure 3.3 and the block diagram given in Figure 3.4.

![Data Flow Diagram](image-url)
As given in the data flow diagram, the recommendation engine collects the functional requirements from the user or animators and the details about the service offerings from the service providers. Then, the functional requirements of the user are matched using the knowledge based similarity reasoning methods and the matched list of services that satisfies the functional requirements of the users are provided as input for ranking based on the multi criteria QoS requirements of the user using the MCDM ranking methods.

The second level of recommendations is based on the QoS requirements of the user. The weights for the QoS requirements attributes that have been identified as important in the field of animation for rendering are collected from the user, as the importance of the QoS attributes may differ based on the type of project and the user requirements. For example, an animation student may give higher weights for cost
compared to rendering deadline, whereas; a professional animator would assign more weights to rendering deadline than cost. Once the user assigned weights to the QoS attributes is collected, the functional requirements matched list of services at the first level are considered as input for ranking them further based on the Multi criteria QoS requirements of the user.

If the Service Providers (SPs) are new and have not registered with the cloud broker by providing their promised QoS data, whereas the QoS data of other matched services in the list is available then, the services without the QoS data are rejected and the other matched services with QoS data are considered for further ranking.

However, if all the services in the functional requirements matched list is not registered and none has the QoS data, then the functional requirements matched list of services are provided at the level 1 of recommendation to the users.

If the service provider considered for ranking is new but have registered their promised QoS data, then the promised QoS data are used for ranking services using any of the three MCDM ranking methods chosen. However, if the service provider is not new and the monitored QoS data of the SP are available to the cloud broker, then monitored QoS data are used for ranking the services using the MCDM method and a second list of services that satisfy both the functional requirements and the QoS requirements of the user are obtained at this stage.

The third stage of recommendation filters the second list of services further based on the rating given by the user for Quality of Experience (QoE) attributes like learnability. The rating scale considered is from one to five for each QoE attribute is rated and if the user has not used and rated the service, it is assigned zero rating. At this stage, it is checked, if the services in the second list have been already used and the Quality of Experience (QoE) attributes has been rated by the user. If the user has not used and rated any of the services in the second list, then the second list of services itself in their ranking order are recommended to the users. However, if the user has rated some of the services and has not used or rated any services, then the second list of services is further re-ranked using the proposed re-ranking method that is based on the content based filtering. The content based filtering algorithm recommends items that
are similar to the ones that the user liked in the past. Hence the second list of services is re-ranked based on the QoE rating of the user and the re-ranked list of services is provided as the third ranked list of recommended services.

In order to provide opportunity for other services as well to be recommended and to break the monopoly of highly rated services, the second list of services is re-ranked only if the QoE rating is 4 or 5 and the services are not re-ranked if the other services have QoE rating less than or equal to three. This condition for re-ranking, the services provides scope for the new services that has not been rated by the user also to be recommended. Moreover, since only the user registered with the cloud broker and has used the service could provide the feedback, the problem of false user feedback could be avoided. Thus, since the cloud renderfarm services are recommended in three different stages, we could overcome the cold problem of recommending the new cloud services like the cloud renderfarm services in an effective manner.

3.5 TEST BED SETUP AND EXPERIMENTS

A test bed was set up for conducting the real time experiments by rendering the selected animation files in the cloud renderfarms selected. The animation files of different sizes from the “Big Buck Bunny” project was selected for conducting the real time cloud rendering. The “Big Buck Bunny” project is an open-source computer animated film that was created using the free Blender software by the Blender software development team. As the film has been created under the creative commons license, all the 3D models, texture and image files have been publicly released and are available as an open source. It contains over 400 files in 1.2 Gb for the entire dataset and the files are organized into 13 scenes by top level folders under the project directory. The final animation is for ten minutes and has more than 15,000 image files. The real time experiments were conducted in the Editorial VFX (Visual Effects) Suite in AVM Studio, Chennai and in Pax Animation Studio, Coimbatore. Three animation scenes with three different file sizes were selected for this purpose as given below in the Table 3.1.
Table 3.1 Details of Animation Scenes Selected

<table>
<thead>
<tr>
<th>Job Size</th>
<th>Scene</th>
<th>No of Frames</th>
<th>Input Size (MB)</th>
<th>Avg. Rendering Time (Sec. per frame)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>12.peach/03.blend</td>
<td>28</td>
<td>40</td>
<td>1260</td>
</tr>
<tr>
<td>Medium</td>
<td>01_intro/02.blend</td>
<td>93</td>
<td>90</td>
<td>5837</td>
</tr>
<tr>
<td>Large</td>
<td>02_rabbit/02.blend</td>
<td>91</td>
<td>290</td>
<td>9248</td>
</tr>
</tbody>
</table>

The small job was about 40MB with 28 frames, the medium size was about 90MB with 93 frames and larger size rendering job was of 290 MB with 91 frames. It should be noted here that in large size job though the file size is huge, the no of frames is less than the medium size job, this is because these frames contain many more complex animation and VFX scenes compared to the medium size job. Thus the job size is based on the file size and not on the number of frames in a job. The three different files given above were first rendered in the Local rendering machines with the following configuration to estimate the average rendering time (Sec. Per frame).

Hardware Specification of the system used for Local Rendering

- CPU Cores - 4
- CPU Speed – 2.2 to 2.4 GHz
- Memory - 4 GB RAM
- Free Hard Disk Space – at least 50 GB
- OS – Linux CentOS 6.2

3.5.1 Collection of Real-Time QoS Data

In order to collect the real time QoS data on cloud renderfarm services, the same files that were rendered locally were later rendered in four different renderfarm services chosen. The renderfarm services were chosen after checking whether they could satisfy the functional requirements for the files to be rendered and an example of the functional requirements is given below in Table 3.2.
Table 3.2 Example of Functional Requirements of users

<table>
<thead>
<tr>
<th>Functional Requirement Attribute</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute Unit Type</td>
<td>CPU</td>
</tr>
<tr>
<td>Supported S/W License Fee</td>
<td>Provided and Fee included</td>
</tr>
<tr>
<td>Job Management S/W requirement</td>
<td>Plugin</td>
</tr>
<tr>
<td>S/W support requirements</td>
<td>Cinema 4D</td>
</tr>
<tr>
<td>Render Engine S/W requirements</td>
<td>Mental Ray</td>
</tr>
<tr>
<td>Plugin requirements</td>
<td>FumeFX</td>
</tr>
<tr>
<td>OS requirements</td>
<td>Mac</td>
</tr>
<tr>
<td>Render Node Cost</td>
<td>0.60 $ per core per hour</td>
</tr>
</tbody>
</table>

The four cloud renderfarm service that satisfies the functional requirements and chosen for real time rendering are the following:

- Render Rocket
- Rebus Renderfarm
- Rendering Fox
- Ranch Computing

The data for the selected QoS attributes ie) Deadline Adherence, Pre-Cost Assessment Accuracy, Functionality and the Render node cost were collected.

3.5.2 Collection of Real-Time QoE Data

For the purpose of collecting the QoE data, the three different rendering jobs of the “Big Buck Bunny” project, namely the small, medium and large were rendered in each of the four real time cloud renderfarms selected by fifty different users, who were a mix of professional animators, freelancers and students from the VFX suite of AVM studio and the Pax Animation Studio. Each user was asked to render each of the three rendering jobs in each of the four selected render farms. Once the user has used a particular cloud renderfarm services, they were asked to rate the services used based
on the five QoE criteria or attributes identified on a scale of 0-5 for each attribute using a questionnaire and the average QoE ratings were calculated so that the services could be re-ranked based on the proposed re-ranking algorithm.

The QoS and QoE data, collected using this experimental setup was used for ranking and re-ranking, the services as explained in the forthcoming chapters. The methods adopted to generate the domain knowledge about the cloud renderfarm services and the knowledge based filtering methods applied to filter the cloud renderfarm services that could satisfy the functional requirements of the animators is explained in detail in the next chapter.