CHAPTER 6

MULTI-SERVER SYSTEM IN A REALISTIC SCENARIO

6.1 INTRODUCTION

Recently, there has been an increasing demand for video service applications related to education, information, and entertainment (Minming Li et al 2007). Many universities and colleges are moving towards the concept of virtual class rooms and on-line education. Such advancements in technologies would not only reduce the cost of the services offered to the end users, but also benefit people living in remote places. A recent study reveals that Video-on-Demand (VoD) is a service that will generate one of the largest revenues for private network companies, but which needs performance-effective architecture solutions that provide high scalability.

In an on-demand video system (i.e., videos are displayed upon user request with negligible delay), a number of repository servers such as tertiary libraries or jukeboxes (collectively referred to as a repository) store all the video contents of interest to a large number of geographically distributed users (Hao Yin et al 2006). If videos were to be streamed directly to the users, the user capacity in the system would be limited by the streaming capacity of the repository. Such capacity can be increased by using a hierarchy of servers, in which multiple streaming servers cache the movies delivered from the repository and stream them to the users. If the streaming servers were co-located with the repository, the transmission cost incurred in streaming videos to remote users might be high (Azzedine Boukerche et al 2006). To overcome this problem, as well as to take advantage of the access locality and demand
characteristics of the user pool, the streaming servers may be placed close to
the user clusters, thus forming a “Multi-server architecture.” Such an
architecture is able to achieve scalable storage and streaming capacities by
introducing more repository servers and local servers as the traffic increases.

In a wireless environment users may enjoy ubiquitous
entertainment services (Wade Trappe et al 2003). For example, they may play
games or watch videos of their interest online wherever they are. This chapter
focuses on technologies that enable (VoD) services over wireless networks. A
VoD system is an interactive multimedia system working like cable
television, the difference being that the client can select a movie from a large
video database stored in a distant video server. Unlike other video services
such as pay-per-view (PPV) and video in demand (VID), individual VoD
clients in an area are able to watch different programs whenever they wish to,
not just in time as in VID or pre-scheduled as in PPV. A VoD system is,
therefore, a realization of the video rental shop brought into the home (Ragab
Hassen et al 2007).

A wireless VoD system has many practical applications. For
instance, airlines could provide VoD services in airport lounges to entertain
passengers on their own PDA (personal digital assistant) while they are
waiting for a flight; a museum could provide video information on the
exhibits on demand over the wireless network; in education, a university
could also install such a system on campus to allow students to watch videos
recorded earlier from lectures they were not able to attend (Junghyun Nam
et al 2005).

Depending on how the servers are operated and maintained, there
are three different approaches of VoD: centralized, independent server nodes,
and proxy. The centralized server approach uses only one server with a huge
storage. But the drawbacks of this approach are its single point of failure and
the lack of scalability. The independent servers approach utilizes $n$ different servers; each has a dedicated storage facility. But it suffers from the problem of poor fault tolerance and high storage requirement (Niklas Carlsson et al 2006). The proxy approach exploits the existing web proxies’ capacity to cache media data, which is cost-effective. The scalability of this approach is limited by the bandwidth availability of the centralized server. In a Multi-server architecture the workload is divided evenly among all the servers and scalability is achieved.

6.2 SYSTEM ARCHITECTURE

In the Multi-server system, a service is often provided by one or more servers running on different machines with the service functions distributed and/or replicated - primarily for increased availability and performance. The servers are organized as members of a server group which exports the service through a generic runtime interface with the distribution and replication of the functions hidden from clients. An application is then implemented as one or more clients communicate with the server group.

Existing VoD architectures do not support, frequent member join/leave and content exchanges, due to limited processing capacity and network support. Each of the local servers, therefore, obtains its movies only from the repository and operates independently from the other, and the repository has to unicast the movies to the local servers. Furthermore, network limitations such as disjoint networks and limited network capacity also lead to performance instability (Sencun Zhu et al 2007). On the other hand, the Multi-server approach for video streaming applications consists of a network of servers, where each maintains a dedicated local cache. This allows the local server to cache the full video as compared to the partial cache facility supported by the previous schemes. This enhances the availability and performance while minimizing the delay.
Two types of services are possible with the Multi-server network model: local service and remote service. In a local service, a client can directly receive video information from its local server-node. Local service requires only local-network resources and server-node resources. If the video requested by a client is not available in the local service the remote service is used. Thus, a remote service requires the resource of an inter-connection network. In order to minimize the resource requirements of remote services, video-mapping strategies have been developed to calculate the optimal number of replicated popular videos in each node.

**Figure 6.1 Multi-server network architecture for VoD application**

In order to reduce the delay in receiving the data from the repository always, the local servers can be connected by a network so that they can communicate and exchange their data with each other. This will also decrease the long-haul transmission cost of the requested video from the central repository to the local server. It is possible for servers to be co-located in a campus area network, with an entertainment network of cooperative service providers, or a private enterprise network. In general, the transmission cost between the local servers is low, relative to that from the repository. The local cache connected to each local server helps to buffer the requested video in advance. This, in turn, helps the server network to satisfy any subsequent
local request(s) among themselves, hence conserving bandwidth from the repository to the server network. The central repository may multicast movies to the local servers, or it may unicast the movie to a particular local server, which in turn, unicasts/multicasts the movie to the other servers using the network between the local servers.

6.3 RESULTS AND DISCUSSIONS

In applications where there are many users and frequent additions or deletions to the group membership, key management schemes can reduce significantly the communication burden. In this thesis, an attempt is made to reduce such a communication burden by improving their efficiency in the face of QoS parameters using a Multi-server approach. Having met the basic design goal, there is a need to examine the performance of the Multi-server approach in realistic scenarios.

Our simulation model consists of a single main multimedia server and a set of streaming servers. Each streaming server is connected to the next level of servers which are interconnected among themselves using the IEEE 802.11. The following are the assumptions made in our simulation model. The user requests for the video that follows the Zipf law of distribution. The sizes of the videos are uniformly distributed over a range, and the number of streaming servers in each group is assumed to be same. The values considered for simulation are as follows. The sizes of videos are 350MB to 550MB and each simulation is carried out for 1500 seconds. The performance parameters are client waiting time for the requested videos and number of videos served.
6.3.1 Waiting Time

Waiting time is the initial latency experienced by a customer before watching the video. It is certainly a significant parameter from the client’s perspective, which determines the QoS in a VoD system. The performance analysis on waiting time as a function of the number of requests for various VoD schemes is shown in Figures 6.2, 6.3, and 6.4. The numerical comparison in terms of initial latency in all the strategies is shown in Table 6.1. Very popular videos are batched and then only serviced, hence incurring initial latency. A majority of requests are for very popular videos, hence initial latency is observed in the graph. It is evident from the graph that in the Multi-server approach, initial latency is nearly uniform irrespective of the number of requests, whereas other schemes show a significant increase in initial latency for a large number of requests due to an inefficient buffer management technique implemented in the local server.

<table>
<thead>
<tr>
<th>No. of requests</th>
<th>Client waiting Time(s) Centralized</th>
<th>Client waiting Time(s) Independent server nodes</th>
<th>Client waiting Time(s) Proxy</th>
<th>Client waiting Time(s) Multi-server system with K=2</th>
<th>Client waiting Time(s) Multi-server system with K=3</th>
<th>Client waiting Time(s) Multi-server system with K=4</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>201</td>
<td>156</td>
<td>52</td>
<td>55</td>
<td>35</td>
<td>21</td>
</tr>
<tr>
<td>100</td>
<td>235</td>
<td>159</td>
<td>64</td>
<td>76</td>
<td>38</td>
<td>26</td>
</tr>
<tr>
<td>150</td>
<td>288</td>
<td>158</td>
<td>86</td>
<td>45</td>
<td>40</td>
<td>39</td>
</tr>
<tr>
<td>200</td>
<td>321</td>
<td>162</td>
<td>98</td>
<td>58</td>
<td>46</td>
<td>41</td>
</tr>
<tr>
<td>250</td>
<td>325</td>
<td>162</td>
<td>134</td>
<td>62</td>
<td>51</td>
<td>47</td>
</tr>
</tbody>
</table>
Figure 6.2 Waiting time for Multi-server system with K=2

Figure 6.3 Waiting time for Multi-server system with K=3
Figure 6.4 Waiting time for Multi-server system with K=4

A Multi-server system achieves negligible initial latency by means of local streaming servers. This advantage, however, comes with the cost of additional servers and storage.

6.3.2 Number of Videos Served

The performance analysis of the number of requests serviced as a function of time for various VoD schemes is shown in Figures 6.5 through 6.7. In the earlier stages (till 30-60 seconds), all the techniques serve the requests and none of them has completed the service. Later, the number of requests served grows linearly in all the schemes. Since local streaming servers are used, more number of requests are processed in the Multi-server system and hence achieve significantly greater throughput than all the other schemes. Requests are served faster using the available resources and the resources which enable other requests to be served are released. For example, at time 270, in the Multi-server scheme with K=4, over 250 requests are served, when compared to the other scheme where only 135 requests are served, which clearly indicates that in our approach more than 90% of the requests are served when compared to the centralized server scheme.
Table 6.2  Comparison of Number. of videos served for various VoD schemes

<table>
<thead>
<tr>
<th>Time(s)</th>
<th>Centralized</th>
<th>Independent server nodes</th>
<th>Proxy</th>
<th>Multi-server system with K=2</th>
<th>Multi-server system with K=3</th>
<th>Multi-server system with K=4</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>35</td>
<td>31</td>
<td>35</td>
<td>54</td>
<td>55</td>
<td>61</td>
</tr>
<tr>
<td>60</td>
<td>41</td>
<td>33</td>
<td>41</td>
<td>59</td>
<td>64</td>
<td>65</td>
</tr>
<tr>
<td>90</td>
<td>46</td>
<td>47</td>
<td>55</td>
<td>61</td>
<td>68</td>
<td>71</td>
</tr>
<tr>
<td>120</td>
<td>49</td>
<td>66</td>
<td>79</td>
<td>62</td>
<td>99</td>
<td>112</td>
</tr>
<tr>
<td>150</td>
<td>62</td>
<td>95</td>
<td>106</td>
<td>89</td>
<td>154</td>
<td>134</td>
</tr>
<tr>
<td>180</td>
<td>81</td>
<td>156</td>
<td>182</td>
<td>168</td>
<td>198</td>
<td>218</td>
</tr>
<tr>
<td>210</td>
<td>135</td>
<td>201</td>
<td>212</td>
<td>196</td>
<td>201</td>
<td>257</td>
</tr>
<tr>
<td>240</td>
<td>199</td>
<td>255</td>
<td>270</td>
<td>245</td>
<td>278</td>
<td>298</td>
</tr>
<tr>
<td>270</td>
<td>201</td>
<td>289</td>
<td>301</td>
<td>285</td>
<td>325</td>
<td>355</td>
</tr>
<tr>
<td>300</td>
<td>225</td>
<td>321</td>
<td>356</td>
<td>311</td>
<td>329</td>
<td>413</td>
</tr>
</tbody>
</table>

Figure 6.5 Number of videos served for Multi-server system with K=2
Figure 6.6 Number of videos served for Multi-server system with K=3

Figure 6.7 Number of videos served for Multi-server system with K=4
6.4 SUMMARY

Thus the performance of the Multi-server system for various K values as the evaluated in the VoD scenario. The results have shown that the proposed Multi-server scheme performs better in terms of initial latency and number of videos being served, compared to the other existing schemes.