Paper Analysis

An alternative paradigm for Scalable On-Demand Applications: Evaluating and deploying the Interactive Multimedia Jukebox

After the H.264 paper, this paper was a welcome change. It was an easier read, had better explanations for the features incorporated in the product and was not just a long list of facts. Another difference between the two papers is that this paper is driven more by commercial and user needs. The focus of this paper is the Interactive Multimedia Jukebox (IMJ), a paradigm for programmed video delivery to multiple users simultaneously. There are two opposite types of video delivery paradigms: Video on Demand (VoD) and cable TV. IMJ is somewhere in between, offering both the services to an extent. The authors argue that VoD like systems have failed because of high costs of deployment and very less demand. Hence, I feel that the motivation behind this idea is more commercial than technical which is pretty much highlighted by the paper's content, which has less technical content and more analysis of IMJ's operation and customer experience, depicting its commercial feasibility. (I could be wrong though!)

The authors start off by telling us what lacks in true VoD and other broadcast video systems. There were a few, but big drawbacks with these systems:

- Scalability: True VoD systems require reservation of resources for each user. Clearly, with a large number of users, a large number of resources would be required and hence the solution is not scalable.
- Program Scheduling: True VOD offers the highest degree of viewer scheduling control. On the other hand, broadcast systems (cable TV) offer no real time scheduling control.

So, intuitively even though VoD looks like a very lucrative option for service providers, the service providers were not very keen to implement it on a large scale basis as it offered little financial benefits. Using these thoughts, the authors conclude that there are two factors based on which video service paradigms can be compared:

- Number of viewers who can watch a particular program stream (direct outcome of scalability)
- How much viewer input is considered in the scheduling of the programs

The idea of using a graph to explain the comparison between the paradigms is brilliant and summarizes clearly the main differences based on the above two factors (each represented on the two axes). The jukebox paradigm lies in between the two extremes of the graph and is all about trade-off between real time and non real time user interaction.

The key design principle here is to have flexible scheduling on a finite set of channels available to all viewers. So, there are a set of channels which are multicast to viewers who tune into those channels. Then if a user requests some program, his request is scheduled on one of the channels and lastly, the users are provided a schedule of the currently playing programs so that they can tune into a channel of their choosing. Thus, even with a fixed number of channels, jukebox achieves scalability by adding users to the multicast group of the channels. Even though I agree with the authors' view that this paradigm is scalable, I think it comes at the price of customer satisfaction. In fact, the authors mention it too that there is a trade-off here with the viewer wait time increasing. I will go into it further when I discuss the results of the IMJ deployment.

The other key component of the system is user request scheduling which is very much like the scheduling of processes to be run on a processor in a computer. Some policies for the same have been proposed in the paper:

- Shortest wait scheduling: Schedule a viewer request on the channel with the shortest wait time. I think this policy can be implemented in conjunction with another to give an optimum result.
- Content based scheduling: Scheduling based on content looks like a good result, but it could lead to lopsided waiting times. For example, during sport seasons, the sport channel would be pretty clogged up (there would be more number of viewers) whereas other channels would be relatively free. Scheduling a sport program request on the same channel would lead to way more waiting time than scheduling it on a free-er channel.
- Service provider scheduling: the service provider can schedule programs according to its own policies.
- Vote based scheduling: this scheduling is based on viewer votes (for programs) but it suffers from the possibility of starvation where some programs may never get scheduled to play.

I think that an ideal scheduling algorithm would be quite hard to achieve as a lot depends upon the viewer choice which is fairly dynamic and a bit unpredictable. But a mixture of a few of such policies could achieve a satisfactory result. In the simulation experiment of the system, the authors have used the 'voting scheme' and show results by changing the parameters of the scheme. It would have been very useful to see and compare the performance of the system with other schemes too (of course with the side effect of more time consumption perhaps for not much interesting results).

Next, the authors describe the architecture for a jukebox system and then the prototype that they developed: the IMJ. The architecture of the IMJ (and hence the architecture proposed for a generic jukebox system) consists of:

- Scheduling control and schedule display component: a web page for the user to select the program from.
- Network: the IMJ uses the MBone network for delivery of videos.
- Receivers: the MBone tools provide the decoding, display and delivery functionalities.
- Video server: the server which transmits the video to the receivers.

The authors also mention the type of content they had in the content library. Since the performance of such a system is highly dependent upon the content provided, it was a very good idea to describe what the system had. It is another measure for the performance of the system. However, from what the authors have described, I feel the system really lacked a good content (which again is highly difficult to obtain). So, I believe that the results to follow would be somewhat short from the actual actual user behavior.

The tracking of the usage of IMJ has been looked at more from the commercial point of view than research point of view. The data has been collected from three sources: program requests, jukebox schedule and the program viewers. The first graphs explore the pattern of program requests, page hits and channel viewers. There are a couple of interesting things to note here. The first spike in the graphs is when the IMJ was announced. So I gather that the data before that (program request and WWW page hits) must be from experiments and testing the software. So why are there no group joins (channel viewers) before that period? Was this not tested? Secondly, there is a period of inactivity near about August '97 with no group joins and no WWW page hits which has not been explained in the paper anywhere. Was it because of lack of content? Or some failure which could bring out interesting results? Some information on the aspect would have been helpful. Also, just out of curiosity, why is 'WWW' used in every place where the word 'web' could be used: WWW page, WWW site, WWW request ? Is it just a 90s terminology? Or is there some difference which I have not noticed?

In table 2 (and its description), it is observed that the most frequent users have hit the 'web' page 807 times and have made about 27 program requests. If I am not wrong, does it mean that people accessed the program list on the web page for as many as 807 times but made a request only 27 times? What could be the reason for such an aberration? Lack of content or lack of time with people accessing the MBone only at work? Also, a measure of customer satisfaction could have helped too (since the prototype is also oriented at evaluating the commercial performance). The metric is however a bit hard to evaluate but maybe could have been derived from the length of time a user maintains his session? That is, a more satisfied user would watch the whole video and a less satisfied one would close it before it is completed.

The authors next present a simulated performance of a large scale jukebox system which is highly important because a small prototype (IMJ) would not provide a satisfactory evaluation of the system. The parameters, nominal values and range of tested values have been stated explicitly and thus help us understand the simulation environment clearly. 150 requests per hour, 100 channels and 500 programs is a good rough estimate of a real system and the key measure is the average waiting time for a request. The system performs reasonably well under the nominal conditions (ave. wait time of 11.7 mins is agreeable). The next set of results point to an expected behavior: more the number of programs available, more the waiting time; more number of channels, less the waiting time. However, customer satisfaction would be directly proportional to number of available programs and inversely proportional to waiting time. Thus, these results draw a paradox where both cannot be simultaneously achieved. Perhaps a well calculated 'sweet spot' could be a found? Next, the impact of two different scheduling policies is evaluated: voting and alternative choices. Voting is a bit iffy proposition as it may lead to starvation of programs which do not receive the threshold amount of votes. Thus, it is quite possible that some customers feel short changed because of it. Even the graphs of figure 10 indicate that the waiting time is too large (somewhat close to 100 mins). So the interesting question is as to how beneficial is this system? If someone requests a program and doesn't know when he is going to be able to see the program (while he is scheduling it), it is bound to create frustration. Wouldn't it be much more

beneficial to know before hand when a program is going to be scheduled (as in pre-scheduled cable tv)? This uncertainty in the waiting time would certainly hamper the popularity of the system.

Lastly, the authors provide an overview of what additional services could be provided with the jukebox including interactivity (ability to seek in a video), distributed server and interface (the servers could be distributed to avoid high loads), advanced scheduling policies (which would take into account the program length or if some program can fit into an empty time slot) and service pricing (various policies that can be implemented for pricing).

In conclusion, I would say that even though the problems are very real and the idea very interesting and intuitive, I am not very convinced to the feasibility of the system, mainly because of the uncertainties involved with the scheduling of the programs. A real VoD holds more lucrative attraction as it gives the user total control and definitely more attractiveness. I highly doubt if the system would be of much value in present times with such high bandwidth and processor power available (and not many ISPs providing support for multicast networks), but I am sure the work was very relevant to the world of 1997 when these ideas were fairly novel!