Open load multimedia server balancing

Gloria García Fernández, Jesús Soto Carrión, Luis Joyanes Aguilar, Ismael Muñoz Collado
Departamento de lenguajes y sistemas informáticos
Universidad Pontificia de Salamanca
Paseo Juan XXIII, nº 3
Spain (Madrid)

Abstract: - One of the main challenges a multimedia server must face is serving to its users in a fast and efficient way. In that sense, it is highly important to develop an appropriate work load management. New advances regarding semantic web technologies are fostering the design of a new ontology. This ontology will allow the description of new state registers, as well as new functionalities applied to searching for certain resources through several sources. In that way, different servers placed in different locations can cooperate in order to provide a higher quality service. In this essay a load distribution protocol proposal through semantic web technologies applied to multimedia servers will be shown. In addition, an user-friendly and transparent resource search schema will be depicted.

Key-Words: - Semantic web, multimedia server, streaming, video, load management.

1 Introduction
Today, the access to multimedia contents can be done via a huge range of devices: portable computers, cellular phones, pdas, and personal computers. This means that the requirements for a video content to be correctly visualized are deeply heterogeneous, specially, when it comes to guarantee a high-class service to the clients. Inside a session, with several servers working at the same time, when an error appears the client is redirected to another node. This server must be able to provide to the client the same service in the same way the previous one was doing. However, this transition cannot be transparent. Thus, it implies an interruption in the transmission which causes a loss of information, or even the need to visualize content that already had been seen. To overcome this trouble, through this essay the use of SFOMS (Server Friends Of Multimedia Servers) will be proposed, which working based on FOAF (Friend of A Friend Project) will be able to deal with load sharing from the client point of view.

2 Problem Formulation
When a client asks for a multimedia content, it opens a connection against a server devoted to develop the content transmission. Inside this transmission several issues (i.e. network jam, link quality) play their role over the quality of the information that is being transmitted. Concerning a network with several node servers, they can operate in three different ways [2]:

- Proxy at Server: Each node acts as a true server.
- Proxy independent: A server takes care of both user validation and load sharing. Inside this kind of architectures, when a node fails the clients must ask for a new connection to another server (put it simply, the fail may be not transparent for the client).
- Proxy at client: The clients must carry out with the transmission management.

2.1 Streaming servers features
There are two main features in a multimedia service, which directly can be applied to a streaming service:

- Scalability. In case of a burst (or a deep decrease) in the number of clients accessing to the service, the servers must be capable of readjust its capacity without redrawing the whole architecture of the service.
- High availability. The servers must guarantee the clients the access to the service within the optimal conditions.

In terms of stream transmissions inside a proxy-based architecture, a load balance can be developed so that the servers within the schema can share the tasks. However, when the client itself asks for a server change, it must send an explicit request to complete the server transition.
2.2 Difficulties to beat

The two main difficulties are [1]:

- **Network capacity issues** When it comes to a multimedia service it is hard to evaluate the traffic volume a streaming transmission can suppose. This happens because the stream size (both in bandwidth and duration) may change depending on its content. So, it is critical to provide the clients with an adaptation dynamic mechanism.

- **Obstacles due to unguaranteed quality of service** Generally speaking, wireless devices must work in networks unable to guarantee a fixed quality of service (with the exception of GSM/UMTS circuit domain). This implies that both the bandwidth and the delay the user is given are supposed to change based on the network load. The adaptation to this parameters is critical. However, if bandwidth decreases during a long period of time, may sound reasonable jumping to another server, with the purpose of getting a higher quality of service.

2.3 Semantic Web – RelationShips between Objects

Semantic web new specifications allow the definition of ontologies. These ontologies allow the establishment of a common vocabulary described in an open standard. Moreover, machines are able to process this vocabulary. Using this specifications relationship semantic network between different objects can be raised.

Image 1 shows existing relationships between different people who publish a resource description file on their web servers (FOAF scheme with relationships defined by Eric Vitiello[6])

![Fig. 1 – Foaf Relationships](image)

Project FOAF [4] has designed a vocabulary that allows the description of somebody, his/her relationships with the other people, and the activities he/she plays. This is a powerful tool used to create personal web pages that can be processed through a machine. In addition, it can promote the creation and administration of virtual communities. In order to do that, personal details are specified over a common scheme based on the semantic web standards. Each single user can search for common hobbies with others, be able to know which is the latest published article, or the last project developed by somebody, as well as many other inferences. Social networks have taken advantages from these technologies [5]. The Semantic Web provides new rooms that foster the interoperability between different systems, as examples, in [9] are defined security resource management mechanisms over the semantic web or in [7] are showed a new scenario for the flexibility in a learning object repository.

Keeping these ideas on mind, in the present research an open communication scheme between several multimedia servers is raised. Each server publishes its resources and its backup servers in order to develop load sharing.

3 Problem Solution

As soon as a client begins a streaming transmission, it will ask for the rest of the nodes to the reached. In other words, the client will be given the “friend servers”, in order to be prepared to jump to another server when the communication quality decreases. The client must warn about the minimum number of friend servers needed. In that way, a transitive search within the related servers will be done if the asked server is not able to provide that minimum number. The search is carried out through a SPARQL [8] over RDF files request. The code below is given as an example:

```
PREFIX mpeg7: http://rhizomik.upf.edu/ontologies/2005/03/Mpeg7-2001.owl

SELECT ?resource
WHERE{
?track rdf:type mpeg7:VideoType .
?track mpeg7:Creator ?Author .
?Author mpeg7:Name “Almodóvar” .
?track mpeg7:Title “Volver” .
}
```

Once the client knows the servers which are able to work with, it will keep on asking about their status so as to have information about them as updated as possible (OWL file). The server also must update this information at defined time intervals. When the client connects to a new server, session parameters must be agreed in order to satisfy the needs of the client. Finally, when the client detects a decrease in the quality of service that is enjoying, it
will perform a new search to locate the most suitable server.

3.1 Server parameters:

There are several parameters that must be beared in mind to determine the status of a server. They are [3]:

- CPU usage. The server must be capable of create new streams.
- Memory. Should the server fall in pagination, it is mandatory to have free memory to deal with this situation.
- Disk and network. Their usage determines the quantity, from the available bandwidth, that is being used for a certain task. As far as stored content is concerned, the capacity on disk is critical because the content must be searched. Network marks the maximum speed for the server to communicate with the client, to be precise, the less bandwidth available, the poor quality of service may be offered.

3.2 Client Parameters

From the client side, there is a number of parameters that must be taken into account in order to trace the quality of the communication [3]:

- Play faults: i.e. lost streams.
- Client rebuffering: In certain situations the client needs to stop the transmission to refill the buffer. This is a symptom of an overloaded server unable to go on with all the requests received. Rebuffering usually happens just before the play fault.
- Amount of time between the start request and the start of the transmission: The client, as a general rule, will not take notice of this time, because it is hided in the initial buffering. But if the time is excessively long it implies that the server is unable to serve new requests.
- Decrease in the store percentage: This means that the client is being served the stream in a poor codification than the server has. On the whole, the client should not start rebuffering until getting all the information. However, giving a codification able to degrade the quality of the stream, the video will keep being visualized but in low quality. From the server side this situation force this node not to send certain frames. In other words, it will delete the frames considered less important because of its overload. In other circumstances the server will be dealing with a video codified with different qualities, defined as different layers. Each layer adds something extra to the video so, if the server is overloaded, it will start sending the first layers (which are the essentials), leaving apart the others. The client will be able to visualize the video, but in low quality.
- Jitter: When the streams are unable to arrive at their precise instant (they must arrive in order and within a fixed period of time) jitter appears. The trembles in an image are the jitters of the streams. As jitter become bigger rebuffering can be invoked. In extreme situations jitters can force a stop in the visualized video.

4 Open Multimedia Server Cluster Ontology

The class oc_DigitalRecord represents the Digital Register abstract concept described in OpenCyc. All the players within a connection are defined by oc_Agent class, also described in OpenCyc. Multimedia servers are defined by MultimediaClient class, and so are servers in MultimediaServers class. Inside the load balance protocol definition there are some restrictions that must be followed. These are defined in descriptive logic over the ClusterNode class: ∀ has_neighbour MultimediaServer, ∃ has_state MSGeneral_State

All properties that surround a multimedia communication have been conceptualized in the MultimediaCommunication_State class. Both the client and server are linked with the has_active_client and has_active_server properties respectively. Furthermore, in order to associate all the multimedia servers that store the resource that has been requested by the client, has

---

1 Upper Ontology - www.opencyc.org
been defined the multiple property has_background_server. The abstract representation from which all metadata multimedia resources derive has been defined as the MultimediaResource class. The multimedia ontologies as mpeg7 [10], MusicBrainz 2 or Simac3, can be used to make a detailed searches using the terms and properties described in this ontologies.

Figure 2 – MSGeneral_State and Session_State class

5 Protocol

1) The client opens a connection against a server (ACON), in order to start a multimedia session. The number of background servers desired to guarantee the quality of service must be reported too.

2) The server verifies that is able to provide the demanded resource. At the same time asks its neighbours about this resource.

3) A session datagram including background and active servers is sent to the client. The client opens a connection against all the background servers.

4) Should an error happen (i.e. a quality of service decreasing) the client requests the activation of the best background connection (stored in the MSGeneral_State register specified on the ontology). To do that, it must send a transmission initiation request (because the connection was already opened).

5) The selected server accepts the session start, in addition it will add the client to its receiver list. In case of a server not accepting the session start, the client will perform several requests (based on the precedence record stored) through the rest of the servers.

6 Conclusion

During a video transmission, optimizing the received quality of service by the clients is getting more importance nowadays. A communication protocol which allows servers to cooperate has been proposed through this paper. In this protocol servers at first have no connections among them, but they will be able to develop a cooperative relationship in order to offer the best quality of service.

7 References

[8] SPARQL, Query Language for RDF. http://www.w3.org/TR/rdf-sparql-query/

2 MusicBrainz Ontology-www.musicbrainz.org
3 Simac Ontology:www.semanticaudio.org