Report on the visit to the University of Klagenfurt[*]

Alon Kama

Computer Science Department, I.I.T. (Technion)

February 8, 2005

1 Introduction

This document reports on my visit to the University of Klagenfurt, Austria, between January 30 and February 5, 2005. The visit's objective was to exchange knowledge on our respective research and to find areas in which cooperation and joint work may be fruitful. An additional objective was to strengthen and foster ties between our research communities, in order to facilitate future joint research ventures.

During the visit I met with Prof. Laszlo Boeszoermenyi and Christian Spielvogel, who is pursuing a PhD under Prof. Boeszoermenyi's direction. I also had the opportunity to meet Prof. Hellwagner, Claudiu Cobarzan, and Adrian Sterca, and to share my research with them.

2 Proxy-to-Proxy Multimedia Distribution Service

Christian Spielvogel, under the direction of Prof. Laszlo Boeszoermenyi, is working on defining a new approach for dynamic content distribution networks. The concept, named *proxy-to-proxy*, is slated to be more dynamic than current distribution networks by allowing a constantly changing set of proxies to serve the content.

The proposed system is composed of a web server, a naming service, multi-purpose servers that will cache the multimedia content, and the clients

[*]This visit was supported by the Middleware for Network Eccentric and Mobile Applications (MiNEMA) programme of the European Science Foundation.

who request this content. The multi-purpose servers, herein also called proxies, form a dynamic set of available resources to the system and are utilized in multiplexing new multimedia content into separate stripe units; storing and caching these stripe units; and finally joining them together and streaming the re-multiplexed data to the clients.

My arrival in Klagenfurt coincided with the team's deliberation as to the distribution of the tasks and responsibilities among the system's entities. The method and extent of communication between the geographically far-flung entities was yet to be determined. Due to my experience with group communication middleware, as well as the designers' wish to someday extend the system to serve mobile clients, I was asked to consult.

2.1 Communication Patterns

Christian Spielvogel and I drafted the general outline of the communication patterns. In a complex system such as this, which vies to be expandable and dynamic, it is important to decide in advance on the extent of synchronization among the entities.

The system has two static and presumably powerful centralized servers: the web server and the naming service. All clients approach the web server in order to request content. The web server surmises the approximate geographical location of the client using ICMP packet data and redirects the request to a proxy server that is nearby the client. In order to find such a proxy server, the web server will consult with a naming service, where all proxy servers register upon joining.

The naming service keeps track of the presence of the servers, as well as important data such as the content they are currently caching, the available bandwidth, and other network measurements as periodically related by the servers. Given this information, the naming server can choose the most appropriate proxy server to service the client request.

Within each geographical area, there may be several proxy servers. One of them is arbitrarily chosen to be the leader, and becomes the spokesman for the group vis-à-vis the naming service. The leader is charged with maintaining connection information about all the server, as well as assigning application-specific tasks such as striping the data, choosing who among the servers will service a client request, etc.

From the above description, as described to me by Christian, I opined that this system exhibits a hierarchy of communication, with potentially hundreds of small geographic groups and many leaders who communicate with a centralized server. The desire to use our group communication toolkit, *JazzEnsemble*, can help in group membership management within the small group, in order to detect changes. This is important for fault-tolerance, as will be discussed in the next section.

However, there may be technical obstacles in using JazzEnsemble for the global communication. JazzEnsemble was designed for collaborative communication among multiple processes, all of which are interested in sending and receiving information to the entire group. Obviously this involves significant overhead that becomes unnecessary in this scenario, as typically the communication will be one-to-one and not many-to-many. Also, JazzEnsemble is geared to work within the confines of a small number of nearby LANs, where the distribution in this case may span several hundred LANs. While JazzEnsemble can be configured to pass messages across several LANs, it may cause the unnecessary flooding of messages that will ultimately not be intended for the majority of reachable nodes.

While Christian initially thought that leaders would need to exchange information among themselves as to the current cache content, I am of the opinion that this would be an expensive operation that would be difficult to scale. Because of the necessary use of a centralized server, I think that the cache distribution as well as the assignment logic should be kept in the naming service instead of being gossiped around by regional leaders. Gossiping would create an unnecessary burden on all servers, as they would have to maintain and update cache information that may not be relevant to their tasks at hand. In a dynamic environment, where servers connect and disconnect frequently, this would bear a high overhead.

2.2 Fault Tolerance

The proposed system is hinged on the voluntary participation of a set of servers around the world. With the same ease of joining, a server may decide to leave the group and thus the content it had stored would be inaccessible. The Klagenfurt team had planned to address this scenario but at the time of my arrival had not gone into specifics.

My suggestion was to initially decide on an acceptable fault-tolerance and recovery parameters, such as “at most X servers leaving during a period of Y minutes.” This would set the ground rules for how the data and entities need to be replicated and what is the acceptable recovery time after detecting a failure.

Christian and I went over the entities and their specific jobs and specified the needs for replicas. We left out the web server and the naming service because those could potentially be implemented using commercial software that has built-in replication capabilities[1]. Under the assumption of only one failure at a time, we came up with the following strategy:

- A *data distributor (DD)*, who is the group leader and also responsible for receiving new content from video servers, will have a *logging DD (LDD)* that will log the demultiplexing decision and the transfer of new content into the caching servers.

- A *data manager (DM)*, who caches a slice of the multimedia content, will have a *replica DM (RDM)* that will actively store a copy of the same data and, in the event of the failure of the DM, will continue serving this data.

- A *data collector (DC)*, which is charged with recombining the slices into a stream for the client, will have a *logging DC (LDC)* that will record the extent of transfer to a client. Upon a failure of the DC, the LDC will continue to stream to the client from the last known position.

In this scenario, JazzEnsemble can serve a useful purpose in coordinating the communication within a regional group. The members can decide among themselves who will replicate or log their activities, and they will all be notified of a change in the membership view. While it may be possible to code a new failure detector and leader/replicator logic, it may be easier to initially prototype the system with the services that JazzEnsemble provides.

2.3 Design Opinion

I was encouraged to find that the video proxy research conducted at Klagenfurt can stand to gain from using group communication middleware, and specifically JazzEnsemble, as an enabler for managing communication, synchronization, and replication in the distributed system. The current design calls for several hundred groups that communicate amongst their geographically-close members. For these groups, JazzEnsemble may be utilized for initial prototyping and ramp-up but may not be the best solution at a more advanced stage of development. The more suitable use for group communication may be in the communication between the web server, naming service, and their respective replicas. It will also prove useful if the communication model is expanded to hierarchical communication, where regional

[1] Incidentally, many such commercial systems use group communication in order to implement failure detection and replication. If the team decides to implement these components themselves, they could certainly do it with JazzEnsemble

leaders communicate amongst themselves in order to share knowledge on the state of the system.

I am also encouraged about the future plans of the researchers to extend their scope into the mobile wireless sphere. In this realm the networking issues would be more complex, and JazzEnsemble would aid greatly in abstracting them away. I look forward in collaborating with the team when they reach this stage, as the issues that will rise will overlap the current issues that I am addressing in my research.

3 Academic Cooperation

My visit helped to introduce the topic of group communication to the teaching staff of the Information Technology department. I gave a tutorial on the general functionality of group communication middleware, and a more in-depth tutorial on the JazzEnsemble package. The intent is to integrate this topic into the curriculum of the Distributed Systems course that is taught in Klagenfurt. By incorporating the topic into lectures and assigning projects or homeworks using JazzEnsemble, the students will gain a better understanding of the strength of such middleware.

4 Final Remarks

I wish to thank my gracious hosts in Klagenfurt, who were happy to share their research with me and were excited to hear about my academic interests. I was made to feel at home and thoroughly enjoyed my visit. I hope that I may repay the favor in a similar setting at my home institute.