Using a Distributed Shared Memory for Implementing Efficient Information Mediators

P. Dechamboux [1] , D. Hagimont [2], M. Lopez [1]

Laboratoire IMAG-LSR, 2 av. de Vignate,
38610 Gières - France
Internet: Daniel.Hagimont@imag.fr

Abstract: With the development of the INTERNET, high performance information servers are made available to clients distributed over the world. Hardware configurations for such servers are, in most of the cases, centralized, therefore costly machines that need to be replaced when the server must scale up.

While a distributed hardware configuration is scalable, running a distributed information server requires adequate system support. We believe that a Distributed Shared Memory (DSM) system support is best suited to address this problem, since it provides the abstraction of a virtual shared memory multiprocessor.

In order to validate this point, we consider the use of our DSM system for supporting a scalable Mediator, an information server that gives access to heterogeneous databases distributed on the INTERNET.

1 Introduction

With the rapid development of computer-based distributed information infrastructures, an increasing number of industrial agents rely on their ability to access information stored in data sources distributed over the INTERNET. Since the variety and the size of these sources is increasing dramatically, the need appeared for dynamic virtual database systems that provide a much more synthetic (and therefore usable) view of these databases. In the following, such virtual databases are called Mediators [Carey 95] [Tomasic 95].

In such an infrastructure (figure 1), the Mediator provides clients with access to the virtual database. A request to the Mediator may be dealt locally by the Mediator (according to the information stored at the Mediator site), or may also require fetching additionnal information from the remote databases. In the latter case, the additional information is processed (synthetized) by the Mediator and possibly stored locally. This organization of distributed information is also widely applyied to data warehousing, but under different conditions and with different goals.

---

[1] Bull S.A. Grenoble

[2] INRIA Rhône-Alpes

Based on this infrastructure, it is obvious that the Mediator site will be a bottleneck of the global system, requiring a very powerful host hardware. This power is very expensive if we only consider centralized computers. Therefore, we believe that such an information server should be managed on a cluster of machines connected through a high speed local area network. The advantages of distributed architectures are: the computing power, the scalability and the low cost.

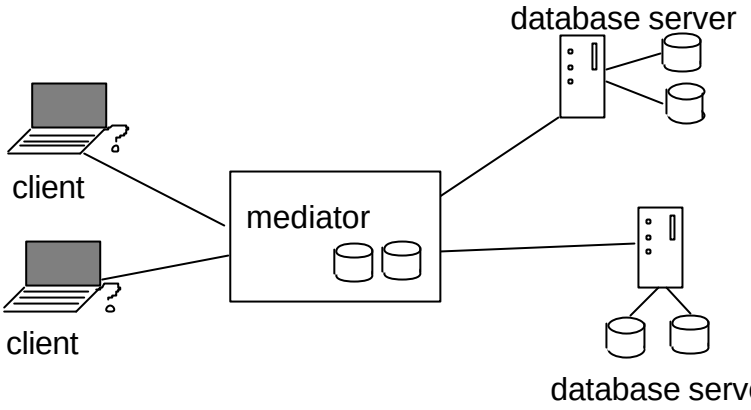

Figure 1. Mediators

In a collaboration between Bull and the IMAG-LSR laboratory, we are currently designing and implementing a distributed system which purpose is to manage a cluster of workstations as a virtual multiprocessor. This new system is based on the Distributed Shared Memory (DSM) paradigm [Han 95]. We are using this DSM service as a basis to support a Mediator.

In this paper, we argue that a DSM system precisely meets the requirements for supporting a scalable and powerful Mediator. The paper is structured as follows. In section 2, we provide the requirements for mediation servers regarding system support. Then, we present in section 3 the DSM-based approach and we analyse its advantages with respect to these requirements. Section 4 provides the state of this work and proposes a workplan for the following.

2 Mediator Requirements

In this section, we present the requirements for managing a Mediator site regarding system support.

As summarized in the introduction, a Mediator site behaves as a server that :

- should be able to deal with a large set of simultaneous queries from remote client sites at the same time,
- should manage synthetized information or metadata (e.g. data source descriptions, data types) locally in order to reduce the response time,
- should be able to simultaneously access a large number of remote databases.

For that purpose, the advantages of a distributed architecture are the computing power, the scalability and the low cost compared to powerful (but costly) centralized architectures.

However, if we want the resources of this distributed architecture to be fully available to the Mediator application, system support must be provided in order to efficiently manage these resources:

- First, the system must be able to balance the load among the machines of the cluster running the Mediator. In this way, it will be possible to efficiently perform queries in parallel on this architecture. Moreover, with the availability of wide bandwidth network for WANs, it will be interesting to manage parallel connections to the INTERNET from the machines that compose the server.
- Second, since all the queries are related to the virtual database implemented by the Mediator, the information from that database that is stored locally (and that need not be fetched from remote databases on the INTERNET) must be accessible from all the machines of the cluster. This is a strong argument for the management of a distributed shared memory.
- This virtual database is managed using the memory resources from the cluster of machines, i.e. the physical memories and the disks from these machines. Indeed, two kinds of data are managed by the Mediator: metadata and synthetized data that need to persist on a stable storage, and data belonging to remote databases that are cached (i.e. temporary data) by the mediator for efficiency purposes.
- This architecture must tolerate a node failure. This has two consequences for persistent data. First, since simultaneous executions may modify this persistent information in the stable storage, transactions must be used for its update in order to keep it consistent. Second, information managed on stable storage must be replicated on disks in order to garantee its availability in case a machine is down.

The previous list is not exhautive, but it includes the main characteristics this system support should gather. Following this overview of the system support requirements, we describe our DSM service and the way it fulfils the above requirements.

3 A System Support for a Mediator

The previous sections pointed out that the Mediator site, which has a distributed architecture, should ideally be viewed as a shared memory multiprocessor in which the shared memory holds in fact the virtual database. In this section, we present the principles of our DSM system and we discuss its adequacy for supporting the Mediator.

Distributed Sharing

DSM systems are based on memory object replication. An object shared between several machines has a local representative or copy on each of these machines. A consistency protocol is responsible for the management of object coherency [Li 89]. This coherency is ensured by invalidation and update operations on the object's copies.

In most of the cases, a reader-writer protocol is applied [Bershad 89]. An object is either in read or write mode. In read mode, all the object's copies may be read in parallel on any host. In write mode, only one copy may be written exclusively to any operation on other copies.

Various update/invalidation protocols may be used, their efficiency depending on the access pattern of the application. In our DSM system, we allow applications to implement the consistency protocol of their choice, i.e. the best suited to their access patterns [Pérez 95]. This is done by developing a new consistency protocol and integrating it in the host Unix system. Thus, each object is assigned a specific consistency protocol allowing it to be managed in an optimal way.

Therefore, in a DSM system, each object managed in the shared memory is made available on each site, giving the illusion of a shared memory multiprocessor. Then, each Mediator's query, that has been dispatched to one node, can be executed locally on that node, the DSM garanteeing information availability and consistency. As mentionned above, our system allows the DSM to be fine-tuned to the needs of the Mediator according to its access pattern. For example, different consistency protocols can be associated with persistent and temporary data (section 2).

Persistent Storage

In our system, objects may be persistent, i.e. managed on stable storage. This storage space is distributed and it is implemented using the disks available on the cluster. However, if we want this storage space to be kept consistent when parallel executions in a Mediator are modifying its content, we need to provide transactions.

The basic mechanism that allows the implementation of transactions is logging. The modifications are registered in a log on disk and then validated in that log. After validation, the modifications can be safely reported to the stable storage. In case of a node crash, the log is used to recover a consistent state. Various logging protocols may be used, differing by the nature of the records managed in the logs. In our DSM system, we allow applications to specify the logging protocol of their choice, i.e. the best suited to their information type [Han 95]. This is done by developing a specialized logging protocol and integrating it in the system. For instance, this feature can be used to implement the most efficient logging polilcy for a Mediator.

Our system also manages object replication in stable storage, in order to garantee object availability if a storage site is victim of a failure. Therefore, all the mechanisms required for the tolerance of failures are provided.

4 Current State and Plans

The DSM system mentioned above has been designed and is currently under development. It is being integrated as an extension of the AIX Unix system, on a network of PowerPC 601 and 604 based workstations. A first propotype including distributed sharing and consistency protocols will be available by the end of 1995, allowing early experiments with database applications. Additional features will be provided in early 1996.

The Mediator that we mentioned in this paper is currently under development by another INRIA team [Tomasic 95]. A prototype of the Mediator on our DSM system support should be available by the end of 1996.

Acknowledgments:

J. Han, A. Knaff, J. Mossière, E. Pérez-Cortés, X. Rousset, F. Saunier contributed to the design of the DSM system. T. Jacquin is a member of the implementation team.

Bibliography

[Bershad 93] B. Bershad, M. Zekauskas, "The Midway Distributed Shared Memory System", COMPCON'93 Conference, PP. 528-537, February 1993.

[Carey 95] M. Carey and al, *Towards Heterogeneous Multimedia Information Systems: the Garlic Approach*, Technical Report, IBM Almaden Research, 1995.

[Han 95] J. Han, A. Knaff, E. Pérez-Cortés, F. Saunier, "Arias: Generic Support for Persistent Runtimes", European Research Seminar on Advances in Distributed Systems, pp. 220-226, L'Alpe d'Huez, April 1995.

[Li 89] K. Li and P. Hudak, "Memory Coherence in Shared Virtual Memory Systems", ACM Transactions on Computer Systems, 7(4), pp. 321-357, November 1989.

[Pérez 95] E. Pérez-Cortés, P. Dechamboux, J. Han, "Generic Support for Consistency in Arias", *5th International Workshop on Hot Topics in Operating Systems (HOTOS-V)*, pp. 113-118, Rosario, Washington, May 1995.

[Tomasic 95] A. Tomasic, L. Raschid et P. Valduriez, Scaling Heterogeneous Databases and the Design of DISCO, Technical Report (N°2704), INRIA, November 1995.