Oléron: Supporting Information Sharing in Large-Scale Mobile Environments

Aline Baggio* Guillaume Pierre*

INRIA Projet SOR
http://www-sor.inria.fr/

Abstract

Oléron allows sharing of small sets of documents among working groups, as well as efficient World-Wide Web browsing in large-scale mobile environments. The target environments provide diverse quality of service, bandwidth, latency, disconnections, storage space, etc. Oléron is designed as a set of extended cooperative proxy caches which provide various mechanisms that deal with this diversity. Configuration is dynamic and depends on the current environment, in order to provide best possible Web access to mobile computers. Oléron is currently in an early specification phase; this article presents its general design.

1 Introduction

Mobile computers are already commonplace and will become even more prevalent in the near future. Their users want to work as normally as possible, remaining within their usual environment. In particular, mobile computers should provide access to the fixed network resources whenever possible. However, wireless networks raise a number of difficulties with respect to that aim: available bandwidth is often limited, disconnections are frequent, and the computers move. For these reasons access to shared information such as files, databases or Web pages will generally be more painful for mobile users.

The problems raised by mobile computing integrated in a large-scale environment can be attributed to three causes: mobility itself, the wireless medium, and the laptop's scarce resources.

A mobile computer cannot rely on statically predefined servers, networks, local resources, etc. [7, 14]; it will be located in new networks and will need to discover new servers, such as file, name, print or cache servers. Mechanisms are therefore needed to dynamically reconfigure resource usage on a mobile computer. A related problem is locating the best copy of a document when it is replicated. A browser should direct its requests to the least costly replica of each document, where cost is measured in terms of freshness, latency, bandwidth, access fees, etc.

Since some computers move, they sometimes access the fixed network via a wireless interface. Such media have low bandwidth, high bandwidth variation, high error rates, and high fees [7]. As computers move, the previously described costs change. Therefore many users will choose to alternate between connected, disconnected, or weakly connected (i.e. connected with a low bandwidth) modes. This will force us to deal with various computing states, in which data can be missing, servers unreachable, etc.

Mobile computers also have restricted power, CPU, memory and storage resources [7]. Storage is critical in stand-alone mode.

To solve these problems in a large-scale information sharing application such as the World-Wide Web, we propose to reuse and extend a number of mechanisms designed for fixed networks. A Web cache, although originally designed for a fixed network can be extended to enable browsing while the mobile computer is disconnected. As it is still in its early specification phase, this article will discuss our general approach.

The paper is organized as follows. Section 2 details our objectives, then Section 3 gives some partial solutions to our problems. Section 4 describes the Oléron's internals. Finally, Section 5 gives some concluding remarks.

2 Oléron

We first describe the expected features of Oléron; and an application scenario to illustrate them.

*Aline Baggio (Aline.Baggio@inria.fr) is a student at the University of Paris VI. Guillaume Pierre (Guillaume.Pierre@inria.fr) is a student at the University of Evry-Val d'Essonne.

2.1 Features of Oléron

The aims of Oléron are: i) to provide working groups with consistency management of their shared documents; and ii) to enable efficient World-Wide Web browsing. Oléron targets large-scale environments which include mobile computers.

Oléron is designed as a set of cooperative Web proxy caches. Each user of a working group will have an access to one of these caches. Users working on fixed computers close to each other may share a cache, whereas each user working on a mobile computer will have one of his own.

Oléron manages the shared documents of the working group. Each cache of Oléron will own the primary copy of a subset of these documents. It will be responsible for allowing its users to modify them, as well as propagating updates to the other caches.

Oléron also enables its users to efficiently browse the Web, while connected to the network as well as while disconnected. For enabling browsing in disconnected mode, Oléron will provide mechanisms for cache control. The user will be requested to give hints about the documents he might access during the disconnection. This means that he may be prompted before the cache removes documents or at load time. It will also be possible to ask the cache to prefetch some documents or document hierarchies that the user expects to access later.

During disconnections, Oléron handles requests with "best effort" strategies. In case of a cache miss, pop-up windows ask the user whether this request must be achieved as soon as possible, can wait and be treated later on, or is unimportant and can be ignored.

Each time computers move, they need to find local resources such as neighbor caches. A related issue is for the caches to locate the nearest replica of the documents they may access. Oléron will be provided with resource discovery mechanisms for handling these issues.

2.2 Application scenario

Let us consider three users (A, B, and C) working together on mobile computers and sharing a set of Web pages. A and B are working in the same laboratory, whereas C is working in a foreign country. Each user is likely to consult or modify a subset of these pages. Each document is owned by a single user, who has the exclusive right to modify it. The document is physically located in the owner's Oléron cache, and there may be secondary copies in the other caches.

At the beginning, the three users are connected to the fixed network. Each user is able to consult the shared documents, to modify the ones he owns, and to browse the World-Wide Web.

Now, A decides to go and see C, for a work meeting. He planned to prepare the meeting during his journey, so he switches his cache to "disconnected mode". He prefetches all the documents he will need: some are documents owned by B or C, and some are ordinary World-Wide Web reference documents. The cache also sends to one of its peers an up-to-date copy of all the documents it owns, in order to make these documents readable by other users. After that, A unplugs his computer from the network, and brings it with him. During the journey, he can consult all the prefetched documents, as well as edit the documents that he needs for the meeting.

During the journey, A requests a document he forgot to prefetch. The Oléron cache handles the cache miss by asking him whether this document is important or not. If it is indispensable, Oléron will try to establish a wireless connection to the public network, whatever the cost, for fetching it. If it cannot be fetched, the cache will download the document as soon as the computer is reconnected, or ignore the request.

When A arrives at C's laboratory, he reconnects his computer to the fixed network. At this moment, his cache performs a number of actions: find a local cache for improving its performance; retrieve the missing documents; control which documents are ejected from the cache if storage space is needed for the new documents; get updates of the working group's shared documents; and propagate the updates on the documents that it owns and which have been modified during the disconnection.

During his meeting with C, A is connected to the network. He can browse the Web, while staying in "disconnected mode". When storage space is needed, he is being asked about the documents to eject from the cache. A can then choose to eject only the documents he will not need during his journey back to his own laboratory.

As A is in a foreign country, he does not know well the topology of his host network. He may have difficulty choosing the best location for the mirrored documents that he requests. Given some information about this topology, his cache is able to automatically select the best replica. So A's cache requests C's cache for this kind of "local" information.

After his meeting, A goes back to his laboratory. When he reconnects to his home network and retrieves his usual environment, he switches his cache to "connected mode". This operation frees him from the intrusive questions asked by his cache about documents to eject.

This simple scenario illustrates the following of Oléron's features:

- An Oléron cache is both a Web proxy cache and a home site for editable documents. It can handle consistency management of these documents.
- Oléron is able to deal with disconnections: it can prepare for them by prefetching the needed documents; it can run while disconnected by having special handling of cache misses; it can manage reconnection by discovering local resources and running cache reconciliations.
- Given information about the topology of the network, Oléron is able to automatically redirect requests for mirrored documents to the best replica.

3 Possible solutions

The above objectives can be partially attained by adapting simple caching techniques.

In the World-Wide Web, caching documents greatly improves the average access time, and decreases server and network load as well [5]. Caching is also beneficial with mobile computers by making documents available during disconnected operation. This approach was first used in the Coda file system [10].

In order to solve issues of replication, replacement, consistency and administration, caches can cooperate to maintain mutual consistency between copies [4, 5, 6, 12]. Cooperation can also improve performance when client caches retrieve documents from one another, instead of always contacting the server in the case of a cache miss.

We can use cooperation as an enhancement to traditional Web caches. Cooperative caches are able to pool their resources by exchanging document contents and meta-information. Such cooperation is expected to increase performance, fault tolerance, continuity of service, and to balance the load. In particular, it can help to improve caching by working around scarce storage space and lack of prefetching by other users.

Another point in favor of cooperation is that classical caches have better performance when shared by a large number of users and when they have access to a large amount of storage space. This is generally not realistic on mobile computers, although caches have proven to be useful for disconnected Web browsing [3, 9, 15]. Cooperation is then a good way to improve caches on mobile computers.

Our approach is to develop a general cache, able to deal with different kinds of uses, and to adapt to its current environment. Integrated in the caches will be libraries of policies to be triggered, depending on the execution mode, for cache replacement, consistency management, filtering of the cacheable documents, prefetching, collaboration between caches, and communication. Mechanisms for locating resources such as caches and documents will be general, and used throughout the whole system.

4 Oléron internals

This section suggests a possible design toward the objectives described above, in the area of cache control and resource access management.

4.1 Filtering

Filtering policies aim at choosing *which* documents to cache, and the type of caching (e.g. on disk or in memory). To make the selection, the filter can consider the type of the document, its size, source, etc.

Some documents should not be cached. For example, CGI (Common Gateway Interface) scripts dynamically create documents. These are expected to change at each call, therefore it is inefficient to cache them. Documents located on a local server are also often considered not useful to cache (on a fixed computer), because the client-server network cost is of the same order as the client-cache cost. On a mobile computer, the policy to not store multimedia documents saves storage space.

The filtering can be done automatically for most purposes. In some cases, such as mobile computers, it will preferably be done by user choice at load time.

4.2 Prefetching

Prefetching policies aim at anticipating users' requests [13]. Automatic prefetching could be based on references that exist between documents, or on access histories provided by other caches. Users can explicitly prefetch some documents (or document hierarchies) that they plan to access later [1].

The choice of policies will likely be based on the type of machine and the kind of work it needs to perform. On a mobile computer, prefetching is necessary to ensure continuous access to documents. On a fixed machine, prefetching could be an optimization.

4.3 Cooperation

A cooperation policy enables caches to benefit from one another. This can be achieved by exchanging information between caches, such as document contents, lists, dates or access histories.

For example the ICP (Inter Cache Protocol) [5] and the Berkeley cache protocol [12] rely on synchronous inter-cache messages at the time of a miss. This minimizes storage usage, but suffers from high latency and high network "burstiness". Moreover, these policies broadcast caches misses to a whole cache group, causing redundant computations and high load. In contrast, the Relais protocol [4, 6] asynchronously replicates the table of cache contents. A local lookup at miss time allows redirecting the miss to a single neighbor with a high hit probability. This minimizes latency and load, but necessitates some storage for each cache to store others' directories.

The choice of policy will depend on the cost of available resources. For example, a mobile computer uses the null cooperation policy while disconnected, and reverts to some standard cooperation policy when connected. We intend to mainly use the Relais protocol.

4.4 Consistency management

A consistency management policy is responsible for maintaining invariants between copies of a document, for example, that all copies be identical. This can be performed by exchanging information between copies or by exploiting document meta-information. The most popular consistency policies are Time-To-Live, client polling and invalidation [8].

The choice of policy is based on the characteristics of the document itself, such as type, liability to change, numbers of replicas, or contents, and whether the machine is mobile or fixed. For example, if a document has few copies then it will be easy to keep track of them, and to use an invalidation or update protocol. On the contrary, the consistency of a widely replicated document will likely be achieved with Time-To-Live protocols.

4.5 Cache replacement

A replacement policy is responsible for choosing documents, considered the least valuable, to remove when cache space is needed. Value functions such as LRU (Least Recently Used), LFU (Least Frequently Used), and document size, are widely used. On a mobile computer, we can also imagine removing documents from the cache only after confirmation or a hint provided by the user using upcalls and pop-up windows.

If the replacement criterion is maximizing hit rate, we can use a size-based policy, which preferably removes the biggest documents, favoring a large number of small documents [18]. However, this favors caching of small useless documents such as icons. Some suggest favoring hits on documents whose load time would be high, by using such algorithms as LRU or LFU. For a mobile computer, one could suggest an algorithm driven by user hints, in order to remove only unimportant documents.

4.6 Resource access

The general location and access mechanism will provide a way to manage resource use. Using this mechanism and cooperation meta-information, the caches will be able to retrieve the nearest copies of some documents. Furthermore, a cache on a mobile host will be able to locate one of its peer on the fixed network with which cooperate.

We intend to use a forwarding-pointer like mechanism, such as the one described in [2, 17], extended with a flexible binding to access the nearest available documents, caches or mirrors [11, 16].

5 Conclusion

Oléron is a system designed for supporting large-scale information sharing. It targets such diverse environments as fixed, disconnected and mobile computing.

The context in which we choose to illustrate our concepts is the World-Wide Web. However, we think that the approach of Oléron should be general enough to be applied to other kinds of environments or documents (e.g. collaborative editing within small groups of users).

To achieve this goal, we designed Oléron as a set of cooperating caches able to exchange information about the documents. This collaboration helps to keep the caches up-to-date, but is also expected to reduce average access time, to balance load among servers, and to improve continuity of service.

Caches, allowing local storage of documents, have already proved their usefulness on mobile computers. They enable the user to access his documents even while disconnected or weakly connected. Nevertheless, caches have better performance when shared by a large number of users and when they have access to a large amount of storage space. Since these assumptions are not realistic on mobile computers, collaboration between caches is a good way to achieve this goal.

To ensure that Oléron caches can dynamically adapt their behavior, we provide replaceable algorithms for needed functionalities; for instance, for filtering of documents, prefetching, cache consistency and replacement, cooperation policies as well as various communication schemes. The current choice of policy is triggered by the kind of machine currently in use, the environment, the user's wishes, etc.

All these mechanisms have a common objective: cache control. In a mobile environment, this is a very important point. The user gives hints to the cache for dealing with misses, replacements or prefetching, and should be able to choose the execution mode of his cache (mobile, normal, etc.).

This article presented the very beginning of the Oléron project and some of its goals. The long-term objective is to provide read-write document editing within groups of users.

References

[1] Baggio, A. Disconnectable web proxy. Internal Coda group report, Carnegie Mellon University, Sept. 1996.

[2] Baggio, A., and Piumarta, I. Mobile host tracking and resource discovery. In *Proc. 7th ACM SIGOPS European Workshop* (Connemara (Ireland), sept 1996).

[3] Baquero, C., Fonte, V., Moura, F., and Oliveira, R. Mobi*Scape*: WWW browsing under disconnected and semi-connected operation. In *Proceedings of the Portuguese WWW National Conference* (Braga, Portugal, July 1995). http://www.di.uminho.pt/~cbm/ps/MobiScape.ps.

[4] Berenguier, E. Cache réparti pour le World-Wide Web. Master's thesis, École polytechnique, 1995.

[5] Chankhunthod, A., Danzig, P. B., Neerdaelsand, C., Schwartz, M. F., and Worrell, K. J. A Hierarchical Internet Object Cache. Tech. Rep. 95-611, Computer Science Department, University of Southern California, Mar. 1995. ftp://ftp.cs.colorado.edu/pub/cs/techreports/schwartz/HarvestCache.ps.Z.

[6] Dugelay, S. Caches coopératifs pour le World-Wide Web. Master's thesis, École Nationale Supérieure des Télécommunications, Sept. 1996.

[7] Forman, G., and Zahorjan, J. The challenges of mobile computing. *IEEE Computer* (Apr. 1994), 39–47. ftp://ftp.cs.washington.edu/tr/1993/11/UW-CSE-93-11-03.PS.Z.

[8] Gwertzman, J., and Seltzer, M. World-Wide Web cache consistency. In *Proceedings of the 1996 Usenix Technical Conference* (Jan. 1996). http://www.eecs.harvard.edu/~vino/web/usenix.196/caching.ps.

[9] Kaashoek, M. F., Pinckney, T., and Tauber, J. A. Dynamic documents: Mobile wireless access to the WWW. In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications* (Santa Cruz, CA., Dec. 1994). http://snapple.cs.washington.edu:600/mobile/mcsa94.html.

[10] Kilster, J. J., and Satyanarayanan, M. Disconnected operations in the Coda file system. *ACM Transactions on Computer Systems 10*, 1 (Feb. 1992), 3–25.

[11] Maisonneuve, J. *Hobbes : un modèle de liaison de références réparties.* PhD thesis, Université Pierre et Marie Curie, oct 1996. http://www-sor.inria.fr/SOR/docs/maisonneuve_thesis96.html.

[12] Malpani, R., Lorch, J., and Berger, D. Making World Wide Web caching servers cooperate. In *Proceedings of the 4th international WWW conference* (Boston, MA, Dec. 1995). http://www.w3.org/pub/Conferences/WWW4/Papers/59/.

[13] Padmanabhan, V. N., and Mogul, J. C. Using predictive prefetching to improve World-Wide Web latency. In *Proceedings od the SIGCOMM '96 conference* (July 1996). http://www.cs.berkeley.edu/~padmanab/papers/ccr-july96.ps.gz.

[14] Pitoura, E., and Bhargava, B. Dealing with mobility: Issues and research challenges. Tech. Rep. CSD-93-070, Purdue University, Nov. 1993. http://www.cs.purdue.edu/homes/pitoura/distribution/Mobile/review.ps.

[15] Schilit, B. N., Douglis, F., Kristol, D. M., Krzyzanowski, P., Sienicki, J., and Trotter, J. A. TeleWeb: Loosely connected access to the World-Wide Web. *Computer Networks and ISDN Systems 28*, 7–11 (May 1996), 1431. http://garuda.imag.fr/www5cd/www172/overview.htm.

[16] Shapiro, M. A binding protocol for distributed shared objects. In *Proc. Int. Conf. on Distributed Computing Systems* (Poznan (Poland), June 1994).

[17] Shapiro, M., Dickman, P., and Plainfossé, D. SSP chains: Robust, distributed references supporting acyclic garbage collection. Tech. Rep. 1799, INRIA, 1992. http://www-sor.inria.fr/SOR/docs/SSPC_rr1799.html.

[18] Williams, S., Abrams, M., Standridge, C. R., Abdulla, G., and Fox, E. A. Removal policies in network caches for World-Wide Web documents. In *Procedings of the ACM SIGCOMM'96 conference* (Aug. 1996), Stanford University, CA. http://ei.cs.vt.edu/~succeed/96sigcomm/.