System Support for Transparency and Network-aware Adaptation in Mobile Environments

Aline Baggio

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

System Support for Transparency and Network-aware Adaptation in Mobile Environments

Aline Baggio

No 3408

Avril 1998

THÈME 1

Rapport de recherche

ISSN 0249-6399

System Support for Transparency and Network-aware Adaptation in Mobile Environments*

Aline Baggio†

Thème 1 — Réseaux et systèmes
Projet SOR — http://www-sor.inria.fr/

Rapport de recherche n°3408 — Avril 1998 — 8 pages

Abstract: Today's applications for mobile computers are inadapted: mobile users must foresee disconnections and plan manually what they will need during these disconnections. To avoid this, our project aims at reconciling transparency with adaptability and flexibility for these users. It is based upon mechanisms for referencing remote objects, and for flexible binding, allowing on-the-fly replication and policy changes. This system-level support hides the disadvantages of disconnected operation, moves, unavailable data or servers, etc. Our project, in its early development phase, already provides support for disconnection and reconnection of references to arbitrary objects.

Key-words: Mobile computing, network-awareness, adaptability, Cadmium project

(Résumé : tsvp)

* This paper appeared in the proceedings of ACM Symposium on Applied Computing, Special Track on Mobile Computing Systems and Applications, February 27th - March 1st, 1998 (Atlanta, Georgia, USA).
† E-mail: Aline.Baggio@inria.fr

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
Téléphone : 01 39 63 55 11 - International : +33 1 39 63 55 11
Télécopie : (33) 01 39 63 53 30 - International : +33 1 39 63 53 30

Support système transparent et flexible pour les environnements mobiles

Résumé : Les applications actuelles sont relativement mal adaptées à une utilisation en environnement mobile : il est encore souvent nécessaire de prévoir les déconnexions, charger manuellement les données indispensables, etc.. Pour pallier à ce problème, notre projet vise à combiner la transparence à la mobilité et l'adaptation à l'état courant de l'environnement. Nous nous basons sur des mécanismes de référençage d'objets et de liaison flexible, qui autorisent la réplication au vol des objets ainsi que la ré-évaluation des stratégies (réplication, cohérence, etc.). Ce support système permet de masquer les désavantages des opérations déconnectées, des déplacements, les données ou serveurs inaccessibles. Notre projet fournit actuellement un support pour la déconnexion et la reconnexion de références vers des objets arbitraires.

Mots-clé : Informatique mobile, adaptabilité, projet Cadmium

1 Today's mobility trends

Most problems encountered in mobile computing are due to mobility itself, the wireless communication medium, or the laptop's scarce resources. Others are induced by the mismatch between current software and mobile user needs.

Today's mobile users generally utilize applications mostly targeted for fixed environments. These applications are not well adapted to mobile environments. They lead to use laptops as relocatable (wired) desktops rather than making them really mobile and independent of the rest of the fixed network.

Some applications have started to take in account mobile users' needs, for instance by integrating prefetching functionalities in browsers, or consistency and conflict resolution tools in e-mail or calendar applications.

In our opinion, more generic mobility facilities should be supplied at the operating system level so that *any* application can take advantage of it. Such support should provide availability of data, performance, and automatic prefetch or "push".

2 Transparency vs. flexibility

Our goal is to provide mobile users with their usual environment everywhere. To do so, two conflicting approaches are generally encountered: transparency and adaptability. With transparency, applications need not to be aware of mobility because the system completely hides disconnections. Unfortunately, this means that an application will block if some expected resource is unreachable, and leads to counter-productive tradeoffs for the applications [10]. Conversely, in the adaptable approach, applications are mobile-aware. The system does nothing to support mobility. It lets the applications manage it if they need to, and adapt for example by switching to an alternate for some missing resource. The drawback of this approach is that the system has no way to control resource allocation neither limit their use [10].

Both approaches are feasible and work, as demonstrated in systems like Coda [7] or specific applications such as some calendars or browsers. However, it appears that neither of these two approaches, used alone, is the best way to handle mobility. Both system and application control are needed and therefore both transparency and adaptability, as demonstrated by Odyssey [16, 10, 11], Katz's approach [6] or Rover [5].

3 Approach

Our approach is to combine adaptation and transparency in order to provide users with a convenient mobile environment.

The goal is to minimize users annoyance when disconnected. They should have available their files and data, as well as other sources of information such as the World-Wide Web. This is what we call *transparency to mobility.* Transparency is seen from the user level and must be provided at the system level so that any application on top of it can use the mobile support with minimal changes.

Nevertheless, since complete transparency is not possible nor even desirable [16, 10], mobile users or applications should be able to switch to the locally available resources, including servers, documents, object replicas, etc. Our support is *network-aware*: it can identify and use locally-available resources, adapt to changes and provide an appearance of continuity of service, managing tradeoffs between transparency, flexibility and efficiency of use.

This approach reconcile transparency with network-aware flexibility by distributing the responsibilities between the system, the applications and the users. Basic policies for replication, conflict detection, etc. are provided by the system, but applications can plug in better ones, adapted to their current needs.

To be completely general, our system supports arbitrary objects designated and invoked via references. The system is transparent because one reference remains available across disconnections and moves. To ensure the latter, we provide a flexible *network-aware binding protocol*. When an object or a computer moves, the corresponding reference bindings are broken and selectively *re-bound* to "better" objects. This provides adaptations to the local environment.

4 Mechanisms

Our approach uses four mechanisms: object references, *funnels*, binding to objects, and dynamic replication.

4.1 References

As a mechanism for referencing and invoking objects, we use Stub-Scion Pair Chains (SSP Chains) [18, 15], which track object movement by extending and short-cutting. SSP Chains provide transparent referencing of remote objects in a distributed system. They garbage collect unused objects, they are cheap and fault tolerant. We also assume an underlying mobile network layer, such as Mobile IP [13] or IPv6 [4], so that SSP Chains do not explicitly reference objects using host addresses.

4.2 Funnel model

To allow SSP Chains to deal with host mobility and disconnections [1], we use distinguished nodes called *funnels*. A funnel is located on the fixed network. A mobile host communicates with the rest of the network via some funnel. So the funnel multiplexes SSP Chains to the mobile host and transparently forwards messages when the mobile moves. The funnel is also in charge of searching and reconnecting disconnected stubs and scions after a mobile host move.

A "port-of-call funnel" is located close to the current location of the mobile host and funnels communications between the fixed network and the mobile hosts. A "home funnel" stands for the mobile host at its home location, keeping track of the current mobile host location, storing files, etc. Communication between a fixed host and the mobile host proceeds initially to the home funnel which forwards it to the port-of-call funnel, which in turn forwards to the mobile host. After short-cutting, the fixed hosts communicates directly via the port-of-call.

A funnel is a convenient place where to store information and replicate or cache objects. A funnel thus serves as a resource informant (being a "well-known" host of the foreign network) [1]. This can be done by using a name server or an object broker approach. It provides information such as local network resources or services, local time, local weather or road traffic conditions, etc.

4.3 Binding

Disconnectable SSP Chains support a flexible binding mechanism [17, 9, 1] that allow a reference to a replicated object to dynamically bind to the "best" replica. The notion of "best" is subject to change, especially in a mobile environment. It can be measured considering availability, load, network distance, network speed, fees, locality, user preferences, etc. These measures control the breaking and rebinding of references.

Finding equivalent object servers relies on the resource discovery scheme offered by funnels [1]. Discovery of equivalent replicas of objects other than servers implies new SSP Chain protocols in order to allow access and management of groups of objects (i.e. replicas). This part is under study.

Bindings are broken according to user or application needs criteria. For example, breaks can arise automatically on a time-out, or after a move, to rebind on a local replica; semi-automatically using user profiles that list preferred objects or servers; or manually by users request. Any of these dynamic reconfigurations should be preceded by checkings to verify if the target object is reachable, if it is available or overloaded, or if the user is allowed to access and use it, etc.

4.4 Replication

The replication mechanisms provide support for downloading data on both fixed and mobile parts of the network before disconnection, or for decreasing latency in the case of low-bandwidth connections. They can also be used to transparently create "mirror" objects (similar to World-Wide Web "mirror servers").

Replication of objects or group of objects [8] can be on-demand, semi-automatic or on-the-fly.

On-demand replication involves that the user or the application specify their needs in term of objects or servers. Such a replication is used in Coda [7] and Bayou [19, 14]. It is generally done just before a disconnection in order to preserve availability of data.

Semi-automatic replication assumes that the user defines a profile specifying what to replicate [7] as well as time hints about when to replicate. For example, users who leave their office at 20:00 can request a prefetch at 19:30 each day.

Finally, on-the-fly replication occurs transparently. It can be achieved by tracking in and out coming references on a mobile host. Tracking itself is ensured by a funnel-like object. By instance, on-the-fly replication of objects can be initiated according to the available bandwidth.

Replicating objects also implies managing consistency between replicas. A very basic conflict detection and resolution tool can by provided by the system. For example, it can search conflicts by comparing replica of directories, then try to resolve file by file if a conflict were found, then line by line. The users can be notified that a conflict were found and successfully solved or they can be asked to solve the conflict manually. The Coda [7] and Bayou [19, 14] projects have investigated such reconciliation for mobile environments.

4.5 Additional mechanisms

Thanks to the flexible binding, objects can select policies from a library of prefetching, mirroring and caching. Application can plug in replication, conflict detection and resolution, filtering policies, to replace the ones predefined by the system.

Filters can be dynamically bound into a communication channel to modify message contents, e.g. using compression and data filtering and removal.

Finally, upcall registration [10] allows applications to track changes of resources such as bandwidth, CPU, or memory.

5 Use and applications

To illustrate our mechanisms, let us take the example of one of our targeted applications: a distributed calendar. The calendar is shared among several mobile users. Consider a shared meeting item. For the connected users, the meeting item is a transparent link to the original meeting item (located at some owner site).

When a mobile hosts disconnects, the item is replicated on-the-fly. Note that the disconnected host might be the one retaining the original object; then the system must replicate the meeting item on a fixed host, for example the owner's home funnel station for other users to see during the owner's disconnection. The binding changes to point towards the right item.

If some changes occur during the disconnection, they can be fetched as soon as a network is available. Calendar items may have properties such as "fetch this item as soon as bandwidth exceeds 10 kb/s". A conflict resolution policy must manage conflicting changes to the calendar such as overlapping items, concurrent updates of the same item, etc.

Our other target applications are a replicated mail-box and a cooperative World-Wide Web cache. Both will benefit from the on-the-fly replication, and dynamic binding to locally available objects.

6 Related work

Like Bayou [19, 14], one of our targets is to address large scale, availability, adaptability problems; but we do not target database replication. We rather choose a finer grain scheme based on object replication similar to Stub-Scion Pair Chains [18, 15], Fragmented Objects [2, 3] and Globe [20]. This scheme allow us to deal with any kind of objects. Our approach is similar to Rover [5], but we do not intend to base our mechanisms exclusively on World-Wide Web objects. Of the Coda [7] approach, we can take advantage of the prefetching and caching functionalities and augment them to provide a more transparent and automatic scheme. Odyssey [16, 10] provides adaptability and flexibility but does not handle distribution in the sense of access and replication of objects. In addition to management of network or host resources, we would like to deal also with servers, objects replicas, etc., that are available episodically. Our mechanisms are layered on top of existing operating systems and communication protocols.

7 Conclusion

Our approach to mobility aims at providing adaptability, availability and transparency to users. To reach our goals we use a remote reference mechanism along with dynamic replication and binding schemes. We bind to available resources as the need arises. Distributed applications that share data, such as calendars, distributed mailboxes, or cooperative World-Wide Web caches are good candidates for using our mechanisms. Our project is in an early development phase. We are currently re-developing the SSP Chain extensions for supporting flexibly disconnected operation. Replication and management of group of objects are under design. Some complementary works are under way, such as [12], implementation of SSP Chain in various languages or the development of a distributed mail box application.

References

[1] Baggio, A., and Piumarta, I. Mobile host tracking and resource discovery. In *7th ACM SIGOPS European Workshop* (Connemara, Ireland, Sept. 1996). http://www-sor.inria.fr/publi/MHTRD_sigops96.html.

[2] Brun-Cottan, G., and Makpangou, M. Adaptable replicated objects in distributed environments. Tech. Rep. RR-2593, INRIA, May 1995. http://www-sor.inria.fr/publi/RCMFGCCOSDS_rr2593.html.

[3] Dickman, P., and Makpangou, M. A refinement of the fragmented object model. In *Third International Workshop non Object-Orientation in Operating Systems* (Dourdan, France, Sept. 1992). http://www-sor.inria.fr/publi/RFOM_iwooos92.html.

[4] Hinden, R. M. IP Next Generation overview. Tech. Rep. Internet Draft, IETF Mobile IP Group, May 1995. http://www.ietf.org/.

[5] Joseph, A. D., deLespinasse, A. F., Tauber, J. A., Gifford, D. K., and Kaashoek, M. F. Rover: A toolkit for mobile information access. In *Fifteenth Symposium on Operating Systems Principles* (Dec. 1995). http://www.pdos.lcs.mit.edu/rover/.

[6] Katz, R. H. Adaptation and mobility in wireless information systems. *Personal Communications 1*, 1 (1994), 6–17. http://daedalus.cs.berkeley.edu/publications/Personal.ps.gz.

[7] Kistler, J. J., and Satyanarayanan, M. Disconnected operation in the Coda file system. In *Thirteenth ACM Symposium on Operating Systems Principles* (Asilomar Conference Center, Pacific Grove, US, 1991), vol. 25, pp. 213–225. http://www.cs.cmu.edu/afs/cs/project/coda/Web/docs-coda.html.

[8] Kuenning, G., and Popek, G. J. Automated hoarding for mobile computers. In *Sixteen ACM Symposium on Operating Systems Principles* (Saint Malo, France, Oct. 1997), pp. 264–275. http://fmg-www.cs.ucla.edu/geoff/sosp97.html.

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

[10] Noble, B. D., Price, M., and Satyanarayanan, M. A programming interface for application-aware adaptation in mobile computing. Tech. Rep. CMU-CS-95-119, Carnegie Mellon University, Feb. 1995. http://www.cs.cmu.edu/Web/Reports/1995.html.

[11] Noble, B. D., Satyanarayanan, M., Narayanan, D., Tilton, J. E., Flinn, J., and Walker, K. R. Agile application-aware adaptation for mobility. In *Sixteen ACM Symposium on Operating Systems Principles* (Saint Malo, France, Oct. 1997), pp. 276–287. http://www.cs.cmu.edu/afs/cs/project/coda/Web/docdir/s16-reprint.ps.Z.

[12] Nutte, O. Object referencing and garbage collection with SSP Chains in CORBA (in french). Master's thesis, Sept. 1997. http://www-sor.inria.fr/~nutte/PS/MEMiie.DEA-O.Nutte.ps.gz.

[13] Perkins, C. Ip mobility support. Tech. Rep. Internet Draft, IETF Mobile IP Group, Jan. 1995. ftp://software.watson.ibm.com/pub/mobile-ip/draft-ietf-mobileip-protocol-08.txt.

[14] Petersen, K., Spreitzer, M., Terry, D., and Theimer, M. Bayou: Replicated database services for world-wide applications. In *7th ACM SIGOPS European Workshop* (Connemara, Ireland, Sept. 1996). http://www.parc.xerox.com/csl/projects/bayou/.

[15] Piumarta, I. SSP Chains - from mobile objects to mobile computing. In *ECOOP'95 Workshop on Mobility and Replication* (Aarhus, Denmark, Aug. 1995). http://www-sor.inria.fr/publi/SSPMobPP_ecoop95-mobility-pp.html.

[16] Satyanarayanan, M., Noble, B., Kumar, P., and Price, M. Application-aware adaptation for mobile computing. In *Sixth ACM SIGOPS European Workshop* (Dagstuhl, Germany, Sept. 1994). http://www.cs.cmu.edu/afs/cs/project/coda/Web/docs-ody.html.

[17] Shapiro, M. A binding protocol for distributed shared objects. In *International Conference on Distributed Computing Systems* (Poznan (Poland), June 1994). http://www-sor.inria.fr/publi/extended-ICDCS.300dpi.html.

[18] 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/publi/SSPC_rr1799.html.

[19] Terry, D., Theimer, M., Petersen, K., Demers, A., Spreitzer, M., and Hauser, C. H. Managing update conflict in bayou, a weakly connected replicated storage system. In *15th ACM Symposium on Operating Systems Principles* (Copper Mountain Resort, Colorado, US, Dec. 1995). http://www.parc.xerox.com/csl/projects/bayou/.

[20] van Steen, M., Homburg, P., and Tanenbaum, A. S. The architectural design of Globe: A wide-area distributed system. Tech. Rep. IR-422, Vrije Universiteit, Netherlands, Mar. 1997. http://www.cs.vu.nl/~steen/globe/publications.html.

Unit´e de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unit´e de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unit´e de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unit´e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unit´e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399