3

The impact of mobility on distributed systems platforms

N. Davies

Distributed Multimedia Research Group,
Department of Computing,
Lancaster University,
Lancaster,
U.K.
Fax: +44 (0)1524 593608
Telephone: +44 (0)1524 65201
E-mail: nigel@comp.lancs.ac.uk

Abstract

Mobile computing environments are characterised by change: heterogeneous end-systems have to operate over underlying communications whose quality of service (QoS) and associated cost may be subject to rapid and massive fluctuations. Such changes present new challenges for all distributed system services and in particular for distributed systems platforms. This paper explores the impact of mobility on distributed systems and distributed systems platforms highlighting shortcomings in both the models and implementations of current platforms. In particular, the lack of platform support for adaptive applications and services is explored and recommendations are made for future distributed systems.

Keywords

Mobile computing, distributed systems platforms, QoS, adaptive applications.

1 INTRODUCTION

The starting point for this paper is the belief that knowledge is power. More specifically, powerful applications require knowledge of their underlying communications and processor infrastructure to operate effectively. Such a view is in conflict with the approach adopted by current distributed systems platforms (RM-ODP [ISO95], DCE [OSF91] and CORBA [OMG91]) which attempt to use transparencies to hide details of the underlying distributed

A. Schill et al. (eds.), *Distributed Platforms*

system from their client applications. This approach works adequately when the characteristics of the underlying system are relatively static and applications and end-users can make assumptions about the levels of service they are likely to experience. However, when the levels of service which can be provided are subject to rapid and significant fluctuations, as is the case in a mobile environment, the approach starts to break down. This paper focuses on the role of distributed systems platforms in mobile environments and examines the new features they must posses if they are to make the transition from information hiders to information providers.

It should be stressed at this point that there are three distinct types of application which users might run on their mobile end-systems; stand-alone applications, existing distributed applications and advanced mobile applications, and that the distributed systems platforms of concern to us in this paper have a role to play in supporting only the latter of these (i.e. advanced mobile applications). Stand-alone applications require relatively little support in order to operate on a mobile end-system. The main requirement is that they have access to their usual file store and this can be provided using a mobile file system such as CODA [Satyanarayanan90] or an extended version of AFS as proposed by Honeyman [Honeyman92]. These file systems generally work by ensuring mobile clients cache copies of files while working on fixed networks and then, when mobile, operate on these cached copies. At re-connection time the cache is re-integrated with the files held on the file-server and any conflicts are marked and must be resolved by the user.

Existing distributed applications commonly run on mobile end-systems also tend to make relatively modest demands on their underlying support system. Examples of these applications include email, job dispatch systems and, more recently, web-browsers [Bartlett94]. Such applications tend to send relatively small amounts of non-time-critical information (with perhaps the exception of web-browsers). They also tend to have relatively simple patterns of interaction, i.e. they are, without exception, simple client-server distributed applications. They do not, in general, require sophisticated distributed systems platforms to support their operation.

In contrast, the third category of application, advanced mobile applications, are characterised by peer-to-peer and group interaction, transmission of safety and time critical information, use of multimedia data and support for collaborating users. Examples of such applications include multimedia mobile conferencing and collaborative applications to support the emergency services and these clearly represent a major advance in the state-of-the-art in mobile applications. In addition, such applications require extensive distributed systems support and it is on the nature of this support that the remainder of the paper focuses.

The paper is structured as follows. Section 2 discusses the main characteristics of mobile environments and highlights the importance of change. Section 3 then considers how information regarding these changes can be exploited and surveys some recent work on change-based systems. Based on this survey section 4 outlines the role of future distributed systems platforms as providers of information on change and section 5 contains some concluding remarks.

2 CHARACTERISTICS OF MOBILE ENVIRONMENTS

When an end-system is mobile its environment is subject to change. In particular, the level of service it experiences from the network may change, the cost it pays for this service may change, its physical location may change and the capabilities of its supporting hardware may change.

2.1 Network quality-of-service

The quality-of-service (QoS) that the underlying communications system can provide is related to the freedom of movement required by the end-user (see figure 1).

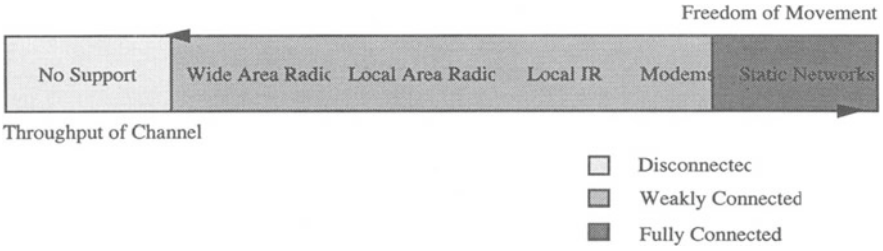

Figure 1 The relationship between movement and network QoS.

In more detail, in a fixed network environment users can expect relatively reliable communications with a bandwidth of between 1 and 100 Mbps. Such characteristics make it possible to design applications and operating systems services with little regard for optimising network traffic. Indeed, in wide-area fixed networks latency is fast becoming the overriding factor in determining the performance of distributed systems while bandwidth is seen as a plentiful resource. In the local area these network characteristics make it possible to rely on centralised services with little need for replication other than to provide fault tolerance.

If an end-system moves from a fixed network connection to a local-area wireless connection then the QoS provided by the supporting network is likely to reduce substantially. Depending on the technology used (directed or diffuse infra-red or radio) the bandwidth available will fall to between 10 Kbps and 2 Mbps. In addition, the characteristics of the channel will also change. For example, the number of bit-errors will increase leading to a significant number of packets being lost. Furthermore, in cellular systems, cell hand-offs and coverage blackspots will also increase the number of packets lost. While it might naively be assumed that these packet losses can simply be treated as a reduction in the overall bandwidth available Cáceres demonstrated in [Cáceres94] that it is important that network protocols are tuned for networks with these packet-loss characteristics. In more detail, he showed that the performance of TCP was significantly reduced when it was operated over a wireless network with packet-losses due to bit-errors and cell-handoffs. Specifically, the loss of packets causes a marked degradation in TCP's performance due to its exponential back-off strategy originally developed to avoid network congestion. The solution adopted by Cáceres uses information from cell managers to inform the end-points of TCP connections when a cell hand-off is taking place in order that they can adopt an alternative back-off strategy.

Finally, when an end-system requires wide-area wireless connectivity the available bandwidth will drop to between 0 and 9.6 Kbps (analogue cellular systems can usually support 1200 bps, both the GSM digital data service and CDPD support 9.6 Kbps). In all cases the latency of establishing a connection also increases such that it can be measured in seconds rather than milliseconds. The fact that the bandwidth is stated as potentially dropping to 0 Kbps reflects the fact that in a wide-area complete disconnection from the network can be a frequent occurrence.

2.2 Cost

The issue of cost reduction has, to date, been largely ignored by the designers of distributed systems. This is because network services have been effectively free to most users and schemes for charging for distributed systems services have been hard to deploy. The exception is the telecommunications companies who in initiatives such as TINA have attempted to maintain a clear distinction between those services which notionally reside 'inside' the network and thus can be charged for and those which lie outside. The challenges of devising and implementing fair, efficient and secure schemes for charging for the use of distributed system services are

common to all distributed systems. However, dealing with the cost of communications is particularly acute in mobile environments.

If we consider the networks described in section 2.1 there are clear cost implications associated with each of the different network types. In the case of local-area and fixed networks the costs are, in the case of most institutions, covered centrally and users are not charged according to usage (though there may be an internal charging strategy based on bytes sent to spread the cost more evenly between groups within the institution). In the case of wide-area wireless communications or when mobile users must rely on dial-up lines the situation is very different. Consider the case of a user whose underlying communications is being provided by a public cellular telephone provider. The actual cost of transmitting information will depend on a vast array of factors including the tariff the user initially signed up for, the time of day, the users physical location, whether or not the user is having to exploit a roaming agreement to obtain coverage and whether the data is being sent via an explicit connection or via a short message or datagram service. Many of these factors will change dynamically and can make a substantial difference to the cost of connectivity. Moreover, while fixed computer network usage is typically based on the amount of data sent or a fixed rate for line rental, mobile users are often burdened with tariffs designed for telephony and hence are charged for connection time regardless of the amount of data actually transmitted or received.

2.3 Location

By definition mobile users' physical location changes over time. These changes can be detected in a number of ways: in the case of cellular systems it is possible to query the system to determine the position of any given user. Technologies such as the active-badge system designed at Olivetti Research Labs [Want92] can also be used to determine the physical location of mobile users. The implication of this is that any point in time it is possible to accurately pinpoint a users physical location and, perhaps more importantly, the location of other services and users with which they may wish to interact. Section 3.3 discusses the relevance of this information and how changes to a users' physical location can be exploited by distributed systems components to offer a better level of service.

2.4 End-systems

Mobile computing devices typically have different characteristics to their desk-top counterparts. Indeed, while significant advances have been made in the state-of-the-art in portable computing there is always a fundamental conflict between portability and performance. In addition, there are two other factors which solely affect mobile computers. Firstly, mobile computers are always subject to the limitations imposed by current battery technology. To address this problem they use a variety of mechanisms for reducing their overall power consumption including low-power processors and i/o devices, the use of doze-modes and hardware suspend-resume capabilities and automatically reducing the processor's clock-speed during periods of relative inactivity. Despite the use of these techniques it is an inescapable fact that most portable computers spend substantial periods of time switched off to conserve power. This is in contrast to most desk-top machines which are often left on continuously.

The second limitation of portable computers is that in order to be portable their size must be limited. In particular, the size of their screens must be reduced to no more than A4 size and, in the case of PDA type machines, substantially smaller. This means that existing user interfaces often don't work and custom interfaces much be developed in order to address this issue.

3 EXPLOITING CHANGE INFORMATION

Given that changes such as those described in section 2 occur in mobile computing systems it is important to determine what, if anything, can be achieved by being aware of these various changes. The following sections contain examples of how change information can be used to improve the overall performance of distributed systems. A number of these examples are based on the results of early research in the areas of adaptive services [Davies94], [Katz94] and context-aware applications [Schilit94] which are encouraging and suggest that substantial improvements in service levels can be achieved if system components are provided with the information necessary to tailor their behaviour to the environment.

3.1 Network quality-of-service

Coping with changes in network QoS has received the most research of any of the areas discussed in section 2 and information regarding network QoS can be used by a wide range of distributed systems components. For example, within a prototype advanced mobile application developed as part of the MOST project at Lancaster [MOST92] information regarding the network QoS is used to tailor the application's behaviour. In more detail, the application supports a component which provides mobile users with access to a remote database. When a mobile user issues a query the number of fields returned for each matching record is determined dynamically based on the number of matches and the network QoS. Hence, when the user is connected by a slow-speed network only information which is likely to be of use in further restricting the search is returned, while in a high-speed network where latency is more critical than throughput, the entire found-set is returned (subject to a specified threshold). The application also gives QoS feedback to users in order to allow them to adapt their behaviour to mach the network characteristics. This stemmed from an original application requirement to provide a graphical monitor similar in appearance to the signal-strength meters commonly found on cellular communications equipment. In practice the display provides substantially more detailed feedback to users allowing them to see, for example, where bottlenecks are occurring in complex group based activities involving sophisticated interaction patterns. This enables users to adjust their patterns of work, e.g. switching between synchronous (shared white-board) and asynchronous (email) communications, to make the best use of the available network QoS.

In addition to applications, system services can also make use of network QoS information. For example, a recent version of the CODA file system discussed in section 1 includes the notion of trickle-reintegration of the log of cache updates when connected by slow-speed networks [Mummert95]. This is in contrast to the original version of CODA which assumed either full network connectivity or complete disconnection. By using trickle-reintegration CODA is able to increase the level of file consistency without exceeding the capabilities of slow-speed networks.

As a final example of how network QoS information can be used consider the case of live video transmission. Video is one of the most demanding media types since it is time-critical and requires substantial bandwidth if a reasonable frame-rate and image size is to be achieved. Techniques such as scalable video compression [Keeton93] help by reducing the overall bandwidth requirement and, more importantly, by allowing the image quality to be selectively degraded to match the capabilities of the underlying network.

3.2 Cost

Very little research has been carried out into optimising the cost of running distributed system services and applications. Despite this there is a significant amount which can be done, relatively simply, to reduce the cost if the tariff structure in use is known. For example, if a connection is charged per second of usage then it clearly makes sense to batch up messages on a give machine before sending them, particularly if the time taken to establish a connection is

significant and is also charged for. This implies that the transmission of some messages may be delayed while the system waits to see if there any subsequent messages about to arrive which can be sent at the same time. However, current communication protocols tend not provide programmer interfaces which allow applications to specify the time-constraints associated with messages and hence it is difficult for the system to arbitrarily decide to delay messages without, for example, running the risk of invoking time-outs and re-transmissions by applications. In addition, if a connection is changed by the second it makes sense to exploit any idle time, e.g. by carrying out trickle-reintegration.

If the tariff-structure subsequently changes to one in which communications are charged by the byte sent then using idle time becomes an irrelevance and efforts must be made to reduce the total amount of traffic sent over the network. Returning to our database access program example given in section 3.2, it would again make sense to send only partial records for matches, this time to reduce the overall cost of running the application rather than to increase the performance.

3.3 Location

Information regarding changes in users' location can be exploited in a number of ways. Early work by [Neuman93] on the Prospero system put forward the idea of matching client's service requests to appropriate services based on location. So, for example, asking to be supplied with a printer service and specifying a constraint such as 'location = nearest' ensures that a client is able to print their document to the printer nearest to them regardless of their physical location.

Location information is also used in the work of Schilit [Schilit94] which provides a framework for what are termed context-aware applications. An example of the types of application Schilit proposes is a memory jogger which allows users to specify that a certain message is displayed when a series of location and temporal based conditions are met, e.g. "remind me to do x when bob and jo are next in my office". Finally, changes in location information may be used to explain anomalies in network transmission as in the work of Cáceres (see section 2.1) where information from cell managers is used to inform transport protocol end-points of the likely cause of packet loss.

3.4 End-systems

There has been a substantial body of work aimed at designing user-interfaces for applications running on portable computers, e.g. [Schilit91]. Of more interest to distributed systems developers is the work of Badrinath et al. on the design of low-power distributed algorithms [Badrinath94]. This has shown that with the use of software controllable doze-modes a reduction in power usage can be made by, for example, sending large computations to remote sites and dozing until the results are ready to be collected. Clearly a number of factors such as other activities on the mobile machine and relative costs of transmission will determine the practical benefits of such an approach but the work highlights the importance of designing mobile applications which are aware of the limited power at their disposal when operating on batteries. Of course, when the portable computer is reconnected to the fixed network and a power supply the applications should be able to readjust to their new environment.

4 THE ROLE OF DISTRIBUTED SYSTEMS PLATFORMS

Given that change is a fundamental characteristic of mobile environments (section 2) and that information regarding change is of use to applications and services (section 3) then one of the key roles of future distributed systems platforms must be to provide and manage change information. As mentioned previously, this is in contrast to the transparency based approach adopted by current platforms. Hence, these platforms must be revised to include support for QoS (change) monitoring and management. In more detail, it should be possible for

applications to register an interest in changes to any of the parameters discussed in section 2, to express requirements in terms of these parameters and to obtain the values of these parameters at any time. In a prototype platform developed by the author and a number of co-workers at Lancaster this support is provided in the form of explicit bindings between client and server objects. The QoS associated with these bindings can be established at bind-time and the platform notifies interested parties if this QoS is violated (more details of this work can be found in [Davies95], [Friday96]).

The disadvantage with the approach described above is that it addresses the issue of change only in network QoS and does not, for example, provide a mechanism for applications finding out about their current location. However, the work has highlighted one important issue regarding QoS management, i.e. the difficulties associated with arbitrating between conflicting QoS requirements. For example, it might be desirable for a user to have an overall preferences file in which they can make statements such as 'always cheapest', 'don't spend more than x', 'everything can wait' and 'always fastest'. In practice however the statements are likely to be more complex then these, e.g. 'always do the cheapest unless it's X in which case do Y iff it is less than 10% more expensive and less than 10% slower than X' (this is the algorithm I use for determining my carrier when booking flights). Supporting these complex QoS statements is clearly the role of a distributed system service to which responsibility for resolution and arbitration can be delegated, particularly when multiple applications operating on the same mobile host express conflicting QoS requirements. This is an area in which substantial further research will be required once platforms are developed with the capability to provide the appropriate change information.

Finally, the services which are included with distributed systems platforms must themselves be capable of adapting to changes in their environment. Hence, for example, communications protocols must be able to operate over a wide-range of network types, naming services must scale well to environments in which they are extensively replicated but where consistency must be sacrificed for cost reduction and patterns of object interaction which prevent portable machines entering doze-mode must be prevented.

5 CONCLUDING REMARKS

This paper begun with the statement that knowledge is power. Current platforms attempt to hide information about the environment from their clients by the use of transparencies. However, it has been demonstrated that this approach is not only likely to fail, but is also counter-productive since given information about their environment, applications and system services can make better use of the available resources. Hence, knowledge can be used to create more powerful and effective applications. The responsibility for providing and managing this knowledge for advanced mobile applications will, in the author's opinion, fall to the next generation of distributed systems platforms.

ACKNOWLEDGEMENTS

The author would like to acknowledge his colleagues in the mobile computing group at Lancaster and in particular Adrian Friday and Phil Adcock for their input to this paper.

REFERENCES

[Badrinath94] Badrinath, B.R., A. Acharya, and T. Imielinski. "Structuring Distributed Algorithms for Mobile Hosts." *Proc. 14th International Conference on Distributed Computer Systems,* Poznan, Poland, June 21-24, 1994.

[Bartlett94] Bartlett, J. "W4-the Wireless World Wide Web." *Proc. Workshop on Mobile Computing Systems and Applications,* Santa Cruz, CA, U.S., December 1994.

[Cáceres94] Cáceres, R., and L. Iftode. "The Effects Of Mobility on Reliable Transport Protocols." *Proc. 14th International Conference on Distributed Computer Systems,* Poznan, Poland, 22-24 June, 1994. Pages 12-20.

[Davies94] Davies, N., S. Pink, and G.S. Blair. "Services to Support Distributed Applications in a Mobile Environment." *Proc. 1st International Workshop on Services in Distributed and Networked Environments,* Prague, Czech Republic, June 1994.

[Davies95] Davies, N., G.S. Blair, K. Cheverst, and A. Friday. "Experiences of Using RM-ODP to Build Advanced Mobile Applications." 2 No. 3, Pages 142-151.

[Friday96] Friday, A. "Extensions to ANSAware for advanced mobile applications." *Proc. International Conference on Distributed Platforms,* Dresden, 1996.

[Honeyman92] Honeyman, P., L. Huston, J. Rees, and D. Bachmann. "The LITTLE WORK Project." *Proc. 3rd Workshop on Workstation Operating Systems,* Key Biscayne, Florida, U.S., 1992. IEEE Computer Society Press, Pages 11-16.

[ISO95] ISO/IEC Draft Recommendation X.902, International Standard 10746-1, "ODP Reference Model: Overview", January 1995.

[Katz94] Katz, R.H. "Adaptation and Mobility in Wireless Information Systems." IEEE Personal Communications Vol. 1 No. 1, Pages 6-17.

[Keeton93] Keeton, K., and R. Katz. "The Evaluation of Video Layout Strategies on a High-Bandwidth File Server." *Proc. 4th International Workshop On Network and Operating System Support for Digital Audio and Video*, Lancaster, U.K., Pages 237-248.

[MOST92] MOST. "MOST: Mobile Open Systems Technology for the Utilities Industry", *Project Proposal* Lancaster University and EA Technology. 1992.

[Mummert95] Mummert, L., M. Ebling, and M. Satyanarayanan. "Exploiting Weak Connectivity for Mobile File Access." *Proc. SOSP,* December 95.

[Neuman93] Neuman, B.C., S.S. Augart, and S. Upasani. "Using Prospero to Support Integrated Location-Independent Computing." *Proc. 1st USENIX Symposium on Mobile and Location Independent Computing,* Cambridge, U.S., August 1993. Pages 29-34.

[OMG91] OMG. "The Common Object Request Broker: Architecture and Specification (CORBA)", *Report 91.12.1,* The Object Management Group. 1991.

[OSF91] OSF. "Distributed Computing Environment: An Overview", OSF, April 1991.

[Satyanarayanan90] Satyanarayanan, M., J.J. Kistler, P. Kumar, M.E. Okasaki, E.H. Siegel, and D.C. Steere. "Coda: A Highly Available File System for a Distributed Workstation Environment." IEEE Transactions on Computers Vol. 39 No. 4, Pages 447-459.

[Schilit91] Schilit, B., M. Theimer, and B. Welch. "Customizing Mobile Applications." *Proc. 1st USENIX Mobile and Location-Independent Computing Symposium,* Cambridge, MA, August 2-3. Pages pp. 129-138.

[Schilit94] Schilit, B., N. Adams, and R. Want. "Context-Aware Computing Applications." *Proc. Workshop on Mobile Computing Systems and Applications,* Santa Cruz, CA, U.S., December 1994.

[Want92] Want, R., A. Hopper, V. Falcao, and J. Gibbons. "The active badge location system." 10 No. 1, Pages 91-102.

BIOGRAPHY

Nigel Davies graduated from Lancaster University in 1989 and later that year joined the department as a research associate investigating storage and management aspects of multimedia systems. As a result of his work in this area he was awarded a PhD in 1994. After a spell as a visiting researcher at the Swedish Institute of Computer Science (SICS) where he worked on mobile file systems he returned to Lancaster, first as site-manager for the MOST mobile computing project and subsequently as a lecturer in the Computing Department. His current research interests include mobile computing, distributed systems platforms and systems support for multimedia communications.