Published in Webnet '97, 1-2, 1997
which should be used for any reference to this work

Towards a Web Operating System (WOS)

Peter Kropf[1] John Plaice[2] Herwig Unger[3]

[1] Université Laval, Département d'Informatique, Sainte-Foy (Québec), Canada, G1K 7P4
[2] University of New South Wales, School of Computer Science and Engineering, Sydney 2052, Australia
[3] Universität Rostock, Fachbereich Informatik, D-18051 Rostock, Germany

Keywords. Distributed Systems, Adaptive Systems, World Wide Web, WOS^{TM} (Web Operating System).

1 INTRODUCTION

With the rapid development of new forms and concepts of networked and mobile computing, it is increasingly clear that operating systems must evolve so that all machines in a given network, even the Internet, can appear to be controlled by the same operating system. As a result, the world-wide interconnected networks, commonly called the Internet or the Web, could potentially be supported and managed by a giant virtual operating system [8].

For example, initially the World Wide Web was created to allow one to view remote hypertext pages on one's own machine, thereby facilitating collective work among geographically removed collaborators. Soon after, virtual pages, generated on the fly using tools such as cgi-bin [2], allowed the widespread remote execution of programs. More recently, with languages such as Java [1], it has become possible to download fully executable programs to one's own machine, and then to make them run on that machine. However, there is no general means for taking an arbitrary program and having it run somewhere on the network.

There are several reasons that this last possibility is actually essential. First, with the development of network-centric computing, there will be more and more limited-capacity machines (slower processors, limited memory or storage space, etc.), such as the NC computers [6], that will be forced to use more powerful computers on the network to effect any non-trivial tasks. Second, an arbitrary program on the network might just be incapable of running on the local machine, simply because it is the wrong platform (hardware, local operating system, running applications, etc.)

Implicit in the above discussion is the heterogeneous nature of most networks. The transparent use of such heterogeneous networks of computers has been partially addressed in work on metacomputing, whose objectives are to transform a network into one single computer system [3]. Recent developments in operating systems such as Inferno [9] or JavaOS [5] provide the user ubiquitous access to resources and information. However, the Web or the future global information infrastructure is more than just a metacomputer or a networked system of computers seen as a virtual machine run by a virtual (network) operating system, in that there is no complete catalog of all resources available. Moreover, such a catalog is infeasible, because of the highly dynamic and distributed nature of the Web or the Internet, continually integrating rapidly developing technologies.

2 WEB OPERATING SYSTEM: WOS

As a result, any attempt to design one single operating system offering a fixed set of resource-management functions will have difficulty adapting to technological innovation or to new demands. Therefore, there is, such as proposed in [4], a need for a Web Operating System (WOS), which would make available, to all sites on a network, the resources available on that network, or at least a reasonable subset thereof, to effect computations for which local resources are missing. These resources could be of many forms, including processor speed, available memory or storage space, available operating systems or applications, and so on. In order to deal with the dynamic changes in the system, the Web Operating System should be a versioned system, in which different versions of the operating system are running simultaneously on the network. Should, for instance, a given version not be capable of dealing with a particular request for a service, then it can pass it on to another version, as is currently done for packet routing.

What distinguishes the Internet from classical distributed systems is the fact that there is no complete catalog of all resources available and central decisions making for resource allocation is not acceptable or even impossible. Rather, the Web Operating System (WOS) [4] should be a versioned system, in which different versions not capable of dealing with a particular request for service, then pass it on to another version, as currently done for packet routing. Generalized software configuration techniques, based on a demand driven technique called eduction [7] are being developed, that can be used to define versions of a WOS to be built in an incremental manner. Software and hardware (description) repositories or warehouses will provide the necessary components for fulfilling a service requested.

Being able to use the global network for parallel/distributed execution of a program in the framework of a WOS is clearly promising. For this to be realistic, mechanisms are required to distribute the work, collect the results and coordinate the participating processes or agents. In particular, for such mechanisms to be effective, dynamic load balancing/sharing must be implemented. Therefore, we intend to apply advanced, prediction based load balancing/load sharing methods in a pre–reserved user space [10].

3 CONCLUSIONS

The general aim of out approach is to develop a family of services for illustrating and studying the concept of a Web Operating System (WOS) based on one single underlying concept, the demand-driven computation using simple warehouses, which hold and provide all the necessary information a system may offer to

a request. The ongoing work includes (1) production of sample resource managers and warehouses, together with the necessary automatic broadcast or 'resource-mining' mechanisms, (2) the implementation of a sample series of WOS-services (e.g. typesetting services, graphics processing, interactive simulations, etc.) and (3) implementation of a prototype user-interface based on browser-like forms to specify user (application) requests, which includes new 'data-mining' search engines.

REFERENCES

[1] K. Arnold and J. Gosling. *The Java Programming Language*. Addison Wesley, 1997.

[2] National Center for Supercomputing Applications. The Common Gateway Interface. `http://hoohoo.ncsa.uiuc.edu/cgi`, 5 February 1996.

[3] National Center for Supercomputing Applications. NCSA Metacomputer. `http://www.ncsa.uiuc.edu/General/CC/CCMetacomputer.html`, 25 July 1997.

[4] S.B. Lamine, J. Plaice, and P. Kropf. Problems of computing on the WEB. In A. Tentner, editor, *High performance computing 1997*, pages 296–301, Atlanta, Ga., 1997. SCS.

[5] Sun Microsystems. JavaOS. `http://java.sun.com/products/javaos` 25 July 1997.

[6] Sun Microsystems. Network computer reference profile. `http://www.sun.com/smi/Press/sunflash/mncrs-profile.html`, 25 July 1995.

[7] J. Plaice and S.B. Lamine. *Intensional Programming II*, chapter Eduction: A general model for computing. World Scientific, Singapure, 1997. to appear.

[8] F. Reynolds. Evolving an operating system for the Web. *IEEE Computer*, 29(9):90–92, 1996.

[9] Lucent Technologies. Inferno. `http://www.lucent-inferno.com`, 25 July 1997.

[10] H. Unger, H. Unger, and G. Hipper. LYDIA – load sharing for PVM applications in a workstation cluster. In *Proceedings of PDCS'97*, Washington, DC, 1997.