

# Exploration of Emerging HPCN Technologies for Web-Based Distributed Computing

H. W. Yau, A. Leung, W. Furmanski and G. C. Fox

Northeast Parallel Architectures Center  
Syracuse University, 111 College Place,  
Syracuse, NY 13244-4100, USA.

Tel. 315-443-1722, Fax. 315-443-1973,

E-mail: [hwyau@npac.syr.edu](mailto:hwyau@npac.syr.edu), [alvin@npac.syr.edu](mailto:alvin@npac.syr.edu), [firm@npac.syr.edu](mailto:firm@npac.syr.edu) &  
[gcf@npac.syr.edu](mailto:gcf@npac.syr.edu)

URL: <http://www.npac.syr.edu/> & <http://www.infomall.org/>

20th of February, 1996\*\*

**Abstract.** The surge in the popularity of the World Wide Web (WWW) has corresponded to a decreasing market for specialised high performance computers. This paper discusses how, by making use of technology developed from the broader end of the computing pyramid, much of the past decade's work in distributed computing can be realised in the context of the larger WWW market. Not only do these new technologies offer fresh possibilities, but their pace of development is unlikely to be matched by the traditional high performance research community. A motivating application, discussions of the pertinent emerging technologies, and NPAC's investigations of them, will be presented.

## 1 Introduction

This paper reports what the authors believe are key emerging technologies in the world of High Performance Computing and Networks<sup>3</sup>. The cornerstone of this is the National Information Infrastructure (NII) and its success within the World Wide Web (WWW), which will enlarge the market for the distributed computing paradigm into the area of Metaproblems[FWM94].

The following section 2 will discuss a driving application for the development of computing on the NII, with regards to activities in NPAC. Section 3 identifies the critical technologies are in use and in development, which will be pertinent for the application described in section 2. NPAC's rôle in evaluating this technology is described in section 4, and we shall attempt to describe our experience. Finally, the summary in section 5 will describe where we think we are, and how much

---

\*\* Submitted to HPCN 1996

<sup>3</sup> Also known as 'High Performance Computing and Communications' in the United States.

further we need to go before attaining the goals in section 2. In addition, we will point out known problems, and speculate on possible future ones.

## 2 Driving Application

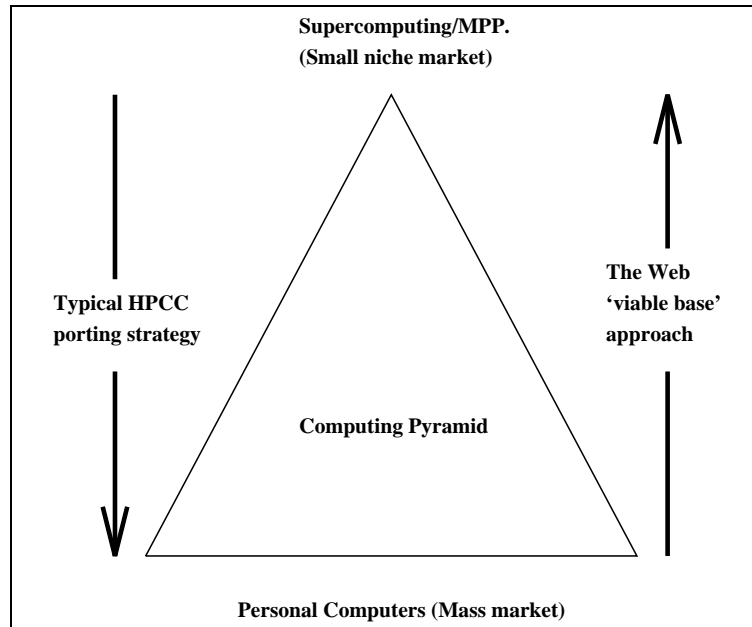
The need for distributed computing is as strong as ever, with the scientific and engineering sectors still at the forefront of its exploitation. However, whilst these applications have typically used specialised, tightly coupled processor, machines we believe the addition of a pervasive, long distance communications network and the emergence of software technology to make use of it, has opened up opportunities for other compute models.

A good example of an application whose use of the NII is being planned from the offset is the Affordable Systems Optimisation Process (ASOP) consortium, made up of major US aerospace companies, NASA Langley, aircraft engine makers, and NPAC, to examine the problems of interdisciplinary collaborative design and manufacture[FF95a]. For this, high speed links ( $\geq 10$  Mbits/s) to each site are expected, delivering full multimedia information. The motivation for this is the complexity of designing a modern aircraft: hundreds of contractors, thousands of engineers, and the use of thousands of computer programs. Web technology is proposed to bind the management of such large projects, provide a uniform interface to all the different computing platforms, and allow the management of remote executions —for example, a database engine on an enterprise server, or a CFD calculation on an MPP.

## 3 Key Emerging Concepts and Technologies

Usage of the Internet has soared in the last few years, from an arcane text-based means of exchanging data amongst academics to a multi-million dollar industry expected to form the next computing wave. Whilst there is an inevitable degree of exaggeration in the purported ability of the World Wide Web's benefit to society and the economy, this popularity has had one very obvious consequence: the remarkable pace of development in Web technology. For the first time, the use of the Web as a distributed computing resource[Fox95] may become a reality, at a far more rapid rate than expected. We can attempt to illustrate the reason for this growth with the 'computing pyramid' in figure 1. Traditionally, high performance computing and networking technology have been developed at the top by high end, mostly federally funded users, where it was then encouraged to move down into the broader user base. However, today we believe that the advent of the Web has allowed technology intended for the broader compute market to be made available to those same high end users. Indeed, just as the market for traditional supercomputers is shrinking, investment and resources have flowed into the development of Web technology.

The first observation is that graphical Web browsers can be looked upon as a portable, open, operating-system independent graphical user-interface. The



**Fig. 1.** The computing pyramid, to illustrate the difference between the traditional and the anticipated flow of distributed computing technology.

logical extension of this was explored by NPAC with WebWindows[FF95b], a project to extend the features of Web browsers to include the applications users expect from a modern computing environment. However, like all browsers from that era, the original components for WebWindows were done using Common Gateway Interface (CGI) programs, which resulted in a client (the browser) server (the host machine) compute model, with the computation taking place in the remote server.

The breakthrough to a server-server model has been made possible by the concept of the Java programming language[Mic95] from Sun Microsystems and the availability of browsers which can interpret so-called Applets written in that language. Java itself is an object-orientated language much simplified from C++, but what makes it important is its ability to be compiled down to a portable intermediate form (Java Bytecode) which can be executed on any machine with the appropriate interpreter (Java Virtual Machine —‘JVM’). This allows Java Applets to be downloaded in Bytecode form from a host machine, to be executed by the JVM inside the client browser. We have evaluated the Java technology from the viewpoint of providing a portable distributed computing environment, and the conclusions from that will be discussed in the following section.

Another technology NPAC is evaluating is the Virtual Reality Modelling Language (VRML), originally proposed by Silicon Graphics Inc. The currently agreed standard is recognised as being mostly a proof of concept, but this may

change with the definition of VRML 2.0, which aims to address the lack of support for dynamic images and multimedia data as well as lower the bandwidth cost of sending the description of a scene.

The application outlined in section 2 depended on traditional HPCC technology for communications between tightly coupled processors, as would be found in an MPP machine. However, for communication across a wide area, the current means would be via custom communication calls written on top of the existing HTTP Web protocol. However, the problem of specifying how data is to be communicated across the Web has yet to be fully addressed. One promising approach is to marry the object-orientated approach of the Java language with the CORBA open standard for linking distributed objects[Gro96] from the Object Management Group. This is currently being pursued by the company Iona as an extension of their existing CORBA compliant object request broker[Ltd96]. On reflection, the use of CORBA to communicate objects should be more promising than the obvious alternative of (for example) an MPI binding in Java.

Finally, one other piece of technology currently being explored in NPAC are parallel databases interfaced to the Web. Virtually all the major database vendors now have a parallel version of their product, running on a variety of (usually shared memory) multiprocessor machines. A web interface would be crucial in allowing the database engine to be accessed remotely, in a portable manner.

In addition, we can assume the further development and use of the following existing technologies: the NII, driven by the needs of commerce and consumers; the Message Passing Interface as the underlying, portable, means of sharing data in a distributed computing environment; and High Performance Fortran, this is presently optimised for data-parallel applications, but work is currently underway to allow it to tackle larger-grained 'task-parallel' problems, expanding its applicability to loosely synchronous problems.

## 4 NPAC Experiences with the Emerging Technologies

NPAC is currently working on a number of projects which involves detailed examination of the technologies described in the above section. The pertinent projects are:

- The visualisation of data from an integrated weather, terrain renderer, and electromagnetic scattering simulation, in the context of a command and control tactical scenario. This will demonstrate the application of VRML as a common graphics display standard, as well as explore the issues in communicating objects between the different applications.
- Monte Carlo simulation of option pricing on demand, running on parallel compute resources, and using the latest online data from the stock market.
- Collaboration with the Syracuse University department of physics to provide simulations on demand to augment the teaching of neural networks. This will replace the current CGI executables of the simulations with Java Applets,

hence removing the computational bottleneck of students overloading the CGI server.

- The Syracuse University Living Schoolbook project[MFC<sup>+</sup>95], has demonstrated use of the NYNET ATM Wide area network in the state of New York, and the use of VRML linked to a geographical database for real time terrain rendering.

From this, we shall draw some conclusions regarding the applicability of the Java and VRML concept with respect to the ASOP problem outlined in section 2.

First, the question of whether one can build a distributed computing environment with Java technology. Before we answer this, we must first distinguish the two instances of Java programs. First there are the aforementioned Java Applets, programs which are downloaded to be executed on a client's browser; and second there are Java Applications which are compiled and executed by a user on his/her workstation as with a traditional program. In both cases, a JVM is used to execute the compiled Bytecode, but with different security restrictions brought about by their intended execution environment. So for example, a Java Applet is unable to write a file to the client system but can do so on the host; and it can only make a socket connection from the client to the host. In contrast, a Java Application can do arbitrary I/O on the machine it is running on, and make remote requests to whatever sites that it can access.

One obvious problem with the Java concept itself is that it is an entirely new language, with a negligible software base. In addition it is an interpretive language and is currently about twenty times slower than an equivalent binary compiled from 'C' (although this performance gap is expected to narrow). The solution to both these problems is with the so-called Native Method Class. This allows the programmer to specify a Java wrapper which calls a library routine already compiled to execute on the client machine, such as a matrix solver package. It is important to note that compiled binary is never sent to the client machine, instead the Java program will make use of libraries already resident on that machine. One interesting variation would be for the Java program to use method overloading to provide an alternative, slower, version of the requested library in Java, should a native compiled version not be available. How this all works in practise is currently under investigation.

With regards to VRML, the main point to note is that it is designed to provide a description of three dimensional objects. This is sufficient for purely visualisation purposes, but means it is unlikely to be sufficient for use as the *lingua franca* between (for example) the CAD, CAM, aerodynamicist and stress engineer in the design of a new aircraft. Instead, VRML should be used as a component part in a wider description protocol, which would include information on the material, mechanical properties, mass, and cost of a given component.

## 5 Opportunities and Pitfalls

We believe the use of the Web in the manner outlined above represents an important computing paradigm shift, that of pervasive distributed computing.

The potential opportunities which can be exploited to attain this goal include the software technologies discussed, as well as the need for high performance servers (most likely MPP) within the NII, and the need for cheap high speed wide area networks. These will continue to be developed largely independently of the state of federal funding to research institutes.

However, we do foresee problems. Some of these are technical, but others are sociological and more difficult to overcome. First of all, there is the reluctance people to store and disseminate sensitive information using the Web. This is particularly necessary for any commercial collaboration to take place, and the techniques for ensuring this can be done securely on the Web is by no means trivial. On the other extreme, we do perceive a need for Java Applets to be less strict with its security under certain circumstances —*e.g.*, when used inside a firewall within which are only trusted clients. These restrictions should not be an issue for Java Applications, but use of this would be at the loss of the common GUI provided by the browser. Finally, we do regard the lack of a Java software base as a problem. We have outlined how this may be circumvented using wrappers to trusted library binaries. The alternative of using machine translators would be difficult given the different design philosophies between object and procedural orientated languages. How the compiled library, the Java code, and the browser interact for different real world applications will be a useful line of investigation, and an essential step towards a Web-based distributed computing environment.

## References

- [FF95a] G C Fox and W Furmanski. The use of the National Information Infrastructure and High Performance Computers in industry. *NPAC Technical Report SCCS-732*, <http://www.npac.syr.edu/techreports/html/0700/abs-0732.html>, 1995.
- [FF95b] G C Fox and W Furmanski. WebWindows: Motivation and Application to Distributed Computing. *Talk and demonstration of WebTools at CRPC Annual review, Houston Texas, also* <http://kayak.npac.syr.edu:2005/WebTools.html>, March 1995.
- [Fox95] G C Fox. High Performance Distributed Computing. *NPAC Technical Report SCCS-750*, <http://www.npac.syr.edu/users/gcf/sccs750/>, 1995.
- [FWM94] G C Fox, R D Williams, and P C Messina. *Parallel Computing Works! Morgan Kaufman Publishers, Inc.*, 1994.
- [Gro96] Object Management Group. Information Resources on CORBA and the OMG. <http://www.acl.lanl.gov/CORBA/>, 14 February 1996.
- [Ltd96] IONA Technologies Ltd. Orbix for Java. <http://www.iona.ie:8000/www/Orbix/OrbixJava.html>, 1996.
- [MFC<sup>+</sup>95] K Mills, G C Fox, P D Coddington, B W Mihalas, and M Podgorny. The Living Textbook and the K-12 classroom of the future. *Proceedings of SuperComputing '95, also see* <http://www.npac.syr.edu/projects/ltb/SC95/index.html>, 1995.
- [Mic95] Sun Microsystems. Java homepage. <http://java.sun.com/>, 1995.

This article was processed using the L<sup>A</sup>T<sub>E</sub>X macro package with LLNCS style