High Performance Computing For All

James Allwright, David Pritchard,
Department of Electronics and Computer Science,
University of Southampton
Southampton
SO9 5NH

Stephen Jones, Nelson Stephens,
Department of Computational Mathematics,
University of Wales College of Cardiff
PO Box 78
Cardiff
CF1 1XL

Andrew Peel, Gordon Makinson,
Computing Laboratory,
University of Kent
Canterbury
Kent
CT2 7NF

John Steel, Heather Liddell
Department of Computer Science
Queen Mary and Westfield College,
Mile End Road
London E1 4NS

Project Coordinator and Author for Correspondence
Dr G.J.Makinson
Computing Laboratory
University of Kent
Canterbury
Kent CT2 7NF

Email g.j.makinson@ukc.ac.uk

Tel No 01227 82 3819
Fax No 01227 762811

High Performance Computing For All

Gordon Makinson
Project Coordinator

Computing Laboratory
University of Kent
Canterbury
Kent
CT2 7NF

Email
g.j.makinson@ukc.ac.uk

ABSTRACT

A Consortium of four partners - the University of Kent, University of Southampton, University of Wales College of Cardiff and Queen Mary and Westfield College have been pooling efforts to produce hypertext courseware to help teach High Performance Computing, jointly funded under the TLTP Phase II Programme by the Higher Education Funding Councils HEFCE, HEFCE,SHEFC and DENI.

By high performance Computing we mean all aspects of parallelism:-

Data Parallel concepts, Paradigms, Algorithms, Languages such as Fortran 90, High Performance Fortran, Occam, Message Passing.

Architectures such as the Single Instruction Multiple Data(SIMD) eg. Maspar, Multiple Instruction Multiple Data (MIMD) eg Transputer Systems, NCube etc

Users can choose the section of material, page order, point to clickable maps and explore to different levels of detail by following hyperlinks. This interaction only allows the user control over navigation.

Electronic forms allow data input and this is going to be used in a variety of ways to provide self assessment and input of source code for example.

Password control is another feature now available which has enabled control of particular resources to be set up and this opens the way for safe remote access.

The multi media capabilities of the WEB system are now well known but at present the project has not sought to avail itself of for example, video or sound, but future enhancements will certainly do so.

1. Introduction

The project to develop courseware for the teaching of High Performance Computing has presented the developers with a succession of challenging choices stemming from the initial project design decision to opt for a networked hypertext system for delivery. The initial choice of authoring with UNIX Guide was abandoned when the potential advantages of using the WEB standards became apparent.

To take advantage of the evolving public domain WEB browsers for a range of machine types it was decided to write material in HTML and to use the WWW server technology.

The consortium brought together for the project consists of four partners - University of Kent, University of Southampton, University of Wales College of Cardiff, and Queen Mary and Westfield College. This gave a natural breakdown of the material to be covered corresponding to the specialised hardware available at the four sites and the expertise that is associated with each site.

Occam and transputer technology has long been a feature of Computing at Kent. Southampton have been actively involved in the use of and the formulation of standards for Message Passing. Cardiff has studied the use of algorithms and techniques and in particular applied them to an N-Cube machine. Queen Mary and Westfield has had a long history of involvement in Data Parallel work, firstly on the DAP and more recently on the Maspar.

It was natural to separate out the development work into the four different specialisms.

The team of academics and developers associated with the project are as follows :-

James Allwright, David Pritchard, University of Southampton

Stephen Jones, Nelson Stephens, University of Wales College of Cardiff

Andreww Peel , Gordon Makinson, University of Kent

John Steel, Heather Liddell, Queen Mary and Westfield College

This paper describes how the project has evolved as a result of the WEB advances and how the strategy for producing courseware that makes use of 'state of the art' technology is succeeding in its original aims. The design of development tools, written in Perl, to speed the authoring process has been necessary. In particular this was needed to provide the interactivity that has always been regarded as essential.

2. A Multi-Site Development Team

Various factors made the early progress in the project less productive than we would have liked. The funding of a half post at each of four sites on a renewable annual basis meant that the staffing of the project has been incredibly difficult. In some instances it was not possible, for a while, to find matching funding to be able to offer a full time post.

In other instances those appointed did not stay in post very long before leaving to take up a post concentrating solely to one activity or one offering more than the prospect of a maximum twelve month horizon. This has of course impeded progress.

Communication between sites has been conducted by email and by producing text in HTML and gifs for viewing over the WEB.

Occasional face to face meetings have been held but the distances involved have precluded holding many of these.

Whilst email is quick and convenient to send and multicast messages are also very simple to direct, the nature of the dialogue and the response to receiving messages makes this form of communication very different from normal day to day discussions with colleagues. It has been responsible for a surprising amount of tension and misunderstanding as well as resulting, at times, in delays in the multi-way discussion process.

It is true to say that lessons have been learned and that the team are now approaching the final stage of the project with a coherent strategy and a more confident perspective than twelve months ago.

Using WEB pages and email for communication puts the onus on all the developers to make a deliberate and conscious effort to study them and respond with a suitably composed reply. The temptation for each developer is to pursue their own pressing programming priorities rather than contribute to the discussion of the project's more general design aspects when these are being raised elsewhere. Basically the developers have been carrying out a role much closer to a research role than a developer's role and this has come about because of the change of direction to a emerging technology.

There is no doubt however that the move, which has made the programming challenge more difficult and, of course, more interesting, is clearly justified in giving the courseware the universal availability that it would not have had.

The use of the public domain browsers such as Netscape or Mosaic ensure that the courseware has a high degree of common 'look and feel' regardless of where it was authored. There is nevertheless a need for extra navigation provision and other functionality which should have a similar common basis for the courseware being authored at each of the four partner sites.

The following features have been agreed to take a common form

1) the enhancements that are to be provided to HTML pages ie. the extra navigation icon buttons in terms of their number and form and their functionality.
2) The 'length' of standard WEB pages ; this reconciles the need to eliminate large amounts of scrolling and repetitive clicking of the mouse button to progress through a section of coherent material.

3. Design Considerations

Various design options were predicated by the decision to use HTML with WEB browsers over a network. The hyperlinks and their presentation, layout, style and other windowing aspects are determined by the particular Browser chosen to view the courseware.

3.1. Interactivity

Interactivity was planned as a key element of the courseware from the outset. It was available in UNIX GUIDE and we would not have changed from this as an authoring system had it not also been possible with the WEB technology.

Interaction has been achieved at the hypertext level by the use of

1) hypertext links
2) active gifs
3) icon buttons/boxes

In addition interactivity has been achieved with the use of

i) electronic forms
ii) use of CGI (Common Graphics Interface) scripts

The CGI scripts allow other application programs to be run. This facility has been used for the input of source code, editing, compilation of source code in various programming languages such as occam,C, Fortran and the running of executable code.

The developers have devised various means for the presentation of the HTML files. As a result there are three modes which can be adopted: pre-processed files, files constructed on the fly, or a mixture of both.

Operational experience will enable us to decide the extent to which the courseware is presented in which mode.

4. Hardware Requirements

The hardware needed to run the courseware has been dictated by the ability of the 'workstation' to run the WEB browsers at sufficient resolution and with enough colours and performance to be effective.

We were aware in our choice of this method of delivery that WEB browsers were being written for many types of 'workstation' ranging from the PC, Macintosh, UNIX workstation or X Terminal thus giving the courseware the degree of machine independence that we sought.

It is also true that what was top end of the range in 1993 has become more commonplace and we are anticipating that HE institutions will be moving towards having an infrastructure that can make use of the courseware as we complete the end of the project in a year's time.

The infrastructure required will consist of network 'workstations' with access to a server which is based on a host capable of processing many interactive processes. Fortunately also the pattern of operation can be modified so that the courseware can be used in other hardware configurations including on a stand-alone machine.

5. Development Tools

There exist some commercial and public domain tools for converting existing text files and such files as Latex or Wordperfect to HTML.

Other development tools for authors producing HTML use the package Framemaker and then convert using Webmaker. More such tools are appearing all the time.

The project began by constructing material directly in HTML. It has subsequently constructed development tools and macros to assist the developer. One such tool has been constructed to reduce the burden of retyping and editing repeated sections and addresses. A typical file name uses a Uniform Resource Locator(URL) which can be exceedingly long.

Other development tools have been built which assist the developer by adding, to the Web pages, navigation icons either at the top or the bottom or both top and bottom of the Web page.

These enable the Browser to view processed pages with a set of universal navigation icons added to the original HTML thereby giving all the pages a common 'look and feel'.

They build into the courseware hypertext functionality which makes the pages into an easily read tutorial with extra facilities for jumping to a glossary, back to a home page, or to related pages, the titles of which are clearly indicated. A student can also search through an index for a specific item from any page.

For a particular course the material to be covered may consist of some core local material and a large amount of background external material.

A student who is following through the material can be given opportunities to divert to the external material. When this happens the student is informed that this material is no longer local core material and an icon is provided to enable a return to be made back to the departure page from the core, regardless of the depth to which the student has departed. This facility was first published as part of the Interact project [Nicol,Smeaton & Slater 1995].

6. Pedagogic Issues

Three issues have dominated the pedagogic debate.

1) Should the courseware be presented to the student in a linear fashion.

2) How flexible can the courseware construction be made so that an academic can tailor the material either by omitting material or adding material or just changing the sequence by which the student might be constrained to view the material.

3) Should the student be presented with a great many small WEB pages and hence have to click repeatedly or should it be a matter of scrolling through a 'long' page.

On the question of linear sequencing it is agreed that for some of the subject matter the 'correct' way to present it is in a linear fashion but that the freedom to explore tutorials in arbitrary sequence should still be allowed. Students may wish to take very different routes through the material, in subsequent returns to the material, as opposed to the first time round.

The length of a normal sized page has been kept small. This has been influenced partly by the operational requirement that delay in loading a page should be kept as low as possible and partly because of the belief that the readability and comprehension was better viewing new pages than scrolling through a long file.

It also means that additional links can be provided between the pages and thereby give greater flexibility in the choice of order of viewing. Experience with this mode of teaching and details of related work has been published by a group including HPC Consortium members at Cardiff. [Hurley,Marshall,McIntosh-Smith & Stephens 1994], [Campbell, Hurley, Jones and Stephens 1995].

7. Remote Compilation and Execution of Code

A tenet of the project has been that the student should not just read about the subject but have the opportunity to study example source code and its behaviour. Further it should be possible for the student to have the means of testing the effect of modifications to the existing code and of testing his or her own code. Courseware to do this has been written. This has been achieved by use of the CGI script mechanism.

The implementation of this feature is not straightforward. In the case of running occam on a transputer for instance there are a good many stages to the process. It requires communication from the workstation to the server on a UNIX system.

>From that server communication is made to the UNIX host for the transputer system. From there, there is communication using the Inmos Occam Toolset to the transputer system itself. The communication process has to be repeated in reverse to allow the executing code to communicate its output back to the workstation on which the student initiated the process.

When the output is returned at whatever stage - compilation errors - run time errors - run time output - it has to be converted by the server to HTML so that the browser can display it on the workstation screen.

Remote access to a specialised machine in this manner can be password protected. It offers the real possibility of the use, by many students, of facilities, which for whatever reason - cost of hardware or support - cannot be located at every HE institution for training purposes.

Integrated Teaching Environment

It is now apparent that the use of a networked WEB courseware facility, such as in this project, brings more than just easy access to some CAL material. The possibility of incorporating an extensive 'library' of related material, which is authored elsewhere can be provided by adding the necessary hyperlinks. Search facilities are included and simplify the finding of any topic.

Sample sections of the courseware are visible on the Web even though none of it has been released yet. The URL is http://larch.ukc.ac.uk:2001/

Self Assessment can be incorporated . In the first instance the use of multi-choice questions and input by means of electronic forms, is planned.

Courseware for related courses can also be interlinked and the whole collection can be integrated with exercises, assignments, notices and other class communications.

8. Limitations of this approach

The most obvious limitation of adopting the WEB approach to teaching is that of not having exclusive control of a single machine. This means that network traffic and other load on the server machine can interfere with the performance of the courseware.

Performance could therefore degrade to unacceptable levels.

Applications, such as simulations, which are illustrating a continuous or time dependent activity can fail to give the correct appearance as the processes becomes delayed in arbitrary ways.

9. Future Possibilities

Already the deficiencies inherent in the system described above are being addressed by organisations such as the hardware manufacturers and others writing new browsers.

Sun Microsystems has produced a new browser called HotJava which allows 'applets', written in the language Java, to be embedded in the HTML. The 'applet' triggers a local application and hence allows simulations to be run which will not be interfered with by the network traffic.

The courseware that the project has developed has not made use of multi media but the extension to include video clips and sound is a logical one and already catered for in the browsers. The work to build in such material has been held back on the grounds that the wide availability of networked 'workstations' with the capability of handling such material is still some way off.

10. Acknowledgements

The High Performance Computing Courseware project has been jointly funded by a grant under the Teaching and Learning Technology Programme(TLTP), from the Higher Education Funding Councils, HEFCE, HEFCW, SHEFC and DENI. This funding is gratefully acknowledged.

11. References

Nicol D, Smeaton C, Slater A.F. "Footsteps : Trail - blazing the Web" in Third International World-Wide Web Conference, Darmstadt, Germany, April 1995.

On line at http://www.igd.fhg.de/www/www95/papers/60/footsteps.html

Hurley S, Marshall A.D, McIntosh-Smith S. N, and Stephens N.M. "Courseware for parallel computing using mosaic and the world wide web" in 2nd World Wide Web Conference, Chicago, Illinois, Oct 1994.

On-line at http://.ncsa.uiuc.edu/STG/IT94/Proceedings/Educ/hurley/www.html

Campell K, Hurley S, Jones S B and Stephens N.M. "Constructing educational courseware using NCSA mosaic and ythe the world wide web" in Third International World-Wide Web Conference, Darmstadt, Germany April 1995

On-line at http://www.igd.fhg.de/www/www95/papers/52/www3.html