Short Communication

The Graphical Benchmark Information Service

MARK PAPIANI, ANTHONY J. G. HEY, AND ROGER W. HOCKNEY

Department of Electronics and Computer Science, The University of Southampton, Southampton, SO17 1BJ, U.K.;
e-mail: mp@ecs.soton.ac.uk

ABSTRACT

Unlike single-processor benchmarks, multiprocessor benchmarks can yield tens of numbers for each benchmark on each computer, as factors such as the number of processors and problem size are varied. A graphical display of performance surfaces therefore provides a satisfactory way of comparing results. The University of Southampton has developed the Graphical Benchmark Information Service (GBIS) on the World Wide Web (WWW) to display interactively graphs of user-selected benchmark results from the GENESIS and PARKBENCH benchmark suites. © 1995 by John Wiley & Sons, Inc.

1 INTRODUCTION

The Graphical Benchmark Information Service (GBIS) provides the following features:

1. Interactive selection of benchmark and computers from which a performance graph is generated and displayed.
2. Default graph options that can be changed:
   a. Available output formats; gif, xbm, postscript, or tabular results.
   b. Selectable ranges or autoscaling.
   c. Choice of log/linear axes.
   d. Key to traces can be positioned as required.

3. Links to PARKBENCH, GENESIS, and NAS information available on the World Wide Web (WWW) (including documentation and source codes).
4. Details of how to send in results for inclusion in the Graphical Interface.

The motivation for the GBIS stemmed from the existence of the GENESIS [1] and PARKBENCH [5] distributed memory benchmark suites. These benchmark suites contain several categories of benchmark codes that test different aspects of computer performance. The PARKBENCH suite is divided into four categories: low-level, kernel, compact application, and HPF compiler benchmarks. The codes within these categories were developed specifically for PARKBENCH or adopted from existing benchmark suites. For example, the low-level codes within PARKBENCH mirror those in GENESIS, whereas several of the kernel benchmarks are taken from the NAS parallel benchmarks [2]. The development of GENESIS

low-level codes will be frozen in the near future, to be replaced by the PARKBENCH suite, which will be used for the GBIS.

The PARKBENCH suite contains single-processor and multiprocessor codes that use the PVM (and in the future MPI) message-passing interface. Both types of codes produce tens of numbers as results and the most satisfactory way of displaying these numbers is in a graphical manner.

Many of the multiprocessor codes are designed to be run numerous times, on the same computer, with the number of processors varied each time. A performance figure in Mflop/s is obtained for each run. By plotting a graph, called a trace, of performance against number of processors, the observed scaleup can be seen, i.e., the change in performance that results when more processors are used to solve the same sized problem. By incorporating performance traces for different computers on the same graph, it is possible to compare the relative performance of different computers, as well as the actual scaleup.

Another factor that can be varied for several of the benchmark codes, is the problem size. Traces of temporal performance (e.g., timesteps per second against the number of processors) can be plotted for the same computer. with a different problem size associated with each trace. The observed scaleup can be seen from the relative position of the traces, i.e., the increase in the size of problem that can be solved in the same time period when more processors are used.

The GBIS allows several other types of graphs to be plotted depending on the type of performance metrics that a particular benchmark produces. These performance metrics include benchmark performance, temporal performance, simulation performance, speedup, and efficiency. These metrics are defined in Hockney [4] and the PARKBENCH report [5].

The low-level benchmarks in the PARKBENCH suite measure performance parameters that characterize the basic machine architecture. These parameters can be used in performance formulae that predict the timing and performance of more complex codes. Hockney and Jesshope [6] describe the low-level parameters. These low-level benchmarks output vectors of numbers that can be satisfactorily displayed and analyzed as graphs. For example, the GBIS allows a choice of two graph types to be plotted for the COMMS1 benchmark from the PARKBENCH (and GENESIS) suites; message transfer time or transfer rate can be plotted against message length.

2 IMPLEMENTATION OF THE GBIS ON THE WWW

2.1 Overview

The aim of the GBIS is to make the graphical results for GENESIS and PARKBENCH as widely available as possible. For this reason the WWW was selected to implement the service, and any software used was to be freely available in the public domain. The GBIS was therefore developed using the National Center for Supercomputing Applications [NCSA], University of Illinois, Mosaic Internet information browser (version 2.4 for the X Window System). In particular, the Mosaic Fill-out forms facility was used to create the user interface. Mosaic forms allow user interaction with the WWW beyond the use of hypertext links available within HyperText Markup Language (HTML). It is possible to use widgets on a WWW page including, radio buttons, text entry fields, and selection buttons. These provide an ideal method for allowing the user to select results of interest and also to select the desired graph options. The method used by the GBIS to process the forms is to use Unix Bourne Shell scripts running on the WWW server machine, at the University of Southampton.

The user first selects a benchmark. A Bourne shell script processes this selection and returns a Mosaic form consisting of a list of manufacturers, for which results exist. A number of manufacturers are then selected, which produces another Mosaic form displaying all machine results for the chosen benchmark plus manufacturers. Transparent to the user, each machine result corresponds to a data file of cartesian coordinates stored on the WWW server machine. These data files are used in conjunction with the public domain plotting package, GNUPLOT (Unix version 3.5, patchlevel 3.50.1.17, 27 Aug 93), in order to plot the graphs.

The GBIS uses the information processed from the Mosaic forms to create an appropriate GNUPLOT batch file. This file contains a list of commands that provide a means of setting graph options (e.g., axes scaling, axes types, etc.) to default or user-selected values and for naming the data files to be plotted. The shell scripts then execute GNUPLOT using this batch file, on the WWW server machine, to produce a postscript file of the graph as output. Depending on the format the user selected for the results, this postscript file is made available for viewing by a hypertext link

Table 1.   Summary of Steps Involved in the GBIS Operation, from the User and System Points of View

User Actions	GBIS System Operation
	1. Display GBIS home page.
2. Select the link to Graphical Benchmark Results (Fig. 1).	
	3. Display Mosaic form showing available benchmarks.
4. Select benchmark of interest (Fig. 2).	
	5. Obtain HTML file which lists all available manufacturers for the selected benchmark and create Mosaic form (see section 2.4).
6. Select manufacturers of interest (Fig. 3).	
	7. Obtain one HTML file for each manufacturer selected for the chosen benchmark. Create Mosaic form of available machine results (see section 2.4).
8. Select machines of interest. Also, accept default graph options or elect to change defaults (Fig. 4).	
	9. Store names of results data files, corresponding to each machine selected, in a temporary file.
	10. If default graph options are being used, create temporary GNUPLOT batch file. Run GNUPLOT to create a temporary postscript file, convert to a GIF file and return this as a WWW page. This ends the process if default graph options are accepted.
	11. If change defaults was selected display the Mosaic form showing available graph options.
12. Select graph options (Fig. 5).	
	13. If text results are selected display contents of each file, the names of which are listed in the temporary file of results data file names (step 9, above).
	14. Otherwise, create GNUPLOT batch file using data file names from the temporary file, and add GNUPLOT commands to change the default graph options, as requested. Run GNUPLOT to create a temporary postscript file. Provide WWW link to this file if postscript output format was requested or convert postscript file to other requested format and return result as a WWW page.

from a WWW page, or, is converted to another format (e.g., GIF) that can be displayed directly on a WWW page.

The processes described above generate a number of temporary files on the WWW server machine. These files are named using the process id of the Unix shell script that creates them. In this way it is possible for multiple users to simultaneously access the GBIS without any conflict.

The GBIS operation described above is shown in Table 1, from the user and system points of view (also see Figs. 1–5).

2.2 Directory Structure Naming Convention for Storing Results Files

The results data files associated with each machine are stored as a hierarchy of Unix text files on the WWW server machine. This method was chosen due to the diversity of different vectors of results that need to be stored for the different benchmarks.

The convention used for the directory structure is benchmark name/machine manufacturer/result file name, where result file name is in the form: (machine model)_(date benchmarked DD.MMM.YY)_(problem size, if applicable)_(benchmark metric). For example, MULTIGRID/IBM/SP-2_10.AUG.94_CLASSB_PERF.

The machine model, date benchmarked, and problem size, from the results files name, are used to label traces on the graph; the benchmark metric is used for choosing labels for the graph axes and in selecting the choice of log or linear axes for the default graph format. For some benchmarks the result of a single run (on a single machine) corresponds to several results files in the GBIS database. This is to allow the results to be plotted in different ways. For example, if the benchmark produces results that can be plotted as benchmark performance, temporal performance, or simulation performance against the number of processors, then three different files of coordinates must be stored. These are distinguished by
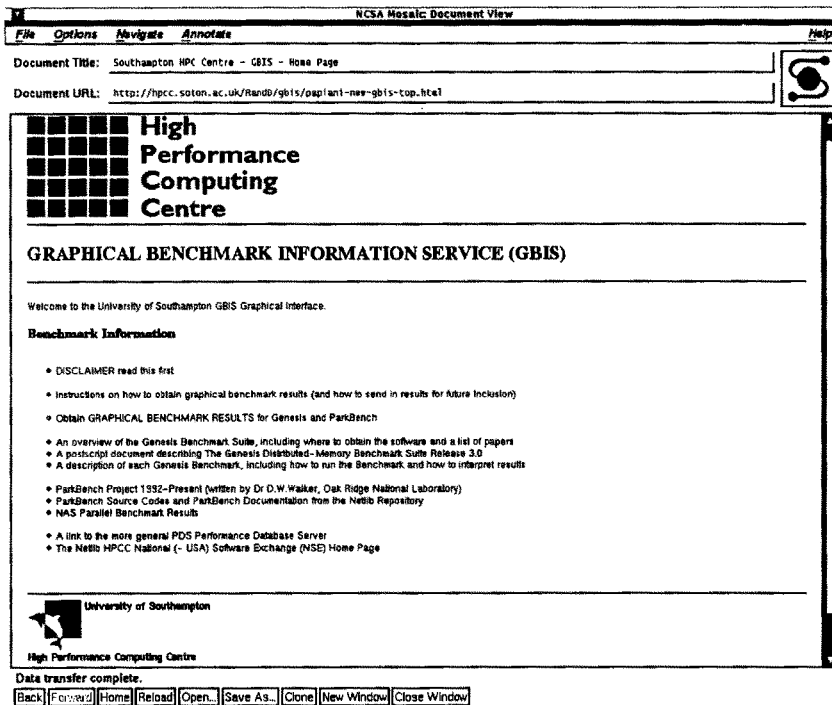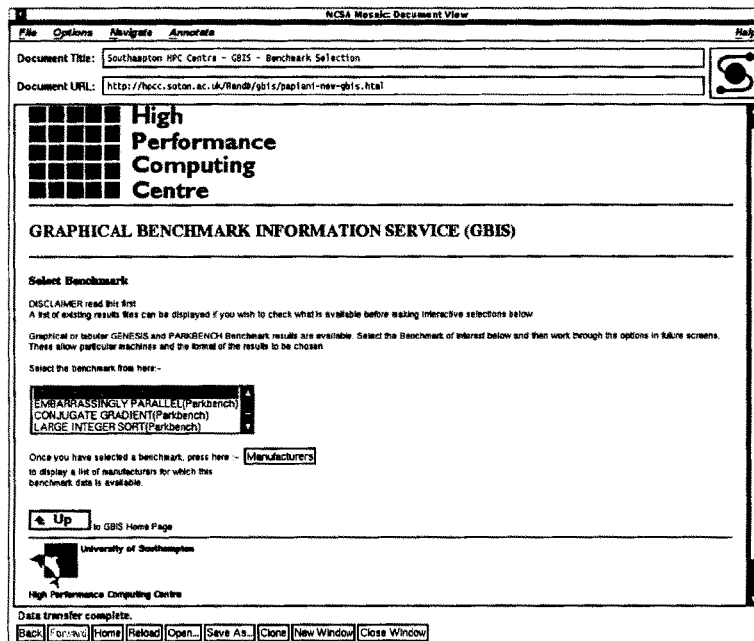
FIGURE 1   GBIS WWW home page.

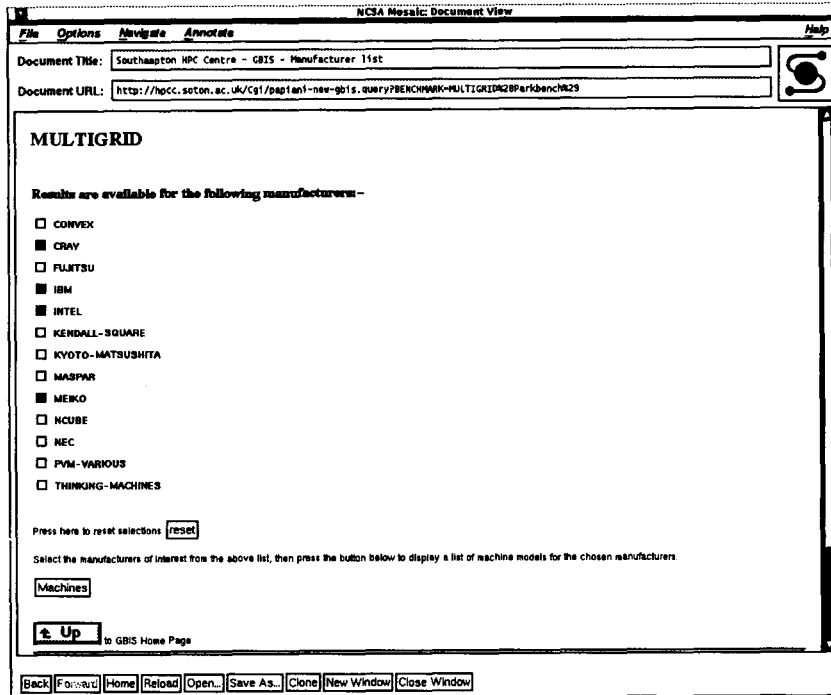**FIGURE 2**   GBIS WWW select benchmark page.

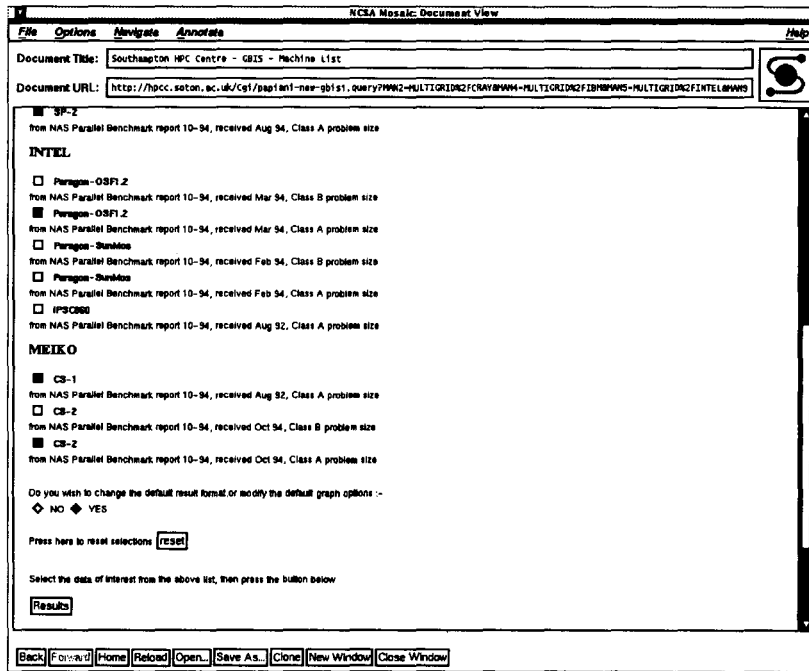**FIGURE 3**   GBIS WWW manufacturer list page.

FIGURE 4   GBIS WWW machine list page.

FIGURE 5 GBIS WWW change defaults page.

using a different benchmark metric in the filename so that the appropriate file can be found, for the type of graph requested.

2.3 Format of Results Files

The actual contents of results data files are in a format that GNUPLOT can use. This consists of lines that contain coordinates and comment lines that begin with a # character. The availability of comment lines was used to incorporate additional information used by the GBIS system. This is explained with reference to the example file contents shown in Figure 6.

The path and filename for this file is: COMMS1/MEIKO/CS-2_26.MAY.94_TIME.

```
#! CS-2 at the University of Southampton.
## Genesis 3.0, PVM, Fortran 77,
## 26 MAY 94
#
#Message Length / Byte        Transfer Time / us
0                             1.983E-04
1                             2.001E-04
2                             2.006E-04
3                             2.007E-04
#
#RINF=9.180 MByte/s
#NHALF=1791.949 Byte
#STARTUP=195.200us
```

FIGURE 6 Example contents of a GBIS result data file.

The file contents are plotted if the user selects transfer time against message length. For the COMMS1 benchmark, the transfer rate can also be plotted against message length. Therefore, a file containing different coordinate values also exists, with a name ending in _RATE.

As well as lines containing coordinates, there are lines beginning with #!, ##, and #. Because each of these lines contains at least one # at the start they are treated as comments by GNUPLOT. The line beginning #! is used to produce an HTML file (see Section 2.4), which is used to construct the Mosaic form which shows the available machine list (Table 1, step 8). The line contains the computer model name which is displayed next to a selection button on the Mosaic form. Lines beginning ## are also included in the HTML file and used to construct the Mosaic form. They are displayed after the machine name to provide additional information to help the user decide whether to select the machine. Lines beginning with # are not displayed, but can be used to store other comment information. This information is accessible if text results are requested, because the entire contents of the result data file is displayed in this case.

Bourne shell scripts and PERL programs are being developed to convert automatically the

results files generated by the benchmarks into the format required by the GBIS. For example, a PERL program has been written, which takes the Latex version of the NAS Parallel Benchmark Results [3] and creates the required directories and results data files for the GBIS.

2.4 Updating the Results Database to Include Additional Machines and Manufacturers

Updating the GBIS database involves three steps:

1. Adding the new results data file to the correct benchmark/manufacturer directory.
2. Running a shell script that updates the list of manufacturers available for each benchmark, if a new manufacturer subdirectory was created in Step 1.
3. Running a shell script to update the list of available machines available for each manufacturer.

The scripts mentioned in steps 2 and 3 produce the HTML files that are used in the creation of the Mosaic forms containing the manufacturer list for a given benchmark (Table 1, step 5) and the machine list for each manufacturer (Table 1, step 7). The script in step 2 above) uses the names of all subdirectories of a given benchmark directory (the subdirectories are named after manufacturers, see Section 2.2) and hides the HTML file created in the benchmark directory for later use. The script in step 3) works through each of the manufacturer subdirectories and hides an HTML file in each one. This file is constructed by extracting the lines which describe the machine benchmarked, from each results data file as described in Section 2.3.

The HTML files, used in constructing the Mosaic forms, are updated in this manner to improve the response time for the GBIS results. Initially, the Mosaic forms listing manufacturers, and then machines for selected manufacturers, where generated entirely at the time they were required. This gave a much slower response time.

2.5 Adding State Information to WWW Pages

A problem that had to be overcome, while implementing the GBIS, was how to retain information processed from Mosaic forms for later use. An example of the problem can be described by referring to Table 1.

If the user selects machines from the machine list WWW page (Table 1, step 8) and accepts default graph options, this is straightforward to process because all the information contained in the Mosaic form is used immediately. If, however, machines are selected from the machine list WWW page and the user opts to change default graph options, the following problem occurs. The WWW page of Table 1 (step 11) must be displayed so that the user can change default graph options. At the same time, however, a means of retaining the machines selected in Table 1, step 8, is required. The machine names can then be processed with the chosen graph options to produce the appropriate output.

The method used by the GBIS to overcome this problem is to store the information required for later use in temporary files, and then to encode this file name in the HTML used to create a subsequent form. When the information in this form is processed the name of the file is also extracted.

Table 1 (step 9) shows that the names of results data files, corresponding to selected machines, are stored in a temporary file. If the user elects to change default options another Mosaic form is displayed in Figure 1, step 11. Part of this Mosaic form as it is displayed on a WWW page is shown in Figure 7. The HTML used to generate this Mosaic form is shown in Figure 8. Figure 7 shows a radio button used to select the required output format. One of the four options (GIF, XBM, postscript, or text) must be selected.

In Figure 8 the four lines beginning '<INPUT TYPE="radio" NAME="FORMAT" VALUE=' are used to create the radio button. To find out which option was selected the Bourne Shell script, which decodes this Mosaic form, checks the value of an environment variable named $QS_FORMAT which contains the contents of the "VALUE=" attribute for the selected radio button. As well as containing the selected output format, this environment variable also has the name of the temporary file encoded in it. Therefore, the environment variable $QS_FORMAT can be tested to see if it contains the string GIF, XBM,

CHANGE DEFAULT OPTIONS

Select the required output format:-
♦ Display graph as .gif image (default)
♦ Display graph as .xbm image
♦ Display graph as postscript using external viewer ♦Colour postscript ♦Monochrome
♦ Display tabular results

FIGURE 7   Part of the GBIS WWW page that allows default graph options to be changed.

PSCOLOUR, or TEXT. This establishes the required output format. This string can then be removed from the environment variable, to leave the name of the temporary file. In Figure 8 the text file name was previously stored in an environment variable named $PIDFILE. This is appended to the "VALUE=" attribute.

3 AVAILABILITY

The GBIS homepage is available on the WWW, as follows:

URL: **http://hpcc.soton.ac.uk/RandD/gbis/ papiani-new-gbis-top.html**

Results data files can also be obtained by anonymous ftp, if the user does not have access to a WWW viewer. This is accomplished as follows:

type,
    ftp par.soton.ac.uk ⟨press enter⟩
at the name: prompt type,
    anonymous ⟨press enter⟩
at the password: prompt type your full e-mail address followed by enter,
then type,
    cd pub/benchmark_results ⟨press enter⟩

This takes you to the top level of the results subdirectories, which is described in the structure described in Section 2.2.

Results data files can also be sent in to the GBIS by anonymous ftp;

type;
    ftp par.soton.ac.uk ⟨press enter⟩
at the name: prompt type,
    anonymous ⟨press enter⟩
at the password: prompt type your full e-mail address followed by enter,
then type,
    cd incoming/benchmark_results ⟨press enter⟩
finally type,
    put result_file_name ⟨press enter⟩
where, result_file_name is the filename of your results file.

A readme file in this directory contains more details, including a suggested convention for the filename of your results file. This readme file is also available via the 'Instructions' link on the GBIS home page. It also contains further information on the implementation of the GBIS.
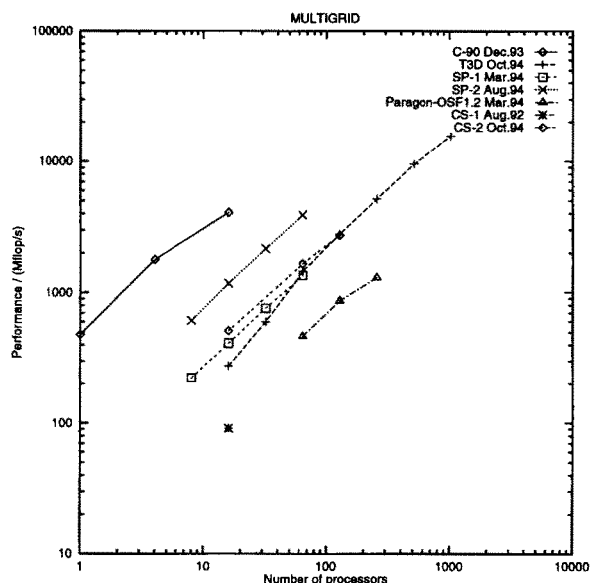
```
<pre>
<h2>CHANGE DEFAULT OPTIONS</h2>
<strong>Select the required output format:-</strong>

<INPUT TYPE="radio" NAME="FORMAT" VALUE="GIF$PIDFILE" CHECKED >Display
graph as .gif image (default)

<INPUT TYPE="radio" NAME="FORMAT" VALUE="XBM$PIDFILE" >Display graph as
.xbm image

<INPUT TYPE="radio" NAME="FORMAT" VALUE="PS$PIDFILE" >Display graph as
postscript using external viewer' '<INPUT TYPE="radio" NAME="PSCOLOUR"
VALUE="color" CHECKED >Colour postscript' '<INPUT TYPE="radio"
NAME="PSCOLOUR" VALUE="monochrome">Monochrome

<INPUT TYPE="radio" NAME="FORMAT" VALUE="TEXT$PIDFILE">Display tabular
results
</pre>
```

FIGURE 8   Part of the HTML document used to generate the GBIS WWW page that allows default graph options to be changed.

FIGURE 9 Graph showing results of the PARKBENCH Multigrid Benchmark (from NAS [3]), generated using the monochrome postscript output option of the GBIS.
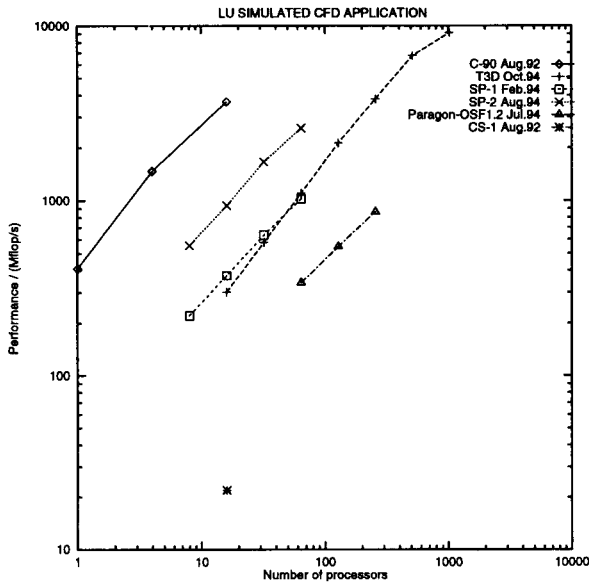
FIGURE 10 Graph showing results of the PARKBENCH LU Simulated CFD Application Benchmark (from NAS [3]), generated using the monochrome postscript output option of the GBIS.

4 EXAMPLE GRAPHS GENERATED BY GBIS

See Figures 9 and 10 for graphs showing results of the PARKBENCH Multigrid Benchmark and the

PARKBENCH LU Simulated CFD Application Benchmark, respectively.

REFERENCES

[1] C. A. Addison, V. S. Getov, A. J. G. Hey, R. W. Hockney, and I. C. Wolton, "The GENESIS distributed-memory benchmarks," In *Computer Benchmarks*, J. J. Dongarra and W. Gentzsch, Eds. North Holland, 1993, pp. 257–271.

[2] D. Bailey, E. Barszcz, J. Barton, D. Browning, R. Carter, L. Dagum, R. Fatoohi, S. Fineberg, P. Frederickson, T. Lasinski, R. Schreiber, H. Simon, V. Venkatakrishnan, and S. Weeratunga, "The NAS parallel benchmarks," NASA Ames Research Center, Moffett Field, CA, Tech. Rep. RNR-94-007, March 1994.

[3] D. H. Bailey, E. Barszcz, L. Dagum, and H. D. Simon, "NAS parallel benchmark results 10-94," NASA Ames Research Canter, Moffett Field, CA, Tech. Rep. NAS-94-001, October 1994.

[4] R. W. Hockney, "A framework for benchmark performance analysis," *Supercomputer 48*, vol. IX-2, pp. 9–22, Mar. 1992.

[5] R. W. Hockney and M. Berry, "Public international benchmarks for parallel computers, PARKBENCH committee: Report number 1," *Sci. Prog.*, vol. 3, pp. 101–146, 1994.

[6] R. W. Hockney and C. R. Jesshope, *Parallel Computers 2*. Bristol and Philadelphia: Adam-Hilger/IOP Publishing, 1988.