

Della-

Copies To: N. Tinklepaugh
B. Andrews
M. James (LW)
C. Robinson (LW) (k)
W. Hermeringer

from - Jim Masten

1-of-3

UNIX Profiling Tool for the IPSC/2

The Concurrent Workbench™ software tool set includes a version of the standard UNIX profiling tool adapted to run on several nodes concurrently. The profiler gives FORTRAN and C users an easy way to evaluate their applications for load balancing and efficiency, and determine which parts of the node programs might be improved to yield the greatest overall performance.

What the Profiler will Do

The profiler tools will give the programmer statistics on number of calls to individual subroutines, and estimates for time spent in each routine. Execution profiles are created for each node and stored in separate files for each node. The tools are easy to use - simply call the compiler with the profiling option switch ("-p") enabled. After the program run, the *prof* tool lets the user browse through the collected statistics.

Profiler components

The profiler tools consist of run time routines linked into applications compiled with the -p option, and a utility, *prof*, that displays the results. The run time libraries make use of the file system to store files of statistics for each node program under the family of file names of the form "mon.out.nnn" where "nnn" is the node id number. *Prof* then access each file as directed by the user.

Typical usage

To use the profiler, a user takes the following steps:

1. The node program is compiled and linked with the -p option, for example:

```
cc -p -o prog prog.c lprof -node
```

compiles the C program "prog.c", links it to the library "lprof" and produces an object module that runs on nodes.

2. The node application is loaded and run on the IPSC/2. At the end of the run, a set of files labeled "mon.out.nnn" are created on the disk system, one for each node.
3. The user then invokes the *prof* utility to look at the execution statistics, which come in the form of tables with the format of the example shown below.

Example output

The following is an example of a table of execution statistics printed by the *prof* tool for a FORTRAN Linpack application that ran on a node for about 271 seconds. *Prof* will sort the table in order of most time, or most calls as the user wishes. Here, the routine "daxpy_" was called 115,319 times, or about 95% of the total execution time.

%Time	Seconds	Cumsecs	#Calls	msec/call	Name
95.1	257.95	257.95	115319	2.2368	daxpy_
2.7	7.25	265.20	6	1208.	matgen_
1.2	3.30	268.50	6	550.	dgefa_
0.6	1.50	270.00	1044	1.437	idamax_
0.3	0.75	270.75	1044	0.718	dscal_
0.2	0.45	271.20			_csendrecv
0.0	0.05	271.25			_mcount
0.0	0.05	271.30	6	8.	dgsel_
0.0	0.05	271.35			monitor
0.0	0.00	271.35	24	0.0	second_
0.0	0.00	271.35	1	0.	MAIN_

Restrictions and Limitations

- Like the original UNIX tool, this profiler tracks CPU usage and subroutine calls, not message traffic.
- Programs to be profiled are limited to 600 subroutines or labels.
- Only one process can be profiled on each node.
- Analysis of multiple nodes assumes the same node program is run on all nodes. Heterogeneous node programs can be analyzed with separate runs of the *prof* tool calling out different program's link tables.
- Applications using the profiler cannot use the *sleep* or *alarm* functions on the node.
- Execution time estimates are based on 50 msec time slices, which means that accurate time estimates typically require run times of 25 to 50 minutes.
- CFS files should be used for the "mon.out.nnn" files where possible, in order to improve turn-around time.
- *Prof* views one node's execution profile at a time. However, users can send *prof* output to another set of files and then use the UNIX utility *awk* to produce reports on data from multiple nodes.

UNIX is a trademark of AT&T. Concurrent Workbench and iPSC/2 are trademarks of Intel Corporation.

UNIX Profiler Utility

Gathers execution statistics on node programs
- numbers of calls - time spent in each routine

Use as compiler option, then view data with *prof* utility.

%Time	Seconds	Cumsecs	#Calls	msec/call	Name
95.1	257.95	257.95	115319	2.2368	daxpy_
2.7	7.25	265.20	6	1208.	matgen_
1.2	3.30	268.50	6	550.	dgefa_
0.6	1.50	270.00	1044	1.437	idamax_
0.3	0.75	270.75	1044	0.718	dscal_
0.2	0.45	271.20			_csendrecv
0.0	0.05	271.25			_mcount
0.0	0.05	271.30	6	8.	dgsel_
0.0	0.05	271.35			monitor
0.0	0.00	271.35	24	0.0	second_
0.0	0.00	271.35	1	0.	MAIN_