JET PROPULSION LABORATORY

INTEROFFICE MEMORANDUM
SFB: 366-91-5
August 15, 1991

TO: Time Warp Group

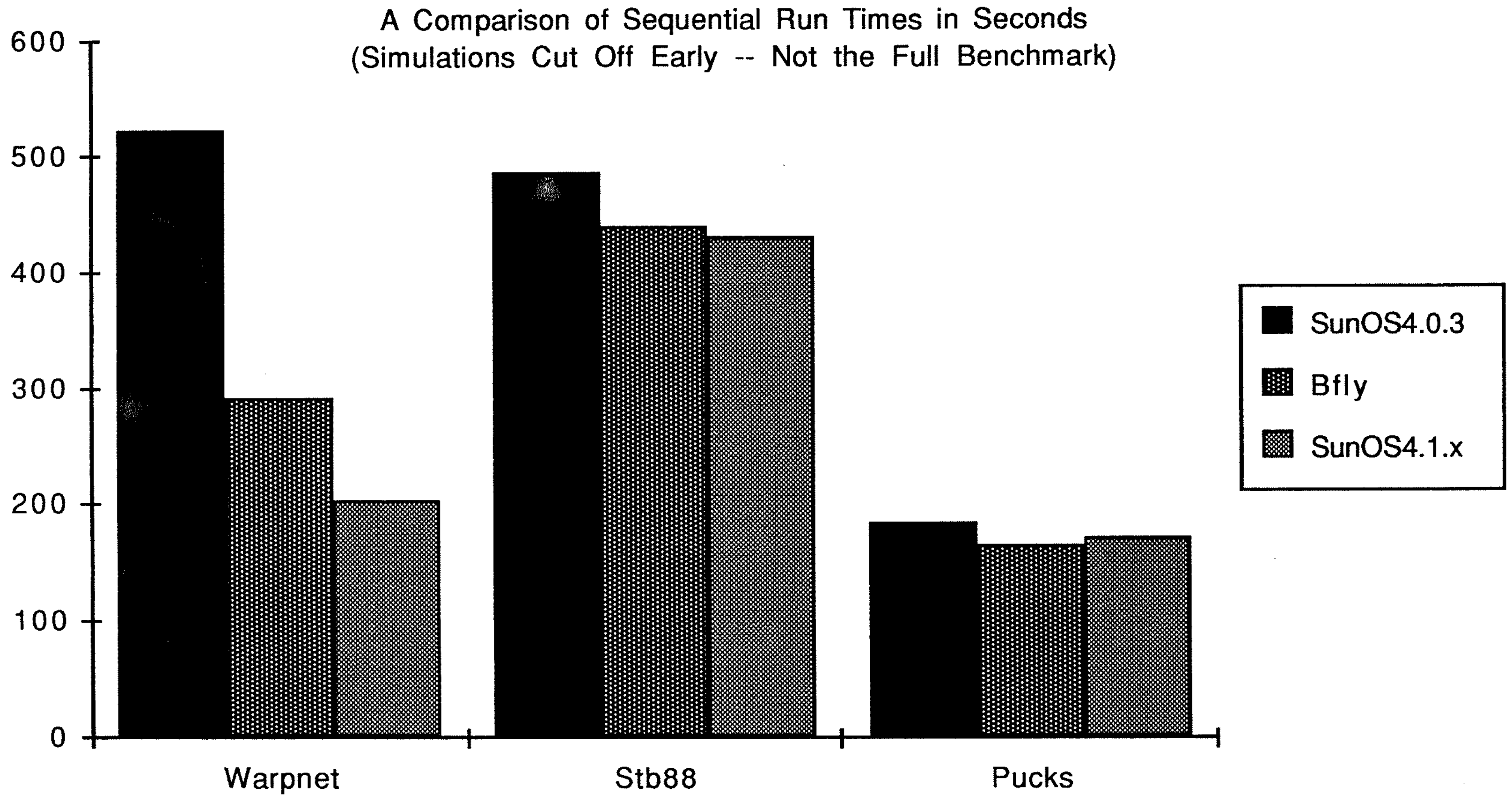
FROM: Steve Bellenot

SUBJECT: Sun TWOS 2.5.1 BenchMinus

If you look at the title carefully, you will see that this is not a benchmark memo. First the application benchmarks were given arbitrary cutoffs to make their run times faster. (For warpnet we used a cutoff of 250, pucks had a cutoff of 40 and stb88 had a cutoff of 25000.) Also the Sun TWOS code executed isn't exactly the TWOS 2.5.1 code. This code has a few additional bug fixes and prints a longer GVT line. However, we believe its performance is very close to the Sun TWOS 2.5.1 code. Configuration files were stolen from prior Butterfly benchmarks and not made special for the Sun. Both Sun and Butterfly run times are with dynamic load management turned off. All Sun runs were made on the Sun 3/60's which have 12 megs of memory each. There are only four 3/60's locally available and we always used all of them in our multiple node runs.

Both the Sun and the Butterfly are 68020 based computers, but have different compilers. Figure one shows the run times for the sequential simulator runs from the Butterfly and for two different Sun compilers. It is possible that the all the difference between the Sun 4.0.3 compiler and the Sun 4.1.x compiler is that the "-O" option (optimize) now defaults to "-O2" instead of "-O1." It seems reasonable to infer that all the difference in run times in Figure one could be compiler differences.

The four node Sun runs were made by a batch file which used the new "xtw" tool. Xtw starts up an "xterm" as well as Time Warp on each of the Sun nodes. The combination of the xterm and Time Warp requires more than the four megs of memory that our Sun 3/50's have. The Sun Time Warp version was run with 3 megs of memory in the Time Warp heap. The Butterfly Time Warp version was run with 2.125 megs of memory in the Time Warp heap. (The Sun version needs a much bigger stack.) Figure two shows the relative four node performance between the Butterfly and three different Sun runs. The difference in the Sun runs is



A Comparison of Sequential Run Times in Seconds
(Simulations Cut Off Early -- Not the Full Benchmark)

Figure 1. Relative Run Times.

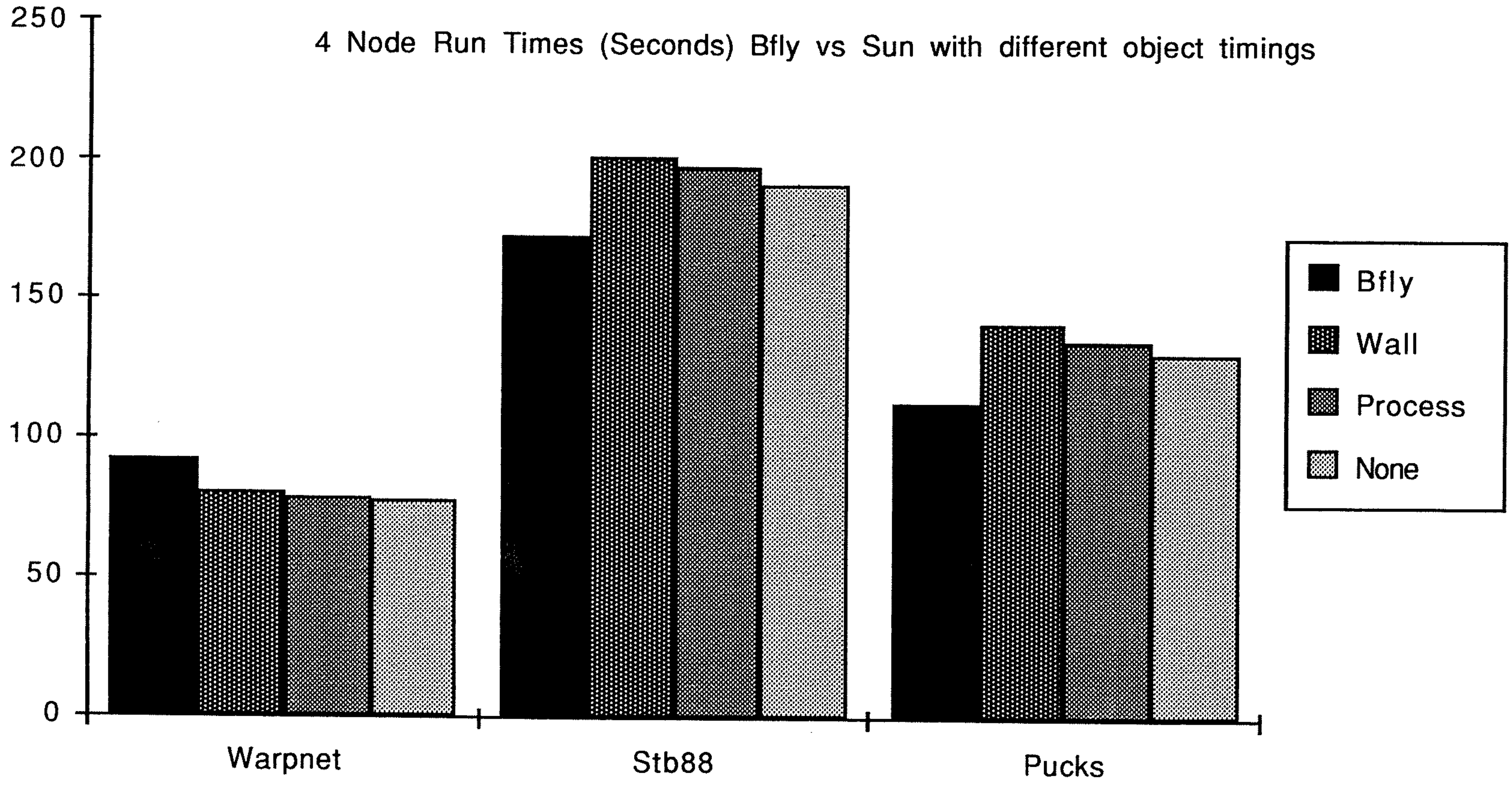


Figure 2. Four Node Run Times

how objects are timed. In "Wall", the wall clock time is used to measure how long objects execute. In "Process", only the time which the program is actually executing is measured. And in "None", no object timings are done. The Butterfly version times objects on a "Wall" basis. The Sun simulator times objects on a "Process" basis. The old Sun versions use to time objects on a "None" basis. Roughly there is a 2-3% difference in run time going from none (fastest) to process and from process to wall (slowest). The reason for the speed difference is due to the number of system calls needed. (Butterfly clocks reads do not require system calls.)

Figure three shows the speedups for these four node runs. Note that the Butterfly always "wins" the speedup race. The timings for the Sun version were taken from a batch file made in the night on Suns which were doing nothing else. Figure four shows all the run times of the "Wall" four node warpnet run versus time of day. The run actually started around 6pm and run to about 4pm the next day. The big gap between 5am and 11am was due to a run which was stuck. During the day a second run stuck but was found faster. Note that even at 3am, run times can suddenly jump. Busy screen savers like "lockscreen" can seriously degrade timings. If one machine doesn't have enough memory for both the xterm and Time Warp, then the batch program will hang.

The socket based communication on the Suns is much slower than the Butterfly communications. We used Ping sending 10000 messages to estimate communication time. On the Butterfly, the one node run time of 42.6 seconds and two node run time of 42.5 seconds indicates that the communication time is less than overhead of scheduling (setting the current object at +inf and rearranging the ready queue.) However, the one node Sun times of 40.5 seconds (wall), 32.4 seconds (process) and 22.7 seconds (none) is about the same as the Butterfly, but the two node run time was 1003 seconds (same run time for wall, process and none) over 20 times longer. Thus as a rough estimate it took an average of 100 milliseconds per message (or 10 messages per second) on the Suns.

We think the reason the slowness of the Sun communications isn't a major factor in our benchmark runs is because these benchmarks are "saturated" at four nodes. (That is, there is always lots of useful simulation work to do on each node.) Since all communication between the Suns is on the same ethernet, increasing the number of Suns will eventually saturate the ethernet bandwidth, slowing communications further.

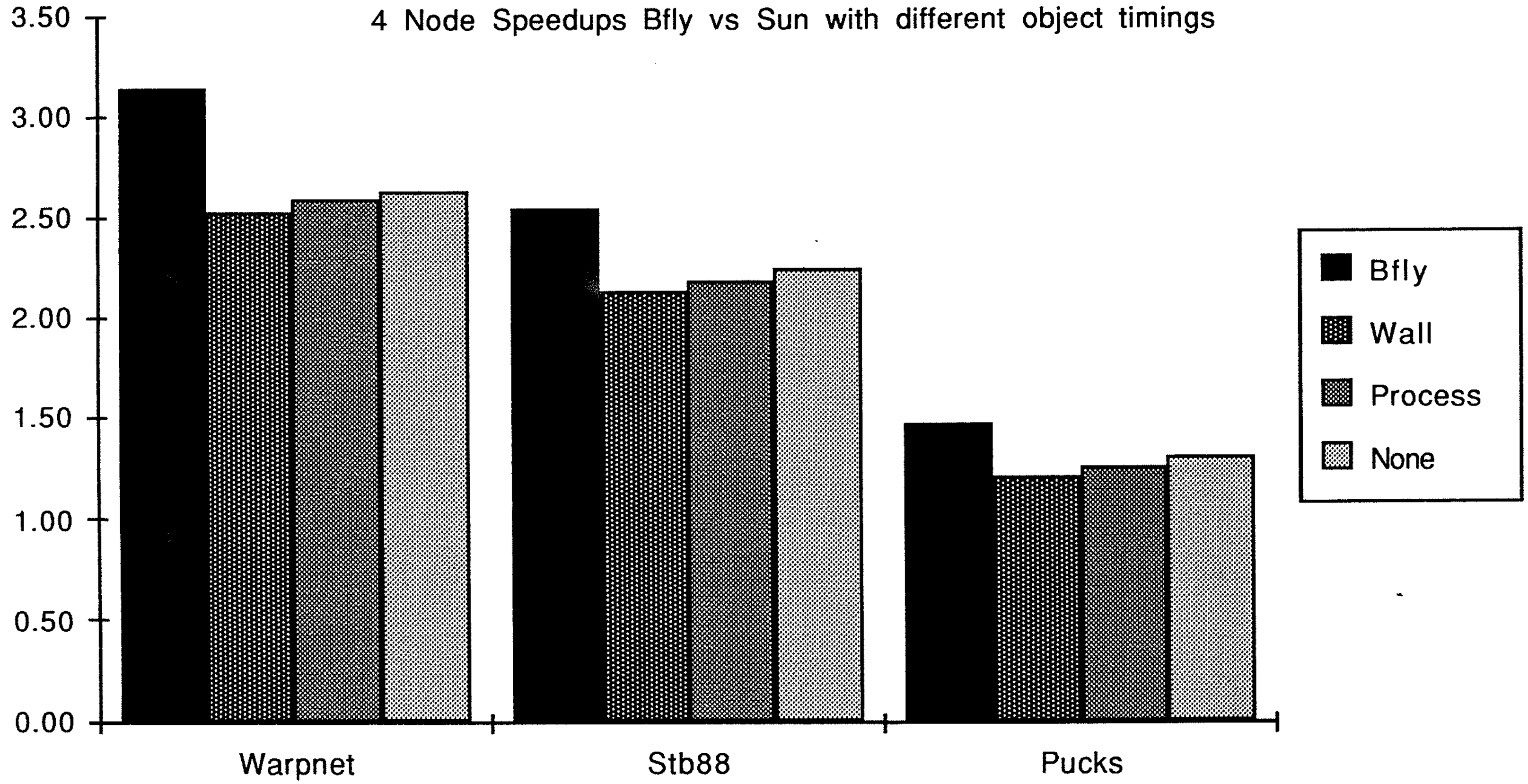


Figure 3. Four Node Speedups

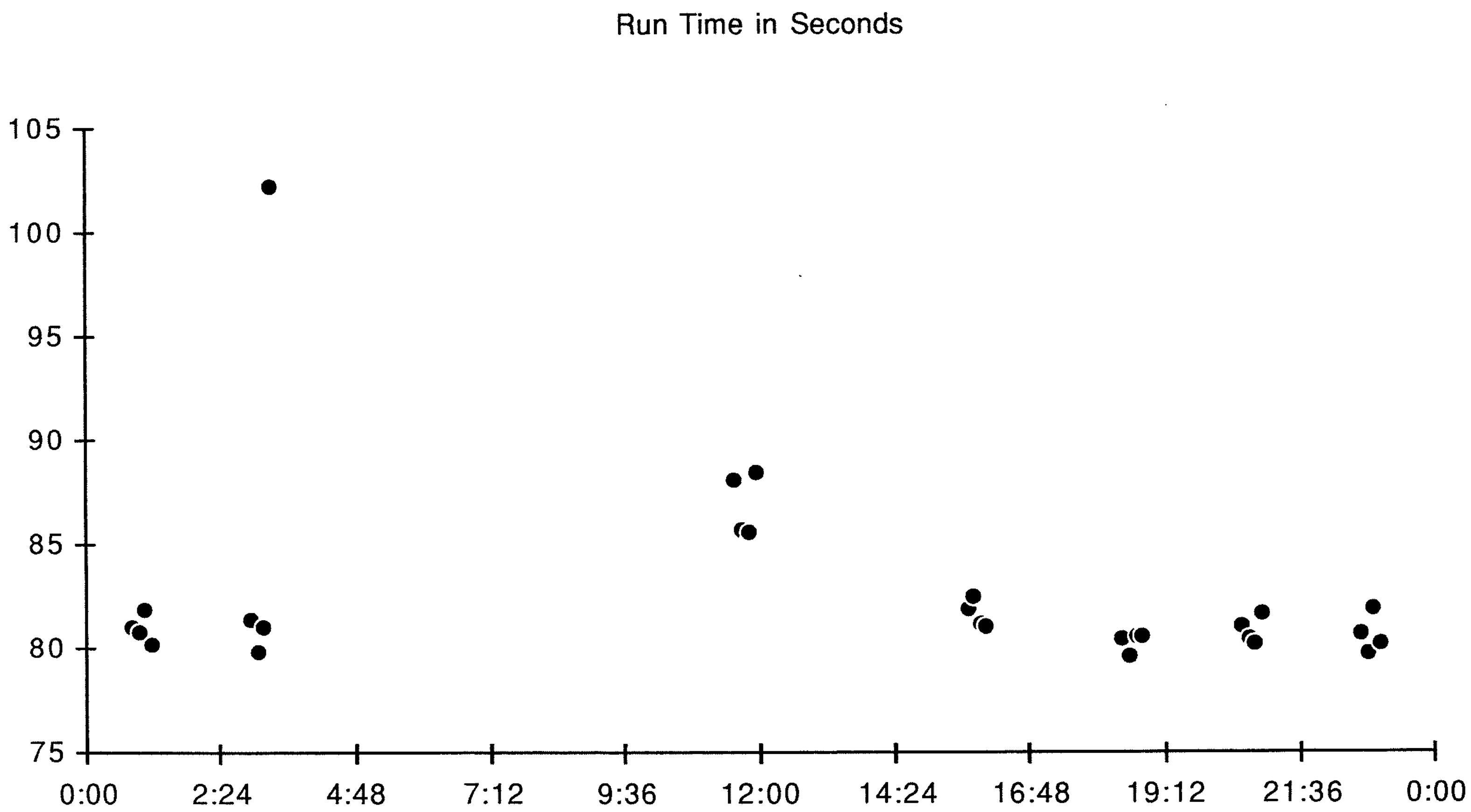


Figure 4. Variations in Run Time vs Time of Day