# A Performance Comparison of the Window Systems of Two LISP Machines

Ramin Zabih
Massachusetts Institute o f Technology
Department of EE & CS
Cambridge, MA 02139

Raj Jain
Digital Equipment Corp.
77 Reed Road (HLO2-3/N03)
Hudson, MA 01749

## EXTENDED ABSTRACT

We have constructed a workload for comparing window system performance of different computers. It consists of key operations that window systems should universally provide. We have used this workload to measure the performance of a the Symbolics ZetaLisp window system on two Lisp Machines, the 3600 [3,4] and the CADR [1].

The workload consists of repeating a sequence of operations (which we call a run) many times. Each run starts by creating a test window, and proceeds to perform various operations on the window in the order below. Finally, the test window is deleted. The details of the operations performed in an individual run of the workload follow.

**Creation:** Create a generic window, of full screen size, capable of doing graphic operations such as the plotting and drawing of lines as well as bitblt.

**Exposure:** Make the test window visible.

**Select:** Make the test window be the "selected" window.

**Resizing:** Shrink the window to half-height, and then grow it back to its normal size twice.

**Point Drawing:** Draw one hundred random points in the window.

**Line Drawing:** Draw one hundred lines by picking two hundred random points and connecting every other pair of points.

**Bitblt:** Create a 320 by 320 array of random bits, and display using the bitblt primitives, in a random location on the window. Perform this operation ten times.

**Character Output:** Output five hundred random ascii (non-control) characters to the window.

**Deletion:** Delete the test window.

Our workload repeated several operations many times. For instance, we draw 100 points and output 500 ascii characters. These numbers were chosen to make the total measured time significant, keeping in mind the clock resolution (10 milliseconds).



We chose total elapsed time as our performance metric. Elapsed time is also referred to as response time. We ran 1000 trials of our workload on a 3600 running Release 5 software with 1 Megabyte of physical memory and on a CADR running release 4.5 software with 386 Kbytes of memory. The results of our tests are reported in Table 1. Using the $P^2$ algorithm [2],we also calculated the fifth and ninety-fifth percentiles for the results. In addition, we have listed the median and the mean.

**Table 1:** Time in milliseconds is shown as $t_1/t_2$,.where $t_1$ is time for 3600, $t_2$ is the time for CADR.

| Operation | 95% | 5% | Median | Mean |
|---|---|---|---|---|
| CREATE | 60/70 | 10/40 | 20/50 | 21.8/48.5 |
| EXPOSE | 230/250 | 180/190 | 190/190 | 198.3/212.7 |
| SELECT | 10/10 | 0/0 | 00/10 | 0.9/7.9 |
| RESIZE | 460/730 | 410/660 | 420/710 | 422.8/700.6 |
| DRAW-LINES | 890/210 | 750/160 | 820/190 | 816.0/190.1 |
| DRAW-POINTS | 20/70 | 00/30 | 20/50 | 14.6/54.2 |
| BITBLT | 320/510 | 140/200 | 220/340 | 225.4/345.4 |
| CHAR-OUTPUT | 180/360 | 130/300 | 160/330 | 154.6/330.9 |
| KILL | 190/260 | 140/190 | 150/200 | 157.2/218.0 |

As is clear from the results, the 3600 window system is quite fast. The time to output 500 characters is almost always below a fifth of a second. Similarly, the time to do graphics output is very small. Even the time for window creation is almost nothing. Even the longest time for line drawing was below a single second. Thus, it seems that the 3600 window system is in fact very fast.

The CADR is generally about 50% slower, with a fairly wide variance. This is, however, still reasonably fast. The major surprise was that the CADR is significantly faster than the 3600 for doing line-drawing. We suspect that this is due to the presence of special microcode.

The workload we present should provide a basis for evaluating the performance of window systems on a wide variety of computer systems.

## REFERENCES

1. Greenblatt, R.D., T.F. Knight, et. al., "The Lisp Machine," In *Interactive Programming Environments*, edited by Barstow, Shrobe and Sandewall. 1984.
2. Jain, R.K., and *I.* Chlamtac, "*The P²* Algorithm for Dynamic Calculation of Quantiles and Histograms Without Storing Observations," To appear in *Communications of ACM*, October 1985.
3. Rea, D.P., "The Symbolics LISP Machine," *IEE Colloquium on Advanced Workstations for Scientific and Office Use* (Digest No. 80), October 1984, pp. 4/1-5.
4. Symbolics, Inc., *3600 Technical Summary*, February 1983.