645 vs 6180

PROJECT MAC

May 17, 1973

Computer Systems Research Division

Request for Comments No. 20

645/6180 PERFORMANCE COMPARISON TESTS

by R.A. Roach and J.H. Saltzer

I Description of Tests

A script was prepared for the tests consisting of two parts. The first part was a functional test of the system including tests of most of the commands found in revision 12 of the MPM. The second part was a test of the compilers and the binder. The script consisted of approximately 221 lines (often there was more than one command per line, especially in the first part) and was read via a special absentee dim which inserted an automatic delay between lines consisting of a random time between two and fifteen seconds. The output from this script was an absout file approximately 1900 lines long.

The tests were performed on a "small" configuration on each machine (see part II below). For the test, a master (interactive) process was logged in during a special session. This master process submitted twenty absentee requests and waited until they were logged in. When all the processes were logged in, a set of system meters was initialized and the scripts started on each of the absentee processes. As each script completed, it signalled the master process which gathered statistics on the run. In the 645 run, one of the processes terminated due to unexplained reasons approximately 80% through the script and for that run, the results had to be extrapolated.

II Configurations

	645	6180
System:	18-11	20.2
Time:	05/12/73 0630 - 1040	05/10/73 0617 - 0750
CPU:	A	A
Memories:	C & D (256K)	A & C (256K)
Interlace:	off	off
Paging Device:	Drum B (2031 pages)	BSU A (2039 pages)
Disks:	12 D270's 14 D170's	6 D190's
Disk Channels:	2	1

III Results

	645	6180
Elapsed Time:	249.7 min	91.4 min
Costs (shift 3):	$598.40	$524.36
Costs (shift 1):	$1,995.07	$685.66
Costs (shift 2):	$1,421.30	$619.25
CPU, total (1):	14,213 sec	4,860 sec
CPU, virtual (2):	9,002 sec	3,985 sec
CPU, good (3):	7,999 sec	4,031 sec
Memory units:	-	25,786
Page Faults:	919,474	339,883
ttm meters:		
Page Faults:	20.35%	14.40%
Drum Interrupts:	8.18%	-
Getwork:	9.75%	2.33%
Seg Faults:	4.61%	1.18%
Bound Faults:	0.16%	0.03%
Interrupts:	0.43%	0.86%
Idle, zero:	2.04%	9.64%
Idle, loading:	0.02%	0.00%
Idle, MP:	0.42%	0.00%
Idle, NMP:	0.71%	0.16%
Idle, total:	3.19%	9.80%
Other:	53.33%	71.47%

Notes:

(1) 6180 value based upon VCPU/TCPU = .82

(2) 645 value based upon VCPU/TCPU = .60

(3) based on ttm data (elapsed time x "other") rather than actual run data as in the other CPU times.

IV Analysis and Interpretation

From the results, the two measurements seem to be comparable, with one minor reservation: the 645 system, under normal load with this configuration, usually runs with time in the "other" category between 55 and 60%. The observed value of 53% may indicate that the test script forced the 645 system into a slight paging traffic overload. (At present, the multiprogramming control algorithm is disabled, so the system does not automatically respond to overload by reducing multiprogramming.) The maximum effect of this reservation is that the measurements understate the true 645 performance by about 5 parts in 53, or a little under 10%.

The total effect of the hardware speedup and the replacement of the firehose drum with the bulk store is most easily read out of the total cpu time for the two runs:

$$\frac{\text{645 total cpu time}}{\text{6180 total cpu time}} = \frac{14213}{4860} = 2.9 \qquad \left\{\begin{array}{c}\text{total effect}\\ \text{of}\\ \text{new system}\end{array}\right\}$$

As mentioned, this effect is the product of two smaller effects: 1) Use of a faster processor, and 2) replacement of the drum with the bulk store. There is enough information in the present measurements to separately estimate the two effects. The "good" cpu time represents the time to perform the work of the scripts, not counting paging. This work should be unaffected by replacement of the drum with the bulk store, so can be used to estimate the effect of the faster processor alone. We obtain:

$$\frac{\text{645 "good" cpu time}}{\text{6180 "good" cpu time}} = \frac{7999}{4031} = \underline{2.0} \qquad \left\{\begin{array}{l}\text{effect of system}\\ \text{changes \underline{other} than}\\ \text{the bulk store.}\end{array}\right\}$$

see RFC 19

as an estimate of the processor speedup. (Note that this effect actually includes four effects: the basic hardware speedup, the replacement of wall-crossing software with wall-crossing hardware, change to the file system locking strategy, and a compiler which uses eight pointer registers rather than four.)

The remaining part of the speedup, from the bulk store/firehose drum change, is deducible in any of several ways. Since the primary effects of the bulk store are to reduce multiprogramming idle and to reduce the level of multiprogramming (thereby giving each process more space to run and a lower paging rate), the most direct calculation is by comparing useful work done. On the 645, the "other" and "zero idle" categories indicate that 53% out of 98% of the elapsed time was useful. On the 6180, 71% out of 91% of the elapsed time was useful.
Thus we have:

$$\frac{\dfrac{\text{6180 "other" time}}{\text{6180 non-idle time}}}{\dfrac{\text{645 "other" time}}{\text{645 non-idle time}}} = \frac{\dfrac{71}{91}}{\dfrac{53}{98}} = 1.45 \quad \left\{\begin{array}{l}\text{effect of}\\ \text{bulk store}\\ \text{in reducing}\\ \text{paging overhead}\end{array}\right\}$$

As a confirmation, the product of the two separately accounted effects, (2.0 x 1.45) = 2.9, which agrees with the measurement of the total effect first mentioned. As a further confirmation, the effect of the processor speed alone, including wall-crossing speedup, agrees closely with that measured directly and reported by RFC 19.

Finally, we may note that the billed charges for first shift correspond to the performance changes. We have

$$\frac{\text{645 Cost (shift 1)}}{\text{6180 Cost (shift 1)}} = \frac{1995.07}{685.66} = 2.9 \quad \left\{\begin{array}{l}\text{ratio of}\\ \text{first shift}\\ \text{charges}\end{array}\right\}$$

This ratio suggests that the first-shift prices for cpu and memory have been accurately set to reflect exactly the performance improvement of the new system. In reviewing the second and third shift charges, it is apparent that the new prices have also been set to make shift differentials much smaller: the second shift user will see a price improvement of 2.3, and the third shift user a price improvement of only 1.1 The shift differentials now amount to about 10%, much smaller than before.