$\boxed{\mathcal{AIPS}\text{ Memo 104}}$

Y2K, a new DDT, and AIPSMark$^{(00)}$ Measurements

Eric W. Greisen

September 8, 2000

Abstract

The $\mathcal{AIPS}$ certification and benchmarking package known as DDT has lost its usefulness as computers have become faster. A similar package, called Y2K, has been constructed to use the $\mathcal{AIPS}$ imaging tasks now in greatest favor on a significantly larger problem. This should let us keep up with computer development for a few years. In this Memo I present the results of performance tests made with the 31DEC99 release of $\mathcal{AIPS}$ using both the old DDT and new Y2K on a variety of computer architectures. A new "$\mathcal{AIPS}$ Mark" is defined.

1 Introduction

The idea of an $\mathcal{AIPS}$ certification and benchmarking package was introduced by Wells *et al.* in 1985 [1] and was substantially updated by Greisen in 1994 [12]. It has been the basis of an additional thirteen $\mathcal{AIPS}$ Memos (see References). The certification test, called "DDT" for "dirty-dozen test," is run regularly on the development version of $\mathcal{AIPS}$ to insure its continued correctness. The package has been used in formal computer procurements and in a number of other tests with commercial or financial implications. This package has been described in detail in previous $\mathcal{AIPS}$ Memos and will not be discussed extensively here.

In the seven years since the previous revision of DDT, $\mathcal{AIPS}$ users have switched from the imaging tasks UVMAP and MX used in DDT to an object-based imaging task named IMAGR. Computers have also become a great deal faster. The IPX computer used to develop the 1993 version of the test required 4000 seconds to complete it. Recently, a Compaq Alpha Digital XP1000 running Red Hat Linux 6.2 with the DEC Fortran compiler ran this test in 68 seconds. Undoubtedly, much of this small time was occupied with overhead in running the test rather than in direct computation. I have created a new test which is an order of magnitude more demanding than the 1993 test. I chose to give it a new name, Y2K, so that we may retain the older, widely-used test for comparison.

2 The revisions to DDT

In Greisen 1994 [12], this section was an exemplar of de Sade's principle ("no good deed goes unpunished"). This time, fortunately, the long saga of bugs and algorithm improvements did not arise. The only abnormality discovered turned out to be a short-lived bug in the paging system in Red Hat Linux systems that affected artemis, a newly installed Pentium system at the AOC. Y2K is very much like DDT except that the UVMAP step was replaced by IMAGR with no deconvolution, the MX step with no deconvolution was dropped, and the MX step with deconvolution was replaced with IMAGR doing the imaging and deconvolution. The re-convolved image from VTESS is now created and checked as well as the pure MEM image.

The MEDIUM and LARGE DDT input data sets, together with their imaging parameters, were made the SMALL and MEDIUM data sets in Y2K. The new LARGE data set was donated by Chris Carilli and Rick Perley. Their data, a four-configuration X-band observation of Cygnus A, proved to be larger than we wanted. However, by discarding the A configuration data and time averaging (to 30 seconds) the remaining B, C, and D configurations, a suitable data set was obtained. The `LARGE` test uses compressed *uv* data while the two smaller ones continue to use uncompressed. The images are 2048 on a side; `APCLN` uses 25000 iterations to make a self-calibration model while `IMAGR` uses 75000 iterations. The parameters given to `VTESS` in all three tests were changed to obtain much better output images.

3 Benchmark Results

The matter of most concern to users is the question of how long it will take to perform the full sequence of jobs needed to reduce their data. To express this simply, Glendenning and Hunt (1991, [10]) invented the concept of "AIPSmarks." They define the total run time of the `DDT` as the real time between the procedure initiation ("`RUN DDTEXEC`") and the (nearly) final print message (`PRINTING ANSWERS, ERRORS, OTHER IMPORTANT MESSAGES`). This time is easily determined from the messages printed at that final print message. From the LARGE test, Greisen 1994 [12] defined an "AIPSMark$^{(93)}$" as

$$A_m^{(93)} \equiv \frac{4000}{T_{LARGE}} ,$$

where T_{LARGE} is the total run time for the `LARGE DDT` in seconds. The scaling factor (4000) was chosen to make the IPX (and Convex C-1) approximately 1.0 AIPSmark. For the new AIPSMark$^{(00)}$, the scaling factor is simply increased by a factor of 10:

$$A_m^{(00)} \equiv \frac{40000}{T_{Y2KL}} ,$$

where T_{Y2KL} is the total run time for the `LARGE Y2K` in seconds. The scaling factor (40000) was chosen to make the new AIPSmarks be on roughly the same scale as the old ones.

The computers tested are summarized in Table 1 and the test results are listed in Table 2. Total real times are affected by many factors. The "Server" column is a qualitative hint at the source of the `HELP` and executable files for the tested machine. "Slow" means they are both served by a normal computer in the local area network. "Fast" means they are served by a special server computer, while "Fast?" means that the `HELP` files are served by a normal computer while the executables are found on the tested computer via a symbolic link through the slow server. "Local" means that all files are found on the tested computer. Note that primate was moved from Charlottesville ("Slow") to Socorro ("Fast"), but its real times went up slightly. Perhaps this was due to an update of the Linux system installed on it after the move. There is no direct evidence that recent Linux updates have slowed performance, but the Solaris upgrade on tesuque clearly had this effect.

Another factor affecting performance is the amount of real memory in the machine and the amount of memory used for the "pseudo-array processor" in `IMAGR`, `APCLN`, `CALIB`, and `VTESS`. With a significant amount of real memory, the 5 and 20 Mbyte pseudo-AP give the same performance on all but the largest test. For images 2048 on a side, the larger AP memory is a distinct advantage. If the pseudo-AP uses up a moderate amount of the real memory, then operating systems that use left-over memory for disk I/O, especially Linux, may show poor performance. An 80-Mbyte AP on a 64 and even 128 Mbyte computer causes a serious degradation in performance due to paging and interference with disk I/O.

Disk systems are frequently tuned by default in the operating system to support many processes each accessing many small files. An *AIPS* environment, however, tends to have one or two very serious processes accessing a few large files. At NRAO, we have been adding a line to each Linux system's `/etc/rc.d/rc.local` file for EIDE disks that has something along the lines of:

```
/sbin/hdparm -c 1 -d 1 -k 1 -m 16 /dev/hda
```

Some of these are a little aggressive, especially the `-m 16`. Read the man pages for `hdparm` for your EIDE disks, some perform better with `-m 8` and it is wise to use caution when tuning your system. The effect of using or not using `hdparm` can come close to doubling the AIPSMark$^{(93)}$ you may get, and the performance boost is (qualitatively) very noticeable in non-benchmark use too.

There are also operating system tuning parameters applicable to Solaris systems. Adding, to systems at revision level 7 and below only, the following line to the file `/etc/system`

```
set priority_paging=1
```

gives priority to program data over file system data. Solaris 8 systems separate program and file memory pages and are thought to perform better without this line. In addition, the disk system controls disk writing with a system of high and low water marks. For all revision levels, adding to `/etc/system.`

```
set ufs:ufs_HW=6291456
set ufs:ufs_LW=4194304
```

changes the water marks to values more suitable to *AIPS*. The first change should not be done on systems with ≤ 64 Mbytes of memory or on Solaris 8 or later revisions. Experiment shows that the second change was important even at Solaris 8.

The ratio of AIPSMarks$^{(00)}$ to AIPSMarks$^{(93)}$ varies from 0.65 to 1.14 (leaving out the irreproducible 1.24 on dosequis[1]). In several cases, low values of this ratio are due to inadequate memory (computer or *AIPS*' pseudo-AP) or malfunctioning I/O systems. In other cases, such as primate and especially arcturus, the cause is less obvious. In the latter case, perhaps, all of the `DDT` fit in memory while the `Y2K(L)` forced concurrent disk operations. The DEC alpha, Pentium III XEON, and Celeron systems maintained or even improved their relative performance on the larger test. The limited cache of the Celeron chip appears to be unimportant in *AIPS* number crunching, making these computers rather more powerful than their low price would suggest.

References

[1]Wells, Donald C., Fickling, Gary A., and Cotton, William D., 1985, "Certification and Benchmarking of AIPS on the VAX-8600," AIPS Memo No. 36, NRAO, Charlottesville, Virginia, June 24.

[2]Hilldrup, Kerry C., Wells, Donald C., and Cotton, William D., 1985, "Certification and Benchmarking of AIPS on the CONVEX C-1 and Alliant FX/8," AIPS Memo No. 38, NRAO, Charlottesville, Virginia, December 24.

[3]Wells, Donald C., Fickling, Gary A., and Cotton, William D., 1986, "Benchmarking AIPS on a VAX-8600 with FPS-120B Array Processor," AIPS Memo No. 44, NRAO, Charlottesville, Virginia, April 19.

[4]Calabretta, Mark and Rayner, Paul, 1986, "Benchmarking AIPS on a VAX 8600," AIPS Memo No. 48, NRAO, Charlottesville, Virginia, September 22.

[5]Hilldrup, Kerry C., 1989, "AIPS Benchmarks on the CLSC and PSC Cray X-MPs," AIPS Memo No. 58, NRAO, Charlottesville, Virginia, January 25.

[6]Greisen, Eric W. and Calabretta, Mark, 1990, "Installing AIPS on an IBM RISC SYS6000 and Performance Results for Convex C220 and Sun Sparc Computers," AIPS Memo No. 65, NRAO, Charlottesville, Virginia, July 16.

[7]Calabretta, Mark and May, Henrietta, 1990,"AIPS DDT Benchmark Results for Sun's SPARCStation 2GX (Sun 4/75)," AIPS Memo No. 67, NRAO, Charlottesville, Virginia, November 28.

[8]Murphy, Patrick P., 1991,"A Comparison of DDT Results for IBM RS/6000 and Convex C-1," AIPS Memo No. 71, NRAO, Charlottesville, Virginia, April 8.

[1]The large `DDT` test ran significantly slower and the large `Y2K` test ran faster on dosequis at the beginning of the summer compared to the identical marconi measured at the end of the summer. Unfortunately, dosequis was not available at that time for a re-test.

Host	Site	Type	Mbytes	Server	Notes
Sparc/Solaris:					
ranger	AOC	SparcStation 4	64	Fast	
aguila	AOC	SparcStation 20	128	Fast	
tesuque	AOC	Ultra 10	128	Fast	Solaris 2.6
				Fast	Solaris 2.8
lemur	CV	Ultra 2: Dual 168	384	Slow	Disks optimized
comanche	AOC	Ultra 2: Dual 168	384	Fast	Disks before & after optimization
arcturus2	GB	Ultra-60	1024	Local	SOL load modules
Intel/Linux:					
valen	CV	450MHz Pentium III	256	Slow	Probable disk hardware problems
maruti	AOC	400MHz Pentium II	128	Fast	Behaved erratically
vegas	AOC	400MHz Pentium II	128	Fast	80-Mbyte AP modules
marconi	AOC	466MHz Celeron	128	Fast	Disks after optimization
dosequis	AOC	466MHz Celeron	128	Fast	Disks before & after optimization
primate	CV	500MHz Pentium III	256	Slow	
	AOC	500MHz Pentium III	256	Fast	Moved, upgraded OS
artemis	AOC	650MHz Pentium III	128	Fast	Memory slows I/O on largest jobs
?	Haystack	750MHz Pentium III	128	Local	Disks before & after optimization
vulcan	CV	550MHz PIII XEON	768	Fast?	
charybdis	CV	700MHz Pentium III	256	Slow	
Alpha/OSF1:					
hominid	CV	400 21164	512	Fast?	
Alpha/Linux:					
(alpha2)		667/XP1000	512	Local	With Compaq (DEC) compiler

Table 1: Computers tested for AIPSMark$^{(00)}$ Results

[9]Langston, Glen, Murphy, Patrick P., and Schlemmer, Dean, 1991,"AIPS DDT History," AIPS Memo No. 73, NRAO, Charlottesville, Virginia, May 16.

[10]Glendenning, Brian and Hunt, Gareth, 1991, "15APR91 DDT Results on a Sun IPC, a Sun Sparcstation 2, a IBM RS/6000 Model 550, and a Convex C1," AIPS Memo No. 75, NRAO, Charlottesville, Virginia, September 23.

[11]Allen, Ernest and Langston, Glen, 1992,"Summary of DDT Accuracy Results," AIPS Memo No. 77, NRAO, Charlottesville, Virginia, September 3.

[12]Greisen, Eric, 1994,"DDT Revised and AIPSMark$^{(93)}$ Measurements," AIPS Memo No. 85, NRAO, Charlottesville, Virginia, February 9.

[13]Murphy, Patrick P., 1995, "*AIPS* Benchmarks on the Sparc Ultra 1 and 2," NRAO, Charlottesville, Virginia, December 12.

[14]Kemball, Athol and Flatters, Chris, 1997, "*AIPS* Benchmarks for the Silicon Graphics Origin200," NRAO, Charlottesville, Virginia, January 29.

[15]Millner, Robert L., Murphy, Patrick P., and Uphoff, Jeffrey A., 1997, "*AIPS* on an ALPHA AXP Clone," NRAO, Charlottesville, Virginia, October 9.

Host	DDTL	AM93	Y2KS	Y2KM	Y2KL	AM00	AP	Notes
Sparc/Solaris:								
ranger	1976	2.0	536	2583	24342	1.6	5	
aguila	1580	2.5	356	1976	18705	2.1	5	
tesuque	422	9.5	102	492	5227	7.7	20	2.6; Compiled at 80 Mbyte
					5587	7.2	80	2.6
	575	7.0	147		5891	6.8	20	2.8; disks not optimal
	423	9.5	109	608	5505	7.3	20	2.8; high-water mark
lemur	411	9.7	147	539	4803	8.3	80	
comanche	538	7.4	147	614	4924	8.1	80	Before disk optimization
	388	10.3	118	490	4045	9.9	80	After disk optimization
arcturus2	166	24.1	53	204	2175	18.4	20	(SOL load modules)
Intel/Linux:								
valen	264	15.1	80	334	4234	9.4	20	Poor cpu/real, bad hardware?
maruti	310	12.9	98	383	4263	9.4	80	
	313	12.8	91	390	4158	9.6	20	
vegas	279	14.3	74	350	3968	10.1	80	
marconi	312	12.8	70	366	3167	12.6	20	After disk optimization
dosequis	527	7.6	157	614	6373	6.3	80	Before hdparm disk opt
	366	10.9	89	444	2963	13.5	80	After disk optimization
primate	222	18.0	64	305	3143	12.7	5	Small AP
	223	17.9	65	275	2724	14.7	20	Larger AP
	325	12.3	78				20	AOC: disks not optimized
	228	17.5	64	281	2739	14.6	20	AOC: hdparm done
artemis	194	20.6	56	243	2370	16.9	20	Y2KL I/O slowed by memory
(Haystack)	144	27.8			2208	18.1	5?	With hdparm on EIDE disk
	264	15.2						Without hdparm setting
vulcan	217	18.4	75	249	2175	18.4	5	
	208	19.2	65	248	1986	20.1	20	
charybdis	171	23.4	55	201	2010	19.9	5	
					1846	21.7	20	
Alpha/OSF1:								
hominid	300	13.3	132	441	2629	15.2	20	
Alpha/Linux:								
alpha2	68	58.8						With Compaq (DEC) compiler

Table 2: New `Y2K` AIPSMark$^{(00)}$ Results