AIPS MEMO NO. 36

Certification and Benchmarking of AIPS on the VAX-8600

Donald C. Wells, Gary A. Fickling, William D. Cotton

National Radio Astronomy Observatory [*]
Edgemont Road, Charlottesville, VA 22903-2475
(804)296-0211, FTS=938-1271, TWX=910-997-0174

24 June 1985

Abstract

The 15APR85 release of AIPS was installed on a VAX-8600 under VMS 4.1 and was tested using the "PFT" benchmarking and certification test. Installation was uneventful, and computed results agreed exactly with NRAO's VAX-11/780 in Charlottesville. Comparative timing data for the 8600 and 780 are tabulated. CPU-bound tasks executed up to 4.8x faster on the 8600 than on the 780. Tasks which had I/O or system calls which were not overlapped displayed lower ratios, some lower than 2x. The overall real-time ratio for the entire PFT test problem (dominated by compute-bound operations) was 3.8x. Data for the 780+AP and for VMS 4.0 versus VMS 3.7 are included to assist in interpretation of the measurements.

CONTENTS

[*] The National Radio Astronomy Observatory is operated by Associated Universities, Inc., under contract with the U. S. National Science Foundation.

1 INTRODUCTION

NRAO tested the new DEC VAX-8600 (often called the "Venus") because it is one of the candidate computers for a proposed procurement to replace the IBM and Modcomp systems in Charlottesville. The goals of the tests were to assure that the new machine would execute AIPS properly and to assess its performance relative to the DEC VAXes which NRAO already owns.

2 ABOUT THE CERTIFICATION AND BENCHMARKING PACKAGE

The programming group in Charlottesville has a standard procedure for assessing any new computer system: install NRAO's AIPS ("Astronomical Image Processing System") on it and run the AIPS certification and benchmarking test package. Successful execution of the test certifies that the computer hardware, the FORTRAN compiler, the operating system components, and the interface between AIPS and the operating system (the "Z-routines") all behave correctly. Benchmarking data can be extracted from the time stamps recorded in the AIPS message file and from the accounting listings produced by the AIPS utility PRTACC. Because the test procedures are written in the AIPS command language POPS ("People-Oriented Parsing System"), and because they read a binary data tape written in FITS ("Flexible Image Transport System") format, they are inherently machine- and operating system-independent. The intent is to perform exactly the same test on all machines being considered in a given procurement. Note that this does not mean that all of the code of the test is the same. Only the portable portions of the application programs, the POPS procedures, and the FITS files should be invariant; it is not only permissible but even desirable that the system-dependent portions of AIPS should be customized to achieve the best performance on each system.

The tests discussed in this memo were all performed with the pre-beta-test experimental version of the test package, which was named "PFT" (the production release will have a different name). The package is two command language scripts ("RUN files") of about 400 lines. The first script compiles the test procedures and the second executes them. The test executes twelve different AIPS programs (AIPS, IMLOD, UVLOD, UVSRT, UVMAP, COMB, APCLN, SUBIM, ASCAL, MX, CNVRT, and VM, often called the "Dirty Dozen") to process a real dataset of about 7000 fringe visibilities obtained with NRAO's Very Large Array. At each step where an image is computed, it is compared against the equivalent master image and residuals are printed. A full description of the details of the test package will be the subject of a future AIPS memo; it is sufficient to say that the package reasonably reflects the actual use of AIPS on real data. Some parts of it are I/O limited, some are CPU-bound, and some are sensitive to various sources of overhead such as opening, closing, and cataloging files.

AIPS is intended to be used with a Floating Point Systems array processor (models 100, 120B, 5105, or 5205). If the host machine does not have an AP the application tasks are linked against a library of

FORTRAN subroutines which have the same names, arguments, and functionality as the library of microcode routines for the FPS AP. This alternate library is called the "pseudo-AP". Ideally NRAO would have tested AIPS on an 8600 with an FPS AP; unfortunately, no such system was available, and so this memo only reports measurements of pseudo-AP performance.

The master data files for PFT were generated on 19 February 1985 on NRAO's Charlottesville VAX-780 using the FPS AP-120B array processor with the versions of the AIPS tasks which were extant at that time. The tests performed on the 8600 and 780 in pseudo-AP mode executed somewhat later versions of the tasks. In some cases changes in the algorithms produce minor residuals, but in any case, maps computed in pseudo-AP mode almost always give non-zero residuals compared to those computed with an AP, because the AP uses 28 bit floating point fractions while the VAX computes with 24 bit fractions. The important point is that the residuals were EXACTLY the same on the 8600 and 780. The reason for using the February version of the package rather than more recent versions is that the February version has been used to test a number of other machines recently and it was desirable to be able to intercompare all results.

3 TESTING THE 8600

Mr. Tom Chisholm, DEC's representative for central Virginia, accompanied two of the authors (DCW & GAF) to Massachusetts to make the tests, which were performed in DEC's Large Systems Center in Marlboro, MA, on 29 April. The center is a large computer room containing examples of all of DEC's large CPUs, both VAXes and DEC-20s, and their peripherals, and with arrangements that enable customers to test a wide variety of proposed system configurations. The LSC is a closed shop; the operator on duty mounts and dismounts tapes and returns line printer output to the guests, who work in a terminal room overlooking the computer room.

NRAO was given exclusive use of an 8600 for about five hours. The 8600 had 32 MB of memory and was running VMS V4.1. A single RA81 disk drive was used; it was interfaced through a Cluster Interconnect and HSC-50 controller, because this is the configuration which DEC recommends for new installations. Note that NRAO's 780s also have RA81 drives, but that they are interfaced with UDA-50 controllers on the Unibus. It is unclear whether using the CI and HSC-50 rather than the Unibus and UDA-50 had any significant effect on performance in the tests reported here.

The 15APR85 release of AIPS was installed on the 8600; this was exactly the same kit which was shipped to NRAO's user sites. Note that the executable images which were installed had been compiled on the Charlotttesville 780 using the 3.X compiler with the optimizer disabled, and had been linked under VMS 4.0; recompiling and relinking all of AIPS on the 8600 would have taken several hours. The main benchmark command procedure file was compiled, executed in "READ" mode to read in the FITS data tape, and then executed again in "TEST" mode to make the actual trial, which ran for about 90 minutes. At the end of the process the AIPS utility PRTACC was executed to produce a listing of the accounting files maintained by AIPS. The details of the timings are tabulated and discussed below.

There was still some time left at the completion of the formal test run. The 8600 had a copy of the new VMS 4.1 FORTRAN compiler, which had not yet been installed in Charlottesville (it was installed a week later). It was decided that a certification test of the large task MX compiled with the new compiler would be a good use of the time. NRAO has never before been able to use the VMS compiler with the optimizer enabled because several bugs were visible (and of obscure origin) and it was necessary to assume that other, less visible, and even more obscure, bugs must have existed.

The pseudo-AP library was recompiled with the optimizer enabled. Then MX itself was recompiled with the optimizer and was relinked; the main application libraries were not recompiled. Finally PFTEXEC was executed again with the adverb TMASK set to select the MX-map and MX-clean steps. It is a pleasure to report that the optimized MX and pseudo-AP still computed the exact same images as with the old compiler, and with 15% faster execution (see Table 1 below). This test suggests that the new optimizer is SIGNIFICANTLY more reliable than the old one for AIPS code. Presumably a 780 would display approximately the same speedup as the 8600 did.

Another astronomical image processing group began testing a VAX-785 in the LSC a short time after NRAO began its run on the 8600. This group was from the Image Processing Lab of the Jet Propulsion Laboratory; they were installing the current version of JPL's VICAR system. Considering that very few large-scale image processing packages exist, and that even fewer of such packages have been constructed by astronomy organizations, the simultaneous benchmarking of AIPS and VICAR at DEC's LSC was certainly a surprising coincidence!

The JPL systems programmer was having some trouble; he said that the 4.1 system was reluctant to install a shared library image which he had linked under 3.x. Such shared images have been proposed for use with AIPS on VMS systems; obviously, incompatibility of VMS releases should be factored into NRAO's deliberations.

4 COMPARISON DATA FOR THE 780

The Charlottesville 780 was reserved for the entire night of 06-07 May. A BACKUP tape which had been taken from the 8600 at the completion of the tests on 29 April was installed under a new login created for the purpose. A single RA-81 was the disk for this installation, analogous to the 8600 installation, but with a UDA-50 (Unibus) interface. Thus, the comparison run on the 780 was performed with exactly the same executable images and data files as on the 8600.

During the MX-clean step of the procedure the night operator did the normal daily disk BACKUP process, which ruined the timing data for that step. The step was rerun later and the total real time has been adjusted by the 958 second difference in real times between the two runs of that step.

One of the authors (WDC) ran PFT on 14 February 1985 on the 780 under VMS 3.7 with the array processor. This test was to generate the master comparison for trials which were performed on the NORD-500 computer in Sweden later that month. On 12 April PFT was executed again to test how the installation of VMS 4.0 on the Charlottesville VAX had affected the performance of AIPS. Results from these runs are also tabulated below.

5 TIMING DATA, SPEED RATIOS, CPU/REAL RATIOS

5.1 Real-Time Speed Ratios

For this memo we choose to emphasize real-time ratios (Table 1) because the 8600 tests did not involve an AP. For VAXes with APs the situation is more complicated than for the pseudo-AP: not only do AP tasks run faster in real-time, but the difference between real-time and CPU-time is available for use by other processes (that is, in some respects having an AP is like having another CPU). The 780 with pseudo-AP is taken to be 1.0 for comparisons to the 8600 and the 780+AP. The 780+AP under VMS 3.7 is ratioed to the 780+AP under 4.0, and the 8600 with optimizer is ratioed to the unoptimized 8600.

5.2 CPU/Real Ratios

The CPU/Real ratios in Table 2 are all just the CPU time in that table divided by the corresponding real time in Table 1 (note that when the AP is involved, the CPU time includes no contribution from the AP except for the handler). The ratios indicate the extent to which a task exhibits un-overlapped I/O or operating system overhead. Some tasks, MX with pseudo-AP is a good example, manage to almost completely hide heavy I/O activity behind their CPU operations, whereas others, VM and UVMAP for example, have rather low CPU/Real ratios. Some tasks are I/O dominated and exhibit quite low ratios; UVSRT and SUBIM are good examples. UVSRT does a disk merge sort and SUBIM is effectively a file copy operation.

Table 1: Real time [1], with speed ratios:

CPU:	780 [2]	8600 [2]		780+AP [3]		780w/AP		8600 [4]	
VMS:	4.0[5]	4.1w/HSC[5]		4.0		3.7		4.1w/HSC	
FOR:	3.5/NOOPT	3.5/NOOPT		3.5/NOOPT		3.5/NOOPT		4.0/OPT	
Date:	06 May	29 April		12 April		14 Feb.		29 April	
	Time	Time	Speed	Time	Speed	Time	Speed	Time	Speed
PFT-load	625::[6]	266	2.3::						
PFT-read	1060:[7]	437	2.4:						
UVLOD(3)[8]	128:	41	3.1:						
IMLOD(5)	125:	50	2.5:						
IM-CC(3)[9]	214:	72	3.0:						
PFT-test	20309[10]	5371	3.8	2306	8.8				
CNVRT(2)	15	7	2.1	16					
COMB (7)	222	91	2.4						
UVSRT(2)	78	53	1.5	82		74	1.11		
UVMAP$[11]	170	59	2.9	81	2.1	78	1.04		
APCLN$	4577	963	4.8	340	13.5	284	1.20		
SUBIM	14	13	1.1	15					
CLN-res$	113	35	3.2	60	1.9				
ASCAL$	3509	1088	3.2	154	22.8	153	1.01		
MX-map$	222	71	3.1	106	2.1	99	1.07	61	1.16
MX-clean$	10033	2340	4.3	712	14.1	683	1.04	2029	1.15
VM$	320	116	2.8	195	1.6	188	1.04		
total:				1670		1559	1.07		

Table 2: CPU times[1], with CPU/Real ratios:

	Time	Ratio	Time	Ratio	Time	Ratio	Time	Ratio	Time	Ratio
UVLOD(3)	70	.55	17	.41						
IMLOD(5)	41	.33	10	.19						
IM-CC(3)	175	.82	43	.59						
CNVRT(2)	11	.75	3	.39	10	.63				
COMB (7)	158	.71	34	.37						
UVSRT(2)	41	.53	10	.19	40		36	.49		
UVMAP$	151	.89	39	.65	41	.51	39	.50		
APCLN$	4485	.98	930	.97	144	.42	120	.42		
SUBIM	6	.41	2	.12	6	.40				
CLN-res$	91	.81	23	.65	32	.53				
ASCAL$	3397	.97	1057	.97	35	.23	32	.20		
MX-map$	191	.86	47	.66	47	.44	43	.43	38	.62
MX-clean$	9920	.99	2274	.97	228	.32	216	.32	1959	.97
VM$	280	.88	71	.61	126	.65	118	.63		

[1] times rounded to nearest second; some ratios from unrounded times
[2] compiled with 3.x FORTRAN compiler with optimizer disabled (NOOPT)
[3] using an FPS AP-120B with custom microcode and a Unibus interface
[4] compiled with 4.1 FORTRAN compiler with optimizer enabled
[5] 780 using RA-81 & UDA-50 (Unibus); 8600 using RA-81 & HSC-50 (CI)
[6] double colon measurement from a run on 25 February (light loading?)
[7] single colon measurement from a run on 27 April ("lightly loaded")
[8] times for "(n)" cases are sums of multiple similar runs
[9] IMLOD processing "clean component" tables (increased CPU load)
[10] time adjusted to correct for operator error during test
[11] tasks marked with "$" use the AP

5.3 AP Utilization

Some tasks execute less than 3x faster with an AP on a 780 than without. For several of these tasks (UVMAP, CLN-res, MX-map, and VM) the 8600 is actually able to beat the 780+AP combination. Some of these tasks have comparatively large overhead or unoverlapped I/O. VM displays a high CPU/real ratio (0.65), which indicates that much of its computing is done in the VAX even when the VAX has an AP. By comparison, MX displays a high degree of AP utilization (14x speedup) combined with a complete overlap of its I/O with its computing (CPU/Real of 99%).

ASCAL has the largest speedup (23x) of any AP task in the 780+AP; it also has the lowest CPU/Real ratio (0.23) of any AP task running in the 780+AP. It is obvious that ASCAL is peculiarly well adapted to the use of the AP. One suggested explanation is that ASCAL may be doing a disproportionate number of sine/cosine calculations, and that the 120B is especially strong on these operations due to special lookup tables and microcode.

5.4 Comparison Of VMS 3.7 With 4.0

Note that the ratio of real time for 3.7 and 4.0 on the 780+AP (1.07) suggests a significant performance degredation with 4.0. A possible explanation of this is the "file high water marking" introduced as a security measure with 4.0. This is enabled by default, and causes disk blocks to be erased as they are allocated. This certainly affected the timings of 4.0 on the Charlottesville VAX on 12 April, and presumably also the timings of the 8600. It has since been disabled on the NRAO VAXes, but the timing tests have not been repeated.

One further change with VMS 4.0 may affect the CPU timings, but not real time. This is the replacement of the disk ACP by the distributed XQP software. Work previously charged to the ACP is now charged to the process originating the IO requests.

5.5 The 8600/780 Performance Ratio

First we must ask: how typical is PFT of AIPS processing, ignoring timesharing? Obviously many real datasets are much larger than PFT's, and they result in much larger images; some images require many more CLEAN cycles. The relative emphasis on various tasks in such cases is likely to vary somewhat from that in PFT, but probably the relative ranking of operations in PFT is not grossly atypical, with the notable exception that PFT under-represents the role of VM, because PFT only does 3 iterations---full convergence would require more like 30 iterations. If the assertion that PFT is typical of the AIPS job mix is accepted, then it follows that the overall PFT-test real-time ratio of 3.8x ("PFT-test" in Table 1) is the single figure of merit that AIPS users would want to know.

Real AIPS machines usually have more than one task running and often support a general timesharing load as well. Is PFT a proper measure of these situations? There have been almost no measurements of real AIPS machines under heavy loading and therefore the strict, literal answer to the question must be:

> The present measurements apply only to AIPS running with no competition, and they even apply only to the particular sequence of AIPS operations and particular dataset that PFT uses. They say nothing at all about the performance with an AP on an 8600.

Note that the 8600/780 real-time ratios in Table 1 for the individual tasks of PFT range from about 1 to almost 5; this represents the spectrum of processes from I/O-bound (both 780 and 8600 using the same model of disk drives) to compute-bound. It would be reasonable to speculate that a general timesharing load would be less dominated by compute-bound processes than is PFT; in that case a ratio of about 3x might be a plausible guess.

5.6 Guessing The Performance Of An 8600+AP

It is reasonable to suppose that an 8600+AP would run faster than a 780+AP because the AP driver would execute faster and because the un-overlapped portions of the VAX execution would execute faster, but the disk drives would be the same and probably the bandwidth between the CPU and the AP would be about the same. All we can do is speculate; the current range of opinions in Charlottesville is that the combination would be 1-2x faster than the 780+AP in real time. CPU times with an AP would probably be 3-5x smaller than for the 780; the extra CPU time would be available for other processes.

6 CONCLUSIONS

NRAO's test of the VAX-8600 demonstrated that it computes radio images exactly as do existing VAX-780s (this is no surprise, of course). Compute bound tasks were up to 4.8x faster, approximately consistent with DEC's advertisements, while some I/O-bound tasks showed little gain in speed over the 780. The tests indicate that "average" performance under a synthesis mapping loading of an 8600 without an AP is 3.8x faster than for NRAO's VAX-780s without an AP. Note that these measurements on the 8600 do not say anything definite about how it would perform on AIPS tasks if it had an FPS AP.