CPG MEMO NO. 20

National Radio Astronomy Observatory
Socorro, New Mexico
Very Large Array Program

VLA Computer Memorandum No. 168

A COMPUTER PLAN FOR THE VLA

Bob Duquet, Gareth Hunt and Bob Burns

September 1983

ABSTRACT

This document contains an analysis of the computing needs of the VLA anticipated for the next three to seven years. It also contains recommendations concerning the equipment that will be needed to accommodate these needs.

This document examines the need for off-line facilities. In a separate document (Gareth Hunt and Ken Sowinski, VLA Computer Memorandum No. 166) a proposal for the upgrade of the VLA on-line system has been made. The proposed capital cost of this is $600k with no increase in the subsequent operating budget.

In order to be able to specify the computer capacity necessary for image formation and enhancement, a group headed by Ron Ekers has estimated the distribution of observing demand expected for the next several years. From their work, it seems clear that computer demands will greatly exceed the present VLA capacity.

No single number gives an adequate measure of computer capacity, especially when very different computer architectures are considered. In this plan a hypothetical machine with simple but well defined characteristics is used as a yardstick.

The effect upon this machine of an equally abstract task typical of the VLA processing load is analysed in some detail. To process 12 hours of observations made for this task, a VAX/array processor combination of the type now used at the NRAO for AIPS systems would need an initial 660 hours to set up the data i.e. for sorting then approximately 680 hours for image formation and enhancement. In other words, processing time would exceed observing time by about 100:1. The average astronomical requirements anticipated by Ekers et al. exceed this abstract task by about 100%.

When the VLA needs are calculated from the anticipated astronomical requirements, it appears that the increased use of computer intensive techniques now call for access to computer capacity of 60 to 75 Mflops. If we assume that this power would be available in a system with sufficient memory and I/O capacity to avoid serious bottlenecks, 60 to 75 Mflops is equivalent to about 50 VAX/array processor combinations or 1 large supercomputer (e.g. a Cray or a Cyber 205).

A possible alternative to a supercomputer might be found in the combination of a conventional machine with a new class of array processor that is now becoming available. More work needs to be done to compare prices and the performance of these alternative approaches.

The cost is expected to range from approximately $10 million if a small supercomputer is needed to approximately $7 million if the array processor approach is feasible.

Operating costs for the required facility will exceed current costs. The cost of utilities will remain about the same; the cost of

annual maintenance will increase from the present level of about \$200k to as much as \$700k; and an additional 3-6 positions in software and 1-5 for operations will be necessary depending upon the type of facility selected.

INTRODUCTION

This memo is a discussion of available options whereby the computing needs of the VLA may be satisfied during the next three to seven years. It has been prepared as part of a more comprehensive planning document covering all NRAO.

The scope of this plan does not encompass all computing done at the VLA. In particular, it excludes on-line and post-processing facilities.

Gareth Hunt and Ken Sowinski have elsewhere proposed an upgrade to the on-line system (VLA Computer Memorandum No. 166; a final draft of this was included as Computer Planning Group Memo No. 7). If this is approved a superior on-line computer system could be installed (optimistically) in 1985 using modern Modcomp computers. If a solution using another vendor's hardware is adopted a target date earlier than 1987 is unrealistic.

The term "post-processing" is used here in the sense of image analysis as opposed to image formation. To a great extent, post-processing has become identified with the use of AIPS software even though AIPS now encompasses both imaging tasks. This plan envisages continued reliance upon modest computer facilities available at observers' home institutions for a significant part of the total VLA post-processing. Regardless of the way in which other VLA computer needs will be satisfied, several hardware systems, typical of those on which AIPS is used outside of NRAO, undoubtedly will be maintained internally for software development and testing as well as for production use.

The role of the large VLA facility, which is the subject of this plan, is to provide high quality input (calibrated data and/or clean maps) upon which the local machines can operate without being overwhelmed by the initial mapping requirements.

The organization of this memo is as follows:

Part 1. As a point of reference we will begin by describing (briefly) the computer resources that are currently available at the VLA. In effect, this is merely an updated summary of the major document prepared last year (1982) for the Computer Advisory Group.

Part 2. The astromonical requirements of the VLA during the pertinent time period have been estimated by a group led by Ron Ekers. The results from that committee form the basis of some calculations of the future VLA computer needs.

Part 3. Next we describe, in broad terms, the options that might be available to meet those needs. Of necessity the discussion is based upon a two to three year extrapolation of perceived current trends in computer science and engineering.

Part 4. The factors that should be considered in evaluating each option are outlined in the next section. These factors include the impact that each option would have upon software productivity, operational convenience, etc. in addition to the obvious question of its impact on our budget.

Part 5. Each of the options is then considered with respect to those factors.

Part 6. Finally, we describe the conclusions that we were able to reach on this subject and we draw attention to those areas in which it was not possible to reach any conclusion at this time.

PART 1 - THE CURRENT VLA COMPUTING FACILITY

Current Purposes of Computing

Computing at the VLA can be categorized as follows:

a) Process Control - Pointing the telescope and monitoring the behavior of the instrument. The telescope cannot be operated without on-line computer service.

b) Data acquisition - Collecting and recording the data. This includes some preliminary evaluation of the validity of the data.

c) Hardware diagnosis - To log the time variation of instrumental characteristics as well as to provide assistance in isolating the cause of failures.

d) Calibration and editing - This activity establishes the basis for quantitative measurements. Improving the quality of the data by identifying and removing the effects of interference and other destructive phenomena requires human judgment, hence interaction between the computer and the observer. It is hoped that the need for such interaction will decrease in the future.

e) Image formation (map making) - The real purpose of the VLA as an instrument. Frequently the results of this operation lead to an iterative process involving re-calibration and further editing. For that reason, this process should not, in general, be isolated from the one listed previously.

f) Image enhancement - Procedures devised for this purpose have greatly increased the value of VLA observations, but are the most computer intensive component of data analysis at the VLA.

g) Image analysis - Once satisfactory images have been obtained, one must usually make quantitative numerical measurements of various aspects of these images. Here a high degree of interactivity is required. One-of-a-kind programs are needed occasionally.

h) General computing including development of new programs. Adequate software development tools must be available in order to ensure reasonable productivity in this area.

The Equipment

At present VLA computing is distributed among the hardware systems shown in Table 1. The letters that appear under the column labelled "Application" refer to the list presented above.

Table 1 - Equipment List

System	Computers	Array Processors	Applications
On-line computer system	8 Modcomp-II	1	a b c
Asynchronous computer system	DEC-10	0	c d (e)(f) h
Pipeline	4 PDP-11	4	e (f)
AIPS	2 VAX-11/780	2	e f g (h)
AIPS (in Charlottesville)	VAX-11/780 Modcomp-IV	1	e f g (h)

Table 2 (below) shows the power, measured in millions of instructions executable per second (Mips), that is theoretically available in each of these subsystems. Those more comfortable with other measures of computer performance may equate 1 Mips to approximately a quarter million floating point instructions per second (0.25 Mflops). In more concrete terms, this unit is very close to the processing speed of a VAX-11/780.

The aggregate of 17 computers and 10 array processors (APs) currently in use at the VLA represents a substantial resource equivalent, at least at face value, to one of the largest computers in existence. Unfortunately, not all of the available computer power can be used effectively. Table 2 shows how much is actually in use at this time.

Actual measurements were used in Table 2 where they were available. The figures for the DEC-10 and for the VAXs fall into this category. The APs on the VAXs were metered using a special device built by Phil Dooley. The figures for the Modcomps are based on a combination of measurements and rather detailed calculations by Ken Sowinski. Unfortunately, the PDP-11s do not have the facilities required for metering so the entries for these machines and their attendant APs in Table 2 are mostly estimates based on fairly rough calculations.

From Table 2 it is clear that APs represent a very significant portion of the presently available computer power at the VLA. The APs dominate even though the potential power quoted for each is not the burst rate of which it is capable but a considerably lower figure based upon estimates of its maximum practical sustained computing rate. The basis for our assessment that each AP is equivalent to a 6 Mips conventional computer is described in Appendix A.

Table 2 - VLA Computer Power (July 1983)

System	Theoretical Mips	Currently Used	Percentage Used/Theoreti
ON-LINE SYSTEM			
7 MODCOMPs (0.4 Mips each)	2.8	1.2	
1 MODCOMP (spare)	0.4	0.1	
AP	6.0	1.5	
	----	----	
Total for On-line System	9.2	2.8	30 %
CALIBRATION & EDITING SYSTEM			
DEC-10	1.0	0.4	40 %
IMAGE FORMATION (PIPELINE) SYSTEM			
SORTER 11/70	0.5	0.2	
SORTER AP	6.0	0.0	
GRIDER 11/70	0.5	0.2	
GRIDER APs (3 x 6.0)	18.0	1.0	
DISPLAY 11/44	0.5	0.1	
WORKER 11/44	0.5	0.2	
XPOSE MEMORY	(1.0)	(0.1)	
	----	----	
Total for Image Formation System	26.0	1.7	7 %
AIPS			
11/780 (2)	2.0	1.0	
AP (2)	12.0	4.0	
	----	----	
Total for AIPS System	14.0	5.0	36 %
TOTAL FOR ALL SYSTEMS	50.2	9.9	20 %
TOTAL FOR APs ONLY (7 x 6.0)	42.0	6.6	16 %
TOTAL FOR NON-APs	8.2	3.3	40 %

Table 2 depicts a system in transition. Until recently the machines known as MAPPER and IMPS were the busiest of all VLA computers but their previous function is now being performed by the AIPS machines. MAPPER and IMPS have been renamed and assigned to the pipeline system where they are not yet used very much because the pipeline system is not yet in full operation. Almost all of the utilization shown for components of the pipeline is the result of program development and testing.

The current utilization of the DEC-10 was measured at 40% but a significant part of the remaining power in that machine has been allocated to the pipeline for calibration and editing. Furthermore, the DEC-10 figures do not reflect the recent doubling of continuum data due to the implementation of the second IF pair and do not include the period in D array when the spectral line usage was high.

The overall impression given by Table 2 is one of excess rather than inadequate computer capacity. The problem is that much of the theoretical capacity is NOT USABLE. One major reason is that on most of the machines, but most importantly on our APs, the processor speeds are not matched by equivalent memory size nor by equivalent I/O (channel) capacity. In other words, both of these are severe bottlenecks.

The magnitude of this mismatch can be appreciated by noting that the APs in the pipeline are meant to produce maps of 2048 by 2048 pixels on a routine basis and 8192 by 8192 on occasion. Yet the largest AP in the pipeline contains only three "pages" of memory (1 page = 64 kwords) which is less than 2% of the space required for a single routine map. Since the gridding machine cannot hold a full map the input data must be sorted at least into pigeonholes. This makes things awkward when the "natural" time-basdeline order would be appropriate (e.g. when editing or calibrating the data).

On the basis of his experience with the pipeline, Wim Brouw has recommended tripling the AP memory and increasing the transpose memory 16-fold i.e. to 64 million words. (VLA Computer Memorandum No. 157, February 1981.)

Even the PDP-11's, which contain relatively large amounts of memory, are afflicted with bottlenecks because the operating system restricts each job to 64 kbytes of usable address space. (A much larger memory space can be used but only for data and only at a great loss of efficiency incurred from register swapping.)

The other condition that limits the available computer power might be called "computational entropy" by analogy to the factor that limits the amount of work derivable from thermal energy. In computing, as in thermodynamics, diffuse power is less usable than concentrated power when working on problems that cannot be partitioned.

Consider the on-line system, for example. Even apart from the equipment allocated to working "spares," some of the Modcomp CPUs are idle while others are near saturation. The lost cycles cannot be used to relieve the overburdened machines because the CPU-limited tasks cannot be shared. The on-line system uses only a quarter of its AP because the overall duty cycle of the system limits the amount of data to be processed. (The utilization figure for this AP has been calculated and confirmed by measurement by Ken Sowinski). Again, the extra cycles in the AP cannot be put to work because they are in the wrong place.

The AIPS system provides further examples of impediments to efficient use of available computer capacity. Despite being in such great demand that it is scheduled around the clock, and despite efficiency-increasing measures such as a "batch" mode to fill otherwise unused time, only 40% of the CPU time available in the AIPS system is used productively. (The actual figures for the period March 28 to June 19 were 37% for one AIPS system and 42% for the other).

One source of this inefficency in AIPS is the multiplicity of data transfers (tape-to-VAX-to-disk, disk-to-VAX-to-AP, AP-to-VAX-to-disk, disk-to-VAX-to-display) which consume an inordinate percentage of available time.

To some considerable extent, the inefficient use of the AP and the surprisingly low CPU utilization in AIPS are unavoidable. Many of the most frequently performed tasks in AIPS do not lend themselves to the use of an AP. Furthermore, AIPS is intended to be an interactive system in which user-incurred idle time is the norm. In a single-user environment both the AP and the host CPU must wait for work to be available unless the user carefully exploits multitasking opportunities. In principle a multi-user environment would increase efficiency but the non-trivial software for time-sharing the AP (resource allocation) is unavailable. The VAX/AP architecture is generally considered to be extremely attractive for AIPS usage because good user interaction is more important than optimizing the use of CPU cycles.

Even though the pipeline is not yet in production, it is already known to suffer from computer entropy problems. Revisions to the system have been undertaken recently to reduce the number of times the data passed through different machines. On the other hand, the extent to which interaction with the observer will reduce the efficiency of the pipeline is still unknown.

In general, the distribution of CPU power in the VLA computer systems greatly worsens the problem of inadequate memory and I/O capacity that was described earlier because so much data must be transfered between processors. Widespread use of computer-demanding deconvolution methods, which was not foreseen at the time the pipeline was designed, has further magnified these problems.

PART 2 - ESTIMATING VLA COMPUTER NEEDS

At present, as ever since the initial observations made by the VLA, the computer facilities have been a limiting component of the telescope. For a long time now the correlator has been capable of delivering more data than could be processed.

Although some of the limitations have been imposed by the on-line computer subsystem, the restraints that have been most severely felt by the astronomers have been due to limitations in the image formation (map-making) subsystem. The mapping requirements of spectral line observers have been served particularly poorly because they involve some of the heaviest data flow. It was hoped that the pipeline subsystem would eliminate this condition. At this time it is fairly clear that, for reasons already explained, the present pipeline hardware cannot satisfy all of the demands that are being placed upon it.

Even a casual comparison of the original purpose of the pipeline with the expectations that exist 5 years later should suffice as a warning of the dangers inherent in predicting total VLA computer requirements for the next decade. Nevertheless, for planning purposes, the only reasonable way of estimating future needs is to assume that future data process techniques in radio astronomy will be neither more nor less computer intensive than current ones.

It will be convenient to consider the total anticipated computer requirement for imaging at the VLA as a multiple of the computer power needed for an arbitrary (but not unreasonable) spectral-line task which will be described shortly. It is the same task that was considered by Barry Clark in VLA Computer Memo No. 127 dated November 1975 and again in VLA Computer Memo No. 137 dated July 1977.

Another convenient yardstick will consist of a hypothetical machine that, at first glance, would seem to be of sufficient power to carry out the canonical task. Relatively few calculations are needed to demonstrate that, in fact, the hypothetical machine would be just barely adequate for image formation but that it would be inadequate for image enhancement. Finally, in the most crucial portion of this section, the reference task will be compared with those provided by Ekers et al.

A Canonical VLA Imaging Task

Consider the observing program that entails:

1- An average input data rate of 9,000 complex visibility values per second maintained continuously over 12 hours

2- Simultaneous construction of 256 maps, each defined on a grid of 2048 by 2048 points

3- An output rate capabable of storing those 256 maps.

The origin and significance of these specifications can be appreciated by noting that spectral line observations using 256 channels with all of the VLA's 351 baselines generate 89,856 complex visibilities i.e., with the usual 10 second integration period, roughly 9,000 values per second.

A Hypothetical Computer

Consider that the task described above is to be carried out on some machine which is unspecified except for:

1- Its architecture, which is conventional (von Neumann) i.e. it is not an Array Processor.

2- Its processing speed, which is sufficient to perform a 2-dimensional 2048 x 2048 complex FFT in 1 minute. (Roughly equivalent to a processor rating of 20 Mips or, even more roughly, 5 Mflops.)

3- The size of its main memory, a minimum of 8 million words of 4 bytes each (i.e. 32 Mbytes), which is sufficient to hold one copy of an entire 2048 by 2048 complex map (or a map and beam) plus program and ancillary data.

4- Its I/O (channel) capacity, which can support an effective continuous transfer rate of 1 Mbyte per second (8 Mbaud).

Mapping (Image Formation) on the Hypothetical Machine

The canonical task described above will be considered in two parts: mapping (image formation) and CLEANing (image enhancement).

For a rough estimate of the computational load represented by the mapping part of the reference task, we can make several simplifying assumptions. The most important of these is the assumption that the maps can be produced automatically i.e. unencumbered by the (typically interactive) process whereby the raw data is normally edited. Another assumption is that for mapping the following operations suffice:

1- Application of gain and passband corrections once to each input datum.

2- Gridding by convolution of each datum onto a cell of 6 by 6 grid points.

3- Calculation of a beam pattern for each map.

4- Production of maps. Each map will require a 2-dimensional fast Fourier transform (FFT) of the gridded data.

5- Calculation of another 2-dimensional FFT for each beam. Because the map is Hermitian, this process will not require additional resources since the calculation of the beam can be combined with the calculation of the map.

To apply gain and passband corrections to each datum requires the equivalent of roughly 100 instruction cycles under optimum conditions, i.e. the gain tables have been prepared ahead of time, they are available in main memory and the input data is available in time sequence order. (The equivalence makes allowance for those operations that will be performed on floating-point values.) If we use the approximate figure of 20 Mips for the processor speed then, over the 12-hour reference period, this task will occupy the CPU for

$$\frac{9{,}000 \text{ values/sec} \times 43{,}200 \text{ seconds} \times 100 \text{ instructions}}{20{,}000{,}000 \text{ instructions/sec}} = 0.5 \text{ hours.}$$

Convolving the calibrated data onto 36 grid points will take approximately 750 instruction cycles per datum if the inner loop of this process has been tightly coded. The hypothetical machine will be occupied by the gridding task

$$\frac{9{,}000 \text{ values/sec} \times 43{,}200 \text{ seconds} \times 750 \text{ instructions}}{20{,}000{,}000 \text{ instructions/sec}} = 4.1 \text{ hours}$$

A weight must be calculated for each map point in order to normalize the map and, coincidentally, to determine the beam pattern. In the usual case (uniform weighting) this will require a preliminary pass through the data involving an additional 250 instructions per datum. The hypothetical machine will be occupied by the beam-making task

$$\frac{9{,}000 \text{ values/sec} \times 43{,}200 \text{ seconds} \times 250 \text{ instructions}}{20{,}000{,}000 \text{ instructions/sec}} = 1.4 \text{ hours}$$

The 2-dimensional FFT will be effected by a set of 2048 1-dimensional FFTs in each of two orthogonal directions. (The total number will be 25% less because the Hermetian properties of the data make half of the FFTs in the first direction unnecessary.) Between each set of transforms the grid must be transposed but, since it is assumed that the entire grid can be in memory simultaneously, no allowance has been made for this process.

From the specification that each (full) 2-dimensional FFT would require 1 minute of processing time we find that each 1-dimensional complex 2048-point FFT should take a bit more than 15 milliseconds. So mapping will require:

15 milliseconds/FFT x (1024 + 2048) FFTs x 256 maps = 3.3 hours.

Turning next to the I/O requirements of the task, we start by determining the total number of bytes involved in the input data set and in the output maps. Each input datum is a complex pair whose real and imaginary parts require 2 bytes each. The 12-hour volume of data is therefore:

9000 values/sec x 4 bytes/value x 43,200 secs = 1.5 Gbytes.

In addition to the raw data, a certain amount of overhead is necessary to identify data clusters and to access the data effectively. The identifying information consists of values of u, v, w, t, baseline and flags. Each identifier requires 16 bytes. If a cluster of data (a record) consists of one identifier for each set of 256 channels the overhead is only 16/1024 or less than 1.5%. This is the minimum overhead and the most efficient use of storage.

Unfortunately, the mapping process cannot use data for all 256 channels at one time so, under the identification scheme described above, each cluster would have to be reread many times and the I/O volume would be multiplied by a large factor. A similar problem is encountered if one goes to the opposite extreme of identifying the data for each channel separately. In that case the data storage overhead would be a prohibitive 16/4 or 400%. A reasonable compromise between these two extremes appears to be one identifier per cluster of 8 channels for an overhead of 16/32 or 50%.

The total input data volume is then 2.2 Gbytes.

but for mapping, each value must be read 8 times (once for each channel in the group since we assume only one channel will be mapped at a time).

The total input data volume for mapping is then 17.6 Gbytes.

The output data are the pixels in the 256 maps. If each pixel is stored in a 4-byte floating point number the output data volume is:

256 maps x (2048 x 2048) pixels x 4 bytes/pixel = 4.3 Gbytes.

Properly identifying maps does not usually entail nearly as much overhead as that required for the identification of the input data. In fact such overhead for maps can be neglected. What we should not neglect, however, is the storage requirement for beams. Since the canonical task deals with spectral line data, no more than 10% of the beams will be needed. Furthermore, since beam patterns are symmetrical, each one will require only half as much space as the corresponding map. So the need to store beams raises the 12-hour volume of output to

4.3 Gbytes + 0.2 Gbytes = 4.5 Gbytes.

Since our hypothetical machine can perform continuous high speed I/O, averaging a full Mbyte per second (8 Mbaud) over the entire 12-hour period, the time required to pass all the input data through the I/O channels will be 2200 seconds (0.6 hours), the input for mapping will be 8 times that much (4.8 hours), and the time required to write the 256 maps and beams will be 1.2 hours.

It is unlikely, however, that the input data can be used synchronously i.e. that it can be processed as soon as it is available and that it can be discarded thereafter. The need to buffer the input data means that it must pass through an I/O channel at least 3 times: once to acquire it from the original source, again to store it on a disk and a third time to read it back when it is needed in the map-making process. As a matter of fact, the standard practice for gridding (use of uniform weighting) requires that the data be read twice from the disk.

The output must be transfered asynchronously to a display system which means that it too must be stored (temporarily at least) on a disk.

In summary, here is how a 5 Mflops machine, with 1 Mbyte/sec average I/O bandwidth and somewhat more than 32 Mbytes of memory available for the process, would handle the canonical imaging task:

	Operation	Hours
I/O	Original Input	0.6
	Output to Buffer Disk	0.6
	First Input to Mapping	0.6
	Second Input to Mapping	4.8
	Original Map and Beam Output	1.2
	Rereading Output from Disk	1.2
	Writing Maps & Beams to Tape	1.2

		10.2

CPU	Calibration	0.5
	Weight Determination	1.4
	Gridding	4.1
	FFT	3.4

		9.4

Thus, even with fully overlapped I/O and computation, with no CLEANing (let alone self-calibration), with no display, no calculation of gain tables, and no editing, the mapping part of the canonical task would come close to saturating the hypothetical 5 Mflops machine, especially if the operating system overhead were a typical 15 to 20% of application program time.

The figures quoted above are obviously only rough estimates. It may not be quite so obvious that each approximation has been biased in the direction of underestimating rather than exaggerating the computer load. For example, in addition to all of the exclusions listed in the previous paragraph, the requirements have been underestimated further by the following:

1. Data is assumed to be available in main memory whenever it is needed. Inadequate memory would certainly increase the time estimates by a significant factor. In particular, the time to transpose the data between the two sets of FFTs would no longer be negligible.

2. The CPU time required by programs that transfer data from the original source to the disk buffer (the FILLER program), to copy the output maps from disk and reformat them for tape (the FITS program), and to write a calibrated version of the database onto tape for export to other installations (the UVFITS program) have all been neglected.

3. It has been assumed that the data could be organized to allow streaming at 1.0 Mbyte per second into and out of the central processor. In particular, it has been assumed that no sorting is required.

4. A multi-user environment would (presumably) require paging and greatly increased system overhead. An interactive environment (even with a single user) would also slow down the process by the amount of time the system would be waiting for user responses.

5. No provision has been made for archiving either the original data or the maps (i.e. neither BACKUP nor RESTORE have been included in the I/O requirements).

The only way in which the necessary resources may have been overestimated is by the assumption that all of the input data will be used to make maps. This ignores the phase, amplitude and passband calibrators which, in general, will not be mapped. However these sources will be used for their respective purposes and will have their own impact on system resources.

Increasing Dynamic Range (Image Enhancement)

The second part of the reference task is CLEANing (image enhancement). This computer-intensive deconvolution procedure, together with the relatively new technique of self-calibration, has upgraded the VLA from an instrument with 100:1 dynamic range into one that routinely achieves 1,000:1 and in some cases even 10,000:1 dynamic range. In other words, the value of VLA data has been increased many times over at the expense of a multifold (and largely unanticipated) increase in the amount of computer power required.

Tim Cornwell has calculated the amount of computer processing required by the CLEAN algorithm in terms of an equivalent number of 2-dimensional FFTs. In a memo distributed informally within the VLA (reproduced here as Appendix D) Cornwell estimates that

1 CLEAN deconvolution = (2 + 2 N) 2-dimensional FFTs

where N is the number of "major cycles" required to achieve the desired dynamic range. Typical values of N range from 4 to 200 but fortunately, for spectral line work, the preponderant value is 4. Even so, this means that it takes at least 10 times as long to CLEAN a map as to make the map initially. On the hypothetical machine considered above, CLEANing the 256 maps (12 hours of data) would take

256 maps x 10 FFTs/map x 1 min/FFT = 2560 minutes = 43 hours !!

It is because CLEANing is so dominant that it has been considered separately from the rest of the canonical task described above!

The CLEAN algorithm requires that the initial map and beam be reread from the disk. Obviously, we will want to save the improved map so it must be written to disk. In summary, the need to CLEAN the 256 maps (2048 by 2048) produced by the canonical task described above imposes the following load on the hypothetical machine:

	Operation	Hours
I/O	Read Maps and Beams from Disk	1.8
	Output CLEANed Maps	1.2

		3.0
CPU	10 FFTs per channel	43.0

Self-calibration (automatic recalculation of antenna gains) is another computer-intensive process that is used in the quest for ever higher dynamic range.

Tim Cornwell has estimated the computer load represented by a self-calibration step. It turns out to be approximately the same as one 2-

dimensional FFT followed by a complete re-mapping. The actual redetermination of gain values is relatively insignificant.

Although self-calibration is a large percentage of the work required by continuum observations, it is a small factor in spectral line work because the gain figures obtained by self-calibrating one channel can be used to map all of the other 255 channels. So, for the canonical task, self-calibration can be ignored.

Summarizing the total computer requirements of the canonical task:

	Operation	Hours
I/O	Mapping	10.2
	CLEANing	3.0

	TOTAL	13.2
CPU	Mapping	9.4
	CLEANing	43.0

	TOTAL	52.4

The conclusion that emerges from these calculations is that the computer requirements of the canonical task exceed the capacity of the hypothetical machine by a factor of 4 or 5. In other words, a machine of 20 to 30 Mflops would be needed merely for image formation to process the data at the rate it is observed. Real circumstances, in which the ideal assumptions made in these calculations were not all valid, plus ancillary tasks (such as data editing) would raise the level of computer power required to handle the canonical task as an average load to about double that figure or to about 40 or 50 Mflops.

It now remains to be shown whether or not the canonical task is representative of the anticipated work to be done at the VLA during the next decade.

Foreseeable Computer Requirements

A study group comprising Ron Ekers, Ed Fomalont, and Frazer Owen with the help of Tim Cornwell, Jacqueline van Gorkom, and Pat Palmer has tried to predict the demands that might be placed against VLA computing facilities from all sources, spectral line and continuum alike, in the next few years.

The study group began by characterizing each type of observational experiment suitable for the VLA in terms of its astronomical objectives. From these the committee deduced the required observational parameters which have been translated, in turn, into implied computer I/O rates and processing needs (for image formation and enhancement only). The conclusions of Ekers et al. are summarized in Table 3.

The work of Ekers et al. is absolutely fundamental to this analysis of VLA computer requirements. Given a fixed set of objectives and the computational effort required to meet each, the size of the machine needed by the VLA follows. (Naturally, professional judgement also influences the size calculations.) Different objectives, of course, would lead to different computer requirements.

The processor load for each research area was first obtained by using Cornwell's equation to determine the equivalent number of 2-dimensional Fourier transforms. The number of data points in this transform was taken to be the appropriate number of pixels on each side of the map as given by Ekers et al. In any actual experiment the map size would probably be a power of 2 but the numbers derived by Ekers et al. were used as given, on the grounds that some users would ask for the next larger size while others would make do with the next smaller size. The number of equivalent 2-dimensional 2048 by 2048 complex FFTs (interpretable as minutes of processing time on the hypothetical 5 Mflops machine) was obtained by scaling according to

$$N^2 \mathrm{LOG}(N).$$

Ekers et al. have estimated the percentage of all VLA observations that would be made for each of the experiment types. Those estimates were used to combine the calculated requirements for each experiment type in a weighted average that represents the computer requirements to process a "typical" 12 hours of VLA observing.

For continuum observations the amount of work required by each of the experimental areas was estimated from the following equations:

$$Nfft = Obs * (DC + 30 * SC\%)$$

where

Nfft — is the number of 2D FFTs (or equivalent) to be performed. The number of data points in each FFT is that appropriate for the experiment i.e. has not yet been scaled to minutes of processing on the hypothetical 5 Mflops computer.

Obs — is the number of sources observed in this type of experiment during a typical 12-hour period.

DC — is the work required for mapping and for CLEANing each source. It is obtained from

$$[Np/2 + (Np-1) * 12 * C\%] * R$$

Np — is the number of polarizations observed. The constant 2 reflects the fact that, while some experiments require a separate map for each IF, others require only one. On average, the number of maps should be about half the number of IFs observed.

C% — is the percentage of maps to be CLEANed. (The constant 12 is based on the belief that, on average, only 5 major cycles are required to CLEAN each CONTINUUM map to the desired level.)

R — is a repetition or "re-try" factor. It is an estimate of the number of times a given data set might be processed.

SC% — is the percentage of maps to be self-calibrated. The constant 30 assumes that self-calibration will be carried out for an average of 2.5 iterations. Approximately, the iterations cost one FFT for the initial map, 2+2*N FFTs for CLEAN, followed by 2 FFTs for the self-calibration step itself. N, the number of major CLEAN cycles, is expected to be approximately 2, 4, 6, etc. for successive iterations. The total equivalent FFT load for the average self-calibration task, therefore, is

$$(1+6+2) + (1+10+2) + (1+14+2)*0.5 = \text{(app) } 30$$

Table 3A (on the following page) shows the result of applying these equations to the values supplied by Ekers et al. for CONTINUUM experiments. The leftmost columns are labelled with symbols that have been used in the equations. The labels on the righthand columns carry the following meaning:

$$\text{Time} = (\text{Equiv} + 3.3 * \text{Vis}) * \text{Mix}$$

where

Time is the number of minutes of processing time required on the hypothetical 5 Mflops machine for the given type of experiment in a "typical" 12-hour mix of CONTINUUM experiments.

Equiv is the equivalent number of 2048 x 2048 2-dimensional complex FFTs. It is given by

$$\text{Equiv} = \frac{(\text{Npix} * \text{Npix}) * \text{LOG}(\text{Npix})}{(2048 * 2048) * \text{LOG}(2048)} * \text{Nfft}$$

Npix is the number of pixels on each side of the maps required for the given experiment.

Vis is the number of words of visibility data read in this experiment (in millions of words per 12 hours). The constant 3.3 is the number of minutes required to grid a million words of data.

Mix is the estimated percentage of CONTINUUM observing at the VLA during which the given type of experiment is conducted.

Table 3A - Processing time on a Hypothetical 5 Mflops Machine for the anticipated 12-hour mix of CONTINUUM observations

Type of Experiment	Obs	Np	R	C%	SC%	Npix	Nfft	Equiv 2K	Vis Mw	Mix %	Time min
Full field mapping											
A array											
1.4 GHz	6	4	2	100	75	5400	591	4762	400	4	190
5 GHz	6	4	2	100	75	6000	591	5918	400	5	295
15,22 GHz	6	4	2	100	75	5400	591	4696	200	2	93
B array											
1.4 GHz	6	4	2	100	75	1600	591	391	130	5	19
5 GHz	6	4	2	100	75	1800	591	471	70	3	14
15,22 GHz	6	4	2	100	75	1600	591	354	16	1	3
C,D arrays											
1.4 GHz	6	4	2	100	75	500	591	39	32	2	0
5 GHz	6	4	2	100	75	600	591	45	8	5	2
15,22 GHz	6	4	2	100	75	500	591	30	4	3	0
Small Objects											
Weak Sources 327 MHz	12	2	2	100	100	9400	672	16993	23	6	1019
1.4 GHz	6	4	2	100	50	5400	546	4287	26	6	257
5 GHz	6	4	2	100	50	3000	546	1232	7	5	61
15,22 GHz	6	4	2	100	50	300	546	10	4	4	0
Strong Sources (all)	6	4	2	100	100	300	636	11	4	5	0
Snapshots											
Weak Sources 1.4 GHz	100	3	1	100	50	5400	4050	31741	13	4	1269
5 GHz	100	3	1	100	50	3000	4050	9126	2	5	456
12,22 GHz	100	3	1	100	50	300	4050	65	2	4	2
Strong Sources (all)	200	4	1	100	100	300	13600	219	4	9	19
Survey											
B array, 1.4 GHz	720	2	1	100	20	1100	13680	3625	3	1	36
C array, 1.4 GHz	4320	2	1	20	0	500	14688	714	2	2	14
Point Sources											
Astrometry (A array)	100	2	1	0	0	100	100	0	2	3	0
Monitoring, Spectra	100	4	1	0	50	100	1700	3	2	2	0
Flare Stars	10	4	1	0	0	100	20	13	40	1	0
Detections	24	2	1	0	0	100	24	0	2	7	0
Solar											
Quiet	2	4	1	100	0	500	76	5	4	1	0
Active	20	4	1	100	0	500	760	63	81	3	1

Time to process 12 hours of "typical" CONTINUUM observations: 63 hours

Visbilities for "typical" CONTINUUM observations: 58 Mwords / 12 hours

For spectral line observations the amount of work required by the experimental mix foreseen for the VLA was estimated from the following equations:

Nfft = Obs * [(Np-1) * (Ch * 1.3) * (1 + B% + 10 * DC%) + 15 * S]

where the symbols are mostly as defined for the CONTINUUM case, i.e.

- **Nfft** is the number of 2D FFTs (or equivalent) to be performed. The number of data points in each FFT is that appropriate for the experiment.
- **Obs** is the number of sources observed in this type of experiment during a typical 12-hour period.
- **Np** is the number of polarizations observed.
- **Ch** is the number of channels observed. The constant 1.3 reflects the belief that, on the average 10% of the channels will be mapped 3 times before the full set is mapped.
- **DC%** is the percentage of maps to be deconvolved (CLEANed). The constant 10 is based on the belief that, on average, only 4 major cycles are required to CLEAN a SPECTRAL LINE map to the desired level.
- **B%** is the percentage of channels for which beams are mapped.
- **S** is a constant which has the value 0 or 1. It will be 1 if a map (for 1 channel at most) is to be self-calibrated, otherwise it will be 0. The constant 15 is an estimate of the work required for self-calibration when it is required.

Table 3B (on the following page) shows the result of applying these equations to the values supplied by Ekers et al. for SPECTRAL LINE experiments. The leftmost columns are labelled with symbols that have been used in the equations. The labels on the righthand columns carry the following meaning:

$$\text{Time} = (\text{Equiv} + 3.3 * \text{Vis}) * \text{Mix}$$

where

Time is the number of minutes of processing time required by this type of experiment in a "typical" 12-hour mix of all types of CONTINUUM experiments.

Equiv is the equivalent number of 2048 x 2048 2-dimensional complex FFTs. It is given by

$$\text{Equiv} = \frac{(\text{Npix} * \text{Npix}) * \text{LOG}(\text{Npix})}{(2048 * 2048) * \text{LOG}(2048)} * \text{Nfft}$$

Npix is the number of pixels on each side of the maps required for the given experiment.

Vis is the number of words of visibility data read in this experement (in millions of words per 12 hours). The constant 3.3 is the number of minutes required to grid a million words of data.

Mix is the percentage of all SPECTRAL LINE observing at the VLA during which the given type of experiment is conducted.

Table 3B - Processing time on a Hypothetical 5 Mflops Machine for the anticipated 12-hour mix of SPECTRAL LINE observations

Type of Experiment	Obs	Np	Ch	DC%	S	B	Npix	Nfft	Equiv 2K	Vis Mw	Mix %	Time min
Extragalactic Emission												
HI Clusters	1	2	16	20	0	12	1100	64	24	21	6	1
HI Galaxies	2	2	64	75	0	3	500	1419	90	65	12	10
ISM (individ. gal)	3	2	130	50	0	15	5400	3118	24712	850	6	1482
Recomb. line	1	2	8	50	1	1	200	77	3	8	1	0
NH3	1	2	8	10	0	0	100	20	2	8	1	0
H2O Masers	3	4	4	0	0	7	6000	50	509	58	1	5
OH Masers	3	4	64	0	0	6	5400	793	6495	839	1	64
Extragalactic absorption												
HI	3	2	64	50	1	2	400	1547	67	65	2	1
OH	2	4	65	50	1	2	400	3082	135	130	3	4
H2CO	2	2	65	50	1	7	1200	1055	368	95	2	7
Galactic Center												
NH3	2	2	16	50	1	0	100	279	5	16	3	0
H2CO	1	2	64	50	1	3	5400	516	4186	419	1	41
HI, OH	1	2	256	50	1	3	5400	2021	16400	1700	1	164
Recomb. Line	1	2	8	50	1	4	1600	77	51	16	1	0
Galactic Absorption												
HI	3	4	130	100	1	3	5400	16821	132377	1700	2	2647
OH	3	4	130	100	1	1	1800	16791	12938	570	2	258
H2CO	3	2	130	100	1	3	6000	5637	55517	950	2	1110
NH3	1	2	128	100	1	0	6000	1845	18377	930	4	735
Galactic Emission												
HI	2	2	52	50	0	1	400	812	41	52	2	0
NH3	1	2	128	10	1	0	100	347	43	130	12	5
Recomb Lines	2	2	65	50	1	0	100	1044	23	66	11	2
OH Masers	12	4	130	50	1	1	1800	36744	28090	570	7	1966
H2O Masers	2	4	32	5	1	2	6000	409	4163	469	6	249
Stars												
OH Masers	12	4	156	50	1	0	100	43984	168	320	6	10
H2O Masers	12	4	32	50	1	0	600	9165	681	65	3	20
SiO Masers	12	4	16	50	1	0	1200	4672	1507	47	2	30

Time to process 12 hours of "typical" SPECTRAL LINE observations: 147 hours

Visbilities for "typical" SPECTRAL LINE observations: 314 Mwords / 12 hours

Table 3C - Processing time on a Hypothetical 5 Mflops Machine for the anticipated 12-hour mix of SPECTRAL LINE observations ASSUMING AN ENHANCED CORRELATOR

Type of Experiment	Obs	Np	Ch	DC%	S	B	Npix	Nfft	Equiv 2K	Vis Mw	Mix %	Time min
Extragalactic Emission												
HI Clusters	1	2	325	20	0	12	1100	1318	490	430	6	29
HI Galaxies	2	2	130	75	0	3	500	2883	182	130	12	21
ISM (individ. gal)	3	2	130	50	0	15	5400	3118	24712	850	6	1482
Recomb. line	1	2	130	50	1	1	200	1030	49	130	1	0
NH3	1	2	325	10	0	0	100	844	109	330	1	1
H2O Masers	3	4	325	0	0	7	6000	4068	41386	4700	1	413
OH Masers	3	4	325	0	0	6	5400	4030	32995	4300	1	329
Extragalactic absorption												
HI	3	2	130	50	1	2	400	3097	135	130	2	2
OH	2	4	65	50	1	2	400	3082	135	130	3	4
H2CO	2	2	65	50	1	7	1200	1055	368	95	2	7
Galactic Center												
NH3	2	2	520	50	1	0	100	8142	185	530	3	5
H2CO	1	2	520	50	1	3	5400	4091	33175	3400	1	331
HI, OH	1	2	520	50	1	3	5400	4091	33175	3400	1	331
Recomb. Line	1	2	325	50	1	4	1600	2566	1725	640	1	17
Galactic Absorption												
HI	3	4	130	100	1	3	5400	16821	132377	1700	2	2647
OH	3	4	130	100	1	1	1800	16791	12938	570	2	258
H2CO	3	2	130	100	1	3	6000	5637	55517	950	2	1110
NH3	1	2	416	100	1	0	6000	5963	59388	3000	4	2375
Galactic Emission												
HI	2	2	52	50	0	1	400	812	41	52	2	0
NH3	1	2	416	10	1	0	100	1096	139	419	12	16
Recomb Lines	2	2	65	50	1	0	100	1044	23	66	11	2
OH Masers	12	4	130	50	1	1	1800	36744	28090	570	7	1966
H2O Masers	2	4	260	5	1	2	6000	3112	31727	3800	6	1903
Stars												
OH Masers	12	4	156	50	1	0	100	43984	168	320	6	10
H2O Masers	12	4	156	50	1	0	600	43984	3272	320	3	98
SiO Masers	12	4	156	50	1	0	1200	43984	14193	460	2	283

Time to process 12 hours of "typical" SPECTRAL LINE observations: 228 hours

Visbilities for "typical" SPECTRAL LINE observations: 835 Mwords / 12 hours

The estimates of future research levels at the VLA are based upon a great deal of painstaking work and many hours of discussion with numerous well informed observers. Nevertheless, the quantitative results in Table 3 must be considered to be no better than rough estimates. The following are some of the sources of uncertainty in these values:

1- They are neither "worst case" nor "average" but somewhere in between; call them "mildly severe" cases. It is not possible to quantify that description very closely.

2- They assume a perfectly distributed work load (i.e. slack time consistently filled by backlogged work). The wasteful effects of uneven use could easily double the computer power needed for busy (or even normal) periods.

3- The introduction of new mathematical procedures that are more demanding computationally than those presently in use could easily double or even quadruple future VLA computer needs.

Such uncertainties must be kept in mind as we examine the implications of Table 3.

If we assume that there will be roughly equal amounts of continuum and spectral line observing at the VLA, the time needed to process 12 hours of observations on a 5 Mflops machine would be 109 hours! Looking at this result in a different way, the mix described by Ekers et al. calls for a machine of at least 45 Mflops (60 Mflops if an enhanced correlator is assumed) merely for image formation and enhancement. A practical machine to meet these and other concomitant needs in a non-ideal environment would have to be capable of delivering more computer cycles, i.e. it should be rated at no less than 60 to 75 Mflops.

If one excludes the on-line and the AIPS systems, the computer power theoretically available at present at the VLA is 27 Mips (see Table 2). That is the equivalent of about 6 or 7 Mflops. The estimated average processing requirements in Table 3 therefore exceed even the theoretical value of the current computer capacity at the VLA by an order of magnitude and, of course, they exceed the actual capacity by much more. There are three fundamental reasons for such a large increase: more data from the telescope (more channels, shorter integration times, etc.), greater detail in the output (larger maps) and, most important of all, greater exploitation of deconvolution techniques (CLEAN).

Frequent use of map sizes in the range of 4000 to 7200 pixels on a side is predicted; at present a map size in excess of 1024 by 1024 pixels is exceptional. It must be pointed out that construction of such large maps is useful even in the absence of display devices that can handle such large images. Large maps are processed for data that is confusion limited in order to remove from the field of view instrumental effects caused by objects outside that area.

The equations used to derive Table 3 underestimate the computational load represented by the astronomical objectives of Ekers et al. in a subtle but significant way: they do not allow for the inefficiency that arises when a map is too large to fit in memory. Of the 27 CONTINUUM experiments listed, one quarter of them, accounting for more than a third of all CONTINUUM observing, involve maps that are larger than 2048 pixels on a side. One third of the SPECTRAL LINE experiments (accounting for 25% of observing in this mode) also require the construction of large maps.

It is clear from Table 3 that some types of experiments are far more costly than others in terms of computer processing requirements. Two CONTINUUM experiments, representing only 10% of observing in that mode, account for 60% of the computer resources estimated for all CONTINUUM processing. In SPECTRAL LINE mode, 5 experiments, involving 21% of observing, require 90% of the computer power. Clearly, decisions by the VLA scheduling committee can limit the amount of observing done at the VLA in each category and thereby can drastically change the overall computer needs.

On the other hand, Table 3C shows that the computer requirements for SPECTRAL LINE experiments would double if the objective were to process as much visibility data as could be used profitably in any astronomical experiment. Of course Table 3C makes the rather optimistic assumption that the VLA would have a correlator capable of delivering such data. But it is not unreasonable to assume some improvement in the correlator during the expected life of the next generation VLA computer facility.

Some Worst Cases

The figures in Table 3 do NOT represent worst cases. The following are some experiments that have been nominated (by Frazer Owen and Pat Palmer) for this honor:

1- Some scientists may want to use the A array spectral line system for continuum snapshots. This could effectively increase the number of "sources" observed in the A array a hundredfold.

2- Outrigger antennae (i.e. VLBA antennae within 100km of the VLA) could be used to map the entire primary beam of the VLA. This would require a map size whose dimensions were increased by a factor of 3 in each direction. It would also require three times as many channels. The result would be an overall increase in equivalent 2048 x 2048 FFTs by a factor of 27. (Combined with the requirements of the previous entry, this would increase processing requirements by a factor of 2700!)

3- Observations of water masers in W49 would call for data in 1300 spectral line channels and maps of 6000 pixels on a side. If the correlator could support this data rate and if 1% of a "typical" 12 hour period were spent in this type of work, then processing that 7 minutes worth of data would take about 90 hours on our hypothetical 5 Mflops machine.

4- A search for stars emitting OH lines (or an OH survey) would introduce the need for spectral line snapshots. Pat Palmer has estimated that this might involve observations of 36 objects per 12 hour interval, data in 120 channels and maps that are 5400 pixels on a side. Each 7 minutes of this type of observing would call for 125 hours of computer processing.

To quote one astronomer: "Of course, this does not exhaust the list. Still worse cases will probably be thought up by the next generation of radio astronomers."

Requirements Foreseen but not Included

The forecasts of Ekers et al. take into account the forthcoming addition of 327 MHz capability at the VLA. Experiments at this new frequency will add significantly to the VLA computer requirements. Although the VLA is undoubtedly the foremost instrument in the world for radio astronomy and is only a few years old, the addition of a 327MHz capability is but one of several major upgrades that are planned or proposed for the next decade. Others are:

1- A 75 MHz capability

2- A millimeter wavelength array

3- A 44 GHz capability

4- An extended array using nearby VLB antennae

The first two items in this list represent essentially independent instruments which could merely share a physical site with the VLA. Computer facilities for these instruments should be considered separately from those needed by the VLA proper. The other two items are sufficiently speculative that their effects on VLA computing needs were not considered by Ekers et al.

The fact that this document has not included any of these items should not be taken to mean that anyone believes that the VLA will remain unchanged during the next decade. It simply means that when plans are being formulated for any major change they must include plans for corresponding changes in the VLA computer facility. The fact that changes are inevitable (and that they will almost certainly require MORE rather than less computing) can only be acknowledged in these plans by attempting to provide a reasonable amount of flexibility rather than by providing actual computer power.

I/O Requirements

The discussion in this section has centered thus far on the processing requirements of the VLA. This should not be taken to mean that the I/O problems are negligible. To the contrary, they are substantial and failure to provide for them adequately is one of the main reasons for the disappointing performance of some of our present equipment. (According to some measurements made by Wim Brouw, the effective data transfer rate through the present pipeline configuration is approximately 35 kbytes/sec or 30 times less than what we need.)

Part of the solution to the I/O problem is to avoid making it worse than it needs to be! In other words, avoidance of multiple data transfers that result from inadequate main memory or from distributed processing.

A second part of the solution is to adopt a computer system for which adequate I/O software exists. Almost every computer I/O channel on the market has hardware characteristics that would be adequate for the data transfer rates required by the VLA if that channel could be used continuously at its rated speed. In practice, software limitations or limitations in a combination of peripheral hardware and software limit the duration of full channel speed to small bursts. Typical problems are:

1- Small block sizes (units of data transfered).

2- Non-contiguous file space that requires excessive physical motion of the read/write mechanism.

3- Non-contiguous file space might also require chaining through a file (multiple disk accesses) to find or store specific data.

4- Complex file structure requiring multiple disk accesses for indexes or other record-keeping operations to support each true data transfer.

5- Excessive operating system calculations for each data transfer.

6- Sluggish response to I/O interrupts.

Another fundamental problem occurs in smaller machines: a fixed limited number of memory access cycles must be shared between the CPU and the I/O process. For example, processing in the VAX-11/780 CPU and I/O activity in the VAX mutually interfere.

The maximum effective continuous data transfer rate of a given hardware configuration is not a performance figure that is generally quoted by computer vendors. The reason is that, in addition to depending upon the factors just mentioned, it depends upon the characteristics of the user's data and, as a consequence, it will vary from one application to another.

Summary of Requirement Estimate

It is possible that, by presenting an analysis of the canonical image preparation task, this plan has been belaboring the obvious. After all, a closely related case was examined by Barry Clark 8 years ago and again 6 years ago (VLA Computer Memoranda Nos. 127 and 137).

The only difference between the task considered by Clark and the canonical task analysed here is that he assumed that a 3-dimensional (u,v,w)-cube would be mapped but that no enhancement (CLEANing) would be required. When adjusted to account for these differences, the machine power necessary to accomplish the task defined in his papers works out to be 5 Mflops for initial mapping and 50 Mflops for cleaning - the same numbers that emerge from our analysis.

At the time the Clark memos were written the canonical task was considered to represent an extreme case. The work done by Ekers et al. bring out the fact that it no longer is, even before considering the ten-fold increase in processing required for CLEANing. Maps as large or larger than 2048 pixels on a side are called for in continuum experiments that account for 30% of VLA observation time in that mode and 25% of observing time in spectral line mode.

The translation of requirements listed by Ekers et al. into computer needs was based on solidly established equations (due to Cornwell) that convert the number of maps needed, their size, and the extent to which they must be CLEANed into an equivalent number of 2-dimensional FFT calculations. These then imply a need for computer power in the range of 60 to 75 Mflops.

Needless to say, that estimate would have to be revised if some unexpected development caused a significant change in the distribution of the projects conducted at the VLA or a change in the processing techniques that were appropriate for that work. If, however, one takes the estimates of Ekers et al. as given, 60 to 75 Mflops is a LOWER bound to the VLA computer needs.

Some rather important characteristics of the estimated VLA computer requirement can be observed by ranking all anticipated observing programs according to the contribution each makes to the overall total. In Table 4 the cummulative sum of the individual requirements is shown along with the cummulative percentage of observing time expected for these programs.

It is immediately obvious from Table 4A that the VLA telescope is expected to be used during a relatively large percentage of the time for observing programs that require almost nothing by way of computer facilities. Furthermore, programs which, in aggregate, are expected to account for about half of the total observing time require no more computer processing than that available from a single system equivalent to the VAX/AP combination currently used at NRAO for AIPS. (In Appendix B it is demonstrated that we currently achieve about 0.2 Mflops from such a combination. Remember though, that the figures in Tables 3 and 4 refer to map creation and enhancement only - other tasks add to these requirements.)

At the opposite end of the scale, Table 4 demonstrates that a few very computer-intensive experiments account for a large percentage of the total VLA computer requirement. These experiments cannot be handled by a single small system of the type refered to above, nor (practically) by any number of such systems. A slight variation in the estimated amount of observing time devoted to any of these experiments can drastically alter the conclusions one reaches about the required VLA computer facility. The sensitivity of our conclusions to such uncertainties is less if an enhanced correlator is assumed (Table 4B) because, in that event, more computer-intensive experiments would be conducted.

Table 4A
Current Correlator

% obs	Mflops	% Mflops
100.0	43.7	100.00
99.0	34.5	78.96
95.5	27.7	63.33
92.5	22.5	51.55
90.5	18.1	41.46
89.5	14.3	32.64
86.5	10.7	24.54
84.5	8.2	18.69
82.0	6.6	15.07
79.5	5.6	12.72
78.5	4.7	10.66
75.5	3.8	8.62
72.5	2.9	6.63
70.5	2.2	5.12
70.0	1.7	3.81
69.0	1.3	3.07
68.5	1.1	2.55
66.0	0.9	2.06
65.5	0.8	1.73
65.0	0.6	1.44
64.0	0.5	1.20
62.5	0.5	1.04
58.0	0.4	0.88
55.5	0.3	0.73
54.5	0.3	0.61
53.0	0.2	0.50
47.0	0.2	0.41
44.0	0.1	0.33
43.0	0.1	0.27
37.0	0.1	0.23
36.5	0.1	0.19
35.0	0.1	0.16

Table 4B
Enhanced Correlator

% obs	Mflops	% Mflops
100.0	60.5	100.00
99.0	51.3	84.80
97.0	43.0	71.16
93.5	36.2	59.87
90.5	29.6	48.94
87.5	24.4	40.43
85.5	20.0	33.14
84.5	16.2	26.76
81.5	12.6	20.91
79.0	11.1	18.29
78.5	9.6	15.91
78.0	8.5	14.00
77.5	7.3	12.10
77.0	6.2	10.21
74.5	5.1	8.51
73.5	4.2	6.88
72.5	3.3	5.39
69.5	2.4	3.91
67.5	1.7	2.82
66.0	1.4	2.26
65.0	1.0	1.72
62.5	0.8	1.36
62.0	0.7	1.16
59.0	0.6	0.99
53.0	0.5	0.86
48.5	0.5	0.75
46.0	0.4	0.63
45.5	0.3	0.54
39.5	0.3	0.44
38.5	0.2	0.36
37.0	0.2	0.28
34.0	0.1	0.22

Twenty additional experiment types require, in aggregate, 0.1 Mflops or less.

PART 3 - DESCRIPTION OF OPTIONS

An estimate of the future image formation objectives of observers at the VLA has indicated that a computer system capable of performing 60 to 75 Mflops will be needed at the VLA for off-line processing. Given the uncertainties in the estimates that formed the basis for this requirement it is possible that more power will actually be needed or that somewhat less power would suffice. At the high end of our estimates it is fairly obvious that only one option exists: a large supercomputer. This may also be true at the low end but there exists at least a possibility that computing power close to 60 Mflops could be obtained by alternative means.

The following pages contain a description and an evaluation of each of four different ways in which the requisite computer power might be supplied. Only three of these options are considered to be viable. The fourth has been included because it is the subject of so much discussion that to omit it might distract the reader from giving serious consideration to the options that deserve it. The three viable options are:

A- A Supercomputer.

B- A large but conventional computer supported by a Super AP.

C- Several smaller machines of conventional architecture supporting a commonly shared Super AP.

The fourth configuration which will be discussed, even though it is not considered a viable alternative, is:

D- A large number of microprocessors, each equiped with a small AP.

It may be worth noting that these options span almost the entire range of computing hardware: supercomputer, mainframe, supermini computer and microprocessor. No one (yet!) has suggested using a million pocket calculators.

There are some computer components that will be needed by the VLA regardless of the option chosen. The following items will be a part of the equipment list for each option:

1- 20 Gbytes of on-line auxiliary storage (disks).

2- 6 high-density (6250 bpi) tape drives.

3- 64 interactive terminals.

4- An additional 12 or more ports for dial-up access.

5- 4 image graphics workstations.

6- A relatively high speed (
with other NRAO sites.

7- Unpredictable "fine tuni

Option A. A Supercomputer

The term "supercomputer" refers to one of only two or three very special machines: a CRAY-1, a Burroughs BSP or a CDC Cyber 205. Supercomputers are distinguished from other large but conventional machines by their logical organization (architecture) which departs from the von Neumann model followed by all other computers since the ENIAC. The supercomputers have been designed specifically for tasks wherein pipelining and parallelism can be exploited; those involving vector computations. Much more radical departures from the von Neumann design are currently under investigation in university laboratories. One particularly intriguing machine, Denelcor's Heterogeneous Element Processor (HEP), is already on the market.

Acquisition of a supercomputer by the VLA would represent the most direct and straightforward approach to its data processing problem since even a relatively small model could easily provide the required 60 to 75 Mflops. Furthermore, the dominance of array operations (such as FFTs) in the VLA task mix ensures that the special properties of this type of machine could be exploited effectively. Provided, of course, that the machine configuration were suitably balanced to avoid bottlenecks!

Even though supercomputer models are not numerous, there are still several choices that can be made under this option. The following is a description of one possible configuration. (It will be observed later that, in fact, this particular machine is not well suited to VLA needs, but alternative supercomputer configurations may be more appropriate.)

A Cray 1-M model 2200 machine with 2 Mwords of main memory.

An Input/Output Processor (IOP) with 8 Mwords of buffer memory.

A second IOP which might be required to control the disks.

A "front end" computer in the class of a VAX-11/780 or IBM 4031.

The disks, tapes, terminals, etc., required under all options. (See the preceding section for more details on this item.)

At full speed the Cray will execute 160 Mflops but it is a rare application that can make use of that speed except in very small bursts. After initial setup a series of 2048 point complex FFTs can be performed on this machine at the rate of 2 ms each or roughly 55 Mflops. This is an upper limit to the rating of this machine vis a vis the VLA workload.

The word size in a Cray is 64 bits. The amount of main memory quoted above is therefore equivalent, in some ways, to 16 Mbytes. This is less than half of the memory we have assumed in our analysis of the VLA processing needs. What is even worse, that analysis was based on a 4-byte word but the Cray instruction set does not include operations on units of that length. Despite the availability of utility routines to pack and unpack data, the conclusion seems inevitable that a large fraction (as much as half) of that very expensive main memory would be wasted for VLA applications. With only 8 Mbytes of usable main memory the effort to transpose a reasonable sized map could seriously compromise the machine's computing power.

The 8 Mwords (64 Mbytes) of buffer memory form a common pool of memory space accessible to both I/O processors. The main purpose of this memory is to minimize idle CPU time caused by I/O blockage but it can also be used for "memory resident" datasets i.e. as a type of fast auxiliary storage. (But it is NOT word addressable i.e. it is not an extension of main memory in the way LCS was on IBM's 360 series machines.)

Although each I/O processor is a small minicomputer programmable in its own right, Cray does NOT offer any software products for realtime data acquisition nor does it support connections to non-standard I/O devices. The basic VLA process of FILLing databases would therefore be limited to an offline mode. It appears, however, that at least one Cray user (Grumman) has written software for direct on-line data acquisition assuring us that a realtime FILLer is at least possible. The Grumman software might be available for our use.

Disks and tapes would be connected directly to the Cray itself rather than to the front end machine. At the present time the only disks available from Cray have a maximum capacity per drive of 600 Mbytes. 32 drives would be needed to amass the 20 Gbytes of storage needed by the VLA (ugh!). It is assumed that competitive pressure will force Cray to adapt its system to the much larger units now on the market. Cray does supply a block multiplexor channel to control fast (200 ips), high density (6250 bpi), IBM compatible tape drives.

User access to the Cray is mediated by a "front end" machine. Although at least one installation has written a time-sharing operating system for the Cray, the systems supplied by the vendor are batch oriented as appropriate for this type of computer. The operating system on the front end machine is typically time-sharing and user-friendly (e.g. VMS on a VAX). The user, nevertheless, must become familiar with at least some of the Cray operating system commands.

It must be remembered that this configuration has been described merely as an example of the supercomputer option. It is entirely possible that a small model of a different supercomputer would be more suitable for the VLA load. In particular, CDC's Cyber 205 contains 4-byte operations which, for typical VLA data, seems to be almost essential in order to make the most effective use of main memory. A detailed comparison of the price:performance ratios of competitive supercomputers is beyond the scope of this plan.

Option B. A Large Conventional Machine with a Super AP

While both the trade press and the non-technical press have been drawing everyone's attention to the advent of microprocessor-based systems, and while academia has been focusing on radically new computer designs, the more conventional machines have undergone significant improvements especially in the amount of power (Mips) available per dollar. According to Computerworld (June 27th, 1983 issue) a typical large mainframe in 1976 was the IBM 370/168 (which Computerworld rated at 2.5 Mips) which cost roughly $2.2M per Mips, whereas in 1982 the typical large mainframe was IBM's 3081K, rated at 14.0 Mips, which cost slightly over $0.3M per Mips.

Other areas in which conventional mainframes continue to improve are:

1- Increased amounts of directly-addressable main memory at reduced unit cost and packaged in a smaller physical volume.

2- Increased recording density and lowered cost of magnetic media used for auxiliary memory (disks and tapes).

3- Increases in data transfer rates commensurate with the improvement in peripheral devices.

There is no conventional computer on the market that can deliver, by itself, the 60 to 75 Mflops required by the VLA. Even if such a machine were made, it would be unaffordable at the price per Mips quoted above. However, an AP attached as a peripheral to a conventional computer can carry out some types of computations (vector processing) much more rapidly and far less expensively than its host. In such cases, the arithmetic load on the host machine is limited to the remaining (non-vectorizable) computations and the most important property of the host is its ability to meet I/O requirements with other high performance peripherals.

Since an AP is very much akin to a supercomputer, in the sense that they are both non-von Neumann machines specialized for vector processing, the same properties that make the VLA computing load appropriate for a supercomputer make it appropriate also for an AP. This is particularly true now that there has appeared a new generation of Super APs with speeds comparable to that of a Cray.

The combination of a moderately large conventional computer with a Super AP appears to be a reasonable alternative to the small supercomputer described as the first VLA option. The following is an example of this second option:

An IBM 3083J main processor

32 Mbytes of main processor memory

8 high speed I/O Channels

A Star Technologies ST-100 Array Processor

96 Mbytes of ST-100 memory

The disks, tapes, terminals, etc., required under all options. (See the preceding section for more details on this item.)

Competition in the area of medium-to-large-scale computers is quite active. Several alternatives to the IBM 3083J are available to use as the host for the Super AP in this example.

In this configuration the bulk of the auxiliary devices (disks and tapes) are appended to the mainframe rather than to the AP. It is assumed that all of the I/O to and from the AP must pass through the host machine. (The third option will show what might be done if this assumption is incorrect.)

The speed of the IBM 3083 is roughly 2.5 Mflops. This speed and the relatively large amount of memory with which the mainframe is equipped should be adequate for the following purposes:

1- Support a timesharing system which could respond in an adequately short time to 60 or more simultaneous active users.

2- Supply many large buffers for high-speed I/O devices.

3- Support terminals and graphic output devices.

4- Accommodate some non-arithmetic tasks (such as sorting) which might need to be performed upon the large data base.

5- Accommodate secondary processing tasks performed upon medium size data files e.g. calculation of gain files from calibration data.

6- Support program development.

The ST-100 is one of two Super APs currently on the market. Several ST-100s have already been delivered. The second Super AP has been announced by FPS, (the makers of the AP120B currently used at the VLA); but first deliveries are 6 months away, and it is rated at only 60% of the speed of the ST-100.

The ST-100 is rated at 100 Mflops. That figure is like all supercomputer and AP ratings: a theoretical upper limit which cannot be reached in practice. A simulation of the ST-100, using vendor-supplied library routines, did a 2048 x 2048 complex FFT in 7.7 seconds which works out to an achieved speed of approximately 40 Mflops. From this, and from other indications, it appears that the effective rate of the ST-100 is about 90% of a Cray even though its (unachievable) upper limit is only 63% of the (likewise unachievable) Cray's upper limit. This is impressive but a bit slower than the VLA requirement; a faster model is expected within a year or two.

A large amount of memory is needed so that multiple jobs can be staged. The software for this Super AP includes a fairly sophisticated Monitor that resides in the AP. It supports the preparation of additional tasks during execution of the current one. This speeds context switching but it requires that data and program space be available concurrently for future as well as present tasks.

The proposed memory for the ST-100 is more than the machine can accommodate at this time (32 Mbytes). It refects the expectation that higher density chips, to be introduced early next year, will raise the limit to the level needed and beyond (128 Mbytes). The memory is organized in 4-byte words accessible in units of 4, 2 or 1 bytes. The path through which data flows between main memory and cache contains a microprogrammable reformatter which can convert, split or repack data on-the-fly.

The ST-100 includes an Input/Output processor which manages data flow between external devices such as disks, host memory, special on-line devices, etc. (An IOP is an optional feature on the VLA's current APs.) Since this IOP was designed to accommodate realtime devices the online FILLer problem encountered with the Cray should be non-existent with this option.

Since the ST-100 can serve more than one host (and the software to support multiple hosts already exists) it might be convenient to exploit this feature by letting the same Super AP serve one or more AIPS systems. AIPS hardware has NOT been included in this option because two factors seem to argue against exploiting the multiple host feature in that way:

1- The ST-100, at least in its present implementation, is barely adequate for the imaging task and would probably be overloaded if it also had to support one or more AIPS systems.

2- All the I/O, hence most of the disks supporting the database, are assumed to be on the host rather than on the Super AP. Transfer of data from host to host would be awkward. (This is not a problem under the third option - see below.)

Option C. Multiple Supermini hosts for a Super Array Processor

There are two main differences between this option and the one that has just been described. The differences are:

1- The mass storage devices are directly connected to the AP rather than accessed through a host.

2- The single host machine is replaced with multiple hosts all of which share access to AP and to its mass storage files.

One possible configuration under this option would consist of:

A VAX Cluster consisting of 3 VAX-11/780s, each equipped with:
- 32 Mbytes of main memory
- 300 Mbytes of local disk storage
- CI780 Computer Interconnect Box

A DEC Star-Coupler

An HSC Intelligent Disk/Tape Server

A Star Technologies ST-100 Array Processor

128 Mbytes of ST-100 memory

The disks, tapes, terminals, etc., required under all options. (See the preceding section for more details on this item.)

This configuration is based upon the assumption that most of the I/O will flow directly between the mass storage system, the outside world (via tapes) and the AP without passing through host channels. The previous option covers the case where this cannot be assumed.

The rationale behind the division of the host functions between multiple smaller machines is based mostly upon economics. The foremost reason for specifying a fairly substantial host machine in the previous option was the need for heavy duty I/O channels. That reason disappears when I/O passes directly between the AP and outside devices. Most of the tasks that must be performed by the host (support for an interactive time sharing system, non-vector computation, program development, support for graphic devices, etc) could be performed by a distributed host system. The task of sorting relatively large data bases is an exception. For this task a single fast machine with a large amount of local memory would be far superior to many slower machines with memory divided between them.

For the multi-host approach to be feasible, all database files must be accessible to all hosts equally. DEC's Computer Interconnect hardware is a "back-end" machine which does for files what a "front-end" concentrator does for terminals. It will permit the use of a common file system, accessible to the hosts and to the ST-100 I/O Processor. As seen from the DEC viewpoint, this option is a "VAX Cluster" with one abnormal peripheral (the Super AP).

Of course the multiple host approach is only possible because the ST-100 contains the necessary hardware and the software to ensure that tasks from each can be properly handled. (These features are far beyond the reach of our first generation APs.) Since staging multiple tasks is even more important than under the second option, the ST-100 has been configured with the (anticipated) maximum memory.

Option D. Other (Rejected) Options

There are ways other than the ones described above to create a computer facility rated at 60 to 75 Mflops. One that has been suggested, at least informally, is a relatively large number of 68000-based microprocessors each equipped with a small AP.

From Table 4 it is clear that a significant fraction of the experiments conducted at the VLA entails quite small amounts of computer processing. For these experiments, relatively small microprocessor systems, of the type now being marketed for individual professional use, would be adequate.

Although small professional systems are attractive for some specific applications, there are many reasons why a solution based upon a large collection of these machines would be unsatisfactory for handling the entire predicted computer load.

1- It would eliminate the possibility of conducting some of the more demanding astronomical experiments since some tasks could not be divided into subtasks small enough to be handled by the individual component machines.

2- It could not totally replace a large central facility, which would still be required to accommodate the predicted voluminous raw input data, and to host special purpose devices (e.g. vector graphics) whose cost would limit their number.

3- It would pose heavy demands on data communication facilities in order to provide access to databases. Networks for microprocessor systems are generally based on "packet switching" which is appropriate for short messages and low volume traffic but it is inefficient for large databases.

4- It would encourage an undesirable form of hardware diversification. (At a lower level of equipment, consider the difficulty experienced by NRAO in standardizing word processors at the various sites or consider the diversity of terminals at the VLA.)

5- It would be difficult to manage, especially in the areas of software standardization and validation, and maintenance of both hardware and system software.

PART 4 - THE BASIS FOR EVALUATING THE OPTIONS

A. Capital cost.

Prices in the computer field have been notoriously volatile. Fortunately, most of the changes have been in the downward direction. The costs that will be quoted for each option are estimates at best. Some of the figures are based upon current published prices, others upon budgetary figures supplied by vendor representatives. In NO CASE are they the result of bids or requests for quotations. The nature of this plan would have made such requests inappropriate.

B. Software.

It has been a long time since one could assess the relative merits of computer hardware without considering its impact on software. The following paragraphs describe some of the ways in which software requirements should influence our assessment of hardware options.

It will be useful to identify four different types of software. Even though such a division is bound to be somewhat arbitrary, it will highlight the areas from which the most important software requirements originate. Here are the four categories:

a) Vendor-supplied operating system components
b) Owner-supplied operating system enhancements
c) Applications software
d) Casual software

We will discuss each of these categories separately.

B1. Vendor-supplied operating system components

This category includes:

a) Compilers
b) Editors
c) File management services
d) Inter-machine communication facilities
e) Utilities.

The important characteristics of vendor-supplied software are balance, power, reliability, flexibility and adaptability.

The term "balance" refers to the relative emphasis placed on efficient use of the hardware versus user convenience. The fundamental usefulness of hardware can be greatly enhanced or seriously undermined by the appropriateness of an operating system in the installation where it is used. In the scientific environment, for example, an interactive system is preferable to a batch operating system, but overemphasis on interaction can lead to degraded system performance.

The "power" of a computer operating system is again a question of appropriateness - that of the software to the hardware. The effectiveness of a given piece of circuitry, e.g. hardware stacks, will surely be undercut by an operating system that does not exploit those stacks. For example, the ability of the DEC KL-10 processor to be used as a virtual memory machine is exploited more effectively in the TOPS20 operating system than in the TOPS10 system.

The term "reliability" is used in its everyday sense. There is a tremendous difference between the attitude taken by vendors toward their responsibility with regard to hardware and with regard to software. If a machine is delivered with malfunctioning hardware it is expected that the vendor will immediately diagnose and repair the problem. In contrast, malfunctioning software is the norm. The user programming staff is expected to detect the ways in which an operating system departs from specification, report the problem to the vendor, and wait for months (if not years) for the problem to be fixed in a later release of the operating system. The hardware options must be evaluated in terms of the ability and willingness of the company supplying that hardware to support its software. Big companies usually have the ability but not the willingness - small companies the reverse.

"Flexibility" and "adaptability" are not exactly synonymous. Flexibility refers to the extent a given operating system is parameterized so that it can be tailored to the demands of a specific installation. Adaptability refers to ease with which the vendor-supplied software can be modified locally.

B2. Owner-supplied operating system enhancements

The second category of software includes:

a) Interfaces to special harware devices.
b) Frameworks under which application programs are controlled and united into larger units.
c) Data-specific I/O and utility routines.
d) Machine utilization and efficiency reports.

The cost of local system enhancements will depend upon the "cleanliness" of the operating system organization, the number of "hooks" provided in the system, the amount and reliability of system documentation and the stability of the system.

These qualities are clearly not identical to the ones listed in the previous category of software. For example, the UNIX system is highly regarded for the flexibility it provides through use of its "pipes," but it is known to be difficult to maintain (this may improve after the release of system V which will be fully supported by Bell).

Local maintenance of systems software will be influenced by the choice of hardware. For example, other things being equal, the amount of local effort required to adopt vendor-supplied operating system updates will be considerably greater in a multiple-machine installation than in a single-machine. In the multiple-machine environment it is inevitable that each machine will differ slightly from its companions in the amount of memory it contains, its appointment of peripherals, its communications links to the other machines, etc. The operating system must be tailored to each machine, care must be taken to keep the operating systems contemporary (or at least compatible) and effort not required in a single-machine installation must be expended on the software required for inter-machine communications when many machines are used in the place of one. Finally, the more a computer system consists of dispersed elements, the more complex is the software needed to support it.

B3. Applications Software

The third category of software consists of the major application programs which are developed and maintained by a cadre of professional programmers. These main line programs are the software that meets the basic objectives of the installation.

The productivity of the professional programming staff will depend heavily on what is now called "the programming environment." It includes both software tools (editors, interactive debuggers, etc) and management tools such as a program librarian.

One aspect of the various hardware options that should be given the most serious consideration is whether or not the option provides sufficient access for software development. A hardware option that requires continuous use of all components for on-line applications is sure to cost dearly in programmer productivity.

B4. Casual software

Programs developed by the general computer user form the fourth, and final, category. These one-shot programs are generally much smaller than the main-line programs described above, but even trivial programs can contribute significantly to scientific productivity by meeting immediate individual needs.

To avoid wasting professional scientific time on amateur programming activities, the legitimate need for casual programming must be satisfied by the availability of the most widely known languages, a well documented library, and a "user-friendly" operating system.

C. Maintenance.

Considering our remoteness (120 miles by road) from the nearest large city where maintenance services can readily be obtained, it is necessary for our operation to have as much maintenance as possible done either by

a) NRAO personnel or
b) a dedicated customer engineer under contract.

Obviously, only a company with a large maintenance contract can afford to dedicate an engineer to our site.

The maintenance performed by NRAO staff requires training, a spare parts inventory, and backup manpower for vacations. In this area we are also very vulnerable to personnel turnover. The cost of a maintenance contract includes all of this.

The proximity of another site with the same hardware is also an important factor. The advantages are in the sharing of common problems and in the availability of help in an emergency. The disadvantages occur when one person or team is required to service both our site and another (as is the case with our present DEC service contract). The availability of help is actually diminished.

C1. Special Device Maintenance

There will always be requirements for special purpose devices which are not provided by the CPU manufacturer. These devices need special maintenance considerations. In the on-line system, there are several devices which must be interfaced to the system by NRAO staff. The hardware (and software) maintenance of these devices has to be performed at NRAO. The maintenance of special devices, e.g. the image display devices, may be performed locally. If this is to be done, a complete set of spares and, in some cases, a completely redundant system should be purchased with the initial system and should be added to the total system cost. Where it is practical, a maintenance agreement should be obtained with the hardware vendor to supply not only maintenance in the case of failure but also regular preventative maintenance schedule and the implementation of hardware upgrades (change orders, etc.).

C2. Mixed Vendor Systems

A mixed vendor system puts a heavy burden both on the software and hardware personnel. Most computer manufacturers can provide a service contract that is acceptable especially for a large installation, but the vendors who have one or two peripherals installed at a site cannot provide nearly the same level of service.

One should seriously consider the demands placed on the local systems programming staff by hardware options that require local software expertise for hardware maintenance. At the VLA we seem to have consistently overlooked some of the hidden costs of hardware options that involve mixing machinery from a variety of vendors. Almost every failure will require that the local software experts interrupt other work in order to perform the initial (usually informal) diagnostic steps which determine

a) whether the failure is due to hardware or software
b) which vendor's hardware is responsible.

At the VLA, our most successful venture into mixed vendor hardware has been with the Century Data Systems disks. At the time of initial purchase, they were about 50% cheaper than the offering from DEC. We have also maintained them at about 40% of the cost. But the hidden costs of downtime and management of the maintenance, such as the spare parts inventory, have never been put into the equation.

Maintaining systems by NRAO personnel can cause problems with hardware similar to those mentioned in section B2. In particular with our APs, we have not upgraded our hardware (or software) to keep it current with the vendor's revision levels, which causes confusion when problems arise. We have also mixed the revision levels of AP hardware so that each of our APs behaves differently; and tailored the appointment of each AP differently so that each has a different software capability. These problems are difficult to avoid when the maintenance team is stretched to its limit.

). Operational considerations.

The operational considerations that bear upon the se omputer hardware are:

1- Standardization

Common programming language and operating systems.
Interfacing between components.
Common user interface. (Command Scanner, POPS)
Terminal Standardization.

2- Software support

Access to archived data.
Backup and Restoration of large private data bases.
Backup and protection of program files.
Updating of operating system releases.

3- Availability

Statistical load distribution vs reservation system.
Every private office vs central public clusters.
Switching network.

4- Physical plant

Air Conditioning
Floor Space
Power
Building Wiring (office terminal access)
Environment (cleanliness)
Altitude Problems

E. Communications.

Communication between computers is a rapidly burgeoning field. Local area networks such as Ethernet are available between computers at a single site and products such as DECNET, SNA, and packet-switching networks allow flexible transfer of information between remote sites over dedicated links, standard phone lines, and satellite channels. NRAO already has a simple link-up between the sites, but the impact of sophisticated communications on our eventual requirements needs to be investigated further.

The following types of communication need to be considered:

Inter-process.
Inter-user.
Inter-machine.
Inter-site.
Inter-institution.

F. Personnel.

Some of the computer hardware options available to the VLA would have a greater impact upon personnel requirements than others. This follows from the complexity of the system, its appropriateness to the VLA objectives, etc. This has already been discussed under the subject of "Software" above.

What should also be kept in mind are the very great demands placed upon VLA personnel by "cut-rate" equipment. Time and time again NRAO has fallen into the trap of buying things that required the least immediate capital expenditure only to incure a continuing debilitating drain on maintenance and programming personnel.

The technical standard of the programming staff is high, but productivity can and should be increased. Programming seems often to be driven by inspiration and is commonly done with highly irregular working hours. So radical suggestions, such as moving the center of programming operations where possible to Socorro, and providing programmers with a programming station at home (modem, terminal or personal computer, and printer) should be seriously considered.

It should also be pointed out that programmers prefer working with interesting hardware. Having such would certainly tend to attract and retain professional programming staff.

PART 5 - EVALUATION OF OPTIONS

A. SUPERCOMPUTER

a) Capital Cost.

Basic Mainframe with 2 Mwords	$ 5.5 (M)
IOP with 8 Mwords aux storage	0.7
6 6250 BPI tape drives	0.3
Second IOP (for disks)	0.2
20 Gbytes of disk storage	1.0
Front end computer	0.7
Terminals, Graphics devices, etc	0.4
Installation	0.2
Contingency	0.5
TOTAL	$ 9.5 (M)

Many of these prices are based upon budgetary figures supplied by the vendor's representative. It is important to note that the figure for disk storage is NOT in that category. The only disks available from Cray at this time are relatively small and expensive units. It would require 32 such units, at a cost of over $2 million to provide the required storage.

An installation cost has been included with this option but not with the following two because a Cray requires some extraordinary expenses. The first is for floor reinforcements; the load per unit area of a Cray is considerably greater than that of a conventional computer. The second is provision for liquid cooling. Even though the refrigerator is included in the price of the machine, the cost of installing pipes etc. would belong to the VLA. Cray also supplies power conditioning equipment (similar to the flywheel now in use, but operating at 440 Hz) which the VLA would have to install.

b) Software.

1- The Cray operating system (COS) is batch oriented and relatively simple because the tasks which justify the purchase of a Cray are generally "number crunchers" that do not require a wide variety of system services. Cray supplies a software specialist to maintain this system and to assist users in general. The cost of this individual is included in the standard maintenance contract.

From the users' point of view the operating system that is seen is that of the front end with a few embellishments to direct interaction with the Cray. The front end system may be as user-friendly as desired (eg UNIX or VMS running on a VAX.)

2- The interface between the Cray and the front-end machine is supplied by Cray but it is likely that we would want to incorporate embellishments of our own. It is also clear that programs such as FILLER will require local systems software expertise. This requirement can be minimized by exploiting the work done by others who have already met this type of need (e.g. Los Alamos, Livermore, etc.) but there will always be a residue of work to be done locally.

A supercomputer would place other demands on the senior VLA programming staff. For example, hand coding the inner parts of the most commonly used programs would probably be needed to use the Cray effectively; certainly that has been the experience of other Cray users! In addition, the peculiar characteristics of the VLA database would probably call for an equivalent "souping up" of the standard I/O routines. Both of these labor-intensive activities would be required at the VLA in order to use the supercomputer to full capacity.

3- Main-line software for a supercomputer would most likely be done in an enhanced version of FORTRAN since that seems to be the lingua franca of supercomputers. Language enhancements, intended to exploit the particular architecture of a given machine, are fairly common. This tends to promote non-standard (therefore non-transportable) programming.

The vendor-supplied FORTRAN compilers do a creditable job of vectorizing user code. Even so, programmers will need to be aware of the implications of what they write. It is much too easy to defeat the compiler's attempts at optimization by minor features that appear completely innocuous to the unaware. (Of course, the same is true of virtual memory utilization in a conventional computer!)

4- Most casual programming would be served best by the front end processor. Exceptions would involve large databases more accessible to the supercomputer than to the front end. A "vanilla flavor" subset of the supercomputer FORTRAN would undoubtedly be adequate for the vast majority of such cases.

A supercomputer is inherently a batch environment. The practice at the VLA of doing all processing in interactive mode would have to be modified. Large problems that need the supercomputer will have to be queued but those that can be served by the front end machine can continue to be done interactively. What is even more important from the point of view of the casual programmer is that program development will also vary since programs for the supercomputer must be compiled, in batch mode, on the supercomputer rather than on the interactive front end machine.

The Cray company supplies some software assistance as part of its maintenance contract but it also recommends that the user dedicate 3 to 6 senior level employees to the 4 programming categories mentioned above.

c) Maintenance.

These machines are not very common (only about 60 Cray computers have been built). This means that hardware expertise is scarce, and parts are both scarce and expensive. A contract for vendor-supplied maintenance, the only realistic course, would certainly cost several times what the VLA pays for maintenance at the present. The quality of such maintenance will undoubtedly be relatively good because of the proximity to the VLA of other supercomputer users (e.g. Los Alamos and Sandia Labs).

The estimated annual cost of maintenance for this proposed configuration is $700k.

Special hardware, such as graphics devices, are generally not available from supercomputer vendors so that a mixed vendor shop is almost inevitable.

d) Operation.

It is almost inconceivable that a $10 million supercomputer would be left unattended, open to pokes, prods, punches and pounding by untrained and impatient users. Since the VLA presently runs such an "Open Shop", protecting this equipment would require the establishment of 4-5 new computer operator positions.

e) Communications.

All communication with supercomputers is traditionally done through the front end computer. This has its attendant problems (see Part 4 section B2).

There may be interest in providing graphical displays at remote facilities as well as locally. Graphics displays on a supercomputer are usually handled by the front-end processor. A typical device will display images of 512 pixels on a side with a resolution of 8 bits. The display device is likely to contain local intelligence, i.e. it will be able to select a subset of a larger image. The data to be transfered from the Cray, through the front-end machine to the destination device will typically be 4 Mbytes (1024 by 1024 4-byte pixels). Locally this transmission will only take seconds. If the graphics terminal is at the end of a 1200 baud phone line then this would take about 15 minutes. For some applications this would be acceptable.

f) Personnel.

A supercomputer is a magnet for highly competent computer personnel (we have lost one member of the computer division to a Cray!).

As noted above, staff increases of 3 to 6 programmers and 4 to 5 operators are probably required.

B. CONVENTIONAL MAINFRAME WITH SUPER ARRAY PROCESSOR

a) Capital Cost.

IBM 3083J with 32 Mbytes memory, 8 Channels	$ 3.0 (M)
ST-100 Array Processor	0.3
Extra memory for the ST-100 (to 32 Mbytes)	0.8
Extra memory for the ST-100 (to 64 Mbytes)	0.8
Extra memory for the ST-100 (to 96 Mbytes)	0.8
6 6250 BPI tape drives	0.3
20 Gbytes of disk storage	1.0
Disk controller	0.2
Communications	0.4
Terminals, Graphics devices, etc	0.4
Contingency	0.5
TOTAL	$ 8.5 (M)

The prices for IBM products shown above were obtained mostly from DATAPRO fact sheets. From the same source it was noted that prices for competitive machines such as Amdahl 470 V/8, NAS AS/9050, CDC 170/875, and Univac 1100/91 are very similar.

Prices for Star Technologies products were obtained from company representatives. The memory shown for the ST-100 is more than the current maximum but the proposed amount should be within the new limits expected when higher density chips are introduced. At that time the price per byte should decrease considerably. Despite this expectation of lower prices, all the ST-100 memory is shown at current prices because the future price is unknown and because any savings would probably be consumed by the need for more cache memory (not a separate item above) and/or by the need for a faster model which might be introduced at approximately the same time.

b) Software.

1- Most mainframes that would be considered for this use are "mature" products, i.e. they, or their predecessors, have been available for many years. This sounds like an advantage but it can be disadvantageous to the extent that the machines and their software have been made upward compatible from very early versions and therefore they retain historic inefficiencies.

Medium to large scale conventional computers are very much "general purpose" machines. The vendors, very much aware of this fact and selling in a very competitive market, generally supply operating systems that serve all possible buyers. As a result, the operating systems tend to be many times larger than they need to be. Even worse, they tend to be overly complex and inefficient in any one application: they are not tailored to a given market as, for example, supercomputers are tailored to "number crunching". They contain complicated accounting systems and security measures (e.g. encryption) most of which is wasted and inefficient in our type of operation.

The system software required to interface the Super AP with the mainframe is supplied by the AP vendor. The same source supplies the compiler and other programming tools (simulator, debugger, etc) that are necessary to create new Super AP modules. As a result, the quality of this type of software is independent of the vendor selected for the mainframe and must be evaluated separately.

2- The general purpose file systems of the mainframe may be inefficient for the very large scientific databases commonly used at the VLA. This may call for local development of a special I/O package. The overwhelming opacity of IBM's operating systems command language would probably require a large amount of systems programming services to assist the casual user. Some competing operating systems are much better; CDC's NOS system, for example, is reputed to be quite user-friendly.

3- It is expected that most application programs would be able to use a standard library of AP routines. Of the remaining programs most would be able to create the necessary AP modules through the FORTRAN compiler supplied by the AP vendor.

It is even more difficult to hand microcode the Super APs than it is to write microcode for the VLA APs (the instruction length is greater because the amount of possible concurrency is greater). Fortunately such activity should not be necessary at the VLA except on very rare occasions. Currently available AP and Super AP compilers for high level languages (mostly FORTRAN subsets) are capable of generating remarkably efficient microcode even from quite ordinary (unoptimized) input. For example, automatic recognition of possible concurrency in DO loops is the norm. Bad habit and the neglect of available software tools are the only reasons why hand microcoding is still relatively common at the VLA.

4- Most casual programming probably would use standard AP library modules if the AP were used at all.

c) Maintenance.

Large mainframes tend to have a richer variety of add-on equipment than all other options. This reduces the necessity of a mixed vendor system. Add-on equipment for such machines tends to be of higher quality, and therefore less of a maintenance problem, than for minicomputers.

d) Operations.

As explained in connection with the supercomputer option, a set of 4 to 5 computer operators will be required to run this equipment. In addition to the arguments already presented there is, in this case, the additional factor that IBM operating systems WILL NOT FUNCTION without operator intervention.

e) Communications.

The question of communications and terminals could be the source of an expense that is only partly reflected in the $0.4 million item labelled "Communications" in the budget shown above. The problem is that the computer world is generally divided into two parts: IBM and all others. For example, all of the terminals owned by the VLA operate in asynchronous mode whereas IBM-compatible terminals operate in synchronous mode. Likewise, the de facto standard AIPS system is based on a VAX whose inter-machine protocol differs from IBM's. A translation package is available but at a cost. A different choice of host machine might avoid much of this cost.

f) Personnel.

Other things being equal, the larger the operating system, the greater is the load imposed on the user's systems programming staff. With a poor choice of operating system, this option could easily place great demands upon the systems programming staff: greater, in fact, than those imposed by the combination of a supercomputer (with a relatively simple operating system) and a front-end VAX (with a well supported system such as VMS).

To deal with such a complicated system requires a team of systems analysts (not just one Al Braun!).

C. MULTIPLE SUPERMINI HOSTS WITH SUPER ARRAY PROCESSOR

a) Capital Cost.

VAX-11/780 front end (3)	0.8
(Each with 2 Mbytes of memory, 300 Mbytes of disk, and a CI 780 Computer Interconnect Box)	
100 Mbytes of extra VAX memory	0.3
ST-100 Array Processor	0.3
Extra memory for the ST-100 (to 32 Mbytes)	0.8
Extra memory for the ST-100 (to 64 Mbytes)	0.8
Extra memory for the ST-100 (to 96 Mbytes)	0.8
Extra memory for the ST-100 (to 128 Mbytes)	0.8
Swapping disk for the ST-100	0.2
6 6250 BPI tape drives	0.3
20 Gbytes of Disk storage (with controller)	1.0
Terminals, Graphics devices, etc	0.4
Contingency	0.5
TOTAL	$ 7.0 (M)

Comments about ST-100 prices that were made during the discussion of the previous option apply here too.

b) Software.

c) Maintenance.

One advantage of the multiple host approach is that a failure of a single host does not incapacitate the entire facility. The counter argument is that there are then more components subject to failure.

d) Operations.

The reasons for having a set of computer operators are less compelling under this option than under the first two.

e) Communications.

The VLA experience with DECNET has been positive enough (outside of the disastrous TOPS-10 situation) to justify considerable confidence in its future adequacy for VLA data communications tasks.

f) Personnel.

This option would seem to require the least increase in programming staff.

PART 6 - CONCLUSIONS

During the past five years the scientific value of the VLA has been increased dramatically through the invention of computer-intensive techniques which increased the dynamic range of the instrument by more than 20 dB. This fortunate development is directly responsible for the fact that during the same period the VLA has been forced to play "catch up" with its computer power.

The present VLA computer facility consists of a relatively complex network of minicomputers and array processors. It is the result of an attempt to assemble, from a collection of disparate machines and special devices, a "computing engine" tailored to VLA needs. Only a relatively small part of the aggregate processor power in that network is actually useable. The problem is excess complexity and imbalances between processor power on the one hand and memory and I/O bandwidth on the other.

The serious lag of VLA computer facilities behind the need for the same, has prompted a committee of astronomers, led by Ron Ekers, to prepare a list of areas in which research at the VLA can be anticipated during the next decade. For each area the committee has estimated the level of activity and the ensueing computer demands. The report of Ekers et al. has been used as the basis for this computer plan.

The anticipated computer demands at the VLA are at least one order of magnitude greater than our current computer capacity. The general characteristics of the necessary computer facility are: processor power ranging from 60 to 75 addressable by the processor(s), 20 Gbytes of on-line disk storage, and I/O channels (and peripherals) capable of continuous achieved average transmission of 1 Mbyte/second.

Fortunately the nature of computing at the VLA will continue to be such that vector computations (primarily Fourier transforms) will form a large percentage of the total. This means that the VLA needs might be satisfied by one of several options which include: a supercomputer, a very large and fast array processor, or some other form of non-conventional machine (one with non-von Neumann architecture). Rough estimates based on current prices indicate that the cost of each option is comparable and is in the neighborhood of $10 million. A final choice would depend upon prices at the time of purchase and upon a technical evaluation of each option far more detailed than appropriate for this plan.

The criteria upon which the options should be judged must include operational factors. It seems almost inevitable that, whatever the final choice, maintenance of a large computer facility by the VLA will require, in addition to an initial capital outlay, on-going expenditures in excess of current levels.

Appendix A

An Assessment of the FPS AP120B Array Processor

The important role played by the half dozen APs in use at the VLA is fairly obvious but is not easy to quantify. The unconventional design of these devices does not permit a direct comparison with ordinary computers. Furthermore, direct use of speed values would be meaningless because, in practice, that speed can be sustained only for short periods. What we needed was an estimate of the effective "equivalence" of an AP to a conventional computer so that we could see how well or how poorly we were using the APs and so that we could estimate what would be needed to replace them.

In Table 2 each AP is listed as being potentially capable of delivering the same amount of computation as a conventional computer running at 6 Mips. This figure was reached by three routes.

The most direct basis is the AP clock cycle: 166nsec. At one instruction per cycle this would be exactly 6 Mips. That this figure coincides with the result obtained by other means must be considered purely coincidental because the AP, as a result of its parallelism, can carry out several instructions per cycle. In fact, the AP can actually perform 12 million floating point operations per second (Mflops) which is far beyond the capability of a typical 6 Mips machine.

The second basis for the 6 Mips equivalency figure was a comparison of the performance of an AP with that of a VAX (which is rated at roughly 1 Mips). The basis for comparison was a 1024-point complex Fast Fourier Transform which can be computed in 150ms on a VAX or 6ms on the AP connected to the VAX. This ratio would indicate that the AP is equivalent to 25 VAXs but, unfortunately, this AP speed can be achieved only in relatively short bursts interspersed with considerable start-and-stop overhead. In the case of the 1024-point FFT, the AP must be triggered by at least 3 VAX system commands (QIOs), each consuming roughly 2ms. Furthermore 24ms are required to transfer the data and answers between the VAX and the AP. This overhead is in addition to time that might be required to read data from auxiliary memory, a delay that would be affect the outcome equally wherever the FFT was computed. Since the AP is so fast, only 6 of the 24ms of I/O time can be overlapped by computation (of other FFTs). So the ratio of 150:6 should really be 150:(6+3*2+24-6) or roughly 5.

The third basis for assigning 6 Mips as the equivalent computer power of an AP was an experiment conducted by Tim Cornwell using a special measuring device built by Phil Dooley. During one 90 minute interval Tim ran a set of AIPS tasks especially selected to keep the AP as busy as possible. Measurements made with Phil's box over that period showed that the AP was running only 25% of the time. Using the same figure of 25 for the speed ratio of an AP and a VAX, we deduced from this test that the effective equivalence of an AP was 25 x .25 = (app.) 6 VAXs.

Appendix B

Timing for the Canonical Task on a VAX-11/780 with AP120B

It is instructive to calculate the time that would be required to perform the canonical imaging task on one of the machines currently being used for AIPS at the VLA. (A combination of VAX-11/780 plus AP120B array processor.) According to Ed Fomalont, the performance of this combination can be determined from the set of equations shown in Table A1 wherein D is twice the number of complex visibilities in millions, and M is the map size divided by 1024. The result of each equation is minutes of wall-clock time assuming no interference from other concurrent tasks.

Since the AIPS system needs to order the visibilities there is an entry in Table A1 for that process. The time for sorting, which is mostly I/O time, obviously has not been included in the total I/O time since sorting was not included in any of the earlier discussions of the canonical task. It is worth noting, however, just how significant that process can impact any system that needs it. The time required for gridding has been set, arbitrarily, to the time for FFTs because the formulae quoted does not seem to match measured values.

The various AIPS modules shown in Table A1 must run sequentially i.e. for a given user the I/O cannot be overlapped with the processing. The total throughput time for the canonical task would therefore add up to 680 hours or roughly 60 times longer than the observing time.

The time required to run the canonical task on the hypothetical 5 Mflops machine was estimated to be 53 hours CPU time. Since the same task requires 680 hours on an AIPS system we conclude that the effective power of the VAX/AP combination under AIPS is only 0.38 Mflops. In Table 2 the theoretical capacity of an AIPS system was listed as 7 Mips or roughly 2 Mflops. This order of magnitude discrepancy is largely due to the memory limitation (especially the AP memory limit) under which this version of AIPS operates.

Table A1

Timing for the Canonical Task on a VAX-11/780 with AP120B

(Based on Equations derived by E. Fomalant)

AIPS Module	Function	Formula (minutes)	Total I/O Time	Total Calc Time
UVLOD	UV data Tape to Disk	$5.5 * D^{1.0}$	73 hrs	
UVSRT	Sort the UV data	$50.0 * D^{1.1}$	(667 hrs)	
UVMAP	Read Visibilities	$4.0 * D^{1.0}$	53 hrs	
	Grid (one map)	$1.8 * D^{0.5} * M^{0.3}$		37 hrs
	FFT (one map)	$2.0 * M^{2.1}$		68 hrs
	Write one map	$2.8 * M^{2.1}$	51 hrs	
APCLN	Initialize CLEAN	$3.0 * M^{1.5}$	36 hrs	
	Finding 1000 Components	$2.0 * M^{2.0}$		136 hrs
	Major CLEAN cycle	$3.2 * M^{1.5}$		191 hrs
	Writing the map	$2.0 * M^{2.0}$	34 hrs	
			--------	-------
			247 hrs	432 hrs

In the equations quoted above, D is the size of the database (number of visibility values) divided by 500,000 and M is the mapsize (pixels per side) divided by 1024.

An independent confirmation of the power of the VAX/AP combination under AIPS comes from measurements of the time consumed by a maximum entropy (MEM) program written by Tim Cornwell. Each iteration through the MEM routine is equivalent to 2 2-dimensional FFTs. The amount of CPU time required for each iteration on a map of given size and the implied system preformance are:

Map size (pixels)	CPU time per iteration (minutes)	Implicit Mflops
128	.3	.03
256	.5	.09
512	2	.10
1024	5	.18
2048	20	.19

Note that wall clock time was roughly 3 times larger than the CPU time recorded in these measurements.

Appendix C

CPU resources required for I/O

It is everyday experience that a heavy I/O load tends to degrade the performance of a computer even when the I/O is conducted in Direct Memory Access (DMA) mode i.e. is handled mostly by the special hardware in computer channels that are independent of the CPU. The reason for this is fairly well understood: the data transfer process must be orchestrated by the operating system which runs in the CPU.

The specification of the hypothetic computer does not include the characteristics of its operating system. Some indication of the load imposed on the CPU by the I/O calculated for the canonical task can be gained by considering what would happen if that load were to be handled by the VMS system on a VAX 11/780.

Dr. Richard Wrenn of Washington University recently presented to a DECUS meeting a paper entitled "Disk System Performance in VAX/VMS" wherein he reports the amount of CPU time consumed by VMS each time it initiates an I/O transfer to disk. (Technically called the "QIO overhead".) The value reported is 9 milliseconds.

Rewriting a previous table, the I/O for the canonical task is:

I/O Operation	Gbytes
Original Input	2.2
Output to Buffer Disk	2.2
First Input to Mapping	2.2
Second Input to Mapping	17.6
Original Map and Beam Output	4.5
Rereading Output from Disk	4.5
Writing Maps & Beams to Tape	4.5

	37.7

If we assume that we need only one I/O initiation for each block of data transfered and that each block contains 10 kbytes, then the CPU time that would be consumed for 12 hours of I/O operations is:

$$\frac{38{,}000{,}000{,}000 \text{ bytes}}{10{,}000 \text{ bytes/block}} \times \frac{.009 \text{ seconds/block}}{3600 \text{ seconds/hour}} = 9.2 \text{ hours}$$

which says, in effect, that the CPU would be almost totally paralyzed by the I/O operations!

Appendix D

Some Anticipated Image Reconstruction Algorithms

Tim Cornwell

The following is intended as a partial list of image construction algorithms that will, or could be used for future VLA processing. The cost of the algorithms will be given in terms of 2-D FFTs. File reading times and other calculations will be ignored but should not represent a major addition to the computing needs.

Algorithm	Purpose	Cost in 2-D FFTs	Comments
CLEAN	Deconvolution	2 + 2 * (number of major cycles). Number depends on structure and ranges between 4 and 200. Average 12 FFTs for continuum, 10 for spectral line.	Widely used. Maximum Entropy (MEM) may be competitive for very extended sources.
Smoothness Stabilized CLEAN	Deconvolution	2 + 2 * (number of major cycles). As above.	Same cost as CLEAN.
Maximum Entropy and Variants	Deconvolution	Several iterations 100 to 400 FFTs Average 120 for continuum.	For extended sources about as good as CLEAN.
Gerchberg-Saxton Algorithm	Deconvolution	Few + 2*(number of iterations). Typically 100 to 400 FFTs Average 150 for continuum.	Can do polarization mapping. Otherwise MEM is probably better. Can interpolate short spacings.
Self-calibrations	Atmosphere Correction	Number of iterations * Slow FT * number of components. Typically 2-3 iterations, and 500 components. (Could be improved to number of iterations * 2 FFTs)	Very useful for observations not S/N limited. At the moment we do a slow transform but could do the inverse of UVMAP.
Broadband mapping of sources with varying spectral index	Improved u-v Coverage, esp. for VLBA.	Costs several deconvolutions.	Speculative. May not be widely used for the VLA.

Monte Carlo estimates of CLEAN errors	Image reliability estimate	Several CLEAN iterations	May be useful for complicated sources
3-D mapping and CLEANing	Maps from non-coplanar aperture	Need 3-D FFTs with about 10 deep in w-plane. FFTs larger.	Needed for wide field mapping.
Snapshot Summation	Many large maps at short interval	Need about 10 maps per 12 hours. Avoids 3-D problem.	Needed for wide field mapping or non-circularity problems.
Mozaicing	Many small maps to cover a large field	May be cheaper than one large map.	Needed for wide field mapping. Better if entire field is sparse.

This short list is not complete but it should be noted that "MEM and variants" covers a multitude of sins. I think that most advances in image construction will be of this general form. Furthermore, for images in which more than about half of the field of view is filled with emission, MEM etc. should be superior to CLEAN in results and efficiency. One interesting possibility is that of adapting the philosophy of Barry Clark's CLEAN to MEM. The gains in speed may be up to a factor of ten. Substantial effort is necessary to investigate this possibility. Also this type of task is well suited to the rapid FFT capability of the pipeline.

Note also that self-calibration can be increased in speed by:

1. Do inverse transform by FFT and regridding.
2. Avoid one sort by using gain table on XY sorted data.

A lot of programming effort is required for these changes.