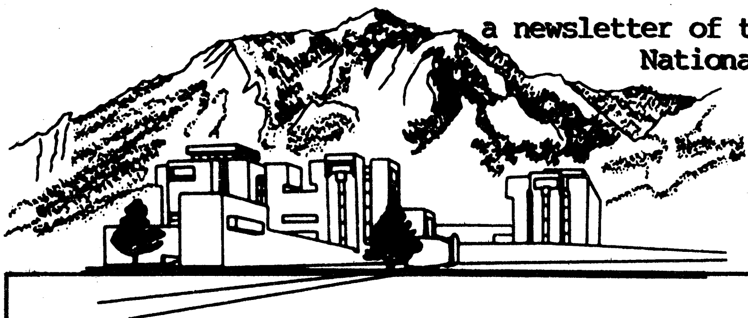
Volume 5, Number 9
September 1, 1984

The Record



a newsletter of the Scientific Computing Division,
National Center for Atmospheric Research

contents

SERVICES DIRECTORY

(303) 497-1000 - FTS prefix 320

		ext.	room
NEW USER INFORMATION			
Computing Resource Applications	Cicely Ridley	1211	119
	John Adams	1213	118
Project & User Number Assignment	Rosemary Mitchell	1235	5
Document & Manual Distribution	Sal Farfan	1292	17g
REMOTE USER INFORMATION			
Data Communications (RJE)	Don Morris	1282	11d
	Marie Working	1250	31f
RJE Password Assignment	Rosemary Mitchell	1235	5
Visitor Information	Kathy Lucero	1231	6a
OPERATIONAL INFORMATION			
Computer Operations	Oper. Supervisor	1200	29
Graphics Operations	Andy Robertson	1241/42	24c
1/2" Tape Librarian	Sue Long	1245	24f
TBM Tape Librarian	Mary Trembour	1232	5
Software Distribution/Output Mailing	Sue Long	1245	24f

CONSULTING OFFICE

The Consulting Office is closed every Wednesday from 13:30-14:30 for staff meetings. Consultants may be reached by calling (303) 497-1278. Messages may also be sent to the CONSULT1 virtual machine on either the IBM 4341 (IO) or the IBM 4341 (IA).

	08:30-12:30	13:00-17:00
MONDAY	Erich Thanhardt/Barb Horner	Harsh Anand Passi
TUESDAY	Ann Cowley	Dan Anderson/Harsh Anand Passi
WEDNESDAY	Erich Thanhardt	Dave Kennison/Ken Hansen
THURSDAY	Barb Horner	Dan Anderson
FRIDAY	Ken Hansen	Dave Kennison

SCHEDULE OF MACHINE UNAVAILABILITY

All machines may be down from 07:00 until 08:30 daily for Systems Checkout. In addition, some machines will be down for Preventive Maintenance as follows: CRAY,CA, 06:00-08:00 (T Th); CRAY,C1, 06:00-08:00 (M W); TBM, 06:00-07:00 (daily); MODCOMP, 08:00-12:00 (1st Monday of month).

The Record is published monthly by the Scientific Computing Division of the National Center for Atmospheric Research. NCAR is operated by the University Corporation for Atmospheric Research and is sponsored by the National Science Foundation. Reference to a company or product name does not imply approval or recommendation of the product to the exclusion of others. Robert Nicol, Editor; Ann Cowley, Head, Information Services; Astrik Deirmendjian, Trouble/Design Reports; Ken Walton, Summary of NCAR Computer Use; Mary Bartels, Computer Resources Allocated.

Special Feature: The CRAY I/O Report

The Record this month includes a special feature: "The Report on Efficient I/O on the CRAY-1 Computers at NCAR", by Bonnie Gacnik, which was originally published in August, 1984. Because of the importance of I/O (input and output) to users with specific limits on computing resources, the staff of The Record felt it appropriate that this report be made widely available immediately.

The report contains results of test runs of various CRAY-1 I/O routines available at NCAR, as well as some general information about the routines themselves. It should be noted that the test results are intended as a basis for comparison and not necessarily as specific recommendations for I/O use. I/O is not a topic which lends itself to absolute generalizations, since the constantly changing computing conditions and the unique aspects of each program directly affect the efficiency of the I/O process. However, if the report is approached as a general study rather than a definitive set of rules or recommendations, it should provide users with some general guidelines for improving I/O efficiency and reducing the percentage of computing allocations used in I/O processes.

Due to size constraints, "The Report on Efficient I/O on the CRAY-1 Computers at NCAR" is the only feature article in this edition of The Record, and has been reprinted in the original form; it may be found at the back of this edition. Items "For The Record" have been included on separate pages so that the report may be detached if desired. Please feel free to inform us of any comments regarding the report or other items via the Your Turn page, which which is included with each edition.

ZOT - The October Mass Store Purge will occur October 6, 1984

The October purge of datasets (VSNs) from the TBM will take place on October 6, 1984, and will affect VSNs not accessed since July 1, 1984. The subsequent purge of datasets will occur on November 3, 1984, and will affect VSNs not accessed since August 1, 1984.

SCD requests all TBM users to help improve the efficiency of the system by refraining from updating unneeded VSNs. Dedicated tape users are also urged to consider the continued need for old "dedicated" VSNs, and release all obsolete or unnecessary volumes. Mary Trembour (303) 497-1232 will gladly accept lists of unneeded VSNs to be deleted from Dedicated and regular reels.

Summary of Daily Bulletin Items

CRAY-1 COMPUTERS

July 24, 1984
CRAY-1 COMPUTERS: System repairs to both CRAY-1 computers were completed at approximately 19:00 yesterday, July 23, 1984. The repairs eliminated a longstanding problem which caused jobs to be incorrectly held in a S-EVENT (system event) condition.

July 26, 1984
COS 1.12 and CFT 1.11 have been installed on the CRAY,C1 computer as of Monday, July 23, 1984. COS 1.12 and CFT 1.11 were installed on the CRAY,CA computer on Monday, July 9.

The MXM, MXMA, and ISAMAX routines were not working properly since installation of the new release. A correction effort was made on Wednesday, July 18, 1984, but software problems continued. The routines have again been corrected. Please contact the Consulting Office in the event of any futher problems with these routines.

Certain library routines are now using more memory than previously, since system software is now required to support both the old and new calling sequences. Jobs that were approaching the memory limitations with COS 1.11 may exceed memory limits with COS 1.12. Users who receive a MORE MEMORY REQUESTED THAN AVAILABLE message should call the Consulting Office (ext 1278) for suggestions on running jobs on the new system.

July 30, 1984
Several bugs have been found in the CRAY-1 system software, one of which has been affecting system performance. All known bugs in COS 1.12 have been corrected as of this morning, Monday, July 30. Users who encounter problems with the COS 1.12 software should contact the Consulting Office (ext 1278).

August 13, 1984
CRAY-1 COMPUTERS: Alterations have been made to the network, tape, and MODCOMP sections of code on both CRAY-1 computers in order to accomodate memory management enhancements. Users should not be affected.

August 20, 1984
CRAY-1 COMPUTERS: The operating systems on both CRAY-1 computers were modified this morning to eliminate system halts. Users should not be affected.

IBM 4341 COMPUTERS

August 20, 1984
IBM 4341 COMPUTERS: Installation of release 3.1 of VM/SP on both IBM 4341 computers was completed this morning, August 20, 1984. The new release affects CP software only; CMS software has not been altered, so any changes should be user transparent.

PACX

August 20, 1984
NOTICE TO PACX USERS: On Monday, August 27, 1984, the parity parameters on several SCD systems will be altered. Users accessing computing systems through the PACX will need to change the parity setting on all terminals. Details on the correct settings for various terminals will be appearing in the Daily Bulletin and in the forthcoming Staff Notes on August 24.

SOFTWARE

July 25, 1984
LIBRARY SOFTWARE CHANGE: The NAG Mark 9 library (old edition) is no longer available due to operating system changes on both the CRAY-1 machines. A new object library for this old version cannot be generated using the new software since the FORTRAN source code is not accessible at NCAR. Users are advised to use the default NAG Mark 10 library. To access the default CRAY-1 object library for NAG Mark 10, include NAG in the list of parameter values for the LIB keyword on the LDR statement. For example:

```
LDR,LIB=NAG.
```

Any problems or questions should be directed to the Consulting Office (ext 1278).

August 1, 1984
GRAPHICS SOFTWARE: The EZMAP and PLOT.8.8 software packages will be updated on Thursday, August 9 to aid in statistical analysis of package use. The change will be transparent to all users who employ the binary versions of these routines from $NCARLB. However, some users who obtain and modify the source code may be affected. Any modifications between lines 2632 and 2638 in EZMAP will cause errors; any modifications between lines 2849 and 2851 in PLOT.8.8 will also cause errors. Any other modifications to the source code will not cause problems. Both files will have the same number of lines as the previous versions.

TROUBLE/DESIGN REPORT
JULY 1984

CRAY No. 139

TROUBLE: Bad code generated for reassignment of loop index in called subroutine.

EXAMPLE:

```
   DO 10 I=1,3
   PRINT *,I          ! Prints loop index. (1,2 OR 3)
   CALL SUB1 (I)      ! SUB1 reassigns I=123456
   PRINT *,I          ! Prints loop index. (1,2 OR 3)
   CALL SUB2 (I)      ! SUB2 prints I as =123456
   PRINT *,I          ! Prints loop index. (1,2 OR 3)
10 CONTINUE
   PRINT *,I          ! Prints I as 4.
```

COMMENTS: The CFT manual states that the DO index (I) generates a value within the loop without affecting the loop iteration. In this case, the value of the loop index is kept in two T registers (T00 for the index itself, T01 for the value of the index which is passed to the subroutines). Because of this, the value of I printed within the calling loop is the expected value of the loop index, while the value of I printed from within the subroutines is the value assigned in the subroutine.

TEMPORARY SOLUTION: Do not change the value of the DO-loop index in a called subroutine. This problem can be avoided by assigning I to a temporary variable, passing the temporary variable as argument in the subroutine call and then reassigning Do-loop index to the temporary variable as follows:

Example:

```
ITEMP=I
CALL SUB1 (ITEMP)
I=ITEMP
PRINT *,I
CALL SUB2 (ITEMP)
I=ITEMP
```

ORIGINATOR: John Vorwald (Harsh Anand Passi)

TROUBLE/DESIGN REPORT
July 1984

CRAY No. 140

TROUBLE: CFT 1.11 will pad character strings differently depending on whether or not the variable is defined before program execution.

EXAMPLE:

```
PROGRAM MULL
CHARACTER*10 A,B
DATA A/´SEA´/
READ (5,*) B
IF (A.NE.B) WRITE (6,*) ´BAD CONDITIONAL´
STOP
END
Input card:   ´SEA´
```

COMMENTS: CFT 1.11 will write the ´BAD CONDITIONAL´ message.

TEMPORARY SOLUTION: Compare substrings which are the length of the input data.

Example:

```
PROGRAM MULL
CHARACTER*10 A,B
DATA A/´SEA´/
READ (5,*) B
IF (A(1:3).NE.B(1:3)) WRITE (6,*) ´BAD CONDITIONAL´
STOP
END
Input card:   ´SEA´
```

CFT 1.11 will not write the ´BAD CONDITIONAL´ message.

ORIGINATOR: Peter Mullen (Michael Pernice)

TROUBLE/DESIGN REPORT
July 1984

CRAY No. 141

TROUBLE: CFT 1.11 does not allow a valid character assignment when the same variable name appears on both sides of a character assignment statement, even though the substrings do not overlap.

EXAMPLE:

```
PROGRAM SUBSTR
CHARACTER*3 CH
DATA CH/'ABC'/
CH(3:3) = CH(1:1)
STOP
END
```

COMMENTS: CFT 1.11 flags CH(3:3) = CH(1:1) as an illegal statement.

TEMPORARY SOLUTION: Use a temporary character variable to move the substring:

Example:

```
PROGRAM SUBSTR
CHARACTER*3 CH,CHTMP
DATA CH/'ABC'/
CHTMP = CH(1:1)
CH(3:3) = CHTMP
STOP
END
```

ORIGINATOR: Fred Clare (Michael Pernice)

COMPUTER RESOURCES ALLOCATED IN JULY 1984

SCIENTIST	PROJECT TITLE	GAU Request	GAU Alloc.
William Smyth Atmospheric and Environmental Research Inc./ Michael McElroy Harvard University	Extended nature and planetary interactions of satellite atmospheres	10.0	10.0
Bernard Durney National Solar Observatory Sunspot, NM	Solar differential rotation	4.67	4.67
Phil Austin/ Marcia Baker Univ. of Washington	Entrainment and mixing in layer clouds	5.33	5.33
Duane Stevens Colorado St. Univ.	Synoptic-scale tropical disturbances	10.0	10.0
James Carton Harvard University	Island eddy current simulations	10.0	10.0
D. G. Hummer/ D. C. Abbott JILA Univ. of Colorado	Atmospheres and winds of hot stars	10.0	10.0
Kuo-Nan Liou Univ. of Utah	Development of a 2-D climate model with mean circulation feedbacks	10.0	10.0
Stella Coakley Univ. of Denver	Climatic variation and plant disease epidemics	6.0	6.0
Philip Brown, Jr. CEM	Analysis of coalescence/ breakup equation	8.0	8.0

SCIENTIST	PROJECT TITLE	GAU Request	GAU Alloc.
Jon Ahlquist Florida St. Univ.	Large scale monsoon fluctuations	10.0	10.0
John Molinari SUNY	Convective parameterization in mesoscale models	10.0	10.0
Allan Robinson/ Leonard Walstad Harvard University	Primitive equation assimilation of the POLYMODE data	10.0	10.0
Martin Maxey Brown Univ.	Settling of aerosol particles in homogeneous turbulence	10.0	10.0
Myrl Hendershott Scripps Institute of Oceanography	Normal modes of the Pacific-observations and theory	10.0	10.0

Note: A request may be supported at a lower level than requested because:

a. It exceeds the five-hour limit above which Panel review is required; or

b. Reviewers consider the amount of time requested to be excessive.

Summary of NCAR Computer Use for July 1984

CRAY,CA COMPUTER

	Jul Total	Jul Day Avg.	FISCAL YTD Total	FISCAL YTD Day Avg.
Clock Hours in the Month	744.00	24.000	7320.00	24.000
less Scheduled PM	14.12	0.455	162.30	0.532
less Hardware Downtime	24.72	0.797	72.66	0.238
less Software Downtime	2.25	0.073	18.93	0.062
less Environmental Downtime	3.40	0.110	82.33	0.270
less Operations Use	5.80	0.187	7.15	0.023
less Other Causes	1.38	0.045	8.52	0.028
Clock Hours Up	692.33	22.333	6968.11	22.846
less Systems Checkout	15.10	0.487	66.39	0.218
Clock Hours Avail. to Users	677.23	21.846	6901.72	22.629
less Idle Time	20.98	0.677	75.62	0.248
Clock Hours in Use	656.25	21.169	6826.10	22.381
% Available Hours Used	96.90 %		98.90 %	

CRAY,C1 COMPUTER

	Jul Total	Jul Day Avg.	FISCAL YTD Total	FISCAL YTD Day Avg.
Clock Hours in the Month	744.00	24.000	7320.00	24.000
less Scheduled PM	17.80	0.574	161.74	0.530
less Hardware Downtime	18.50	0.597	128.70	0.422
less Software Downtime	1.72	0.055	20.52	0.067
less Environmental Downtime	2.93	0.095	75.22	0.247
less Operations Use	1.87	0.060	9.06	0.030
less Other Causes	1.35	0.044	16.93	0.056
Clock Hours Up	699.83	22.575	6907.83	22.649
less Systems Checkout	7.20	0.232	49.59	0.163
Clock Hours Avail. to Users	692.63	22.343	6858.24	22.486
less Idle Time	11.85	0.382	81.24	0.266
Clock Hours in Use	680.78	21.961	6777.00	22.220
% Available Hours Used	98.29 %		98.82 %	

REPORT ON EFFICIENT I/O ON THE CRAY-1 COMPUTERS AT NCAR

by Bonnie Gacnik
Scientific Computing Division
National Center for Atmospheric Research

August 1984

REPORT ON EFFICIENT I/O ON THE CRAY-1 COMPUTERS AT NCAR

by Bonnie Gacnik
Scientific Computing Division

INTRODUCTION

Every computer user at one time or another must address the problem of moving data between the Central Processing Unit (CPU) and a peripheral device. This process is referred to as Input/Output (I/O) and is a source of interest if not concern to many computer users. Since computer resources are always in demand, using them can be expensive, particularly if I/O processes are mismanaged or poorly understood. One of the ways to use these resources wisely and reduce the cost of I/O activity, is to perform efficient I/O.

There are a number of software and hardware constraints on the CRAY-1 computers at NCAR which make I/O activity expensive. Because of the importance of this activity for users and the costs associated with some types of I/O, this study was undertaken to look at various I/O utilities under a variety of scenarios. The results are presented in this report. It is hoped that these results will assist the reader in choosing more efficient I/O strategies for existing and future computing tasks on the CRAY-1 computers.

DEFINITIONS

Blocked Datasets

Unless otherwise specified, all CRAY-1 datasets are blocked. This means that the CRAY-1 will take many of your small logical records and, using a control word structure, fill out a 512-word physical record (block). Alternately, it will take your large records and break them up to fit into 512-word blocks. These blocks are then moved by the operating system to the requested peripheral device. A block control word (BCW) is the first word of every 512-word block, and a record control word (RCW) occurs at the end of each record, file, and dataset. The dataset blocking structure is transparent to the user.

Figure 1. The CRAY-1 dataset blocking structure.

Figure 1

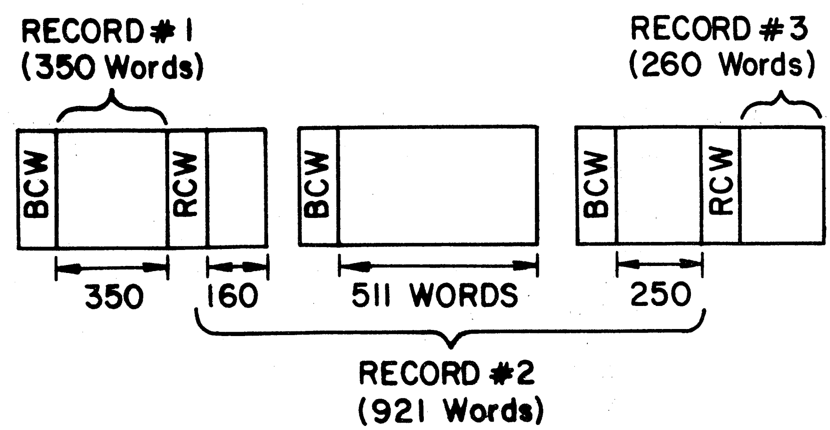
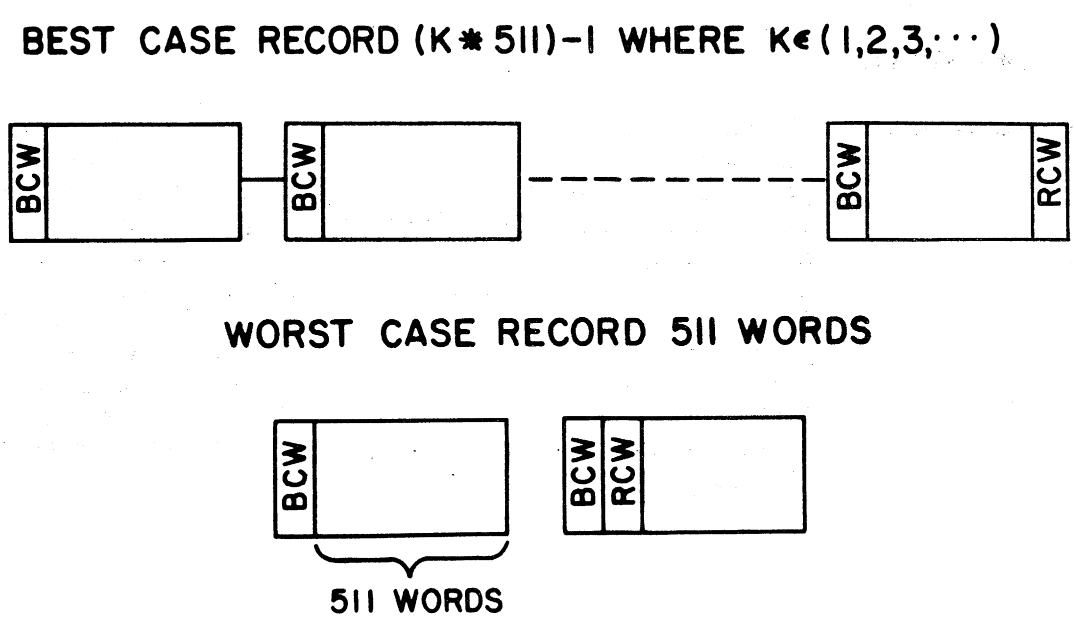


Figure 2. Best and worst case blocking structure.

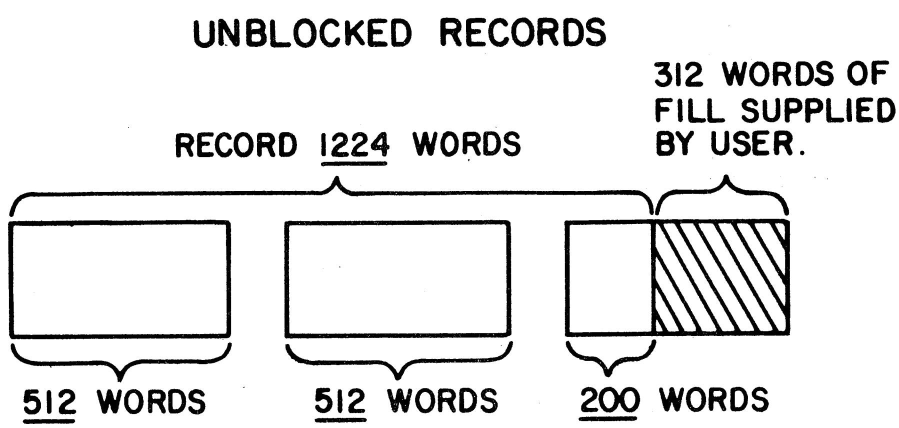
Figure 2



Unblocked Datasets

To declare a dataset as unblocked, the "U" option must be specified on the ASSIGN statement (e.g., ASSIGN,DN=dn,..,U.). For unblocked datasets, the CRAY operating system does not put record and block control words into the dataset structure. Therefore, in an unblocked dataset, all records must have lengths which are a multiple of 512 words. That is, each logical record written or read must be the size of a physical block (physical record) on the peripheral device. When the data cannot be arranged into 512-word multiple records, the user must provide filler words, forcing each logical record to contain a multiple of 512 words. Figure 3 illustrates a 1536-word logical record containing 1224 words of data and 312 words of fill.

Figure 3



Sequential Access Operations

There are two methods of access to a blocked or unblocked dataset: sequential and non-sequential. Sequential access operations are based on the sequential storage of records within files and files within datasets. The order of the records is the order in which they are written. The first record accessed is the record written first, the second record accessed is the record written second, and so on.

Non-Sequential Access Operations There are two types of non-sequential I/O available to CRAY users: random and direct access. In random access operations, records can be read or written in any order. Random datasets may be positioned to the appropriate record, then read or written. In general, records are written sequentially, and may be read or re-written in a random order. Records on a random dataset need not be of equal length.

In direct access operations, records can be read or written in any order. All records of the dataset have the same length and each record of a dataset has a unique record number. The record number is a positive integer specified when the record is written. Once established, the record number cannot be changed. A record can be overwritten but not deleted. Direct access is available through the FORTRAN 77 READ and WRITE statements, which include the REC= keyword option. Direct access was not among the I/O utilities tested in this study.

Synchronous Operations I/O may be performed synchronously or asynchronously. Synchronous I/O operations suspend CPU activity while the I/O activity is in progress.

Asynchronous Operations Asynchronous I/O operations do not suspend CPU activity during data storage and retrieval. Asynchronous I/O utilities always have an additional call to determine when the I/O has been completed. Because asynchronous I/O operations and CPU activity may occur concurrently, the user must take care that the I/O information (e.g., the portion of memory referenced in the I/O statement) is not used or changed until the current I/O process has terminated.

TEST STRATEGY Six I/O utilities were selected for study in this CRAY I/O report:

1. the FORTRAN unformatted READ/WRITE statement
2. BUFFER IN/OUT
3. RDTAPE/WRTAPE
4. READMS/WRITMS
5. READDR/WRITDR
6. BRANIO

NOTE: FORTRAN formatted I/O was not tested, although it can be performed sequentially or non-sequentially on blocked datasets. Formatted I/O on the CRAY-1 is presently quite inefficient with respect to the utilities tested. Direct access I/O operations were not performed. They can be performed only on blocked datasets. This report does not discourage the use of formatted or direct access I/O, it merely does not include these I/O methods among those tested.

Dataset sizes of 1.25 million words were chosen. These datasets were configured using record sizes of 100, 500, 5000, and 10000 words. The datasets were blocked or unblocked and were read and written with random or sequential access either synchronously or asynchronously. Each I/O utility tested used every I/O method available to it. The test cases were run on the CRAY,CA. All test cases were run during normal job mix on the CRAY,CA as well as during a block of dedicated time in which each job was run with no other jobs in the machine. GAU statistics, CPU execution time, and I/O wait time were recorded for each test case. The statistics provided will allow the reader to compare the I/O utilities relative to one another on the basis of the improvement or degradation of an individual utility's performance over the range of record sizes.

To assure nearly equal memory usage, all test programs contained one large I/O array dimensioned by 10240 words. Some programs, depending upon the I/O utility and the test scenario, also contained index or table arrays. The code logic for each test category was identical for each I/O utility.

THE I/O UTILITIES TESTED

Unformatted FORTRAN READ/WRITE

The unformatted READ/WRITE is a synchronous FORTRAN instruction. Therefore, no asynchronous tests were possible. The CRAY library routines GETPOS and SETPOS were used in conjunction with the FORTRAN READ/WRITE in order to test this utility using random access.

It should be pointed out that for FORTRAN READ/WRITE statements, the present version of the CRAY compiler generates more efficient code when the array name is used as the I/O list item than when the I/O list item is defined using an implied-DO statement.

EFFICIENT	INEFFICIENT
DIMENSION ARRAY(1000) WRITE(unit) ARRAY	DIMENSION ARRAY(1000) WRITE(unit) (ARRAY(I),I=1,1000)

Unformatted FORTRAN READ/WRITE tests using synchronous I/O on blocked datasets were run both ways. The following table shows the GAU usage when these cases were run under a normal job mix (e.g., in a non-dedicated environment). Observe the inefficiency of the implied-DO code. See the "DESCRIPTION OF TABLES" section of this report for an explanation of the test cases referred to below.

Test Case	ARRAY name = list	Implied-DO
1.	0.00491	0.01575
2.	0.00976	0.01294
3.	0.00519	0.01267
4.	0.00656	0.01243

BUFFER IN/OUT

The BUFFER IN/OUT utility was run both synchronously and asynchronously, with blocked and unblocked datasets. The CRAY utilities GETPOS and SETPOS were used in conjunction with BUFFER IN/OUT in order to test this utility using random access. At present, it is not possible to use BUFFER OUT on random blocked datasets due to inadequacies in the CRAY I/O buffer management software. However, it is possible to use BUFFER IN on random blocked datasets.

RDTAPE/WRTAPE

The RDTAPE/WRTAPE utility was run both synchronously and asynchronously, with blocked and unblocked datasets. The CRAY utilities GETPOS and SETPOS were used in conjunction with RDTAPE/WRTAPE in order to test this utility using random access. Since RDTAPE/WRTAPE uses BUFFER I/O, random I/O is not possible with blocked datasets for WRTAPE.

READMS/WRITMS

The CRAY I/O utility READMS/WRITMS uses unblocked datasets. However, the READMS/WRITMS utility does not require the user to pass record lengths of 512-word multiples. Therefore, the READMS/WRITMS cases were set up using 512-word multiple records (in the tables these are referred to as UNBLOCKED cases) and using records which were not 512-word multiples (these are referred to as BLOCKED cases in the tables). The READMS/WRITMS BLOCKED cases should more correctly be referred to as PSEUDO-BLOCKED cases.

READDR/WRITDR The CRAY I/O utility READDR/WRITDR uses unblocked datasets. However, the READDR/WRITDR utility does not require the user to pass record lengths of 512-word multiples. Therefore, the READDR/WRITDR cases were set up using 512-word multiple records (these are referred to as the UNBLOCKED cases) and using records which were not 512-word multiples (these are referred to as the BLOCKED cases). The READDR/WRITDR BLOCKED cases should more correctly be referred to as PSEUDO-BLOCKED cases.

NOTE: For asynchronous I/O, the READDR/WRITDR documentation **STRONGLY** recommends that record lengths be 512-word multiples.

BRANIO This utility was used by old CDC 7600 codes which were ported to the CRAY-1 when the CDC 7600 was disconnected. BRANIO is a synchronous utility using blocked datasets.

DESCRIPTION OF TABLES

Information Presented Several tables present the results of this study. Each table has the same structure. The first, Table 1, contains GAUs used, CPU execution time, and I/O wait time for the sequential tests run during normal job mix on the CRAY,CA. Table 2 contains the same information for the corresponding random tests. Tables 3 and 4 contain results for these same tests when run under dedicated conditions on the CRAY,CA. ("Dedicated" means that each case was run while no other jobs were in the machine).

Table Headings The heading at the top indicates whether the table refers to random or sequential test information and if the statistics refer to a dedicated or non-dedicated machine. The next three lines describe what statistical information is found in each box (e.g., GAUs used, CPU execution time, and I/O wait time).

Column Headings The six I/O utilities selected for this study are named at the top of each column. The column contains statistical information for that utility.

Row Headings The column heading for the leftmost column contains row information. Each row contains information for a particular dataset type (blocked or unblocked) and an I/O strategy (synchronous or asynchronous). As many as five cases appear in each box. The cases are numbered 0-4. Each case refers to a dataset configuration. Each dataset is defined using a specified record size and a specified number of data records (e.g., the dataset for case 1 contains 12500 records, and each record is 100 words long). The numbers appearing in front of each row refer to the dataset configuration being accessed.

Each box in the table represents a particular I/O strategy for a particular I/O utility. The rows of boxes are grouped under synchronous or asynchronous processes using blocked or unblocked datasets.

How To Read The Tables

Below is a box, taken from Table 1, which contains the results for the BUFFER IN/OUT statements. The I/O was performed in a **SEQUENTIAL** manner using a **SYNCHRONOUS** process on **BLOCKED** datasets.

```
                        |BUFFER IN/         |
                        |BUFFER OUT         |
    ===================================================
SYNCHRONOUS       1     |  0.00513          |<--- GAUs Used
                        |        02.4078    |<--- CPU Time
                        |        15.9176    |<--- I/O Wait Time
                  2     |0.00344            |
                        |        00.7566    |
                        |        15.5277    |
BLOCKED           3     |  0.00427          |         ---------
                        |        00.3696    |         Box from
                        |        27.1278    |         Table 1
                  4     |0.00367            |         ---------
                        |        00.3026    |
                        |        18.8727    |
                        |-------------------
```

The datasets were configured in four ways:

1. 12500 record dataset with 100 words per record.
2. 2500 record dataset with 500 words per record.
3. 250 record dataset with 5000 words per record.
4. 125 record dataset with 10000 words per record.

Each line (1-4) in this box contains test run statistics for the particular dataset configuration being accessed. The numbered line contains the GAU usage for the case tested. The next two indented lines contain CPU time and I/O wait time respectively. CPU and I/O wait time are given in seconds or in minutes:seconds.

Case Winners

In each row, the I/O utility which appears in boldface is the case winner. The numbers represent the GAUs used for a given I/O scenario. These numbers are not statistical averages; non-dedicated runs are the results of a single I/O scenario running with a non-repeatable job mix on the CRAY,CA, whereas dedicated runs produce repeatable results. The programmer should weigh both sets of statistics when attempting to determine an efficient I/O strategy.

Zero Cases Notice that all test cases using unblocked datasets contain a zero (0) case. As discussed earlier, unblocked datasets require that record sizes be 512-word multiples. In boxes where zero cases appear, the reader should compare case one with case zero. Case one writes 100 words of data and 412 words of fill for every I/O statement (12500 reads/writes) in the test. This is extremely wasteful since 6.4 million words are moved in order to write 1.25 million words of data information. Case zero writes 512 words of data (e.g., no fill) for every I/O statement (2448 reads/writes) in the test. Approximately the same number of words of user data are moved in each case; however, case zero moves the words of data in 512 word multiples with no fill. Comparing test cases zero and one demonstrates that the user can optimize I/O by efficiently organizing the data into record sizes of 512 words. Case zero proves to be much more efficient.

CPU Activity Some of the I/O utilities selected could be tested synchronously as well as asynchronously. For these I/O utilities, a method was designed to measure the amount of CPU activity which could occur during each asynchronous I/O request. This simulated the effect of a user "covering" I/O † wait time with computation.

These same I/O utilities were then tested using synchronous I/O requests. The synchronous tests were run with the same CPU activity of the corresponding asynchronous tests (BLOCKED/CPU, UNBLOCKED/CPU). The synchronous tests were also run without the simulated CPU activity (BLOCKED, UNBLOCKED).

These tests should be used to compare the advantages and disadvantages of "covering" asynchronous I/O with computation.

† "Covering I/O" refers to a strategy whereby the CPU remains active during an I/O process.

Table 1
<u>Sequential I/O (Non-dedicated)</u>
CRAY-1 Termination in GAUs
Time Executing in CPU
Time Waiting for I/O

REC SIZE/# RECS
0. 512/2442
1. 100/12500
2. 500/2500
3. 5000/250
4. 10000/125

Mode	Rec	UNFORMATTED FORTRAN READ/WRITE	BUFFER IN/ BUFFER OUT	RDTAPE/ WRTAPE	READMS/ WRITMS	READDR/ WRITDR	BRANIO
SYNCHRONOUS BLOCKED	1	0.00491 01.0321 21.8849	0.00513 02.4078 15.9176	**0.00451** 02.7285 16.2471	0.01202 02.1087 49.7796	0.07906 04.4286 05:06.3155	0.19494 09.4203 24:37.9104
	2	0.00976 00.5730 01:26.1545	**0.00344** 00.7566 15.5277	0.00443 00.8336 23.8795	0.01492 01.0537 01:35.7186	0.02156 00.9708 02:02.0238	0.03752 02.1366 04:06.8127
	3	0.00519 00.4771 31.9416	0.00427 00.3696 27.1278	**0.00377** 00.3932 20.3152	0.00677 00.4820 49.6748	0.00687 00.2940 52.7129	0.00938 01.1926 43.2013
	4	0.00656 00.4703 51.3429	**0.00367** 00.3026 18.8727	0.00430 00.3608 29.8555	0.00725 00.4465 01:00.8672	0.00546 00.2406 40.6478	0.00813 01.1335 54.2356
UNBLOCKED	0	0.01706 00.6075 01:04.7146	0.01666 00.9458 57.0502	0.02145 01.0378 02:04.7907	**0.00533** 00.6962 35.5366	0.01484 00.9383 30.4255	N/A
	1	0.13022 02.4759 17:28.4139	0.08415 04.1547 06:23.9963	0.07422 04.4994 04:00.3171	**0.01121** 02.7075 01:03.3182	0.08089 04.1679 05:33.2680	
	2	0.01938 00.6139 01:33.8223	0.01776 00.9575 01:08.8906	0.01860 01.0524 01:21.0733	**0.00682** 00.7053 56.1847	0.01638 00.9487 48.8952	
	3	0.00552 00.2158 35.0347	**0.00460** 00.2664 20.9202	0.00842 00.3029 01:15.7443	0.00665 00.4658 52.5636	0.01017 00.2876 01:40.0942	
	4	0.00467 00.1859 30.6878	0.00457 00.2220 28.7382	**0.00430** 00.2481 25.0892	0.00833 00.4554 01:16.2452	0.00622 00.2408 51.4712	
BLOCKED/CPU	1	N/A	**0.01417** 23.0997 17.8229	0.01838 36.0244 33.9424	0.10785 05:49.5321 33.5404	0.12384 31.7783 13:38.2353	N/A
	2		**0.00821** 06.3015 16.7916	0.01089 09.8218 50.6506	0.02852 52.8520 01:23.5874	0.01754 04.3732 51.3124	
	3		**0.00599** 07.4038 12.8538	0.00765 06.6807 40.1860	0.00912 14.7906 26.2626	0.00830 05.6287 52.0220	
	4		0.00656 03.7929 43.6138	**0.00484** 03.8342 18.5386	0.00737 03.4836 49.8642	0.00534 04.9693 20.7179	
UNBLOCKED/CPU	0	N/A	**0.01773** 05.2254 54.5764	0.02045 17.6448 44.4282	0.02009 51.8146 42.5523	0.02741 04.7204 03:10.7116	N/A
	1		0.08012 29.1962 03:42.3310	0.08784 48.0906 04:15.7679	**0.07705** 03:56.2994 01:12.0133	0.10557 23.0174 09:57.6074	
	2		0.01777 08.3700 39.9059	0.02155 07.6548 01:35.7576	**0.01614** 45.7302 10.4232	0.01923 08.2395 01:00.0890	
	3		**0.00538** 05.9041 10.3342	0.00589 05.6384 18.1784	0.00617 08.5004 13.7301	0.01070 06.1004 01:24.6046	
	4		0.00489 05.0973 14.3079	0.00635 08.2238 22.2279	**0.00480** 03.2659 14.9377	0.00756 05.5285 50.1535	
ASYNCHRONOUS BLOCKED/CPU	1	N/A	**0.01489** 24.3139 12.0740	0.01833 36.7149 11.5154	0.14861 04:28.7162 01:20.6971	0.06579 30.1697 14.8773	N/A
	2		**0.00920** 06.3503 03.1685	0.01052 09.9619 07.4972	0.04011 54.3774 01:20.2528	0.01434 54.3774 07.3292	
	3		0.00682 07.4556 24.6343	**0.00508** 06.7478 02.4497	0.00869 15.0266 11.3182	0.00629 05.6304 24.0196	
	4		0.00440 03.8043 11.5035	**0.00399** 03.7033 06.3333	0.00572 03.5727 15.2309	0.00569 05.0078 25.4778	
UNBLOCKED/CPU	0	N/A	**0.01472** 04.9860 13.1523	0.01871 17.8504 18.6485	0.02909 53.2274 07.6933	0.01561 04.4496 27.0519	N/A
	1		0.06417 27.9115 02.4892	0.06991 47.5094 05.2714	0.13051 04:02.7345 03.5304	**0.06324** 21.5818 12.8012	
	2		0.01568 08.1529 10.6792	**0.01492** 07.4622 02.5517	0.02800 47.0080 13.9218	0.01658 08.0103 23.8804	
	3		0.00574 05.9525 14.7422	**0.00476** 05.7297 01.7954	0.00596 08.6200 06.5054	0.00637 06.1293 23.2899	
	4		**0.00412** 05.1368 02.8777	0.00494 08.1787 02.1439	0.00490 03.3293 11.7118	0.00496 05.5839 12.5372	

Table 2

Random I/O (Non-dedicated)

CRAY Termination in GAUs
Time Executing in CPU
Time Waiting for I/O

REC SIZE/# RECS
0. 512/2442
1. 100/12500
2. 500/2500
3. 5000/250
4. 10000/125

		UNFORMATTED FORTRAN READ/ WRITE	**BUFFER IN/ BUFFER OUT**	**RDTAPE/ WRTAPE**	**READMS/ WRITMS**	**READDR/ WRITDR**	**BRANIO**
SYNCHRONOUS	1	0.09344 04.0530 08:08.7935			**0.02648** 02.9887 01:54.6700	0.09859 04.6569 09:41.8853	0.20190 09.5841 26:18.6363
	2	0.07451 01.3383 13:15.7447			0.03017 01.8779 03:08.0278	**0.02227** 01.0240 02:12.0950	0.03915 02.1472 04:29.6852
BLOCKED	3	0.01280 00.5836 01:39.2169	N/A	N/A	0.00787 00.5495 57.5259	0.00679 00.3038 51.9351	**0.01155** 01.1922 01:12.6522
	4	0.00660 00.5460 29.3111			0.01000 00.5205 01:29.2365	**0.00420** 00.2555 22.9019	0.00785 01.1174 39.6302
	0	**0.01553** 00.7016 41.9661	0.01966 01.0742 01:39.0776	0.01926 01.1601 01:32.3464	0.01715 01.3982 01:02.4751	0.01641 01.0040 52.8112	
	1	0.09320 02.8866 08:34.9338	0.08368 04.6203 06:12.4632	0.11072 04.9682 12:33.7496	**0.08289** 06.2536 05:59.4053	0.09853 04.3372 09:42.0369	
UNBLOCKED	2	0.01817 00.7109 01:16.1854	0.01902 01.0875 01:26.5966	0.01948 01.1747 01:32.5111	0.02095 01.4241 01:53.4147	**0.01753** 01.0194 01:05.2984	N/A
	3	0.00582 00.2527 38.8428	0.00671 00.3242 51.5849	**0.00526** 00.3511 30.1631	0.00580 00.4953 37.4443	0.00851 00.3068 01:16.7044	
	4	**0.00399** 00.2045 20.3562	0.00499 00.2682 34.9541	0.00644 00.2959 54.5600	0.00641 00.4864 45.4821	0.00490 00.2539 33.0798	
	1				**0.06445** 02:26.9738 01:22.3114	0.15987 03:04.6645 12:04.0776	
	2				0.04032 47.3841 02:31.3838	**0.03551** 51.3210 02:00.4873	
BLOCKED/CPU	3	/N/A	N/A	N/A	0.01024 13.3528 39.9079	**0.00808** 09.5430 33.8549	N/A
	4				0.01004 16.5240 26.1585	**0.00731** 07.7316 37.4728	
	0		**0.02291** 26.5893 41.8087	0.03261 50.3934 01:24.2967	0.03407 39.7252 02:29.5287	0.02856 38.9580 01:12.4020	
	1		**0.12548** 02:44.4929 05:23.6007	0.15097 03:15.6825 09:16.9796	0.15727 03:50.4632 08:41.8325	0.19751 03:58.0155 17:20.9202	
UNBLOCKED/CPU	2	N/A	**0.03042** 39.8287 01:34.5969	0.03178 47.6668 01:22.6074	0.03113 42.5068 01:34.8678	0.03110 48.6459 01:08.1787	
	3		**0.00828** 13.3356 21.7842	0.00937 12.5958 40.3276	0.00902 12.6288 34.9670	0.01027 13.5684 48.5503	N/A
	4		0.00578 07.7061 16.1068	0.00787 08.5105 42.0590	**0.00558** 03.9163 20.1105	0.00782 09.8369 36.3299	
ASYNCHRONOUS	1				0.13623 02:32.7010 03:28.5404	**0.10893** 03:05.4790 08.2366	
	2				0.03674 47.7591 01:12.9195	**0.02771** 51.6604 08.4745	
BLOCKED/CPU	3	N/A	N/A	N/A	0.00899 13.4226 22.5988	**0.00616** 09.5601 06.4016	N/A
	4				0.01055 16.4680 33.6329	**0.00649** 07.7662 25.7281	
	0		**0.02040** 26.7074 06.5600	0.02729 50.3975 09.4225	0.02493 40.0624 18.2474	0.02508 39.1481 23.7850	
	1		**0.10351** 02:45.6396 11.6942	0.11165 03:15.1586 08.9045	0.12162 03:52.6519 06.8788	0.12406 04:00.0362 06.0375	
UNBLOCKED/CPU	2	N/A	**0.02419** 40.1502 04.6403	0.02615 47.5658 02.7509	0.02493 42.8296 04.9646	0.02827 48.8480 28.0602	N/A
	3		0.00736 13.5938 07.6598	**0.00671** 12.6203 01.9999	0.00744 12.7461 11.6451	0.00742 13.6270 08.1930	
	4		0.00505 07.8214 05.3151	**0.00501** 08.5584 01.6372	0.00517 03.9220 14.4166	0.00718 09.8920 27.3531	

Table 3
<u>Sequential I/O (Dedicated)</u>
<u>CRAY Termination in GAUs</u>
Time Executing in CPU
Time Waiting for I/O

REC SIZE/# RECS
0. 512/2442
1. 100/12500
2. 500/2500
3. 5000/250
4. 10000/125

		UNFORMATTED FORTRAN READ/ WRITE	**BUFFER IN/ BUFFER OUT**	**RDTAPE/ WRTAPE**	**READMS/ WRITMS**	**READDR/ WRITDR**	**BRANIO**
SYNCHRONOUS	1	0.00390 01.0732 09.0694	0.00308 02.4200 04.0146	**0.00297** 02.7290 03.4336	0.01054 02.1063 28.7933	0.06515 04.5224 49.0752	0.17363 09.5386 19:49.1311
	2	**0.00259** 00.6065 06.7322	0.00284 00.8321 06.5679	0.00262 00.9253 05.6530	0.01031 01.0567 29.4083	0.01459 00.9947 23.4601	0.03240 02.1716 02:56.9794
BLOCKED	3	0.00246 00.4931 05.2850	0.00263 00.4304 05.8859	**0.00245** 00.4888 05.2494	0.00418 00.4848 12.6902	0.00374 00.2948 08.8237	0.00734 01.2248 26.7548
	4	0.00253 00.4847 06.2879	0.00257 00.3900 06.0128	**0.00251** 00.4611 06.2698	0.00376 00.4510 10.8524	0.00304 00.2446 07.0129	0.00541 01.1463 19.5322
	0	0.01474 00.6070 31.6835	0.01408 00.9536 20.4415	0.01417 01.0420 21.5261	**0.00363** 00.6995 11.3393	0.01432 00.9694 22.9467	
	1	0.06668 02.4730 02:21.9063	0.06402 04.1902 01:37.2157	0.06427 04.5410 01:38.7771	**0.01003** 02.7144 46.5976	0.06506 04.2161 01:49.2527	
UNBLOCKED	2	0.01487 00.6163 29.8755	0.01431 00.9742 20.4271	0.01438 01.0590 20.5865	**0.00362** 00.7101 10.8483	0.01456 00.9876 22.9848	
	3	0.00379 00.2277 10.3454	0.00373 00.2784 09.1311	**0.00359** 00.3101 06.8011	**0.00359** 00.4693 08.8185	0.00371 00.2905 08.4540	N/A
	4	0.00310 00.1616 08.4949	0.00301 00.2235 06.6378	**0.00300** 00.2535 06.5781	0.00363 00.4572 09.3989	0.00303 00.2429 06.8409	
-----------	1		0.01602 32.0085 01.9635	**0.01579** 31.3427 02.0808	0.11394 06:17.2127 10.7147	0.07248 20.0670 02:26.5702	
	2		**0.00704** 06.0426 03.5537	0.00722 07.5885 04.9589	0.02513 54.9539 26.7835	0.01627 05.9708 27.3647	N/A
BLOCKED/CPU	3	N/A	**0.00479** 04.3158 08.3671	0.00498 04.7424 09.2150	0.00656 09.1005 12.4284	0.00568 06.5449 11.5664	
	4		0.00445 05.3176 07.1674	0.00448 05.4953 06.8216	0.00499 04.7588 11.2205	**0.00442** 04.8764 07.8110	
	0		0.01595 05.2923 29.5392	0.01595 05.0898 30.0458	0.01683 48.5031 08.9771	**0.01585** 04.6663 29.0518	
	1		0.07433 29.2921 02:19.8786	**0.07265** 21.1383 02:27.7758	0.07662 04:02.5429 41.7203	0.07399 26.9866 02:20.4119	
UNBLOCKED/CPU	2		**0.01607** 04.6677 30.2980	0.01621 05.1349 30.0613	0.01873 55.0446 10.4588	0.01657 07.0754 27.4545	
	3	N/A	**0.00520** 05.5733 08.8832	0.00536 05.4937 11.3814	0.00575 08.1335 09.1448	0.00539 05.8562 10.2379	N/A
	4		0.00453 05.3469 07.7572	0.00446 04.9521 08.6081	0.00448 03.3267 09.9518	**0.00436** 04.7049 07.7093	
ASYNCHRONOUS	1		0.01645 32.5118 01.5185	**0.01628** 31.9344 01.4822	0.18097 06:24.5832 01:21.2618	0.06161 18.2763 01.7467	
	2		**0.00899** 06.1184 01.4929	0.00943 07.6695 01.4201	0.04050 56.6412 01:16.6513	0.01441 05.6995 02.0086	
BLOCKED/CPU	3	N/A	0.00462 04.2608 05.8703	**0.00446** 04.7250 01.6776	0.00688 09.1689 08.7402	0.00500 06.5739 01.7310	N/A
	4		0.00403 05.2068 01.2747	0.00405 05.3013 01.3006	0.00588 04.9286 12.6007	**0.00401** 04.9135 01.7212	
	0		0.01391 05.0544 01.4608	**0.01381** 04.5971 01.8487	0.02776 49.9185 02.0016	0.01382 04.3953 01.9688	
	1		0.06416 28.0190 01.6308	**0.06190** 19.8545 01.6716	0.13233 04:09.7037 02.0057	0.06368 25.6115 01.8139	
UNBLOCKED/CPU	2		**0.01395** 04.3981 01.3999	0.01412 04.8979 01.4002	0.02977 56.5782 01.7022	0.01469 06.8318 01.7366	
	3	N/A	**0.00467** 05.6201 01.3183	0.00470 05.5963 01.4225	0.00552 08.1716 01.8558	0.00478 05.8824 01.5876	N/A
	4		0.00407 05.4026 01.3376	0.00398 04.9099 01.4986	0.00460 03.4135 07.9283	**0.00395** 04.7450 01.9193	

Table 4
Random I/O (Dedicated)
CRAY Termination in GAUs
Time Executing in CPU
Time Waiting for I/O

REC SIZE/# RECS
0. 512/2442
1. 100/12500
2. 500/2500
3. 5000/250
4. 10000/125

		UNFORMATTED FORTRAN READ/ WRITE	**BUFFER IN/ BUFFER OUT**	**RDTAPE/ WRTAPE**	**READMS/ WRITMS**	**READDR/ WRITDR**	**BRANIO**
SYNCHRONOUS							
BLOCKED	1	0.07905 04.2552 04:46.3512	N/A	N/A	**0.02404** 03.0134 01:20.0568	0.07241 04.6499 03:31.6077	0.17334 09.7031 19:43.5585
	2	0.02629 01.5030 04:47.8468	N/A	N/A	0.02200 01.9030 01:10.9421	**0.01611** 01.0313 45.2226	0.03166 02.1971 02:52.4420
	3	0.00694 00.6030 22.8949	N/A	N/A	0.00531 00.5573 20.8655	**0.00442** 00.3143 18.5201	0.00775 01.2164 26.8534
	4	0.00523 00.5688 18.1445	N/A	N/A	0.00499 00.5268 18.2678	**0.00336** 00.2567 11.0969	0.00550 01.1549 20.3794
UNBLOCKED	0	0.01533 00.7003 38.8948	**0.01501** 01.0788 32.9477	0.01654 01.1666 54.0211	0.01622 01.4041 49.0049	0.01583 01.0035 44.4375	N/A
	1	0.07486 02.8794 04:14.7101	0.07354 04.6495 03:48.3265	0.07362 04.9839 03:47.6931	0.07412 06.2701 03:53.7097	**0.07203** 04.3329 03:27.1152	N/A
	2	0.01667 00.7111 54.8957	0.01621 01.0929 46.6068	0.01643 01.1813 48.8685	0.01645 01.4303 49.0283	**0.01596** 01.0232 42.9379	N/A
	3	0.00427 00.2560 16.8857	0.00421 00.3313 15.9457	0.00417 00.3587 14.7891	**0.00411** 00.5014 13.3023	0.00418 00.3109 15.1607	N/A
	4	**0.00329** 00.2132 10.7198	**0.00329** 00.2728 10.7345	0.00334 00.2970 10.8012	0.00418 00,4901 14.1284	0.00331 00.2534 10.5956	N/A
BLOCKED/CPU	1	N/A	N/A	N/A	**0.05555** 01:55.9751 01:18.4720	0.12617 03:26.2871 02:48.3519	N/A
	2	N/A	N/A	N/A	0.02913 27.2175 01:11.9929	**0.02743** 43.4922 37.0971	N/A
	3	N/A	N/A	N/A	**0.00837** 11.2184 22.3305	0.00866 14.7564 21.4048	N/A
	4	N/A	N/A	N/A	0.00627 04.9851 18.4101	**0.00569** 08.5239 11.3974	N/A
UNBLOCKED/CPU	0	N/A	**0.02313** 27.9647 40.2137	0.02479 29.8746 55.3848	0.02644 38.5917 45.9585	0.02575 37.4596 38.8477	N/A
	1	N/A	0.12878 03:30.6278 03:09.3146	0.13104 03:38.0751 03:10.6364	0.13107 03:30.7411 03:48.3026	**0.12389** 03:15.3723 02:59.0207	N/A
	2	N/A	0.02848 46.7347 39.2894	0.02789 44.6294 39.4571	0.02826 43.9751 48.1188	**0.02656** 41.0662 34.4020	N/A
	3	N/A	0.00760 12.8633 14.3619	0.00746 12.1377 15.0291	**0.00651** 09.0051 13.8617	0.00710 10.7708 14.8766	N/A
	4	N/A	0.00523 07.0673 11.1961	**0.00510** 06.5641 11.0817	0.00550 04.8910 15.5195	0.00537 07.5955 10.6369	N/A
ASYNCHRONOUS							
BLOCKED/CPU	1	N/A	N/A	N/A	0.12635 02:00.3611 03:16.8331	**0.11455** 03:27.1339 01.8376	N/A
	2	N/A	N/A	N/A	0.03001 27.3863 58.0656	**0.02500** 43.7456 01.7189	N/A
	3	N/A	N/A	N/A	0.00751 11.2376 10.0172	**0.00734** 14.8953 01.9510	N/A
	4	N/A	N/A	N/A	0.00582 05.0753 11.3116	**0.00501** 08.6136 01.5678	N/A
UNBLOCKED/CPU	0	N/A	**0.02062** 28.1038 04.6826	0.02098 29.9131 01.7848	0.02336 38.5797 02.0706	0.02316 37.6401 01.8541	N/A
	1	N/A	0.11584 03:32.1838 01.8676	0.11743 03:37.7212 01.6065	0.11508 03:30.6106 01.8329	**0.11148** 03:16.3882 01.8621	N/A
	2	N/A	0.02594 47.1220 01.8511	0.02522 44.5335 01.6861	0.02501 43.9730 01.8336	**0.02433** 41.2954 01.7691	N/A
	3	N/A	0.00678 13,0922 01.3459	0.00658 12.1691 01.9136	**0.00571** 08.9985 02.0320	0.00621 10.8566 01.7709	N/A
	4	N/A	0.00460 07.1509 01.3522	**0.00444** 06.5901 01.4984	0.00514 04.9621 09.4310	0.00474 07.6766 01.5552	N/A

CONCLUSIONS AND RECOMMENDATIONS

General Statements For comparable dataset structures, unformatted I/O is more efficient than formatted I/O, and sequential access operations are more efficient than random access operations.

Blocked Datasets For blocked datasets, the system allocates an area in high user memory which serves as a buffer between the user's I/O statements and the physical I/O performed to and from the peripheral device(s). Typically, user I/O requests are much smaller than the buffer size; the buffer serves to reduce the number of physical requests to the peripheral storage devices (disks). The NCAR default buffer size is four CRAY blocks (2048 words). The buffer size of a dataset may be specified by the user on the ASSIGN statement with the BS parameter. In general for small records, blocked datasets are more efficient than unblocked datasets.

It is usually the case that the larger the buffer size of a dataset, the smaller the I/O wait time of the entire job. However, this is **NOT** true for random datasets, unless the record size and buffer size are equal. Thus, for random datasets where record size and buffer size are not equal, large buffer sizes are inefficient.

Unblocked Datasets For unblocked datasets, each user I/O request results in one physical request to the peripheral storage devices. Because of this, no buffers are allocated by the system for unblocked datasets, and the user must read and write records with lengths of 512-word multiples. In general for large records (records approaching or exceeding the system buffer size for blocked datasets of 2048 words), unblocked datasets are more efficient than blocked datasets.

Random Access Operations Random I/O is more efficient for unblocked datasets than for blocked datasets. Because of this, a user wishing to do random I/O on blocked datasets should attempt to make the record size equal to the buffer size. This will eliminate unnecessary physical I/O.

In general, random I/O on the CRAY-1 is quite inefficient. (With COS 1.14, the low level library I/O routines will be rewritten. This should significantly improve random and sequential I/O). Unless you have a real need for random I/O, its use is discouraged. **Caveat:** If you are dealing with very large records (nearly equal to or larger than the buffer size), random I/O is almost as efficient as sequential I/O.

Asynchronous/ Synchronous

In comparing the BLOCKED vs BLOCKED/CPU cases, the statistics demonstrate that it is advantageous for the user to "cover" the I/O wait time.

For non-dedicated tests, GAU use in asynchronous tests is consistently less than that for dedicated tests. This is because, under normal system loads at NCAR, a user's job will be interrupted by the job scheduler in favor of another job. On dedicated time, no other job is available for scheduling. The dedicated tests (as shown by CPU time) were able to perform more computations than the corresponding non-dedicated tests and thus accumulate more GAUs.

BIBILOGRAPHY

CRAY Blocked/Unblocked Dataset Structure

CRAY-OS Version 1 Reference Manual, SR-0011, Revision L-01, Part 1, Section 2.

CRAY FORTRAN (CFT) Reference Manual, SR-0009, Revision J-01, Appendix H.

Sequential I/O and Direct I/O

CRAY FORTRAN (CFT) Reference Manual, SR-0009, Revision J-01, Part 2, Section 5.

Random I/O

CRAY FORTRAN (CFT) Reference Manual, SR-0009, Revision J-01, Appendix E.

CRAY Library Reference Manual, SR-0014, Revision G-01, Chapter 5.

CRAY-OS Version 1 Reference Manual, SR-0011, Revision L-01, Part 2, Section 3.

Synchronous I/O

CRAY FORTRAN (CFT) Reference Manual, SR-0009, Revision J-01, Part 2, Section 5.

CRAY Library Reference Manual, SR-0011, Revision L-01, Chapter 5.

I/O UTILITIES

BUFFER IN/OUT
CRAY FORTRAN (CFT) Reference Manual, SR-0009, Revision J-01, Part 2, Section 5.

CRAY Library Reference Manual, SR-0011, Revision L-01, Chapter 5.

READMS/WRITMS
CRAY FORTRAN (CFT) Reference Manual, SR-0009, Revision J-01, Appendix E.

CRAY Library Reference Manual, SR-0011, Revision L-01, Chapter 5.

NCAR publication, "CRAY Random Access Library Routines",(in press).

READDR/WRITDR
NCAR publication, "CRAY Random Access Library Routines",(in press).

SETPOS/GETPOS
CRAY FORTRAN (CFT) Reference Manual, SR-0009, Revision J-01, Appendices B,C and E.

CRAY Library Reference Manual, SR-0011, Revision L-01, Section 5.

CRAY Macros and Opdefs Manual, SR-0012, Original printing, Chapter 3.

BRANIO
NCAR publication, "MANAGING DATASETS AND PROGRAMS AT NCAR: The Mass Storage Subsystem", Chapter 4.

RDTAPE/WRTAPE
NCAR publication, "MANAGING DATASETS AND PROGRAMS AT NCAR: The Mass Storage Subsystem", Chapter 4.

System Level I/O

CRAY Library Reference Manual, SR-0011, Revision L-01, Chapters 5 and 6.

CRAY Macros and Opdefs Manual, SR-0012, Original printing, Chapters 2 and 3.

CRAY COS Table Descriptions Manual, SM-0045 (available from CRAY site analyst).

CRAY COS EXEC/STP/CSP Internal Reference Manual, SM-0040, Revision B

Overview: Pages 1-24 through 1-28
Circular I/O routines and buffer management: Section 4.2
Task I/O routines: Section 4.1
Dataset allocation: Section 6.3
Dataset tables: Section 8

Non-Profit Organization
U.S. POSTAGE PAID
Boulder/Colorado
Permit No. 558

The Record

Issued by the
National Center for Atmospheric Research
Scientific Computing Division
P.O. Box 3000
Boulder, Colorado 80307

PRINTED MATTER

SUMMERS, BARBARA
Mesa Lab
NCAR
Boulder, CO (MAIL ROOM)