Appendix A3

Appendix for Chapter 3

This chapter appendix provides tables of information about the physical configuration of the CDF system. It also provides information about course accounts in the CDF system and the data collection process.

A3.1 Hardware and Software

Table A3.1 contains information about the hardware and software that was installed on each CDF machine at the time of our data collection. This information is further described in Section 3.1.1.

Machine	Total Disk Capacity	Main Memory Size (MB)	CPU Speed (MHz)	Number of CPUs	SunOS Version
eddie.cdf SPARCsystem 600	1.7 GB	128	40	4	4.1.2
marvin.cdf SPARCserver 490	7.2 GB	98	33	1	4.1.1
dev.cdf IPC SPARCstation	1.4 GB	24	25	1	4.1.1
ws111.cdf IPC SPARCstation	200 MB	24	25	1	4.1.1
21 colour IPC SPARCstations	200 MB	16	25	1	4.1.1
46 mono IPC SPARCstations	200 MB	12	25	1	4.1.1

Table A3.1: Hardware and Software Configuration

A3.2 Disk Drives

Provided in this section are tables that display the CDF physical disk information that is explained in Section 3.1.5.

Disk Type	Capacity per disk	Device Name	Partition names
Maxtor LXT-213S or Quantum PD210S (SCSI controller)	200 MB	sd0	/ /usr /var /tmp
Floppy Disk Drive		fd0	

Table A3.2: Disks for 68 Sun IPC SPARCstations

Disk Types	Capacity per disk	Device Names	Partition names
Maxtor LXT213S (SCSI controller)	200 MB	sd0	/proto/root /tmp /proto/usr
Fujitsu M2266S (SCSI controller)	1.2 GB	sd1	/ /usr /var /tmp /local /admin
Exabyte EXB-8200 8mm tape (SCSI controller)		st0	
Floppy Disk Drive		fd0	

Table A3.3: Disks for Sun IPC SPARCstation, dev

Disk Types	Capacity per disk	Device Names	Partition names
Fujitsu M2654SA (SCSI controller)	2 GB	sd0	/homes/u4 /solaris/usr /solaris/local /cdf
4 Seagate ST9722 Sabres (two IPI controllers)	1 GB	id0 id1 id2 id3	/ /usr /local /var /backup /homes/u1 /tmp /export/root /homes/u2 /tmp /export/root2 /homes/u3 /tmp
3 Fujitsu M2351 Eagles (SMD controller)	400 MB	rf0 rf1 rf2	/backup1 /backup2 /backup3 /backup4 /export/exec/sun3 /homes/u5

Table A3.4: Disks for file server, marvin

Disk Types	Capacity per disk	Device Names	Partition names
Maxtor LXT213S (SCSI controller)	200 MB	sd0	/news
Fujitsu M2652SA (SCSI controller)	1.5 GB	sd1	/ /var /usr /local /src /tmp

Table A3.5: Disks for compute server, eddie

A3.3 CDF Course Information

Table A3.6 lists all courses that had student accounts in the CDF system at the time of the data collection. All course names started with "csc" and were followed by a three digit (for undergraduate courses) or four digit (for graduate courses) course number. The courses listed in Table A3.6 that have two course numbers were offered as both graduate and undergraduate courses.

CS Course	**Course Title**
csc104	The Why and How of Computing
csc108	Introduction to Computer Programming
csc148	Introduction to Computer Science
csc228	File Structures and Data Management
csc260	Introduction to Scientific, Symbolic, and Graphical Computation
csc300	Computers and Society
csc318	The Design of Interactive Computational Media
csc324	Principles of Programming Languages
csc350	Numerical Algebra and Optimization
csc378	Data Structures and Algorithm Analysis
csc418/2504	Computer Graphics
csc428/2514	Human-Computer Interaction
csc434/2509	Data Management Systems
csc465	Programming Methodology
csc468/2204	Operating Systems
csc470	Computer System Modelling and Analysis
csc485/2501	Introduction to Computational Linguistics
csc2206	Computer Systems Modelling
csc2307	Numerical Software
csc2321	Matrix Calculations
csc2535	Computation in Neural Networks

Table A3.6: Courses having CDF accounts during the 1993 Fall Semester

The first digit of a 3-digit course number represents the year of the course. For example, csc104 was a first year course because its course number started with a "1." All 4-digit

graduate course numbers began with a "2."

The login identification names of students in these courses corresponded to the course numbers. Course login names were composed of an "a," followed by the course number, followed by the first four digits of the user's surname. For example, a student named John Smith, taking course csc104 would have had a login name of "a104smit."

All undergraduate students who were enrolled in the Computer Science program were also given an account that started with a "g," followed by the last digit of the year in which they were first accepted into the program, followed by a sequence of characters chosen by the student. A user with the name of John Smith who was first accepted in 1992 may have had the user name "g2smith," where "g2" would be mandatory, but "smith" would have been chosen by the student. This account, which exists until the student graduates, may be used by the student for work in any course.

A3.4 Data Collection

Table A3.7 describes the flags that were used with the `acctcom` command in the analysis of the process accounting data. The `acctcom` command was used to convert the encoded process accounting data files (`/var/adm/pacct`) on each host into ASCII form.

Command Line Option Used	Option Description
- a	Show some average statistics about the processes selected. The statistics will be printed after the output records.
-h	Instead of mean memory size, show the fraction of total available CPU time consumed by the process during its execution. This is called "hog factor" and is computed as total CPU time divided by elapsed time.
-i	Print columns containing the I/O counts in the output.
-k	Instead of memory size, show total kcore-minutes.
-r	Show CPU factor: user CPU ÷ (system CPU + user CPU).
-t	Show separate system and user CPU times.

Table A3.7: Description of the `acctcom` Options Used

The "collect.sh" file that is shown in Figure A3.1 is the UNIX C-Shell script code that was used to start up eight different collection scripts on each host in the CDF environment. The "collect.sh" script was started by giving an `rsh` command for each host from marvin.

To start the "collect.sh" script on frogstar, for example, the following command was issued from marvin:

```
rsh frogstar.cdf "~/collect.sh" &
```

The "iostatD.sh" file, shown in Figure A3.2, contains the UNIX C-Shell script code that was used to collect data from the `iostat -D` command on a particular host (determined by the `hostname` command) on December 7^{th}. As the actual collection of the `iostat -D` data started at 12:00 am, we started this script before this time.

A `while` loop was used to continuously check the date until the day is the 7^{th}. At that time, the loop exits and the `iostat -D` collection begins. The data are stored on the local disk in a file that is named according to the host's name: `/tmp/$host.ioD`.

The `iostat -D` command was used with the numerical parameters `20` and `3782`. The first parameter is 20 seconds, for the frequency of the collection (at 20 second intervals). The second parameter is 3782, for the number of displays. Since data were collected for 24 hours, this parameter was determined by dividing 86400 seconds (24 hours) by 20 seconds, and then adding two extra displays for safety.

When the `iostat -D` command completed, the data file was compressed and then remotely copied from the local workstation disk to an account on another computer system. Following this remote copy, the data file was then deleted from the workstation's local disk.

```
#! /usr/bin/csh -f

# Name: collect.sh
#
# Desc: This script is started on each host by issuing the rsh command
#       from marvin.  It starts up a number of collection scripts, one
#       at a time.  This script must be started before midnight on
#       December 6th.
#
#       The one_time.sh script collects static system configuration
#       information.  The top.sh script collects top info continuously
#       for 4 days.  The other scripts collect info from the various
#       stat commands (over a 24 hour period starting on Dec 7th).
#
#       A slightly different version of these stat scripts are started
#       on hosts that only collect data for an 11 hour period.
#
#       The nice command was used to run scripts at reduced priority.

# Collect static info about CDF's physical characteristics

nice ~/collection /one_time.sh &

# Top command runs continuously for four days

nice ~/collection/top.sh &

# Stat commands collect information for a 24 hour period on Dec 7th.
#  (they are later modified and restarted to collect info for Dec 9th.)

nice ~/collection/vmstat.sh &
nice ~/collection/iostatD.sh &
nice ~/collection/iostatd.sh &
nice ~/collection/netstat.sh &
nice ~/collection/pstat.sh &
nice ~/collection/nfsstat.sh &
```

Figure A3.1: Shell Script Used on Each Host to Start Collection Scripts

```
#! /usr/bin/csh -f

# Name: iostatD.sh
#
# Desc: This script collects iostat -D info on a host starting at
#       12:00 am on Tuesday December 7th.  This script must be
#       started before this time.  It collects iostat -D data at
#       20 second intervals over a 24 hour period.

# Tuesday's date (December 7th) and the name of the host is stored.

set  start_day = "07"
set host = `/bin/hostname | sed "s/.cdf//"`

# loop (checking new date every 15 seconds) until $start_day

while 1
   set get_day = `\date +%d`
   if ($get_day == $start_day) then
      break
   endif
   sleep 15
end

# Collect iostat data for 24 hours

/usr/bin/iostat -D 20 3782 > /tmp/$host.ioD

# Compress and remote copy the file to another account; then delete.

compress /tmp/$host.ioD
rcp /tmp/$host.ioD.Z \
    andria@univac.sys:/data/data2/andria/cdf/Tue_Dec_7/raw_iostat/.
rm /tmp/$host.ioD.Z
```

Figure A3.2: UNIX C-Shell Script for `iostat -D` Data Collection