Appendix A1

Appendix for Chapter 1

This chapter appendix provides additional information about some of the problems that we experienced in our study, resulting in our decision to abandon our original goal of simulating the model that we design in this thesis. Readers who are unfamiliar with the Sun OS operating system and its collection tools may wish to read Chapter 3 before continuing with the discussion in this chapter appendix.

A1.1 Data Analyst's Survival Guide

In this section, we provide additional information for each checklist item of the "data analyst's survival checklist" presented in Section 1.3.2. The subsection numbers in this section correspond to the checklist numbers in Section 1.3.2.

For each checklist item, we explain the possible consequences of not heeding the advice that we have suggested in the "data analyst's survival checklist." We explain the potential problem that each checklist item addresses and how it relates to our study. We classify each problem as "avoidable" or "nonavoidable," and suggest ways to avoid or to handle each problem. Our hope is that this discussion will provide useful information for other data analysts in this area.

A1.1.1 Understand all Command Options

$\surd$ *1. Read the manual pages of the data collection tools very carefully, and study all command options.*

In our study we did not collect information from the `iostat` command for four of the eight disks on marvin. Unless the user explicitly states for which disks the information is to be collected, the `iostat` command only provides information for the first four disks.

This information would be needed in a simulation to determine the load of each disk. Without the information for all of marvin's disks, there was no way to determine the percentages of requests that were scheduled on each disk, as the process accounting records do not indicate to which disk each I/O request was scheduled.

This problem could have been avoided by reading the manual page for the `iostat` command more carefully.

A1.1.2 Functionality of Hardware Components

$\surd$ *2. Determine all hardware components (such as caches, disks, I/O controllers) that exist in the system and understand their functionality (such that they could be simulated if required).*

We did not realize that marvin had a Prestoserve cache installed at the time of our data collection (because the `sysinfo` command did not support marvin's hardware). Without information about the frequency of the Prestoserve flushes and the hit rate of the Prestoserve, it is not possible to simulate the I/O requests to marvin's disks.

The problem of not collecting this information would have been avoided if we had had a better understarting of which hardware components were installed on marvin at the time of our data collection. The problem of simulating the Prestoserve, however, is still very difficult, as the data are flushed to specific disks at non-periodic intervals, and it is not possible to determine if a given read or write in the accounting records was satisfied by the Prestoserve cache or by a disk on marvin.

A1.1.3 Granularity of Measurement

$\surd$ *3. Study the granularity of measurements and determine if this level of granularity is appropriate for the study at hand.*

The interarrival time of jobs was determined by examining the time between the starting time of consecutive jobs in the process accounting records. As the starting time was only measured in seconds, jobs that arrived within a second of each other were recorded as having an interarrival time of zero seconds.

This problem is not avoidable if you wish to use the existing process accounting software. It may, however, be possible to modify the source code of the existing process accounting collection program, or to use other software to determine a finer granularity of the interarrival time between jobs.

A1.1.4 I/O Scheduling

√ *4. Collect information about to which disk(s) each I/O request should be scheduled in a simulation.*

The process accounting records do not provide information about to which disk(s) on which host(s) the I/O requests of each job should be scheduled. The number of disk blocks reported for a job in the process accounting records may have been used on the workstation's local disk or on one of the file server's disks.

If this level of detail is needed for the simulation, this problem is only avoidable by modifying the process accounting software. If this level of detail is not needed, it is possible to work around this problem by determining the approximate percentages of usage of each disk using the `iostat` command.

A1.1.5 Background Processes

√ *5. Determine if there are any background processes that contribute to the system workload, but have not been collected by the basic data collection tool that you are using.*

There were no process accounting records for long-running jobs that were started prior to the data collection period, but continued to run during and after the collection period. As the process accounting records were written out at the time of job completion, long-running jobs that did not complete during the data collection period were not recorded.

This problem is avoidable, but difficult to handle. The UNIX `ps` command, which can be used to provide a "snapshot" of all existing processes on a host, could be used to get a "snapshot" of the cumulative resource usage of these long-running commands at the end of the collection period. It may also be possible to force the accounting records to be written out (for all existing processes) at the end of the data collection period. The resource usage for these processes would then have to be extrapolated for the desired data collection period.

A1.1.6 Impact of Data Collection

√ *6. Consider the effect of your data collection on the observed system workload, and determine how to minimize and compensate for its effect.*

The processes that we created for our data collection must be taken into consideration in the analysis of the data. We minimized the effect of our data collection on the network by collecting our files to the `/tmp` partition of the workstations' local disks. Our data collection files were small enough that they did not interfere with the normal paging and swapping activity on these local disks.

Although the model that we design does not include our data collection jobs, the absence of these jobs must be compensated for when the model is validated. If you are using the system performance measures collected from tools, such as the `top` command, to validate your model, you must keep in mind that the workload produced by your data collection is incorporated into these performance measures, but will not be generated by the model.

A1.1.7 Limitations of Collection Tools

$\surd$ *7. If the data collection tools are known to have inaccuracy problems, determine if these limitations are acceptable for the purpose of the study at hand.*

The UNIX process accounting facility is known to have inaccuracies. Some of the problems that we noticed in the accounting records that we collected were that several records had been duplicated, and that the node-locked commands that were executed from the client workstations were recorded with unrealistic resource usage.

Another problem with the process accounting records is that variable sized disk blocks were not indicated. The disk blocks used are 8192 bytes in size, except for the last disk block that may be some multiple of 512 bytes. The process accounting records show only the number of disk blocks, and do not indicate the size of the last disk block.

The inaccuracy problems are not avoidable if you wish to use the provided process accounting software to collect your data. If the inaccuracy of the tools is not acceptable for your study, then you will need to use alternative tools or you will need to write your own tools.

As the accounting tool displayed its information using fixed-width fields, it did not properly display the resource usage of commands that had used very large resource amounts. We were able to avoid this problem by modifying the source code of the `acctcom` process accounting display program to print fields using wider columns.

A1.1.8 Resource Interaction

$\surd$ *8. Determine if the necessary level of resource interaction is represented by the data that will be collected.*

The process accounting records provide information about the total I/O and total CPU used by each process, but there is no indication of how these resources interact or the amount of user think time that may occur between resource requests. This problem is very difficult and its solution will depend upon the level of representation that is required for the study at hand. This problem can only be avoided if you wish to modify the process accounting collection software, or to use alternative data collection tools.

It may be possible to estimate the resource interaction based on side studies that examine the interleaving of the CPU and I/O requests. The user think time could also be estimated from experiments that examine the user interaction behaviour.

A1.1.9 Network Transfers

$\surd$ *9. Depending on the level of representation of the components in the model, it may be necessary to collect additional information that indicates where time is spent in network transfers.*

Consider a command that arrives on a client workstation that requires I/O from the NFS file server. Depending on the level of representation of the components in the model, in a simulation the time required to service this command may have several components, which could include:

- Total CPU time (on client)
- Total I/O time
- Ethernet transfer time
- Queueing time for resources
- Remote software overhead

The CPU and I/O time required can be determined from the process accounting records. The Ethernet transfer time can be determined using the packet size to be transferred and the known bandwidth of the Ethernet. The queueing time is the unknown time component that will be determined by the simulation. The remote software overhead time (i.e., the time required to package the request on the remote server) may, however, be difficult to determine. It may be necessary to perform packet timing experiments to determine this time component, if this level of detail is required for your model.

A1.1.10 Plan Carefully

$\surd$ *10. Spend a lot of time planning your study and carefully consider every element that might potentially be needed. When in doubt, it is better to collect too much data than not enough.*

Sometimes it becomes impossible to recollect missing data at a later time. In our case, both the hardware and the operating system software of the CDF system changed shortly after our data collection. This made it impossible to recollect missing information at a later time.

Problems of this nature can be avoided by planning the study carefully in advance and by collecting information about all aspects of the system that might possibly be needed. If you are unsure if you will need a particular piece of data, we suggest proceeding to collect it.

Some of the items that are important to collect for a distributed system study, but may be overlooked, are the password file (`/etc/passwd`) and the file of user login and logout times (`/usr/adm/lastlog`).