Chapter 7

Conclusions and Future Work

The work presented here represents an initial foray into the areas of workload characterization and modelling of distributed computing systems. In this thesis, we took the reader through the processes of data collection, workload characterization, and model design to provide insight into the tools, techniques, and methodology that can aid in the study of distributed systems.

The techniques presented in this thesis could be applied to other distributed computing systems, such as research or industrial systems. The workload characterization techniques used in this thesis to simplify a large set of workload data could be useful in a number of different types of studies, including system performance evaluation, workload management, capacity planning, and model design. The techniques described in this thesis are primarily intended for distributed file systems, but some aspects of the methodology may also benefit the study of centralized systems.

In this concluding chapter, we will summarise the work of this thesis in Section 7.1, by outlining the contents and contribution of each chapter in this thesis. In Section 7.2, we will discuss some general observations and methodology generalizations that we discovered in the process of our study. The final section in this chapter, Section 7.3, will suggest some direction for future work in this area. The appendix for this chapter, Appendix A7, contains a discussion of dynamic models.

7.1 Summary

In Chapter 1, we stated the original goal of this thesis, which was to study a hierarchical load sharing policy for a large-scale distributed academic computing system. We planned to design a model for a 71-host system that would be replicated to produce a model for an
Andrew-like large-scale distributed system [How88]. Due to complications with the data collected from our 71-host study system, however, it would not have been possible to validate a simulation of our model. Consequently, we outlined our intention to refocus our study on the workload characterization and model design of this 71-host system. We presented our goals to produce a static distribution-based workload model that would be suitable for a load sharing or a capacity planning study.

We explained some of the general terms and methodologies that have been used in workload characterization and model design in Chapter 2. We explored studies in the area of centralized computer systems and noted that these techniques can also be applied to distributed systems. We examined several distributed system model design papers, and observed the importance that all studies had placed on characterizing the system workload prior to their model design efforts. We investigated cluster analysis as a technique to aid in the model design process. As the literature showed that not much previous work had been done in the area of using clustering to investigate large amounts of “live” workload data, this provided motivation for our research.

In Chapter 3, we provided an overview of the 71-host CDF academic computing system that was used in our study and the data collection process. We collected data from the CDF system using existing data collection tools to extract both static and dynamic information about the system workload.

In Chapter 4, we studied the workload in the CDF system to determine the collection interval (Thursday December 9th 1993 from 1:00 pm to 5:00 pm) and the feature set (CPU and disk blocks) to be included in our model. Our examination of the workload revealed useful information that could be used by system administrators to improve the current operation of the CDF system. We also found that the CDF environment mainly comprised low resource usage jobs with short lifetimes that were not suitable for remote execution. We questioned whether load sharing in the CDF system would be of any practical benefit. We grouped the users in the CDF system according to their known function in the system and examined these groupings to see if they might constitute reasonable classes for a model. An examination of the variance within each of these classes suggested that alternative classification techniques may provide a better classification.

The work done in Chapter 4 contributes to the understanding of the characterization of massive amounts of “live” workload data. The general approach that was most useful was to
examine both typical and outlying hosts to get an understanding of the range of the overall workload. As well as identifying components for the model, the workload characterization in this chapter revealed information about the workload that could be utilized in model design decisions.

We further examined the workload in Chapter 5 using cluster analysis dissection. Our goal in this chapter was to determine if cluster analysis could be used to procure meaning from a large unprocessed set of data. We studied different clustering methods and used known information about the purpose of our study and the nature of our data set to choose the $k$-th nearest neighbour two-stage density clustering method to produce the clusters that we examined in this chapter. We summarised the resultant clusters in terms of their user, command, and host variability. We found that a particular user was more likely to give jobs in several different clusters, whereas a particular command did not occur in as many different clusters. We found that a particular client workstation contained jobs from a large set of the clusters.

Chapter 5 contributes to the understanding of how cluster analysis can be used to characterize large amounts of workload collected from distributed systems. A major finding in this chapter was that the methods that are most commonly chosen for clustering workload data may not always be suitable because they do not handle outliers well. We found that methods based on nonparametric density estimates produced clusters that could be analyzed to reveal information about the workload, and to make model design decisions.

In Chapter 6, we used a graphical tool to identify the periodic components in the workload. These components were extracted from the overall workload data set and the remaining data were passed to the clustering routines. We used the information about the workload that we learned in Chapter 5 to determine that our distribution-based model should include a number of user classes that sampled from a number of global command classes. Timing distributions would be used to model the arrival of these components in the model.

The first step was to identify classes of users for the model. As there were often large gaps of inactivity for particular users (which would skew the command interarrival time distributions), we used the knee criterion heuristic to determine the length of user inactivity that would be used to designate separate user sessions. In Section 6.4, we determined the set of user classes for the model by clustering on the average interarrival time between
commands, the average number of disk blocks used, and the average total CPU usage for each user session. The CV graph technique, which is based on minimizing the coefficient of variation for different numbers of cluster values, was used to determine the number of user classes that was included in our model.

In Section 6.5, we used cluster analysis to determine the global command classes for our model. The number of command classes was determined using information about the known use of the clusters in the model. For the resource data within each command cluster, we examined the chi-squared error of the fitted cumulative distribution function determined using the MLE parameters provided by SAS to determine a suitable number of command classes for our model. As often the centroids for clusters are incorrectly used to represent the resource usage of classes in a model, we examined the resource distributions for each command cluster; these distributions could be used to generate the resource usage values for the model.

We concluded Chapter 6 by examining the distributions of the timing elements of the model. We found that the command interarrival time distributions were approximately exponential. As these distributions had an overabundance of interarrival time values that were smaller than 1, fitting two separate exponential distribution functions to the data provided a better fit than using a single exponential distribution function.

The work done in Chapter 6 contributes to the understanding of techniques and tools that can be used for distributed system model design. One contribution is its demonstration of how a heuristic, borrowed from another area, can be put to use in this area. It furthers the understanding of how cluster analysis can be used to identify model components for a large distributed system. It also furthers the work in the area of determining suitable numbers of clusters (classes) to use in a model.

7.2 Recommendations

As stated in Chapter 1, a major contribution of the work in this thesis is the insight that it provides into the tools and methodology that can be used in the workload characterization and modelling of distributed computing systems. In this section, we make some general observations and recommendations about the tools and methodology, based on what we have learned by carrying out the study this thesis. These suggestions may further the understanding of others in the same area of research.
7.2.1 General Observations

It is important to understand the nature of the workload before becoming too deeply involved in the particulars of a study. Time and trouble can be saved by determining well in advance if the workload looks feasible for the goals of the study at hand. Had we discovered the lack of demand for load sharing in our study environment sooner, we may have decided to examine a system with a more intensive workload.

In distributed system workload characterization studies, simpler techniques may prove to be the most useful. In our initial attempts (which are not presented in this thesis) to classify the data for our model, we used complicated strategies that made it cumbersome to manage our large data set. Try to let the methods do as much of the work as possible. Simple techniques and tools that can analyze large portions of the data set with little supervision are most desirable.

The amount of data collected from distributed systems is often so large that there is no choice but to use automated tools. If the analyst has to carefully monitor every step, the amount of time required would make the task unmanageable. It is important, however, to keep control on automated routines. Although they may seem fine in principle, in practice they do not always work properly. Expert knowledge of the underlying structure should be used to guide and monitor the use of these automated routines.

We found that it is best to be as ruthless as possible when examining large sets of data. All data should be examined, as one never knows what is to be found. In particular, be diligent when examining extreme regions of operation and error conditions, as these may be most influential in your study. Examine the behaviour of outliers to determine their suitability with the tools being used, and to determine how or if they should be included in the model.

Do not limit your workload characterization study, as side investigations may uncover insightful information about other aspects of the current operation of the system. They may even show something about more urgent system problems that are independent of the model design intentions. For example, in Section A4.3, we discovered that a section of the Ethernet had higher than usual collision rates that may indicate a transceiver or connection problem.

We found that there was a lack of well-integrated all-encompassing tools for workload characterization and model design of distributed computing systems. We suggest that using
a variety of different tools may offer the diversity that is needed to deal with the wide range of issues in workload characterization and modelling.

Employ heuristics wherever possible, and do not be afraid to borrow from other areas. We made use of the knee criterion heuristic, which has more generally been used in virtual memory management policy design, to define the user sessions for our model.

Examine different types of distribution families to represent your data. Do not blindly assume that an exponential distribution, for example, is best in all situations. The resource distribution types for the command clusters that were produced in our study varied from cluster to cluster. Assumptions about the nature of distributions should not be made before examining the data extensively.

### 7.2.2 Clustering Generalities

Cluster analysis was a suitable tool for the analysis of the data that were collected from the CDF distributed system. It was particularly suitable because of its ability to process large amounts of data relatively quickly. Not only did cluster analysis work well in the identification of the components for our model (in Chapter 6), but it also helped to provide an understanding of the workload (in Chapter 5). As the clusters that were developed in the workload characterization process were similar to the clusters for the model, insight gained from studying these clusters provided insight into model design decisions.

The large number of different clustering methods that are now available complicates the task of choosing an appropriate method for a particular study. Expert knowledge of the system and the intended usage of the clusters should be used to aid in the selection of a method. Examine clustering methods of different types and choose the one that is most appropriate for the data type and purpose of the study.

Be sure to consider at least one density-based method when you are choosing a method to analyze a set of workload data. The density method is not as sensitive to extreme outliers, which are typical of workload data. The ability of the density method to create clusters that are approximately equal in size is particularly desirable for model design studies. When the final goal for the clusters is inclusion in a distribution-based model, each class must have sufficient data or else it may not be possible to find a distribution that fits the data well. A clustering method with equal-size bias suits this purpose well.

The issue of how many clusters should be chosen for a model is one that has received a
lot of attention; however, it is still primarily unanswered. We suggest that the known use of the clusters in the model should be used to govern the choice of the number of clusters.

7.3 Future Work

In this section, we examine some of the interesting areas of future research relating to the work done in this thesis. We briefly examine applications of the model, alternative workload studies, alternative validation techniques for cluster-based models, the use of dynamic models, and parallel simulation. The areas for future work suggested in this section are by no means an exhaustive list of the studies that might transpire from the work done in this thesis.

7.3.1 Applications of the Model

The most obvious area of future research for our study would be to use the techniques discussed in this thesis to implement a model using data from a system that could be simulated and validated. When collecting the data from the new study system, the suggestions made in Appendix A1 should be kept in mind. Some modification of the existing system software may be needed to ensure that all necessary data are collected at the required level of granularity.

The actual design of the simulator will not be straightforward. There are still issues, such as the amount of user think time, the interaction of resource usage, and the time spent in remote software, that will need to be addressed. The task of monitoring distributed systems is difficult in itself; so it is understandable that the numerous issues that must be considered in an actual simulation of a distributed system make the task exceptionally difficult. In implementing the simulator, it is important to focus on accurately representing the components that the workload characterization study has identified as important. In addition, the requirements for the key purpose(s) of the study should always be kept in mind.

Once validated, the model designed in this thesis could be used in a myriad of studies. The goal of designing our model to be flexible makes it useful for capacity planning studies. The workload could be increased by modifying the parameters in the model, and the effects of such variation could be studied. The model is particularly suitable for studying the impact that additional users or an intensified workload might have on the system.
Models similar to the one designed in this thesis could be used to test either initial-placement or migration load sharing policies. In an initial-placement study, the composition of jobs within each command cluster that are candidates for remote execution would be stochastically represented in the model. For a migration study, the resource usage accumulated by long running jobs could be examined should remote execution become necessary on a particular host.

Another potential application of our model is the hierarchical load sharing policy that we outlined in Chapter 1. As previously stated, a validated version of our 71-host model would be required before extrapolating the model to a large-scale distributed system. This extrapolation could be performed on a per sub-network basis.

Questions such as whether or not it would be desirable to share the load within only the local sub-networks, or within the entire large-scale system could be examined. The smaller amount of overhead required to maintain load information in only the local sub-network may or may not outweigh the advantages of implementing a global policy that is able to locate powerful compute centres on other sub-networks.

Another area of future research would be to compare the hierarchical load sharing policy to some of the existing load sharing policies for distributed systems. Examination of the Utopia load sharing software package [ZZWD91], which has previously been used in the CDF system, would make an interesting comparison study.

7.3.2 Alternative Systems and Workloads

The workload characterization analysis in Chapter 4 indicates that load sharing is unlikely to be merited in the CDF academic computing environment. There were very few jobs that would be candidates for remote placement, even during a period of high user activity. An area of future research would be to use the techniques discussed in this thesis to examine the workload of more resource-intensive environments, such as research or industrial environments.

A comparison study of the workload in several different types of distributed computing systems is another possible area for future research. As the sub-networks in existing distributed systems are unlikely to have homogeneous workloads (e.g., a particular distributed system for a company may have sub-networks for various purposes, which might include administration, research, and development), combining models made for several
different types of distributed systems may provide a more realistic large-scale environment to study. If load sharing were implemented in such an environment, a transfer of jobs from the resource-intensive to the lightly loaded sub-networks is likely to occur.

It would also be interesting to follow-up our study with a study of the workload in the current 1996 CDF system. As the current CDF system has undergone numerous hardware and software upgrades since 1993, a comparison of the current workload with the 1993 workload might prove insightful. Such a comparison could be used to examine questions, such as (1) “Are the system upgrades able to handle the increased new user workload that is a result of the new applications and the presence of world wide web access software?” and (2) “Is the bandwidth of the Ethernet still adequate, even with the increased requirements of the new network applications?” A detailed workload characterization study of the current CDF system may be able to answer these and other questions.

7.3.3 Alternative Validation Techniques

An alternative validation approach that uses a variation of the whitebox validation scheme that is outlined in Figure 2.4 in Chapter 2 could be investigated. The technique that we explain in this section could be used to gain confidence in models that are generated using cluster analysis.

In this validation scheme, cluster analysis would be applied to the actual job script data \( J \) collected from the study system \( S \) to produce a workload model \( W' \). Workload model \( W' \) would contain a number of clusters of known type, as well as information about the interaction and timing of the cluster components of the model. Representative distributions would be used to simulate the arrival times and resource requirements of jobs.

The workload model \( W' \) would be run to generate a synthetic job script \( J' \), which is then reclustered to produce a new set of clusters, which define the regenerated workload \( W'' \). This validation technique would then compare the characteristics of the regenerated clusters to the clusters in workload model \( W' \).

Although this technique is not able to determine if the performance indices are representative of those in the actual system, it does provide information about how well the interaction of the components in the model and the distributions that represent each component have been selected. It can generally be used to gain increased confidence in a model.

The difficulty involved in comparing the characteristics of the two sets of clusters is an
area that could use more research. One approach might be to ensure that the same number of clusters are generated in \( W' \) and \( W'' \), and then to compare the most similar set of cluster pairs.

### 7.3.4 Examine Dynamic Models

Although we have been careful to base the static workload model designed in this thesis on an interval that had consistently high load, it is often difficult to ensure the existence of steady-state conditions. As the unpredictable composition of workload often causes peaks having highly variable intensities and durations, higher order models for distributed computer systems is an area of future work that should be examined.

In the appendix for this chapter, Appendix A7, we outline a one-step transition matrix for the model designed in this thesis. As dynamic models are generally less compact than static models, a validated simulation should be used to determine if the extra overhead required to produce a dynamic model is worthwhile.

### 7.3.5 Parallel Simulation

The final area of research that we suggest, is the use of parallel or distributed simulation. In this type of simulation, some elements of the simulation are assigned to different hosts in a distributed system, or to different processors. Each processor works on a separate portion of the simulation in parallel, to reduce the overall time that is required for the simulation.

Parallel simulation is particularly suitable for simulations of complex large-scale distributed systems that would take a long time to run sequentially. In the hypothetical large-scale Andrew-like environment that we have outlined in this thesis, simulating each

![Figure 7.1: Alternative Whitebox Model Validation Scheme](image)
71-host sub-network on a separate processor is a logical approach. The task of communicating between the processors and properly synchronising the operation of each sub-system is a complicated area of future work.