December 9, 2000

CLIENT / SERVER REQUEST AND RESPONSE SIMULATION

Aaron Morris
CS522
Semester Project
Fall 2000

INTRODUCTION

This document outlines the functionality and makeup of a program written to simulate a client-server request and response environment. Requests from clients are routed through a central director object, which chooses an optimal server based on the number of requests being sent to each. The servers then are responsible for generating a file of random size and the director for successfully returning it to the source of the request.

The simulation tracks and reports statistics on the average response time, average file size, average idle servers, and average server load throughout the run. The simulation also has a GUI (graphical user interface) which allows the user to input many parameters affecting the situation being modeled.

CONTENTS

SYSTEM OVERVIEW

Basics

This system is a model of a client-server relationship, with the addition of a central routing object called a Director. The Director is responsible for receiving requests from client systems, and routing that request to what is to the best of its knowledge the optimal server for handling the request. The Director is then required to route the response from the Server back to the client who initiated the request.

User Interface

In order to create a flexible and useful simulation, the program includes a GUI (graphical user interface), which allows the user to change various parameters affecting the simulation run. Parameters such as the end of simulation time, the number of clients and servers, the speed of all the objects and also the request frequency all can radically change the results of a run. All output is also routed to the screen as well as a text file for the user to look at as needed, and track the simulation from start to finish.

Statistics

For a simulation to be useful, some statistics representing how the model performed are necessary. In this case, the user most likely will be interested in the average response time as well as the average server load and average idle servers per unit time. This will help show the relationship between response time and the number of servers required to meet that response time based on the given number of clients and their average request frequency. For perspective, the simulation also reports on the average file size sent back by the servers throughout the run.

Model

The model used will be that of an object-oriented, event-based simulation. The model strives to represent random requests and file retrievals as realistically as possible, given the scope of the project, and also to effectively route requests in a random but accurate fashion. The implementation details follow in the next section.