Design Project #1 - Web Server Replication

Introduction. One of the features of 6.033 is that we discuss real systems, both successful and unsuccessful. To get beyond discussion and give you some direct experience with designing systems, we also assign two design projects. As in the real world, these projects have (we hope) a simple high level problem statement but when you get into the system design you will find that there are hard choices to make. And, as in the real world, it is your job to explore, understand, and explain what the best choices are, how to reconcile sometimes conflicting goals, and how to keep the complexity of your design under control. This term, the first of these design projects relates to World-Wide Web technologies.

The Problem.

The ACME company has encountered a significant problem with the speed of their Web service. Their server currently consists of a single machine (www.ACME.com) in San Francisco connected to the Internet via a high speed link. They are currently serving serveral million documents per day. However, there are days when they serve significantly greater number of requests. The average number of requests has also been going up steadily since they started the Web site. In addition, they also expect a major influx of users when they start adding interesting dynamic content to their Web site. To add to ACME's problems, most users from abroad and some users in the U.S.A. have begun to complain about unacceptable response times. The are unsure whether their link to the Internet, the speed of their server and/or the bandwidth of the backbone Internet links are the cause of the poor performance. However, ACME does know that the bulk of their customers have upgraded to new fast cable and ADSL links and that the customers' Internet links are not the performance bottleneck.

To summarize their problems:

- The system must serve many requests each day.
- The number of requests served each day is growing quickly.
- People access their server from all over the world.
- The current server setup cannot support the additional load and provide adequate performance to all users.
- The bandwidths of the clients' Internet links are not a bottleneck.

A clever 6.033 student studies this problem and proposes solving it by creating identical copies of the Web server at various locations in the U.S. and around the world. Users will use the replica that best suits their needs and access data from it. This will make it easy to handle a large number of requests and ensure that all individuals get good response times. Easier said than done.

In the Web today, support for such mirror sites requires significant user intervention. Often, users are presented with a list of sites on a Web page and are asked to pick one. This is how the download sites for popular items such as Netscape browser are done. Other times, users must chose a server site by picking the correct server name (such as www.ibm.com, www.ibm.com.uk, www.ibm.com.ca, etc.).

Your job is to design a new automated Web server replica system. A user should be able to request the Web page http://www.ACME.com and your system should fetch the Web page from the replicated servers. Your primary goal is to maximize the number of requests that the entire system can handle while trying to minimize each individual's observed response time.

You may change clients, servers and/or protocols as needed. However, ease of deployment is an important factor in a good design. If your design changes significant portions of the Web infrastructure, you should provide a deployment plan as part of your design.

At a minimum, your design must meet the following requirements:

- It must provide a mechanism for directing requests to one of the server replicas.
- It must use a dynamic policy for picking a replica to serve a given request. It cannot serve all requests for an object from a preconfigured server(s).
- It must be easy for ACME to add new replicas to the system.
- It must increase the number of requests that ACME can handle and improve the response time of the users.

In addition you should answer the following questions in your report:

- Which part of your system (clients, servers, routers, DNS, etc.) chooses which replica to direct a request to? How does this part of the system learn of the available replicas?
- How will you choose among the different servers for an given request? Will this minimize the response time observed by individual clients? What metric(s) did you use to predict the performance of the Web transfer: network hop count? round trip times? or something else?
- Once your system has chosen a server for a request, how does it deliver this request to the correct server?
- How well will your system work once ACME starts hosting compute intensive dynamic web pages?
- What resource overheads does your system add to 1) the servers, 2) the network and 3) the clients? What would happen if every server site used your scheme?
- How does your system scale as the number of replicas increases? What will be the bottleneck in your system?
- How quickly does your system adapt if a server goes down or becomes extremely slow?
- What are the failure modes of your system? When failures occur in your system, are user requests not served or are they served more slowly?

You may also want to consider the following issues:

- Can anyone create a replica of www.ACME.com or must this be done by the administrators of ACME? We haven't discussed methods for achieving security yet, but you might point out in your report what the requirements are.
- How do you decide when and where to add a new replica to the system?

You do not need to address the following concerns in your design:

- The design does not need to work through network firewalls (if you happen to know what they are)

- In general, do not worry about solving security issues in your system.
- Often, protocol specifications do not match implementations exactly. You may assume that protocols do match the specifications.
- The design does not need to address how the Web server content is replicated or kept consistent. You may assume that some existing part of the system provides this service.

You should assume that HTTP/1.0 is currently used by the server and browsers. See http://web.mit.edu/rfc/rfc1945.txt for the specification. Athena users can access the RFC directly by the following command "attach rfc; more mit/rfc/rfc1945.txt". This RFC is very long and not all of it is relevant to this project. You may want to skim or only read parts of the document.

In addition, www.w3.org has a wealth of other information. You may also wish to look at www.netscape.com and www.microsoft.com for background information on their browsers. Also, since you are allowed to change protocols as necessary, you may incorporate any of the proposed extensions of HTTP or features of HTTP 1.1 in your solution.

The following reading will also be useful.

- The description of DNS by P. V. Mockapetris and K. Dunlap, Development of the Domain Name System, In Proceedings of SIGCOMM '88, Stanford, CA, August, 1988), and the Domain Names - Concepts and Facilities RFC (RFC 1034) (http://web.mit.edu/rfc/rfc1034.txt). These readings should help you understand how Web server names are currently resolved into specific machines.An optional additional reading is Section 7.2 of Computer Networks (Third Edition) by Tanenbaum (Note: This is different from the recommended Operating Systems textbook by the same author!),
- The description of TCP in V. Jacobson, Congestion Avoidance and Control, In Proc. ACM SIGCOMM '88 (Stanford, CA, August, 1988) and the TCP Congestion Control Internet-Draft (http://www.ietf.org/internet-drafts/draft-ietf-tcpimpl-cong-control-05.txt). These should help explain how TCP behaves and what network factors affect its performance. An optional additional reading is the description of TCP in Section 6.4 of Computer Networks (Third Edition) by Tanenbaum.
- The description of Cisco's DistributedDirector. This describes a commercial replica support system and should give you some ideas about where to start your design.

Your report. Your paper should be 8 to 10 pages in length. You should start by explaining to your intended audience the background of the problem in terms that the audience can understand. Next, describe why what you have decided are desirable properties of a solution. Then give your solution and explain how well it achieves (or fails to achieve) the desirable properties. Throughout the paper you should justify each of your design decisions, especially in relation to alternative decisions that you could have made. You will be more convincing if you say not just why your idea is good, but why it is better than the alternatives. (For example, if another approach would meet all of the objectives perfectly, but the cost would be 100 times higher, then you should mention that as a reason for choosing your less general but cheaper approach.)

Write for an audience consisting of colleagues who took 6.033 five years ago. That is, they understand the underlying system and network concepts and have a fair amount of experience applying them in

various situations, but they have not thought carefully about the particular problem you are dealing with. Assume that your paper will also be used to convince your friend's computer guru that you have the right idea. Finally, give enough detail that he can turn the project over to that guru for implementation with some confidence that you won't surprised by the result.

When evaluating your report, your instructor will be looking at both content and writing.

Content considerations:

- Does your solution actually address the stated problem?
- Do you explain your decisions and the trade-offs?
- How complex is your solution? Simple is better, yet sometimes simple won't do the job. But unnecessary complexity is bad.

Does your solution fit well with the rest of the system? If your solution requires modifying every piece of hardware, software, and data in sight, it won't be credible, unless you can come up with a very good story why everything needs to be changed.

How extensible is your design? Are there opportunities for later addition of desirable features that you decided to omit?

Writing considerations:

- Is the report easy to comprehend?
- Is it well organized and coherent? Does it start by summarizing the problem and your solution? (Remember, you aren't writing a mystery novel. Perhaps you should start with an abstract)
- Does it explain the approach or architecture conceptually before delving into details, components, and mechanics?
- Does it use diagrams where appropriate? (A frequent problem when people use word processors is that they try to express everything in words, either because the word processor doesn't make it easy to include diagrams, or they haven't ever learned how to use the drawing features. Pictures can communicate some ideas far better.)
- Does it use the concepts, models, and terminology introduced in 6.033? If not, does it have a good reason for using a different universe of discourse?
- Does it address the intended audience?

You can find other helpful suggestions on writing this kind of report in the M.I.T. Writing Program's on-line guide to writing Design and Feasibility Reports.

Phase Two writing considerations: If you are enrolled in the 6.033 writing practicum, you don't need to do anything special; your practicum instructor will explain how the report will get you credit for the Phase II writing requirement. If you are not enrolled in the practicum, AND you want us to forward your design project report to the writing program as your phase II writing project, please say so on the cover page, and make sure that your report is at least 8 pages long. Note also that the writing program has a rule that they will accept only reports that earn a B or better from the class in which they originate. Finally, be aware that the second design project will probably be a team project, and thus much more difficult to tailor to the needs of the writing program than this one.

Collaboration: This project is an individual effort. You are welcome to discuss the problem and ideas for solution with your friends, but if you include any of their ideas in your solution you should explicitly give them credit, and you should be the sole author of your report.

Schedule: Your report is due in recitation *Thursday, March 18, 1999*.