B

Dan, This proposal has good paragraph structure in general but it would be easier to read if the information order and connectivity between paragraphs were improved. For example, you evaluate the effectiveness of the design before you've fully described the design so your audience doesn't have the necessary context to understand the evaluation. If you were to move the 3rd ¶ to the end of the paper, ¶ 4 would immediately follow ¶ 2. Paragraph 2 ends by introducing queues and ¶ 4 is about queues, so this change would significantly improve connectivity. When you write DP1, take care to use good information order and connectivity both between sentences and between paragraphs. I also recommend including a diagram when you first describe the design.

6.033 **Dan Ports**

Design Project 1 Proposal

2004/02/26 R9: Karger TR1

A Concurrently-Staged Approach For Surveillance Monitoring

1 Architecture

This proposal describes the design for the central computer of a video surveillance system that monitors multiple video streams over the network and identifies and serves suspicious frames via the HTTP protocol. The principal design objectives for the system are fault isolation (particularly for subroutines known to be buggy) and scalability.

This implementation is based on the Staged Event-Driven Architecture (SEDA) model proposed by Welsh.[1] This architecture divides the processing of requests into stages. Each stage has a controller that maintains a queue of requests ("events") to be processed. The controller invokes an event handler process to handle each event; multiple handler processes for the same stage can operate concurrently. Individual stages only communicate with each other by inserting events into each other's event queue.

the diagram would be helpful here.

The SEDA architecture is well-suited for this problem, because processing camera data is naturally divided into stages. It allows all the major design objectives to be met. Fault isolation is ensured by having stages implemented as separate processes with their own address spaces, and tightly restricting inter-process communication. Moreover, if a handler process crashes, the stage controller can restart it. The design scales well, as the number of handler threads for each stage can be dynamically adjusted as needed. In particular, since queues are used at each stage, graceful degradation is possible: when the system is overloaded, the controller manages the queue, ensuring that events are processed in a "fair" order such that every camera receives attention.

Move this assessment to the end of the document, after the stages have been described

Stages in this architecture communicate by enqueuing events. Since stages never share address spaces, inter-process communication is necessary. This is achieved by having each stage controller process listen on an internal port; processes from other stages can connect to this port and submit a properly formatted event to be placed into the queue. Thus, the basic model for a controller process is an event loop that listens for new requests to be inserted into the queue, and removes events from the queue in an appropriate order and spawns the event handler processes.

redundant, combine these ¶s?

connection?

The SEDA design can achieve extremely high levels of performance through its use of concurrency. The number of event handler threads executing at each stage of the pipeline is dynamically adjusted based on the number of requests in the queue. If the length of the queue is beyond some threshold, a new thread is added at that stage, up to a maximum limit. If threads remain idle, they will be shut down.

2 Stages

The flow of data through the stages is shown in Figure 1. Video stream data is obtained from the cameras via a HTTP client. It is then decoded using the transcoder, and passed to the AI

[1] An Architecture for Highly Concurrent, Well-Conditioned Internet Services, Matt Welsh. Ph.D. Thesis, University of California, Berkeley, August 2002

3 Acknowledgements

The architecture for this implementation was based on the SEDA architecture developed by Matt Welsh as part of his Ph.D. thesis research; more information is available at
`http://www.eecs.harvard.edu/ mdw/proj/seda/`.

Austin Clements provided many helpful insights during the design process.

March 9, 2004

6.033 students,

Today you will receive your DP1 proposal with comments from the Writing Program. We hope that you find these comments helpful as you write your DP1 report.

Why did my proposal not receive an "A"?
The best proposals provided an overall approach to the design solution as well as design details in a well-written convincing manner. These proposals also provided a diagram to help illustrate how the various modules were integrated. These proposals did not simply repeat the assignment description.

How can I make my writing better for my design report?
Many of the design proposals provided some level of explanation of your approach. You will need to expand and clarify that explanation for your report. Also keep in mind the following:

- Restatements of the assignment are not meaningful to readers. Your readers already know the context of the project. Readers want to know about <u>your</u> design. The best reports explain the design and show how the proposed design is better than others.

Organization

- An effective design proposal begins by stating the purpose of the project and then providing a high level overview of the design. Remember that there is more than one approach to this assignment, so you need to summarize your approach to readers in the beginning.
- Subheads and chunking. It is easier for readers if you divide information into logical units with subheadings.
- Figures. Diagrams show a process. Arrows show direction. All figures need a number and a caption. Please computer-generate the figures for your report. Code is treated like a figure (and usually typeset in `Courier` font).
- Figure descriptions. Explain the figure to your readers by first describing the figure in the text; the figure should appear *after* the reference to it in text."
- Title. Give your design report a title.
- Smooth transitions between paragraphs and effective topic sentences are important in all writing. Smooth transitions eliminate ambiguity.

Style

- Verb tense. Use the simple present tense for your report. Future tense (e.g., the server will . . .) or conditional tense (e.g., the server should . . .) are not as effective.
- The word "this" always needs an antecedent.
- Jargon is inappropriate. Rather than "killing" a process, you terminate it.
- You may use "I" in your report, when appropriate. Over-use of personal pronouns, however, in technical writing sounds amateurish.

Citations

- Citations and acknowledgements. Cite your sources and make sure that you give credit to anyone who helped you "think through" your project.

To hear more about design report writing (and how your report will be graded), please come to the DP1 lecture on Friday, March 12th.

DP1 Writing Tutorials: Monday, March 15, 5:30 – 8:30pm, Tuesday, March 16 (pm), Wednesday, March 17 (pm). Please contact Mary Caulfield <u>mcaulf@mit.edu</u> for an appointment.

For further questions about the Writing Program, please contact Mya Poe (myapoe@mit.edu)

Regards, the 6.033 writing instructors
Atissa Banuazizi, Karen Boiko, Mary Caulfield, Kim DeVries, Les Perelman, Mya Poe, Susan Ruff, Amy Yu, and Mary Zoll