The Art of Computer Systems Performance Analysis; Techniques for Experimental Design, Measurement, Simulation and Modelling

by Raj Jain

John Wiley and Sons, 1991, ISBN 0471-50336-3

Reviewed by Allison Mankin, The MITRE Corporation

I read Raj Jain's new book *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling* while doing the final work on a performance evaluation project. Note the word, ''read.'' This is an unusual object, a textbook that one wants to sit down and peruse. The prose is clear and fluent, but more important, it is witty. Wit is a good strategy in a textbook, because a joke that uses knowledge the reader has just gained reinforces the learning as well as the enjoyment of the sub- ject. The book is also practical. I found it was easy to relate its treatments of topics such as queue modelling and experiment factors to my hands-on work.

If you are doing a performance or systems engineering analysis for any kind of project, you will appreciate the list of ''Reasons for Not Accepting the Results of Analysis'' (10.8). Here's a sample:

> Reason 3: It improves performance only for long IOs/packets/jobs/files, and most of the IOs/packets/jobs/files are short.
>
> Reason 4: It improves performance only for short IOs/packets/jobs/files, and who cares for the performance of short IOs/packets/jobs/files, it's the long ones that impact the system.
>
> Reason 5: It needs too much memory/CPU/bandwidth and memory/CPU/bandwidth isn't free.
>
> Reason 6: It only saves us memory/CPU/bandwidth and memory/CPU/bandwidth is cheap.
>
> Reason 15: This will violate the IEEE, ANSI, CCITT, or ISO standard.
>
> Reason 17: The standard says nothing about this and so it must not be important.

With rueful humor, born no doubt of long (and successful) experience persuading people to accept analyses, Jain suggests that the list can be used to show reviewers that all of their arguments are old, or to help reviewers critique an analysis that is competing with yours. I also imagined handing the list out to meeting attendees to cut hours off of almost any systems analysis meeting. This list appears in a ''Box'' at the end of its chapter. The Boxes differ from the ''soapboxes'' found in Rose's *The Open Book*; Rose's boxes break the text for flames, Jain's complete the text and help collect the reader's thoughts.

The major sections of the book are: measurement and workload techniques, probability and statistics, experimental design, simulation, and queueing models. There is also an introductory section that sets the tone for the others by featuring the mistakes often made in performance analysis. As you read on, there are many dissections of mistakes, myths, and games. Take, for instance, the section, ''Setting Performance Requirements'' (3.5). The acronym SMART summarizes what performance specifications should be, though they rarely are: Specific, Measurable, Acceptable, Realizable, and Thorough. This is not a surpris- ing summary, but it's worth being reminded that the following example fails on at least four out of the five:

> There should be an extremely low probability that the network will duplicate a packet, deliver a packet to the wrong destination, or change the data in a packet.

Benchmarking pitfalls are well treated also. Some of these are honest mistakes; for instance, if you write a benchmark program which runs a random set of I/O or network requests, it will not accurately represent the system's performance for multiple requests in succession to one destination Others are deliberate and are usefully anatomized. Further along these lines are discussions of myths about random number generation, and common mistakes in simulation and experiment design. I do not want to give away punchlines, so I recommend reading the book.

Jain works in the Distributed Systems and Performance Group at Digital Equipment Corporation. He specializes in networking, and has done important work on congestion management, OSI protocols, traffic characterization and FDDI performance. The book grew from a graduate course that Jain teaches at the Massachusetts Institute of Technology. The reader benefits, because a number of the well-explained examples come from projects carried out by Jain's students. This adds depth to the illustrative material. The examples work well to explain the more difficult points in the text because they are substantive, and, of course, because they come from real people who have used the text. In contrast to most other textbooks on performance, there are as many examples on networked computers as single computer systems.

A few minor areas are less fully developed than the rest. The chapter on monitoring techniques implies that there has been few changes in the technology since the late 1970's, Distributed systems monitoring is presented only sketchily. I would like to see more on the barely mentioned ''implicit spying'' technique (measurement by tapping a bus or network) and on issues such as clock synchonization and hybrid measurement approaches. Similarly, the book presents Remote Terminal Emulation (a technique that generates the workload of terminal users for testing another computer) in some detail. It observes only in passing that local area networking, workstations, and X-Windows have greatly changed the way that people now use time-shared computers. It would be nice to read some pointers on how to modify the old Remote Terminal Emulation approach for the new environment. But the book covers a large amount of territory, both well-established and new, so a few less-developed areas are inevitable.

The book covers its statistical and analytical topics lucidly. The coverage is full enough to define network, operating systems, and systems engineering problems equally well. An introductory knowledge of probability theory and the motivation to work through examples are enough mathematical preparation for these. Many useful techniques here are made easier to apply by good descriptions of visual diagnostic tests. I found this especially useful in the chapters on the analysis of variance. Experiment design, approximations, analytical operating rules, and much more, are presented accessibly. Jain has prepared fine summaries of the mathematical results, which seem to me to be better organized and more broken out than those in Kleinrock's *Queueing Systems*, for example. The summaries are another use of the Boxes.

The Art of Computer Systems Performance Analysis, by Raj Jain, is a practical, rich and enjoyable addition to the library of anyone whose work includes performance evaluation, simulation of systems, or system testing.