

Apdex Finds More Valuable Information



Don't rely on reported averages to ascertain what kind of performance users are getting

In the APM (application performance management) best practices survey documented in this issue (see pp. 22–27), Rebecca Wetzel and I found a strong correlation between good communication about application performance and actually delivering good performance. The top group excelled at communication relative to the other best practices. Accordingly, this column explores various ways to communicate information about application performance.

First we must start with application performance measurements. NetForecast has been conducting ongoing measurements of the Apdex Alliance Contributing Members' websites (there are currently 10 members). We are using a synthetic agent service that gets the home page of each website from various locations around the world every 15 minutes.

For the purposes of this article, we are limiting the discussion to the data gathered from 5 locations in the U.S.: California, Colorado, Florida, Minnesota and New York.

The data was grouped so as to track the overall performance that users in each of the states would experience as they tried to access the 10 sites. The

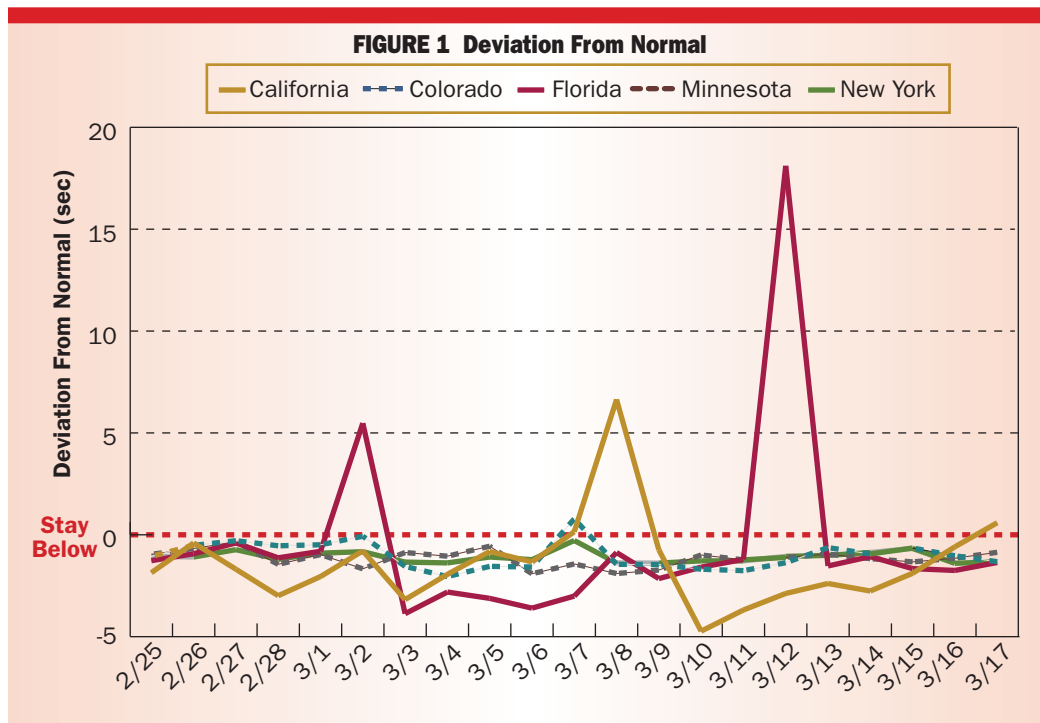
tests simulate a business-to-business ecommerce type of interaction. The synthetic agents simulate such a user because they are generally well wired to the Internet, with access lines of 100 Mbps.

Readers should not focus on the actual 10 websites the testing service used for the experiment, or even the choice of the 5 "user" locations. The quality of the experiment is sufficient to have a good data set that represents Internet performance over the three-week period of February 25 to March 17. The purpose of this article is to discuss what one can now do with the data set.

We developed three types of reports which represent three distinct ways by which the person developing the report interacts with his enterprise.

Deviation From Normal

The simplest way to report the data is to not have any conversation with the application users or management. Let the data report on itself. The premise is simple. Figure out what is the typical response time for each region, and then report days that were significantly slower than normal as poor performance. Some measurement tools have this feature built in as a standard report.



We start by averaging all the measurements per region per day. Then we perform a five-day rolling average of daily performance in each region. The algorithm next adds 50 percent to the rolling average, which becomes the good/poor performance threshold. Each day's average is then compared to the automated threshold.

The result shown in Figure 1 is the delta from the threshold for each day, where negative values are the time in seconds that the day has performed below the threshold.

For example, if the recent rolling average is 4 seconds, then the threshold is 6 seconds (i.e., increasing the 4-second average by 50 percent, as explained above). If today's average performance was 5 seconds, then the result shows that the day was 1 second below the threshold, or negative-1

second. Below-zero scores are good; scores that are above zero indicate poor performance relative to the norm.

Figure 1 shows that over the three weeks, performance was generally good across all the regions except for three specific incidents. Florida saw poor performance on March 2 and March 12 while California was hit on March 8.

Average With Threshold

Let us now become more proactive, engaging some members of corporate management in the process, and figure out a specific threshold that represents the good/poor performance boundary in this scenario. In this case, we applied the target of 4 seconds. It turns out that on a good day, the average from all regions to all sites is just over

Measuring deviation from normal or average readings will miss a major part of the problem

FIGURE 2 Average With Threshold

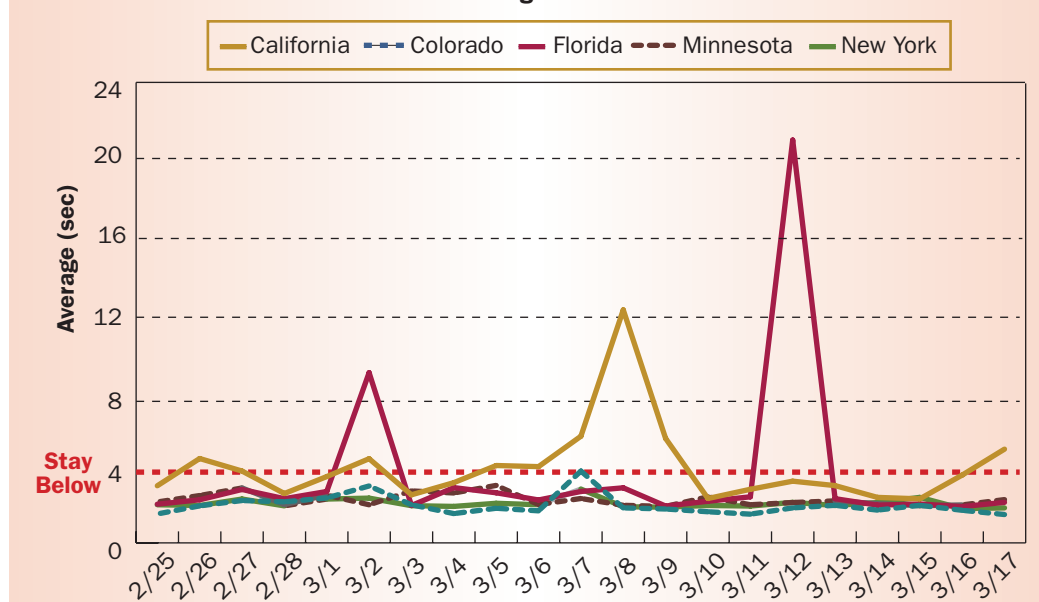
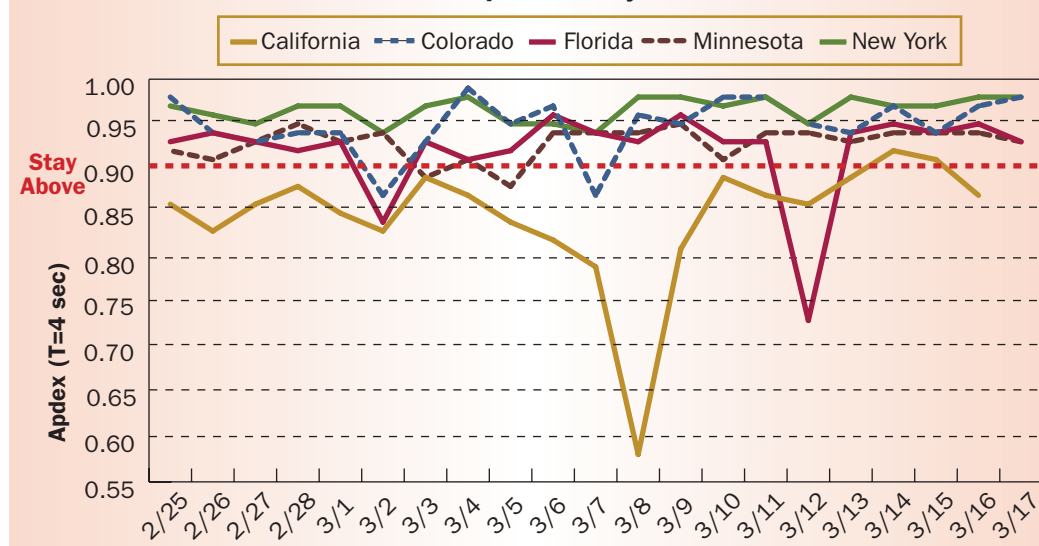


FIGURE 3 Apex With Objective



Looking only at averages, you miss bad performance that users find frustrating

2 seconds. The average for all days (good and bad) is 2.9 seconds.

So the results are what one would generally see in a B-B type of interaction; several organizations have recently stated that such interactions should stay within 4 seconds. Let us assume that the person who is generating these reports has an informed conversation with the business manager who cares about this application, and they agree on a 4-second response time target.

Figure 2 shows the exact same averages per region per day as used in the first report. The agreed-upon target of 4 seconds is clearly marked. Any region-day below the 4 seconds is operating within the service level agreement.

In this case, the result is remarkably similar to Figure 1. The same three days show poor performance being delivered to the same regions. The only new information is that California goes a little above the threshold a few times.

Apdex With Objective

The more sophisticated dialogue between the IT staff and business managers would be to apply the Apdex methodology (see www.apdex.org). Here the conversation has two parts.

First there needs to be an agreement on the Apdex performance target called *T*. *T* is the threshold between satisfied and tolerating performance as seen by the user. Let us assume that the 4 seconds agreed to above is used here as well.

Now the second part of the conversation. Given that the performance when reported by Apdex will have a range of 1 (excellent) to 0 (disaster), what should be the performance objective? The Apdex specification says that an Apdex score of 0.94 to 1.00 is excellent, while 0.85 to 0.94 represents good performance delivery. Let us assume that the new service level agreement using Apdex is for a target of 4 seconds delivered with an Apdex score of 0.90 or better.

Figure 3 shows the exact same data gathered over the three weeks using the Apdex methodology. In this chart, better scores are up rather than down as in the previous charts. So the poor-performing region-days are those below 0.9.

This view of the same data shows strikingly different information. The first observation is that the users in California are seeing chronically poor performance.

So what happens when IT starts getting phone calls from the users or managers in California complaining about performance? If the IT department were using either of the previous two averaging methods of reporting, they would have shrugged off the complaints with, “I don’t know what they are talking about, it all looks normal from here.” But if they were using Apdex reporting, then they would have been working on the California problem already, since it is clearly below the SLA.

The other interesting result is that while the two averaging methods both reported the worst incident as Florida on March 12, the actual worst incident, as reported by Apdex, was California on March 8. The performance in California on March 8 was so bad that the Apdex score dropped to nearly 0.5, which is considered unacceptable! This is a completely different conclusion than found by the averaging methods. Furthermore, Colorado and Minnesota saw some poor performance days that went undiscovered in the averaging methods.

Figure 4 shows the three approaches’ ability to find poor performance days. Deviation from Norms found 6, Averages with a Threshold found 10, and Apdex found 25—Apdex found more than twice as many performance incidents than the averaging methods. Apdex finds more of them because it finds the “tolerating” and “frustrated” user sets. Averaging washed these groups away.

Conclusion

The conclusion is clear. If you want to really find out what is going on with user performance, you should process the numbers through the Apdex formula. And if you want to have a good meaningful dialogue with business managers about application performance that is linked to their business needs, then you should base that dialogue on the Apdex methodology.

The Apdex approach is free for the using. Anyone can export their numbers into an Excel spreadsheet and produce the charts shown here for their network.

Peter Sevcik is president of NetForecast and is a leading authority on Internet traffic, performance and technology. Peter has contributed to the design of more than 100 networks, including the Internet, and holds the patent on application response-time prediction. He can be reached at peter@netforecast.com.

