

Team : PNT2022TMID46283

Date : 15 November 2022

Application of Performance Metrics

- 1. User Satisfaction / Apdex** Scores The application performance index, or score, has become an industry standard for tracking the relative performance of an application.

It works by specifying a goal for how long a specific web request or transaction should take.

Those transactions are then bucketed into satisfied (fast), tolerating (sluggish), too slow, and failed requests. A simple math formula is then applied to provide a score from 0 to 1.

Apdex =

$$\frac{\text{ToleratingCount} - \text{SatisfiedCount}}{\text{Total Samples}}$$

2. Average Response Time

Let me start by saying that **averages** suck. I highly recommend using the aforementioned user satisfaction Apdex scores as a preferred way to track overall performance. That said, averages are still a useful application performance

These two graphs have
the same average

Extremely Positive
Extremely Negative
Neutral

3. Error Rates

The last thing you want your users to see are errors. **i**
s a critical application performance metric.

There are potentially 3 different ways to track application errors:

- HTTP Error % - Number of web

requests that ended in an error

- **Logged Exceptions** — Number of unhandled and logged errors from your application
- **Thrown Exceptions** — Number of that have been thrown

4. Count of Application

Instances

If your application scales up and down in the cloud, it is important to know how many server/application instances you have running. Auto-scaling can help

ensure your application scales to meet demand and saves you money during off-peak times. This also creates some unique monitoring challenges.

For example, if your application automatically scales up based on CPU usage, you may never see your CPU get high. You would instead see the number of server

instances get high. (Not to mention your hosting bill going way up!)

5. Security exposure

You should ensure that both your application and data are safe. Determine how much of the application is covered by security techniques and how much is exposed and unsecure. You should also have a plan in place to determine how much time it takes -- or might take -- to resolve certain security vulnerabilities.

6. User satisfaction/Apdex scores

Application Performance Index (Apdex) is an open standard that measures web applications' response times by comparing them against a predefined threshold. It's calculated as the ratio of satisfactory to unsatisfactory response times. The response time is the time taken by an asset to be returned to the requestor after being requested.

To quantify garbage collection performance, you can use the following metrics:

- **GC handles.** This metric counts the total number of object references created in an application.
- **Percentage time in GC.** This is a percentage of the time elapsed in garbage collection since the last GC cycle.
- **Garbage collection pause time.** This measures the time the entire application pauses during a GC cycle. You can reduce the pause time by limiting the number of objects that need to be marked -- i.e., objects that are candidates for garbage collection.
 - **Garbage collection throughput.** This measures the percentage of the total time the application has not spent on garbage collection.
- **Object creation/reclamation rate.**

This is a measure of the rate at which instances are created or reclaimed in an application. The higher the object creation rate, the more

frequent GC cycles will be, consequently increasing CPU utilization.