

Performance Engineering Cookbook

Ingredients for Performance and Capacity Success

8. Transaction Structures: the part is not the whole

This is a series of brief articles explaining the basic concepts of systems performance and capacity planning. Motivated by the Computer Measurement Group, these concepts are applicable to IT systems and beyond.

Transaction Structures and Performance

In an earlier note on response times, we mentioned that not all transactions are atomic, or are viewed in the same way by all stakeholders. Whatever is perceived by the users or consumers as a single transaction is almost often the result of the integration of a series of more detailed transactions by one or more service providers.

As an example, consider fetching a webpage. From the perspective of the web user, it starts with a click and results in a webpage being displayed. Under the hood, however, the following steps are executed. The request is transmitted over a network to the web server, the web server sends a page back over the network to the browser, and the browser renders the page to the display. All these steps are executed by different providers, and there is hardly a provider in full control.

Breaking down the transaction, and figuring out the responsible parties for all sub transactions, is key to managing the overall capacity and performance of the transaction. An effective technique for finding the optimal breakdown is to match it with the breakdown in responsibilities over different contributors, often different service or technology providers. In our example, the parties are: the network provider, the web server, and the code that runs on the front-end (web browser).

The overall transaction can only be made to have adequate performance if there is an overall integrator. If there is not a formally assigned integrator, this job will fall into the user's lap. That is not always good, as these users and their managers are not necessarily qualified to fill that role. So it is important to find a committed and competent integrator.

Now that you have understood the basic idea of transactions, you are ready for some of the complications. The simplest one is that each sub transaction can itself be broken down into sub-sub transactions. The web server breaks down into an application server and a database server, the database server breaks down into processing and storage. This of course could mirror the breakdown in the organization by support group or skill set.

Another complication is that sub transactions can be done in parallel. In our example, a typical web page contains multiple elements, which can be retrieved simultaneously. Properly understanding the capacity and performance implications is a bit more complicated.

Common ways of visualizing structured transactions and their components are time-sequence diagrams and waterfall diagrams. See the link farm for more information.

Link farm

Wikipedia: http://en.wikipedia.org/wiki/Parallel_computing
http://en.wikipedia.org/wiki/Sequence_diagram
<http://www.webperformancetoday.com/2010/07/09/waterfalls-101/>