Article 1 of the series How to apply performance modeling
Applications have performance-DNA or do they?

The purpose of performance modelling is to explain the performance of applications and to predict their performance in varying circumstances. Predicting application performance takes knowledge about its performance-DNA. Every application has its unique performance-DNA. What does it look like and why is it interesting?

Many applications are just a set of transactions that subsequently make use of resources, both hardware and software. A transaction is an action of the user with a response time. For each resource we need to know how long it is engaged by the application in servicing the transaction, i.e. for each resource in the hardware chain that the transaction passes. In other words the transaction has a service demand for each resource. The complete series of service demands for one transaction is called a transaction profile. The complete set of all transaction profiles is called the application profile, or as I like to call it the performance-DNA of the application.

Example: configuration shown in figure 1.

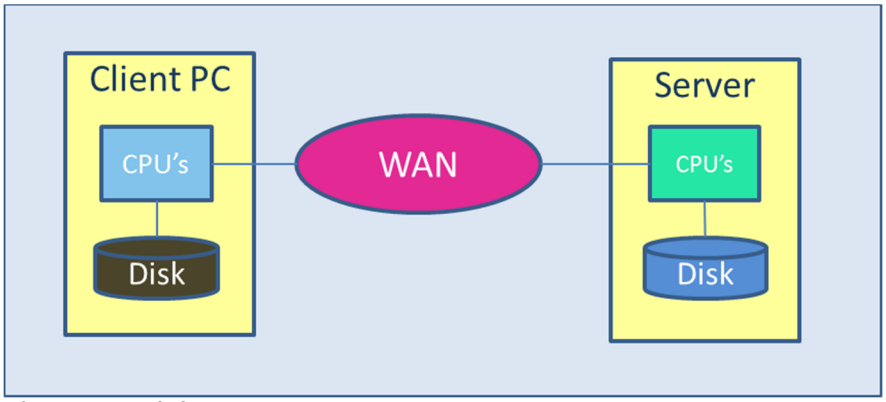

Figure 1: An infrastructure

The next figure depicts in a simplified manner the way time is spent by transactions being processed by the system.

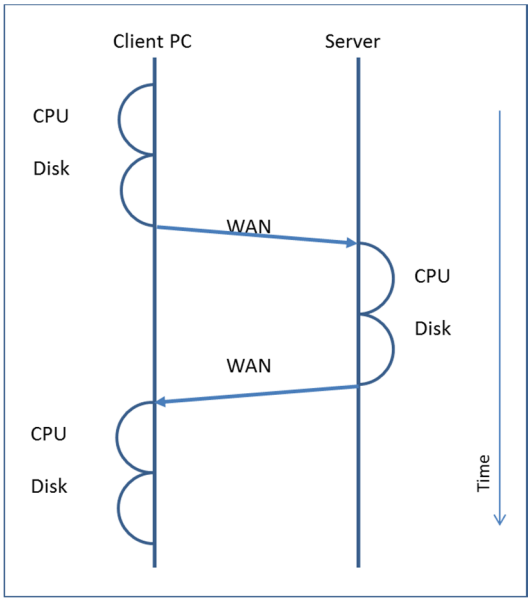

Figure 2: The time spent to process a transaction

The coloured horizontal line in Figure 3 shows a summary of how much time the transaction takes of each of the resources of the infrastructure in Figure 1. Notice that the colours of the horizontal line correspond with the components of the infrastructure shown in Figure 1. Each of the transactions may have such a transaction profile with varying values. The whole set of transaction profiles for all transactions of the application is called the application profile, or as I like to call it: the application's **performance-DNA**.

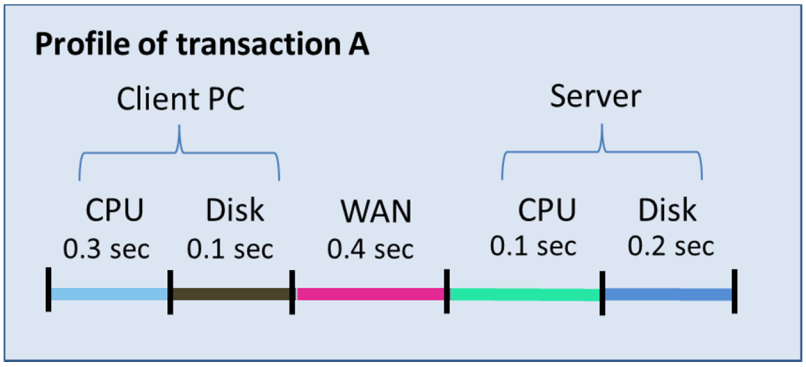

Figure 3: Profile of one transaction

The transaction profile shows by the total length of the horizontal bar the single user response time of the transaction, which is the lowest possible value of the response time of that transaction.
When the application is used in a high transaction volume its performance-DNA remains invariant. However that does not mean that the response times remain the same. They will very likely increase. We will later see how that can be explained and estimated.

It is practical to visualise the transaction profiles in a graphical way. Therefore the layout of Figure 3 is reshaped into Figure 4. Again the length of the bar corresponds with the response time and the colours correspond with the colours of the infrastructure components of Figure 1. An application may have dozens even thousands of different transactions, so its performance-DNA may contain a stack of bars like the one in Figure 4. In most of the cases we only deal with a subset of the transactions when we analyse its performance with the model.

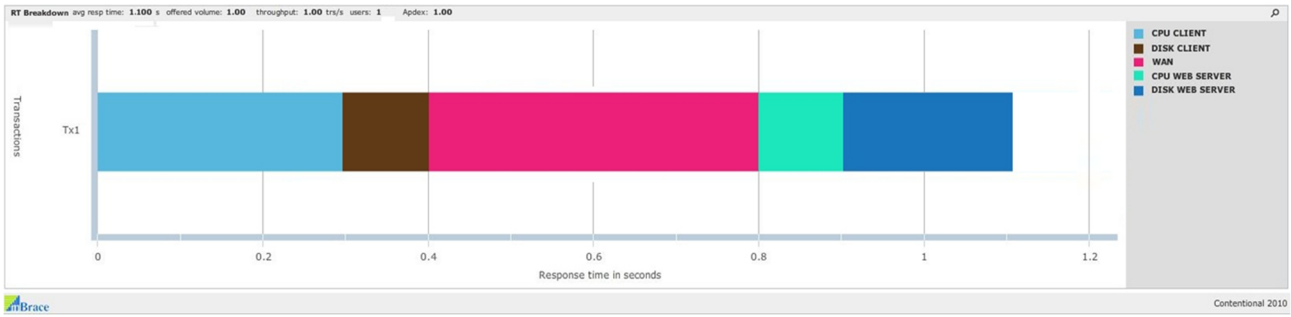

Figure 4: Each application has performance-DNA and for each application this DNA may look different.

The next three examples of varying performance-DNA are shown.

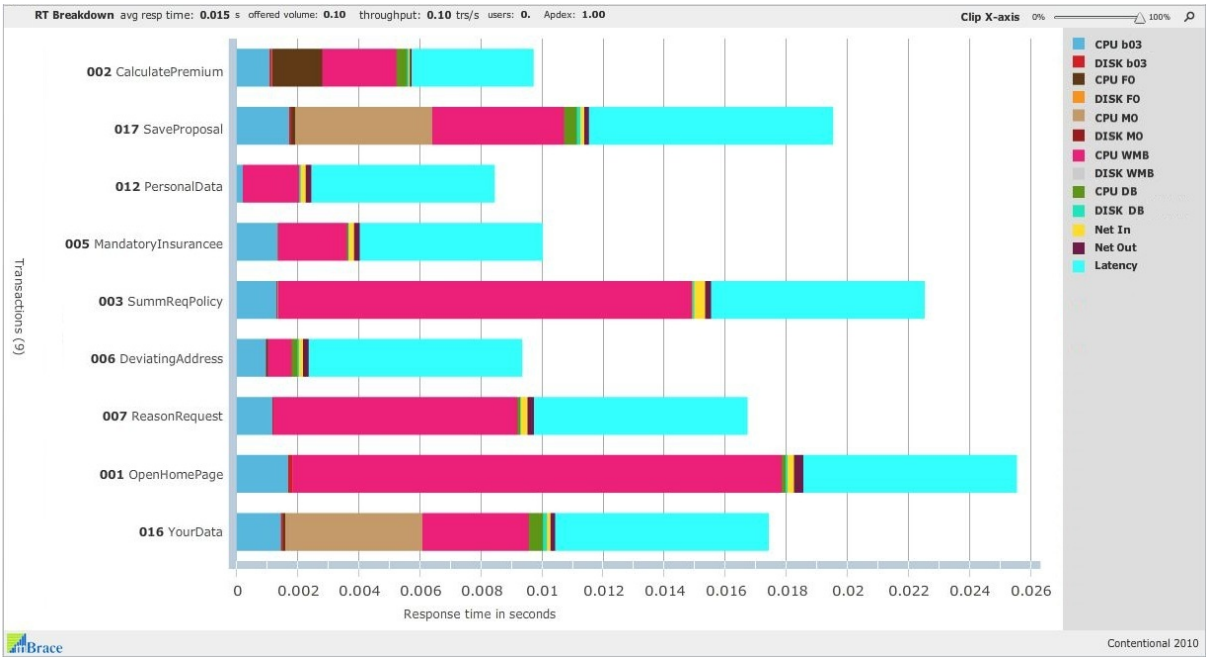

Figure 5: Light weight application with server side response times in the range of 3 to 40 msec.

Each bar shows one transaction profile. The legend shows that there are 5 different CPU's involved, which means that there are 5 servers involved. The Client PC is not included. So in this case the performance-DNA is not entirely complete. For each server the service demands for CPU and disk are included as well as the network. Network includes the WAN and all LAN's, split in inbound and outbound. This example shows extreme short service demands.

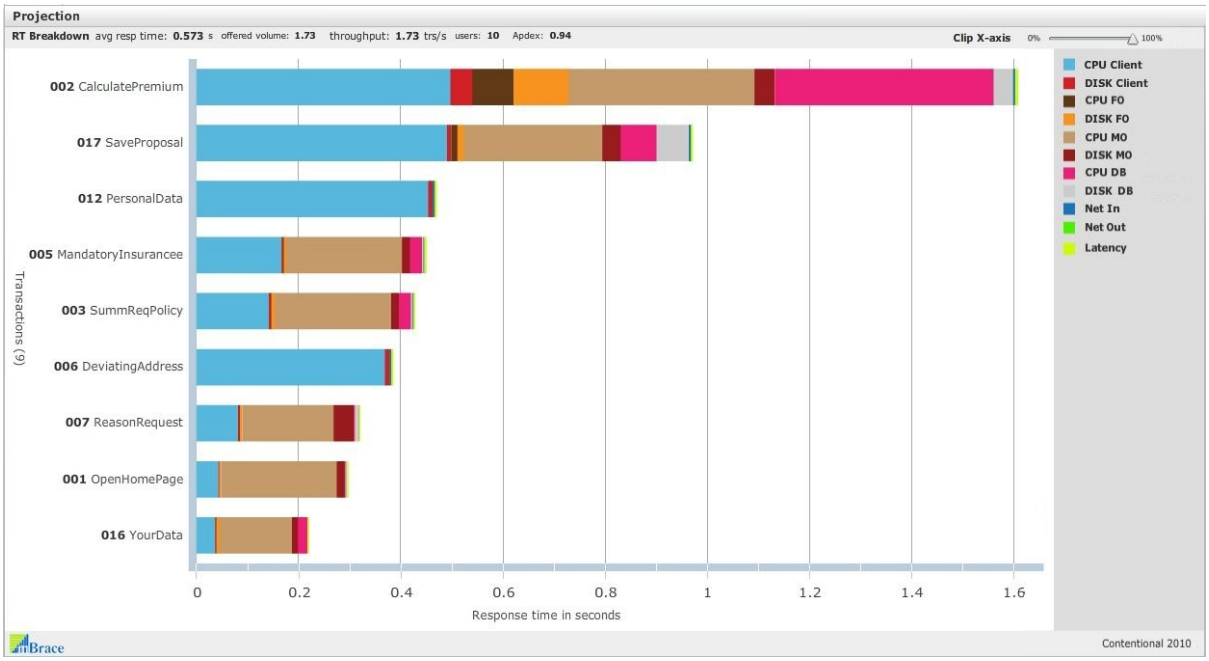

Figure 6: An ordinary web application with ordinary response times

Figure 6 shows the performance-DNA of an ordinary web application sorted by decreasing response times. The application runs on an infrastructure with 3 servers. The Client PC is included in the transaction profiles. Transactions 012 and 006 are executed on the Client PC only. The others use the full chain.

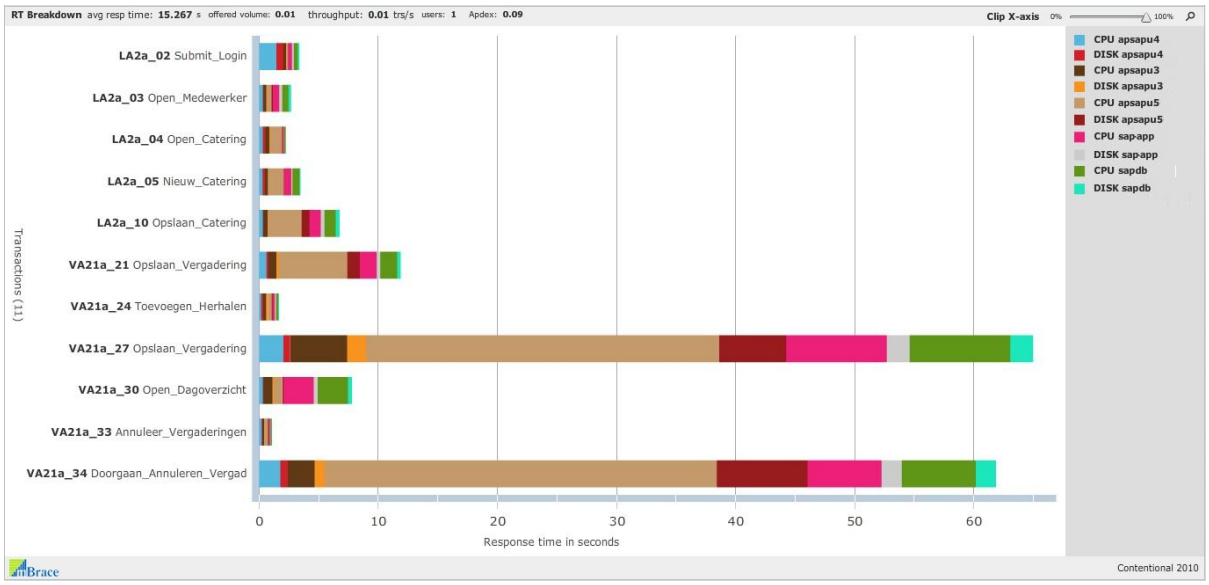

Figure 7: An application with long response times.

Figure 7 shows an application with transaction response times varying from 1 up to 70 seconds. The infrastructure has 5 servers. The Client PC and networks are not included in the profiles. In this example the transaction profiles are sorted by process sequence rather than decreasing response times.

In the above examples only subsets of 9 or 11 transactions are shown for the sake of legibility. The numbers of transactions that applications may have vary from dozens to thousands. It is pragmatic to limit the number of transactions in the performance-DNA when analysing application performance. In practice only the transactions that are used in 95% of the transaction volume are taken and in many cases that is a surprisingly small number. Over the life span of an application new transactions may be added and existing ones may be changed. This can make it necessary to repeat measurements to update the performance DNA as a sort of maintenance.

Obtaining a performance-DNA is not trivial because our industry does not support the concept of transaction. It takes bizarre measurement procedures to collect all elements of the transaction profiles. On many occasions costs are cut by leaving out parts of the profile. The choice of service demands measured may be determined by a risk assessment. At least the elements should be included that are needed to analyse the performance aspects with high risk. As an example the service demands of the Client-PC are often omitted to save time and money.

It is important to bear in mind that when the application software changes its performance-DNA changes too. So the transaction profiles have to be taken by measurement again each time the software of the transaction is changed.

Once we have succeeded in capturing the application's performance-DNA the world lies open for analysis of the application's performance.

The application's performance-DNA is invariant a great deal. It will show the same picture in many varying circumstances. E.g. the performance-DNA remains the same when the transaction volume increases. The exception is with vertical scaling of resources, which will be treated in the third article of this series.

A transaction profile shows the response time for a single user. A single user does not cause any queuing. So the performance-DNA shows the minimal response times for the application without waiting times. When the transaction volume increases, the response times may increase due to queuing, but the performance-DNA remains the same. Waiting times may come up for each of the resources. So if we want to explicate why a response time is as it appears we first observe its performance-DNA. In many cases that tells the tale on poor performance already.

When we want to predict how much the application will load the resources we have to fill in two more items: usage volume (number of users, transactions per second) and hardware capacities.

Commonly the term workload characterisation refers to determining both the performance-DNA and the usage volume of the application.

The next article will outline how usage volume is handled.

Any questions, debate, criticism? Email: michael.kok@mbrace.it.