Article 2
Applications have a Usage Volume Too
Michael Kok

In the previous article we saw that applications have performance-DNA, which tells us how heavy each transaction charges the hardware components and how long it takes to execute them. In this edition we are going to have a look at the usage volume, the amount of use that is made of the application by the users. When the usage volume and the performance-DNA are multiplied we get a view on the load on the resources as a result. This allows us to determine the impact of the application on the hardware and software resources in our quest to predict application performance.

Transaction types and transactions

In order to determine the usage volume, transactions must be counted in some way. So we need to know what they are. Defining the concept of transaction can be complicated, but here is the pragmatic version: *a transaction is some basic action of the user with the application that has a response time*. A transaction type is a specific transaction with a name. *Transaction type* is the generic thing, the *transaction* is the occurrence. For analyzing application performance we want to know how its transaction types perform and how they blend into the total volume of all users, rather than how the whole unnamed lot performs. So we would like to be able to count them by name. Out of hundreds of transaction types most of them can be OK and three may perform poorly. Then of course we want the names of those three. So inevitably we want to see the transaction types in the model.

Usage volume and the load on resources

Once we know the transaction rates and the performance-DNA we can determine the load on the resources. The next two figures give an example of combining usage volume and performance-DNA to determine the load on the resources. They show the load of an application with two transaction types named Tx01 and Tx02 on the CPU's of one server. The figures show three sections:

1. Top-left: The orange bars show the transaction volume in transactions per second for each transaction.
2. Top-right: here we have the performance-DNA of the two transactions. These two are inputs for the model.
3. Below: the vertical green bars represent the percentage CPU utilization. The two bars correspond with the CPU's and disks respectively. This is one of the outcomes of the model.

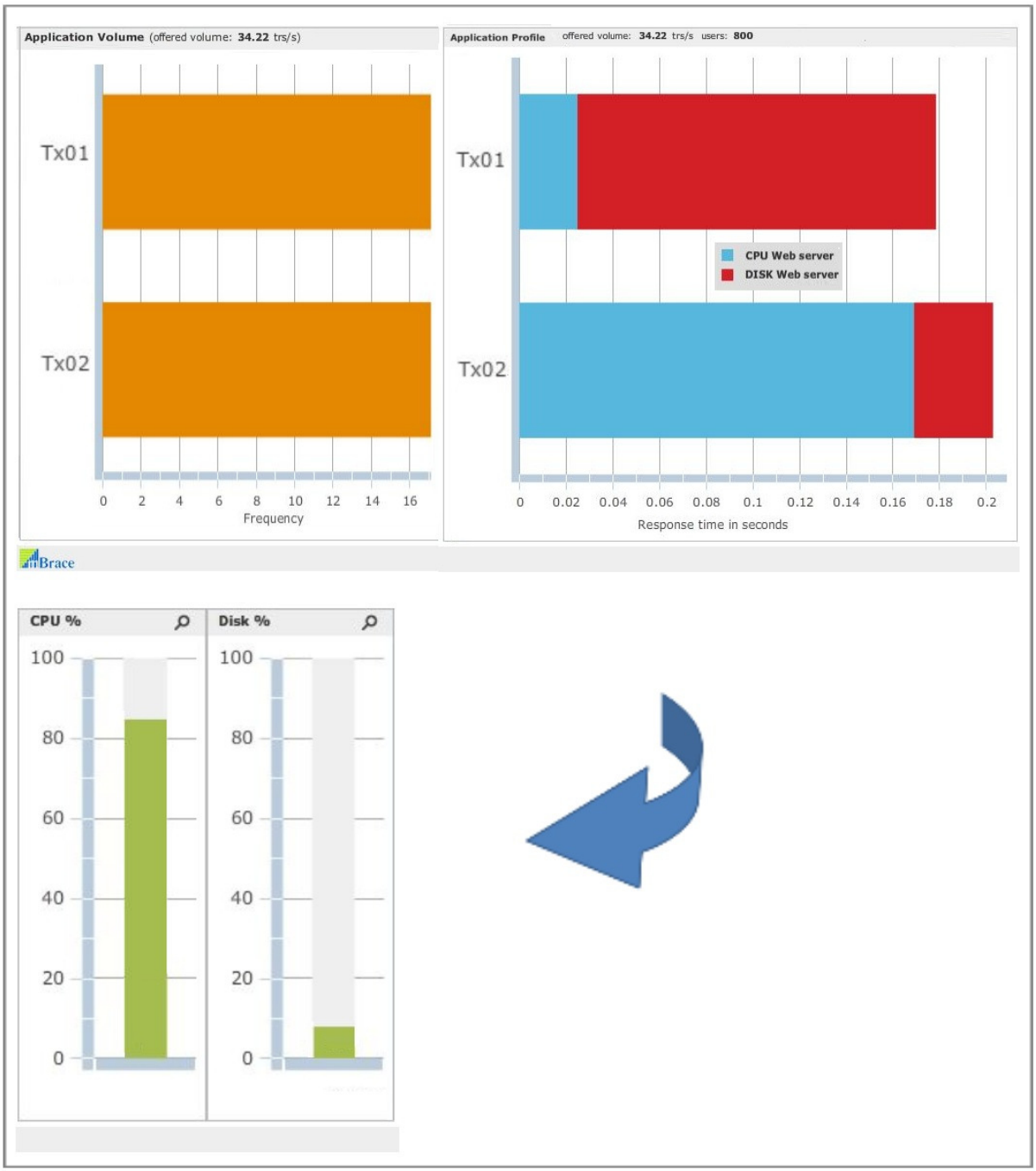
In this example we have two transactions that make use of one server. No service demands are included in the DNA other than for CPU and disk on only one server. It is not that there is no client-PC, other servers or networks, they are just left out of sight.

The upper transaction uses much CPU, and little disk IO. The lower transaction consumes only little CPU, but produces many IO's to the disks.

The first figure shows the load on the two resources, CPU and disk for a usage volume of 200 users producing a total of 8.56 transactions per second. The second one shows the load from 800 users producing a total of 34.22 transactions per second. In this case both transactions have exactly the same transaction rate. In that respect this is an exceptional case. Commonly transactions have different transaction rates.

The lower sections of the figures show the load on the resources in terms of a utilization percentage. This utilization percentage (called %utilization in the charts) of a resource, between 0% and 100% by definition, indicates how much capacity the application requires. The greater the transaction volume the higher the %utilization (this is fairly obvious isn't it?).

It is no surprise that the percentage utilization on the resources is higher in the second figure (below) and it is also hardly a surprise that the percentage utilization on the resources of the second figure is four times as high.



Usage volume specification

There are two ways to specify a usage volume:

- Number of transactions per second by transaction type
- Number of concurrent users together with think times (the time between the successive transactions of one user) by transaction type

They both, either or, can do the job and these two quantities are interrelated. Next two tables show how the usage volumes are fed into the model.

It's either Table A or Table B.

Table A
Transactions executed per second by type

Id	Transaction name	Transactions per second	Think time
1	Visit Homepage	400	10
2	Sign in	400	2
3	RequestOverviewAccounts	400	2
4	Visit PageMoneyTransfer	400	2
5	EnterMoneyTransferOrder	1,600	30
6	VisitPageSubmitMTOrders	400	2
7	SubmitMTOrder	800	6
8	EnterToken	800	40
9	ProduceDailyReport	400	2
10	Sign off	400	6
	Totals	6,000	
	Averages		15.9

Table B
Number of concurrent users by transaction type

Id	Transaction type name	#Concurrent users	Think time
1	Visit Homepage	4,200	10
2	Sign in	1,040	2
3	RequestOverviewAccounts	1,840	2
4	Visit PageMoneyTransfer	1,000	2
5	EnterMoneyTransferOrder	52,320	30
6	VisitPageSubmitMTOrders	960	2
7	SubmitMoneyTransferOrder	6,720	6
8	EnterToken	34,000	40
9	ProduceDailyReport	2,040	2
10	Sign off	2,720	6
	Totals	106,840	
	Averages		15.9

The tables could be the result of analyzing a transaction log showing the counts and measurements of transaction types that have been executed. The tables show transactions per second, think times and concurrent users. It shows that 106,840 users were concurrently using the application. Most of them, 52,320 were busy using transaction type 5, EnterMoneyTransferOrder. Transactions were processed at a total rate of 6,000 per second. The blend of different transaction types is clearly defined by the column *Transactions per second*.

The model can use this input (either Table A or Table B) together with the performance-DNA to calculate the resource utilizations and response times.

Interrelationship between transactions per second and concurrent users: Little's Law

How are these two volume specifications in Table A and Table B interrelated?
If you know the response times (and the performance-DNA provides us a lower limit of the response time for a start) and the think times you can use Little's formula to derive the transactions per second:

$N = X * (R+Z)$

- N – number of concurrent users
- X – throughput in transactions per second
- R – response time

Z – think time

Example:

X (throughput) = 10 transactions per second
R (response time) = 2 seconds
Z (think time) = 50 seconds

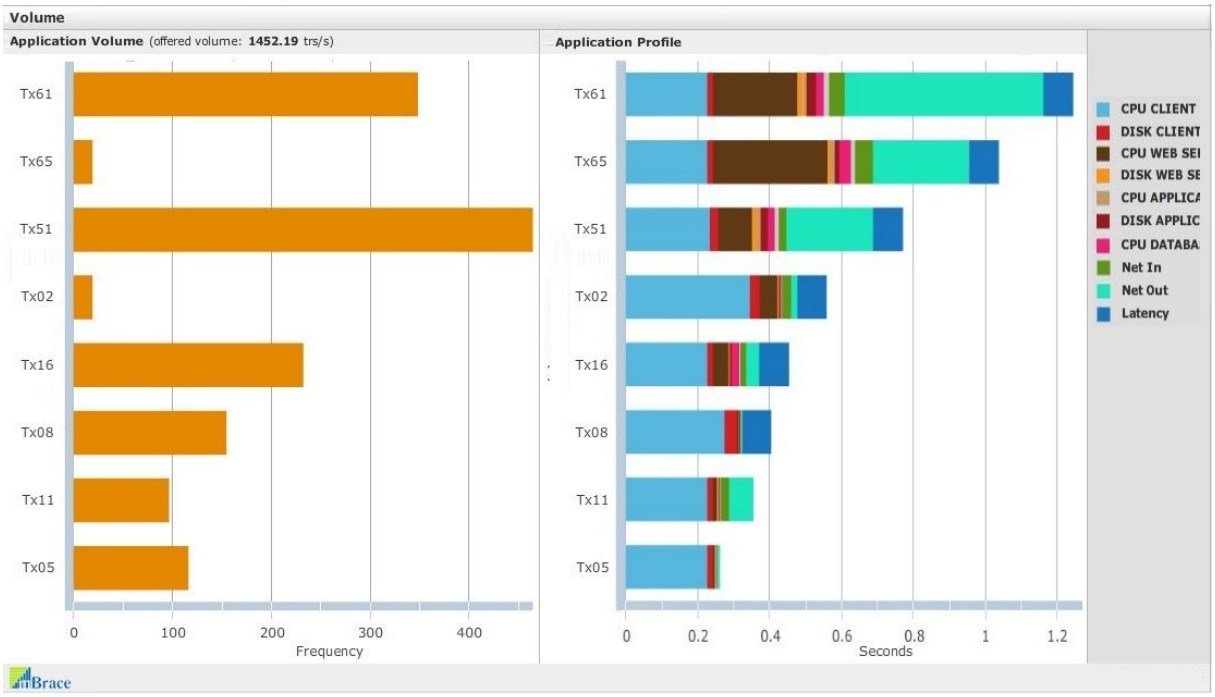
As a result we know that there are N = 520 concurrent users in this simple example. Obviously this also works the other way around.

Not only is this handy for preparing the input for your performance model, it is also handy for preventing confusion. Many times there is a Tower of Babel confusion about the usage volume of the application. Your colleagues may talk about so many transactions per second, while you try to handle the number of users in your model. Little's Law bridges this gap for you quite easily. And by the way, this innocently looking formula amazingly is one of THE corner stones of modeling technology.

As I promised this is the only math in this series of articles.

Usage volume and the load on resources, continued

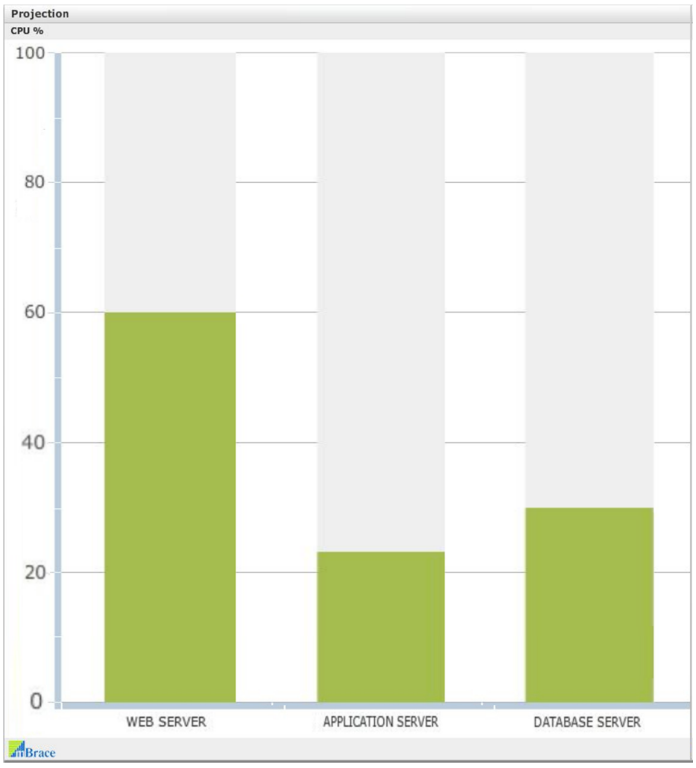
Next pictures give an example showing a transaction blend at a total rate of 1,452 transactions per second and the consequent utilization percentage on a range of hardware and software resources.



The figure shows transaction volume, transaction blend and performance-DNA of an application with 8 transactions. The left section of the figure shows the usage volume. The horizontal orange bars show a transaction frequency for each one of the transaction types adding up to 1,452.19 transactions per second. The right section shows the performance-DNA of the application. The legend at the right reveals that the application deals with an infrastructure chain consisting of Client

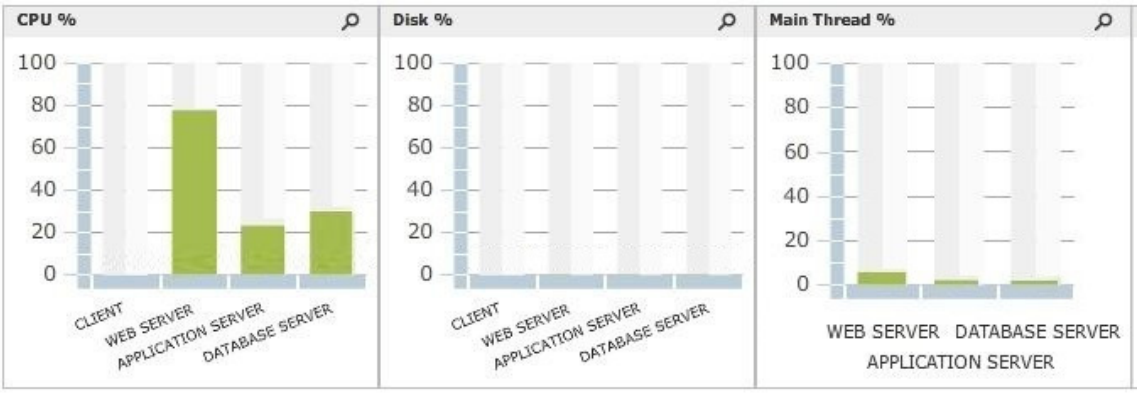
PC's, a Web server, Application server and database server. They are interconnected by 4 network connections, A, B, C and D.

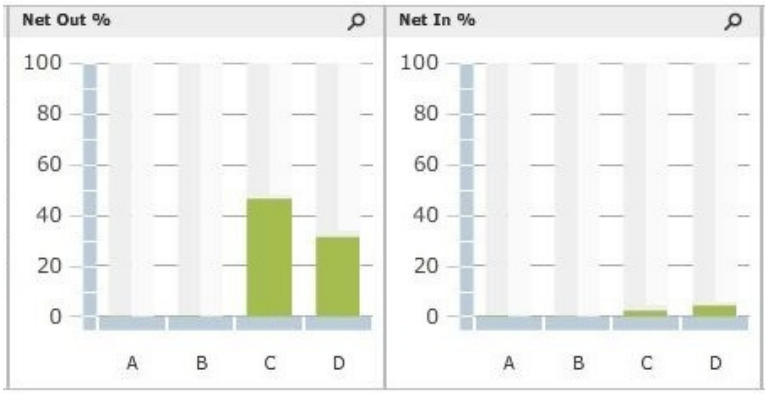
The next figure shows the load that the application with this usage volume causes on the CPU's of each server.



CPU's are not the only resources used by the application. The next figure (split in two parts) gives a more complete example of the load on CPU's, Disks, Outbound network resources, Inbound network resources and Main processing threads of each server in the chain.

Each resource has a utilization percentage determined by usage volume and performance-DNA.





The above figure (in two parts) shows the load on the CPU's and Disks of 3 servers, 4 Networks in- and outbound, and the processing threads related to the usage volume. The processing threads of the servers are software resources. Software resources are also charged by the application and consequently have a utilization percentage. The load on the disks and networks A and B are so low that no green bars are visible.

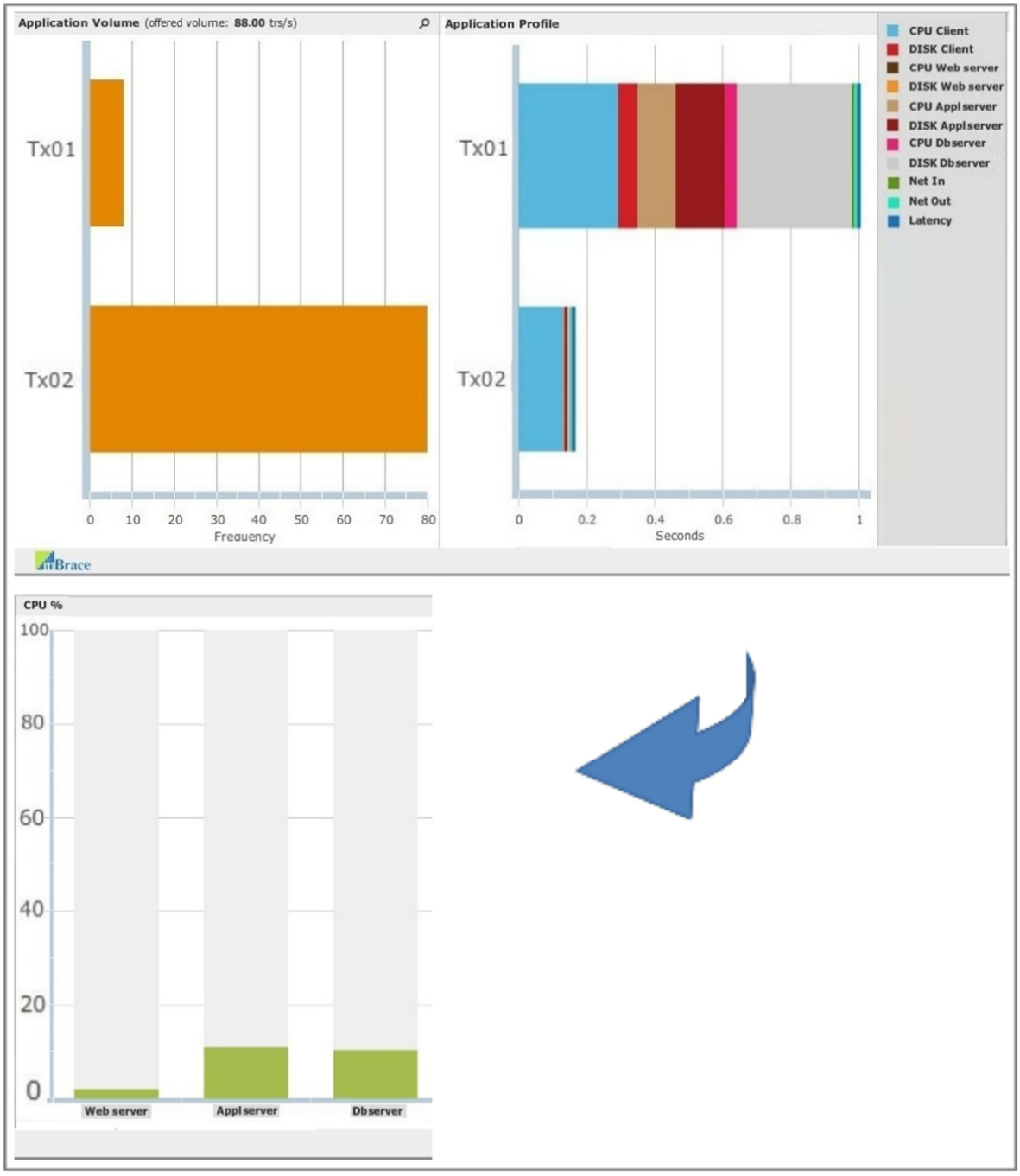
Transaction blend

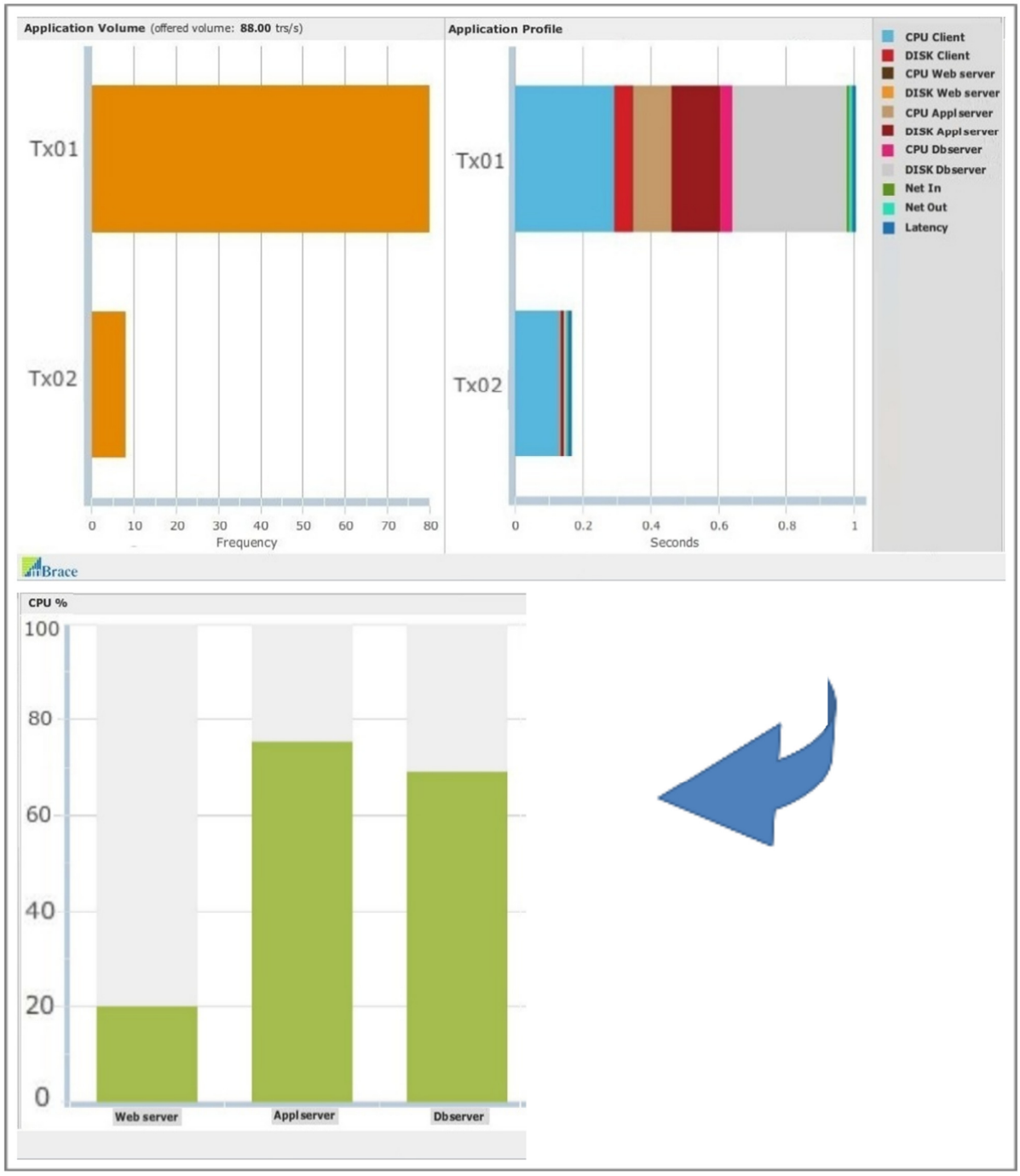
It is important to stress that not only the transaction volume (e.g. 10 transactions per second) determines the load on the resources, but the transaction blend also plays its part. The next two figures stress this. They compare the load of an application with two transaction types named Tx01 and Tx02 on the CPU's of three servers. The total transaction rates and the performance-DNA's are the same. The only difference is the transaction blend.

Each of the two figures consist of three sections, from left to right:

1. Top-left: The orange bars show the transactions per second for each transaction.
2. Top- right: here we have the performance-DNA of the two transactions. The upper one has a heavy transaction profile, the lower one has more modest service demands.
3. Bottom: the vertical green bars represent the percentage CPU utilization. Each bar corresponds to one server and shows the utilization percentage on the CPU's of that server.

The first two items are input for the model, the third is part of the outcomes of the model.


Application Volume (offered volume: 88.00 trs/s)
Tx01
Tx02
0 10 20 30 40 50 60 70 80
Frequency
Application Profile
Tx01
Tx02
0 0.2 0.4 0.6 0.8 1
Seconds
CPU Client
DISK Client
CPU Web server
DISK Web server
CPU Appl server
DISK Appl server
CPU Db server
DISK Db server
Net In
Net Out
Latency
Brace
CPU %
100
80
60
40
20
0
Web server
Appl server
Db server


Application Volume (offered volume: 88.00 trs/s)
Tx01
Tx02
0 10 20 30 40 50 60 70 80
Frequency
Application Profile
Tx01
Tx02
0 0.2 0.4 0.6 0.8 1
Seconds
CPU Client
DISK Client
CPU Web server
DISK Web server
CPU Appl server
DISK Appl server
CPU Db server
DISK Db server
Net In
Net Out
Latency
InBrace
CPU %
100
80
60
40
20
0
Web server
Appl server
Db server

In both cases the transaction volume is 88 transactions per second. The first case has a low volume on Tx01, the transaction with the heavy profile and a high volume on Tx02, the transaction with the light profile. In the second case it is the other way around. As a result we can see a considerable difference in load on the CPU's in the bottom sections of the figures. In the upper figure the CPU utilizations hardly reach 10% (even as low as 2% on the CPU's of Web server), while the second figure shows utilizations up to 75% for the CPU's of Appl server.

This example illustrates two things:

1. Usage volume and performance-DNA together determine the load on the resources. In this example only the CPU's are involved, but this goes the same for the other resources.
2. Not only the total transaction volume but also the transaction blend determines the load.

Notice that so far the time behavior or response times of the transactions at these transaction volumes, another outcome of the model, hasn't been shown yet.

This article revealed the ins and outs of the usage volume of an application and its impact on the load on the resources. A thorough description of usage volume is not only important for performance model based approaches, but also for conventional load testing approaches. You need to know the usage volume if you want to control application performance.

Performance-DNA and usage volumes are two out of three main aspects that determine the performance of an application. They also are two out of the three main inputs for a performance model. The subject of the next article will be hardware, configurations and capacities, which is the third main input for an application performance model.

Any comments or debate: michael.kok@mbrace.it.