Securing System Performance
with
Transaction Aware Performance Modelling
A Case Study

Version 0.4
4 June 2009
Author: Michael Kok

Content

1 Introduction and Summary

Transaction Aware Performance Modelling (TAPM) allows us to analyse the performance and capacity needs of each transaction type of an information system individually. This requires special techniques for modelling and measuring. As a result extra power is added to model driven performance analysis providing capability to support System Performance Engineering (SPE). In this paper the power of TAPM is demonstrated by referring to an example of an analysis that was conducted on a newly-created business application deployment.

The analysis process conducted, included:

- The identification of the individual transactions, and the workload that made up the business workflow of the developed application, followed by the collection of metrics for each transaction.
- The analysis of the relationship between transaction response times and hardware capacity usage at various levels, including server, transaction and total workload.
- Proposed improvements for the efficient usage of hardware capacity for the application were tested and their effects verified with support from the model.
- Finally, the impact of two software servers on the performance of the application as a whole was analysed, and the capacity of those servers were appropriately optimised.

As a result of the performance analysis the following benefits were realized:

- An Application Performance Profile was established prior to application deployment.
- Application performance problems could be identified and resolved early in the development cycle.
- The efficient use of the hardware by the application could be improved considerably.
- Application production hardware environment could be sized efficiently.
- Viable performance enhancement and stability solutions.

Jointly, considering the strategic and tactical advantages made available to the application project management and development teams early in the development life cycle, the benefits mentioned above represent a significant time and materials cost saving.

2 The System Analysed

The system under analysis was a newly developed n-tiered workflow management application, entailing a Web Server, Application Server and Mainframe Server.

Application workflow was provided by an external application (Application STP) which processes business cases in an automatic (Straight Through Processing) manner. A percentage of these business cases would require “manual” processing and were transferred to the Application under Analysis for further processing. The business purpose of the Application Under Analysis, was to accept these cases and allow application users to interact with them in a controlled way.

Application users would interact with the application using the following main transactions:

- Login – User logs on.
- Case Selection – User selects cases, with varying number and sort type.
- Case Assignment – User assigns cases to the relevant business units.
- Case processing, which entails:
  - Start Order – User starts order.
  - Preserve Order – User saves order.
  - Complete Order – Closing procedure.
- Collection of items under workflow regime from the STP Application.
- Logoff.

The following list of transaction types were selected for performance analysis:

1	1 Login
2	2A Select 50 A Nosort
3	2B Select 50 B Sort
4	2C Select 50 A Sort
5	2D Select 50 B Nosort
6	2E Select 100 A Nosort
7	2F Select 100 B Sort
8	2G Select 100 A Sort
9	2H Select 100 B Nosort
10	3 Show lists
11	4 Assign order to employee
12	5A Assign order to unit A
13	5B Assign order to unit B
14	6A Start order A
15	6B Start order B
16	6C Start order C
17	7 Preserve order
18	8A Complete order A
19	8B Complete order B
20	9 Collect item
21	10 Logoff

List. Transaction List

Transaction 2 is of particular interest and has 8 sub-types (2A to 2H). Each sub-type is from the same transaction type but only differs in its preconditions. These preconditions are determined by three parameters:

- Option A or B
- A selection size, in this example either 50 or 100 work items.
- The option to sort or not.

The heavy use and expected impact of Transaction 2, required a number of sub-types to be analysed. This would allow for an in-depth study of the impact of the various preconditions.

Transactions 5, 6 and 8 show variants that are truly different transaction types accommodating different other applications depending on the workflow application.

3 The Model

3.1 Overview of the Model

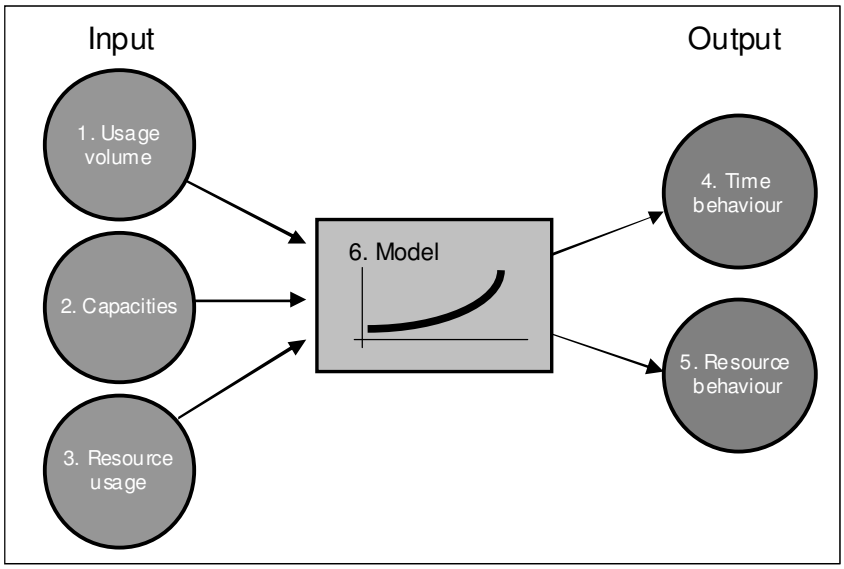


Fig. Model Overview

The analysis was conducted using the mBrace model schematically depicted above. The model requires three inputs and provides two outputs as depicted above. Usage volumes (transaction workloads) and Capacities (hardware capacity available) metrics were collected from the organization. Resource usage metrics are obtained through measurement.

The inputs provided are inserted into the model which calculates the outputs and shows the time behaviour and resource behaviour results in a graphical format. The impacts due to changes in transaction volumes and available server capacity are studied through comparative analysis of the outputs.

The displayed outputs are shown in greater detail in the next sections.

Fig. Use Case Configuration

The above snapshot from the model GUI shows the use case with the transaction types that were selected for further analysis.

A use case is a clustering of transaction types. When defining a usage volume for the system we may compose it of one or more use cases. Commonly a use case is recognisable for the business. E.g. we may have a use case “Sell a policy”. The business knows how many policies it sells (or intends to sell). So we can easily determine how many times the use case is executed. At the bottom right we may fill in the execution rate of the entire use case in numbers per second.

For simplicity sake all transactions wer grouped in one use case.

Each transaction type may be executed once each time the whole use case is executed, but it may also be executed multiple times (multiple being smaller, equal or greater than 1). It requires accurate process analysis to determine how frequent transactions are executed within a use case. For each transaction type we may fill in the Freq column at the right. Ultimately the transaction volume for each transaction type is determined by multiplying the values under Freq with the Use case Frequency. All this taken together yields the overall transaction volume in number of transactions per second. Moreover this also yields the transaction mix. Not only the transaction volume, but also the transaction mix is of significant influence to the performance and capacity needs of the system.

As can be seen not all transactions measured were included in the use case. The data filled in display the result of careful analysis of the expected usage. However this is a forecast with a speculative nature to some extent.

3.3 Configuration

The graphic below provides a schematic overview of the application infrastructure. Web and AS were UNIX servers, Mainframe was a z/OS server.

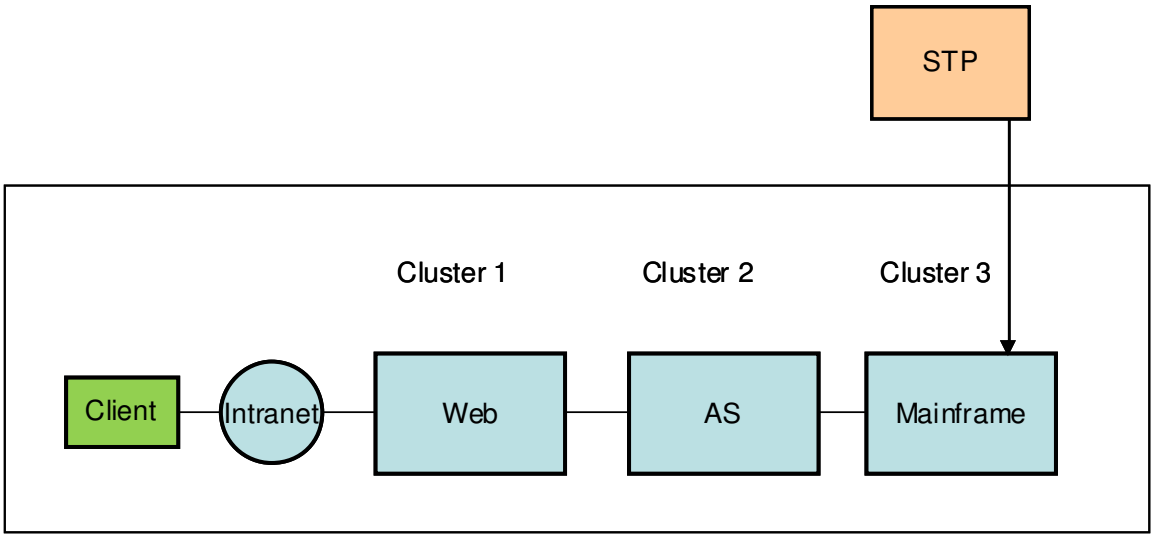


Fig. Application Infrastructure

The test environment provided was not representative and varied extensively from the planned application production environment, but this did not prevent effective analysis. These variations included:

- Server clusters were composed of single server nodes only.
- The AS application server provided in the test environment was an old machine and would be replaced by a server with CPU's four times as fast in Production.
- The test environment Mainframe CPU's were slightly faster than those planned for the production environment.

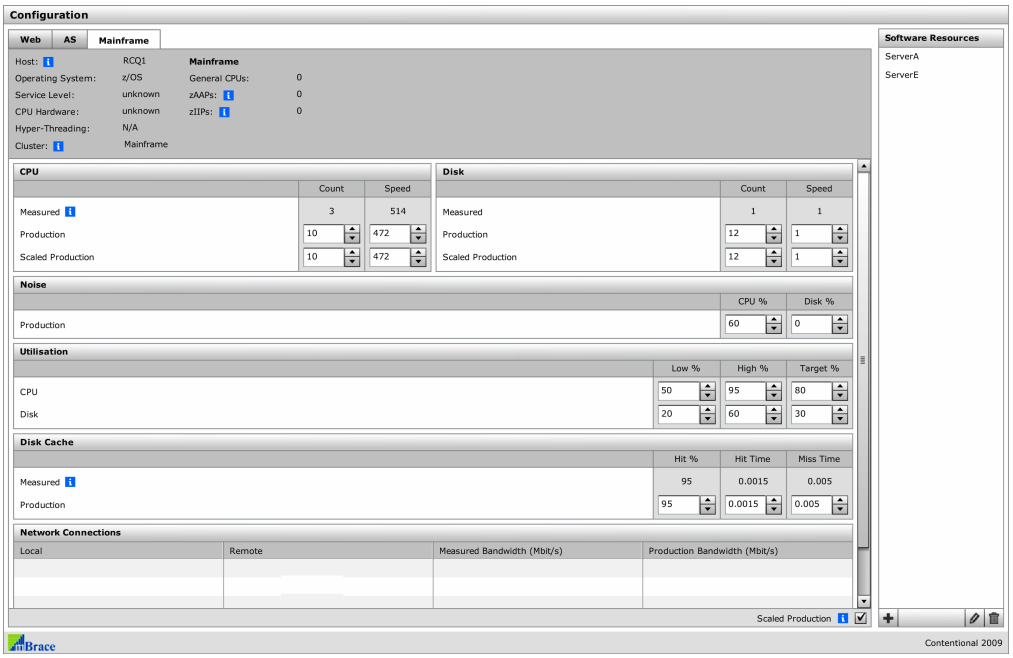


Fig. Server Resource Configuration – Mainframe Server

The Configuration window shown has a tab for each server in the infrastructure chain.
In the above screenshot the tab for the mainframe server of this window was opened to show the above picture. The window is composed of several sections:

The upper grey part shows the characteristics that are known from the test environment.

Next below we have the sections for CPU and Disk. This part displays the capacities of the Test environment, the Production Baseline and the Scaled Production environment. Here we can enter the numbers of CPU's and disks as well as their speeds. We can scale horizontally in the model by changing the number of the devices and vertically by changing their speeds.

Next below is the section Noise. Here we can enter the load imposed on the devices by other applications.

The section Utilisation holds parameters that can be used to determine how to scale up or down. If the outcome of the model shows a value of %utilisation above the High % we have to scale up to approach the target value for %utilisation of the device as close as possible.

The section Disk cache shows the %hitrate on the cache and the times to fulfil a hit or miss in seconds.

The section Network Connections holds the data for all network connections of the server.

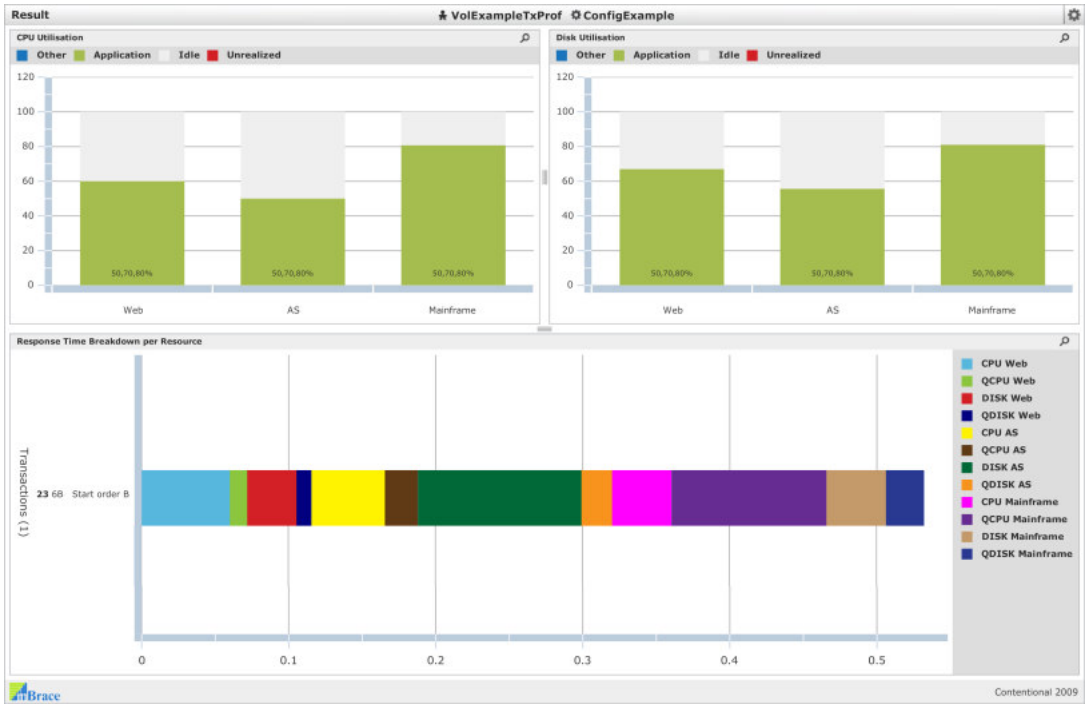
The section Software resources at the right shows the software resources that are modelled. To show the parameters of these resources another window must be opened. This will be shown later in this document.

Each time a parameter has changed the model recalculates the results and produces new outputs.

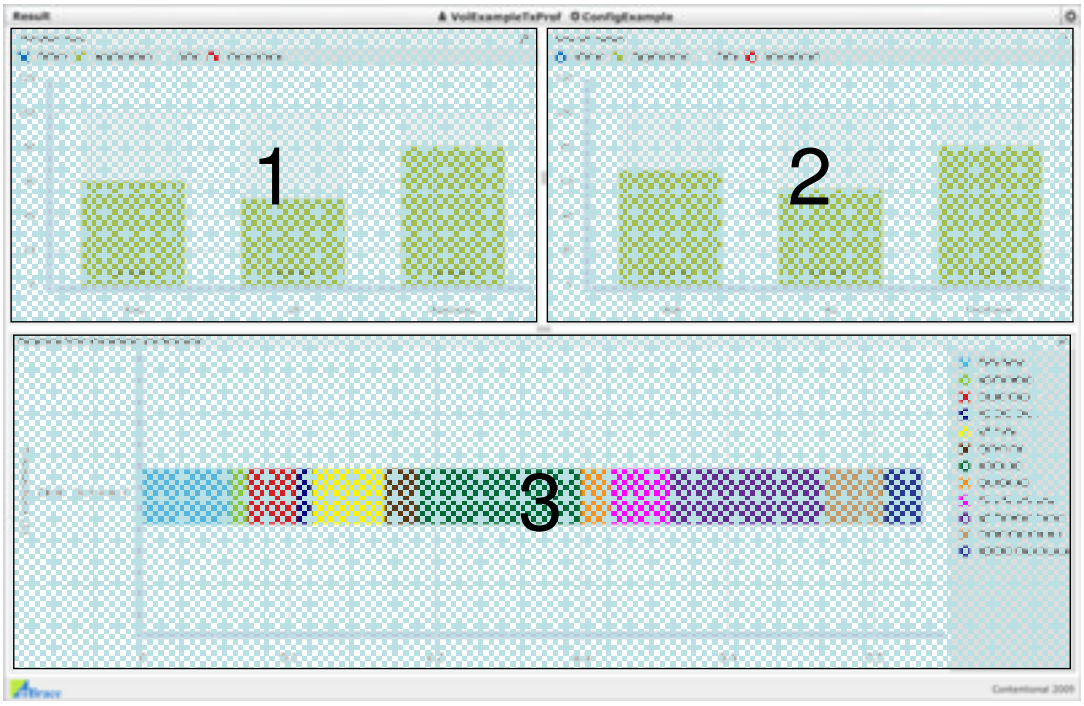
Following the completion of the model input parameters, the model outputs were calculated and displayed.

4 Explanation of the output of the model

Both outputs of the model – time behaviour and resource behaviour – are displayed in one chart of which a simple example is shown below.



What does this chart tell us? This chart tells all there is to know about the capacity need and performance of a system at one glance. Once someone is familiar with the graph reading an mBrace performance report becomes as easy as reading a book with cartoons.



The chart has three main sections:

Section 1 shows the %utilisations of the CPU's of the servers. For each server there is one vertical bar showing the utilisation as a percentage of the total capacity of the resource of that server.
Section 2 shows the %utilisations of the access paths of the disks of each of the servers.
Section 3 shows the time behaviour of one transaction type. The total length of the coloured bar indicates the end to end response time on this transaction type. The length of each coloured component of the bar shows the time spent by either the activity on or waiting time for a resource.

Text balloons are added to the chart in the next picture. Note that colours of the bar are also explained by the legend at the right.

To simplify the chart, in this example only one bar is shown for one transaction type. Commonly a larger number of transactions are displayed in the chart. Sometimes the number of transaction types involved in the analysis is greater than can be displayed in one chart. Then the transaction types can be scrolled up and down. Also in many cases the transaction types are sorted in decreasing order of response times so one can see the most interesting transaction types in on one page.

5 Outcomes of the Performance analysis

5.1 Resource Usage

Transactions were executed in the test environment a number of times and their resource usage metrics were collected. These metrics were then parsed and "scrubbed" to remove outliers and secure the quality of the model outputs.

Next, transaction volume and test and production environment server capacities for each server were inserted into the model.

The mBrace model outputs reported the resource usage and response time behaviour of the application at the transaction level according to the server configurations provided in the previous step.

The response time of each transaction type was broken down to display the amount of time contributed by each separate server resource component.

The two graphs below show the outputs displayed for the **test** and the **production environments**.

The results shown below, demonstrated the transaction response time breakdown for the transactions as they were captured in the **test environment.** Response times indicated are for single user executions of the transactions only. Multiple user impact is considered at a later point in this paper.

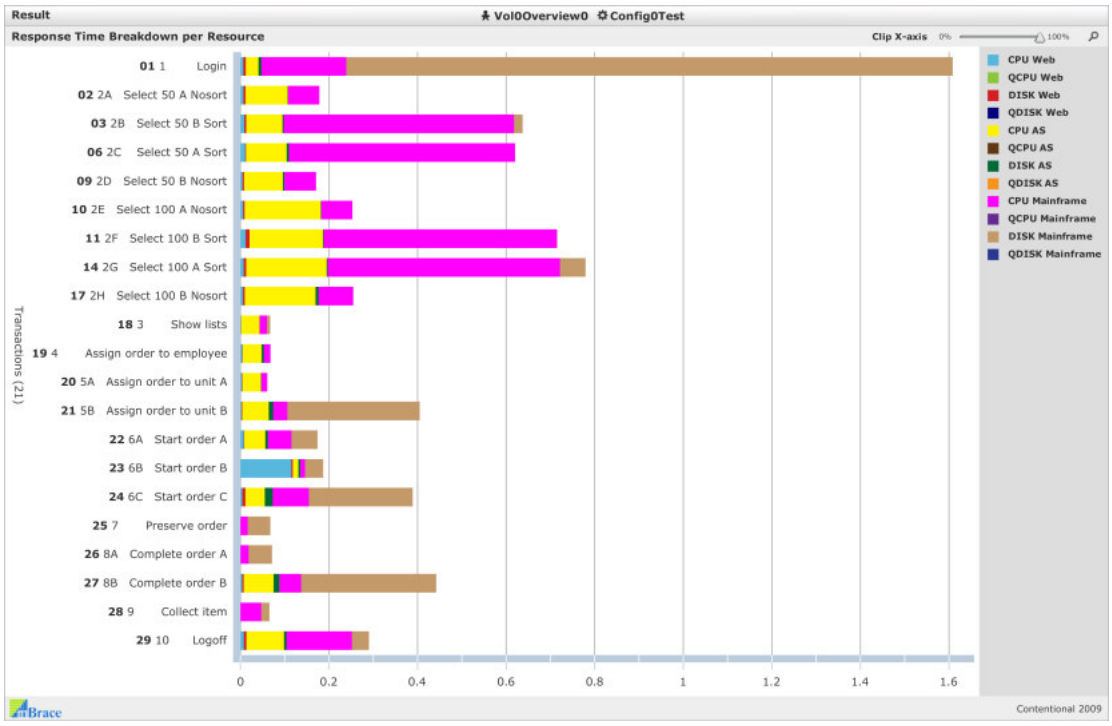


Fig. Transaction Response Time Breakdown - Test Environment

Observations:

- Login transaction - had the highest response time at 1.6 seconds. The main response time contributions were:
    - Mainframe Disk - approx. 1.35 seconds

    - Mainframe CPU – 0.2 seconds.
- Selection transactions - had the following characteristics:
    - Selection response times were mainly due to contributions by AS CPU and Mainframe CPU resources.
    - Selection Sort types:
        - All Sort selections increased the transaction response times by 0.5 seconds, irrespective of the number selected. Main contributor - Mainframe CPU. So these transaction types use five times as much CPU as their Nosort counterparts!
    - The number of cases (50 or 100) had the following impacts:
        - No visible impact on the Mainframe CPU seen.
        - AS CPU contribution increased proportionately to the number of cases. (50 cases – 0.1 seconds, 100 cases – 0.2 seconds).
- Assign, Start and Complete transactions - had response time contributions by the Mainframe server (CPU and Disk, of about 75%.

The next graphic, depicts the transaction response times if they had been executed on the proposed **production environment**. Response times indicated are for single user executions of the transactions only.

Note that a major difference between the test and production environments, was the 4x increase in the speed of the AS CPU's.

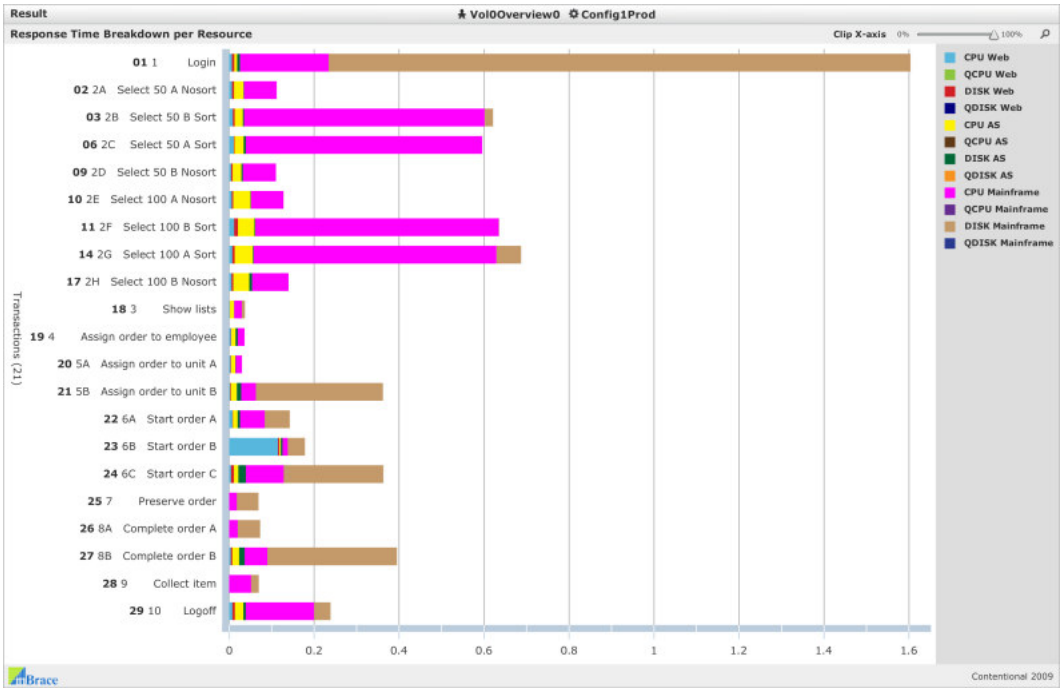


Fig. Transaction Response Time Breakdown - Production Environment

Observations:

- As expected, the main reductions in response time were due to the 4x increase in the AS CPU speed – shown in yellow.

The output relative to production systems, also allows for comparative analysis of varying production environment configurations and varying user loads.

Now that we've set up the initial model, we can explore the predicted production results in more detail with the goal of optimizing the performance and capacity of the production environment. We'll explore that in next month's continuation of this article.

6 Performance - Hardware Resources

Following the individual transaction analysis, we proceeded to investigate the application with a multiple user load.

6.1 Time and Resource Behaviour - Target Volume Baseline

The following graph depicts the application server resource utilisation under load (16 transactions per second) and serves as a baseline measurement in order to allow for later comparative analysis.

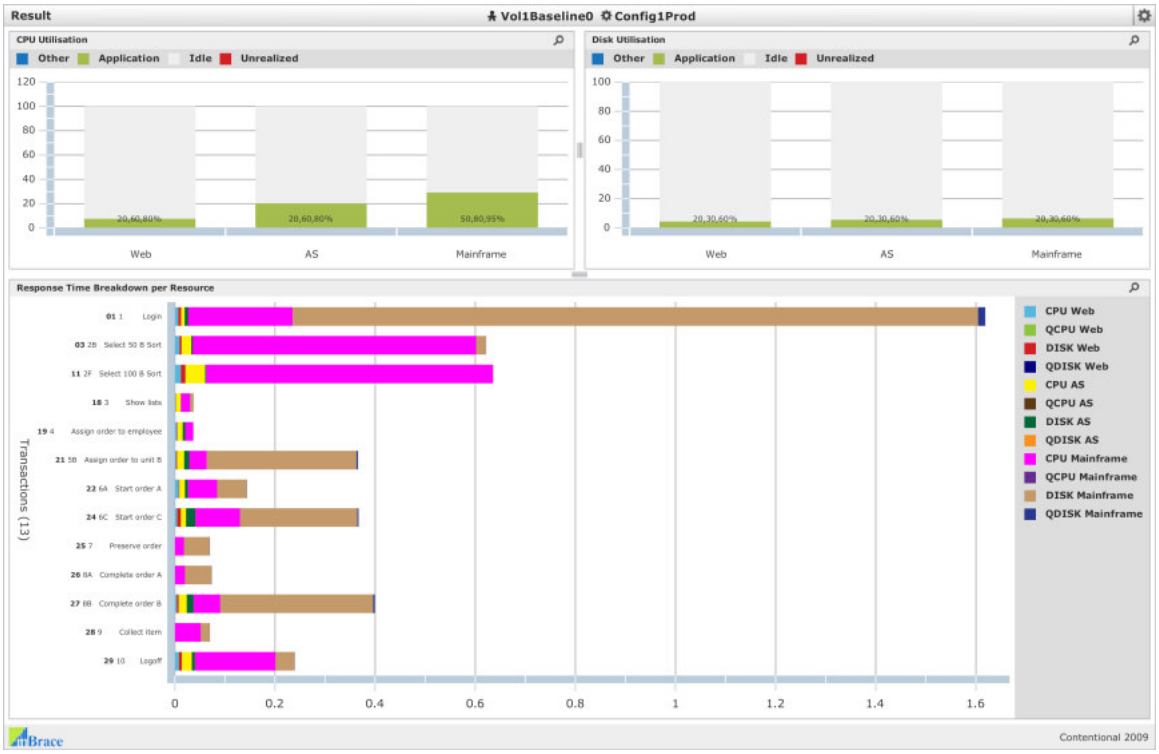


Fig. Transaction Response Time Breakdown - Production Load, Production Environment

- The above graph is explained in more detail in section Explanation of the output of the model.

Observations:

- **CPU Utilisation**
  - CPU utilizations on the servers seem low: 8%, 20% and 29% respectively for Web, AS and Mainframe servers.
  - Note, that for a Mainframe server with 10 CPUs, 29% CPU utilization is a considerable capacity demand.

- **Disk Utilisation**
  - High response time contributions due to the Mainframe disk resource can be seen on the Login, Order Assignment, Order Start and Order Complete transactions, denoting extensive IO to disk in those transactions.
  - However, those transactions have a lower rate of occurrence than the Selection transactions making them a secondary priority for improvement.

- **Response Time Breakdown**
  - Response times of all transaction types were within acceptable limits (1 second), with the exception of the 01 Login transaction (1.6 seconds).
  - The transaction 2 variants with sorting functionality (2B and 2F) show the high response time contribution due to the Mainframe CPU resource.

Transaction Type 2 was a primary candidate for optimization – It had a high consumption of Mainframe CPU resources relative to other transactions and a high rate of transaction occurrence.

6.2 Time and Resource Behaviour - Target Volume with External Application Impact

The next perspective taken into consideration, was the impact on the Application under Analysis due to external influences such as other production environment applications. Whilst the Web and AS servers were dedicated to the application, the Mainframe server was shared with other production environment applications.

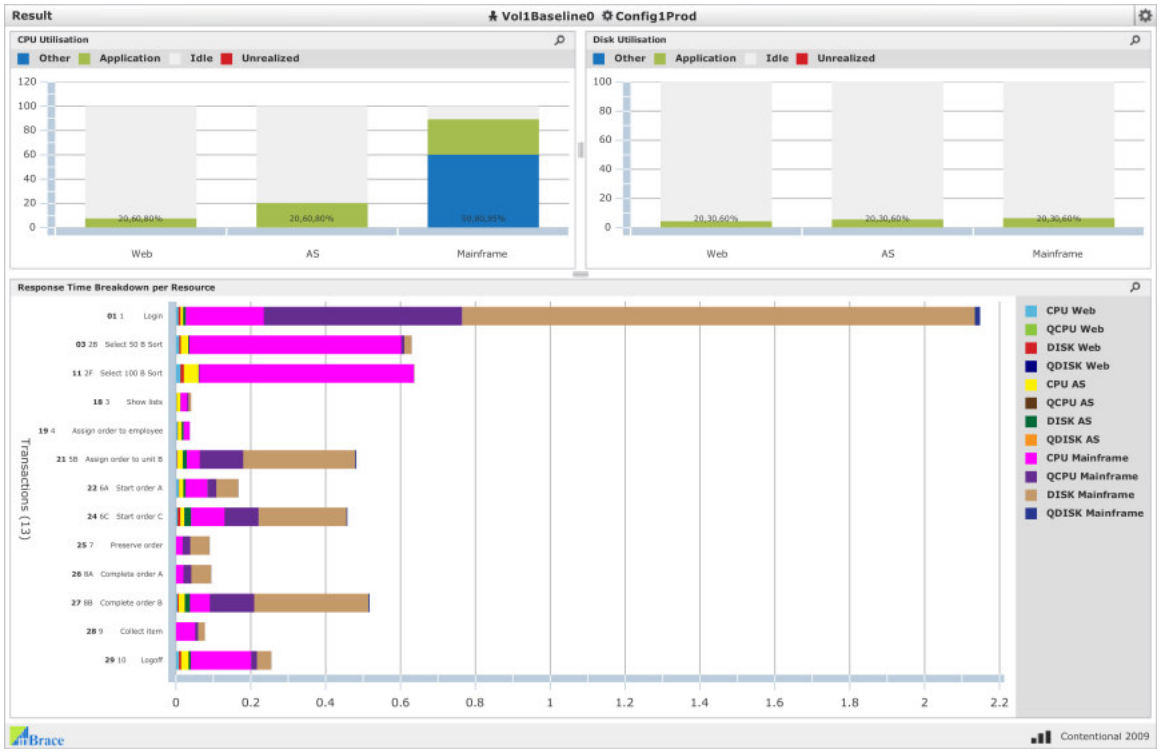


Fig. Production Environment – Production Load Resource Utilisation with External Apllication Impact

The above chart demonstrated the impact that the production environment Mainframe server applications would have on the performance of the transactions under analysis.

These contributions are clearly shown in blue in the picture above.

Observations:

- **CPU Utilisation**
  - External application Mainframe server utilization was high at 60%, resulting in a total of around 90%. This was a major concern.

- **Disk Utilisation**
  - Negligible impact on Mainframe disk subsystem.

- **Response Time Breakdown**
  - Login transaction response time increased from 1.6 seconds to 2.2 seconds.
  - Transaction 2 variants (with and without sorting functionality) do not appear to be impacted by the external applications.

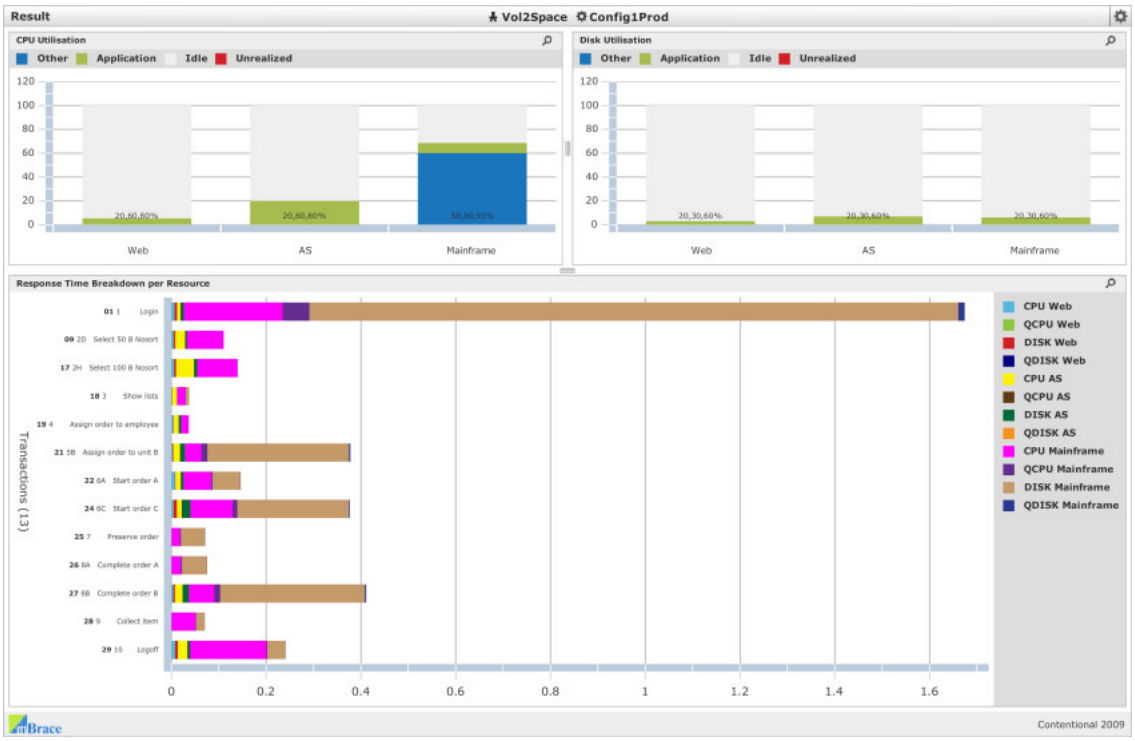


Fig. Production Environment – What If Examination.

In the picture displayed above, transactions 2B and 2F (sort functionality) were replaced by 2D and 2H (nosort functionality).

Observations:

- **CPU Utilisation**
  - Mainframe server utilization drops from 29% to 8%.

As a consequence the application developers were advised to re-engineer the Case Selection transaction (Transaction 2), with specific focus on the optimization of the Mainframe impact due to the sort functionality.

6.4 Transaction 2 Optimised – Response Time

Following advisement, the Case Selection transaction (Transaction 2) was optimised by the developers in two stages. The final optimization produced better results than the preliminary optimization. After each optimization delivery, the transactions were executed on the test environment. Measurements were taken and their metrics imported into the model.

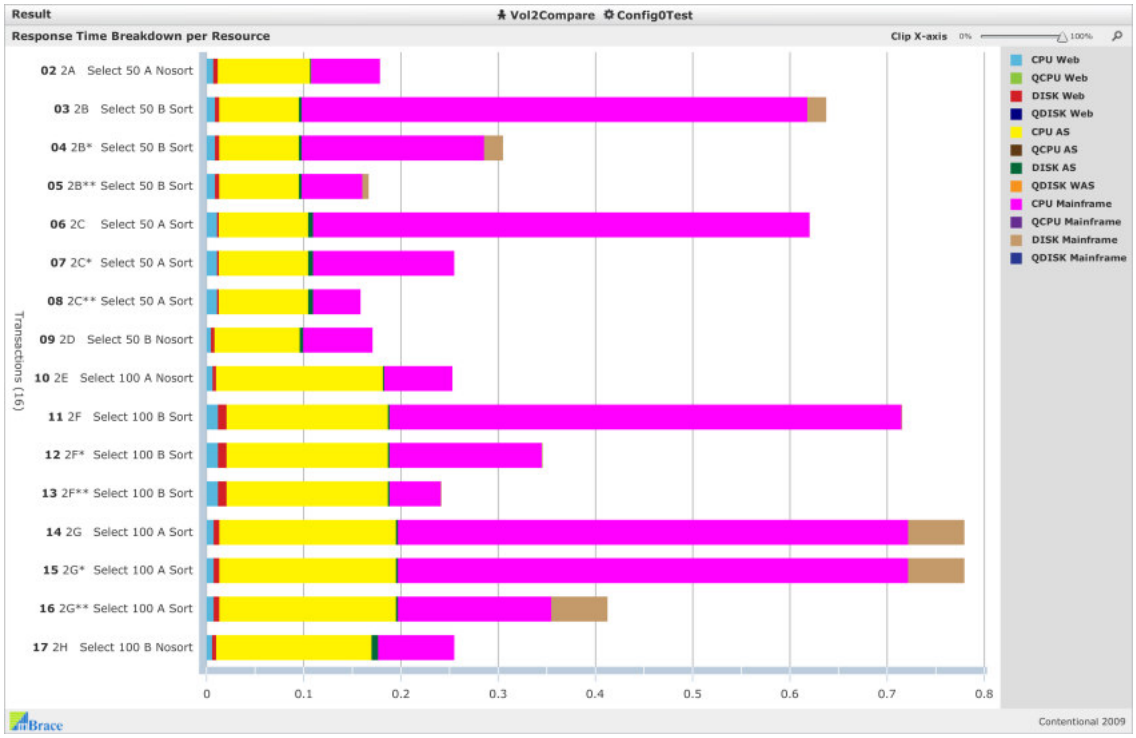


Fig. Response Time Breakdown – Improved Transaction 2.

The figure above displays all the variations for Transaction 2 only. The result of improving transaction type 2 can be seen in the figure above. The transactions above marked (*) denote the preliminary improvement results, whilst those marked (**) denote the final improvement results.

Observations:

- **Response Time Breakdown**
  - Transaction 2B (Select 50 B Sort) and Transaction 2C (Select 50 A Sort)
    - response times decreased from 0.6 seconds to less than 0.2 seconds
    - Transactions were as Mainframe CPU efficient as their related no-sort transactions.
  - Transaction 2G (Select 100 B Sort) experienced similar improvements in efficiency.
  - Transaction 2H (Select 100 A Sort) experienced moderate improvements in efficiency after the second optimisation.
- **Purpose of the improvements**
  - Decreasing the response times of 2C and 2G was not the point. No user will notice an improvement from 0.4 to 0.2 seconds.
  - This improves efficient use of Mainframe CPU capacity. This will be shown in the next sections.

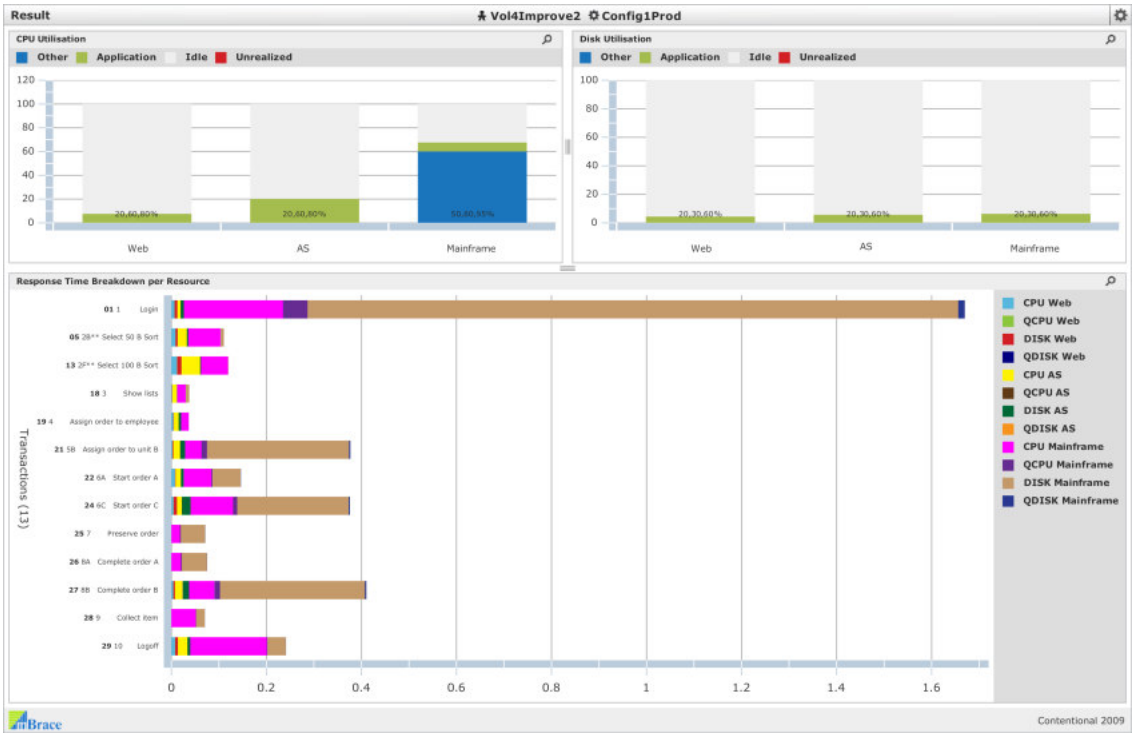


Fig. Server Resource Utilisation – Improved Transaction 2.

The above chart shows the impact of the "sort" Transaction 2 improvements on the applications server resource utilization.

Observations:

- **CPU Utilisation**
  - The CPU consumption on the mainframe has dropped considerably from 29% to 8% on 10 CPU's.
- **Disk Utilisation**
  - Negligible impact on Mainframe disk subsystem due to improvements.
- **Response Time Breakdown**
  - The Case Selection sort transactions decreased from 0.6 seconds to 0.2 seconds.

Considering the high application operational costs due to the increased Mainframe CPU capacity requirement prior to optimization, considerable savings were made through the Transaction 2 bottleneck identification and optimization efforts.

7 Performance - Software Resources

At this point we had analysed the performance profile of the application at various levels:

- Individual transactions were analysed for potential impacts at load.
- Server resource deficiencies were identified in individual transactions.
- Transactions were prioritized on the basis of which transaction optimization efforts would be most efficient.
- Transaction optimizations were verified for efficiency after delivery.
- The impact of co-hosted applications was also considered.

Whilst the application interacts with a number of hosts two software servers were identified on the Mainframe that would require special attention and were designated ServerA and ServerE. Both ServerA and ServerE are single-threaded non-reentrant software servers that handle the processing of the transaction types in the workflow management middleware. These servers are being hosted on the Mainframe.

- ServerA was used when starting and stopping the application and was not scalable.
- ServerE was used for all other transaction types and could be horizontally scaled by deploying more instances. In the analysis baseline, two instances of ServerE were deployed.

The next sections show the way the behaviour of ServerA and ServerE was analysed and how their capacity was optimised.

This is done by initially viewing the impact of the software resources on the following levels:

- Impact on the application performance baseline (before the Transaction 2 optimisations).
- Impact on the baseline after the Transaction 2 optimisations.
- Impact on the baseline with different ServerE instance configurations.

7.1 Software Resources - Impact on Baseline

The first perspective required, was to see what impact ServerE and ServerA had on the application baseline with a production environment workload.

Fig. Interactive Software Resources – Impact on the Application Performance Baseline.

Note that the above picture denotes the application performance when running with a single instance of ServerA and two instances of ServerE.

Observations:

- ServerE instances would be overloaded.
- Response times consequently are extremely long.
- The Login and Logout transactions, which depended on ServerA appear to be unaffected, as expected.
- Throughput drops down from 16 transactions per second to 9 transactions per second.

7.2 Software Resources - Impact of Transaction 2 Optimisation

The next perspective, was to see what impact ServerE and ServerA had on the application following the Transaction 2 (Case Selection with Sort) optimizations.

Fig. Interactive Software Resources – Impact on Application Performance after Transaction 2 Optimisation.

ServerA and ServerE were running on the Mainframe server, configured with one and two instances respectively. ServerE impact consisting of waiting time for that server is denoted by the pink tips of the horizontal bars above.

Observations:

- The improvements made to Transaction 2 not only reduced the Mainframe CPU usage for Transaction 2 Sorts, but have resolved the unacceptably long response times previously encountered in the baseline projection due to ServerE overload.
- Note that the ServerE capacity (two instances) was not changed yet. Still their %utilisation dropped considerably.
- All transactions still show waiting times by ServerE impact. Some of them show considerable waiting times namely:
  - 5 B Order Assignment to Unit B,
  - 6 C Start Order C
  - 8 B Complete Order B
- Though response times look acceptable now we know that with such waiting times they may be unstable.

7.3 Software Resources – Impact of Increasing ServerE Instances

One of the questions presented by the application development team, was to consider the impact of running the application with a greater number of server E instances, given that the instances are single-threaded.

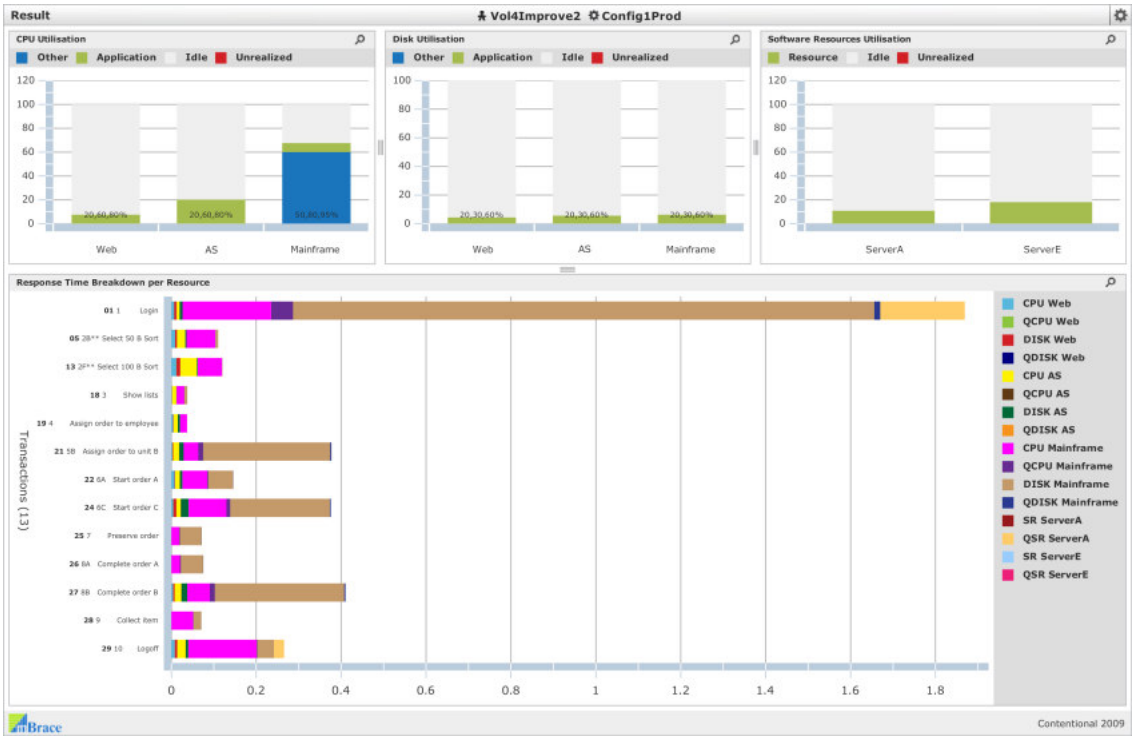


Fig. Interactive Software Resources – Impact after ServerE Instance Reconfiguration.

The above graph depicts the application transaction results after the number of ServerE instances were increased from 2 to 8.

Observations:

- Login and Logout transactions appear unaffected by the changes, but they interact exclusively with ServerA.
- ServerE response time contributions to the response times of all other transactions appear to be eliminated.

It was therefore suggested that ServerE be configured with at least 8 instances in production.

Knowing the dynamic behaviour of software servers, concerns were raised with regards to the stability of ServerE.

This concern was investigated in the next section.

7.4 Software Resources - ServerE Stability Investigated

When for one reason or another a component of the infrastructure chain causes or experiences some extra delay the software server may be overloaded.

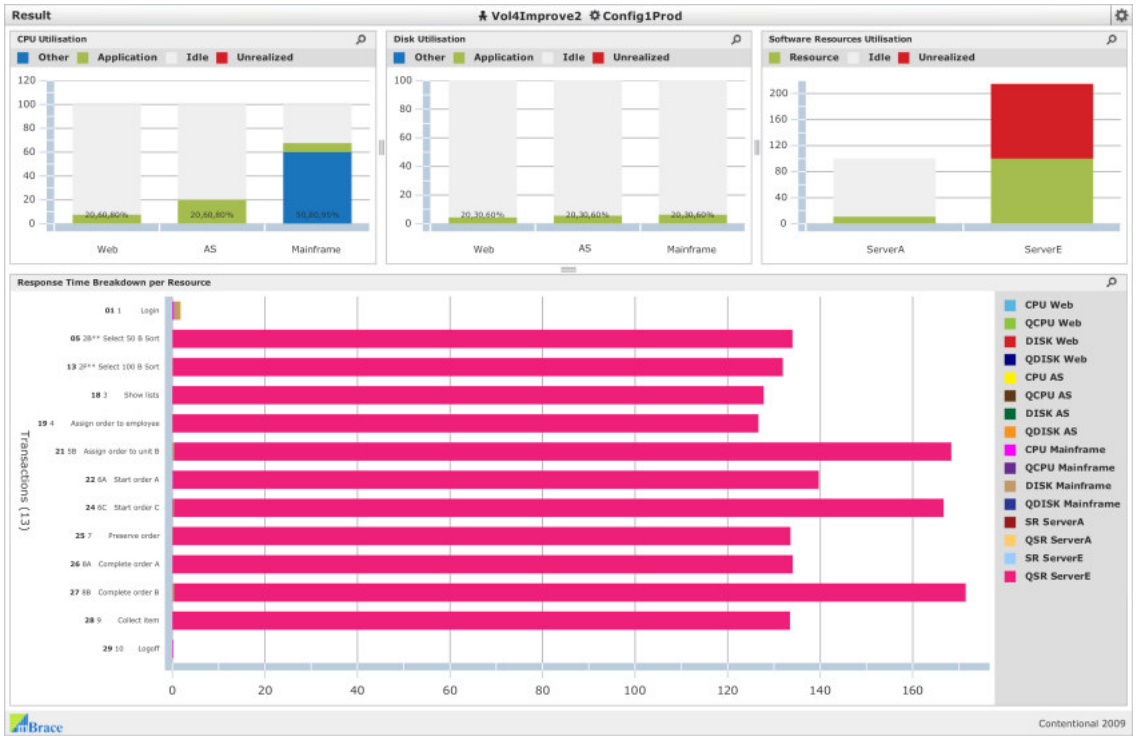


Fig. Software Resources – Impact due to Delay

The model allows for the ability to delay response times artificially in order to mimic delay behaviour in the application. In this specific case, the response times were extended artificially by one second. This is indicative of a possible occurrence when for one reason or another a component of the infrastructure chain causes some extra delay.

Note that serverE was running with 8 instances.

Observations:

- Login and Logout transactions appear unaffected by the changes, but they interact exclusively with ServerA.
- ServerE instances are overloaded causing excessive transaction response times. This indicates that software server ServerE is sensitive to small disturbances when running with 8 instances.

At this point, an increased number of instances for ServerE were tried in the model in order to resolve the sensitivity issue with ServerE and create stability.

7.5 Software Resources – Sensitivity Analysis – More ServerE Instances

After trying a number of combinations, it was discovered that application stability was attained when running the "amended" transactions with 40 ServerE instances, contrary to expectations.

The model output of that configuration is shown below.

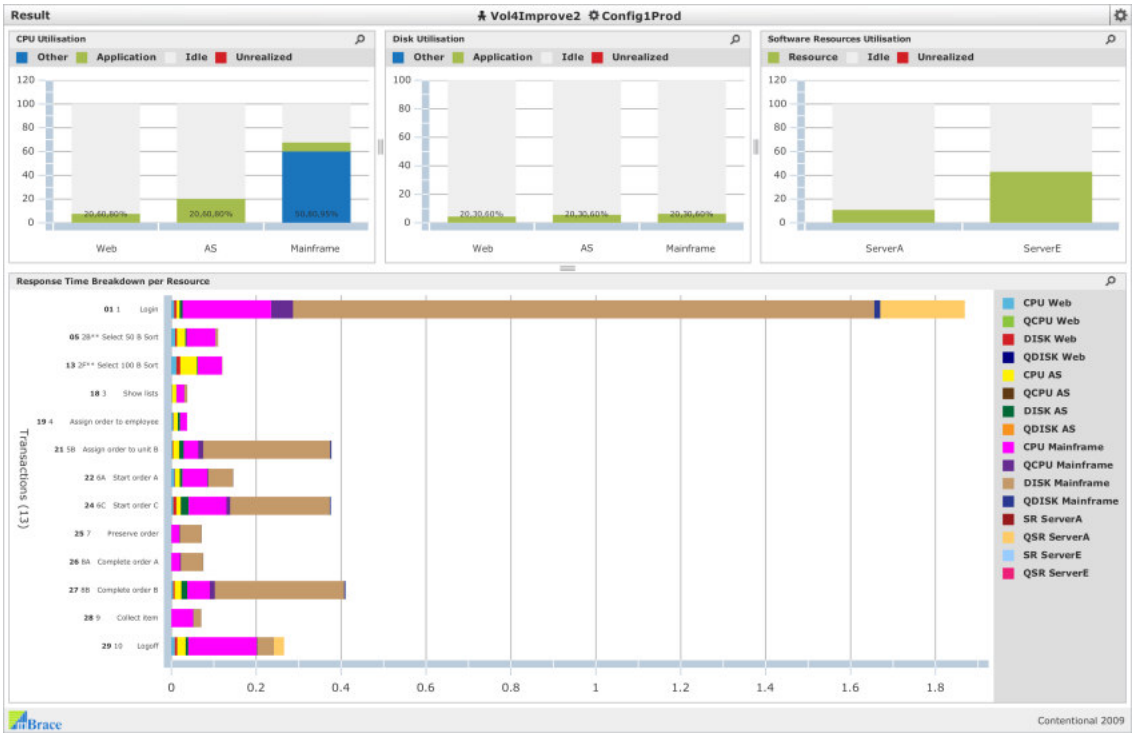


Fig. Software Resources – Impact due to Delay

In the above model output, application transactions have been delayed by a second, and ServerE was running with 40 instances.

Observations:

- ServerE impact on application response times had been eliminated.

The memory configuration requirement of running with 40 ServerE was also studied though not shown here. The increases in memory capacity usage were considerable but did not require additional memory capacity.

8 Conclusion

The case study shows an example of analysing the capacity needs and performance of a newly built application applying Transaction Aware Performance Modelling (TAPM). The transaction awareness of the model allowed us to identify significant room for improvement of efficient use of hardware by one of the transaction types of the application. The project manager of the application development project was advised from the outcomes of the study to have (at least) one transaction type improved for efficient use of the hardware. This resulted in improved efficiency and in considerable savings on mainframe CPU capacity. The costs of implementing these improvements were minimal.
Further, the performance behaviour of two single threaded software servers was analysed. The performance of these software servers showed to be all right, however its stability was not sufficient. This was due to the configuration of software servers used. This problem could be resolved easily by deploying more instances of one of the software servers.