



## SHUNRA PERFORMANCE ROOT CAUSE ANALYSIS REPORT

Prepared for Business Unit on the REPORTING application  
Friday, May 7<sup>th</sup>. 2009



## SHUNRA ENGAGEMENT REPORT

### REPORT INFORMATION

<b>Customer:</b>	Business Unit
<b>Application:</b>	REPORTING
<b>Document Type:</b>	Performance Root Cause Analysis Report
<b>Author:</b>	Amichai Lesser
<b>Document Revision:</b>	1.0
<b>Revision Date:</b>	Thursday, May 07, 2009
<b>Performance Status</b>	<b>Orange</b>



## SHUNRA ENGAGEMENT REPORT

### TABLE OF CONTENTS

<b>REPORTING LATENCY TEST RESULTS</b>	<b>4</b>
<b>OVERVIEW</b>	<b>4</b>
<b>KEY FINDINGS</b>	<b>4</b>
THE EXCEL MACRO TRANSACTION	4
S DRIVE NAVIGATION	5
<b>SUPPORTING DATA</b>	<b>6</b>
DEEP TRANSACTION ANALYSIS OF THE EXCEL MACRO TRANSACTION	7
NETWORK FOOTPRINT ANALYSIS FOR THE PERFORMANCE CHALLENGED TRANSACTIONS	11
ERRORS AND WARNINGS	11
<b>RECOMMENDATIONS</b>	<b>12</b>
S DRIVE NAVIGATION	12
EXCEL MACRO	12
<b>NEXT RECOMMENDED STEPS</b>	<b>13</b>



## REPORTING LATENCY TEST RESULTS

### OVERVIEW

In preparation for the data center migration to the Main Data Center/Contingency Data Center the REPORTING application was tested. The test goal was to assess the performance of REPORTING transactions across latency conditions from Head Office to the Contingency Datacenter and to the Main Datacenter. The test was conducted by introducing a network emulator (Shunra VE) with 30 milliseconds (Contingency Datacenter) and 2.5 milliseconds (Main Datacenter) of latency between the REPORTING Client and the server that is hosting the REPORTING application.

The performance test on May 1<sup>st</sup>. identified several transactions that would experience noticeable performance degradation under the latency conditions to both the main and the contingency datacenters. This report presents the analysis of the root cause behind the performance degradation for those transactions as well as recommendations on how to mitigate the performance bottlenecks.

### KEY FINDINGS

The following describes the root cause behind each of the performance degradation for each of the transactions:

---

#### THE EXCEL MACRO TRANSACTION

1. The first 30 seconds of the Excel Macro is spent on downloading 31 Tables one at a time. Each table download is blocking the next table download from proceeding.



## SHUNRA ENGAGEMENT REPORT

2. The Excel Macro spends several minutes on downloading the same Global\_Macro.ini file 31 times. This redundancy contributes to a 50% increase in the response time.
3. The Excel Macro opens a new Excel file for each report that it generates, only to consolidate all the reports into one file at the end. The process of opening an Excel file in this network environment involves a session with the print server (to retrieve fonts and page layout information) this session is very latency sensitive and contributes at least 50% to the response time increase. Repeating this session 31 times (one for each report) significantly adds to the latency sensitivity of this transaction.

---

### S DRIVE NAVIGATION

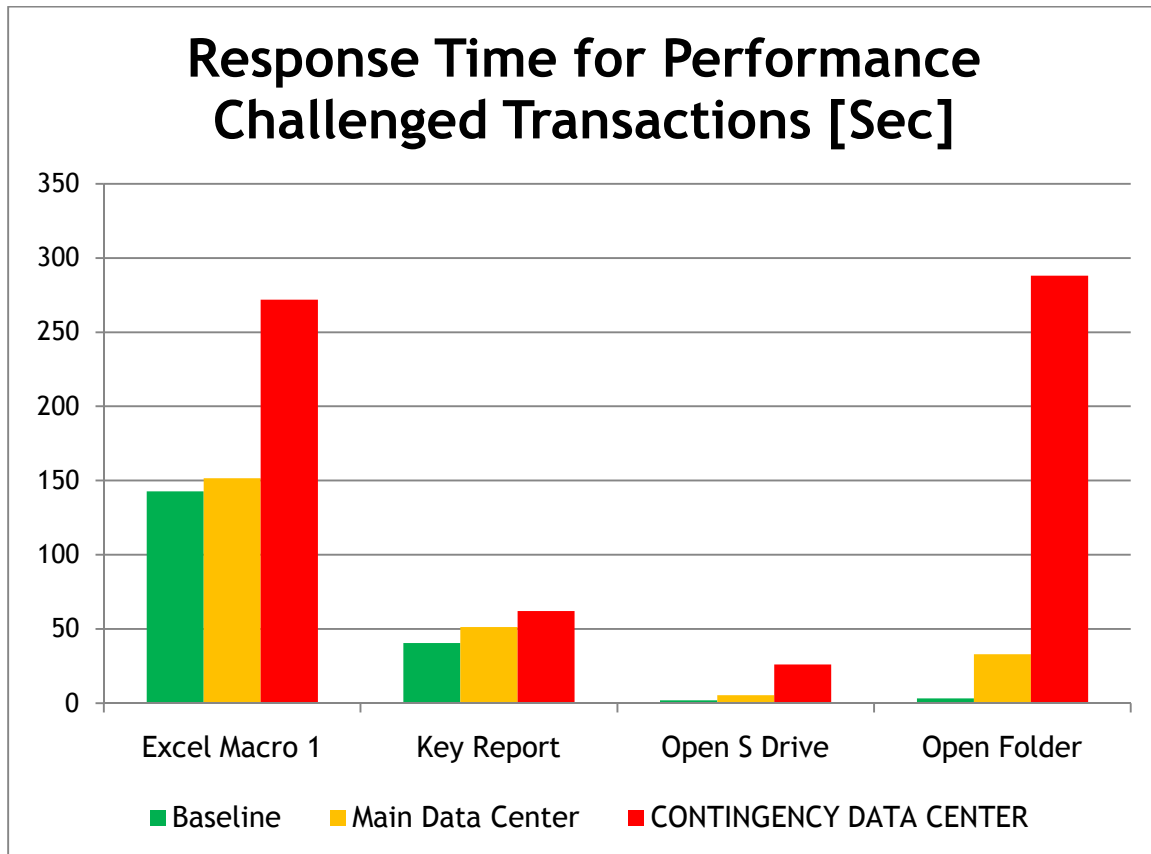
4. The S Drive navigation uses a proprietary Oracle plug-in that pulls the folder content, file by file, so a folder with a few hundred files will experience a 60 msec latency penalty for each file in that folder adding up to the several minute wait time just to navigate through the S drive.



## SUPPORTING DATA

The following presents the response-time test results for the performance challenged transactions (copied over from the performance test results report)

<b>REPORTING Transaction (The problematic transactions)</b>	<b>Baseline [TRT in SEC]</b>	<b>Main Data Center [TRT in SEC]</b>	<b>Contingency Data Center [TRT in SEC]</b>
Excel Macro 1	142.750	151.546	271.9
Key Report	40.562	51.328	62.15
Open S Drive	1.984	5.468	26
Open Folder	3.234	32.968	288



DEEP TRANSACTION ANALYSIS OF THE EXCEL MACRO TRANSACTION

The entire transaction took 272 seconds in the Contingency scenario

Start Time [Sec]	End Time [Sec]	Operation
0 second	32 seconds	This is an Oracle forms web session that downloads the 31 table files, this is done in serial one after the other, a better download process via a backend web service can crash about 15 – 20 seconds from this part.

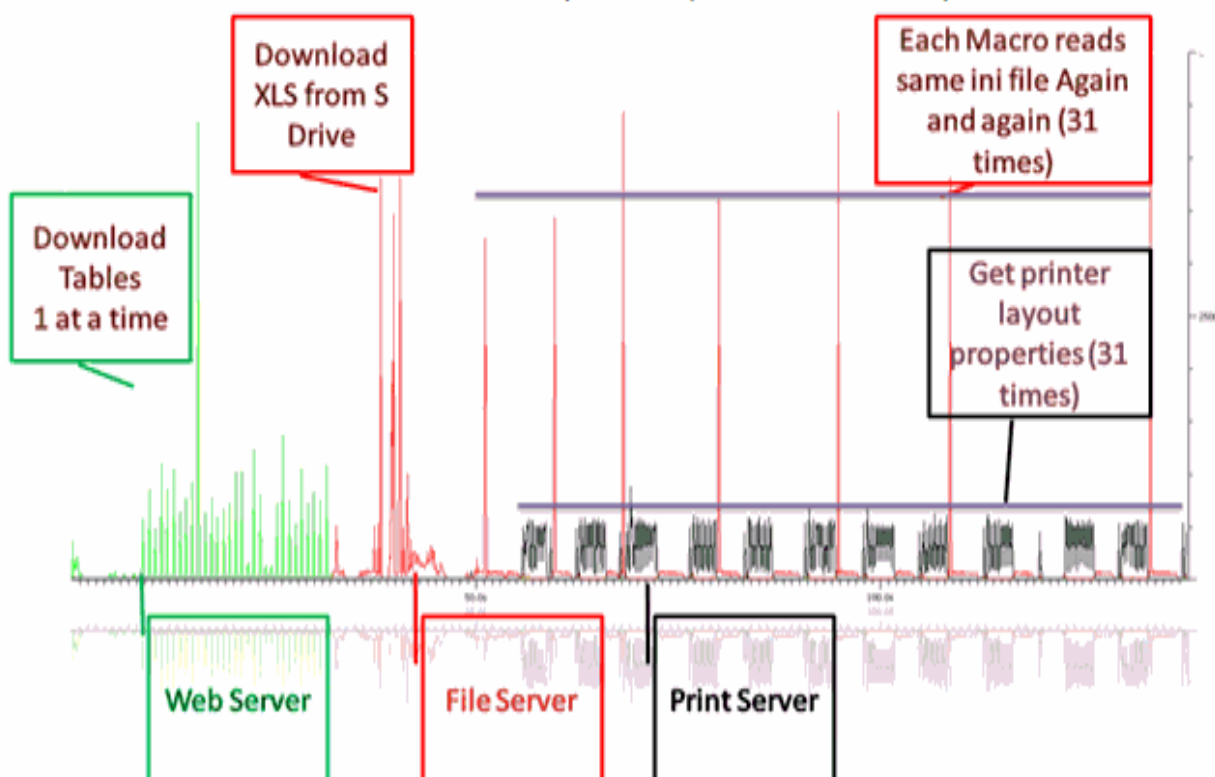


SHUNRA ENGAGEMENT REPORT

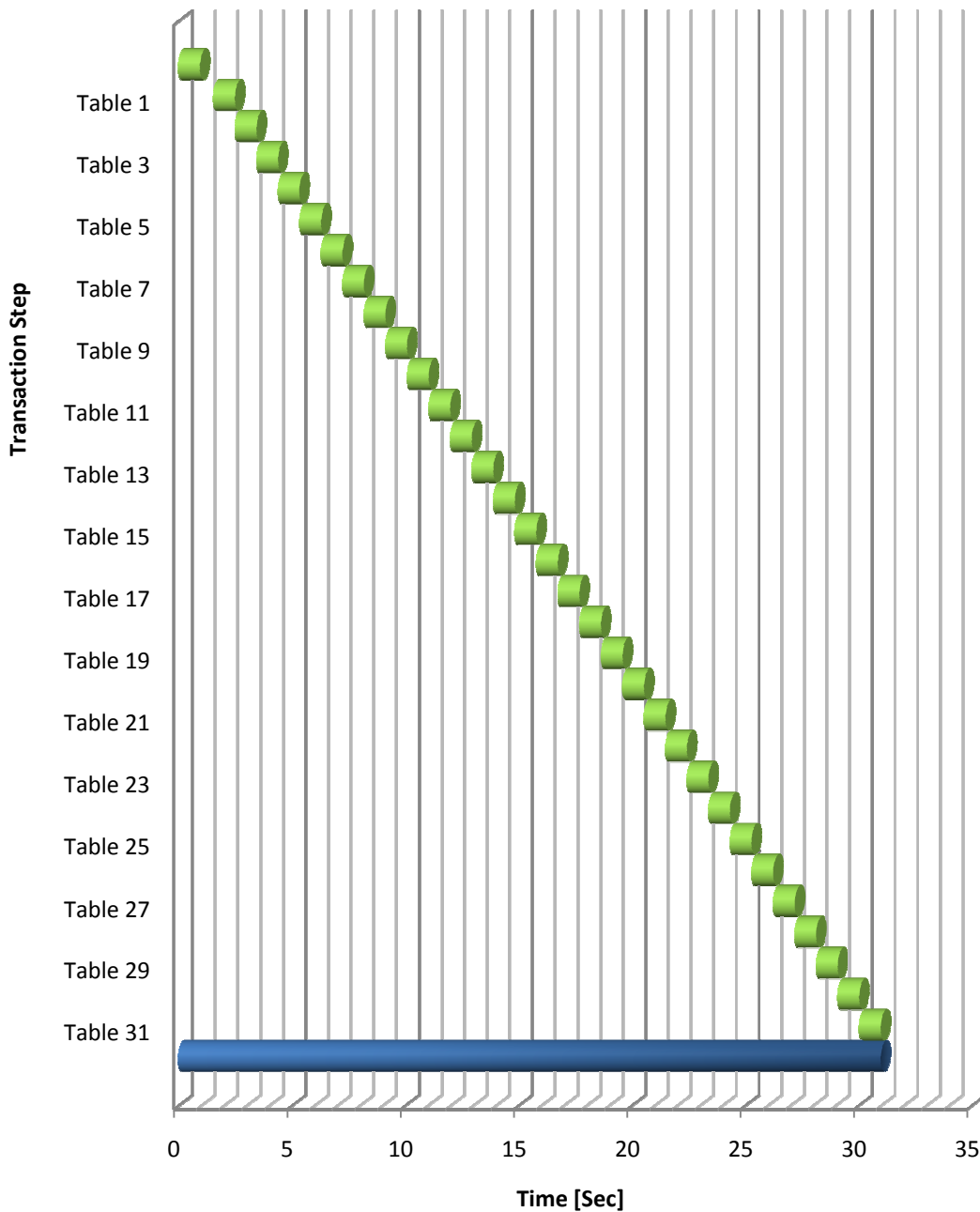
33 seconds	46 seconds	Excel.exe launches and opens a file from the S drive called Global_Macro.xls, this actually causes the XLS file to download to the client. There is also a 2 second gap as the user is prompted to allow Macros to run (a security setting). Another file called Desktop.ini is fetched from the S drive
46 seconds	49 seconds	Client processing time
49 seconds	55 seconds	The first workbook macro reads data from the S drive from a file called Global_Macro.ini
55.5 seconds	59.5 seconds	Excel opens a new file for each table and reads printer property data from a print server (fonts and page layout)
59.5 seconds	271 seconds	The above 2 steps repeat for each of the 31 tables. This causes the same Global_Macro.ini file to be read again from the S drive and the printer properties are fetched again since Excel opens a new table file.

271 seconds	272 seconds	All 31 table files are consolidated into one file
-------------	-------------	---

Below is a timeline chart of the above process (first 150 seconds):



### Excel Macro – Blocking downloads of table files in an Oracle Forms process



NETWORK FOOTPRINT ANALYSIS FOR THE PERFORMANCE CHALLENGED TRANSACTIONS

The following network footprint analysis highlights the latency sensitivity of the transactions as indicated by application turns and the transaction size

Transaction	Transaction Duration time [sec]	App Turns	Traffic details			Protocol Overhead [%]
			Total KB	KB per App Turn	Total Packets	
<b>Excel Macro 1</b>	<b>271.9</b>	<b>7411</b>	<b>5299.36</b>	<b>0.72</b>	<b>15731</b>	<b>11.88</b>
Key Report	62.15	35	235.87	6.74	386	6.59
Open Navigation Screen	17	32	20.49	0.64	80	16.07
Open S Drive	26	659	120.02	0.18	784	26.13
<b>Open Folder 2</b>	<b>288</b>	<b>8828</b>	<b>2054.77</b>	<b>0.23</b>	<b>12469</b>	<b>24.28</b>
Open Testing Folder	12.5	295	53.86	0.18	299	22.2
Open File	5.953	51	22.1	0.43	107	19.59
Start Loading	14.64	32	17.89	0.56	72	16.19

ERRORS AND WARNINGS

As mentioned before, the Excel macro makes redundant calls. Some of the redundancy can be seen from the below warning report:

Protocol	Error Type	Error Description	Number of Errors
HTTP	Not Cached	Found multiple responses instead of a Not Modified (304) response	33



## SHUNRA ENGAGEMENT REPORT

Sub-Transaction	Error / Warning	Description	Number of Errors
POST	Not Cached	Found multiple responses instead of a Not Modified (304) response	33

## RECOMMENDATIONS

### S DRIVE NAVIGATION

1. Upgrade the Oracle file navigation module to improve the S drive navigation performance.
2. Until a new file navigation module is implemented, use direct path links for each file instead of using the S drive navigation application.

### EXCEL MACRO

3. Co – Locate the Global \_Macro.ini file, Global\_ Macro.XLS and the Tables on the client (by downloading the ini file at the beginning of the transaction and referencing it locally in the Macro code, just like it already does for the tables), this should reduce over 30 seconds from the transaction but may transfer the bottleneck to the client so exact numbers are hard to predict, a quick test can verify the exact savings. [\[update Aug 2009 : a new version with this implementation was tested yielding improved performance for both no latency baselines \(response time was cut in half\) as well as remote datacenter simulations \(over 10X improvement in response time\)\]](#)
4. Do not open a new Excel file for each table, instead have one Excel file read the table data into a new tab, this will eliminate the redundant calls to the print server.
5. For the initial retrieval of the 31 tables, consider a backend web service/stored procedure that reads the ini file once, determines which tables are needed, builds them in 1 XML file and pushes that



## SHUNRA ENGAGEMENT REPORT

file to the client via 1 get request. This should reduce approx 20 seconds from the transaction.

6. If code change isn't feasible consider defining a local printer (even a print service like a fax or PDF) as the default printer, this way the page margin print properties will not be fetched over the network. Update [6-15-2009] – testing this option reduced 90% of the transaction time.
7. Test S Drive Access and the Excel Macro with the WAAS setup. Update [6-15-2009] testing this option reduced 50% of the transaction time.

### NEXT RECOMMENDED STEPS

1. Meet with business unit to discuss the recommendations and determine next steps (PM to schedule) – DONE
2. Test the various implementation options. [options 3, 5 and 6 were tested]