Performance Testing: SAP Volume Testing Facts

Author : Anshuman Das
Email: anshudas@in.ibm.com

Address: IBM Global Services, India
Subramanya Arcade,
No. 12, Bannerghata Main Road
Bangalore 560 029

Abstract

SAP is the most successful product for enterprise resource planning (ERP), touching nearly every facet of business like accounting, logistics, CRM, HR, data warehouse, etc. In many companies, transactions of thousands of users must be processed concurrently by SAP and the underlying database system. Hence, it is truly a mission-critical technology that requires solid performance and high availability to deliver the level of service the business requires. Failure of this critical system introduces tremendous business risk, both to the successful execution of internal processes and to the service level agreements as established.

The purpose of this paper is to give an overview of various facts and to large extent the best practices followed during SAP volume and stress testing. This paper describes best methods of creating automation scripts and experiences from real time execution of various SAP applications involving modules like Sales & Distribution, Material Management, Human Resource Management, CRM, Business Warehouse. The paper also shows how an SAP system can be monitored during volume test. This paper is structured in a format to give an overview of the activities to be done before the test, during the test and after the test. LoadRunner 8.1 from HP-Mercury is the automation tool used to create automation scripts and mimic end user workload.

Table of Contents

1. Test Engagement

The entry criteria for volume and stress in any SAP system are due to following reasons.

- Increasing the number of users or adding new users
- A major software upgrade like database or operating system migration
- A major hardware upgrade like implementing BIA infrastructure for BW
- Implementing a new module or new functionality like accessing through SAP Portal instead of SAP GUI

The common concern which arises in minds of project team before any SAP application go live can be classified as follows.

- How does the database react with existing or desired concurrent SAP users?
- Will the implemented SAP application meet the end user performance requirements with respect to response time and throughput?
- Is this SAP implemented solution scalable?
- Will there be any dumps related to RFC connections and local printing hanging issues during production operation?
- Can BIA infrastructure improve the performance of BIW queries?
- Can specified number of CRM users create desired volume of tickets per hour?
- Is this implemented solution capable of handling estimated production level transaction volumes within the available time frame?
- Was the hardware sizing done accurately?
- Are Application Servers and Database Servers configured accurately to withstand desired number of processes or requests?
- Which component can fail for what time?
- Which component of the system limits the number of concurrent users?

In every volume test the engagement model begins in the following manner (refer Figure 1).

- Request from Project Team: The request can be made in various ways like Service Request (SR), by email or phone call and sometimes project team member walking in with request for testing.
- Initiate Kick off meeting: The testing team should initiate kickoff meeting to discuss and understand requirements. During this meeting the scope and objectives of testing should be defined.
- Explain the strategy of testing to project team as per the requirement.
- Create Service Request with a unique id.
- Create the test plan.
- Assign resources as per availability in pool and required skill set.

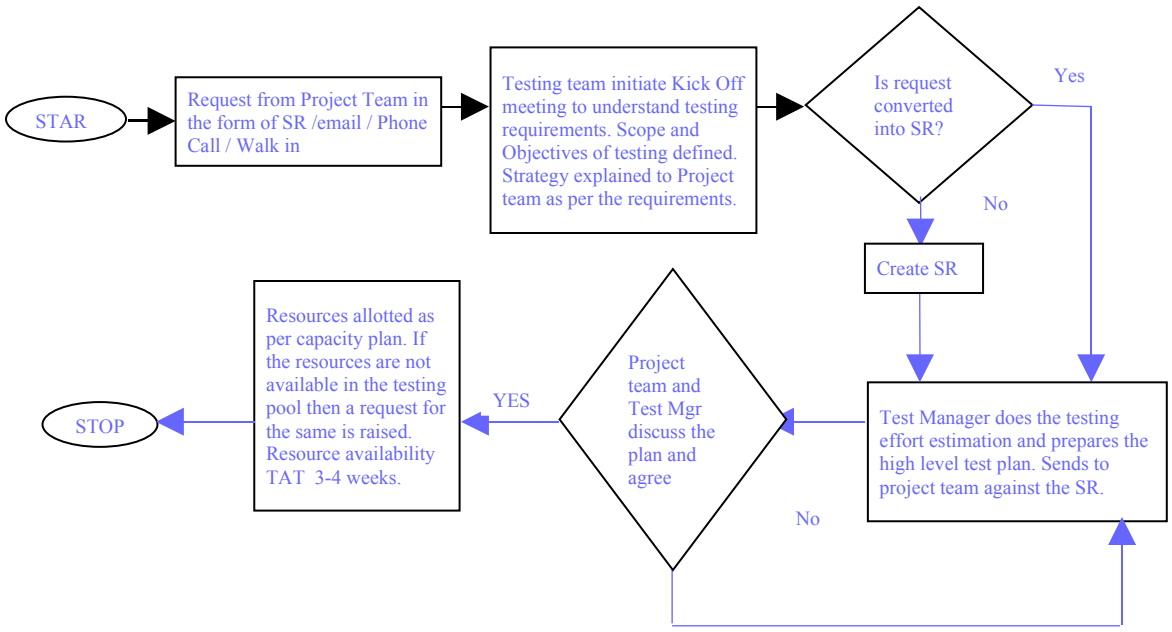


Figure 1: Volume Testing Engagement Model/Framework

2. Effort Estimation

To estimate an effort for volume testing is a key task. The effort can be measured based on 3 phases.

Phase I: Script Design

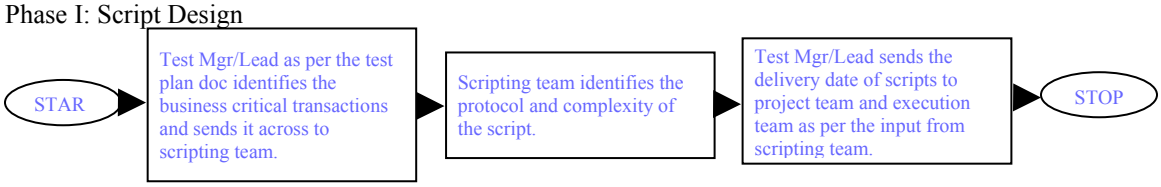


Figure 2: Script Design

According to Figure 2 the script complexity is measured by the scripting team. Effort involved is estimated based on this scripting. As a best practice, the scripting effort is designed using a formula keeping in mind the complexity level.

Considering a medium complex script a script analyst can cover 1 script in 2 days. Hence the formula is

*# of Scripts * 2 = # of Man Days*

Note: This formula also considers script analyst skill in LoadRunner scripting.

Phase II: Test Environment Design

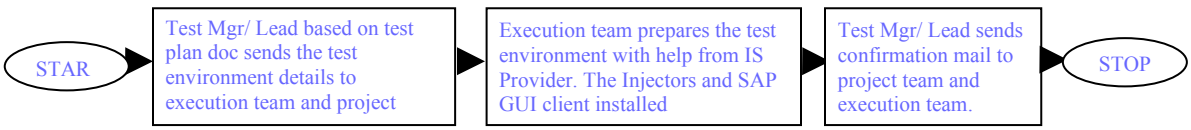
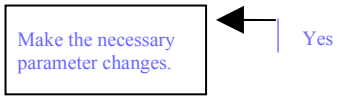


Figure 3: Test Environment Design

Test environment setup is joint venture between execution team and IS Provider. Based on requirements captured in test plan document the environment is prepared. This involves setting up:

- SAP test environment: It includes identifying and creating a landscape similar to production environment consisting of Application Servers and Central Instance with Database.
- SAP test data: Load copy of production data in test environment. Data copying involves few steps to be followed in order to be sure that the data copied is of equal size in the test environment. If we copy data to different environment, say production to QA, then it is normal backup from production and then restore in QA. And if we want to copy data to different platform (say Windows to UNIX) or different DB we need to use "r3load". This is a tool from SAP to import and export table during installation. In certain cases the data used should be confidential. As a result scrambled data is prepared using a third party tool called DSM (Data Sync Manager from EPI-USE). This tool is used to scramble and depersonalize data.
- LoadRunner Infrastructure: The number of injectors to simulate or generate the load is prepared based on number of concurrent users captured in test plan document. As a best practice in case of SAP R/3 50 virtual users can be simulated from a single injector with configuration of 512 MB RAM and 40 GB hard disk. With similar configuration as a best practice in case of SAP BIW 20 virtual user can be simulated from a single injector. To get better and consistent connectivity between LoadRunner controller and injectors, it is advisable to have same version of LoadRunner package and fix pack installed in controller and injector.
- Enable scripting at client: The scripting option should be enabled in SAP GUI client. If the scripting option is disabled the LoadRunner scripts can be neither created nor executed. In order to enable scripting at client level go to Run → regedit. Set the following parameter value to 1.
    - [HKEY_LOCAL_MACHINE\SOFTWARE\SAP\SAPGUI Front\SAP Frontend Server\Security]
      UserScripting = 1
- Enable scripting at server: The scripting option can be enabled at server level by executing transaction “rz11”. Check for parameter “sapgui/user_scripting” and set it to ‘True’.

Phase III: Execution Phase



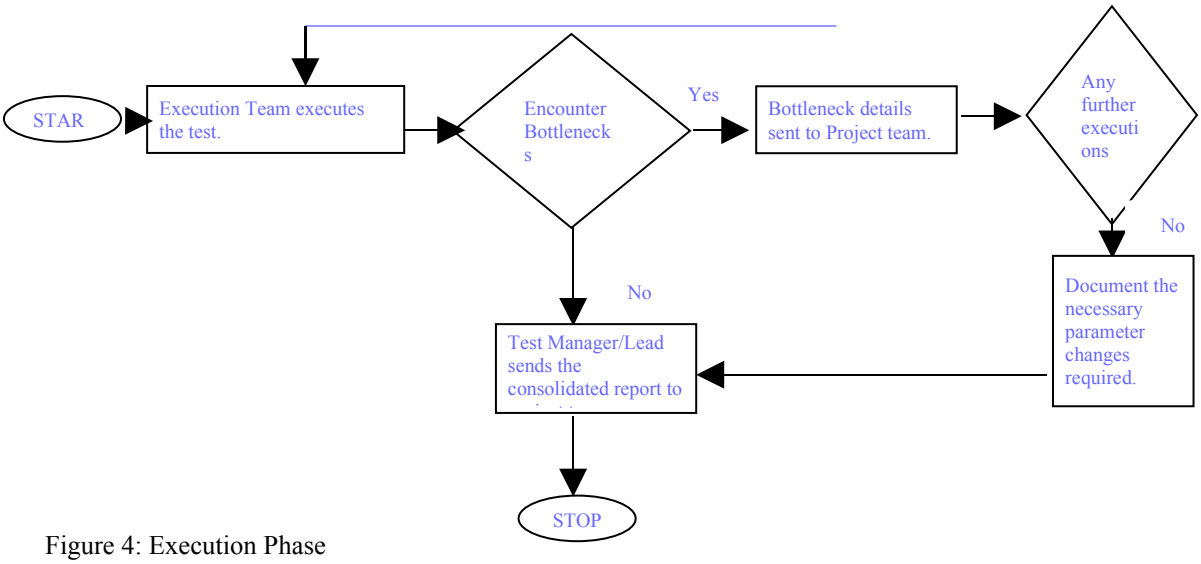


Figure 4: Execution Phase

During execution phase a specific execution plan is created considering all the activities involved during execution. This master execution plan is captured in excel sheet with following worksheets:

- Project Activities: This work sheet consists of details tasks or activities to be accomplished before and during execution.
- Script Plan: This work sheet consists of all the scripts as per defined business critical scenarios.
- Scenarios: This work sheet consists of various types of execution scenarios to be executed during the volume testing. These scenarios are captured initially in test plan document.
- Injectors: This work sheet consists of all the injector details with IP address or machine name and location where they are installed. This list also consist of spare or backup injectors
- Systems and Monitoring: This work sheet consists of detailed information of Servers which will be monitored and parameters which will be considered for monitoring.
- Stakeholders: This worksheet is replica of contacts info section in test plan document. This information helps to contact right person during execution if any problem arises.
- Test Log: This worksheet is updated by the execution team as a best practice to maintain a log of activities done during execution. For example during execution if any bottleneck is found in application server then extra application server is added and again tested. This kind of details should be captured in test log which during final reporting helps to prepare a good report.
- Actions Issues: Any action items to be taken care by stakeholders or issues encountered during any phase of volume testing are registered in this worksheet.
- Lessons Learnt: The captured data from lessons learnt are documented in SAP Volume Testing Best Practices for future reference.

3. SAP LoadRunner Scripting

SAP scripting begins with creating few utility scripts like users creation, resetting password and finally scripts for identified business critical scenarios. In order to create users, initially a template user should be defined with help of functional team or consultant.

Caution: In order to create users the user id should have SAP_NEW access. The users should be created as a replica of template user defined by functional team. The template user is defined based on the business critical scenarios.

Caution: No test user should be created with SAP_ALL access. A user with SAP_ALL security access goes through least security check during login.

4. What & how to record?

LoadRunner scripting is done in recording environment called as Vugen. Vugen is a component of LoadRunner. SAP scripting is done with a specific protocol called "SAPGUI" or "SAP-Web"(refer Figure 5) . As a best practice the login and logout are recorded in "vuser_init" and "vuser_end" sections respectively. Rest other SAP transactions and other navigations are recorded in user defined Actions which are reusable and can be iterated.

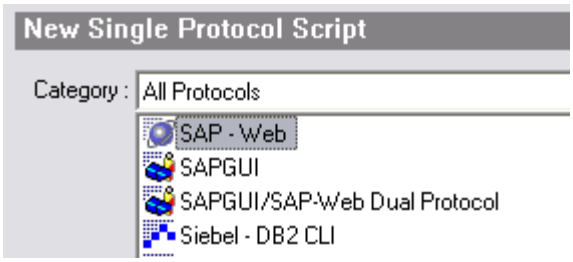


Figure 5: LoadRunner protocol menu

The script correlation should be taken care to avoid duplication of dynamic value. Automatic correlation will help to make the script more dynamic to deal in a better way against Session_ID's, database primary keys and almost all security done over HTTP in case of SAP-Web.

Parameterization is another factor which should be handled carefully.

5. SAP Scripting Challenges

LoadRunner scripting challenges while using SAPGUI or SAP-Web protocol arises due to many factors. This may either due to functionality or due to custom developments.
Before starting recording using Vugen it is always advisable to change the recording options.
In Vugen go to Recording Options → General → Capture Screen Shots → None. This will make the script much faster to record.

Lessons leant:

Lesson Learnt 1: While performing a Purchase Order (PO) transaction it becomes necessary sometime to capture PO number to pass onto subsequent transaction. Manually add the sapgui_status_bar_get_param to retrieve the parameter from the status bar. It is necessary to be careful to pick up the right PO number

Lesson Learnt 2: If you are going to record in multiple sessions you can use the Record at the Cursor option - which plays the script until that point, and then you can record from that point – useful for SAP where there may be several long-winded transactions as pre-requisites for that transaction.

Lesson Learnt 3: One action should be created per SAP Screen.

Naming convention:

Xnnn_TransactionName_shortdesc

Where
x = unique script code (every script in the load test has a different code)
nnn = sequential number

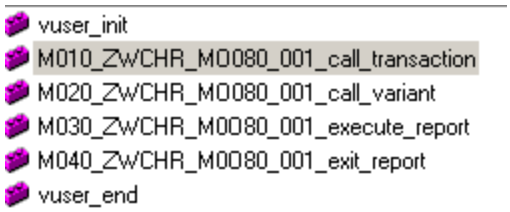


Figure 6: LoadRunner Actions

Lesson Learnt 4:

For every Action

- use lr_start_transaction('action_name')
- and lr_end_transaction('action_name')

Put all think time outside of this transaction. This will allow us to track the actual transaction response time during the test.

Lesson Learnt 5: Recovery from errors

The script can be instructed to exit the iteration and start the next iteration on error. We need to include some code at the top of the first action to select the window and enter the neutral transaction ns000.

The instructions in the script will override the runtime settings of continue on error (which don't make sense in a long script with dependencies as we know that nearly all the subsequent steps will fail).

Include the following in the first lines of the first action:

```
// start of fix continue on error
        lr_continue_on_error(3);

        sapgui_select_active_connection("con[0]");
        sapgui_select_active_session("ses[0]");
        sapgui_select_active_window("wnd[0]");
        sapgui_set_ok_code("/ns000",
                BEGIN_OPTIONAL,
                        "AdditionalInfo=sapgui408",
                END_OPTIONAL);
// end of fix
```

Include the following in the last lines of the last action:

```
lr_continue_on_error(0);
```

6. SAP Servers Monitoring

SAP servers can be monitored either through LoadRunner or manually executing defined transactions in SAP.SAP monitoring is one of the most important task in SAP Volume Testing. As a best practice a caution should be taken not monitor too many parameters using LoadRunner controller. Too many parameters generate huge amount of data and log files. As a result at the end of execution while LoadRunner is collating data from all injectors and servers there are high chance of controller crash or incomplete collation of results.

Parameters to be monitored

Application Server:

- % Disk Time
- % Processor Time
- File Data Operations/Sec

- Interrupts/Sec (Memory)
- Page Faults/Sec (Memory)
- Pages/Sec (Memory)
- Processor Queue Length

SQL Database Server

- % Disk Time
- % Processor Time
- Pages/Sec (Memory)
- Processor Queue Length

Oracle Database Server: Import tnsnames.ora file in controller to connect to Oracle database.

- DBWR transaction table writes (V$SYSSTAT 1)
- physical reads (V$SYSSTAT 1)
- physical reads direct (V$SYSSTAT 1)
- physical writes (V$SYSSTAT 1)
- physical writes direct (V$SYSSTAT 1)

UNIX Server: If central instance is in UNIX.

- Average load (Unix Kernel Statistics)
- CPU Utilization (Unix Kernel Statistics)
- Disk Traffic (Unix Kernel Statistics)
- Paging rate (Unix Kernel Statistics)

Monitoring SAP transactions

- To monitor user distribution in application servers and central instance execute transaction “AL08”.

- To view runtime errors during volume testing execute transaction “SM21” & “ST22”. Figure 7 and 8 shows execution of SM21 transactions.

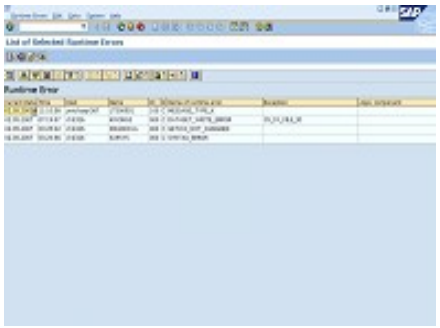


Figure 7: SM21

Figure 8: SM21 Detailed

- To monitor all RFC execute transaction “SM66”.

- To monitor dialog and report user steps executed after volume testing execute transaction “ST03N”. This transaction is a crucial transaction as this helps to calculate the volume of transactions done in the whole testing. Figure 9 shows clearly the number of dialog user steps and report user steps executed during volume testing.

Report/Transaction	# Steps	T Response Time / s	Ø Response Time / ms	T CPU Time / s	Ø CPU time / ms	Ø DB Time / ms
PA20	205	208	1,013.3	49	241.1	302.0
PA30	268	427	1,594.6	88	327.4	483.6
PA40	6,131	3,606	588.2	826	134.7	143.2
PO13	1,121	876	781.0	182	162.3	146.9
PP01	1,157	1,711	1,478.7	350	302.2	276.2
S_AHR_61016129	131	4,898	37,390.3	2,132	16,274.7	7,767.0
S_AHR_61016494	186	230	1,236.5	96	517.8	144.4

Figure 9: ST03N to calculate dialog and report steps.

In the above figure PA20, PA30, PA40, PO13 and PP01 are dialog user steps. Whereas S_AHR_61016129 and S_AHR_61016494 are report user steps.

7. Case Study

Based on above best practices here is a case study based on volume testing conducted for a ARA oganization.

Organization Experience

ARA is a world famous manufacturing company. They are the oldest and well known company in their niche of manufacturing and production. They employ more than 85,000 people worldwide with 50 factories in different locations and product sold in 180 countries. They are about to implement a complete SAP HR solution. The objectives of this implementation are as follows

- Improve the HR service level to ARA and its employees
- Reduce the overall business complexity and operating costs
- Focus on HR value-added activities and their support to the business
- Implement a better payroll system
- Implement a better reporting tool using BIW

Sample SAP HR System for ARA

Figure 10: Sample SAP HR system

IBM's Proposal

Test Objectives

As per the best practices the goal of the test was identified and was documented in test plan document during kickoff meeting. The goal of the tests were

- To run 300 HR users in central instance and application servers.
- Ensure running a load on the backend.
- Ensure with load on backend the desired number of transactions are executed in specified time.
- Ensure CRM application can support creating 500 tickets/hour with 100 concurrent users.
- Ensure there is a drastic difference between executing BIW queries with and without BIA infrastructure.

Scoping of test

The following SAP interfaces were identified from the above landscape to conduct volume testing

- Standard SAP foreground reports –SAP GUI
- Custom Development and Queries – SAP GUI
- Backend transactions – SAP GUI
- e-learning – *Content player*
- CRM – Telephony and Web interface
- Pay stub display
- Time entry and display / leave request
- BIW (Web)

Scoping of Scenarios

With reference to above interfaces the business critical scenarios were identified.
A group of complex, medium and simple scenarios were selected. The scenarios were

- PA20 - DialogUser
- PA30 - DialogUser
- PA40 - DialogUser
- PO13 - DialogUser
- PP01 - DialogUser
- S_AHR_61016129 – Long Reports
- S_AHR_61016494 – Long Report
- S_PH9_46000223 – Long Reports
- YBPMOM00_FUNTDATAKEY – Short Reports
- YBPMOM01_HCOUNT_CC – Short Reports
- YBPMOM01_HEADCNTACT – Short Reports
- YBPMPA00_EMP_LIST – Short Reports
- YBPMPA01_SERV_ANNIV – Short Reports
- ZBMXOM00_HEADCNT – Short Reports
- ZBMXPYO_HR2U_SIPRE_022 – Short Reports

Users Distribution

Transactional Users : 243
Long Running Reports : 7
Short Running Report : 50

Execution Results

- There was a peak in response time when users logged in.
- Dialog or transactional users showed good response times without the report users.
- Report users were loaded after dialog users.
- Both report and dialog users experienced longer response time.
- CPU reached 99% in central instance when report users were loaded (refer Figure 11).
- The transaction frequency decreased when report users logged in (refer Figure 12).
- It was concluded that report queries should be tweaked before we conduct next phase of testing.

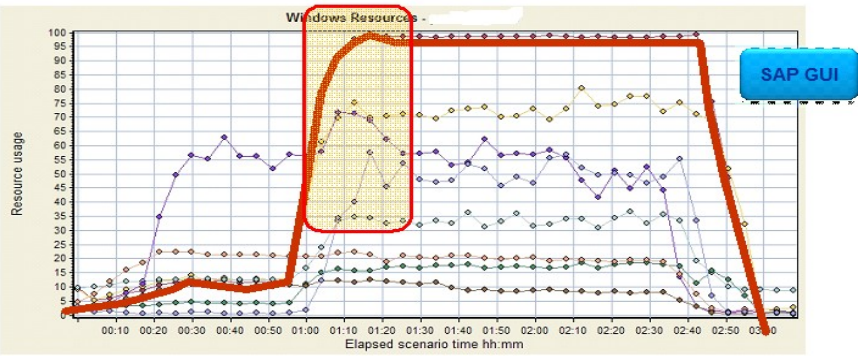


Figure 11: Central Instance Windows Resources

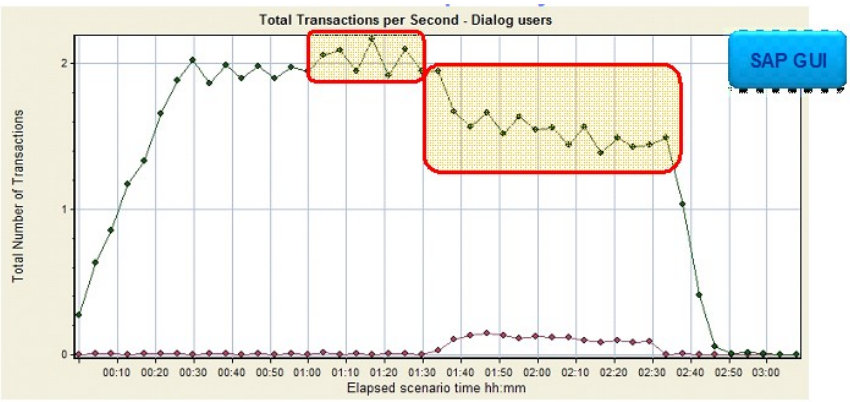


Figure 12: Transactional Frequency

Scoping CRM Scenarios

Two scenarios were identified

- Search with employee code and create a ticket
- Search with employee name and create a ticket
- 70 concurrent users over LAN and 30 users over WAN.

Execution Results

Scenario	Users	Tickets / hour	Test duration / mins	SAP steps	CPU
Test 1	70 LAN 30 WAN	535	100	14000	'normal'
Test 2	70 LAN 30 WAN	594	50	8600	'normal'

Table 1: CRM Results
During this test a specific parameter was tuned to increase the RFC calls which increased the tickets creation/hour.
Parameter Name : 'rdisp/tm_max_no' value can be increased based on desired results.

Scoping BIW Scenarios

-

The objective of this test was to compare the response time of different BIW queries with and without BIA for 40 concurrent users. The Characteristics of the test scenario were as follows:

- 5 query scripts were created that accessed different cubes in order to test different types of queries. Most queries had 2 drilldowns
- 40 users were ramped up with 5 users at a time – the 5 users ran for 10 minutes before the next were added. Each of the 5 users executed one of the 5 queries
- Tests were made using HTTP and not HTTPS.
- Caching: OLAP Caching on the server was off
- Caching: Workstation caching was disabled for the LoadRunner scripts

Execution Results

In order to be sure that the LoadRunner response times were representative of the relative end-user response time we conducted a 'single user test' where we compared:

BW	BW with BiAccelerator
QTP test	QTP test
Manual test	Manual Test
LR test	LR test

Table 2: BW execution strategy

Test Results – Single User Test

Summary Table for HR Dashboard Query		
	BW	BW with BiA
QTP	39.7s	4.8s
Manual	Approx: 35s	Approx 1-2 s
LR	36.7s	1.5s

Table 3: BW single user results

The results show that:

- QTP adds a few seconds to its time calculation
- The results are largely in line between the different methods
- Execution time for the scripts remained more or less constant across iterations

The results validated the test approach for using LoadRunner with no cache for executing the tests.

Test Results : BW with/without BIA comparison

The test scenario was executed in an identical manner with and without the BI Accelerator.
The results showed a significant decrease in response times with the BI Accelerator switched on.

Key results are:

Summary Results comparing BW with and without BiA	BW	BW with BiA
Response times		Queries were x times faster
CPU	Approx: 35%	Average 42%
Response time variation with number of users	Response time was erratic – in general was 5-9 times slower with 40 users compared to 5 users	No significant change in response time
Transactions per second	Ramp up correlating with number of users – peaked at 4.5 tps	Ramp-up correlating with number of users. Peaked at 22 tps and remained consistent.
Total number of transactions		4-7 times more transactions created at peak compared to without BiA (depending on query)

Table 4: BW Comparative Results

The following chart shows the acceleration effect per script as the number of users increased. We have selected only the BW actions (not the logins and log offs) for comparison.

For some actions the effect of the accelerator was in the range of 4x to 13x faster. For others, the acceleration effect was 150x – 547x faster.

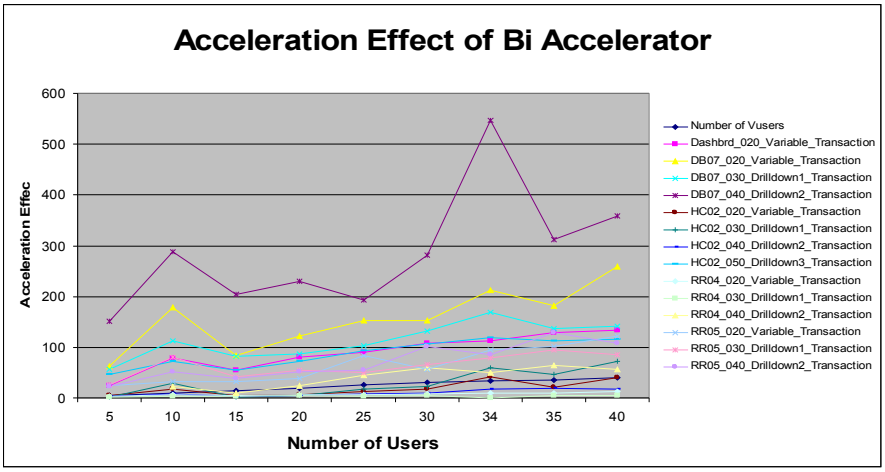


Figure 13: Acceleration of BIA

8. Summary

In this paper we gave an overview of testing various SAP systems which are robust and complex. After over viewing best practices implemented in SAP volume testing, we took up a case study to describe how we implemented those practices to find bottlenecks and various tuning options in transaction processing and query processing. We also briefly surveyed the monitoring part of SAP with LoadRunner and through executing SAP transactions.

9. Reference Materials

- **www.sdn.sap.com**
- www.support.mercury.com

10. Author's Biography

Anshuman Das is working with IBM since 2004. He has over 6 years of experience in IT and Testing. Currently, Anshuman is working as a SAP Performance Testing consultant. He is involved in consulting and creating performance testing strategies to ensure application performance enhancement. He has vast experience in domains like container and terminal management, logistics and ERP. He is CSTE certified from QAI.

11. Appendix

SR : Service Request. Every project request in testing is tracked through a unique id.
TAT: Total Turn Around Time
IS : Information Services
QA : This is used in context with environment which is a test or Quality Assurance environment.
PO : Purchase Order.
TPS: Transactions Per Second
BW/BIW: Business Intelligence Warehouse
BIA: Business Intelligence Accelerator is a separate analysis platform, including both software and hardware, which is network attached to the SAP NetWeaver BI server via gigabit Ethernet link.