



SAP White Paper
SAP Consulting – Process and Solution Quality

Testing SAP HANA

Sven Friess

Test Management Consultant
SAP Deutschland AG & Co. KG
E: sven.friess@sap.com

Sebastian Geissler

Test Management Consultant
SAP Deutschland AG & Co. KG
E: sebastian.geissler@sap.com

Version 1.0

April 2012



The Best-Run Businesses Run SAP™

© Copyright 2012 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company.

Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase, Inc. Sybase is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

The information in this document is proprietary to SAP. No part of this document may be reproduced, copied, or transmitted in any form or for any purpose without the express prior written permission of SAP AG.

This document is a preliminary version and not subject to your license agreement or any other agreement with SAP. This document contains only intended strategies, developments, and functionalities of the SAP® product and is not intended to be binding upon SAP to any particular course of business, product strategy, and/or development. Please note that this document is subject to change and may be changed by SAP at any time without notice.

SAP assumes no responsibility for errors or omissions in this document. SAP does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. This document is provided without a warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. This limitation shall not apply in cases of intent or gross negligence.

The statutory liability for personal injury and defective products is not affected. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third-party Web pages nor provide any warranty whatsoever relating to third-party Web pages.

SAP "How-to" Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

Disclaimer

Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressly prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP's Support Services and may not be modified or altered in any way.

Document History

Document Version	Description
1.00	First official release of this White Paper



The Best-Run Businesses Run SAP™

Typographic Conventions

Type Style	Description
<i>Example Text</i>	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation
Example text	Emphasized words or phrases in body text, graphic titles, and table titles
Example text	File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
Example text	User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
EXAMPLE TEXT	Keys on the keyboard, for example, F2 or ENTER.

Icons





Icon	Description
	Caution
	Note or Important
	Example
	Recommendation or Tip

Table of Contents

1.	Introduction.....	1
2.	SAP HANA Motivation for Testing	1
3.	SAP HANA Testing Approach.....	2
3.1	Phase 1: Technical Tests of SAP HANA.....	3
3.1.1	Design & Execute manual Test Cases.....	4
3.1.2	Build a Test Framework.....	4
3.2	Phase 2: End-to-End Functional Tests	7
3.2.1	Definition and manual Execution of Test Cases	7
3.2.2	Automation of Test Cases.....	8
3.3	Phase 3: End-to-End Performance Tests	9
3.3.1	Test Preparation and Mass Data Generation.....	9
3.3.2	E2E Performance Test using SAP LoadRunner by HP	10
3.3.3	Detailed Performance Analysis of SAP HANA DB/ NewDB	10
4.	Conclusions	13
5.	Appendix	14
	Official Documentation on HANA.....	14
	Security.....	14

List of Figures

Figure 1: Triggers for System Changes.....	1
Figure 2: Three-Phase Process Model for SAP HANA Testing.....	3
Figure 3: Building Blocks forming a functional Test Framework	5
Figure 4: Sample for Automated Test case	7
Figure 5: Components of manual Testing Approach.....	8
Figure 6: Measurement of Processing Time from SQL Traces	11
Figure 7: Sample SAP HANA DB/ NewDB result performance analysis.....	12

List of Tables

Table 1: Key Components of a Functional Test Framework	6
Table 2: Expensive SQL Statements	11

1. Introduction

SAP High-Performance Analytic Appliance (HANA) is a data warehouse appliance for processing high volumes of operational and transactional data in real-time. As such, SAP HANA will be the new leading and central SAP infrastructure component connected to other SAP applications.

This critical infrastructure component requires a professional and elaborated testing approach addressing technical, functional and non-functional needs. The goal is to ensure a smooth introduction and operation of SAP HANA in daily business at minimal costs.

This White-Paper delivers SAP Best Practices with respect to testing SAP HANA and outlines how SAP Consulting can assist SAP Customers in developing and implementing a tailored testing approach by taking those SAP Best Practices into account.

We would like to acknowledge the contribution of our interview partners from SAP, whose input, ideas and support were the foundation for this document.

2. SAP HANA Motivation for Testing

As outlined in Figure 1, three key drivers for change events, such as business, compliance or technology drivers, cause ongoing and discrete changes on SAP HANA. Reasons for change events could be:

- new implementations (e.g. the first introduction of SAP HANA)
- new processes
- consolidation activities,
- upgrades
- custom developments
- implementation of Support Packages, Patches & Builds

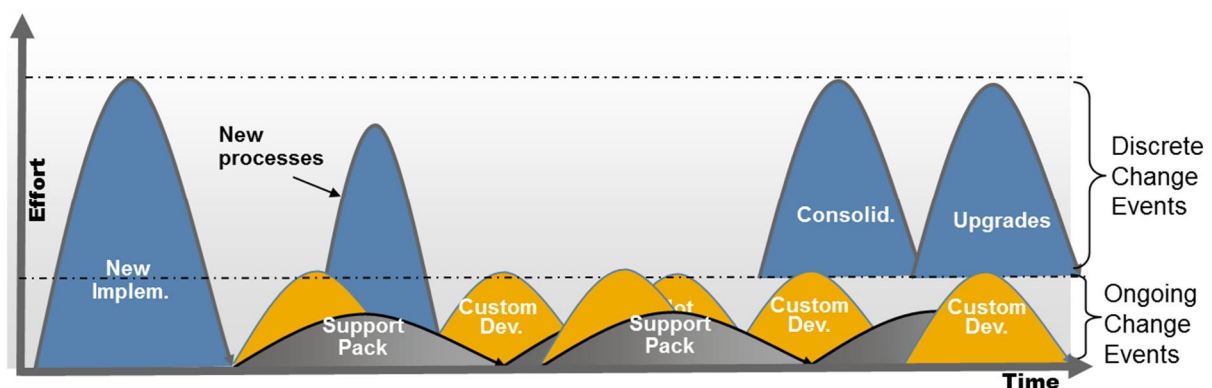


Figure 1: Triggers for System Changes

Taking into account, that SAP HANA is a central infrastructure component, which faced dozens of Support Packages, Patches & Builds within the last two years, and is expected to receive more updates with the same frequency on a mid-term perspective, there is a strong need to ensure testing of SAP HANA after each change.

Testing SAP HANA in this context has the following key challenges:

- Complexity - SAP HANA (HANA DB/NewDB, HANA Server & corresponding user clients) is a complex construct of different engines working in concert
- Change - SAP HANA (HANA DB/NewDB, HANA Server & corresponding user clients) requires to be tested after any change without the need of having connected all application servers
- Reliability - SAP HANA is a central infrastructure component, which serves as basis for critical business processes
- Expectation Management - due to many different stakeholders

Testing all different SAP HANA scenarios including all user clients and application servers in a manual way is not possible. Therefore, an elaborated testing approach including a selection of representative test cases and automation is necessary.

3. SAP HANA Testing Approach

Having understood the challenges and demands for testing SAP HANA outlined in the chapter above, there is a strong need for a professional and elaborated testing approach. SAP proposes three different phases of testing SAP HANA, which can be established either isolated or independent of each other, or in a combination of an elaborated three-phase-testing-approach. This high sophisticated approach is outlined in Figure 2 below and contains the following phases:

- Phase 1: Technical Tests of SAP HANA
- Phase 2: End-to-End Functional Tests
- Phase 3: End-to-End Performance Tests

Phase 1: Technical Tests of SAP HANA focus on technical and functional aspects of SAP HANA DB/NewDB in an isolated manner. Main goal of phase 1 is to enable the owner of SAP HANA DB/NewDB to ensure technical and functional correctness after any technical change.

Phase 2: End-to-End Functional Tests ensure functional correctness of the SAP HANA Server, user clients and Applications Servers, mainly focusing on their interaction.

Phase 3: End-to-End Performance Testing complements phase 2 by not only focusing on the functional correctness, but also by addressing non-functional aspects.

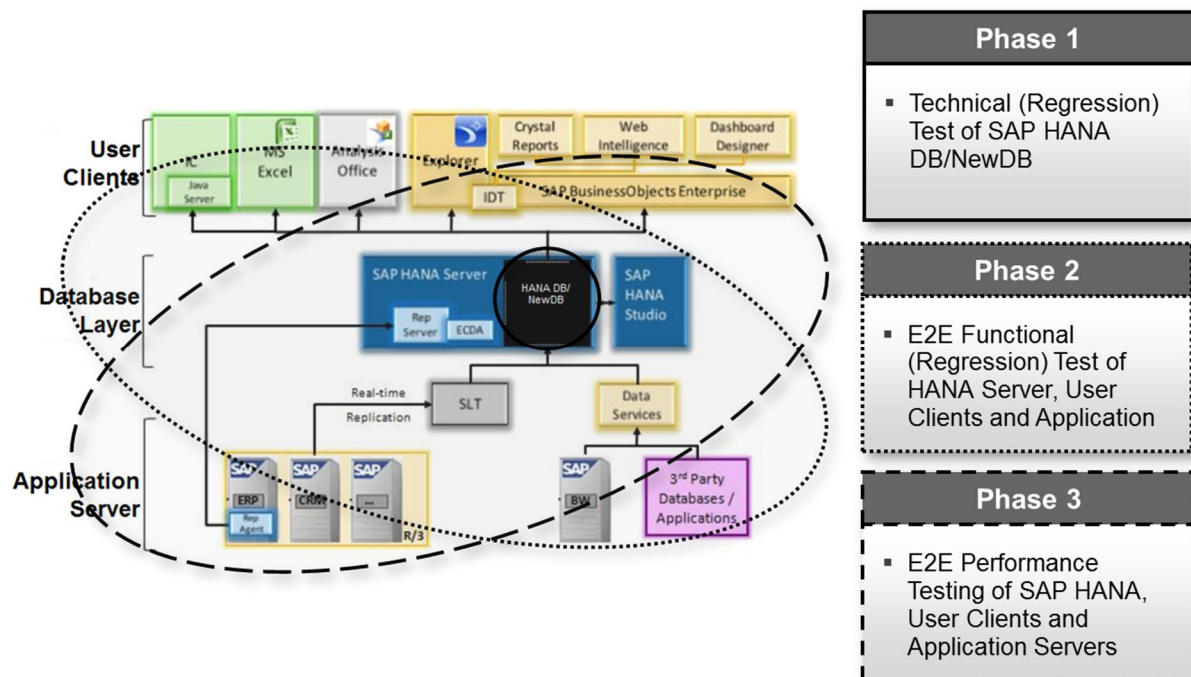


Figure 2: Three-Phase Process Model for SAP HANA Testing

Please note that Consulting Services and Consulting Solutions are provided by SAP for all of the different phases. These Consulting Services and Solutions provide a comprehensive methodology orchestrating the three-phase-testing-approach.

3.1 Phase 1: Technical Tests of SAP HANA

The first phase of the three-phase-testing-approach outlined above, addresses SAP HANA DB/NewDB only. From a SAP perspective, SAP HANA DB/NewDB needs to be tested after each technical upgrade in an isolated manner without the need of having all user clients and application servers connected to it. Stakeholders representing the business expect SAP HANA DB/NewDB to be working correctly after each technical change and are not willing to invest into testing; especially if the technical change was not requested or initiated by them. Therefore, the owner of SAP HANA DB/NewDB needs to ensure quality and gain confidence technical and functional wise in SAP HANA DB/NewDB after each and every change in a standalone mode.

Using the fact, that HANA DB/NewDB is equipped with technical bindings for Python Scripting, Java, JDBC, ODBC and other methods connecting to databases, SAP recommends a two-step-approach to reach Phase 1: Technical Tests of SAP HANA.

3.1.1 Design & Execute manual Test Cases

The first step is to design and manually execute test cases for all technical layer- parts of SAP HANA DB/NewDB. The ultimate principle to make sure, that all parts of SAP HANA DB/NewDB are covered, is to measure test coverage by means of code coverage. As such, a full blown approach is very difficult to realize. SAP recommends including a minimum set of aspects in order to maximize code coverage. These should cover tests for:

- row store
- column store
- join functionality
- OLAP functionality
- calculation engine
- planning engine
- SQL optimizer
- persistence
- table merge
- table split

3.1.2 Build a Test Framework

The second step within phase 1 is building a test framework, which will allow automation of the test cases. From a conceptual point of view, we recommend to establish the following building blocks as outlined in Figure 3:

- Building Block 1: Definition & Management
- Building Block 2: Measurements & Results

The Definition & Management comprises SAP HANA Models (Views, Packages, Model Size, etc.), SAP HANA Queries (SELECT Statements etc.), Test Features (Connection Manager, Expected Results Display Function, etc.) and Tests (Combination of Models and Queries). Building Block 2: Measurements & Results contains Sessions (Test Runs) and Results (e.g. Row Count).

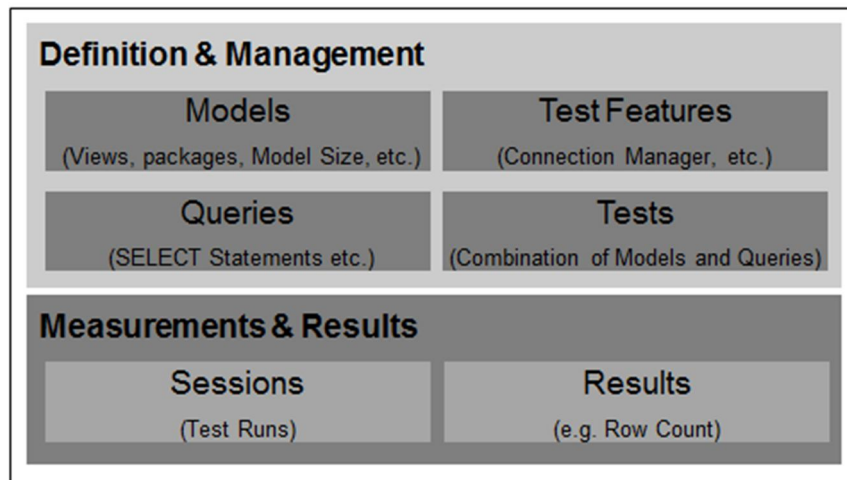


Figure 3: Building Blocks forming a functional Test Framework

A test framework based on the building blocks outlined above, enables a separation of Definition & Management data, as well as the separation of Measurement & Result data. In addition, it allows a separation of models and queries, which enables reuse, reduces maintenance efforts and guarantees comparable results over several test executions.

From a technical point of view, SAP recommends building a test framework and making use of technical bindings for e.g. Python. Such a test framework can be used to automate SAP HANA/NewDB test cases. According to SAP Best Practices for such a test framework, implementing the test features for the building block Definition & Management listed in the following table are key:

Feature	Example
Connection Manager	<pre> conn_mgr = ConnectionManager() # create unnamed connections c = conn_mgr.createConnection() # you can overwrite default connection-parameters # (see testscripts/config/testConfig.ini for the list of available parameters) c = conn_mgr.createConnection(autocommit=True) c = conn_mgr.createConnection(autocommit=True, port=4242) # create named connections c2 = conn_mgr.createNamedConnection("conn2") # createNamedConnection can overwrite default-connection parameters as well c = conn_mgr.createConnection("conn2", autocommit=True, port=4242) # close connection conn_mgr.closeConnection(c) # obtain named connection c2 = conn_mgr.getConnection("conn2") # close named connection conn_mgr.closeNamedConnection("conn2") # close all connections (created by create[Named]Connection) conn_mgr.closeAllConnections() </pre>
Expected Results Display Function	<pre> class MyTestCase(SqlTestCase): def test1(self): c = self.conn cur = c.cursor() cur.execute("select * from tab01") e = ExpectedSimpleResultSet(cur) print e </pre>
Printing Result Sets Function	<pre> { 'column_infos': # meaning of tuples: (name, type_code) [('X', 4), ('Y', 7), ('Z', 29), ('A', 4)], 'resultset': [[1, 10.0, 'asdf blafisch kdkdkd', 100], [2, 20.0, 'asdf kdkdkd kdkdkd', 100], [3, 30.0, 'asdf dkdkdkd kdkdkd', 200], [4, 40.0, 'asdf asdkd kdkdkd', 200], [5, 50.0, 'asdf test12334 kdkdkd', 300]] } </pre>
Test Suite Builder	<pre> from lib.suiteUtilities import TestSuiteBuilder tsb = TestSuiteBuilder() tsb.add("testBasicJoin") tsb.add("testBaseFunctionality", configurations=["configColumnStore", "configRowStore"]) tsb.add("testBaseFunctionality", "test2", "test2", "test2") if __name__ == '__main__': tsb.run() </pre>

Table 1: Key Components of a Functional Test Framework

In addition, there might be need to implement command line parameters for test cases and test suites.

After having implemented the test framework with its different building blocks and test features, test cases defined within the first step can be automated. An example for an automated test case could look like indicated in Figure 4:

```
import unittest

class MyTest(unittest.TestCase):

    def testArithmetic(self):
        self.assertEqual(2+2, 4)

if __name__ == '__main__': unittest.main()
```

Figure 4: Sample for Automated Test case

3.2 Phase 2: End-to-End Functional Tests

A key challenge with SAP HANA is the fact, that the SAP HANA Server, user clients and application servers are a complex construct of different engines that work in concert. Therefore, SAP HANA Server, corresponding user clients and applications servers need to be tested after each technical upgrade of any of those entities. In order to do so in an efficient manner, we propose the steps outlined in the subsequent chapters.

3.2.1 Definition and manual Execution of Test Cases

A manual testing approach for SAP HANA Server, user clients and application servers requires design and execution of test cases. The design of relevant test cases comprises the following activities:

- Define naming conventions
- Identify test objects
- Set priority and severity of test objects
- Create test case descriptions

A manual testing approach should comprise test cases for all aspects listed in Figure 5 below. Especially for the test of user clients and application servers, only those tests for user clients and applications servers, which are in use, are applicable.

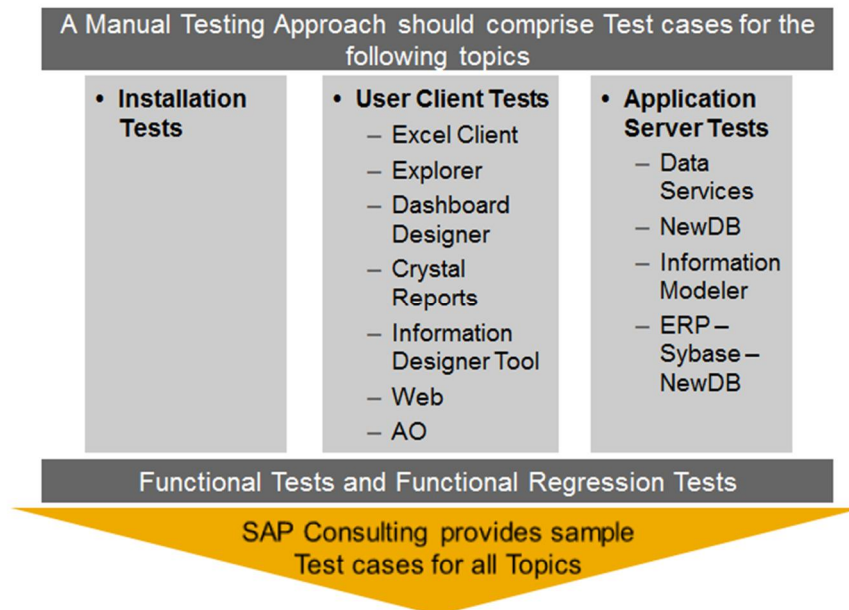


Figure 5: Components of manual Testing Approach

Sample test case descriptions for all the above mentioned topics are available as consulting solutions of SAP.

3.2.2 Automation of Test Cases

After having designed test cases for SAP HANA, user clients and application servers as well as at least having executed them once successfully, the test cases are ready to be automated for regression test purposes.

In order to do so, SAP recommends to use SAP Quick Test Professional by HP for the following reasons:

- Quick Test Professional is the state of the art test automation tool
- Quick Test Professional supports nearly all user clients and application servers
- Quick Test Professional is part of SAP Testing Options 1 & 2 and therefore integrates with SAP Solution Manager and SAP Quality Center by HP
- Quick Test Professional is already licensed at many SAP customers
- Quick Test Professional is an easy to use record & replay test automation tool

The test automation approach using SAP Quick Test Professional by HP is similar to standard test automation approaches for ERP-Systems. This means from a customer perspective: there is already a high degree of test automation knowledge available, which can easily be reused for test automation of SAP HANA, user clients and application servers.

3.3 Phase 3: End-to-End Performance Tests

With SAP HANA extremely fast data analysis with scan times of one billion rows per second for fact tables, and a join performance of ten million rows per second, when joining attribute tables to fact tables, are expected by customers. With this expectation, high query performance is one of the key features of SAP HANA based applications.

In order to ensure a high E2E performance of SAP HANA, user clients and application servers, a proper E2E Performance Testing approach is required. It fully unleashes the entire SAP HANA performance and the associated technology stack. In particular, the utilization of customer individual extensions, modifications, different customizing and data models, and Z-Transactions fortifies comprehensive activities to assure performance stability.

Based on SAP Best Practices, an E2E performance test of SAP HANA Server, user clients and application servers should comprise at least a three-step-procedure as outlined in the following chapters to measure and optimize standard KPI's. Standard KPI's addressing the goal of reducing the Total Cost of Ownership (TCO) are:

- End-to-end execution time including specific known queries
- Resource consumption of end-to-end scenarios including long running queries
- CPU consumption
- Memory consumption
- Degree of parallelization
- Throughput
- Volume restrictions

SAP offers support for all steps of the E2E performance testing approach either on-site, or, if desired, in a test factory approach off-/near shore.

3.3.1 Test Preparation and Mass Data Generation

The generation of Mass Data and its proportionate distribution according the underlying data models and queries is a key success factor for any Performance Test. Therefore, planning and provisioning of test data is an essential and fundamental preparation step. Data structures and volumes have to be aligned from a content perspective and have to fit into the data models of the testing scope. The test cases must not only be fully functional but also repeatable. This requires a sufficient amount of data.

According to SAP Best Practices, an option for data generation is the HANA JavaScript Tool, which is a Java based engine for the execution of JavaScript files for data processing in SAP HANA DB/NewDB. Another option would be to use a tool based on Eclipse with JavaScript support (JEE or JavaScript versions). With this second option, HANA Java classes are exposed as JavaScript classes for script developers. As several generators for sequences, numbers, strings, dates, UUIDs are provided, both tools can be used for in a mass data generation scenario.

3.3.2 E2E Performance Test using SAP LoadRunner by HP

In order to analyze the E2E Performance an expected average/peak load profile has to be simulated. This average/peak load profile needs to reflect users, background jobs and various other workloads. SAP recommends using SAP LoadRunner by HP for simulating this average/peak loads for the following reasons:

- Easy scripting using SAP LoadRunner by HP as SAP HANA is capsulated by either user clients or SAP applications (e.g. Business Objects/SAP BI/SAP ERP)
- For all SAP applications, there is the required scripting know-how available
- Scripts can be easily modified and parameterized
- Integration of monitors from the environment under test so that most of the critical resources are correlated to the response times
- Stable behavior in long-running tests (no crashes in the middle of a run)
- Not only the number of data but also the number of users executing queries in different manners is critical to the performance of SAP HANA. Simulating a high number of users is one of the key strengths of SAP LoadRunner by HP
- Monitoring and Analysis feature included in SAP LoadRunner by HP
- Ability to monitor multiple KPI's

3.3.3 Detailed Performance Analysis of SAP HANA DB/ NewDB

After having executed an E2E Performance Test using SAP LoadRunner by HP, KPI's have been collected by using the monitoring features of SAP LoadRunner by HP or standard transactions for analysis such as STAD or ST5. If KPI's measured during the E2E Performance Test using SAP LoadRunner by HP have not been met, a detailed performance analysis of SAP HANA DB/ NewDB is required as part of the root-cause analysis. The first step of such a root-cause analysis is e.g. to understand the processing time spent in the SAP HANA DB/NewDB, when executing a query.

Measuring the overall processing time of an SQL statement on SAP HANA DB/ NewDB

This can be achieved by either investigating on:

- Expensive SQL statements using the corresponding view within SAP HANA Studio
- SQL traces via the Diagnostics feature with SAP HANA Studio

Using the expensive SQL statement view within SAP HANA Studio the insights as listed in Table 2 can be gained:



START_TIME	Time when the query started
DURATION_MICROSEC	Runtime of the query in micro seconds
OBJECT_NAME	Name of the object(s) accessed
RECORDS	No. of records returned  Please note that the default fetch size is 32 records
STATEMENT_STRING	The SQL statement sent to HANA  Double-clicking on the statement shows all details

Table 2: Expensive SQL Statements

Using SQL traces will provide certain time stamps (e.g. start and stop of the query) as indicated by Figure 6 below. By calculating the difference between stop and start time, the processing time of the SQL statement on SAP HANA Server can be measured and calculated.

```
# tracing PrepareStatement execute call (thread 15099) at 2011-09-27 11:42:39.204239
cursor 139932745875456 c12.execute('' SELECT "MANDT" , "CARRID" , "CONNID" , "FLDATE" , "BOOKID" , "CUSTOMID" , "CUSTTYPE" , "SMOKER
# tracing finished PrepareStatement execute (thread 15099) at 2011-09-27 11:42:39.352294
# tracing ResultSet.columnLabel = [MANDT:NVARCHAR, CARRID:NVARCHAR, CONNID:NVARCHAR, FLDATE:NVARCHAR, BOOKID:NVARCHAR, CUSTOMID:NVARC
## tracing ResultSet.row[1] = [001, AA, 0017, 20100407, 00002220, 00003815, P, , 45.9, KG, X, F, 1989.66, CAD, 1268.82, USD, 20091216
# tracing ResultSet.row[2] = [001, AA, 0017, 20100407, 00002221, 00003578, P, , 45.5, KG, X, F, 1268.82, USD, 1268.82, USD, 20091230,
```

Figure 6: Measurement of Processing Time from SQL Traces

Analyzing SQL statements using Explain Plan

The EXPLAIN PLAN statement is used to evaluate the execution plan that is followed by SAP HANA DB/NewDB in order to execute an SQL statement. The result of the evaluation is stored into the EXPLAIN_PLAN_TABLE view for later user examination. Exactly this result can be used in order to understand the different execution steps and their influence on the processing time of an SQL statement on SAP HANA DB/NewDB.

Analyzing SQL execution with the profile trace

The profile trace is one of the most detailed trace levels that is available in the SAP HANA Studio. The profile trace is particularly helpful to understand how the various engines inside HANA work together. E.g. the number of records that are passed from one engine to another can be examined. Using the profile trace will help to understand the SQL statement and its execution procedure, which might enable to understand the root cause of a very high processing time.

Capturing and analyzing performance traces for Attribute, Analytic & Calculation Views

The detailed analysis of SQL statements as described above helps to analyze pure SQL statements that are executed by the SQL engine of SAP HANA DB/NewDB. However, all models created with the SAP HANA Studio, i.e. Attribute Views, Analytic Views and Calculation Views, will be executed in the OLAP, JOIN or Calculation Engine of SAP HANA rather than the SQL Engine. Thus these models require a performance trace to analyze their runtime behavior in detail.

In order to capture performance traces the following options are available:

- Capturing a performance trace from the HANA Studio
- Capturing a performance trace with the HANA DB admin tool

After having captured a performance trace an appropriate analysis is required. In order to get the most out of such an analysis, we recommend executing the following steps:

- Identify performance relevant SQL queries in the HANA DB admin tool
 - Select specific known queries
 - Select the top time consuming queries
- Understand which engine executes which plan operations
- Understand the degree of parallel execution
- Drill down into single plan operations

Measuring query performance using a Test Framework (see chapter 3.1)

If a test framework as described in chapter 3.1 has been set up, it is also possible to use this test framework in combination with the detailed analysis steps described above for measuring SAP HANA DB/NewDB query performance in an easy manner. This has been done SAP internally, and enabled SAP to have a detailed analysis of the Calculation Engine, the Join Engine and OLAP Engine.

Figure 7 below summarizes key findings of a sample SAP HANA DB/NewDB performance analysis.

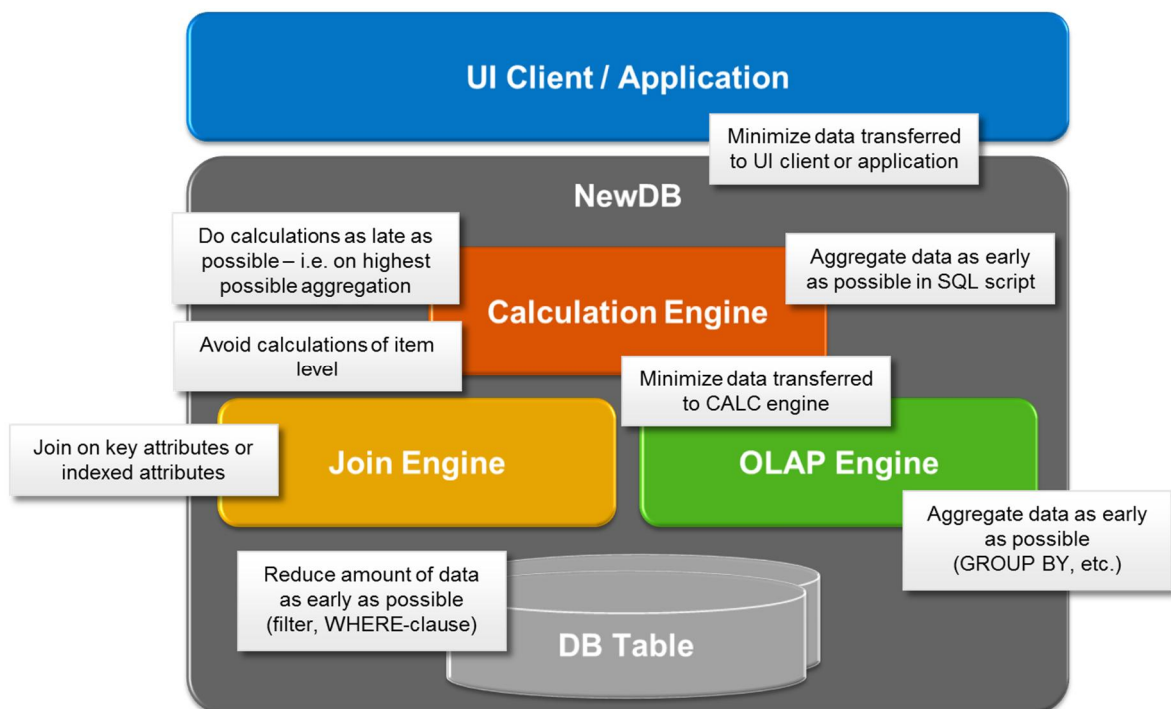


Figure 7: Sample SAP HANA DB/ NewDB result performance analysis

In general, findings of a detailed SAP HANA performance analysis may include recommendations for:

- Data transfer between different layers within the Technology Stack

- Aggregation rules for mass data processing
- Indexing and primary key determination and mechanisms
- Calculation and data processing routines

and therefore can serve as initial input customizing guidelines, when defining customer specific models and queries.

4. Conclusions

With the SAP HANA environment representing a complex infrastructure, intense and severe testing is a must and a key success factor running this multipart framework. The test approach described in this document points out a three-phase integrative testing approach incorporating multiple requirements of testing and quality assurance. By following this three-step testing approach a smooth introduction and operation of SAP HANA at minimal costs can be ensured.

SAP will assist its customers with all testing activities by means of scalable and reliable Consulting Services and Consulting Solutions. SAP will assist its customers as Trusted Advisor with the goal of safeguarding the introduction and operation of SAP HANA.

5. Appendix

The following contains multiple Links to SAP HANA documentation.

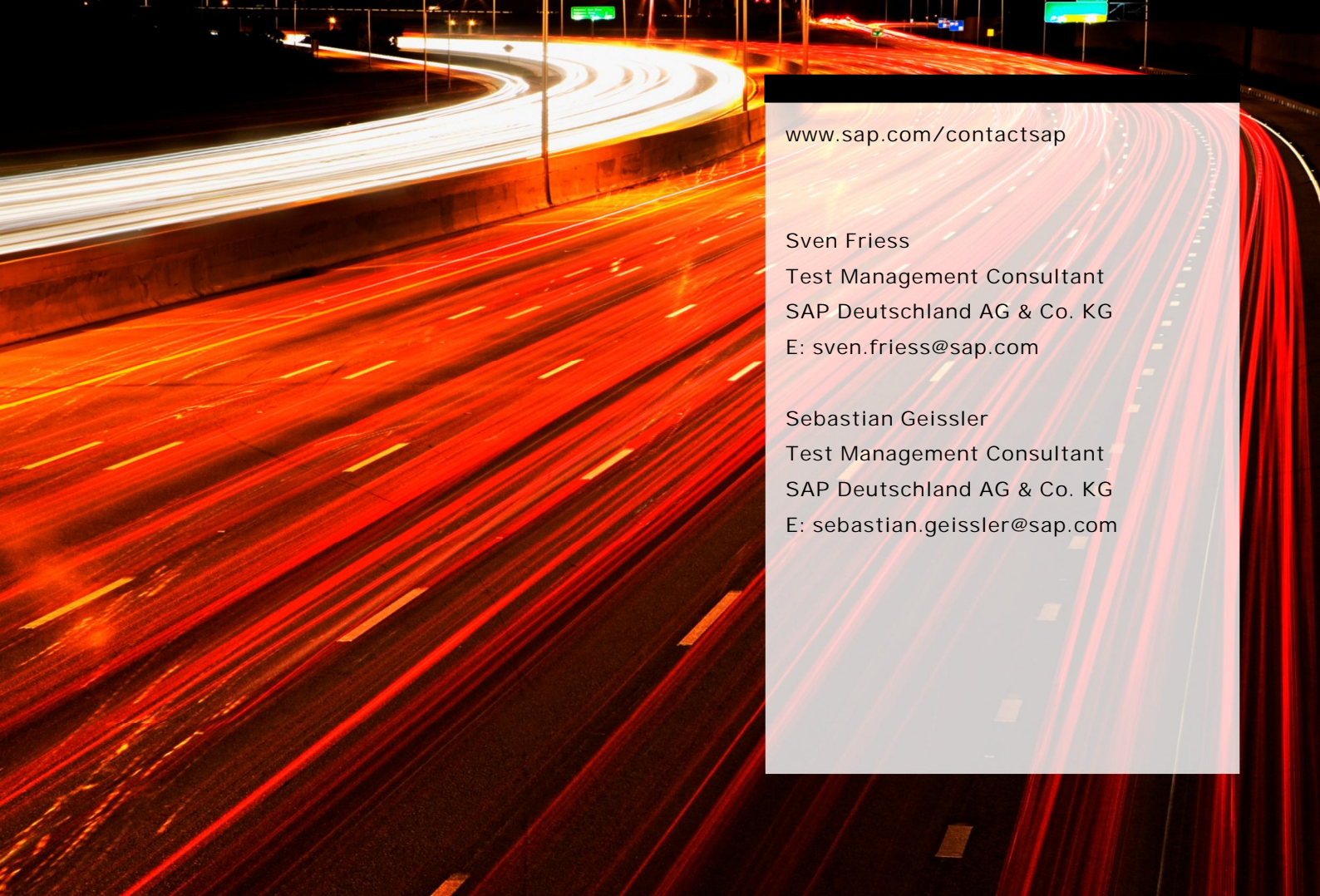
Official Documentation on HANA

SAP Help Portal: http://help.sap.com/content/hana/overview/hana_overview.htm

Service Market Place: <https://service.sap.com/rds-hana-erp>

Security

Security guide: http://help.sap.com/hana/hana1_imdb_sec_en.pdf



www.sap.com/contactsap

Sven Friess

Test Management Consultant

SAP Deutschland AG & Co. KG

E: sven.friess@sap.com

Sebastian Geissler

Test Management Consultant

SAP Deutschland AG & Co. KG

E: sebastian.geissler@sap.com



The Best-Run Businesses Run SAP™