

IBM SAN Distance Matrix Project

Trace coverage and modeling across IBM test labs world-wide

Yoram Adler
IBM Research
Haifa, Israel
adler@il.ibm.com

Tara Astigarraga
IBM Corporate Head
Quarters
Rochester, United
States
asti@us.ibm.com

Sheri Jackson
IBM Systems and
Technology Group
Tucson, United States
sheribj@us.ibm.com

Jose Roberto Mosqueda
Mejia
IBM Software Group
Guadalajara, Mexico
mosqueda@mx1.ibm.com

Orna Raz
IBM Research
Haifa, Israel
ornar@il.ibm.com

Abstract—Storage Area Networks (SAN) solutions are highly complex, often with enterprise class quality requirements. To perform end-to-end customer-like SAN testing, multiple complex interoperability test labs are necessary. One key factor in field quality is test coverage; in distributed test environments this requires a centralized view and coverage model across the different areas of test. We define centralized coverage models and apply our novel trace coverage technology to automatically populate these models. Early results indicate that we are able to create a centralized view of SAN coverage across the multitude of IBM test labs world-wide. Moreover, we are able to compare test lab coverage models with customer environments. Based on these views and comparisons, we expect to obtain an increased coverage, resulting in increased discovery rate of high-impact defects.

Keywords—Software Test; Software Engineering; SAN Test; System Test; Distance Matrix; Trace Coverage Models; SAN Hardware Test Coverage

I. INTRODUCTION AND MOTIVATION

IBM is a global technology and innovation company with more than 400,000 employees serving clients in 170 countries [1]. The IBM test structure consists of thousands of test engineers world-wide. In addition to function test teams for product streams, there is also an entire world-wide organization of many hundreds of people dedicated to systems and solution test. IBM has interop and complex test labs world-wide [2]. Systems test strategies focus on customer-like, end-to-end solution integration testing designed to cover the architectural design points of a broad range of customer environments and operations with the end goal of increased early discovery of high-impact defects, resulting in increased quality solutions. One key area of systems and solution test is innovation. As configurations supported continue to climb, with over 180 million configurations supported on the System Storage Interoperation Center (SSIC) site, test engineers are continually challenged to find ways to test smarter [3].

One IBM test transformational project we have been working on is the storage area network (SAN) distance matrix project. This project arose from the IBM Test and Research divisions as a joint-project aimed at better quantifying and understanding the systems test SAN coverage across IBM test groups world-wide.

At the start of this project we had lots of questions related to world-wide hardware and SAN coverage, but we did not have a centralized view of the test labs across IBM. Test labs were built, monitored and architected on an individual basis without the ability to easily extract coverage models across the test locations and understand on a global scale the total IBM coverage model. Another piece missing was the ability to do broad coverage reviews looking at IBM test labs in comparison to its clients. We have always worked hard to build our test environments to include key characteristics from a diverse range of IBM clients, however we did not have a data environment modeling tool to take customer environment variables and map them against our test environments. The IBM Distance Matrix project was designed to address these concerns and help to centralize visibility and configuration details about the systems and solution SAN test labs across IBM and its clients.

The SAN distance matrix project has the abilities to look at key architectural design points across the SAN environments and extract coverage summaries for deep-dive reviews, comparisons and ultimately architecture changes to continually improve our solution test coverage, scalability and customer focus. In this paper, we will further describe the SAN distance matrix project and the early results we have achieved.

II. RELATED WORK

There are existing tools, including Cisco Data Center Network Manager [4] and Brocade Network Advisor [5] that provide in-depth and detailed modeling capabilities for single environments or environments managed by a single entity; however there is a gap in the ability to easily look across a heterogeneous group of environments controlled by different companies, divisions or organizations.

In functional modeling and one of its optimization techniques Combinatorial Test Design (CTD), the system under test is modeled as a set of parameters, respective values, and restrictions on value combinations that may not appear together in a test. A test in this setting is a tuple in which every parameter gets a single value. A combinatorial algorithm is applied in order to come up with a test plan (a set of tests) that covers all required interactions between parameters. Kuhn, Wallace and Gallo [6] conducted an empirical study on the interactions that cause faults in software that is the basis for the rationale behind CTD. Nie

and Leung [7] provide a recent survey on CTD. The SAN distance matrix that we create can be viewed as a functional model. In our case, we automatically extract the model from switch dumps. We term the creation of coverage models from existing traces 'trace coverage'.

III. PROJECT STRATEGY

The project strategy is composed of 2 main phases as shown on Fig.1. The goal of the SAN distance matrix project is to extract data quarterly across team world-wide. By identifying key switch data, the script we execute has little impact to the regular activity of the switches. In the following sections, we describe each phase and activities in detail.

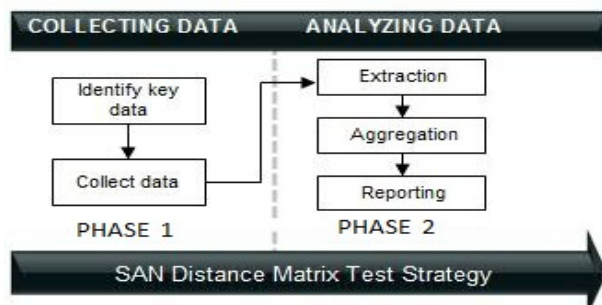


Figure 1. SAN Distance Matrix Project Strategy

IV. COLLECTING DATA

The data collection phase is composed of 2 main activities:

A. Identify Key Data

Using switch dump data, we selected specific switch query commands which are used to systematically extract the key data for usage and coverage statistics across different IBM test teams and select customers. The switch query commands allow us to extract dump data focused on topologies, coverage points, utilization and other environmental aspects in our SANs. Topology data points include port speeds, port counts and port types. Environmental data points include the switch hardware platforms, protocols used (ex: Fibre Channel, Fibre Channel over Ethernet, Fibre Channel over IP), code levels, switch up-time and switch special functions/features enabled.

Architectural design points include port-channel/trunk usage, vsan/vlan coverage, virtualization data and initiator/target to inter-switch link ratios. Using this dump data and subsequent processing logic, we were able to create a summary of all the different port speeds being tested, switch utilization rates, general architecture modeling and software and hardware versions being covered across the initial scope of IBM systems test and customer environments.

This approach helped us easily gather promising data, avoid limitations of manual investigation and create a model that is scalable and easy to use for ongoing analysis and trending.

B. Collect Data

For data collection, we designed automated scripts to collect the dump data from IBM test labs; the scripts use a source csv file which contains the list of switches, switch types, IPs and credentials. It uses a telnet connection to login the different switches, then executes the appropriate switch query commands and generates a log containing the switch dump data for each switch. For the initial scope of this project a subset of IBM test labs was chosen, that subset group included fourteen IBM test labs which contained a total of four hundred and eighty five SAN top of rack, edge and core switches.

V. ANALYZING THE DATA

The problem: SAN switch dump data is heterogeneous based on switch vendor, platform and code levels. Further, the data is collected from various sources and unique collection methods across IBM test laboratories and customer locations.

The switch dump data is a text file created for each switch. It contains output from multiple switch queries/commands that are executed against the switch. Each switch type has its own set of commands and a unique output format.

The goal: Parse the various switch dump semi structured data and transfer it to structured format.

The solution: the solution relies on the novel notion of trace coverage and the IBM EASER [8] easy log search tool.

Trace coverage extracts report data from traces that already exist in a system or are easy to create according to a defined coverage model. The coverage model can be code coverage – automatically created from the code locations that emit trace data, or functional coverage – manually created to define the system configuration or behavior. In SAN coverage, the traces are created by switch dumps, and the coverage model is a functional coverage of the possible SAN environments. A functional coverage model describes the test space in terms of variation points or attributes and their values. For example, attributes may be port types, ports rates, and ports utilization percentages. The IBM EASER tool supports extraction of semi-structured data from traces and its transformation into structured format. It provides both a graphical user interface (GUI) for interactive exploration and a headless mode of operation for automating the extraction and analysis process.

After defining a functional coverage model, the IBM EASER tool is used to extract, aggregate, and compare data:

- Extract functional model values from switch dumps.
- Aggregate the coverage of multiple logs from both customers and IBM test laboratories.
- Compare coverage between a defined set and subsets of labs by generating multiple summary reports.

The SAN Test functional coverage model is extendable; it can be updated to include additional values seen in customer environments. The collected data is aggregated by

IBM test groups and customers and definitions are flexible and can be supplied by the end-user.

The automated functional coverage analysis process includes three phases: Extraction, Aggregation and Reporting.

A. Extraction

The functional model attributes' values are extracted from each switch dump file. By using EASER, the log is divided into entries and then the relevant data is extracted, computed and inserted into the relevant model attributes' values. One file with attributes and values is created for each switch log file.

Fig. 2 shows a sample of a single cisco_fc switch log file, which is created using the automated scripts. In addition to the switch summary, the log file includes the switch query commands and corresponding switch data output.

```

2014-03-24 14:24:55 INFO Switch Summary
  Name:  sswc10f2cis
  IPAddr: 9.11.195.75
  Brand:  cisco
  Type:   fc
  Area:   cisco san
  Location: tucson
2014-03-24 14:24:55 INFO Log in to device sswc10f2cis.tuc.stglabs.ibm.com
2014-03-24 14:25:00 INFO Log in to sswc10f2cis.tuc.stglabs.ibm.com successful
2014-03-24 14:25:00 INFO -----
    
```

Figure 2. Switch Log File Sample

The EASER parser extracts values from the entry in Fig. 2 and updates them into the attributes shown in Fig 3.

Line	Area()	Location()	SwitchType()	SwitchNumber()	Command()	Name()	Value()
1	cisco_san	tucson	cisco_fc	1	Header	sswc10f2cis	##

Figure 3. Parser extracted data Sample

Fig. 4, shows a sample of a Cisco switch dump data extract, which the parser will use to compute values, then inserts the combined values into a model.

Interface	Vsan	Admin Mode	Admin Trunk Mode	Status	SFP	Oper Mode	Oper Speed (Gbps)	Port Channel
fc1/30	200	F	auto	up	sw1	F	16	--
fc1/31	200	auto	off	up	sw1	F	4	--
fc1/32	200	auto	off	notConnected	sw1	--	--	--
fc1/33	200	auto	off	up	sw1	F	16	--
fc1/34	200	auto	off	up	sw1	F	16	--
fc1/35	200	auto	off	up	sw1	F	16	--
fc1/36	200	auto	auto	notConnected	sw1	--	--	--
fc1/37	200	auto	off	up	sw1	F	16	--
fc1/38	200	auto	off	up	sw1	F	16	--
fc1/39	200	auto	off	up	sw1	F	8	--
fc1/40	200	auto	off	up	sw1	F	8	--
fc2/20	100	auto	auto	up	sw1	E	8	4
fc2/21	100	auto	auto	up	sw1	E	8	1
fc2/22	100	auto	auto	up	sw1	E	8	1
fc2/23	100	auto	auto	up	sw1	E	8	1
fc2/24	100	auto	auto	up	sw1	E	8	1
fc3/43	200	E	off	up	sw1	E	8	9
fc3/44	200	E	off	up	sw1	E	8	9

Interface	Vsan	Admin Trunk Mode	Status	Oper Mode	Oper Speed (Gbps)	IP Address
port-channel1	100	auto	up	E	64	--
port-channel2	200	auto	up	E	64	--
port-channel4	100	auto	up	E	32	--
port-channel8	200	off	up	E	8	--
port-channel9	200	off	up	E	16	--

Figure 4. Cisco MDS extract data snippet

Fig. 5 shows an example of an abbreviated model. For the sake of brevity, only a small portion of the parser extract and model data are shown in these figures.

Area	Location	SwitchType	Name	Value
test_team1	Tucson, AZ	cisco_fc	TotalSwitchWithDataCount	1
test_team1	Tucson, AZ	cisco_fc	TotalFcPortsCount	192
test_team1	Tucson, AZ	cisco_fc	TotalFcPortsLoggedIn	74
test_team1	Tucson, AZ	cisco_fc	PercentageFcPortsUtilized	38
test_team1	Tucson, AZ	cisco_fc	TotalVSANsCount	2
test_team1	Tucson, AZ	cisco_fc	AvgFcEPortRate	8
test_team1	Tucson, AZ	cisco_fc	HighestFcEPortRate	8
test_team1	Tucson, AZ	cisco_fc	AvgFcFPortRate	10
test_team1	Tucson, AZ	cisco_fc	HighestFcFPortRate	16
test_team1	Tucson, AZ	cisco_fc	FcFPortSpeedsUsed	(16,4,8)
test_team1	Tucson, AZ	cisco_fc	PortChannelUsage	yes
test_team1	Tucson, AZ	cisco_fc	LongestKernelUptime	41
test_team1	Tucson, AZ	cisco_fc	ShortestKernelUptime	41
test_team1	Tucson, AZ	cisco_fc	AssociatedSWHWVersions	6.2(7),cisco_MDS_9710

Figure 5. Cisco MDS single switch abbreviated base model.

B. Aggregation

All data from **Extraction** output files is grouped by switch type and switch locations into three files:

1. Summary of all entries,
2. Summary of all samples that contains “full data”
3. Summary of files with “no” or “partial” data.

The contents of the first two files reflect the model: Attributes and their aggregated values from the extraction phase output files. The third file contains an ‘illegal’ list that should be reviewed by IBM experts for the cause of the failure during collection. Fig. 6 contains a subset example.

Area	Location	SwitchType	Name	Value
Test-lab-i	Austin, TX	cisco_fc	#Switches	6
Test-lab-c	Tucson, AZ	cisco_fc	#Switches	26
Test-lab-n	Raleigh, NC	brocade_fc	#Switches	5
Test-lab-d	Tucson, AZ	brocade_fc	#Switches	13
Test-lab-o	China	cisco_fc	#Switches	4
Test-lab-b	Tucson, AZ	brocade_fc	#Switches	20
Client1	NY	cisco_fc	#Switches	14
Test-lab-i	Austin, TX	cisco_fc	PortCount	516
Test-lab-i	Austin, TX	brocade_fc	PortCount	112
Test-lab-c	Tucson, AZ	cisco_fc	PortCount	1394
Test-lab-n	Raleigh, NC	brocade_fc	PortCount	568
Test-lab-d	Tucson, AZ	brocade_fc	PortCount	496
Test-lab-o	China	cisco_fc	PortCount	340
Test-lab-b	Tucson, AZ	brocade_fc	PortCount	2380
Client1	NY	cisco_fc	PortCount	2213
Test-lab-i	Austin, TX	cisco_fc	VSANCount	6
Test-lab-c	Tucson, AZ	cisco_fc	VSANCount	22
Test-lab-n	Raleigh, NC	cisco_fc	VSANCount	6
Test-lab-d	Tucson, AZ	cisco_fc	VSANCount	5
Test-lab-o	China	cisco_fc	VSANCount	34
Client1	NY	cisco_fc	VSANCount	23
Test-lab-n	Raleigh, NC	cisco_fc	PortSpeedsUsed	(4,8)
Test-lab-i	Austin, TX	cisco_fc	PortSpeedsUsed	(2,4,8)
Test-lab-b	Tucson, AZ	brocade_fc	PortSpeedsUsed	(4,8,16)
Test-lab-c	Tucson, AZ	cisco_fc	PortSpeedsUsed	(4,8,10,16)
Client1	NY	cisco_fc	PortSpeedsUsed	(1,2,4,8,10)

Figure 6. Summary of select full data samples

C. Reporting

Data from the **Aggregation** phase is broken into several reports. There are two summary reports types: code levels and machine types which are based on aggregation summary of all entry files and results report which contains data including: switch functions, SAN design principles, switch utilization, port speeds, errors, peak traffic rates and average traffic rates.

Fig 7. contains an example of number of switches running select Cisco NX-OS code levels from two IBM test labs and one client location.

Cisco NX-OS	Test-lab-c	Test-lab-i	Client1
6.2.x	6	5	0
5.2.x	8	1	10
5.0.x	5	0	0
4.1.x	3	0	2
3.2.x	4	0	2

Figure 7. Code level sample report

VI. EARLY RESULTS

We have created a functional model that has enabled us to:

- Identify key SAN coverage and test variants
- Better understand our global SAN test environments using trace coverage analysis to aid in gap identification and improve our system test coverage strategies.
- Ensure test groups remain on IBM supported Cisco and Brocade switch code levels
- Create central list of SAN switch hardware across IBM test, allowing us to identify groups utilizing dated switch hardware and place them into a hardware refresh pool.
- Look at switch utilization and stress rates to ensure we are accurately stressing our equipment and if not put plans in place to help increase load coverage.
- Review environment architecture designs and recommend changes or complexity additions where appropriate.
- Foster technical interaction and deep dive environment cross-test-cell reviews with test technical leads from IBM test labs world-wide.

Overall, we were able to systematically collect data from global IBM System test labs and create a centralized view of SAN switch equipment and coverage across IBM systems test. We were also able to gather dump data from select customers and compare our test lab coverage models with customer environments.

VII. CONCLUSION AND FURTHER DEVELOPMENT

As solution complexity and the number of supported configurations increase in the IT industry, we must continue to re-invent the ways we do solution testing. In our global test environment, the need to have procedures in place to extract data and create advanced comparison and coverage

models is essential. This project, although in its early deployment stages, has already shown tremendous promise for being able to systematically extract and model coverage across a large number of test and client SAN environments. One of the key factors of this models continuing success is its scalability.

In the future, we plan to extend the distance function beyond reducing the data to a single dimension. For example, today one distance function is the difference in the average rates among different groups. We could instead compute a distance metric over the rate vectors.

As we continue to implement the distance matrix project across test labs within IBM we are gathering key data and making methodical changes is SAN test architecture to provide better test coverage points for IBM products and solutions.

REFERENCES

- [1] "IBM Basics," ibm.com [Online]. Available from: <http://www.ibm.com/ibm/responsibility/basics.shtml>. [Accessed: May 15, 2014].
- [2] T. Astigarraga, "IBM Test Overview and Best Practices" SoftNet 2012, Available from: http://www.iaaria.org/conferences2012/files/VALID12/IBM_Test_Tutorial_VALID2012.pdf [Accessed: May 15, 2014].
- [3] "IBM System Storage Interoperation Center (SSIC)," ibm.com [Online]. Available from: <http://www-03.ibm.com/systems/support/storage/ssic/interoperability.wss> [Accessed: May 15, 2014].
- [4] "Cisco DCNM Overview," cisco.com [Online]. Available from: http://www.cisco.com/c/en/us/td/docs/switches/datacenter/mds9000/sw/5_2/configuration/guides/fund/DCNM-SAN-LAN_5_2/DCNM_Fundamentals/fmfundov.html [Accessed: May 28, 2014].
- [5] "Brocade Network Advisor," brocade.com [Online]. Available from: <http://www.brocade.com/products/all/management-software/product-details/network-advisor/index.page> [Accessed: May 28, 2014]
- [6] Kuhn, D. Richard, Dolores R. Wallace, and Jr AM Gallo. "Software fault interactions and implications for software testing." Software Engineering, IEEE Transactions on 30.6 (2004): pp. 418-421
- [7] Changhai Nie and Hareton Leung. 2011. A survey of combinatorial testing. ACM Comput. Surv. 43, 2, Article 11 (February 2011)
- [8] Y. Adler, A. Aradi, Y. Magid, O. Raz, "IBM Log Analysis Tool (EASER)," unpublished.