

vAnalyser: vSphere Analysis and Recommendation Tool

Sachin D Joshi¹, Anisha B S², Rajavardhan Bhat³

M.Tech IT, ISE Department, RVCE, Bangalore¹

Assistant Professor, ISE Department, RVCE, Bangalore²

Member of Technical Staff, R&D, VMware Inc, Bangalore³

Abstract: In the current era of huge datacenters opting for Virtual infrastructure, vSphere provides a vast array of features which makes virtualization very effective and flexible. Major challenge ahead of vSphere administrators is to analyse the capabilities of each ESXi hosts in datacenter and to find out the server which is capable to run a specific vSphere feature. This paper proposes a utility 'vAnalyser' to scan the servers in vSphere datacenter and generate a 'feature to server' compatibility matrix. Datacenter Administrator can refer to this compatibility matrix and quickly identify the server(s) which fulfil the requirements to use a vSphere feature. Currently users do not have such a tool to get a consolidated report to analyse vSphere features usable on a server. For example, Administrators want to know the servers in datacenter on which they can deploy a Virtual Machine of size more than 2TB or list of servers in datacenter between which vMotion is possible. Report generated by vAnalyser answers these queries of feature compatibility requirements. This tool can even be used by vSphere testers to identify servers with desired capabilities to test a specific vSphere feature. The tool will use VMware vSphere Management SDK to trace through the datacenter and prepare the matrix. We show the approach in which data can be fetched from datacenters and presented using an elementary prototype.

Keywords: vSphere, ESX, vAnalyser, Recommendation tool

I. INTRODUCTION

According to latest Gartner Magic Quadrant Positions [1], "As of mid-2013, almost two-thirds of x86 architecture workloads have been virtualized on servers." And it positions VMware in leaders Quadrant for x86 Server Virtualization Infrastructure. Thus VMware and virtualization are becoming more essential for IT infrastructure, offering great stability and flexibility.

A. Problem Statement

With this scenario of dependency and the enormous size of datacenters getting virtualized using VMware vSphere, vSphere administrators are critical part of maintaining datacenters. vSphere provides number of features which enable administrators to get the maximum out of virtualized datacenters. But while maintaining the existing datacenters the administrators face the challenge of identifying the correct servers for a particular operation. Administrators have to go through a manual process of analysing the architecture of whole datacenter to pick the servers which can execute the feature in need.

B. Proposed Solution

In this paper we propose vAnalyser a datacenter analysis tool which can extensively scan through the number of servers in datacenter, and combine the analysis of capabilities of each server for recommendation.

- It generates a report which recognizes the vSphere features which can be handled by a standalone server.
- It will generate a matrix to elaborate inter compatibility of servers to execute vSphere features which will need more than one server. E.g. vMotion, FT, HBR.

The structure of this paper is as follows. In section 2, we elaborate how we analyse the hardware capabilities of each server and map/recommend it to vSphere features. In section 3, we showcase our prototype design to demonstrate that we can achieve the goals we proposed in the paper. In section 4, we provide various use cases with existing related work. Finally, we conclude in section 5 along with the future prospects.

II. vANALYSER AS A RECOMMENDATION TOOL

vAnalyser can be used as a datacenter analysis, planning and decision assistant tool in vSphere infrastructure. vAnalyser works using a collection of components using the methods of data analysis, data comparison and presentation. It can recommend you the most apt server(s) in datacenter based on the vSphere feature requirements. Key phases of the decision support system are as described below.

- **INPUT** the servers and the list of features on which the recommendation is needed in the User Interface generated by web services. User can input either the hostnames of several standalone servers or else he can only input the IP of Virtual Center hosting the datacenter. User will have to select the vSphere features using the checkboxes available in UI.
- **ANALYSE** the current hardware capabilities of each mentioned server with respect to mentioned features. Each server is scanned thorough and assessed for current state of hardware for each of the desired features. All the data collected will be compared to see if it meets the pre-defined criteria for the feature

execution on a standalone host (Pre-defined criteria as specified in vSphere Administrator guides [6]). Also data fetched is compared across hosts to check if they are compatible for inter-host features. All the comparison data is compiled into a dynamic report which categorically defines each host with capable features. Report will also contain a matrix of "servers to features", requiring multiple hosts.

- With the matrix/report available, now administrators can PLAN AND DECIDE the servers, and the approach to execute the desired vSphere feature. Report will recommend the servers which match the eligibility criteria of given vSphere feature.
- If user feels that he needs more filtering, he can RESTRUCTURE the dynamic report generated to add or remove servers/features and generate a more specific report again.

III. PROTOTYPE DESIGN

In this section we appraise our idea by portraying the elementary prototype of this approach. Our prototype uses vSphere Management API [2], Java web services and eclipse IDE for java programming [3]. Mapping between objects in our program and vCenter/ESX server objects are done with the help of Managed Object Browser (MOB).

A. Process Flow

The basic flow of information in our prototype is as follows:

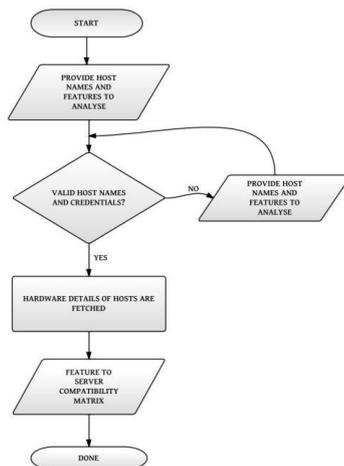


Fig. 1. Flowchart Explaining the Process Flow

B. System Design

- Administrators provide number of servers and their credentials to the system through User Interface (UI) generated by web services.
- Administrators also have to choose the vSphere features of their requirement.
- The server hostname/IP and its credentials are validated.
- Using VMware API, license information of each server is fetched. Then license information is validated for the desired feature availability.
- The details of the hardware elements on each server are collected, pertaining to the given feature.

- The state of hardware is verified and constraints for feature are validated.
- Administrators are presented with a consolidated matrix of servers and requested features.

The Service Instance managed object is the singleton root object of the inventory [4]. The host agent hierarchy has the same general form as the vCenter hierarchy, but most of the objects are limited to one instance.

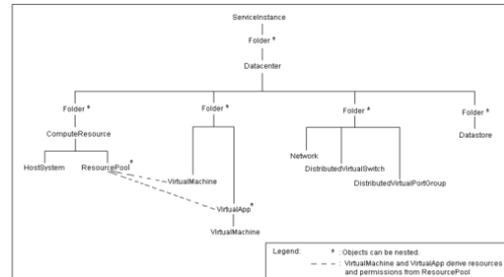


Fig. 2. vCenter Server Inventory Hierarchy as seen through MOB [3]

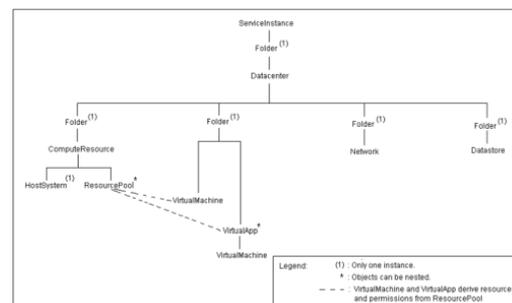


Fig. 2. ESXi Server Inventory Hierarchy as seen through MOB [3]

C. Illustration

When we start a session, vSphere creates a Service Instance with one root folder, one Data center, and four folders that hold the different types of inventory objects. We collect various properties of the given server by using API functions available. These are the properties required for specific feature to be used.

For Example, suppose the Administrators want to know the vMotion feature compatibility among a group of servers present in a vCenter Server. Here we will elaborate how the approach in our prototype will perform the task of fetching details and comparing. In our system, the login function is executed using the credentials provided by the administrator. Server is logged in using Service Instance function available in VIM API [5].

>> ServiceInstance(url, username, password, ignoreCert)

Properties of each host in the data center are collected by tracing the inventory hierarchy mentioned as in above Figure 4. The properties might be license of server or any hardware details to make sure the server is compatible for vMotion. For any feature compatibility check, the license of the host is checked primarily.

Service Instance being the root folder of inventory, all the folders are traced through it. To get license manager object we trace from Service Instance object [3].

```
>> ServiceInstance.getLicenseManager().getLicenses()
```

This returns an array of type License Manager License Info[] containing the license information. Using this we will verify if the license includes the required vSphere feature. Similarly, to collect hardware details, we start from Managed Entity[] of data center type. The following code snippet from our prototype demonstrates how we are using VIM API [5] objects to fetch details of hosts.

```
>> ManagedEntity[] m1 = ServiceInstance.getRootFolder().getChildEntity();  
//Managed Object Reference(MOR) for each managed entity is drawn.  
>> Datacenter dc = new Datacenter(ServiceInstance.getServerConnection(), m1.getMOR());  
// dc is MOR for datacenter  
>> Folder hf = dc.getHostFolder();  
>> ManagedEntity[] m2 = hf.getChildEntity();  
>> ComputeResource cr = new ComputeResource(si.getServerConnection(), m2[0].getMOR());  
// cr is MOR for ComputeResource  
>> HostSystem[] cr = cr.getHosts();  
>> Datastore[] ds1 = cr[0].getDatastores();  
// Returns names of datastores array  
>> ds1[0].getName();  
// returns name of datastore  
>> ds1[0].getSummary().getUrl();  
// returns UUID of the datastore  
>> cr[0].getHardware().getCpuPkg()[0].getDescription();  
// returns String summary description of CPU  
>> PhysicalNic[] pl = cr[0].getConfig().getNetwork().getPnic();  
>> pl[0].getLinkSpeed().getSpeedMb();  
// returns the speed of NIC  
>> cr.getConfigManager().getHostStorageSystem().getDiskName();  
// returns naa id
```

For instance, In order to verify if the given host individually is eligible for vMotion, we check for NIC speed. We use `pl[0].getLinkSpeed().getSpeedMb()` to get the NIC information and confirm it to be minimum of 1Gbps. This is one of the primary constraints of vMotion [6].

Another constraint for vMotion to be possible between any two hosts is, Shared Data store. To validate this we compare the data store name and the disk backing the data store to be same across hosts.

```
>> if( ds1[i].getName().equals(ds2[i].getName()) && ds1[i].getDiskName().equals(ds2[i].getDiskName())  
// Shared Datastore is confirmed
```

These kind of comparisons are made on each criteria of vMotion and is presented in the form of a matrix to the administrators with one-to-one mapping of all hosts

The administrator is presented with physical eligibility of each server for vMotion discretely, and a matrix with consolidated report of all the possible combinations of hosts for vMotion.

IV. RELATED WORK AND USE CASES

A. Related Work

Motivation behind the idea we proposed derives clues from tools which give insights on a specific vSphere feature. VMware fling tool "ESX System Analyser" [7] is an example which only helps administrators to plan the VM migration. But currently according to our research

across the web we see that users do not have a tool which can provide an extensive matrix of available vSphere features in a datacenter. Given that vCenter Server already has the operational ability to validate the host for any feature, why do we need a tool like this? But in VC server, validation of the feature is done per host basis only when you try to perform the operation. Here we are looking at a tool which validates and gives consolidated information in-hand, prior to performing the task.

B. Use Cases

1. Administrators of an existing datacenter want to perform vSphere operations and are not aware of the capabilities of each server in datacenter. They have to go through a rigorous task of going through architecture of datacenter or manually check the feasibility of servers in contention.
2. Challenge for vSphere testers trying to create a test environment to execute a specific test will be to identify a server/datacenter with desired hardware capabilities.

vAnalyser will help administrators and testers with an comprehensive feasibility report of each server assisting in decisions.

V. FUTURE WORK AND CONCLUSION

vSphere is the most used virtualization solution across the globe and is already a major factor of IT infrastructure. With commendable number of features, vSphere makes virtualization very effective. But in the same time numbers of features increase the complexity of vSphere administrator's job. vAnalyser with its approach can fetch and analyse a treasure trove of data and can be a great recommendation tool. So far we have traversed through only limited set of vSphere features. In future vAnalyser can be apprehended for almost every feature vSphere provides. Right now vAnalyser is a standalone tool. As a logical step in future we can have prospects of offering this as a plugin to the vCenter Server, since it uses existing VIM API and MOB relevant to vSphere. vAnalyser provisions vSphere testers or administrators with a user-friendly recommendation tool.

REFERENCES

- [1] Gartner "Magic Quadrant for x86 Server Virtualization Infrastructure," by Thomas J. Bittman, George J. Weiss, Mark A. Margevicius, Philip Dawson, June 27, 2013
- [2] Java: The Complete Reference, Seventh Edition- Mcgraw-Hill Education
- [3] vSphere Web Services SDK Programming Guide vSphere Web Services SDK 5.5 http://pubs.vmware.com/vsphere-55/index.jsp#com.vmware.wssdk.pg.doc/PG_Preface.html
- [4] Developer's Setup Guide VMware vSphere Web Services SDK 5.5 [http://pubs.vmware.com/vsphere-55/index.jsp#com. vmware.wssdk.dsg.doc/sdk_sg_preface.html](http://pubs.vmware.com/vsphere-55/index.jsp#com.vmware.wssdk.dsg.doc/sdk_sg_preface.html)
- [5] VMware vSphere API Reference Documentation <http://pubs.vmware.com/vsphere-55/index.jsp#com.vmware.wssdk.apiref.doc/right-pane.html>
- [6] VMware vSphere Documentation. <https://www.vmware.com/support/pubs/vsphere-essxi-vcenter-server-pubs.html>
- [7] ESX System Analyser. <https://labs.vmware.com/flings/esx-system-analyzer>