TRACK VM USAGE

(Amit/Shayan/Sashank)

**1.1 INTRODUCTION**

This chapter of the document explains about the technical details involved in tracking the virtual machine usage. We try to figure out all the VM statistics to be considered for collecting provenance data, we first analyze the VM parameters which are already present in VCL database which we can use and then look to figure additional parameters such as CPU utilization and memory utilization of individual VMs and the VM-host on which the VM instances are running.

**1.2 SPECIFICATIONS**

This section of the document explains in detail the categorization and technical aspects involved in tracking the VM usage.

**1.2.1 Categorization:**

We have classified the VM characteristics to be grouped as two major categories. One would be dealing with VM Load Statistics(like CPU, Memory utilization) collected hourly/daily based on a parameter which is tunable. The other category gathers the VM metadata of specific VM’s.

As a result of this categorization we have come out with the idea to create two tables in the VCL database.

**1)** VMusage (id,stateid,vmname,mnid,last updated, CPU(%) , Mem(%) )

|  |  |
| --- | --- |
| **Column** | **Description** |

|  |  |
| --- | --- |
| id | The id of the VMusage entry |
| state id | Determines the state of the VM |
| vmname | Virtual Machine hostname |
| mnid | Management node-id corresponding to the VM |
| lastupdated | Time at which the entry has been recorded in to the table |
| CPU(%) | CPU utilization at that point of time (tracked by lastupdated field entry) |
| Mem(%) | Memory utilization at that point of time (tracked by lastupdated field entry) |

This table will have multiple entries for each VM which mainly reflects the CPU, Memory load metrics collected hourly/daily identified by the “lastupdated” field. Also as the table grows large enough we will be rolling out the older entries. This effectively controls the size of the table.

2)VMmeta(id,stateid,vmid,vmname,OS,RAMinfo,Start\_time,End\_time,PrivIP,PublicIP,eth0,eth1,listening\_ ports)

|  |  |
| --- | --- |
| **Column** | **Description** |

|  |  |
| --- | --- |
| id | The id of the VMmeta entry. |
| stateid | Determines the state of the VM |
| vmid | Vm id corresponding to the VMhost |
| vmname | Hostname of the virtual machine |
| OS | Operating system loaded on the VM |
| RAMinfo | RAM allocated to the VM |
| Start\_time | Time at which the VM has been allocated |
| End\_time | Time at which the VM reservation ends |
| PrivIP | Private IP address of the VM |
| PublicIP | Public IP address of the VM |
| eth0 | MAC address of eth0 interface |
| eth1 | MAC address of eth1 interface |
| listening\_ports | Ports on which the virtual machine is listening |

This table reflects per VM’s statistics. This is similar to the “computer” table already existing in the VCL database but it gathers some additional details like listening\_ports.

**1.2.2 Technical Aspects**

We developed a script which will run on the management node to generate a statistic log having all the cpu and memory utilization details of all vm instances collected hourly/daily using a Crontab scheduler.

a) We will follow the below steps for generating the VMusage table:

1. Script(vm\_loadmetrics.sh) running as cron job at specific interval on management node.

2. Script(cpu\_mem\_check.sh) is remotely executed on the vmhost using ssh and generates the statistic log(CPU and Memory usages) for all vm's running on the box.

3. Statistic log is pushed back to the management node where the corresponding database table is populated with an entry which reflects the gathered CPU,Mem stats.

b) For generating the VMmeta table we will use some of the existing data resources like “computer” , “computerloadflow” , “computerloadstate”, “log” and “image” tables of the VCL database. Apart from these, for collecting the information regarding listening ports on each vm we will be using “nmap” utility on vmhost. (Ex:-- nmap -v -sV 192.168.50.10/32 ).

**1.3 TASKS**

We have identified certain ways to gather the CPU, Memory utilisation(by using the “ps aux” command on the Vmhost) and started with the coding for the project relying on the above assumption.

**1.3.1 Action Items Completed:**

1). Analyzed the existing tables of VCL and determined the statistic parameters which are present in the database that can be used to provide statistical VM information.

2).Script (cpu\_mem\_check.sh) - Task for gathering cpu and memory utilization on specific vmhost of all the VM’s running on it. Currently the data is being persisted in statistic log which is rotated with a tunable parameter.

3) Script (vm\_loadmetrics.sh) - Script set as cron job and the implementation of remote execution of cpu\_mem\_check.sh on vmhost and scping the log back from vmhost to management node to gather memory and cpu percentage utilization.

4). Have come up with the table schema for the two new tables which we will be creating as mentioned in section 1.2.1

**1.3.2 Pending Items:**

1) Need to implement the database table for the VMusage , VMmeta on the VCL database.

2) Develop a perl script which polls the existing tables in the VCL database and gather the

statistics which will be used in generating the new tables.

3) Rolling out the old logs, entries from the database table(VMusage) to restrain the size

of the database table, log.

**1.4 Conclusion:**

Our final implementation will provide useful provenance data to monitor the important VM characteristics. This statistical information from the database tables and logs can be used by the Team 4 working on Provenance policy.