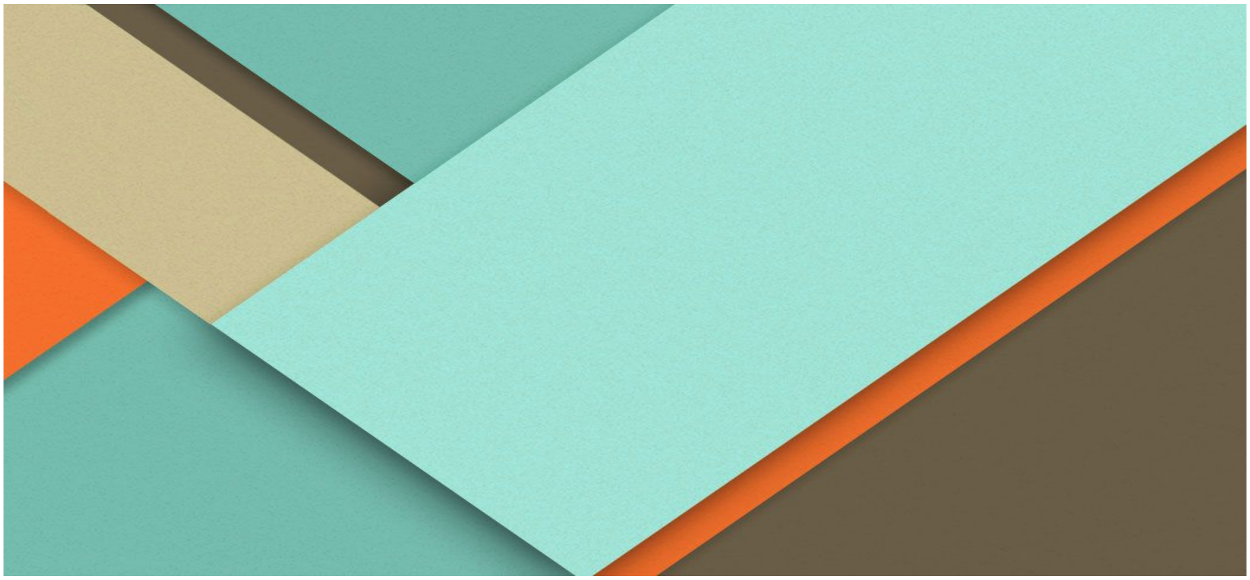


Cloud Computing

Monitoring Dashboard for Baadal

17.11.2017

Submitted by: [GROUP 15]

Aditi (2014CS10205)
Aditi Singla (2014CS50277)
Aditi Gupta (2017BSY7508)

Contents

Introduction

Baadal is a management and orchestration software for the IITD private cloud for high performance computing based on 32 blade servers each with 48 core Intel(R) Xeon(R) CPU and 10 TB of virtualized storage. Some of the main features of Baadal are:

1. Dynamic resource scheduling and power management.
2. Facilities for suspend, resume, shutdown, power off, power on and specifying resource requirement of virtual machines.
3. Dynamic resource utilization monitoring.

Assignment Purpose

We have designed, developed and integrated a Monitoring Dashboard module in Baadal displaying information about different aspects of (at any time instance or over a period of time):

1. CPU and RAM Utilisation for Hosts and VMs.
2. Disk Load attached to VMs for Read & Write.
3. Network Traffic of VMs for Read & Write.

This dashboard is meant for the administrator to monitor the resource consumption to avoid any overloading on any host.

Installation of Baadal

We tried to use one of the VMs provided to us, but we realised that, it would not be able to support an 8 GB RAM VM on it, since it itself can have at max 8 GB RAM.

We were finally able to install Baadal on virtualbox in a desktop machine in Software Telecom Lab,Bharti School.

Steps for installation of Baadal on a VM (8 GB RAM, 100 GB HDD, 8 CPUS), with ubuntu 14.04 are as follows.

1.Specification of host machine:

RAM:16 GB
Processor: 8 core
Operation System 16.04
Hard disk: 1TB

2.Prerequisite for Baadal server:

Operating system:clean Ubuntu 14.04 Desktop Version
RAM:minimum 8 GB
Processor 8 core

3.Settings for configuration:

A. Set environment variable for proxy server:

```
$export http_proxy="http://10.10.78.61:3128"
$export https_proxy="https://10.10.78.61:3128"
```

B. Updating Repositories Source list

```
$sudo apt-get update
```

C. Fetch Baadal repo

```
$git clone https://github.com/iitd-plos/baadal2.0
```

D. Run Installation command

```
$ cd ~/baadal2.0/baadaltesting/devbox
$ sudo -E make devbox
```

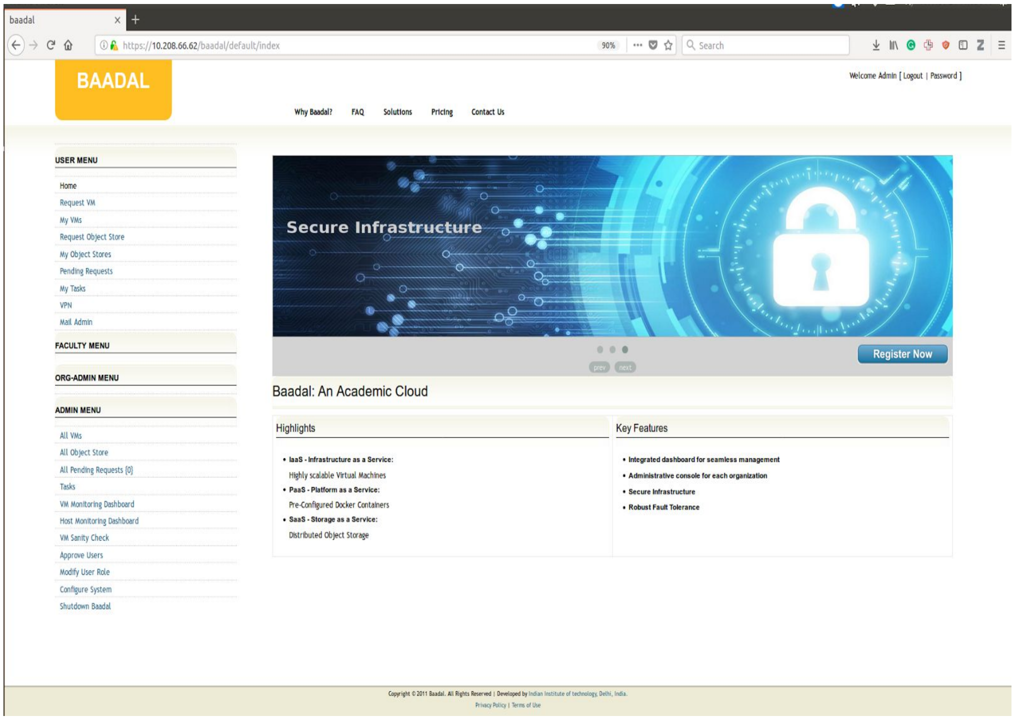
It takes nearly 15-20 mins depending upon your internet speed to complete the installation.

4.Post Successful Installation

The makefile installs an Apache server. At this point, the default baadal GUI is visible on localhost. Open https://10.208.66.62/baadal/default/index in browser.

5. Explore Baadal GUI

1. Login into baadal as webpage admin:
    a. Userid:admin
    b. Paasword: baadal



Baadal: An Academic Cloud

Highlights

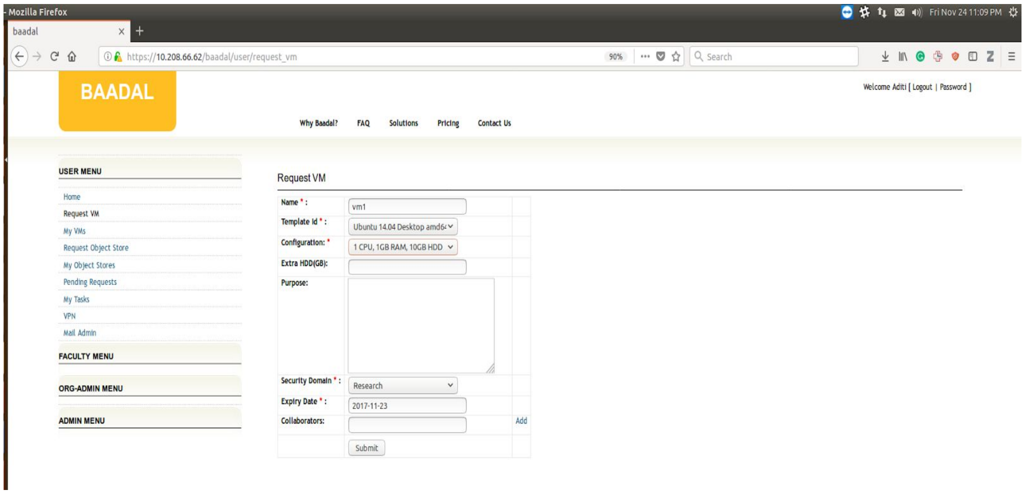
- **IaaS - Infrastructure as a Service:**
  Highly scalable Virtual Machines
- **PaaS - Platform as a Service:**
  Pre-Configured Docker Containers
- **SaaS - Storage as a Service:**
  Distributed Object Storage

Key Features

- **Integrated dashboard for seamless management**
- **Administrative console for each organization**
- **Secure Infrastructure**
- **Robust Fault Tolerance**

Copyright © 2011 Baadal. All Rights Reserved | Developed by Indian Institute of technology, Delhi, India.
Privacy Policy | Terms of Use

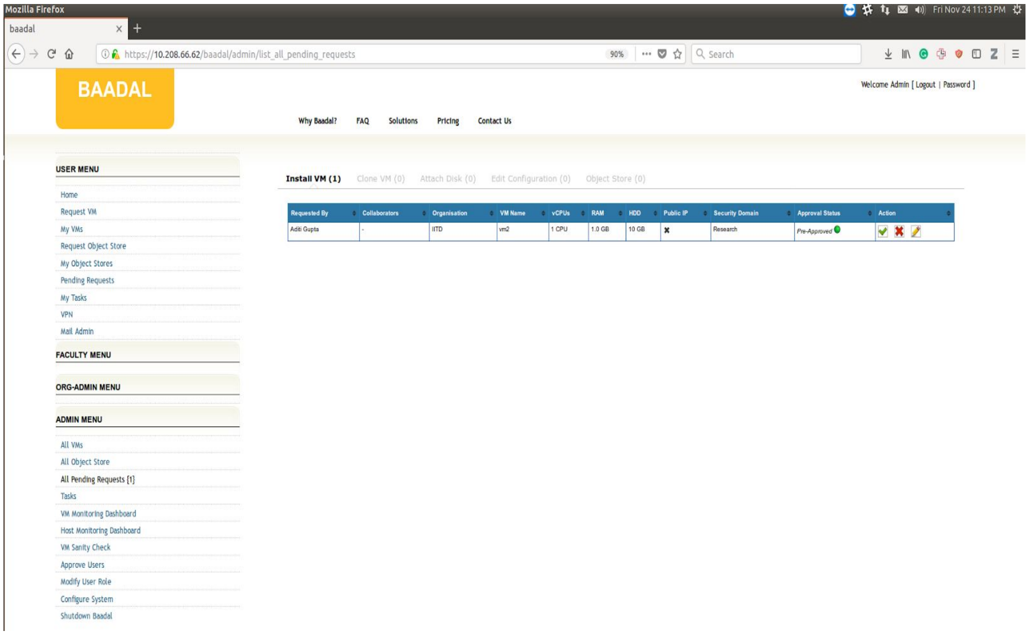
2. Create new users and request new VMs from them



Request VM

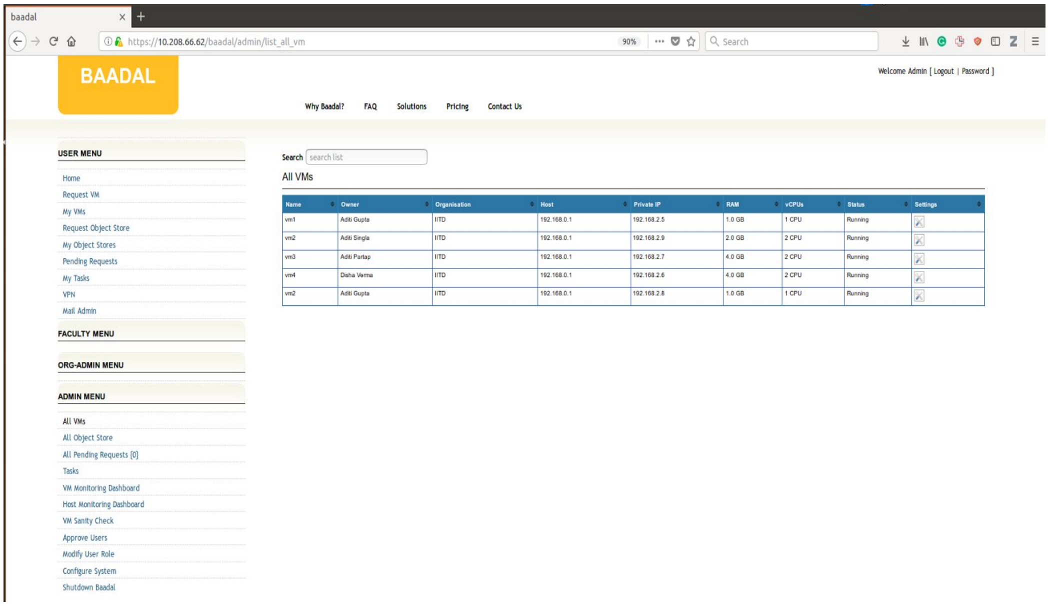
Name * :	vm1
Template Id * :	Ubuntu 14.04 Desktop amd64
Configuration: *	1 CPU, 1GB RAM, 10GB HDD
Extra HDD(GB):	
Purpose:	
Security Domain * :	Research
Expiry Date * :	2017-11-23
Collaborators:	

3. VM approval by admin



Requested By	Collaborators	Organisation	VM Name	vCPUs	RAM	HDD	Public IP	Security Domain	Approval Status	Action
Aditi Gupta	-	IITD	vm2	1 CPU	1.0 GB	10 GB	✖	Research	Pre-Approved ●	✓ ✖ ✎

4. Check VM status and configuration for all at once.



All VMs

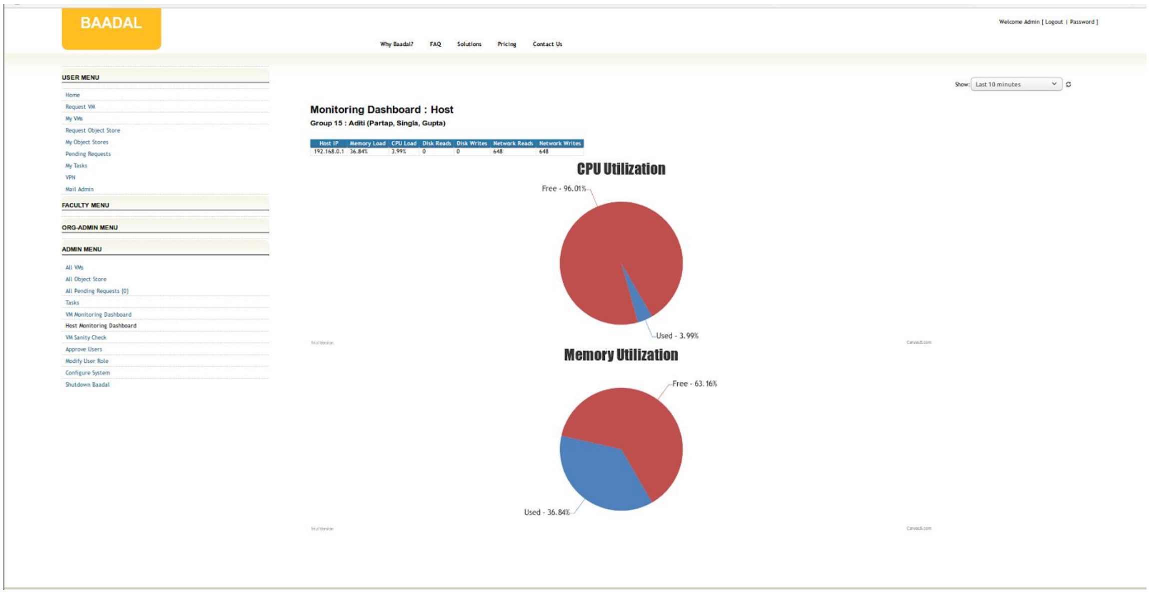
Name	Owner	Organisation	Host	Private IP	RAM	vCPUs	Status	Settings
vm1	Aditi Gupta	IITD	192.168.0.1	192.168.2.5	1.0 GB	1 CPU	Running	
vm2	Aditi Singla	IITD	192.168.0.1	192.168.2.9	2.0 GB	2 CPU	Running	
vm3	Aditi Partap	IITD	192.168.0.1	192.168.2.7	4.0 GB	2 CPU	Running	
vm4	Disha Verma	IITD	192.168.0.1	192.168.2.6	4.0 GB	2 CPU	Running	
vm2	Aditi Gupta	IITD	192.168.0.1	192.168.2.8	1.0 GB	1 CPU	Running	

5. Modify the web2py code to make our own webpage, with better graphical representation of the VM and Host status.

Dashboard Design

1. **Host Monitoring Dashboard :** A button to view this dashboard has been added in the menu on the left side. This dashboard shows the information about the Baadal Host, on which the VMs are running. The statistics for various types of informations are displayed in a table and in pie charts, for better understanding from graphical representation. The information displayed is:
    a. Memory Utilisation
    b. CPU Utilisation
    c. Disk Reads & Writes
    d. Network Reads & Writes

    There is a drop down menu to select a duration over which the average stats need to be displayed. They range from last 10 minutes to last 1 year.



Monitoring Dashboard : Host

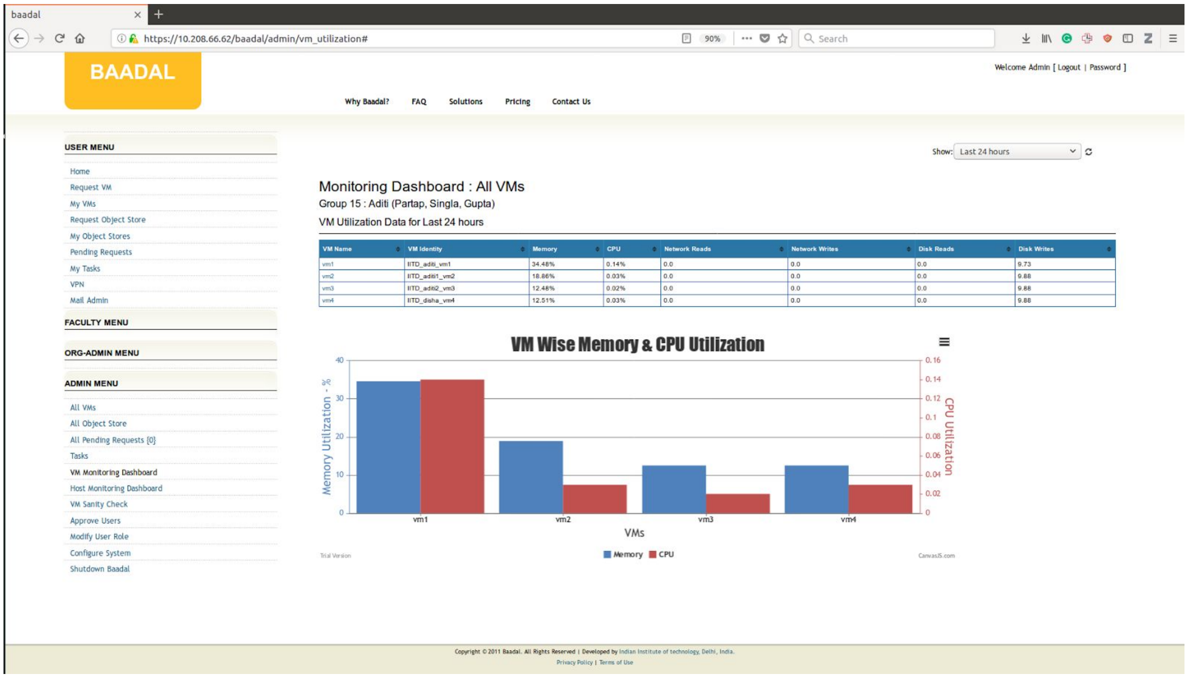
Group 15 : Aditi (Partap, Singla, Gupta)

Host IP	Memory Load	CPU Load	Disk Reads	Disk Writes	Network Reads	Network Writes
192.168.0.1	36.84%	3.99%	0	0	648	648

2. **VMs Monitoring Dashboard :** Again, the button is in the window in the left menu. On clicking it, one can the list of all the VMs present, with their Vm Id, Memory & CPU loads, Network and Disk Reads & Writes.

   For easy evaluation of the current status, a bar graph comparing the memory and cpu loads for all the VMs side by side. This helps the administrator to analyse the consumption patterns, and avoid overloading of resources.

   Again, an option is present, to change the duration over which the data is to be averaged.



Monitoring Dashboard : All VMs

Group 15 : Aditi (Partap, Singla, Gupta)

VM Utilization Data for Last 24 hours

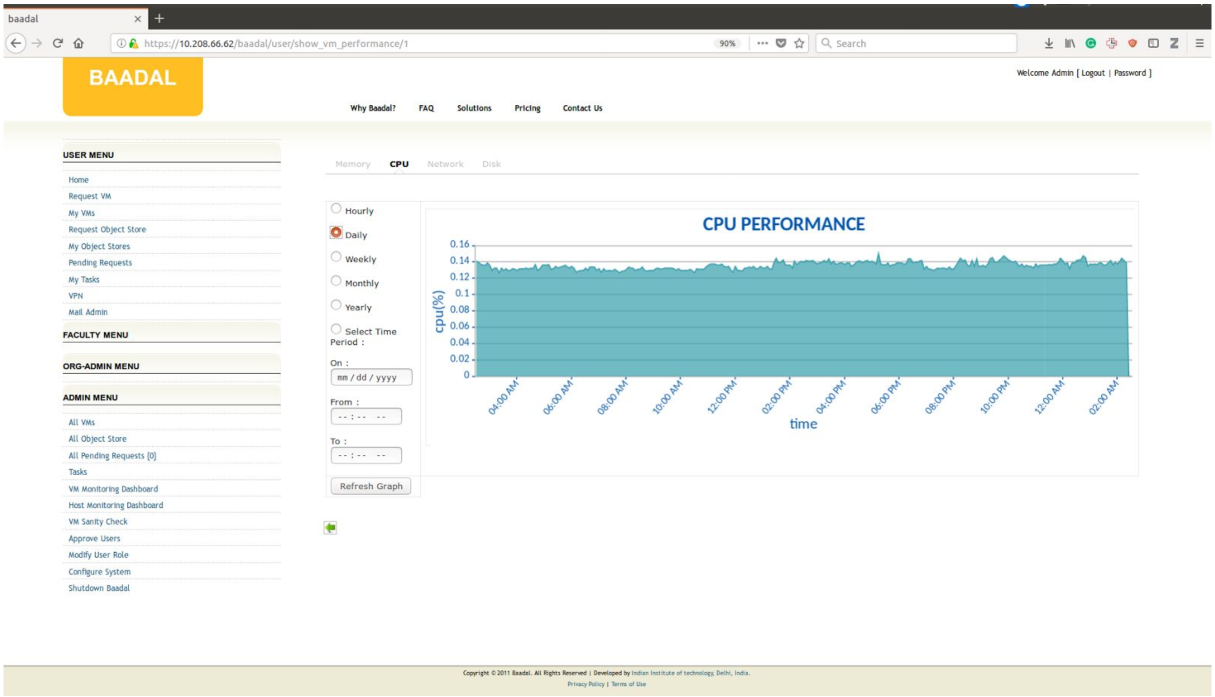
VM Name	VM Identity	Memory	CPU	Network Reads	Network Writes	Disk Reads	Disk Writes
vm1	IITD_aditi_vm1	34.48%	0.14%	0.0	0.0	0.0	9.73
vm2	IITD_aditi1_vm2	18.86%	0.03%	0.0	0.0	0.0	9.88
vm3	IITD_aditi2_vm3	12.48%	0.02%	0.0	0.0	0.0	9.88
vm4	IITD_disha_vm4	12.51%	0.03%	0.0	0.0	0.0	9.88

3. VMs with excessive memory or CPU usage (over 80%) are displayed in red color to bring it into the notice of the admin, for extra resource provisioning.

4. **Per VM Monitoring Dashboard :** From the list of VMs displayed in the VMs Monitoring Dashboard, one can click on the name of any of the VMs to view the details of that VM. On the new page, on can see different tabs for Memory, CPU, Network & Disks. Here under each tab, the graph for usage pattern is displayed, which is very useful to note down the points of failure and identify any patterns for better utilisation of resources.

   Again, we have different options for different periods, over which the pattern from the graph needs to be identified. Also, one can view the graph for a specified time period, with date and start & end times.

   There is an option to go back to the VMs Monitoring Dashboard as well.



https://10.208.66.62/baadal/user/show_vm_performance/1

○ Hourly
● Daily
○ Weekly
○ Monthly
○ Yearly
○ Select Time Period :

On : mm / dd / yyyy

From : -- : -- --

To : -- : -- --

Copyright © 2011 Baadal. All Rights Reserved | Developed by Indian Institute of technology, Delhi, India.

Implementation & Code Changes

The dashboard functionality has been integrated with the existing Baadal User interface, using web2py framework, used by Baadal.

Design :

- The web2py employs the MVC design pattern, i.e. Model View Controller, to implement the website.
- The models manage all the data and state, independent of the UI. All the information about the VMs' cpu, memory, etc, as well as about the hosts' cpu, etc is stored in a Round Robin database periodically.
- The view is comprised of HTML pages, which present information to the user. It also allows some manipulations by the user.
- The controller is the intermediary between the model and the view. It updates the view when model changes and vice versa. The application logic lies mostly in the controllers.
- This structure ensures easy reusability and modifications on the application.

Code Changes :

Models -

The file modules/vm_utilization.py contains the functions that collect the CPU, Memory, Disk & Network statistics of all the VMs. The function **fetch_rrd_data** in this file is used by the model files to get the stats.

The function rrd_database_lookup() is used for querying the stats for different time durations.

In the admin_model.py file in the models folder, we made the following changes :

- The time period for which the user wants the statistics is taken as input, and sent to the model, so as to fetch data accordingly from the RRD database. The data for the last few minutes till several years is available.
- **get_host_util_data** : This function was modified so that now it returns an object containing {memory, cpu, disk reads, disk writes, network reads, network writes} for each host. (This info was already being returned by the fetch_rrd_data function)
- **get_vm_util_data** : This function was modified so as to also return the RAM and Identity of each VM.

Changes in models/menu.py file are done to add the buttons to the admin menu, since the UI fetches the list of buttons from this file.

- 'Host Monitoring Dashboard' button added to the Admin menu. It calls the 'hosts_vms' API.
- 'VM Monitoring Dashboard' button added to the same menu. It calls the 'vm_utilization' API.

Views

The HTML files were updated to add the extra functionalities ad render the graphs or pie charts as per the user requirements.

CanvasJS was used to build and render all the bar & pie charts.

- admin/hosts_vms.html : This is the file that is displayed on clicking on the Hosts Monitoring Dashboard button. This was modified to display our group information. A custom table was added to show all the statistics of CPU, Memory of different hosts on this page. Also, **pie charts** - 2 per host, that display the hosts' relative %age of free memory and CPU are also displayed to build a more intuitive interface for the users.
- admin/vm_utilization.html : This is the page displayed on clicking on the VMs Monitoring Dashboard button. The existing table was modified to also display the identity of the VMs, which allows for easy identification of the VMs. Also, a **double bar chart** was added, which displays the Memory and CPU utilization of each VM in one graph. Every vm Id is clickable, which takes the user to a separate page.
- users/show_vm_performance.html : This is the page which is displayed when the user clicks on a VM Id. Here, we provide the user with 4 different tabs, one for each of Memory, CPU, Disk and Network stats. Under each tab, the user has the option of viewing an **area graph** with information about the usage/numbers for different time durations. The options of hourly, daily, weekly, monthly, yearly, as well as the option of selecting a time period on a particular date is provided.

All the graphs are animated, and show the details of a particular data point as the user hovers over it.

Controllers

Controller files are used to convey the data fetched by the models file to the user interface.

In the controllers/admin.py file, the following changes were made so as to send the relevant data to the UI :

- **hosts_vms** : This is the API that the UI calls when the user clicks on 'Host Monitoring Dashboard' button in the admin menu. This API was updated so as to return all the information like, CPU usage, Memory usage, disk & network read - writes about the host to the view.
- **vm_utilization** : This API is called on opening the 'VM Monitoring Dashboard'. This was updated to show the VM info on the page appropriately.

In the controllers/user.py file, the following functions were modified :

- **show_vm_performance** : This API is called when the user clicks on a particular VM. It calls the function get_vm_info to get all the details of the VM.

Modules

- The file vm_utilization.py in modules folder was updated so as to return the required data to the controller function.
- Needed to plot graphs in per vm page.