High Level Design Document
Bin-packing VM Consolidation Algorithm

Atchutuni Bhavana 13MCMT01 Terli Venkatesh 13MCMT55
Surineni Sampath Kumar 13MCMT49

April 1, 2014

Contents

1 Introduction

The purpose of this document is to depict the high level design and the data flow diagram of bin packing vm consolidation algorithm project.

2 Modules in the system

We have decided to divide the whole project into 3 modules. They are

1. **User Interface**

   This module is the main interface to the user and is responsible for building, editing and updating the GUI.

2. **Parser**

   This module reads the input from file and initializes PMs and VMs as specified in it.

3. **PM modifier**

   This module is responsible for creating physical machines(PM), adding virtual machines(VM) and doing modifications to them. This is also responsible for consolidation operation.

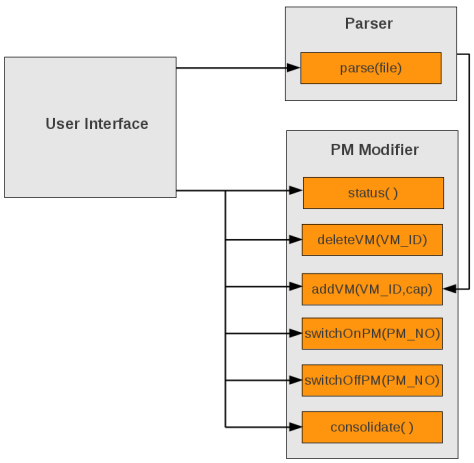

Figure 1: Interfaces between modules

2.1 Data flow diagram

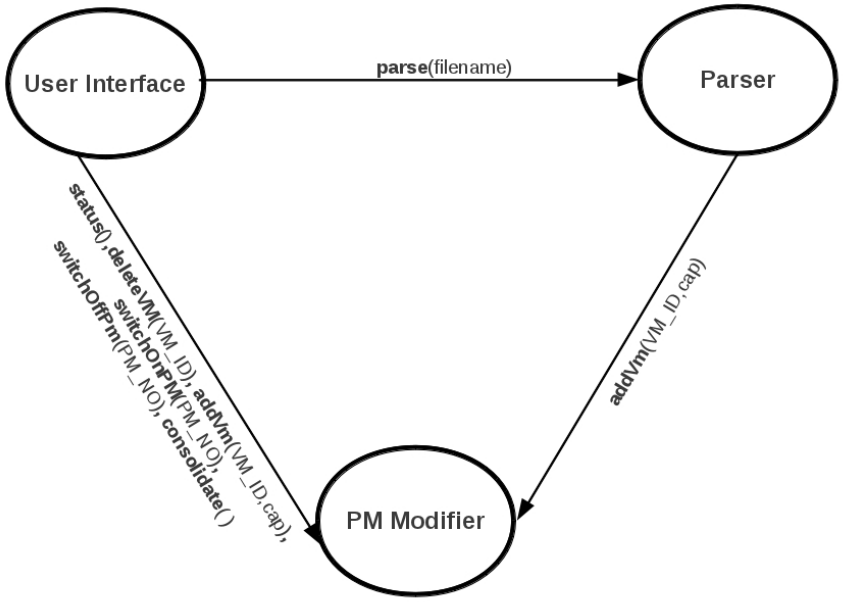

Figure 2: Data flow diagram

2.2 API Specification

2.2.1 Modules of the architecture

User Interface Module

- **Functionality**

  The main purpose of this module is to take input from the user and reflect the system state to the user.

Parser

- **Functionality**

  The aim of this module is to parse a text file specified by the user, extract the information about PM's and VM's in it. It then initializes the PM's and adds VM's to them by the help of PM modifier module.

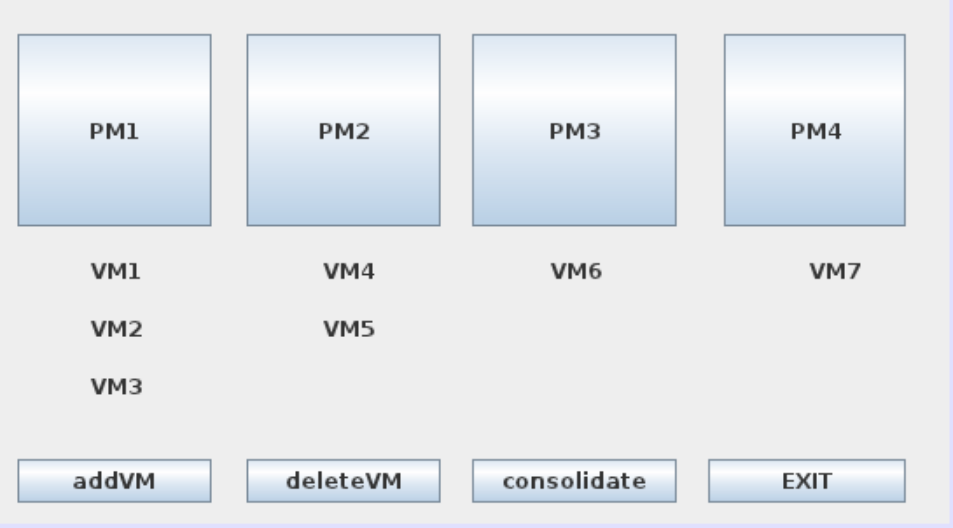

Figure 3: Main user interface

- **Interface Description**

  parse(filename)

Purpose	:The purpose of this function is to parse the file specified by the user and extract information about PM's and VM's.
Input Parameter	: file path specified by the user
Output Parameter	: VM ID and capacity
Return Value	: If the file is not in the specified format it would return **wrong file format** message
Called by	: User Interface module
Calls	: addVM function of PM modifier

PM modifier

- **Functionality**

  The operations of this module include creating a PM, adding VM's to it, calculating the residual capacity and consolidation.

- **Interface Description**
  addVM(VM_ID, cap)

Purpose	: The purpose of this function is to add VM's to the PM as specified by Parser module and UI module.
Input Parameters	: VM ID and VM capacity.
Output Parameter	: none

Return Value : If there is no enough space to add VM to a PM it outputs **No enough space** message
Called by : Parser module and UI module
Calls : NONE.

deleteVM(VM_ID)

Purpose : The purpose of this function is to delete a VM specified by the user.
Input Parameters : VM ID.
Output Parameter : none
Called by : User Interface module
Calls : none.

switchOffPM(PM_NO)

Purpose : The purpose of this function is to switch off a PM specified by user.
Input Parameters : PM number.
Output Parameter : none
Return Value : Returns **It is not possible to switch off this PM at this time** if switching off was not successful.
Called by : User Interface module
Calls : none.

switchOnPM(PM_NO)

Purpose : The purpose of this function is to switch on a PM specified by user.
Input Parameters : PM number.
Output Parameter : none
Return Value : none
Called by : User Interface module
Calls : none.

consolidate( )

Purpose : The purpose of this function is to run Bin packing algorithm and consolidate all the VM's into minimum number of PM's.
Input Parameters : none
Output Parameter : none
Called by : User Interface module
Calls : none.

status( )

Purpose	: The purpose of this function is to return the information about all the PM's and the VM's in them.
Input Parameters	: none
Output Parameter	: information about the system
Called by	: User Interface module
Calls	: none.