High Level Design Document
Bin-packing VM Consolidation Algorithm

Surineni Sampath Kumar 13MCMT49

Contents

1 Detailed Design

1.1 PM Modifier Module

This module will be called by Parser module and User Interface module for

- Adding a Virtual Machine(VM),

- Deleting a VM,

- Switching off a PM,

- Switching on a PM and

- Consolidation

1.1.1 Interface Data Structures

1. PMstruct

2. VMstruct

PMstruct
   Different fields in PMstruct data structure are

1. PM_id - final String

2. res_cap - integer

3. VM_list - class VMstruct

   This is the data structure returned to status() function which is called by User Interface

1.1.2 Internal Data Structures

NONE

1.1.3 Interface Functions

 void deleteVM(VM_ID)
   Description : The purpose of this function is to delete the VM which is passed as an input parameter to while calling this function.

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, parses it and initializes PMs and VMs as specified in it.

3. **PM modifier**

   This module is responsible for adding virtual machines(VM) and doing modifications to Physical Machines(PM)s. This is also responsible for consolidation operation.

1.2 Data flow diagram

1.3 API Specification

1.3.1 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.

- **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 which are homogenous and have capacity of 100. Adds VM's to them by the help of PM modifier module.

  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	: none
Return Value	: If the file is not in the specified format it would display **wrong file format** message
Called by	: User Interface module
Calls	: createPM and addVM functions of PM modifier
Data type	: **file** - Class File

PM modifier

- **Functionality**

  The operations of this module includes adding VM's to PM, 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 the parser.
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
Calls	: none.
Data type	: **VM_ID** - Class VMstruct, **cap** - int

deleteVm(VM_ID)

Purpose	: The purpose of this function is to delete a VM specified by the user.
Input Parameters	: PM number in which VM resides and VM ID.
Output Parameter	: none
Called by	: User Interface module
Calls	: none.
Data type	: **VM_ID** - Class VMstruct

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	: Displays **It is not possible to switchoff this PM at this time** if switching off was not successful.
Called by	: User Interface module
Calls	: none.
Data type	: **PM_ID** - Class PMstruct

switchOnPm(PM_NO)

Purpose	: The purpose of this function is to switch on a PM specified by user.
Input Parameters	: PM number.
Output Parameters	: none
Return Value	:
Called by	: User Interface module
Calls	: none.
Data type	: **PM_ID** - Class PMstruct

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 inform the User Interface module the current staus of all the PMs all the VMs in it.
Input Parameters	: none
Output Parameter	: head of the linked list of PMs
Called by	: User Interface module
Calls	: none.
Data type	: **PM_ID** - Class PMstruct