

北京航空航天大学计算机学院

## 硕士学位论文中期检查报告

论文题目：IAAS 平台中虚拟机节能整合策略及其仿真环境研究

专    业：计算机软件与理论

研究方向：云计算

研  究  生：王    龙

学    号：SY1106517

指导教师：兰  雨  晴

北京航空航天大学计算机学院

2013 年 08 月 29 日

## 目 录

<b>1</b>	<b>论文工作计划 .....</b>	<b>3</b>
1.1	选题背景.....	3
1.2	论文研究目标.....	3
1.3	论文主要研究内容.....	4
1.4	论文预期成果形式.....	4
1.5	论文研究进度.....	5
<b>2</b>	<b>已经完成的工作 .....</b>	<b>6</b>
2.1	节能整合策略的整体架构.....	7
2.2	虚拟机部署策略.....	8
2.3	虚拟机节能整合策略.....	9
2.3.1	策略框架.....	9
2.3.2	虚拟机选择策略.....	10
2.3.3	物理主机过载检测算法.....	11
2.3.4	物理主机欠载检测算法.....	12
2.3.5	物理主机关机休眠策略.....	13
2.4	算法评估的指标.....	14
2.4.1	SLA 违反率.....	14
2.4.2	耗电量.....	15
2.4.3	迁移次数.....	15
2.5	模拟仿真软件实现.....	15
2.5.1	核心框架.....	15
2.5.2	数据中心事件处理.....	17
2.5.3	实体之间的通讯.....	18
<b>3</b>	<b>关键技术或难点 .....</b>	<b>19</b>
3.1	虚拟机节能整合模型.....	19
3.2	节能整合策略.....	21

3.3 算法策略的验证.....	21
<b>4 下一阶段工作计划 .....</b>	<b>22</b>
4.1 存在的问题.....	22
4.2 尚未完成的工作.....	22
4.3 解决问题的技术思路或措施 .....	22
4.4 下一阶段计划.....	22
<b>5 主要参考文献 .....</b>	<b>22</b>

# IAAS 平台中虚拟机节能整合策略及其仿真环境研究

## 1 论文工作计划

本章简要介绍了论文的工作计划，包括论文选题背景、论文研究目标、主要研究内容、论文预期成果形式和论文研究进度等内容，如下所示：

### 1.1 选题背景

随着云计算[1]的到来，更多的云计算资源存储在云端，给数据中心的管理带来了很大的挑战。数据中心的高能耗不仅仅是由于物理服务器的增多，处理能力的不断增强，还由于这些资源过低的使用率导致了巨大的电能浪费。

2010 年全球数据中心的能耗占据到所有能耗的 1.1%-1.5%，而美国的数据中心能耗占据到全美总能耗的 1.7%-2.2%[2]。此外，2011 年美国数据中心消耗的电能大约为 2006 年的 2 倍，约为 1000 亿千瓦时的电能，电费成本约为 90 亿美元，并且数据中心对电能的需求仍以每年 12% 的速度增长[3]。

据预计，数据中心 2020 年将成为世界上最大的能源消耗行业，此外，数据中心在消耗电能的同时，会排放出大量的二氧化碳，加重了温室效应，据估计，数据中心导致排放的二氧化碳占到全球总排放量的 2%[4]。

所以，数据中心的高能耗是一个亟待解决的问题。本论文针对中标软件有限公司对 IAAS 数据中心节能的切实需求，对如何降低数据中心的能耗问题进行相应的研究。

### 1.2 论文研究目标

1)、给出 IAAS 数据中心节能整合策略：本论文针对数据中心高能耗问题，特别是对目前云计算中 IAAS 数据中心高能耗的问题，通过分析研究其负载特点[5]和应用模式，提出基于虚拟机整合的节能策略，数据中心利用该节能整合策略能够有效的降低其电能消耗。

2)、设计并实现能够模拟仿真 IAAS 云计算环境的软件工具：本论文的另一个目标是研究如何利用仿真环境对节能整合策略进行有效性验证，设计并实现一个能够模拟仿真 IAAS 云计算环境的仿真软件。

### 1.3 论文主要研究内容

本论为的研究内容分为两部分：

#### (1)、基于虚拟机迁移的节能整合策略

基于虚拟机迁移的节能整合策略方法的核心是通过将虚拟机部署到尽可能少的物理主机上，然后将空闲的物理主机关停或者切换到低功耗状态，从而达到减少数据中心的电能消耗和降低云计算服务提供商运营成本的目的。

#### (2)、云计算仿真环境研究与实现

对于研究内容（1）中所提出的节能整合策略，需要对其有效性和节能效果进行验证，由于利用真实的云计算 IAAS 环境来验证算法策略代价很高，云计算运营商也不会轻易允许利用其生产环境来进行算法策略的验证。所以，本论为研究云计算的仿真环境，利用该模拟仿真软件，可以模拟 IAAS 云计算数据中心，并将研究内容（1）中的算法策略在其上进行模拟仿真，从而验证算法策略的有效性和节能效果。

### 1.4 论文预期成果形式

本论文预期达到的研究成果如下：

- 研究并提出 IAAS 平台中虚拟机节能整合策略，包括其理论，伪代码等
- 针对 IAAS 云计算环境的模拟仿真工具软件

- 发表 1—2 篇小论文
- 申请专利一项
- 软件著作权一项

## 1.5 论文研究进度

本论文的工作计划如表 1 所示,论文已经完成了总体计划的大约 80%的工作,通过研究 IAAS 平台中电能消耗的特点,从软件算法角度,完成了基于虚拟机迁移的节能整合策略,该策略使用伪代码进行描述。为了验证节能整合策略的有效性和节能效果,在 CloudSim[6]的基础上,设计并实现了针对 IAAS 的云计算模拟仿真软件工具,并利用该仿真软件对已有的节能整合策略进行模拟仿真,下一步计划利用该仿真软件对本论文提出的节能整合策略进行模拟仿真验证。

表 1 论文工作计划表

阶段	起始时间	工作内容
阶段 1	2012 年 12 月	研究分析影响云平台能耗的因素
	2013 年 1 月	研究 IAAS 云计算平台的应用场景特点
	2013 年 2 月	分析 IAAS 中能耗与性能的约束关系
	2013 年 3 月—4 月	根据前期的研究结果,提出 IAAS 平台中虚拟机节能整合策略
	2013 年 5 月	撰写小论文
	2013 年 6 月—7 月	设计并实现 IAAS 云计算环境模拟仿真软件工具
	2013 年 8 月	利用模拟仿真软件对已有的策略进行模拟仿真,并根据结果,修正前期提出的虚拟机节能整合策略
阶段 2	2013 年 9 月—10 月	利用模拟仿真软件,对论文前期提出的节能功能整合策略进行模拟仿真,验证其有

		效性，并分析其节能效果，根据仿真结果，对策了进行适当的修正。
阶段 3	2013 年 11 月	撰写大论文
	2013 年 12 月	修改大论文，准备毕业答辩

## 2 已经完成的工作

自论文开题以来，在调研国内外相关现状后，得出如下结论：目前国内外已有的 IAAS 节能方法的原理[7]是，将云中的虚拟机整合到尽可能少的物理主机上，并将富余的物理主机关停或者切换到低功耗状态，从而达到节能的目的，当数据中心负载急剧上升时，就会启动未开启的物理主机，来满足要求，其示意图如下（图 1）：

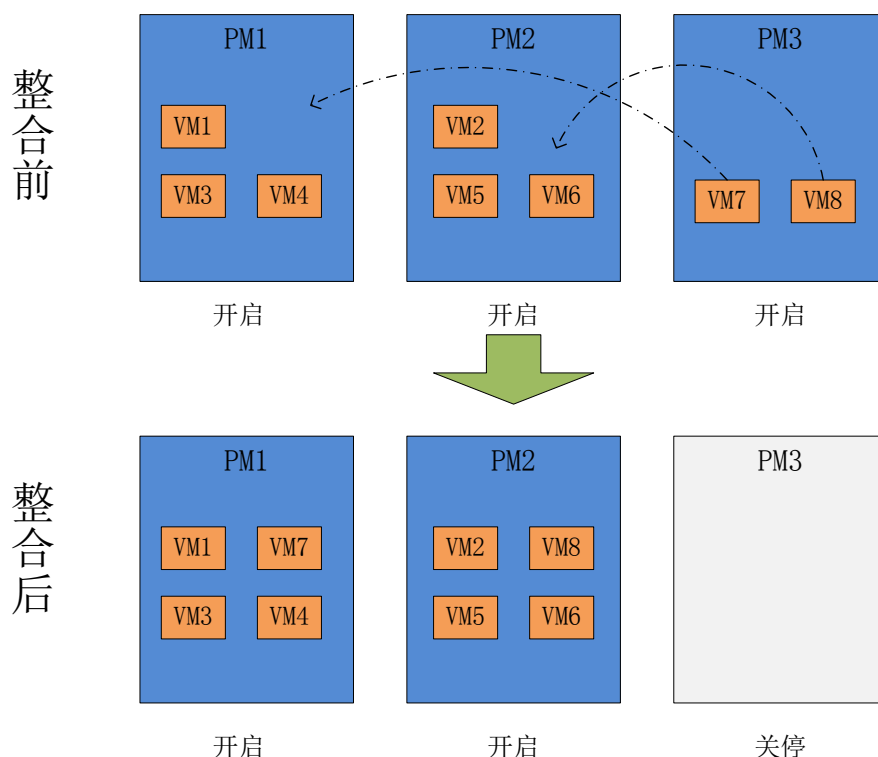


图 1 虚拟机节能整合

图 1 中将 PM3 中的虚拟机 VM7 和 VM8 分别迁移到物理主机 PM1 和 PM2 上，然后将 PM3 关闭掉，从而达到节能的目的。

基于以上结论，学习了包括 IAAS 云计算环境、开源 IAAS 平台的实现技术和降低能耗方法等在内相关技术，并在此基础上完成了如下的工作：

## 2.1 节能整合策略的整体架构

我们提出的虚拟机节能整合策略，是将虚拟机整合到尽可能少的物理主机上，关停空闲的物理主机或者将其切换到低功耗状态，从而达到节能的目的。在整个整合过程中，首先找出数据中心中过载的物理主机，即 CPU 利用率超过所设定的上限阈值的物理主机，然后迁移其上的一个或者多个虚拟机，使其 CPU 利用率低于所设定的上限阈值，接着找出数据中心的欠载的物理主机，即 CPU 利用率低于所设置的下限阈值的物理主机，将其上的虚拟机全部迁移走，进而关闭这些物理主机。

下文对我们所提出的节能整合策略进行详细的阐述。第一部分是节能整合策略的架构，接着分别介绍了虚拟机部署策略、虚拟机节能整合策略、虚拟机选择策略以及虚拟机过载欠载检测算法，最后介绍了数据中心中物理主机的关机和开机策略。通过实验表明，我们的节能整合策略大约能节省 20% 到 50% 的电能耗费（百分比是针对未采用为何节能措施的数据中心而言）。

其节能整合策略的总体架构图如图 2 所示。

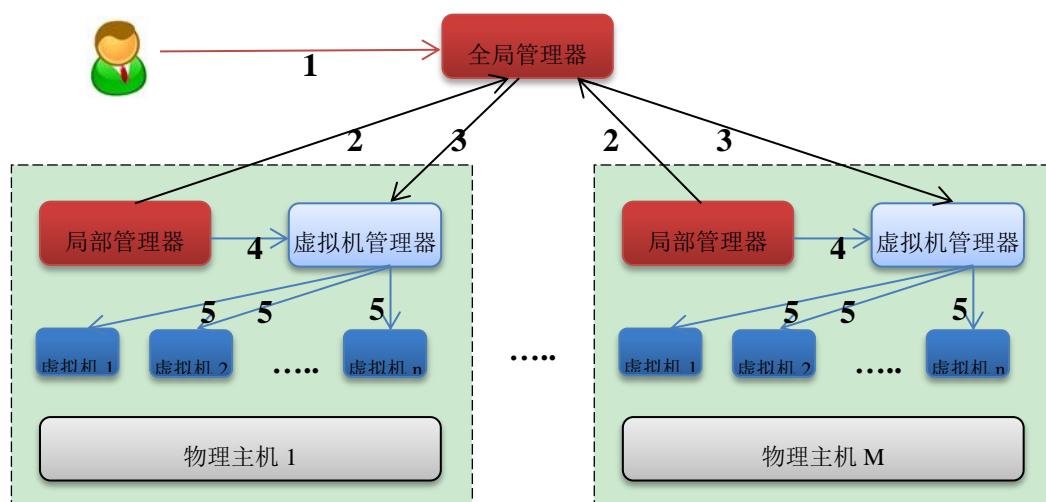




图 2 节能整合策略架构图

实现该节能整合策略的软件由全局管理器和局部管理器组成。局部管理器以虚拟机管理器(VMM)的一个模块的形式安装在每一个计算节点的物理主机上。它的任务是持续不断的监控计算节点的 CPU 利用率、根据资源使用情况调整虚拟机的大小和决定何时迁移哪一个虚拟机等(图 2 中的箭头 4)。全局管理器安装在主控节点上,它负责从局部管理器上手机所有计算节点的资源使用信息(图 2 中的箭头 2),并且通过给虚拟机管理器发送命令进行优化虚拟机的部署(图 2 中的箭头 3)。虚拟机管理器(VMMs)进行实际的虚拟机迁移操作、调整虚拟机大小(图 2 中箭头 5)。用户通过调整全局管理器的配置来更改数据中心的节能策略(图 2 中箭头 1)。

## 2.2 虚拟机部署策略

虚拟机部署可以抽象成一个装箱问题[8],其中物理主机代表箱子,虚拟机代表待装箱的物品。与经典的装箱问题相比,该问题中的箱子的大小是可变的,箱子的大小是用计算节点的 CPU 计算能力、内存大小、网络带宽三个变量进行刻画,同样的,代表物品的虚拟机也用这三个变量进行描述其大小。由于虚拟机部署策略的主要目标是为了节能,所有将物品装进箱子里的花费就是物理节点上主机的电能消耗大小。由于装箱问题是一个 NP 难问题,为了解决虚拟机部署问题,我们使用了经过改进的 BFD (Best Fit Decreasing )算法。该算法将虚拟机按照大小进行降序排序,然后将虚拟机分配到由于分配导致电能消耗最小的物理主机上。该算法允许主机异构的情况下,优先选择最节能的物理主机进行虚拟机的放置。

改进的 BFD 算法如下表所示:

算法名称	改进的 BFD 算法
算法输入	主机列表 hostList, 虚拟机列表 vmList
算法输出	虚拟机的部署结果 allocation
vmList.sortDecreasingBySize()	

```

for vm in vmList do
    minPower = MAX
    allocatedHost = NULL
    for host in hostList do
        if host has enough resources for vm then
            power = estimatePower(host, vm)
            if power < minPower then
                allocatedHost = host
                minPower = power
    if allocatedHost != NULL then
        allocation.add(vm, allocatedHost)
return allocation

```

## 2.3 虚拟机节能整合策略

### 2.3.1 策略框架

我们提出的虚拟机节能整合策略根据资源使用率的历史信息进行动态的虚拟机整合。该节能整合分为四个步骤：

- 1)、决定何时一个物理主机过载，需要迁移其上的一个或者多个虚拟机；
- 2)、决定何时一个物理主机欠载，需要迁移其上的所有的虚拟机；
- 3)、选择过载的物理主机上的需要迁移的一个或者多个虚拟机；
- 4)、为待迁移的虚拟机寻找新的物理主机进行放置。

我们的策略可以分为两个阶段，第一个阶段算法遍历所有的物理主机，寻找是否有物理主机过载，如果有物理主机过载，算法应用虚拟机选择策略选择一个或者多个需要迁移的虚拟机，一旦找到虚拟迁移的虚拟机后，使用虚拟机放置算法，为这些需要迁移的虚拟机寻找新的放置的物理主机。第二个阶段，寻找欠载的物理主机，并为其上的所有虚拟机寻找新的放置位置。策略最终返回所有需要迁移的虚拟机的迁移信息（迁移的虚拟机和它们的迁移目的物理主机）。该算法的时间复杂度为  $2N$ ， $N$  为物理主机的个数。

节能整合策略描述如下：

算法名称	节能整合算法 optimizationConsolidation
算法输入	主机列表 hostList, 虚拟机列表 vmList
算法输出	虚拟机的迁移信息 List<map<String, Object>>
<pre> /* 阶段 1 */ <b>for</b> host <b>in</b> hostList <b>do</b>     <b>if</b> isHostOverLoad(host) == true <b>then</b>         vmsToMigrate.add(getVmsToMigrationFromOverloadHost(host)) migrationMapDueToOverLoad = getNewPlacement(vmsToMigrate) migrationMap.add(migrationMapDueToOverLoad) vmsToMigrate.clear()  /*阶段 2*/ <b>for</b> host <b>in</b> hostList <b>do</b>     <b>if</b> isHostUnderLoad(host) == true <b>then</b>         vmsToMigrate.add(host.getVmList)         migrationMap.add(getNewPlacement(vmsToMigrate))         vmsToMigrate.clear() <b>return</b> migrationMap  UnusedHosts = getUnusedHosts() shutdownOrChangeToLowMode(UnusedHosts) </pre>	

在上面的策略中，使用了第二节介绍的虚拟机放置算法。还使用了虚拟机选择策略、物理主机过载检测算法和物理主机欠载检测算法，下面具体描述这些算法的细节以及伪代码。

### 2.3.2 虚拟机选择策略

当物理主机过载时，即物理主机的 CPU 利用率超过上限阈值时，需要将其上的一个或者一些虚拟机迁移走，来减低物理主机的 CPU 利用率，使其 CPU 利用率处在上限阈值之下。而虚拟机迁移也是有一定的代价的，该策略就是如何在物理主机上选择需要迁移的虚拟机。我们所提的策略的目标是：最小化迁移时间

和最小化迁移虚拟机的个数。

虚拟机选择策略描述如下：

算法名称	最小化迁移时间和最小化迁移次数的虚拟机选择策略 getVmsToMigrateFromHosts
算法输入	主机列表 hostList（即所有过载的物理主机）
算法输出	需要迁移的虚拟机列表 vmMigrationList
<pre> <b>for</b> host <b>in</b> hostList <b>do</b>     vmList = host.getVmList()     vmList.sortDecreasingBySize()     hUtil = h.getUtil()     bestFitUtil = MAX     <b>while</b> hUtil &gt; UPPER_THRESHOLD <b>do</b>         <b>for</b> vm <b>in</b> vmList <b>do</b>             <b>if</b> vm.getUtil() &gt; hUtil - UPPER_THRESHOLD <b>then</b>                 t = vm.getUtil() - hUtil + UPPER_THRESHOLD                 <b>if</b> t &lt; bestFitUtil <b>then</b>                     bestFitUtil = t;                     bestFitVm = vm             <b>else</b>                 <b>if</b> bestFitUtil = MAX <b>then</b>                     bestFitVm = vm                 hUtil = hUtil - bestFitVm.getUtil()                 vmMigrationList.add(bestFitVm)                 vmList.remove(bestFitVm)             <b>if</b> hUtil &lt; LOWER_THRESHOLD <b>then</b>                 vmMigrationList.add(host.getVmList())                 vmList.remove(host.getVmList())         <b>return</b> vmMigrationList </pre>	

### 2.3.3 物理主机过载检测算法

在进行节能整合的过程中，需要判断物理主机是否过载。检测物理主机过载

的方法就是看其 CPU 利用率是否查过上限阈值。而物理主机的上限阈值是根据该主机的历史信息进行动态调整的。

过载检测算法如下：

算法名称	主机过载检测算法 isHostOverLoad
算法输入	要检测的物理主机 host
算法输出	是否过载，过载返回 true，否则返回 false
<pre> hostUtilHistory = getHostUtilHistory(host) upper_threshold = estimateUpperThreshold(hostUtilHistory) vmList = host.getVmList() <b>for</b> vm <b>in</b> vmList <b>do</b>     totalRequestMips += vm.getCurrentRequestMips() utilization = totalRequestMips / host.getTotalMips() <b>if</b> utilization &gt;= upper_threshold <b>then</b>     <b>return</b> true <b>else</b>     <b>return</b> false         </pre>	

#### 2.3.4 物理主机欠载检测算法

寻找数据中心中欠载的物理主机，然后将其上的虚拟机都迁移走，将其关停或者切换到低功耗状态，可以减少数据中心的能耗。我们提出的欠载检测算法是一个非常简单的算法。算法除过过载的物理主机和有虚拟机迁入的物理主机，在其他物理主机中寻找 CPU 利用率最小的物理主机，并且尝试着将其上的所有虚拟机迁移走，如果并非所有的虚拟机都可以成功的迁移走，那么该台物理主机应该保持现有的状态，算法结束。如果所有的虚拟机可以成功的迁移走，那么就将该台物理主机关机或者切换到低功耗状态。这个过程可以迭代进行，直到不能将所有的虚拟机迁移走为止。

欠载检测算法如下：

算法名称	主机欠载检测算法 isHostUnderLoad
------	--------------------------

算法输入	要检测的物理主机列表 Host
算法输出	是否欠载，过载返回 true，否则返回 false
<pre> hostList = getHostList() hostList.sortDecreasingByUtilization() <b>for</b> host <b>in</b> hostList <b>do</b>     <b>if</b> getNewPlacement(host.getVmList()) == true <b>do</b>         <b>if</b> host == Host <b>then</b>             <b>return</b> true         <b>else</b>             <b>return</b> false     <b>else</b>         <b>break</b> <b>return</b> false </pre>	

### 2.3.5 物理主机关机休眠策略

当物理主机关闭时，毫无疑问会节省电能。但是，应该将多少富余的物理主机关闭而不至于当负载上升时违反 SLA，是一个应该考虑的问题。所以，我们设计了以下关机策略：我们在每一次迭代（每隔一定的时间间隔）时将所有的空闲的物理主机中的一部分关闭掉，而另一部分保持开机状态，使用参数  $a$  来决定关闭物理主机的个数。

$$number\ of\ PM\ to\ switch\ off = \frac{number\ of\ Unused\ PMs}{a}$$

这个算法意味着，当空闲物理主机数时常量时，经过若干次迭代后，总有 1 个物理主机保持开机状态，这台物理主机就用来为突发的负载进行服务，从而有效减少 SLA 违反率。

关机策略的详细描述如下：

算法名称	物理主机关机策略 shutdownOrChangeToLowMode
算法输入	空闲物理主机列表 unusedHostList, 表示关闭一部分机器的参数 $a$

算法输出	无输出
<pre> numberOfUnusedHosts = unusedHostList.size() numberOfHostToPoweroff = numberOfUnusedHosts/a while  numberOfHostToPoweroff &gt; 0 do     hostToPoweroff = getTheLeastEnergyEfficientHost(unusedHostList)     poweroff(hostToPoweroff) end </pre>	

当数据中心负载增加时，我们需要将关闭了物理主机重新开机，来满足负载的需要。在具体先择哪一台物理主机进行开机和关机的上，我们的策略总是选择能效最低的物理主机进行关机，选择能效最高的物理主机进行开机。

## 2.4 算法评估的指标

### 2.4.1 SLA 违反率

在整个节能整合策略的运行过程中，云计算服务提供商应尽可能满足用户签订 SLA[9]，为了评估节能整合算法的有效性，需要定义跟负载不相关的指标来反映违反 SLA 的情况。我们定义了一下两个指标来反映 SLA 的违反情况。

1) 平均违反 SLA 时间(SLA Mean Time)。

$$SLA_{MT} = \frac{1}{N} \sum_{i=1}^N \frac{T_{s_i}}{T_{a_i}}$$

其中 $N$ 是物理主机的个数， $T_{s_i}$ 是主机 $i$  CPU 利用率维持在 100%的时间总和， $T_{a_i}$ 是主机 $i$ 保持开机状态的时间总和。

2) 由于迁移导致的平均性能下降 (Performance Degradation due to Migration)

$$PDM = \frac{1}{M} \sum_{j=1}^M \frac{C_{d_j}}{C_{r_j}}$$

其中 $M$ 是虚拟机的个数， $C_{d_j}$ 是虚拟机 $j$  由于迁移导致的性能下降估计， $C_{r_j}$ 是虚拟机 $j$ 请求得 CPU 处理能力总和。

由于 $SLA_{MT}$ 和PDM是两个独立的反应 SLA 违反率的指标，并且期望他们的值越

小越好。所以我们使用了他们的乘积作为一个综合的指标来反映 SLA 违反率。

$$SLA = SLA_{MT} * PDM$$

#### 2.4.2 耗电量

在节能整合策略中，反映算法有效的最直接的指标就是看应用了节能整合策略后数据中心的耗电量是否降低以及降低的程度。

#### 2.4.3 迁移次数

由于虚拟机迁移[10]是有一定的代价的，我们应该尽可能减少迁移次数，所以迁移次数也是一个反应算法有效性的指标[11]。

### 2.5 模拟仿真软件实现

由于已有的云计算模拟仿真软件 CloudSim 的最小单位是对云服务(Cloudlet)进行建模，它无法以虚拟机为最小单位来建模进行模拟仿真。所以本论文基于 CloudSim 软件框架，设计实现了一个支持以虚拟机为最小单位的云计算平台模拟仿真软件，利用该仿真软件，可以方便的对虚拟机整合策略进行建模，从而验证策略有效性和节能效果。

#### 2.5.1 核心框架

所设计的软件沿用 CloudSim 的事件管理框架。图 3 显示了类图的核心部分，以下是相关类：

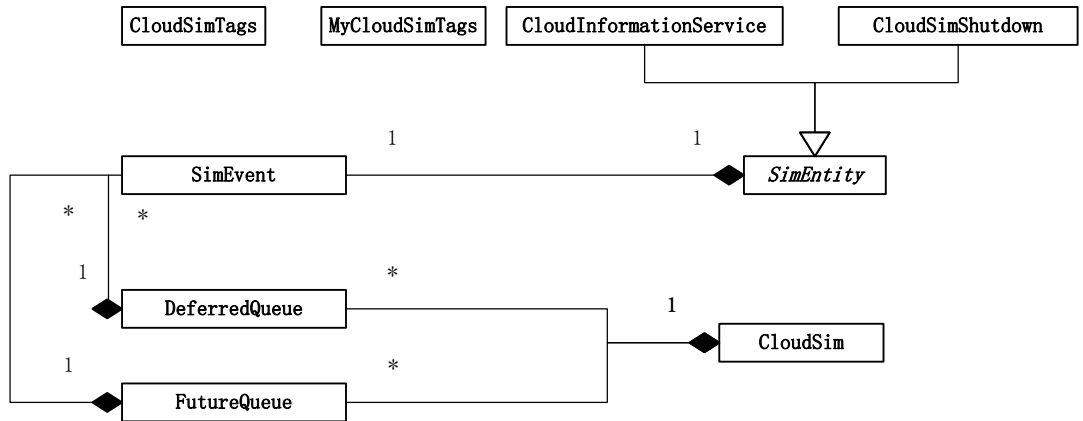




图 3: CloudSim 核心类图

CloudSim: 这是主类, 主要负责管理事件队列和仿真事件的按步骤执行。每一个 CloudSim 实体在运行时刻生成的实体都被存储在称为未来事件 (Future event) 队列中。这些事件按照事件和参数排序并插入到队列中。然后, 仿真中的每步被调度事件从 Future event 队列中移除并且被转移到延时事件 (Defer event) 队列中。接着, 每一个实体激发一个事件处理过程, 主要是从延时队列中选择事件和执行适宜的动作。

DeferredQueue: 该类实现了供 CloudSim 使用的延时事件队列。

FutureQueue: 该类实现供 CloudSim 使用的未来事件队列。

CloudInformationService: CIS 是一个提供了注册、索引、发现资源能力的实体。

SimEntity: 这是一个抽象类, 复杂发送消息给其他实体以及处理接收的消息, 如放弃事件或者处理事件, 所有实体都必须扩展该类和重写它的三个核心方法: `startEntity()`, `processEvent()` 和 `shutdownEntity()`, 它们分别用来实体的初始化、事件的处理和实体的销毁等行为。

CloudSimTags 和 MyCloudSimTags: 这两个类包含了多个静态事件和命令标签, 用来指出当接收或发送事件时需要由 CloudSim 实体承担的行为类型。

SimEvent: 该类描绘了两个或者多个实体之间传送仿真事件的过程。SimEvent 存储了一个事件的如下信息: 事件类型、初始时间、事件发生时间、完成时间。

CloudSimShutdown: 该实体类主要是等待所有终端用户和代理实体的结束, 然后发送仿真结束信号给 CIS。

Predicate: 该类是用于从延时队列中选择事件, 是一个抽象类, 必须创建一个新类来扩展它。图 4 展示了一些标准的 predicate 类。

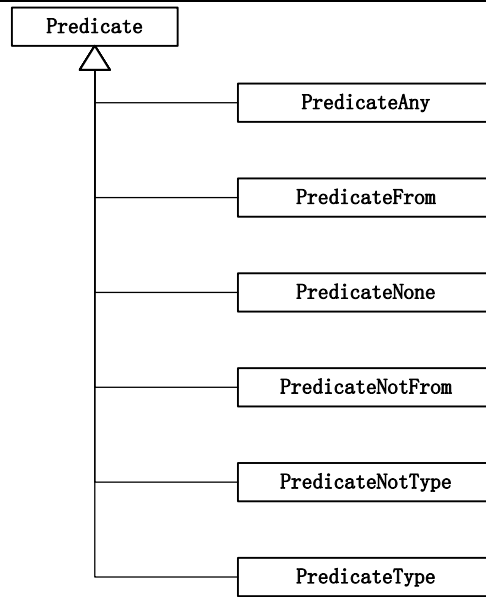


图 4: CloudSim 的核心类 predicate

PredicateAny: 该类描述了从延时队列中匹配任意事件的 predicate 类。

PredicateFrom: 该类描述了从延时队列中匹配来自某个实体的事件的 predicate 类。

PredicateNotFrom: 该类描述了从延时队列中放弃来自给某个实体的事件的 predicate 类。

PredicateNone: 该类描述了从延时队列中不匹配任意事件的 predicate 类。

PredicateType: 该类指的是从匹配特殊标签中选择事件的 predicate 类。

PredicateNotType: 该类指的是从不匹配特殊标签中选择事件的 predicate 类。

### 2.5.2 数据中心事件处理

各个任务单元流程都是由各自的虚拟机处理的。因此它们在每一仿真步骤中的进展都是必须连续更新和监测的。因此，在每一仿真步骤中，每个数据中心实体都激发一个称为 `updateVmsProcessing()` 的方法来以便管理每一个物理主机。按照此流程，相互联系的污泥及可以实时更新主机当前活动的任务流程。该方法的输入参数是当前仿真事件，返回类型为一个任务单元预计完成的时间。下次内

部时间指由主机返回的所有时间中的最短时间。

在主机层面上, `updateVmsProcessing()` 方法触发了 `updateVmProcessing()` 方法, 用来指导每一个虚拟机更新他们在数据中心实体中的任务单元状态(结束、悬挂、执行)。图 5 以顺序图的形式描述了该过程。

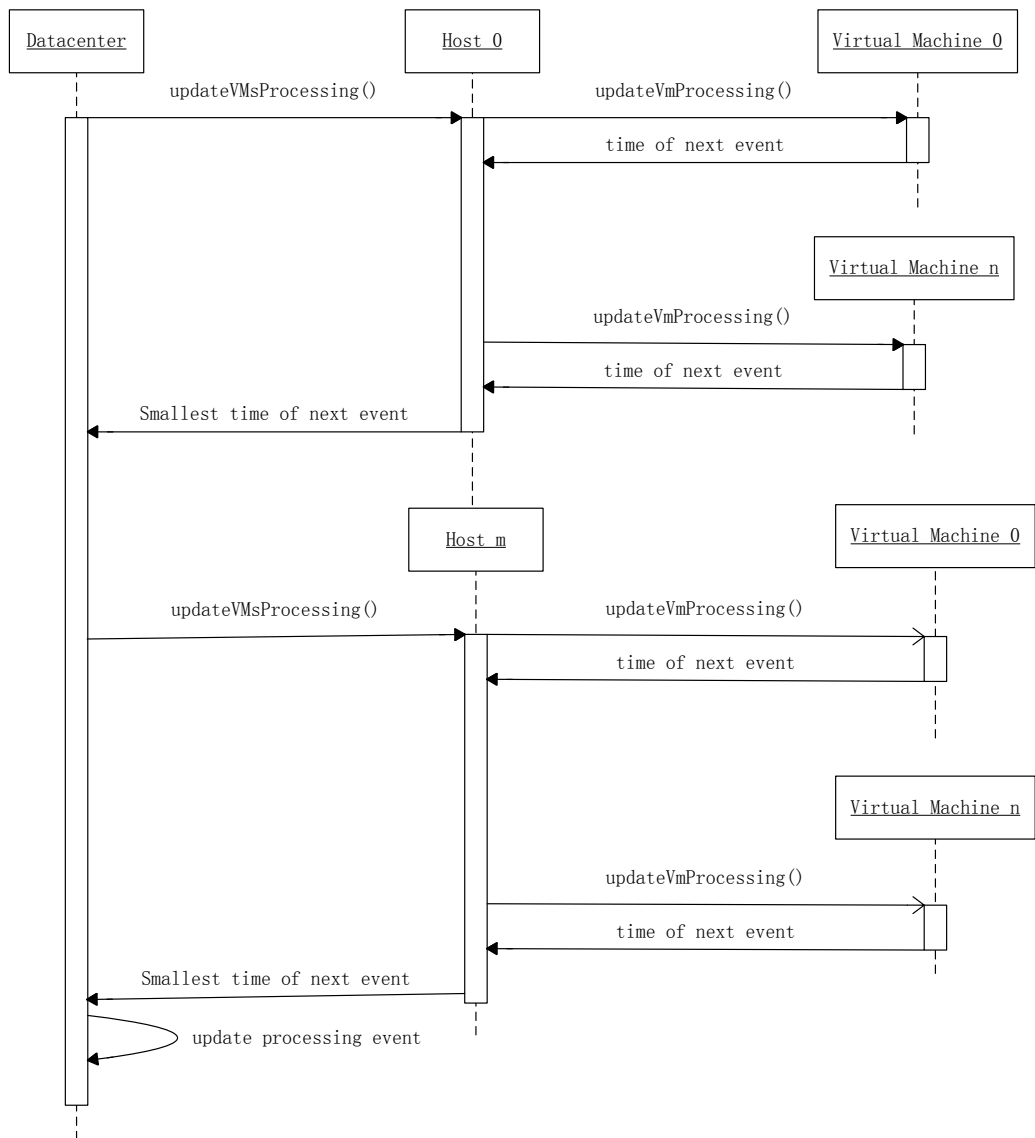


图 5 数据中心事件处理顺序图

### 2.5.3 实体之间的通讯

图 6 描述了核心的 CloudSim 实体之间的通信流量。在仿真开始时, 每个数

据中心实体在 CIS 注册表中注册，然后 CIS 提供信息注册类型功能。如为用户请求匹配合适的云服务商和合适的云服务。下一步，数据中心代理器扮演着用户角色咨询 CIS 服务来获得可提供基础设施并满足应用 QoS、硬件和软件要求的云服务提供商列表。然后代理实体给数据中心实体发送创建虚拟机的事件，接着调度事件来进行模拟，最后在模拟结束后，销毁所创建的虚拟机，关闭各个实体。

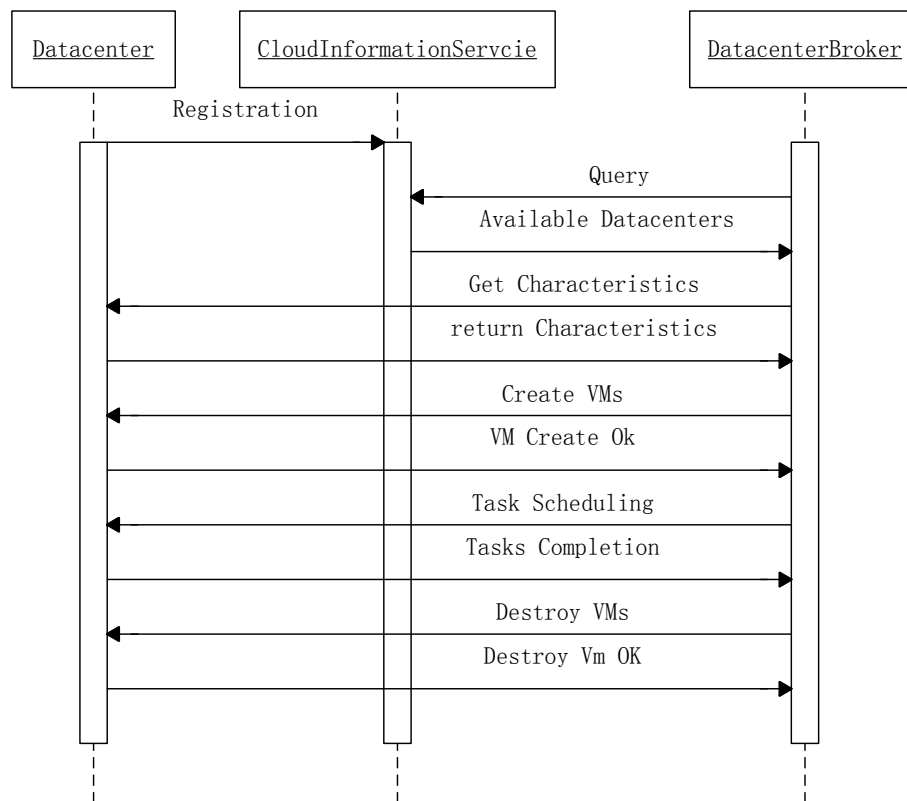


图 6: 核心的 CloudSim 实体之间的通信流

### 3 关键技术或难点

已经完成的关键技术或技术难点的说明如下所示:

#### 3.1 虚拟机节能整合模型

本论文的一个难点是对虚拟机节能整合进行建模，通过建立的模型，才能提出恰当的算法策略。本论文针对 IAAS 平台，进行深入地研究，得出如下模型:

在 IAAS 平台中，资源都是虚拟化出来的，应用程序都是运行在虚拟机之上的。虚拟机服务器整合可以抽象成装箱问题，其中物理服务器可以看作箱子，虚拟机服务器可以看作是待装箱的物品，目标是使用最少的箱子（服务器）来容纳所有的物品（虚拟机）。经典的装箱问题是一个 NP-Hard 问题，没有最优解，只有一些近似的算法的来求次优解，比如 FF（First Fit），BF（Best Fit），NF（Next Fit）和 FFD（First Fit Decreasing）等。而虚拟机整合这个问题和经典的装箱问题还有些不同，在对虚拟机服务器整合进行建模时，首先需要了解虚拟机服务器整合和经典的装箱问题的不同之处[12]，主要有以下四点：

第一，经典的装箱问题中箱子的大小和物品的大小都是用一个整数来描述的，而虚拟机整合中，物理服务器和虚拟机服务器都是使用多个变量进行描述的，比如，CPU 处理能力、内存大小、磁盘大小、网络带宽、磁盘 IO 等。由于在物理服务器和虚拟机服务器都是使用多个变量进行刻画的，是多维度的装箱问题。所以就需要考虑矩形装箱（rectangle packing）和矢量装箱（vector packing）的区别。矩形装箱是将一组  $n$  维对象装到一个  $n$  维的箱子里，这些  $n$  维对象必须装到箱子里，并且不能相互交叉。而在矢量装箱问题中， $n$  个维是相互独立的，装进箱子里的一组对象中每一个维度量值之和都不能超过箱子的对应的维度的量值。当  $n=2$  时，可以用下图（图 7）来清楚的描述两种装箱问题的区别。我们所研究的虚拟机整合属于矢量装箱问题（vector packing）。

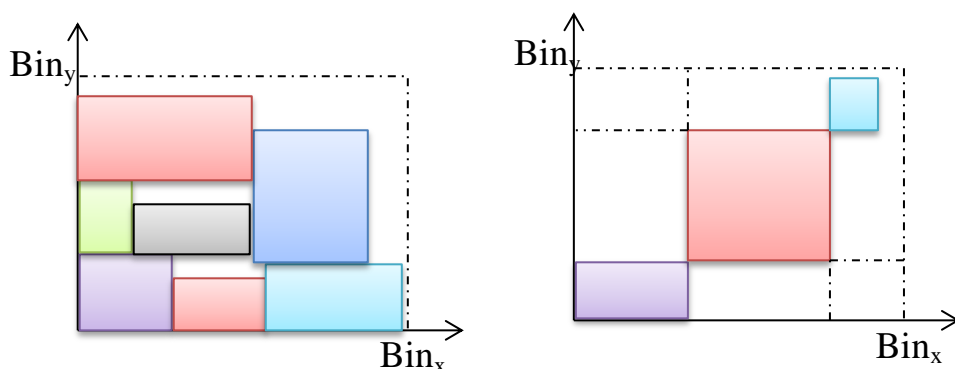


图 7 矩形装箱和矢量装箱的区别

第二，经典的装箱问题中箱子的大小都是一样的，而虚拟机整合中，箱子（物理服务器）的大小有可能是相同的，也有可能是不同的，这主要是因为异构的物理服务器的性能不同。

第三，初始状态不同，经典的装箱问题尝试着用最少的箱子来容纳所有的物品，初始状态所有的箱子都是空的。而虚拟机整合中，虚拟机已经加载到物理主机里，即初始状态所有的箱子不都是空的。

第四，有两种类型的装箱算法，online 和 offline。Online 算法中，不知道所有待装物品的信息，只知道下一个待装物品的信息。而在 offline 算法中，在装箱之前，所以待装物品的信息都可以获取到。在虚拟机服务器整合中，我们使用 offline 算法，所有物理主机和虚拟机的容量大小信息都可以作为算法的输入。

### 3.2 节能整合策略

本论文的另一个难点是在提出节能整合策略时，需要考虑很多约束。因为将虚拟机整合到尽可能少的物理主机上，必然导致一个物理主机上放置数目较多虚拟机，这些虚拟机共同竞争物力资源，这就可能导致服务提供商不能满足承诺给客户的资源请求。即要保证节能整合策略对 SLA 的影响较小。此外，虚拟机迁移也会导致性能下降和能耗的增加，所以算法策略应该尽可能减少虚拟机的迁移次数。

### 3.3 算法策略的验证

由于在真实的云计算环境中对算法策略进行验证存在一定的难度，所以，本论文中提出的算法的有效性验证和节能效果验证也是一个难点。

本论文为了验证算法的有效性和节能效果，作者基于云计算模拟仿真软件 CloudSim 开发了一个专门针对 IAAS 云计算平台的云计算模拟仿真软件，利用该模拟仿真软件，可以对已有的节能整合算法和本论文中提出的节能整合算法进行模拟仿真，并验证其节能效果。

## 4 下一阶段工作计划

针对论文研究计划，指出尚未完成的工作内容、到目前为此尚未解决的关键

### 4.1 存在的问题

存在的一个问题是在进行模拟仿真时，IAAS 数据中心的负载数据很难获取，目前，在对已有算法进行验证时，采用的负载数据来自开源项目 CoMon 中的数据。

### 4.2 尚未完成的工作

对本论文中提出的算法策略使用模拟仿真软件进行验证其有效性和节能效果。并将其与已有的节能算法的节能效果进行比较，从而说明本论文提出的算法策略具有更好的节能效果。

### 4.3 解决问题的技术思路或措施

对于 4.2 中尚未完成的算法模拟仿真，本论为拟采取模拟仿真的方法对其进行验证，并根据模拟仿真结果对节能整合算法策略进行适当修正，迭代以上过程，直到算法策略具有更好的节能效果为止。

### 4.4 下一阶段计划

2013 年 9 月——10 月	利用模拟仿真软件，对论文前期提出的节能功能整合策略进行模拟仿真，验证其有效性，并分析其节能效果，根据仿真结果，对策了进行适当的修正。
2013 年 11 月	撰写大论文
2013 年 12 月	修改大论文，准备毕业答辩

## 5 主要参考文献

- [1] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. A view of cloud computing. *Commun. ACM*, 53(4):50–58, April 2010.
- [2] Jonathan G. Koomey. Estimating total power consumption by servers in the u.s. and the world. Technical report, Stanford University, 2007.
- [3] Jonathan G. Koomey. Growth in data center electricity use 2005 to 2010. Technical report, Oakland, CA: Analytics Press. August 1.  
<http://www.analyticspress.com/datacenters.html>, 2011.
- [4] U. S. Environmental Protection Agency. Report to congress on server and data center energy efficiency public law 109-431. Technical report, U. S. Environmental Protection Agency, 2007.
- [5] Jyun-Shiung Yang, Pangfeng Liu, and Jan-Jan Wu. Workload characteristics-aware virtual machine consolidation algorithms. In *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, pages 42–49, 2012.
- [6] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose, and Rajkumar Buyya. Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software - Practice and Experience*, 41(1):23–50, 2011.
- [7] Sina Esfandiarpour, Ali Pahlavan, and Maziar Goudarzi. Virtual machine consolidation for datacenter energy improvement. *CoRR*, abs/1302.2227, 2013.
- [8] Eric W Weisstein. "bin-packing problem." from mathworld—a wolfram web resource. <http://mathworld.wolfram.com/Bin-PackingProblem.html>, 3 2013.
- [9] Anton Beloglazov, Jemal Abawajy, and Rajkumar Buyya. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems*, 28(5):755–768, 2012.
- [10] Christopher Clark, Keir Fraser, Steven H, Jacob Gorm Hansen, Eric Jul, Christian Limpach, Ian Pratt, and Andrew Warfield. Live migration of virtual machines. In *In Proceedings of the 2nd ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 273–286, 2005.
- [11] Richa Sinha, Nidhi Purohit, and Hiteshi Diwanji. Power aware live migration for data centers in cloud using dynamic threshold. *International Journal of Computer*



*Technology and Applications*, 02(06):2041–2046, 2011.

[12] Aziz Murtazaev and Sangyoon Oh. Sercon: Server consolidation algorithm using live migration of virtual machines for green computing. *IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India)*, 28(3):212–231, 2011.