(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 105159752 B
(45)授权公告日 2018.03.30

(21)申请号 201510607539.8

(22)申请日 2015.09.22

(65)同一申请的已公布的文献号
申请公布号 CN 105159752 A

(43)申请公布日 2015.12.16

(73)专利权人 中国人民解放军国防科学技术大学
地址 410073 湖南省长沙市开福区德雅路109号

(72)发明人 朱晓敏 陈黄科 邱涤珊 李志猛 祝江汉 马满好

(74)专利代理机构 国防科技大学专利服务中心 43202
代理人 郭敏

(51)Int.Cl.
G06F 9/455(2006.01)
G06F 9/48(2006.01)

(56)对比文件
CN 103605567 A,2014.02.26,
CN 103677990 A,2014.03.26,
CN 102541651 A,2012.07.04,
US 2014/0101663 A1,2014.04.10,
CN 103152414 A,2013.06.12,
CN 103414784 A,2013.11.27,
CN 103957231 A,2014.07.30,
CN 105159752 A,2015.12.16,
陈超等.基于滚动优化的虚拟云中实时任务节能调度方法.《软件学报》.2015,第26卷(第8期),第2111-2123页.

审查员 许光华

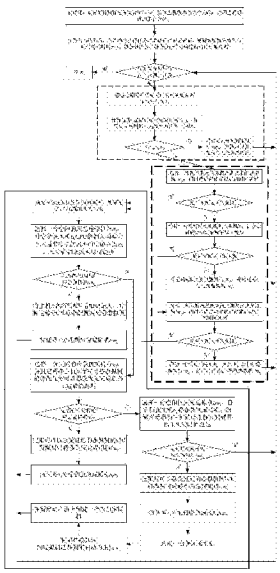
权利要求书5页 说明书8页 附图3页

(54)发明名称

虚拟化云中机器启动时间感知的实时任务与资源调度方法

(57)摘要

本发明公开了一种虚拟化云中机器启动时间感知的实时任务与资源调度方法,目的是减缓启动主机和创建虚拟机的时间开销对突增任务时效性的冲击,提高任务的完成率,减少云服务系统的能耗。技术方案是在每台启动的主机上放置一台空闲的应急虚拟机;当新任务到达时,取消所有等待任务与虚拟机的映射关系,并将新任务加入到等待任务队列中;然后按照截止期最早优先的原则逐个调度等待任务,即先将任务映射到已有的虚拟机,若不可行,则增加新虚拟机来执行任务,若还不可行,则增加空闲应急虚拟机的CPU频率来执行任务。采用本发明可转移启动主机和创建虚拟机的延迟对截止期较短任务的影响,尽可能地提高任务的调度成功率。



CN 105159752 B

1.一种虚拟化云中机器启动时间感知的实时任务与资源调度方法，其特征在于包括以下步骤：

第一步，初始化：

1.1设置等待执行任务集合WT为空，即$WT=\varnothing$；

1.2令活跃主机的集合为H_a，令关闭主机的集合为H_o；

1.2.1令j＝1；

1.2.2如果j＞m，m为主机数量，m为正整数，转第二步；否则，执行第1.2.3步；

1.2.3如果$l_j^t=1$，则$H_a=H_a\cup h_j$；如果$l_j^t=0$，则$H_o=H_o\cup h_j$；$l_j^t\in\{0,1\}$，若$l_j^t=1$表示主机h_j在时刻t处于活跃状态，$l_j^t=0$表示主机h_j在时刻t处于关闭状态；主机$h_j\in H$，$1\leqslant j\leqslant m$，表示云服务系统中的第j台主机，主机的集合$H=\{h_1,h_2,\cdots,h_m\}$；

1.2.4j＝j+1，执行第1.2.2步；

第二步，当新任务动态到达后，更新系统的状态：

2.1将新到达任务加入到等待执行任务集合中，即$WT=WT\cup T_{new}$，其中，$T_{new}=\{t_1',t_2',\cdots,t_{|T_{new}|}'\}$表示新到达的任务集合，其中$|T_{new}|$为新到达任务的数量，$t_i'\in T_{new}$，$1\leqslant i\leqslant|T_{new}|$，为新到达任务集合中的编号为i的任务；

2.2取消等待执行任务与虚拟机之间的映射关系，即令$x_{i,jk}=0$，$\forall t_i\in WT$，映射任务t_i到非应急虚拟机vm_{jk}表示让非应急虚拟机vm_{jk}执行任务t_i；变量$x_{i,jk}$表示任务t_i是否被分配到非应急虚拟机vm_{jk}，如果任务t_i被分配到非应急虚拟机vm_{jk}，则$x_{i,jk}=1$，如果任务t_i没有被分配到非应急虚拟机vm_{jk}，则$x_{i,jk}=0$；任意一个任务t_i表示为$t_i=\{a_i,l_i,d_i\}$，$1\leqslant i\leqslant n$，其中a_i为任务t_i的到达时间，l_i为任务t_i的执行长度，单位为Hz，d_i为任务t_i的截止期，n为正整数；

2.3更新每台非应急虚拟机vm_{jk}的就绪时间rt_{jk}，如果非应急虚拟机vm_{jk}正在执行某个任务，vm_{jk}的就绪时间就是虚拟机完成正在执行任务的时间；如果非应急虚拟机vm_{jk}处于空闲状态，vm_{jk}的就绪时间为当前时间；

2.4更新每台活跃主机h_j上应急虚拟机lvm_j的CPU频率分配，$h_j\in H_a$；

第三步，按任务截止期最早优先的顺序，从WT中选出截止期d_i最早的任务t_i；

第四步，将任务t_i分配到可用的非应急虚拟机上，方法是：

4.1令selVmSet为一个虚拟机集合，初始化为$selVmSet=\varnothing$；minFt为一个变量，初始化为minFt＝+∞；

4.2令j＝1；$|H_a|$表示活跃主机的数量；

4.3如果j＞$|H_a|$，则转第4.4步；否则，执行第4.3.1步；

4.3.1令k＝1；$|VM_j|$表示主机h_j上非应急虚拟机的数量；

4.3.2如果k＞$|VM_j|$，则j＝j+1，转第4.3步；否则，执行第4.3.3步；

4.3.3计算任务t_i在虚拟机vm_{jk}上的完成时间$ft_{i,jk}$，计算公式如下：

$$ft_{i,jk}=rt_{jk}+\frac{l_i}{c(vm_{jk})} \qquad (3)$$

其中，$\frac{l_i}{c(vm_{jk})}$为任务t_i在虚拟机vm_{jk}上的执行时间；

4.3.4如果任务t_i在虚拟机vm_{jk}上的完成时间$ft_{i,jk}$满足$ft_{i,jk}<minFt$，那么，清空虚拟机集合selVmSet即令$selVmSet=\varnothing$，并将虚拟机vm_{jk}加入到selVmSet中；令$minFt=ft_{i,jk}$；$k=k+1$，转第4.3.2步；

4.4如果任务t_i的最小完成时间满足它的时效性要求即$minFt\leq d_i$，取出虚拟机集合selVmSet中的虚拟机，记为selVm；

4.5将任务t_i映射到虚拟机selVm上即令$x_{i,jk}=1$，同时更新虚拟机selVm的就绪时间rt_{jk}为$rt_{jk}=minFt$，转第八步；否则，执行第五步；

第五步，增加一台新虚拟机，并分配任务t_i到新增加的虚拟机上，方法是：

5.1确定新增加虚拟机的CPU频率f_{new}，计算公式如下：

$$f_{new}=\frac{l_i}{d_i-ct-ct(vm)} \qquad (4)$$

其中，l_i和d_i分别表示任务t_i的执行长度和截止期；ct为当前时刻；ct(vm)为创建虚拟机的时间延迟；

5.2令selHostSet为一个主机集合，初始化为$selHostSet=\varnothing$；minRemFre为一个变量，初始化为$minRemFre=+\infty$；

5.3寻找一台剩余资源大于f_{new}的活跃主机：

5.3.1令$j=1$；

5.3.2如果$j>|H_a|$，$|H_a|$表示活跃主机的数量，转第5.4步；否则，执行第5.3.3步；

5.3.3计算主机的剩余资源rf_j，如下所示：

$$rf_j=f_j^{max}-\sum_{k=1}^{|VM_j|}c(vm_{jk})-c(lvm_j) \qquad (5)$$

其中，f_j^{max}表示主机h_j的最大频率；$\sum_{k=1}^{|VM_j|}c(vm_{jk})$和c(lvm$_j$)分别表示非应急虚拟机集合和应急虚拟机使用的CPU频率；

5.3.4如果主机的剩余资源rf_j大于新增加虚拟机的CPU频率f_{new}，且主机的剩余资源rf_j小于minRemFre，即$(f_{new}<rf_j)\&(rf_j<minRemFre)$，则清空主机集合selHostSet，然后将主机$h_j$加入到selHostSet，令$minRemFre=rf_j$，更新$j=j+1$，转到第5.3.2步；

5.4如果选择的主机集合selHostSet为非空，即$selHostSet\neq\varnothing$，执行第5.5步；否则，执行第5.6步；

5.5创建新虚拟机，并映射任务t_i到新虚拟机，转第八步；

5.6确定新增加虚拟机的CPU频率f_{new}，计算公式如下：

$$f_{new}=\frac{l_i}{d_i-ct-ct(vm)-st(host)} \qquad (6)$$

其中，l_i和d_i分别表示任务t_i的执行长度和截止期；ct为当前时刻；ct(vm)为创建虚拟机的时间延迟；st(host)为启动主机的时间延迟；

5.7启动一台关闭的主机，然后创建一台新虚拟机：

5.7.1令j＝1；

5.7.2如果j＞|H_o|，|H_o|表示关闭主机的数量，转第六步；否则，转第5.7.3步；

5.7.3如果主机h_j的最大CPU频率f_j^{max}小于新增加虚拟机的CPU频率需求f_{new}，即$f_j^{max} < f_{new}$，则j＝j+1，转第5.7.2步；否则，执行第5.7.4步；

5.7.4启动关闭的主机h_j；

5.7.5将主机h_j加入到活跃主机集合H_a中，即$H_a = H_a \cup h_j$；同时，将主机h_j从关闭主机集合H_o中移除；

5.7.6在主机h_j上创建一台CPU频率为f_{new}的虚拟机，记为vm_{jk}；

5.7.7在主机h_j上创建一台CPU频率为0的应急虚拟机lvm_j；

5.7.8将新建的非应急虚拟机vm_{jk}加入到可用非应急虚拟机集合AVM中；

5.7.9将任务t_i映射到新非应急虚拟机vm_{jk}；

5.7.10更新非应急虚拟机vm_{jk}的就绪时间rt_{jk}，即$rt_{jk}＝d_i$，转第八步；

第六步，将任务t_i分配到应急虚拟机上，方法如下：

6.1确定应急虚拟机的最小CPU频率$f_{lash-up}$，计算公式如下：

$$f_{lash-up} = \frac{l_i}{d_i - ct} \qquad (7)$$

其中，l_i和d_i分别表示任务t_i的执行长度和截止期；ct为当前时刻；

6.2使用主机的剩余资源来增加应急虚拟机的CPU频率，然后将任务映射到应急虚拟机上：

6.2.1令j＝1；

6.2.2如果j＞|H_a|，|H_a|为活跃主机的数量，转第6.3步；否则，转第6.2.3步；

6.2.3如果主机h_j上的应急虚拟机lvm_j为空闲，即$c(lvm_j)＝0$，执行第6.2.4步；否则，更新j＝j+1，转回第6.2.2步；

6.2.4如果主机h_j的剩余CPU频率rf_j不小于应急虚拟机的最小CPU频率$f_{lash-up}$，即$rf_j \geq f_{lash-up}$，执行第6.2.5步；否则，更新j＝j+1，转回第6.2.2步；

6.2.5将主机h_j的剩余CPU频率rf_j分配给应急虚拟机lvm_j，即$c(lvm_j)＝rf_j$，$rf_j＝0$；

6.2.6将任务t_i映射到应急虚拟机lvm_j上，转第八步；

6.3转移非应急虚拟机的CPU频率到应急虚拟机，然后将任务映射到应急虚拟机上：

6.3.1令j＝1；

6.3.2如果j＞|H_a|，转第6.4步；否则，转第6.3.3步；

6.3.3如果主机h_j上的应急虚拟机lvm_j为空闲，即$c(lvm_j)＝0$，执行第6.3.4步；否则，更新j＝j+1，转回第6.3.2步；

6.3.4令k＝1；

6.3.4.1如果k＞|VM_j|，|VM_j|为主机上非应急虚拟机数量，更新j＝j+1，转第6.3.2步；否则，执行第6.3.4.2步；

6.3.4.2如果非应急虚拟机vm_{jk}的CPU频率满足$c(vm_{jk}) \geq f_{lash-up}$，执行第6.3.4.3步；否则，更新k＝k+1，转回第6.3.4.1步；

6.3.4.3令MT_{jk}为映射到非应急虚拟机vm_{jk}的任务集合；

6.3.4.4如果MT_{jk}中每个任务t_i'的截止期与完成时间之差不小于$l_i/c(vm_{jk})$，即d_i'-

$ft_{i',jk} \geq l_i / c(vm_{jk})$，$\forall t_{i'} \in MT_{jk}$，执行第6.3.4.5步；否则，更新k＝k+1，转回第6.3.4.1步；

6.3.4.5转移非应急虚拟机vm_{jk}的CPU频率给应急虚拟机lvm_j，即$c(lvm_j)=c(vm_{jk})$，$c(vm_{jk})=0$；

6.3.4.6将任务t_i映射到应急虚拟机lvm_j上，转第八步；

6.4转移非应急虚拟机的CPU频率到应急虚拟机，然后将任务映射到应急虚拟机，接着启动一台关闭的主机，并将非应急虚拟机迁移到新启动的主机上：

6.4.1令j＝1；

6.4.2如果$j>|H_a|$，则转第七步；否则，执行第6.4.3步；

6.4.3如果主机h_j上的应急虚拟机lvm_j为空闲，即$c(lvm_j)=0$，执行第6.4.4步；否则，更新j＝j+1，转回第6.4.2步；

6.4.4令k＝1；$|VM_j|$为主机上非应急虚拟机数量；

6.4.4.1如果$k>|VM_j|$，更新j＝j+1，则转到第6.4.2步；否则，执行第6.4.4.2步；

6.4.4.2如果非应急虚拟机vm_{jk}的CPU频率满足$c(vm_{jk}) \geq f_{lash\text{-}up}$，执行第6.4.4.3步；否则，更新k＝k+1，转回第6.4.4.1步；

6.4.4.3令MT_{jk}表示映射到非应急虚拟机vm_{jk}的任务集合；

6.4.4.4如果MT_{jk}中每个任务的截止期与完成时间之差满足$d_i-ft_{i,jk} \geq st(host)+mt(vm_{jk})$，$\forall t_i \in MT_{jk}$，$st(host)$和$mt(vm_{jk})$分别表示启动主机和迁移虚拟机的时间，执行第6.4.4.5步；否则，更新j＝j+1，转到第6.4.2步；

6.4.4.5转移非应急虚拟机vm_{jk}的CPU频率给应急虚拟机lvm_j，即$c(lvm_j)=c(vm_{jk})$，$c(vm_{jk})=0$；

6.4.4.6将任务t_i映射到应急虚拟机lvm_j上；

6.4.4.7启动一台最大CPU频率f_j^{max}大于$c(lvm_j)$的关闭主机h_j；

6.4.4.8迁移非应急虚拟机vm_{jk}到刚启动的主机h_j上；

6.4.4.9配置虚拟机vm_{jk}的CPU频率为$c(vm_{jk})=c(lvm_j)$；

6.4.4.10在主机h_j上创建一台CPU频率为0的应急虚拟机lvm_j；

6.4.4.11将主机h_j加入到活跃主机集合H_a中，即$H_a=H_a \cup h_j$；同时，将主机从关闭主机集合H_o中移除，即$H_o=H_o-h_j$；

6.4.4.12转第八步；

第七步，如果任务t_i没有被映射到任何一台虚拟机上，则拒绝任务t_i，即$WT=WT-\{t_i\}$；

第八步，如果待分配任务集合WT中存在未调度任务，则转第三步；否则，执行第九步；

第九步，结束。

2.如权利要求1所述的虚拟化云中机器启动时间感知的实时任务与资源调度方法，其特征在于2.4步所述更新每台活跃主机h_j上应急虚拟机lvm_j的CPU频率分配的方法是如果应急虚拟机lvm_j没有正在执行的任务，则降低应急虚拟机的CPU频率到0，即$c(lvm_j)=0$；否则，不改变应急虚拟机lvm_j的CPU频率。

3.如权利要求1所述的虚拟化云中机器启动时间感知的实时任务与资源调度方法，其特征在于第5.5步所述创建新虚拟机，并映射任务t_i到新虚拟机的方法是：

5.5.1取出selHostSet中的主机，该主机记为selHost；

5.5.2在主机selHost上创建一台CPU频率为f_{new}的虚拟机，记为vm_{jk}；

5.5.3将新建的非应急虚拟机vm_{jk}加入到可用非应急虚拟机集合AVM中；

5.5.4将任务t_i映射到新非应急虚拟机vm_{jk}；

5.5.5更新非应急虚拟机vm_{jk}的就绪时间rt_{jk}，即$rt_{jk}=d_i$。

虚拟化云中机器启动时间感知的实时任务与资源调度方法

技术领域

[0001] 本发明属于计算机软件和云服务系统中任务调度和资源管理技术领域，涉及云计算平台中任务和资源调度方法。

背景技术

[0002] 为了满足急剧增长的计算服务需求，云服务系统中的主机规模不断扩大。一个数据中心的主机数量就高达几万台，甚至几十万台。正常运行这些主机，云服务系统需要消耗大量的电能。据统计，从2005年到2010年全球数据中心的能耗提高了56％，占全球能耗的1.5％。对企业而言，高能耗就意味着高成本。另外，高能耗对生态环境产生较大的负面影响，因为使用煤矿发电会向空气中释放大量的废气。云服务系统的高能耗问题已经引起工业界和学术界的极大关注，并成为学术界研究的热点。

[0003] 大量的研究表明，在云服务系统中，能够提高活跃主机资源的有效利用和减少电能消耗的有效途径是：在云服务系统负载下降的时候，动态整合虚拟机到尽可能少的主机上，然后关闭空闲的主机，以减少能量消耗。由于主机处于完全空闲的状态，功耗仍然是它最大功耗的50％以上，关闭空闲的主机就意味着减少大量的电能消耗。

[0004] 但是，这种资源调度方法带来了另一个具有挑战性的问题：当云服务系统的负载突增时，在伸展虚拟机的过程中，创建新虚拟机或者先开启关闭的主机然后再创建虚拟机都需要一定的时间开销，使得某些任务不能及时开始，从而延误了它们的截止期。例如，一个在0s到达的新任务，它的执行时间是5s，假设它的截止期是执行时间的五倍(即25s)。启动主机的时间大概为30s，创建一台虚拟机的时间近似于启动一个操作系统的时间，大概是30s。当新任务到达后，启动一台主机，然后再创建一台新虚拟机来执行新到达任务，很明显，新到达任务的截止期将被延误。

[0005] 一段时间内，云服务系统需要执行的任务集合可表示为$T=\{t_1,t_2,\cdots,t_n\}$，其中n为正整数。任意一个任务$t_i\in T$可表示为$t_i=\{a_i,l_i,d_i\}$，$1\leqslant i\leqslant n$，其中a_i为任务t_i的到达时间，l_i为任务t_i的计算量(单位为Hz)，d_i为任务t_i的截止期。云服务系统中，主机的集合用$H=\{h_1,h_2,\cdots,h_m\}$表示，主机$h_j\in H$，$1\leqslant j\leqslant m$表示云服务系统中的第j台主机，m为正整数。另外，任意一台主机$h_j\in H$可描述为$h_j=\{m_j,s_j,n_j,p_j^{max},(f_j,v_j)\}$，其中$m_j$、$s_j$、$n_j$和$p_j^{max}$分别表示主机$h_j$的内存大小(GB)、硬盘大小(GB)、网络传输速度(Mb/s)和功耗(W)；$(f_j,v_j)=\{(f_j^1,v_j^1),(f_j^2,v_j^2),\cdots,(f_j^{max},v_j^{max})\}$表示主机$h_j$中CPU主频-电压对的离散集合，max为正整数，表示主机的主机主频-电压对的数量，其中$(f_j^d,v_j^d)\in(f_j,v_j),\ 1\leq d\leq max$表示主机$h_j$的第d对主频-电压，并且$(f_j^1<f_j^2<\cdots<f_j^{max})$，$(v_j^1<v_j^2<\cdots<v_j^{max})$。主机$h_j\in H$可同时运行一个虚拟机集合，表示为$VM_j=\{vm_{j1},vm_{j2},\cdots,vm_{j|VM_j|}\}\cup\{lvm_j\}$，其中下标$|VM_j|$为可变的正整数，表示主机$h_j$上非应急虚拟机的个数；$vm_{jk}\in VM_j$，$1\leqslant k\leqslant|VM_j|$表示主机$h_j$上的第k台虚拟机；$lvm_j$为主机$h_j$上唯一的一台应急虚拟机。对于虚拟机$vm_{jk}$，符号$c(vm_{jk})$、$r(vm_{jk})$和n

(vm_{jk})分别表示分配给该虚拟机的CPU能力、内存和网络带宽。类似地，$c(lvm_j)$、$r(lvm_j)$和$n(lvm_j)$分别表示分配给应急虚拟机lvm_j的CPU能力、内存和网络带宽。

[0006] 朱晓敏等人在期刊《IEEE Transactions on Cloud Computing》上发表的文章《Real-time tasks oriented energy-aware scheduling in virtualized clouds》(虚拟化云中面向实时任务的节能调度方法)提出了EARH算法用于在虚拟化云中调度实时任务到虚拟机上，同时根据云计算数据中心的负载动态增加和减少虚拟机的数量。其中，增加虚拟机的主要步骤是：(1)在已经启动的主机上创建一台新虚拟机；(2)如果步骤1中不能创建新虚拟机，则通过虚拟机动态迁移，把主机的空闲资源汇聚到某台主机上，然后把虚拟机创建在该主机上；(3)如果前两个步骤都不能成功地创建新虚拟机，则开启一台主机，然后在该主机上创建新虚拟机。

[0007] 然而，以上增加虚拟机的方法需要一定的时间开销，使得某些实时任务不能及时开始，从而延误了它们的截止期。

发明内容

[0008] 本发明要解决的技术问题是针对启动主机和创建虚拟机的时间开销，造成实时任务不能及时开始执行，从而延误了一些实时任务截止期的问题，提出机器启动时间感知的任务与资源调度方法，以减缓启动主机和创建虚拟机的时间开销对突增任务时效性的冲击，尽可能提高任务的完成率，同时减少云服务系统的能耗。

[0009] 本发明所指的任务是指元任务，即只需一台虚拟机独立完成的任务。

[0010] 本发明以任务完成率和系统能耗作为任务分配优化目标。

[0011] (1)任务完成率目标：使得f_1的值，即被成功分配的任务数与所有任务数之比尽可能的大。

[0012] $$f_1=\frac{\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{|VM_j|}x_{i,jk}}{n} \qquad (1)$$

[0013] 其中，变量$x_{i,jk}$表示任务t_i是否被分配到虚拟机vm_{jk}。如果任务t_i被分配到虚拟机vm_{jk}，则$x_{i,jk}=1$，如果任务t_i没有被分配到虚拟机vm_{jk}，则$x_{i,jk}=0$。

[0014] (2)能量消耗：使得f_2的值，即云服务系统完成任务的能量消耗最小化。

[0015] $$f_2=\sum_{i=1}^{n}\int_{st}^{et}(k_j\cdot p_j^{max}\cdot l_j^t+\frac{(1-k_j)p_j^{max}}{(f_j^{\max})^3}\cdot(f_j^d)^3)\cdot dt \qquad (2)$$

[0016] 其中st和et分别表示执行任务的开始和结束时间；k_j为主机h_j完全空闲时功耗与最大功耗的比值；$l_j^t\in\{0,1\}$表示主机h_j在时刻t的状态，若$l_j^t=1$指主机h_j在时刻t处于活跃状态，否则，$l_j^t=0$表示主机h_j在时刻t处于关闭状态。

[0017] 本发明具体技术方案包括以下步骤：

[0018] 第一步，初始化：

[0019] 1.1设置等待执行任务集合WT为空，即$WT=\varnothing$；

[0020] 1.2令活跃主机的集合为H_a；关闭主机的集合为H_o；

[0021] 1.2.1令j＝1；

[0022] 1.2.2如果j＞m，m为主机数量，转第二步；否则，执行第1.2.3步；

[0023] 1.2.3如果$l_j^t=1$，即主机h_j处于活跃状态，则$H_a=H_a\cup h_j$；如果$l_j^t=0$，则$H_o=H_o\cup h_j$；

[0024] 1.2.4j＝j+1，执行第1.2.2步；

[0025] 第二步，当新任务动态到达后，更新系统的状态：

[0026] 2.1将新到达任务加入到等待执行任务集合中，即$WT=WT\cup T_{new}$，其中，$T_{new}=\{t'_1,t'_2,\cdots,t'_{|T_{new}|}\}$表示新到达的任务集合，其中$|T_{new}|$为新到达任务的数量，$t'_i\in T_{new}$，$1\leqslant i\leqslant|T_{new}|$为新到达任务集合中的编号为i的任务；

[0027] 2.2取消等待执行任务与虚拟机之间的映射关系，即$x_{i,jk}=0,\ \forall t_i\in WT$。其中，映射任务$t_i$到虚拟机$vm_{jk}$表示让虚拟机$vm_{jk}$执行任务$t_i$；

[0028] 2.3更新每台非应急虚拟机vm_{jk}的就绪时间rt_{jk}。如果虚拟机vm_{jk}正在执行某个任务，那么它就绪时间就是虚拟机完成正在执行任务的时间；否则，虚拟机处于空闲状态，那么它的就绪时间为当前时间；

[0029] 2.4更新每台活跃主机h_j上应急虚拟机lvm_j的CPU频率分配，$h_j\in H_a$。如果应急虚拟机lvm_j没有正在执行的任务，则降低应急虚拟机lvm_j的CPU频率到0，即$c(lvm_j)=0$。否则，不改变应急虚拟机lvm_j的CPU频率；

[0030] 第三步，按任务截止期最早优先的顺序，从WT中选出一个未调度的任务t_i(即任务t_i的截止期d_i最早)；

[0031] 第四步，将任务t_i分配到可用的非应急虚拟机上，方法是：

[0032] 4.1令selVmSet为一个虚拟机集合，初始化为$selVmSet=\varnothing$；minFt为一个变量，初始化为minFt＝+∞；

[0033] 4.2令j＝1；$|H_a|$表示活跃主机的数量；

[0034] 4.3如果j＞$|H_a|$，则转第4.4步；否则，执行第4.3.1步；

[0035] 4.3.1令k＝1；$|VM_j|$表示主机h_j上非应急虚拟机的数量；

[0036] 4.3.2如果k＞$|VM_j|$，则j＝j+1，转第4.3步；否则，执行第4.3.3步；

[0037] 4.3.3计算任务t_i在虚拟机vm_{jk}上的完成时间$ft_{i,jk}$，计算公式如下：

[0038] $$ft_{i,jk}=rt_{jk}+\frac{l_i}{c(vm_{jk})} \qquad (3)$$

[0039] 其中，$\frac{l_i}{c(vm_{jk})}$为任务t_i在虚拟机vm_{jk}上的执行时间；

[0040] 4.3.4如果任务t_i在虚拟机vm_{jk}上的完成时间$ft_{i,jk}$完成时间满足$ft_{i,jk}$＜minFt，那么，清空虚拟机集合selVmSet(即令$selVmSet=\varnothing$)，并将虚拟机vm_{jk}加入到selVmSet中；令minFt＝$ft_{i,jk}$；k＝k+1，转第4.3.2步；

[0041] 4.4如果任务t_i的最小完成时间满足它的时效性要求(即，minFt≤d_i)，取出虚拟机集合selVmSet中的虚拟机，记为selVm；

[0042] 4.5将任务t_i映射到虚拟机selVm上(即令$x_{i,jk}$＝1)，同时更新虚拟机selVm的就绪时间rt_{jk}为rt_{jk}＝minFt，转第八步；否则，执行第五步；

[0043] 第五步，增加一台新虚拟机，并分配任务t_i到新增加的虚拟机上，方法是：

[0044] 5.1确定新增加虚拟机的CPU频率f_{new}，计算公式如下：

[0045] $$f_{new}=\frac{l_i}{d_i-ct-ct(vm)} \quad (4)$$

[0046] 其中，l_i和d_i分别表示任务t_i的执行长度和截止期；ct为当前时刻；ct(vm)为创建虚拟机的时间延迟；

[0047] 5.2令selHostSet为一个主机集合，初始化为$selHostSet=\varnothing$；minRemFre为一个变量，初始化为minRemFre＝+∞；

[0048] 5.3寻找一台剩余资源大于f_{new}的活跃主机：

[0049] 5.3.1令j＝1；

[0050] 5.3.2如果j＞|H_a|，则转第5.4步；否则，执行第5.3.3步；

[0051] 5.3.3计算主机的剩余资源rf_j，如下所示：

[0052] $$rf_j=f_j^{max}-\sum_{k=1}^{|VM_j|}c(vm_{jk})-c(lvm_j) \quad (5)$$

[0053] 其中，f_j^{max}表示主机h_j的最大频率；$\sum_{k=1}^{|VM_j|}c(vm_{jk})$和c($lvm_j$)分别表示非应急虚拟机集合和应急虚拟机使用的CPU频率；

[0054] 5.3.4如果主机的剩余资源rf_j大于新增加虚拟机的CPU频率f_{new}，且主机的剩余资源rf_j小于minRemFre，即($f_{new}\leq rf_j$)&(rf_j＜minRemFre)，则清空主机集合selHostSet(即令$selHostSet=\varnothing$)，然后将主机h_j加入到selHostSet，令minRemFre＝rf_j，更新j＝j+1，转到第5.3.2步；

[0055] 5.4如果选择的主机集合selHostSet为非空，即$selHostSet\neq\varnothing$，表示新增加虚拟机能够创建在活跃主机上，执行第5.5步；否则，执行第5.6步；

[0056] 5.5创建新虚拟机，并映射任务t_i到新虚拟机：

[0057] 5.5.1取出selHostSet中的主机，该主机记为selHost；

[0058] 5.5.2在主机selHost上创建一台CPU频率为f_{new}的虚拟机，记为vm_{jk}；

[0059] 5.5.3将新建的虚拟机vm_{jk}加入到可用非应急虚拟机集合AVM中，即AVM＝AVM∪{vm_{jk}}；

[0060] 5.5.4将任务t_i映射到新虚拟机vm_{jk}；

[0061] 5.5.5更新虚拟机vm_{jk}的就绪时间rt_{jk}，即$rt_{jk}=d_i$，转第八步；

[0062] 5.6确定新增加虚拟机的CPU频率f_{new}，计算公式如下：

[0063] $$f_{new}=\frac{l_i}{d_i-ct-ct(vm)-st(host)} \quad (6)$$

[0064] 其中，l_i和d_i分别表示任务t_i的长度和截止期；ct为当前时刻；ct(vm)为创建虚拟机的时间延迟；st(host)为启动主机的时间延迟；

[0065] 5.7启动一台关闭的主机，然后创建一台新虚拟机：

[0066] 5.7.1令j＝1；

[0067] 5.7.2如果j＞|H_o|，|H_o|表示关闭主机的数量，转第六步；否则，转第5.7.3步；

[0068] 5.7.3如果主机h_j的最大CPU频率f_j^{max}小于新增加虚拟机的CPU频率需求f_{new}，即

$f_j^{max} < f_{new}$，则j＝j+1，转第5.7.2步；否则，执行第5.7.4步；

[0069] 5.7.4启动关闭的主机h_j；

[0070] 5.7.5将主机h_j加入到活跃主机集合H_a中，即$H_a = H_a \cup h_j$；同时，将主机h_j从关闭主机集合H_o中移除，即$H_o = H_o - \{h_j\}$；

[0071] 5.7.6在主机h_j上创建一台CPU频率为f_{new}的虚拟机，记为vm_{jk}；

[0072] 5.7.7在主机h_j上创建一台CPU频率为0的应急虚拟机lvm_j；

[0073] 5.7.8将新建的虚拟机vm_{jk}加入到可用非应急虚拟机集合AVM中，即$AVM = AVM \cup \{vm_{jk}\}$；

[0074] 5.7.9将任务t_i映射到新虚拟机vm_{jk}；

[0075] 5.7.10更新虚拟机vm_{jk}的就绪时间rt_{jk}，即$rt_{jk} = d_i$，转第八步；

[0076] 第六步，将任务t_i分配到应急虚拟机上，方法如：

[0077] 6.1确定应急虚拟机的最小CPU频率$f_{lash-up}$，计算公式如下：

[0078] $$f_{lash-up} = \frac{l_i}{d_i - ct} \qquad (7)$$

[0079] 其中，l_i和d_i分别表示任务t_i的长度和截止期；ct为当前时刻；

[0080] 6.2使用主机的剩余资源来增加应急虚拟机的CPU频率，然后将任务映射到应急虚拟机上：

[0081] 6.2.1令j＝1；$|H_a|$为活跃主机的数量；

[0082] 6.2.2如果$j > |H_a|$，转第6.3步；否则，转第6.2.3步；

[0083] 6.2.3如果主机h_j上的应急虚拟机lvm_j为空闲，即$c(lvm_j) = 0$，执行第6.2.4步；否则，更新j＝j+1，转回第6.2.2步；

[0084] 6.2.4如果主机h_j的剩余CPU频率rf_j(如公式(5)所示)不小于应急虚拟机的最小CPU频率$f_{lash-up}$，即$rf_j \geq f_{lash-up}$，执行第6.2.5；否则，更新j＝j+1，转回第6.2.2步；

[0085] 6.2.5将主机h_j的剩余CPU频率rf_j分配给应急虚拟机lvm_j，即$c(lvm_j) = rf_j$，$rf_j = 0$；

[0086] 6.2.6将任务t_i映射到应急虚拟机lvm_j上，转第八步；

[0087] 6.3转移非应急虚拟机的CPU频率到应急虚拟机，然后将任务映射到应急虚拟机上：

[0088] 6.3.1令j＝1；$|H_a|$为活跃主机的数量；

[0089] 6.3.2如果$j > |H_a|$，转第6.4步；否则，转第6.3.3步；

[0090] 6.3.3如果主机h_j上的应急虚拟机lvm_j为空闲，即$c(lvm_j) = 0$，执行第6.3.4步；否则，更新j＝j+1，转回第6.3.2步；

[0091] 6.3.4令k＝1；

[0092] 6.3.4.1如果$k > |VM_j|$，$|VM_j|$为主机上非应急虚拟机数量，更新j＝j+1，转第6.3.2步；否则，执行第6.3.4.2步；

[0093] 6.3.4.2如果虚拟机vm_{jk}的CPU频率满足$c(vm_{jk}) \geq f_{lash-up}$，执行第6.3.4.3步；否则，更新k＝k+1，转回第6.3.4.1步；

[0094] 6.3.4.3令MT_{jk}为映射到虚拟机vm_{jk}的任务集合；

[0095] 6.3.4.4如果MT_{jk}中每个任务$t_{i'}$的截止期与完成时间之差不小于$l_i/c(vm_{jk})$，即$d_{i'}-ft_{i',jk}\geq l_i/c(vm_{jk}),\ \forall t_{i'}\in MT_{jk}$，表示所有已经映射到虚拟机$vm_{jk}$的任务都可以忍受应急虚拟机$lvm_j$完成任务$t_i$造成的延迟，执行第6.3.4.5步；否则，更新k＝k+1，转回第6.3.4.1步；

[0096] 6.3.4.5转移虚拟机vm_{jk}的CPU频率给应急虚拟机lvm_j，即$c(lvm_j)=c(vm_{jk})$，$c(vm_{jk})=0$；

[0097] 6.3.4.6将任务t_i映射到应急虚拟机lvm_j上，转第八步；

[0098] 6.4转移非应急虚拟机的CPU频率到应急虚拟机，然后将任务映射到应急虚拟机，接着启动一台关闭的主机，并将非应急虚拟机迁移到新启动的主机上：

[0099] 6.4.1令j＝1；$|H_a|$为活跃主机的数量；

[0100] 6.4.2如果j＞$|H_a|$，则转第七步；否则，执行第6.4.3步；

[0101] 6.4.3如果主机h_j上的应急虚拟机lvm_j为空闲，即$c(lvm_j)=0$，执行第6.4.4步；否则，更新j＝j+1，转回第6.4.2步；

[0102] 6.4.4令k＝1；$|VM_j|$为主机上非应急虚拟机数量；

[0103] 6.4.4.1如果k＞$|VM_j|$，更新j＝j+1，则转到第6.4.2步；否则，执行第6.4.4.2步；

[0104] 6.4.4.2如果虚拟机vm_{jk}的CPU频率满足$c(vm_{jk})\geq f_{lash\text{-}up}$，执行第6.4.4.3步；否则，更新k＝k+1，转回第6.4.4.1步；

[0105] 6.4.4.3令MT_{jk}表示映射到虚拟机vm_{jk}的任务集合；

[0106] 6.4.4.4如果MT_{jk}中每个任务的截止期与完成时间之差满足$d_i-ft_{i,jk}\geq st(host)+mt(vm_{jk}),\ \forall t_i\in MT_{jk}$，其中$st(host)$和$mt(vm_{jk})$分别表示启动主机和迁移虚拟机的时间，表示所有已经映射到虚拟机vm_{jk}的任务都可以忍受启动主机和迁移虚拟机造成的延迟，执行第6.4.4.5步；否则，更新j＝j+1，转到第6.4.2步；

[0107] 6.4.4.5转移虚拟机vm_{jk}的CPU频率给应急虚拟机lvm_j，即$c(lvm_j)=c(vm_{jk})$，$c(vm_{jk})=0$；

[0108] 6.4.4.6将任务t_i映射到应急虚拟机lvm_j上；

[0109] 6.4.4.7启动一台最大CPU频率f_j^{max}大于$c(lvm_j)$的关闭主机h_j；

[0110] 6.4.4.8迁移虚拟机vm_{jk}到刚启动的主机h_j上；

[0111] 6.4.4.9配置虚拟机vm_{jk}的CPU频率为$c(vm_{jk})=c(lvm_j)$；

[0112] 6.4.4.10在主机h_j上创建一台CPU频率为0的应急虚拟机lvm_j；

[0113] 6.4.4.11将主机h_j加入到活跃主机集合H_a中，即$H_a=H_a\cup h_j$；同时，将主机从关闭主机集合H_o中移除，即$H_o=H_o-h_j$；

[0114] 6.4.4.12转第八步；

[0115] 第七步，如果任务t_i没有被映射到任何一台虚拟机上，则拒绝任务t_i，即$WT=WT-\{t_i\}$；

[0116] 第八步，如果待分配任务集合WT中存在未调度任务，则转第三步；否则，执行第九步；

[0117] 第九步，结束。

[0118] 采用本发明可以达到如下效果：在满足用户时间要求前提下，提供一种面向应急

的任务分配和虚拟机扩展方法，从而转移启动主机和创建虚拟机的延迟对截止期较短任务的影响，以尽可能地提高任务的调度成功率。

[0119] 1)本发明的第5.7.7和6.4.4.10步，在每台活跃主机上放置一台应急虚拟机，避免创建虚拟机的时间延迟任务的开始时间，提高了任务的调度成功率。

[0120] 2)本发明的第6.3和6.4步采用转移其他虚拟机CPU资源给应急虚拟机的方法，使得任务尽可能早开始执行，从而进一步提高了任务的调度成功率。

附图说明

[0121] 图1是本发明总体流程图。

[0122] 图2是第6.2.4到6.2.6步的示意图。

[0123] 图3是第6.3.4.4到6.3.4.6步的示意图。

[0124] 图4是第6.4.4.4到6.4.4.10步的示意图。

具体实施方式

[0125] 图1是本发明总体流程图，具体包括：

[0126] 第一步，初始化：初始化等待任务集合WT为空；初始化活跃主机集合H_a和关闭主机集合H_o；

[0127] 第二步，任务动态达到后，取消等待执行任务与虚拟机之间的映射关系，更新每台虚拟机vm_{jk}完成已安排任务的时间rt_{jk}，并将新到任务和等待执行任务作为待分配任务集合WT；

[0128] 第三步，按任务截止期(d_i)最早优先的顺序，从WT中选出一个未调度的任务t_i；

[0129] 第四步，将任务分配到已经开启的虚拟机上，如图1中细虚线框包含部分，方法是：

[0130] 4.1将任务t_i与每个已启动的虚拟机vm_{jk}逐一配对，并计算由此产生的每对任务-虚拟机的任务的完成时间$ft_{i,jk}$；

[0131] 4.2从4.1的计算结果中，选出具有最小值的$ft_{i,jk}$；

[0132] 4.3如果任务t_i的最小完成时间满足它的时效性要求(即，$ft_{i,jk} \leq d_i$)，则将任务t_i映射到vm_{jk}虚拟机上(即，$ft_{i,jk} \leftarrow 1$)，同时更新虚拟机vm_{jk}完成已安排任务的时间rt_{jk}，然后转到第八步；否则，转到第五步；

[0133] 第五步将任务分配到新增加的虚拟机上，如图1中粗虚线框包含部分，方法如：

[0134] 5.1选择一种能够在任务截止期内完成任务的虚拟机vm_{jk}，同时考虑创建虚拟机的时间延迟。如果存在这样的虚拟机，则执行5.2步；否则，转第六步。

[0135] 5.2选择一台剩余MIPS尽可能少的活跃主机来创建该虚拟机。如果存在活跃主机满足以上条件，那么创建虚拟机vm_{jk}，然后将任务t_i映射到虚拟机vm_{jk}上，同时更新虚拟机vm_{jk}完成已安排任务的时间rt_{jk}，接着转到第八步；否则，转5.3步；

[0136] 5.3选择一种能够在任务截止期内完成任务的虚拟机vm_{jk}，同时考虑启动主机和创建虚拟机的时间延迟。如果存在这样的虚拟机，则执行5.4步；否则，转第六步。

[0137] 5.4启动一台主机，然后创建新虚拟机vm_{jk}，接着将任务t_i映射到虚拟机vm_{jk}，同时更新虚拟机vm_{jk}完成已安排任务的时间rt_{jk}，再接着转到第八步；

[0138] 第六步，将任务分配到应急虚拟机上，如图1中黑实线框包含部分，方法如：

[0139] 6.1确定应急虚拟机的最小CPU资源，使得任务能在截止期内完成；

[0140] 6.2选择一台空闲的应急虚拟机lvm_j，并且它所在主机h_j的剩余MIPS大于应急虚拟机的最小CPU资源，且主机h_j剩余MIPS尽可能少。如果存在这样的应急虚拟机lvm_j，则将主机的CPU频率增加到最大，并将这部分CPU频率配置给应急虚拟机lvm_j，然后将任务t_i分配到该应急虚拟机上lvm_j，接着转到第八步；否则，转6.3步；

[0141] 6.3选择一台空闲的应急虚拟机lvm_j，并且它所在主机h_j存在某些虚拟机的分配任务可以忍受应急虚拟机完成任务造成的延迟。如果存在这样的应急虚拟机lvm_j，那么把那些可以忍受延迟的虚拟机的CPU资源暂时转移到应急虚拟机lvm_j，然后将任务t_i分配到应急虚拟机lvm_j，接着转到第八步；否则，转6.4步；

[0142] 6.4选择一台空闲的应急虚拟机lvm_j，并且它所在主机h_j存在某些虚拟机，能够忍受启动主机然后迁移它们到其他主机的时间延迟。如果存在这样的应急虚拟机lvm_j，那么把这些虚拟机的CPU资源转移给应急虚拟机使用，然后将任务t_i分配到应急虚拟机lvm_j。同时开启一台关闭的主机，待关闭主机开启后，再将那些能够容忍延迟的虚拟机，迁移到刚启动的主机上，然后恢复它们的CPU资源供给。

[0143] 第七步，如果任务t_i没有被映射到任何一台虚拟机上，则拒绝任务t_i，即$WT=WT-t_i$；

[0144] 第八步，如果待分配任务集合WT中存在未调度任务，则转第三步；否则，执行第九步；

[0145] 第九步，结束。

[0146] 图2是第6.2.4到6.2.6步的示意图。图2的三个子图中，横轴表示主机的内存资源，纵轴表示主机的CPU资源，虚线框表示主机的最大内存和CPU资源。坐标中的一个矩形表示一台虚拟机，矩形的长度和高度分别表示该虚拟机占用主机的内存和CPU资源大小。如图2(a)所示，在任务分配到应急虚拟机之前，应急虚拟机处于空闲状态，占用主机的CPU资源为$c(lvm_j)$。另外，主机有足够的剩余CPU资源，表示为如图2(a)中的rf_j，当准备分配任务到应急虚拟机上时，如图2(b)所示，增大应急虚拟机提供CPU资源，如图2(b)中$c(lvm_j)$所示。最后，当应急虚拟机完成任务后，降低应急虚拟机CPU资源，如图2(c)中$c(lvm_j)$所示。

[0147] 图3是第6.3.4.4到6.3.4.6步的示意图。如图3(a)所示，主机的CPU资源大部分已经被虚拟机1和虚拟机2使用，当准备分配任务到应急虚拟机上时，主机没有足够的CPU资源供应急虚拟机使用。另外，若虚拟机2上的执行任务和等待执行任务能够忍受应急虚拟机执行任务造成的延迟。如图3(b)所示，将虚拟机2的CPU资源暂时转移给应急虚拟机。待应急虚拟机完成任务以后，再把CPU资源返还给虚拟机2。

[0148] 图4是第6.4.4.4到6.4.4.10步的示意图。如图4(a)所示，当准备分配任务到应急虚拟机上时，主机1没有足够的剩余资源给应急虚拟机1使用。同时，主机1上的其他虚拟机都不能忍受运行应急虚拟机带来的延迟。如果虚拟机2能够忍受启动主机和被迁移的时间开销，那么把虚拟机2的CPU资源转移给应急虚拟机1使用，同时开启一台关闭的主机，如图4(b)所示。待关闭的主机启动后，把虚拟机2从主机1迁移到主机2，如图4(c)所示。待虚拟机2迁移到主机2后，主机2为虚拟机2供给CPU资源，同时在主机2上创建一台应急虚拟机，如图4(d)所示。

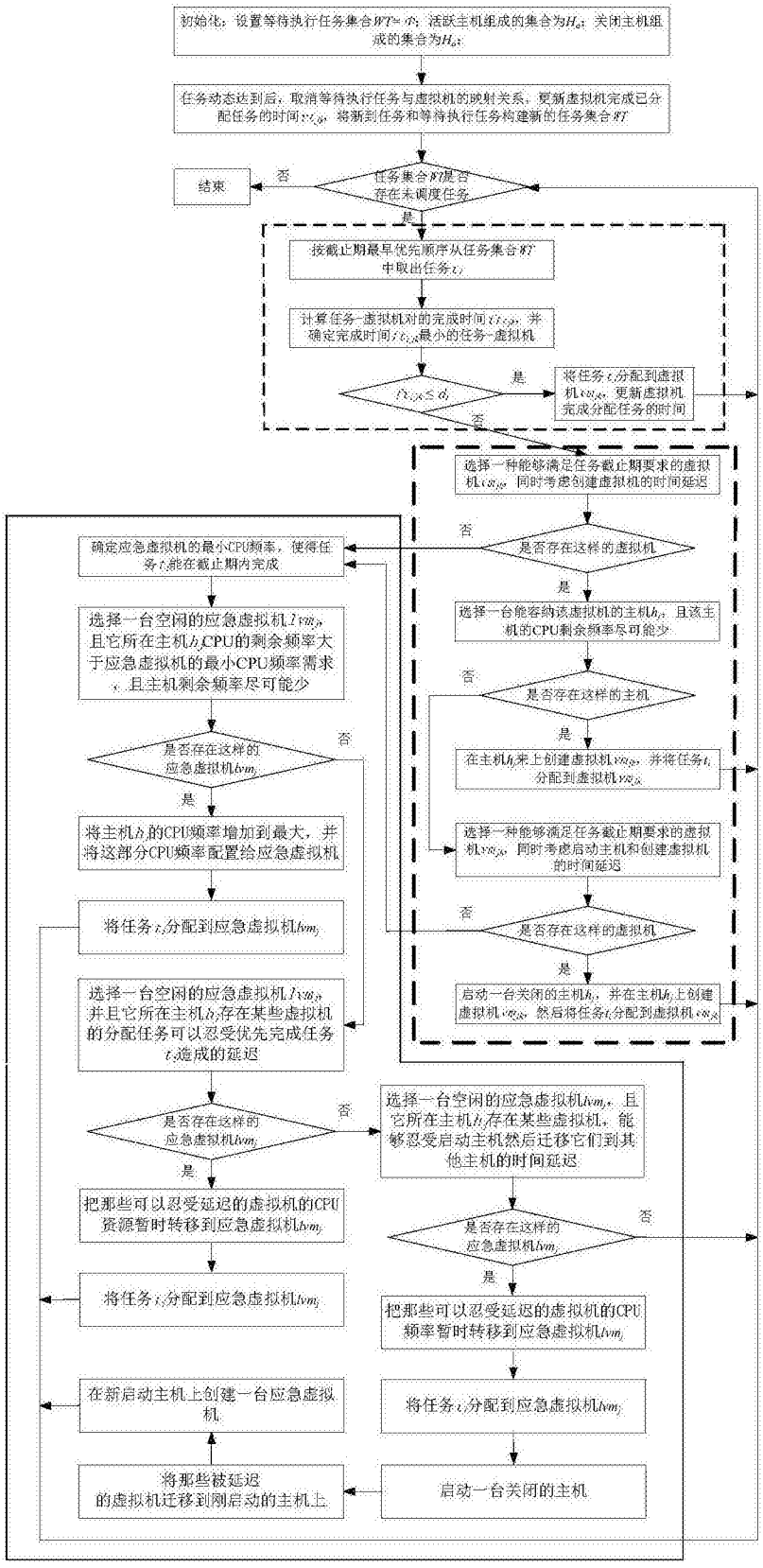

初始化：设置等待执行任务集合$WT=\varnothing$；活跃主机组成的集合为H_a；关闭主机组成的集合为H_s
任务动态达到后，取消等待执行任务与虚拟机的映射关系，更新虚拟机完成已分配任务的时间ct_{ij}，将新到任务和等待执行任务构建新的任务集合RT
结束
否
任务集合RT是否存在未调度任务
是
按截止期最早优先顺序从任务集合RT中取出任务t_i
计算任务-虚拟机对的完成时间ct_{ij}，并确定完成时间ct_{ij}最小的任务-虚拟机
$ft_{ij}\le d_i$
是
将任务t_i分配到虚拟机vm_j，更新虚拟机完成分配任务的时间
否
选择一种能够满足任务截止期要求的虚拟机vm_{js}，同时考虑创建虚拟机的时间延迟
否
是否存在这样的虚拟机
是
选择一台能容纳该虚拟机的主机h_j，且该主机的CPU剩余频率尽可能少
否
是否存在这样的主机
是
在主机h_j来上创建虚拟机vm_{js}，并将任务t_i分配到虚拟机vm_{js}
选择一种能够满足任务截止期要求的虚拟机vm_{js}，同时考虑启动主机和创建虚拟机的时间延迟
否
是否存在这样的虚拟机
是
启动一台关闭的主机h_j，并在主机h_j上创建虚拟机vm_{js}，然后将任务t_i分配到虚拟机vm_{js}
确定应急虚拟机的最小CPU频率，使得任务t_i能在截止期内完成
选择一台空闲的应急虚拟机lvm_j，且它所在主机h_jCPU的剩余频率大于应急虚拟机的最小CPU频率需求，且主机剩余频率尽可能少
是否存在这样的应急虚拟机lvm_j
否
是
将主机h_j的CPU频率增加到最大，并将这部分CPU频率配置给应急虚拟机
将任务t_i分配到应急虚拟机lvm_j
选择一台空闲的应急虚拟机lvm_j，并且它所在主机h_j存在某些虚拟机的分配任务可以忍受优先完成任务t_i造成的延迟
是否存在这样的应急虚拟机lvm_j
否
是
把那些可以忍受延迟的虚拟机的CPU资源暂时转移到应急虚拟机lvm_j
将任务t_i分配到应急虚拟机lvm_j
在新启动主机上创建一台应急虚拟机
将那些被延迟的虚拟机迁移到刚启动的主机上
选择一台空闲的应急虚拟机lvm_j，且它所在主机h_j存在某些虚拟机，能够忍受启动主机然后迁移它们到其他主机的时间延迟
是否存在这样的应急虚拟机lvm_j
否
是
把那些可以忍受延迟的虚拟机的CPU频率暂时转移到应急虚拟机lvm_j
将任务t_i分配到应急虚拟机lvm_j
启动一台关闭的主机

图1

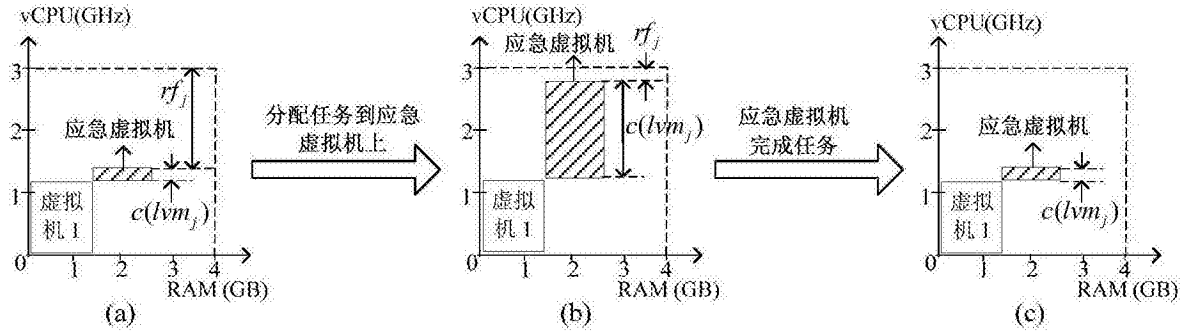

vCPU(GHz)
应急虚拟机
rf_j
$c(lvm_j)$
虚拟机 1
RAM (GB)
(a)
分配任务到应急虚拟机上
(b)
应急虚拟机完成任务
(c)

图2

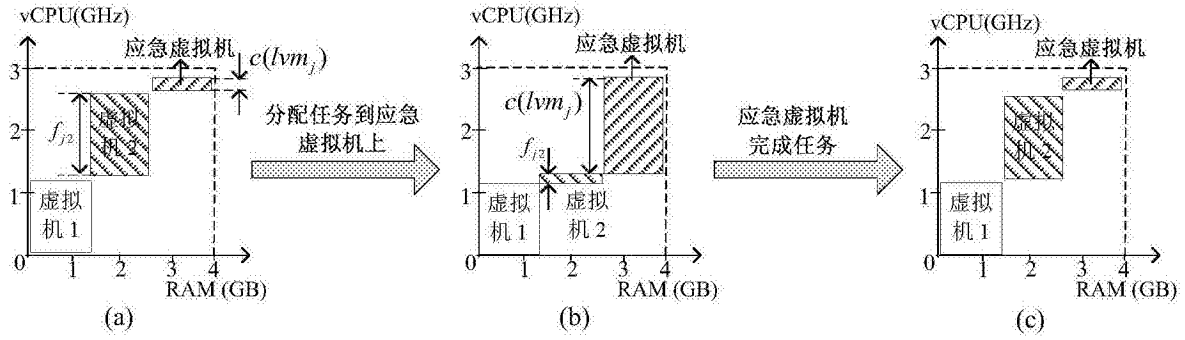

vCPU(GHz)
应急虚拟机
$c(lvm_j)$
f_{j2}
虚拟机 2
虚拟机 1
RAM (GB)
(a)
分配任务到应急虚拟机上
(b)
应急虚拟机完成任务
(c)

图3

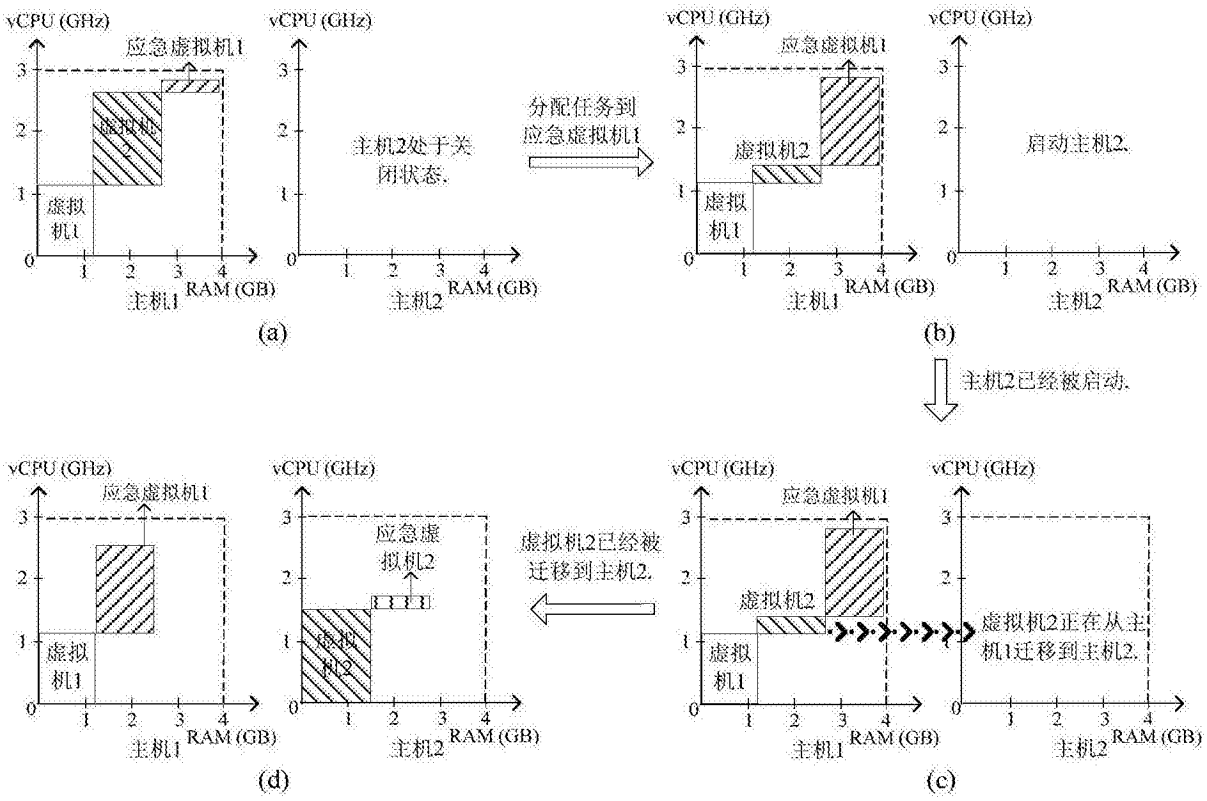

vCPU (GHz)
应急虚拟机1
虚拟机2
虚拟机1
主机1
RAM (GB)
(a)
主机2处于关闭状态.
主机2
分配任务到应急虚拟机1
(b)
启动主机2.
主机2已经被启动.
(c)
虚拟机2正在从主机1迁移到主机2.
虚拟机2已经被迁移到主机2.
(d)

图4