计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2014.20120950
51(6):1199-1205,2014

# 一种可分片预留接纳控制算法研究

吴黎兵[1,2] 党 平[1] 聂 雷[1] 何炎祥[1,2] 李 飞[1]
[1](武汉大学计算机学院 武汉 430072)
[2](软件工程国家重点实验室(武汉大学) 武汉 430072)
(wu@whu.edu.cn)

# A Fragmentable Admission Control Algorithm for Resource Reservation

Wu Libing[1,2], Dang Ping[1], Nie Lei[1], He Yanxiang[1,2], and Li Fei[1]
[1](*School of Computer, Wuhan University, Wuhan* 430072)
[2](*State Key Laboratory of Software Engineering (Wuhan University), Wuhan* 430072)

**Abstract** Admission control algorithms have direct impact on the overall performance of advance reservation mechanisms in the distributed computing. Based on the advantages and disadvantages of existing flexible admission control algorithms, a fragmentable admission control algorithm for resource reservation is proposed. When the fixed resource reservation cannot be achieved, the algorithm allows reserving resources in fragments with the maximum fragment interval. And when there are slots whose remaining amount of resources is less than the requested amount in a fragment, it allows setting aside the minimum amount of resources. Experiments with other three kinds of extensible admission control algorithms (shorten the duration and increase the reserved bandwidth; reduce the reserved bandwidth and extend the duration; change the start time) show that the fragmentable admission control algorithm for resource reservation can effectively reduce the resource fragmentations, and obtain better performance in acceptance rate and effective resource utilization.

**Key words** distributed computing; advance reservation; admission control algorithms; fragmentable admission control algorithm; extensible admission control algorithms

**摘 要** 接纳控制算法的好坏直接影响分布式计算中资源提前预留机制的总体性能.针对现有灵活资源预留接纳控制算法的优缺点,提出了一种可分片预留接纳控制算法.当无法实现固定资源预留时,该算法在保证最大分片间隔的前提下,允许对资源进行分片预留;在各分片中,若存在剩余资源量小于请求预留资源量的时隙,允许用最小资源量进行预留.通过与3种可拓展预留接纳控制算法(缩短持续时间,增大预留带宽(shorten the duration and increase the reserved bandwidth, SDIB);减小预留带宽,延长持续时间(reduce the reserved bandwidth and extend the duration, RBED);改变预留的开始时间(change the reserved start time, CST))的对比实验,从接纳率和有效资源利用率方面进行了评估.实验结果表明,可分片预留接纳控制算法能有效减少资源碎片,具有更优的综合性能.

**关键词** 分布式计算;提前预留;接纳控制算法;可分片预留接纳控制算法;可拓展预留接纳控制算法

**中图法分类号** TP316.4

在高性能分布式计算环境中,一些应用需要同时访问异构的分布式资源,这些资源往往位于不同的地域,受制于不同的管理策略,因而难以进行协同分配.资源预留能较好地解决这一问题,它通过事先申请资源,以保证在将来的某一特定时间段内所预留的资源同时可用,并且能够保证资源的服务质量[1].

收稿日期:2012-10-23;修回日期:2013-01-10
基金项目:国家自然科学基金项目(61070010,61170017,61272112)

资源预留是分布式资源管理中一种普遍采用的机制,也成为分布式资源管理的研究热点.

接纳控制作为一种预防性的资源控制方法,是确保获得网络资源的重要手段.实施资源接纳控制的网格系统要求用户在请求使用网格资源时,将自己的资源特征和要求的服务质量告诉网格,系统根据用户的需求和网格现存的资源情况,决定是否接纳用户的资源请求.目前主要存在两类 QoS(quality of service)接纳控制算法:基于模型的接纳控制算法(model-based admission control)和基于测量的接纳控制算法(measurement-based admission control)[2].根据预留参数是否可变,可以将资源预留分为固定资源预留和灵活资源预留两类[3-4].由于固定资源预留的参数(开始时间、持续时间、预留资源大小)固定,往往会造成多个资源请求的叠加,在某个时刻形成资源使用高峰而拒绝后续请求;同时资源预留机制容易产生大量的"资源碎片",造成分布式资源的浪费.针对这一问题,本文提出了一种可分片预留接纳控制算法(fragmentable admission control algorithm for resource reservation, FACA),它在资源处于利用的高峰时间段里可减小对资源量的要求,同时允许对资源进行分片预留,从而获得更高的请求接纳率,提高资源的有效利用率.

本文研究的对象是分布式环境中的计算资源、网络资源、存储资源等(为阐述方便,文中以带宽为例);进行预留操作的设备是端系统(主机、服务器等),不涉及中间的通信系统.分布式环境下的资源预留系统主要包括客户端、资源提供者和资源调度器,其核心是资源调度器.资源调度器拥有各分布式资源的动态信息,它接收客户端发来的请求后,根据最新资源总信息来判断能否接纳该请求.如果请求能够被接纳,则根据该请求的资源需求情况发送资源预留消息,并修改相应的网格资源信息;否则,拒绝该请求.由此可见,相比于固定资源预留协议(resource reservation protocol, RSVP),可分片预留接纳控制算法虽然把长时间预留分解成若干短预留,但其预留操作仍然是一次性完成的,因而不会增大处理延迟,也不会影响到应用的性能.

## 1 相关工作

为了提升固定资源预留的请求接纳率和系统性能,文献[5]提出和实现了一种自适应的接纳控制算法(adaptive measurement-based admission control, AMBAC).文献[6]针对基于测量的接纳控制中的非公平性,提出了一种高效公平的半预留算法(half reserve, HR),该算法能在保证网络服务质量的同时,以较小降低接纳率的代价,保证对各种资源请求(QoS 保障)的接纳公平性.文献[7]针对系统空闲资源耗尽,但还为已接纳而未被激活的业务预留资源这一情况,提出了一种适当借用此预留资源去接纳新的立即被激活业务的接纳控制算法.文献[8]提出了一种任务代理结构,用来表明资源的差额;同时,提出了最早开始时间估计算法,预计排队时间并提交给任务代理.

在已有的灵活资源预留研究中,文献[9]最早探讨了资源预留对系统性能的影响,并提出了可拓展预留机制,可以在运行时动态调整预留请求的参数,从而提高有效资源利用率.然而,由于各时间参数和带宽的变化在一定范围之内,如果参数值的限制条件苛刻,预留请求被拒绝的概率仍然很大.

文献[10]针对确定性资源能力预留的"资源能力碎片"问题,提出了一种支持松弛时间的灵活资源能力预留机制,引入松弛时间 $SlackTime$,使得原来需要在 $StartTime$ 开始的请求可以在区间[$StartTime$, $StartTime+SlackTime$]内开始.文献[11]引入了参数最早可用时间,判断并拒绝预留开始时间早于资源最早可用时间的请求,从而避免任务抢占.文献[12]提出了一种基于可分割任务的网格资源预留机制(task dividable based reservation, TDR).在 TDR 中,资源预留在多个方向具有可变的灵活度,并在一定的条件下被分割为多个子预留,在资源预留请求中增加了参数相邻子预留之间的最大时间间隔 $T_{dis}$、子预留的最短持续时间 $T$ 等.

固定资源预留在不改变请求参数的情况下,采取其他方式优化性能虽然有一定的效果,但因条件苛刻效果并不显著.文献[9-11]仍然是连续预留,在"资源碎片"较多的情况下并不能改善系统性能.在文献[12]中,对于每个子预留,每个时隙预留的带宽是恒定的,由于 $T_{dis}$ 和 $T$ 的限制,某些请求可能因为个别时隙内剩余的带宽不够而被拒绝.本文提出的可分片预留机制对各分片在带宽不足的情况下,各个时隙允许预留最小带宽值,从而提高请求被接纳的概率和有效资源利用率.

## 2 接纳控制算法对比分析

### 2.1 数据结构

在固定资源预留中,请求的数据结构为 $Req(bw,$

$ts$,$td$),其中,$bw$ 表示单位时间内需要预留的带宽;$ts$ 表示预留的开始时刻;$td$ 表示预留的持续时间.在灵活资源预留中所需要的带宽总量不变,开始时间、终止时间、持续时间和单位时间内预留的带宽可以在一定范围内变化.不同的灵活资源预留算法在固定资源预留请求的基础上添加了其他必要的参数,如表 1 所示.表 1 中,$min\_td$ 表示最小持续时间;$min\_bw$ 表示预留的最小带宽;$tb$ 和 $te$ 分别表示最早开始时间和最晚结束时间;$max\_i$ 表示分片间隔的最大时隙数目.

采用时隙数组 $timeslot[i]$来表示当前系统剩余的资源量,如图 1 所示.若每个时隙表示的时间长度为 $len$,则图 1 所示为在$[0,len)$时间段内,系统剩余的带宽为 100,在$[len,2len)$时间段内,系统剩余的带宽为 90,以此类推.每个时隙的初始值为系统能预留的最大带宽.当系统为用户预留带宽($ts$,$td$,$bw$)时,修改时隙数组的值:对所有的时隙 $i$,若 $ts\leqslant i<ts+td$,则 $timeslot[i]=timeslot[i]-bw$.

Table 1 Structure of Flexible Resource Reservation Requests

表 1 灵活资源预留请求的结构

| Type | $bw$ | $ts$ | $td$ | $min\_td$ | $min\_bw$ | $max\_i$ | $tb$ | $te$ |
|---|---|---|---|---|---|---|---|---|
| SDIB | √ | √ | √ | √ | | | | |
| RBED | √ | √ | √ | | √ | | | |
| CST | √ | √ | √ | | | | √ | √ |
| FACA | √ | √ | √ | | √ | √ | | √ |

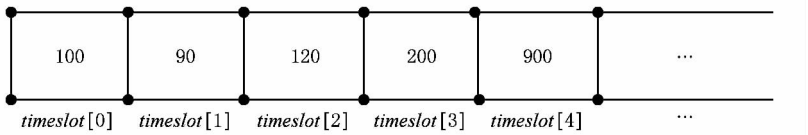



Fig. 1 Time slot array.

图 1 时隙数组

## 2.2 可拓展预留接纳控制算法描述

文献[9]提出的可拓展预留接纳控制算法时间复杂度较高,达到 $O(n^3)$,而且对于固定资源预留的改变总是倾向于缩短持续时间,增大预留带宽,提前资源预留的开始时间.这样经历一段时间后,前面时间段内的资源将被预留完,导致后续请求要推迟开始,进入恶性循环.因此,本文对文献[9]的接纳控制算法进行了研究和改进,在其理论基础上,通过减少参数的改变个数,实现了 3 种复杂度较低的可拓展预留接纳控制算法:1)缩短持续时间,增大预留带宽(shorten the duration and increase the reserved bandwidth, SDIB);2)减小预留带宽,延长持续时间(reduce the reserved bandwidth and extend the duration, RBED);3)改变预留的开始时间(change the reserved start time, CST).本文实现这 3 种可拓展预留接纳控制算法的主要目的是为了和可分片预留接纳控制算法进行对比.

这 3 种算法都是先试图以固定参数进行带宽预留,无法实现时才尝试修改相关参数.且预留的带宽具有如下特点:时隙连续,每个时隙内预留的带宽值恒定.

### 2.2.1 SDIB 算法

SDIB 算法在收到资源预留请求 $Req(bw,ts,td,min_td)$后,计算预留的带宽总量为 $sum\_bw=bw\times td$:

① 检查能否满足预留($bw$,$ts$,$td$),如果可以,则在时间段$[ts,ts+td)$内进行带宽为 $bw$ 的预留,算法终止;否则,转②.

② 若 $ts$ 与当前不能满足预留的时隙 $new\_s$ 之间的间隔不小于 $min\_td$,则新的持续时间为 $new\_td=new\_s-ts$,每个时隙预留的带宽变为 $new\_bw=sum\_bw/new\_td$,转③;否则,预留失败,算法终止.

③ 检查能否满足预留($new\_bw$,$ts$,$new\_td$),如果可以,则在时间段$[ts,ts+new_td)$内进行带宽为 $new\_bw$ 的预留,算法终止;否则,转②.

在最坏情况下,查找资源时将对时隙数组在时间段$[ts,ts+td)$内进行两次遍历,其时间复杂度为 $O(n)$.

### 2.2.2 RBED 算法

与 SDIB 算法类似,RBED 算法在接到资源预留请求 $Req(bw,ts,td,min_bw)$后,也先计算预留的带宽总量 $sum\_bw=bw\times td$:

① 检查能否满足预留($bw$,$ts$,$td$),若能则在时间段$[ts,ts+td)$内进行带宽为 $bw$ 的预留,算法终止;否则,转②.

② 以当前不能满足预留的时隙剩余的带宽量作为新的带宽值 $new\_bw$,若 $new\_bw<min\_bw$,则预留失败,算法终止;否则计算新的持续时间 $new\_td=sum\_bw/new\_bw$,转③.

③ 检查能否满足预留($new\_bw$,$ts$,$new\_td$),如果可以,则在时间段$[ts,ts+new_td)$内进行带宽

为 $new\_bw$ 的预留，算法终止；否则，转②.

在最坏情况下，查找资源时将对时隙数组在时间段$[ts, ts+sum_bw/min_bw)$内进行一次遍历，其时间复杂度为 $O(n)$.

2.2.3 CST 算法

在 SDIB 算法和 RBED 算法中，当固定资源预留无法实现时，我们尝试改变了持续时间和带宽，但它们有一个共同点：不改变预留的开始时间. 而在 CST 算法中，将尝试改变预留的开始时间，而维持持续时间和带宽不变. 改变预留的开始时间包括提前开始时间和推迟开始时间两种. 在 CST 算法中，先尝试提前预留开始时间；当提前开始时间无法满足时再尝试推迟预留开始时间. CST 算法的流程图如图 2 所示：

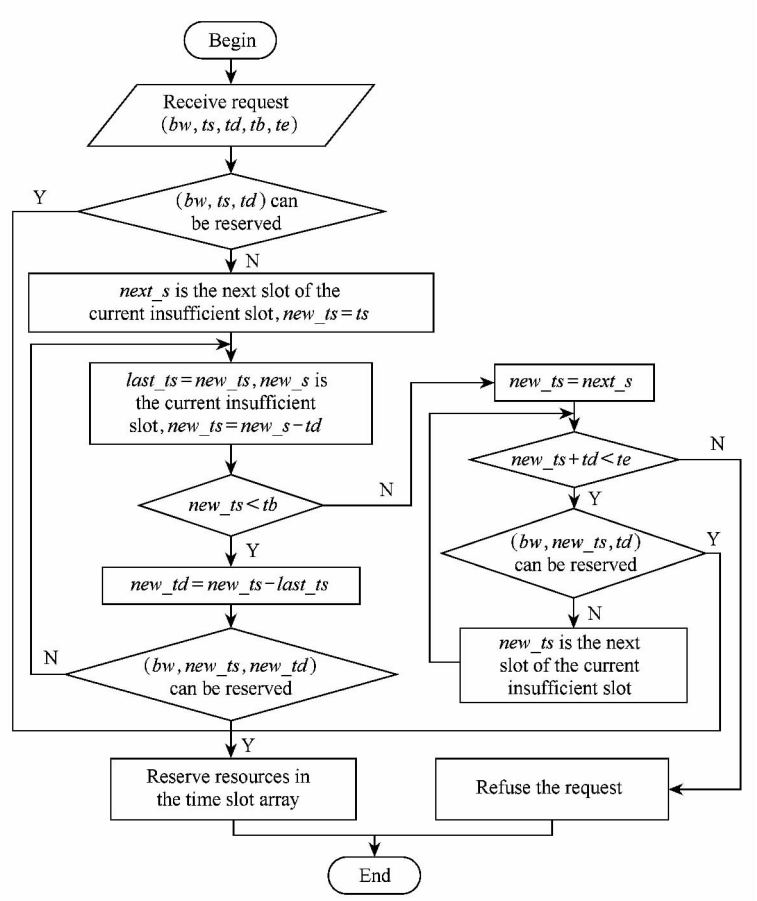



Fig. 2 Algorithm flowchart of CST.

图 2 CST 算法的流程图

CST 算法在收到资源预留请求 $Req(bw, ts, td, tb, te)$后：

1) 检查能否满足预留$(bw, ts, td)$，若能则在时间段$[ts, ts+td)$内进行带宽为 $bw$ 的预留，算法终止；否则，令 $next\_s$ 为当前不能满足预留的下一个时隙，$new\_ts=ts$，转 2).

2) 令 $last\_ts=new\_ts$，$new\_s$ 为当前不能满足预留的时隙，计算新的开始时间为 $new\_ts=new\_s-td$. 若 $new\_ts$ 不小于 $tb$，则由于 $last\_ts$ 到 $new\_s$ 时间段已经被检查过，所以 $new\_td=new\_ts-last\_ts$，转 3)；否则，转 4)，进行推迟预留.

3) 检查能否满足预留$(bw, new_ts, new_td)$，若能则在时间段$[new_ts, new_ts+td)$内进行带宽为 $bw$ 的预留，算法终止；否则，转 2).

4) $new\_ts=next\_s$. 如果 $new\_ts+td$ 不大于 $te$，则转 6)；否则，预留失败，算法终止.

5) $new\_ts$ 表示当前不能满足预留的下一个时隙. 如果 $new\_ts+td$ 不大于 $te$，则转 6)；否则，预留失败，算法终止.

6) 检查能否满足预留$(bw, new_ts, td)$，若能则在时间段$[new_ts, new_ts+td)$内进行带宽为 $bw$ 的预留，算法终止；否则，转 5).

在最坏情况下，查找资源时将对时隙数组在时间段$[tb, te]$内进行一次遍历，其时间复杂度为 $O(n)$.

### 2.3 可分片预留接纳控制算法 FACA

在上述 3 种可拓展预留算法中，预留的时隙是连续的且每个时隙内预留的带宽值相等，这保证了用户在使用资源时不会出现中断和抖动. 而对于某

些应用来说,比如流媒体服务,并不要求带宽的恒定和连续,只需要在客户缓冲区内的数据耗尽之前,流媒体数据继续下载即可.对于内容分发系统也是如此,只需要在既定时间内完成传输,而对传输过程中的带宽变化不敏感.对于生物计算或其他工程计算,只需要在预留的时间 $td$ 内完成计算任务,对于执行过程中 CPU 使用率的抖动、内存分配的变化并不敏感.因此对于多种类型的应用,其带宽的预留可以是不连续的,且在每个时隙内预留的资源量可以变化.为此,我们提出了可分片预留机制,当带宽不足时可进行分片预留,而且在每一段的预留中,各个时隙除了可以以要求的带宽进行预留,还能以最小带宽进行预留,从而获得较高的接纳率和资源利用率,如图 3 所示,其中,$MAX\_B$ 表示最大可用带宽.

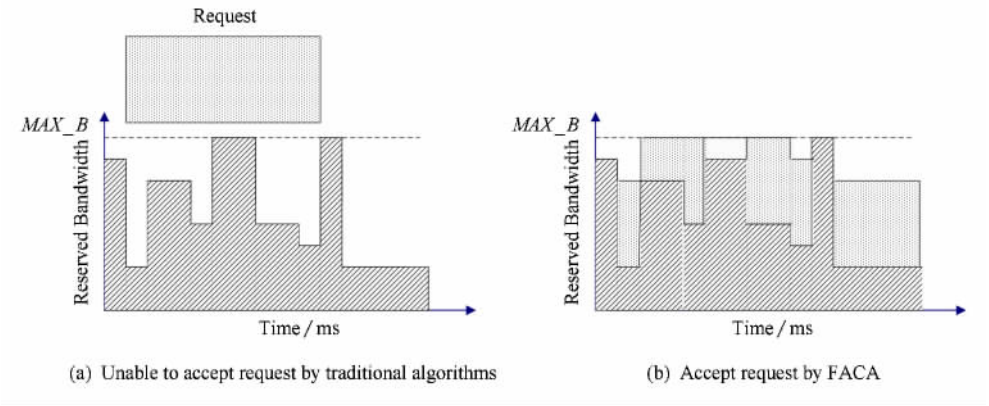



Fig. 3 Example of FACA algorithm.

图 3 FACA 算法示例

在 FACA 算法实现中,$[ts,te)$ 内查找剩余的可用于预留的资源总量:对每个时隙,若其剩余带宽不小于 $min\_bw$ 或者 $bw$,则可预留;同时,在查找的过程中,应该保证分片的间隔不大于最大间隔 $max\_i$.在最坏情况下,查找资源将对时隙数组在时间段 $[ts,te)$ 内进行一次遍历,其时间复杂度为 $O(n)$.

**算法 1.** FACA 算法.

**输入**: $Req(bw,ts,td,min\_bw,max\_i,te)$; $sum\_bw=bw\times td$; /*需预留的资源总量*/

**输出**: 若被接纳,返回 1;否则,返回 0.

```
① begin
② ReceiveRequest(bw, ts, td, min_bw, max_i); /*接收到请求*/
③ if MeetDemand(bw, ts, td) then /*在时间段[ts, ts+td)内的每个时隙中剩余的资源量都大于等于 bw*/
④   Reserve(bw, ts, td); /*在时间段[ts, ts+td)内为该请求预留资源*/
⑤   return 1;
⑥ else
⑦   sum←已遍历时隙个数×bw; /*当前能预留的资源总量*/
⑧   new_s←当前不能满足预留的时隙的下标;
⑨   new_i←max_i;
⑩   if Find(new_s, new_i, min_bw, slot) then /*找到从 new_s 开始 new_i 个时隙内第一个大于 min_bw 的时隙 slot*/
⑪     if 此时隙 slot 内剩余的资源量大于等于 bw then
⑫       sum+=bw;
⑬     else
⑭       sum+=min_bw;
⑮     end if
⑯     if sum≥sum_bw then /*当前当前能预留的资源总量已大于或等于需预留的资源总量*/
⑰       ReserveD(bw, min_bw, ts, slot); /*在时间段[ts, ts+slot)内预留资源*/
⑱       return 1;
⑲     else
⑳       for j from slot+1 to te step 1
㉑         if timeslot[j]≥bw then
㉒           sum+=bw;
㉓           if sum≥sum_bw then
㉔             ReserveD(bw, min_bw, ts, j);
㉕             return 1;
㉖           end if
㉗         else
㉘           new_s←当前不能满足的时隙的下标 j;
㉙           new_i←max_i;
㉚         end if
```

```
㉛          end for
㉜        end if
㉝      else  /* 不能满足预留的时隙个数超过 max_i 个,或者已遍历到结束时间 te */
㉞        break;
㉟      end if
㊱  return 0;
㊲  end if
㊳  end
```

## 3 实验和分析

### 3.1 实验环境和参数

实验采用 Visual Studio 2010 作为开发平台,编程语言为 C#.为了保证实验的可靠性,这 4 种接纳控制算法接收到的请求是相同的;采用足够大的时隙数组存放已接纳的预留信息,每个时隙最大可用带宽为 5 000.时隙数组采用定时器进行更新,每个时隙表示的时间长度是 10 个时间单位(ms).为与当前时间同步,每经过一个时隙的时间,时隙数组下移一个单位.

每两个请求到来的时间间隔遵循参数为 10 个时隙的泊松分布.带宽 $bw$ 遵循(10,200)范围内的均匀分布,持续时间 $td$ 遵循参数为 0.01 的指数分布,开始时间 $ts$ 为系统当前时间加上一个遵循(10,100)范围内的均匀分布的偏移量 $temp$.灵活资源预留参数——最早开始时间 $tb$、最晚结束时间 $te$、最小带宽 $min\_bw$、最短持续时间 $min\_td$ 和最大时间间隔 $max\_i$ 分别由式(1)~(5)刻画:

$$tb = ts - 1/2 \times temp; \tag{1}$$

$$te = ts + td + 1/10 \times td; \tag{2}$$

$$min\_bw = 1/10 \times bw; \tag{3}$$

$$min\_td = 1/10 \times td; \tag{4}$$

$$max\_i = 1/10 \times td. \tag{5}$$

### 3.2 评估指标

本文选用接纳率(acceptance rate, AR)、有效资源利用率(effective resource utilization, ERU)来评价上述 4 种接纳控制算法的性能.AR 表示一段时间内被接纳的请求总数量与到达的请求总数量之比;ERU 表示一段时间内被接纳的请求消耗的资源总量与到达的请求所需预留的资源总量之比.

### 3.3 实验结果与分析

图 4 和图 5 分别表示 4 种灵活资源预留接纳控制算法与固定资源预留接纳控制算法的接纳率和有效资源利用率.由图 4 和图 5 可知,接纳率和有效资源利用率都是随着时间降低的,直到收敛于一个较稳定的值.这是因为刚开始时每个时隙内的剩余带宽都是最大值,而随着时间的迁移,为一些请求预留带宽后可进行预留的带宽减少,而所有请求的开始时间和系统的时间都是往前迁移的,所以最终会收敛于某个值.FACA 具有最高的接纳率和有效资源利用率,因为它允许不连续的资源预留,且不要求每个时隙预留的资源量相同,可以有效地填补资源碎片.CST 算法的性能其次,RBED 的有效资源利用率相比于固定资源预留有所提高,但其接纳率并没有得到改善,而 SDIB 的性能相比于固定资源预留几乎没有提高.

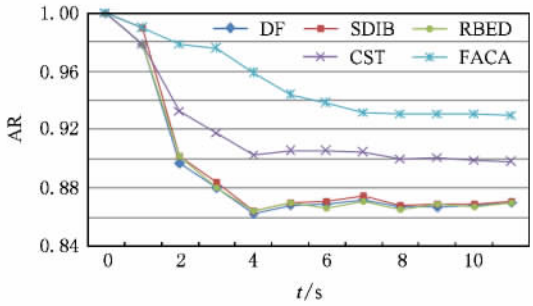



Fig. 4 Comparison of acceptance rate.

图 4 接纳率比较图

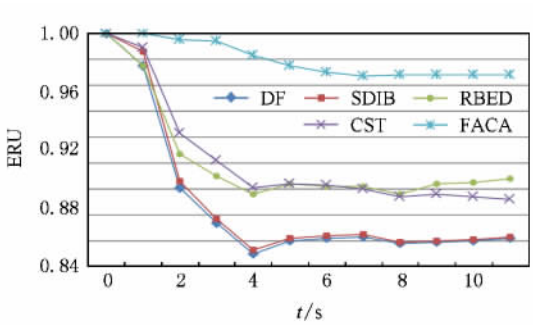



Fig. 5 Comparison of effective resource utilization.

图 5 有效资源利用率比较图

## 4 结 论

为了提高分布式计算环境中资源提前预留的性能,需要良好的接纳控制算法来决定是否对收到的请求予以接纳.可拓展预留接纳控制算法要求预留的资源量为连续恒定的值,导致时间短而资源量大和时间长而资源量小的资源碎片的产生.为解决这一问题,本文提出了可分片预留接纳控制算法,它允许对资源进行分片预留,只要片段与片段之间的间隔不大于某个固定值;在各分片中,若某些时隙剩余的带宽不足可根据最小资源量进行预留.通过与 3 种可拓展预留接纳控制算法在接纳率和有效资源利

用率方面的对比实验表明,可分片预留接纳控制算法可获得更优的性能.

本文对所有的资源请求类型都是平等对待的,然而这与现实的状况存在少量的差异:在一些应用场景中,申请者提交的任务可能具有不同的优先级和事务特征参数.因此,下一步的研究工作将围绕引入优先级的接纳控制算法展开.

## 参 考 文 献


[1] Gao Zhan, Luo Siwei. Dynamic grid resource reservation mechanism based on resource-reservation graph [J]. Journal of Software, 2011, 22(10): 2497-2508 (in Chinese)
(高瞻, 罗四维. 基于资源-预留图的动态网格资源预留机制[J]. 软件学报, 2011, 22(10): 2497-2508)

[2] Gao Qian, Luo Junzhou. Study and comparison of QoS admission control algorithms [J]. Journal of Computer Science, 2004, 31(8): 29-31 (in Chinese)
(高茜, 罗军舟. QoS接纳控制算法的研究与比较[J]. 计算机科学, 2004, 31(8): 29-31)

[3] Farooq U, Majumdar S, Parsons E W. Impact of laxity on scheduling with advance reservations in grids [C]//Proc of IEEE Computer Society's Annual Int Symp on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS'05). Los Alamitos, CA: IEEE Computer Society, 2005: 319-322

[4] Naiksatam S, Figueira S. Elastic reservations for efficient bandwidth utilization in lambda grids [J]. Future Generation Computer Systems, 2007, 23(1): 1-22

[5] Ma Xiaojun, Gu Guanqun. An adaptive measurement-based admission control algorithm [J]. Chinese Journal of Computers, 2001, 24(1): 40-45 (in Chinese)
(马小骏, 顾冠群. 基于测量的接纳控制研究[J]. 计算机学报, 2001, 24(1): 40-45)

[6] Wu Liang, Wang Wei. Measurement-based admission control algorithm for dynamic fairness [J]. Computer Measurement & Control, 2008, 16(3): 339-342 (in Chinese)
(吴亮, 王玮. 一种基于测量的动态公平接纳控制算法[J]. 计算机测量与控制, 2008, 16(3): 339-342)

[7] Zuo Yong, Pan Ke, Liu Xueyong, et al. An algorithm for call admission control based on connection two-phase activation model in IEEE 802.16 networks [J]. Journal of Electronics & Information Technology, 2011, 33(7): 1537-1543 (in Chinese)
(左勇, 潘科, 刘学勇, 等. 基于连接两段激活模型的IEEE 802.16接纳控制算法[J]. 电子与信息学报, 2011, 33(7): 1537-1543)

[8] Liang Feng, Ma Shilong, Schnor B, et al. Earliest start time estimation for advance reservation-based resource brokering within computational grids [C]//Proc of the 8th Int Symp on Parallel and Distributed Processing with Applications. Los Alamitos, CA: IEEE Computer Society, 2010: 8-15

[9] Burchard L O, Heiss H U, De Rose C A F. Performance issues for bandwidth reservations for grid computing [C]// Proc of the 15th Symp on Computer Architecture and High Performance Computing (SBAC-PAD'03). Los Alamitos, CA: IEEE Computer Society, 2003: 82-90

[10] Hu Chunming, Huai Jinpeng, Wo Tianyu. Flexible resource capacity reservation mechanism for service grid using slack time [J]. Journal of Computer Research and Development, 2007, 44(1): 20-28 (in Chinese)
(胡春明, 怀进鹏, 沃天宇. 一种基于松弛时间的服务网格资源能力预留机制[J]. 计算机研究与发展, 2007, 44(1): 20-28)

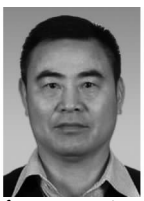
[11] Rblitz T, Schintke F, Reinefeld A. Resource reservations with fuzzy requests [J]. Concurrency and Computation: Practice and Experience, 2006, 18(13): 1681-1703

[12] Pu Jing. Task dividable based reservation for grid computing [J]. Computer Engineering and Applications, 2008, 44(12): 118-120 (in Chinese)
(蒲静. 基于任务可分的网格资源预留机制[J]. 计算机工程与应用, 2008, 44(12): 118-120)


**Wu Libing**, born in 1972. PhD and professor. Senior member of China Computer Federation. His main research interests include distributed computing, trusted software, and network management.

**Dang Ping**, born in 1988. Master candidate in the School of Computer Science of Wuhan University. Her main research interest is distributed computing (dangpingwhu@126.com).

**Nie Lei**, born in 1989. PhD candidate in the School of Computer Science of Wuhan University. His main research interest is distributed computing and network management (lnie@whu.edu.cn).



**He Yanxiang**, born in 1952. PhD and professor. Senior member of China Computer Federation. His main research interests include trusted software, distributed parallel processing, and software engineering (yxhe@whu.edu.cn).

**Li Fei**, born in 1980. PhD and associate professor. His main research interests include distributed parallel processing and computer network (kevin.lifei@gmail.com).