(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 103491174 A
(43)申请公布日 2014.01.01

(21)申请号 201310445288.9

(22)申请日 2013.09.26

(71)申请人 中国船舶重工集团公司第七一六研究所
地址 222006 江苏省连云港市新浦区海连东路 42 号

(72)发明人 殷进勇 袁丽 姚小城 苏培培 李大习 孙忠义

(74)专利代理机构 南京理工大学专利中心 32203
代理人 马鲁晋

(51)Int.Cl.
H04L 29/08(2006.01)

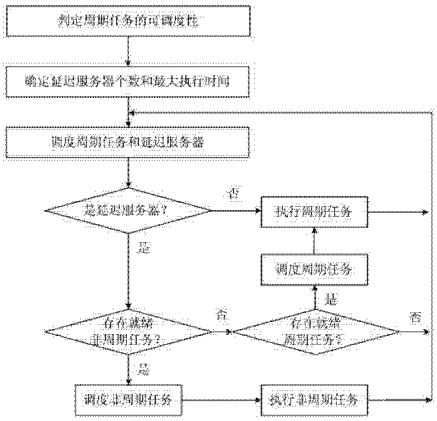
权利要求书2页 说明书8页 附图2页

(54)发明名称

基于延迟服务器的周期/非周期混合实时任务调度方法

(57)摘要

本发明公开了一种在共享内存系统中调度周期/非周期混合强实时任务的方法，该方法在保证周期任务可调度的前提下为非周期任务预留一定的执行时间，能够保证每个周期任务和接收的非周期任务都能满足其截止期限并能有效提高非周期任务的并发性。具体包括以下步骤：(1)判定周期任务集的可调度性；(2)确定延迟服务器个数和最大执行时间；(3)按照DM算法调度周期任务和延迟服务器；(4)如果步骤(3)选中的任务是延迟服务器并且存在就绪非周期任务，则在延迟服务器上调度执行非周期任务；(5)如果步骤(3)选中的任务是周期任务，则执行周期任务；(6)如果步骤(4)中不存在就绪非周期任务，则调度执行周期任务。



CN 103491174 A

1. 一种基于延迟服务器的周期 / 非周期混合实时任务调度方法，其特征在于，包括以下步骤：

步骤 1、根据周期任务的负载判定周期任务的可调度性，如果可调度则执行步骤 2，否则结束操作；

步骤 2、确定延迟服务器个数，在不影响周期任务可调度的前提下，确定每个延迟服务器最大执行时间；

步骤 3、采用 DM 调度算法调度周期任务和延迟服务器；

步骤 4、判定步骤 3 选中的任务是否为延迟服务器，若是延迟服务器，则执行步骤 5，否则执行周期任务，之后返回步骤 3；

步骤 5、判断是否存在就绪的非周期任务，若存在就绪的非周期任务，则在延迟服务器上调度执行非周期任务，之后返回步骤 3，否则执行步骤 6；

步骤 6、判断是否存在就绪的周期任务，若存在就绪的周期任务，则采用 DM 算法调度执行周期任务，之后返回步骤 3，否则直接返回步骤 3。

2. 根据权利要求 1 所述的基于延迟服务器的周期 / 非周期混合实时任务调度方法，其特征在于，步骤 1 根据周期任务的负载判定周期任务的可调度性所用公式为：

$$q=\max\left\{\left(\sum_{i=1}^{k}\beta_i\right)\Big/\left(1-\frac{C_k}{D_k}\right)\,\middle|\,k=1,2,3,\cdots,\ n\right\}$$

其中 q 为周期任务需要的处理器核心数，如果 q 大于系统处理器核心数 m，则周期任务不可调度，否则，周期任务可调度；n 表示周期任务个数；β_i 表示任务 T_i 对任务 T_k 的负载，β_i 计算如下：

$$\beta_i=\begin{cases}\dfrac{C_i}{P_i}\left(1+\dfrac{P_i-\delta_i}{D_k}\right), & \dfrac{C_i}{P_i}\le\lambda\\[2ex] \dfrac{C_i}{P_i}\left(1+\dfrac{P_i-\delta_i}{D_k}\right)+\dfrac{C_i-\lambda P_i}{D_k}, & \dfrac{C_i}{P_i}>\lambda\end{cases}$$

其中 $\lambda=C_k/P_k$；当 $i<k$ 时，$\delta_i=C_i$，当 $i=k$ 时，$\delta_i=D_i$；C_i、D_i、P_i 和 C_k、D_k、P_k 分别表示任务 T_i 和任务 T_k 的执行时间、相对截止期限和周期。

3. 根据权利要求 1 所述的基于延迟服务器的周期 / 非周期混合实时任务调度方法，其特征在于，步骤 2 中延迟服务器个数 h=m，m 为处理器核心数，每个延迟服务器的执行时间和周期与其他服务器完全相同即 $C_i^s=C_j^s$，$P_i^s=P_j^s$ $(1\le i\le m,\ i\ne j)$，C_i^s、C_j^s 和 P_i^s、P_j^s；分别表示延迟服务器 S_i 和 S_j 的执行时间、周期；

确定每个延迟服务器最大执行时间的公式为：

$$C_i^s=\min\left\{\frac{(2P_i^s+D_k)-\sqrt{(2P_i^s+D_k)^2-4P_i^sD_k\beta_i^s}}{2}\,\middle|\,1\le k\le n\right\}$$

其中，C_i^s 表示服务器 S_i 的执行时间；P_i^s 表示服务器 S_i 的周期；D_k 表示任务 T_k 的相对截止期限；β_i^s 为延迟服务器 S_i 对任务 T_k 的负载，β_i^s 计算方法如下：

$$\beta_i^s = \frac{1}{m} \times \min\left\{ m \times \left(1 - \frac{C_k}{D_k}\right) - \sum_{j=1}^{k-1} \beta_j \,\middle|\, 1 < k \le n \right\};$$

其中，m 表示处理器核心数，C_k 和 D_k 分别表示任务 T_k 的执行时间和相对截止期限；β_j 表示任务 T_j 对任务 T_k 的负载。

4. 根据权利要求 1 所述的基于延迟服务器的周期 / 非周期混合实时任务调度方法，其特征在于，步骤 5 中在延迟服务器上调度执行非周期任务，具体包括以下步骤：

步骤 5-1、确定非周期任务的释放时间，非周期任务 J_k 在延迟服务器 S_i 第 j 个周期内的释放时间计算公式如下：

$$R_{kj} = \begin{cases} (j-1)P_i^s, & d_k \le jP_i^s \\ jP_i^s - C_i^{res}, & d_k > jP_i^s \end{cases}$$

其中，C_i^{res} 表示服务器 S_i 在第 j 个周期内的剩余执行时间；P_i^s 表示服务器 S_i 的周期；d_k 表示非周期任务 J_k 的绝对截止期限；

步骤 5-2、判断新到达非周期任务 J_k 的可调度性，所用公式为：

$$SU_i + C_k/D_k \le C_i^s/P_i^s, \quad (1 \le i \le m)$$

其中，SU_i 表示延迟服务器 S_i 的综合利用率，$SU_i = \sum_{J_x \in S(t)} \frac{C_x}{D_x}$，S(t) 表示前忙碌期内截止期限还没有过期的所有非周期任务；

步骤 5-3、对延迟服务器剩余执行时间进行处理，具体方法为：在调度非周期任务时，如果就绪非周期任务的执行时间总和大于延迟服务器的剩余执行时间，则创建延迟服务器时间耗尽事件，触发时间为当前时刻加剩余执行时间，否则迟服务器剩余执行时间减去非周期任务执行时间总和；

步骤 5-4、采用 EDF 算法调度每个延迟服务器上的非周期任务。

基于延迟服务器的周期 / 非周期混合实时任务调度方法

技术领域

[0001] 本发明涉及一种共享内存系统的实时任务调度方法，特别是一种基于延迟服务器的周期 / 非周期混合实时任务调度方法。

技术背景

[0002] 随着电子技术、计算机技术的发展，以多核处理器、SMP 为代表的共享内存系统已在实时系统中得到比较广泛的应用，如何发挥共享内存系统的并行性以及在共享内存系统上调度周期和非周期混合实时任务是一个急需解决的问题。

[0003] 当前混合调度周期 / 非周期实时任务的算法主要有 DS(deferrable server) 算法 (《The deferrable server algorithm for enhanced aperiodic responsiveness in hard real-time environments》)、RB(Reservation-Based) 算法 (《A reservation-based algorithm for scheduling both periodic and aperiodic real-time tasks》)、SSA 算法 (Slack Stealing Algorithm) 算法 (《An optimal algorithm for scheduling soft aperiodic tasks in fixed priority preemptive systems》) 以及其它们扩展版本。DS 算法在系统设置一个高优先级的周期任务服务器，在服务器预留的时间内调度非周期任务，在满足实时周期任务的同时降低了非周期任务的响应时间 ;RB 算法的设计思想与 DS 算法相似，在每个单位周期 (所有周期任务周期的最大公约数) 内为非周期任务预留一定的处理时间，单位周期对应于 DS 算法的服务器周期，降低非周期实时任务截止期限的错失率 ; 而 SSA 算法离线计算周期任务未占用的时间，在未占用的时间上调度非周期任务，需要记录所有周期任务周期的最小公倍数时间内的未占用时间片，浪费大量的存储空间。

[0004] DS 算法、RB 算法以及 SSA 算法都是针对单核处理器系统提出的，并且仅能调度实时周期任务和非实时或软实时非周期任务。殷进勇在其博士论文《可重构系统中实时任务调度算法研究》和《允许多处理机故障的实时任务容错调度算法》文章中从理论上分析了周期任务和延迟服务器的可调度性，并给出了非周期任务在延迟服务器上的可调度性判定条件，但没有给出混合实时任务调度的具体技术方案。

发明内容

[0005] 本发明的目的在于提供一种基于延迟服务器的周期 / 非周期混合实时任务调度方法。

[0006] 实现本发明目的的技术方案为 :一种基于延迟服务器的周期 / 非周期混合实时任务调度方法，包括以下步骤 :

[0007] 步骤 1、根据周期任务的负载判定周期任务的可调度性，如果可调度则执行步骤 2，否则结束操作 ;根据周期任务的负载判定周期任务的可调度性所用公式为 :

[0008] $$q=\max\left\{\left(\sum_{i=1}^{k}\beta_i\right)\Big/\left(1-\frac{C_k}{D_k}\right)\,\middle|\,k=1,2,3,\cdots,\ n\right\}$$

[0009] 其中 q 为周期任务需要的处理器核心数，如果 q 大于系统处理器核心数 m，则周期任务不可调度，否则，周期任务可调度；n 表示周期任务个数；β_i 表示任务 T_i 对任务 T_k 的负载，β_i 计算如下：

[0010] $$\beta_i=\begin{cases}\dfrac{C_i}{P_i}\left(1+\dfrac{P_i-\delta_i}{D_k}\right), & \dfrac{C_i}{P_i}\le\lambda\\ \dfrac{C_i}{P_i}\left(1+\dfrac{P_i-\delta_i}{D_k}\right)+\dfrac{C_i-\lambda P_i}{D_k}, & \dfrac{C_i}{P_i}>\lambda\end{cases}$$

[0011] 其中 $\lambda=C_k/P_k$；当 $i<k$ 时，$\delta_i=C_i$，当 $i=k$ 时，$\delta_i=D_i$；C_i、D_i、P_i 和 C_k、D_k、P_k 分别表示任务 T_i 和任务 T_k 的执行时间、相对截止期限和周期。

[0012] 步骤 2、确定延迟服务器个数，在不影响周期任务可调度的前提下，确定每个延迟服务器最大执行时间；

[0013] 所述延迟服务器个数 h=m，m 为处理器核心数，每个延迟服务器的执行时间和周期与其他服务器完全相同即 $C_i^s=C_j^s$，$P_i^s=P_j^s$ $(1\le i\le m,\ i\ne j)$，C_i^s、C_j^s 和 P_i^s、P_j^s；分别表示延迟服务器 S_i 和 S_j 的执行时间、周期；

[0014] 确定每个延迟服务器最大执行时间的公式为：

[0015] $$C_i^s=\min\left\{\frac{(2P_i^s+D_k)-\sqrt{(2P_i^s+D_k)^2-4P_i^sD_k\beta_i^s}}{2}\,\middle|\,1\le k\le n\right\}$$

[0016] 其中，C_i^s 表示服务器 S_i 的执行时间；P_i^s 表示服务器 S_i 的周期；D_k 表示任务 T_k 的相对截止期限；β_i^s 为延迟服务器 S_i 对任务 T_k 的负载，β_i^s 计算方法如下：

[0017] $$\beta_i^s=\frac{1}{m}\times\min\left\{m\times\left(1-\frac{C_k}{D_k}\right)-\sum_{j=1}^{k-1}\beta_j\,\middle|\,1<k\le n\right\};$$

[0018] 其中，m 表示处理器核心数，C_k 和 D_k 分别表示任务 T_k 的执行时间和相对截止期限；β_j 表示任务 T_j 对任务 T_k 的负载。

[0019] 步骤 3、采用 DM 调度算法调度周期任务和延迟服务器；

[0020] 步骤 4、判定步骤 3 选中的任务是否为延迟服务器，若是延迟服务器，则执行步骤 5，否则执行周期任务，之后返回步骤 3；

[0021] 步骤 5、判断是否存在就绪的非周期任务，若存在就绪的非周期任务，则在延迟服务器上调度执行非周期任务，之后返回步骤 3，否则执行步骤 6；在延迟服务器上调度执行非周期任务，具体包括以下步骤：

[0022] 步骤 5-1、确定非周期任务的释放时间，非周期任务 J_k 在延迟服务器 S_i 第 j 个周期内的释放时间计算公式如下：

[0023] $$R_{kj}=\begin{cases}(j-1)P_i^s, & d_k\le jP_i^s\\ jP_i^s-C_i^{res}, & d_k>jP_i^s\end{cases}$$

[0024] 其中，C_i^{res} 表示服务器 S_i 在第 j 个周期内的剩余执行时间；P_i^s 表示服务器 S_i 的周期；d_k 表示非周期任务 J_k 的绝对截止期限；

[0025] 步骤 5-2、判断新到达非周期任务 J_k 的可调度性，所用公式为：

[0026] $SU_i+C_k/D_k \le C_i^s/P_i^s,\ (1 \le i \le m)$

[0027] 其中，SU_i 表示延迟服务器 S_i 的综合利用率，$SU_i = \sum_{J_x \in S(t)} \frac{C_x}{D_x}$，S(t) 表示前忙碌期内截止期限还没有过期的所有非周期任务；

[0028] 步骤 5-3、对延迟服务器剩余执行时间进行处理，具体方法为：在调度非周期任务时，如果就绪非周期任务的执行时间总和大于延迟服务器的剩余执行时间，则创建延迟服务器时间耗尽事件，触发时间为当前时刻加剩余执行时间，否则迟服务器剩余执行时间减去非周期任务执行时间总和；

[0029] 步骤 5-4、采用 EDF 算法调度每个延迟服务器上的非周期任务。

[0030] 步骤 6、判断是否存在就绪的周期任务，若存在就绪的周期任务，则采用 DM 算法调度执行周期任务，之后返回步骤 3，否则直接返回步骤 3。

[0031] 与现有技术相比，本发明显著优点为：(1) 能够调度周期 / 非周期混合实时任务，保证满足所有周期实时任务和接收的非周期实时任务的截止期限；(2) 非周期任务可调度性判定方法时间复杂度低，时间复杂性为 O(1)；(3) 每个处理器核心设置一个延迟服务器，提高了非周期任务执行的并发性。

附图说明

[0032] 图 1 为周期 / 非周期混合实时任务调度流程图。

[0033] 图 2 为延迟服务器示意图。

[0034] 图 3 为改进的 EDF 算法调度结果图，其中图 (a) 为调度结果图，图 (b) 为延迟服务器剩余执行时间图。

[0035] 图 4 为周期非周期任务就绪队列图，其中图 (a) 周期任务就绪序列图，图 (b) 非周期任务就绪序列图。

具体实施方式

[0036] 由于多处理机系统是分布式内存系统，而多核处理器和 SMP 系统是共享内存系统，两者存在很大的差异，现有的周期 / 非周期混合实时任务调度算法不适用于共享内存系统。本发明在现有技术的基础上，设计了一种适用于共享内存系统的周期 / 非周期混合实时任务调度方法。在实时系统中，周期任务是预先确定的，而非周期任务是随机到达的，该调度方法在保证周期任务可调度的前提下，最大限度地为非周期任务预留 CPU 处理时间，在预留的 CPU 处理时间上调度执行非周期任务。

[0037] 本发明提供一种在共享内存系统中混合调度周期 / 非周期实时任务的调度方法，调度流程如图 1 所示，该方法首先离线判定周期任务可调度性、设置延迟服务器的个数并计算延迟服务器的执行时间，其次在线调度执行周期任务和延迟服务器，并在延迟服务器上调度非周期实时任务，具体步骤描述如下：

[0038] 1. 判定周期任务的可调度性

[0039] 假设系统中有 n 个周期任务，周期任务按周期从小到大的顺序排列，用 $T=\{T_1,$

T_2，…，T_n} 表示；每个周期任务 $T_i \in T$ 用 3 元组 {C_i，D_i，P_i} 表示，其中 C_i 表示任务 T_i 的执行时间；D_i 表示任务 T_i 的相对截止期；P_i 表示 T_i 的周期。周期任务集 T 满足以下不等式，则任务集 T 可调度：

[0040] $$\max\left\{\left(\sum_{i=1}^{k}\beta_i\right)\Big/\left(1-\frac{C_k}{D_k}\right)\Big|\,k=1,\cdots,\ n\right\}\le m$$

[0041] 其中 m 为处理器核心数；β_i 表示任务 T_i 对任务 T_k 的负载，β_i 计算如下：

[0042] $$\beta_i=\begin{cases}\dfrac{C_i}{P_i}\left(1+\dfrac{P_i-\delta_i}{D_k}\right), & \dfrac{C_i}{P_i}\le\lambda\\[2ex] \dfrac{C_i}{P_i}\left(1+\dfrac{P_i-\delta_i}{D_k}\right)+\dfrac{C_i-\lambda P_i}{D_k}, & \dfrac{C_i}{P_i}>\lambda\end{cases}$$

[0043] 其中 $\lambda=C_k/P_k$，当 $i<k$ 时，$\delta_i=C_i$，当 $i=k$ 时，$\delta_i=D_i$。

[0044] 2．设置延迟服务器的个数并计算每个延迟服务器的执行时间

[0045] 延迟服务器是一组特定的最高优先级周期任务，用 S={S_1，S_3，…，S_h} 表示 h 个延迟服务器，每个服务器 $S_i \in S$ 用 2 元组$\{C_i^s,\ P_i^s\}$表示，其中C_i^s和P_i^s分别表示服务器 S_i 的执行时间和周期；只要延迟服务器的执行时间没有耗尽，剩余执行时间可在本周期内的任何时刻执行，如图 2 所示。

[0046] 为了提高非周期任务执行的并发性，该调度方法设置 h=m 个延迟服务器，每个延迟服务器的执行时间和周期与其他服务器完全相同即$C_i^s=C_j^s$，$P_i^s=P_j^s$ $(1\le i\le m,\ i\ne j)$。

[0047] 延迟服务器的最大执行时间$C_i^s(1\le i\le m)$的计算方法如下：

[0048] $$C_i^s=\min\left\{\frac{(2P_i^s+D_k)-\sqrt{(2P_i^s+D_k)^2-4P_i^sD_k\beta_i^s}}{2}\Big|\,1\le k\le n\right\}$$

[0049] 其中，P_i^s表示服务器 S_i 的周期；D_k 表示任务 T_k 的相对截止期限；β_i^s为延迟服务器 S_i 对任务 T_k 的负载，β_i^s计算方法如下：

[0050] $$\beta_i^s=\frac{1}{m}\times\min\left\{m\times\left(1-\frac{C_k}{D_k}\right)-\sum_{j=1}^{k-1}\beta_j\Big|\,1<k\le n\right\}$$

[0051] 3．调度周期任务和延迟服务器

[0052] 采用 DM 调度算法调度周期任务和延迟服务器，如果调度到的任务是延迟服务器时，则采用改进的 EDF 算法调度该服务器上的非周期任务。

[0053] 4．在延迟服务器上调度非周期任务

[0054] 用 J={J_1，J_2，…} 表示系统中存在的非周期任务，每个非周期任务 $J_i \in J$ 用 4 元组 {A_i，C_i，D_i，d_i} 表示，其中 A_i，C_i，D_i，和 d_i 分别表示非周期任务 J_i 的到达时间、执行时间、相对截止期限和绝对截止期限。

[0055] 4.1 当非周期任务 J_k 到达时，查找能够接收此任务的延迟服务器，接收条件为：

[0056] $SU_i+C_k/D_k\le C_i^s/P_i^s$，$(1\le i\le m)$

[0057] 其中，SU_i 表示延迟服务器 S_i 的综合利用率，$SU_i = \sum_{J_x \in S(t)} \frac{C_x}{D_x}$，S(t) 表示前忙碌期内截止期限还没有过期的所有非周期任务。

[0058] 4.2 若服务器 S_i 能够接收非周期任务 J_k，则把任务 J_k 插入本服务器的非周期任务等待序列并修改综合利用率和非周期任务集 S(t)：

[0059] $SU_i=SU_i+C_k/D_k$；

[0060] $S(t)=S(t) \cup \{J_k\}$；

[0061] 4.3 采用改进的 EDF 算法在每个延迟服务器上调度非周期任务，改进的 EDF 算法与标准 EDF 算法的区别为：在保证非周期任务满足截止期限的前提下，尽量延迟非周期任务在延迟服务器每个周期内的释放时间，以便接收更高优先级的非周期任务，调度示意图如图 3 所示。现以延迟服务器 S_i 为例说明非周期任务调度步骤。

[0062] 4.3.1 确定非周期任务 J_k 的在服务器 S_i 的第 j 个周期内的释放时间，计算方法如下：

[0063] $$R_{kj}=\begin{cases}(j-1)P_i^s, & d_k \le jP_i^s \\ jP_i^s - C_i^{res}, & d_k > jP_i^s\end{cases}$$

[0064] 其中 C_i^{res} 为服务器 S_i 在第 j 个周期内的剩余执行时间。

[0065] 4.3.2 如果到达任务 J_k 的释放时间，则把任务 J_k 从非周期任务等待队列转移到非周期任务就绪队列。

[0066] 4.3.3 如果非周期任务就绪队列不空，则执行绝对截止期限最小的非周期任务；否则调度执行周期任务并对综合利用率和集合 S(t) 做相应的处理：

[0067] $SU_i=0$；

[0068] $S(t)=\Phi$；

[0069] 4.4 在非周期任务 J_k 的绝对截止期限 d_k 时刻对综合利用率和集合 S(t) 做相应的处理：

[0070] $SU_i=SU_i-C_k/D_k$；

[0071] $S(t)=S(t)-\{J_k\}$；

[0072] 下面结合附图对本发明作进一步详细描述。本发明一种基于延迟服务器的周期/非周期混合实时任务调度方法分为静态和动态两部分：静态部分完成周期任务可调度性判定、设置延迟服务器的个数并计算每个延迟服务器的执行时间，在系统运行前离线实施，计算结果在系统运行时不再改变；动态部分完成周期任务和延迟服务器的调度以及在每个延迟服务器上的非周期任务调度。具体实施包括如下步骤：

[0073] 1. 判定周期任务的可调度性

[0074] $$q=\max\left\{\left(\sum_{i=1}^{k}\beta_i\right)\Big/\left(1-\frac{C_k}{D_k}\right)\Big| k=1,2,3,\cdots,\ n\right\}$$

[0075] 其中 q 为周期任务需要的处理器核心数，如果 q 大于处理器核心数 m，则周期任务不可调度；β_i 表示任务 T_i 对任务 T_k 的负载，β_i 计算如下：

[0076] $$\beta_i=\begin{cases}\dfrac{C_i}{P_i}\left(1+\dfrac{P_i-\delta_i}{D_k}\right), & \dfrac{C_i}{P_i}\le\lambda\\ \dfrac{C_i}{P_i}\left(1+\dfrac{P_i-\delta_i}{D_k}\right)+\dfrac{C_i-\lambda P_i}{D_k}, & \dfrac{C_i}{P_i}>\lambda\end{cases}$$

[0077] 其中 $\lambda=C_k/P_k$，当 $i<k$ 时，$\delta_i=C_i$，当 $i=k$ 时，$\delta_i=D_i$。

[0078] 2．设置延迟服务器个数并计算每个延迟服务器的执行时间

[0079] 为提高非周期任务的并发性，设置 h=m 个延迟服务器，每个延迟服务器的执行时间和周期与其他服务器完全相同即$C_i^s=C_j^s$，$P_i^s=P_j^s$ $(1\le i\le m,\ i\ne j)$。

[0080] 延迟服务器的最大执行时间$C_i^s(1\le i\le m)$的计算方法如下：

[0081] $$C_i^s=\min\left\{\frac{(2P_i^s+D_k)-\sqrt{(2P_i^s+D_k)^2-4P_i^sD_k\beta_i^s}}{2}\,\middle|\,1\le k\le n\right\}$$

[0082] 其中，P_i^s表示服务器 S_i 的周期；D_k 表示任务 T_k 的相对截止期限；β_i^s为延迟服务器 S_i 对任务 T_k 的负载，β_i^s计算方法如下：

[0083] $$\beta_i^s=\frac{1}{m}\times\min\left\{m\times\left(1-\frac{C_k}{D_k}\right)-\sum_{j=1}^{k-1}\beta_j\,\middle|\,1<k\le n\right\}$$

[0084] 3．调度周期任务和延迟服务器

[0085] 系统设置一个周期任务就绪队列 PRdyQueue、一个周期任务运行队列 PRunQueue 和一个周期任务等待队列 PWaitQueue，分别用于保存就绪的周期任务、运行的周期任务和阻塞的周期任务。周期任务就绪队列 PRdyQueue 和周期任务运行队列 PRunQueue 中的任务按照任务周期从小到大的顺序排列。

[0086] 每个延迟服务器 S_i 有一个非周期任务就绪队列 $APRdyQueue_i$、一个非周期任务运行队列 $APRunQueue_i$、一个非周期任务等待队列 $APWaitQueue_i$ 和一个变量 SU_i，分别保存延迟服务器 S_i 上就绪的非周期任务、运行的非周期任务、阻塞的非周期任务和非周期任务综合利用率。

[0087] 周期和非周期任务就绪队列示意图如图 4 所示，其他队列省略。

[0088] 周期任务、延迟服务器和非周期任务采用事件驱动的方式进行调度，系统中设置一个事件队列EventQueue，按照触发时间从小到大的顺序排列。设置一个定时器，用于触发事件队列 EventQueue 中的事件。

[0089] 每个事件用一个 4 元组 $Event_i=\{ID_i,Type_i,RTime_i,TaskID_i\}$ 表示，其中 ID_i，$Type_i$ 和 $RTime_i$ 分别表示事件 $Event_i$ 的标识、类型、触发时间和相应的任务编号。

[0090] 事件的类型包括周期任务就绪事件、延迟服务器就绪事件、延迟服务器时间耗尽事件、非周期任务就绪事件和非周期任务截止期限事件。每一类事件都有一个事件处理函数，在事件触发时刻调用事件处理函数对此类事件进行处理。

[0091] 与周期任务、延迟服务器调度相关的事件处理函数包括周期任务就绪事件处理函数和延迟服务器就绪事件处理函数。

[0092] 周期任务就绪事件 $Event_i$ 的处理函数在周期任务就绪时调用，主要功能为：

(1) 把编号为 $TaskID_i$ 的周期任务插入到周期任务就绪队列 PRdyQueue 中，从时间队列 EventQueue 中删除事件 $Event_i$；(2) 创建一个新事件 $Event_j$ 插入时间队列 EventQueue 中，事件 $Event_j$ 的类型为周期任务就绪事件，触发时间为该任务下一个周期的开始时刻；(3) 启动周期任务调度函数。

[0093] 延迟服务器就绪事件 $Event_i$ 的处理函数在延迟服务器就绪时调用，主要功能为：(1) 把编号为 $TaskID_i$ 延迟服务器插入到周期任务就绪队列 PRdyQueue 中，从时间队列 EventQueue 中删除事件 $Event_i$；(2) 创建一个新事件 $Event_j$ 插入时间队列 EventQueue 中，事件 $Event_j$ 的类型为延迟服务器就绪事件，触发时间为该延迟服务器下一个周期的开始时刻；(3) 补充延迟服务器 S_i 的剩余执行时间 C_i^{res} 为 C_i^s；(4) 确定非周期任务等待队列 $APWaitQueue_i$ 中每个非周期任务 J_k 在服务器 S_i 的第 j 个周期内的释放时间并创建与每个非周期任务对应的新事件 $Event_j$ 插入时间队列 EventQueue 中，事件 $Event_j$ 的类型非周期任务就绪事件，触发时间为非周期任务 J_k 的释放时间，非周期任务 J_k 的释放时间计算方法如下：

[0094] $$R_{kj}=\begin{cases}(j-1)P_i^s, & d_k \le jP_i^s \\ jP_i^s - C_i^{res}, & d_k > jP_i^s\end{cases}$$

[0095] 其中 C_i^{res} 为服务器 S_i 在第 j 个周期内的剩余执行时间；(5) 启动周期任务调度函数。

[0096] 周期任务调度函数在中断处理从内核态切换为用户态时刻、周期任务就绪时刻、周期任务的一个作业执行完毕时刻、延迟服务器时间耗尽时刻以及周期任务由于某种原因受到阻塞的时刻被调用，周期任务调度函数描述如下：(1) 如果 PRunQueue 的长度小于 m，则直接将 PRdyQueue 队头插入到 PRunQueue 中，在空闲处理器核心上执行；(2) 依次比较 PRdyQueue 队头与 PRunQueue 队尾任务的优先级，如果 PRdyQueue 队头任务的优先级低于 PRunQueue 队尾任务的优先级，则终止这一比较过程；否则将 PRdyQueue 队头任务从 PRdyQueue 删除并插入到 PRunQueue 中，将 PRunQueue 队尾任务从 PRunQueue 删除并插入到中 PRdyQueue 中，并交换两个任务的上下文。

[0097] 4. 在延迟服务器上调度非周期任务

[0098] 与非周期任务调度相关的事件处理函数包括延迟服务器时间耗尽事件处理函数、非周期任务就绪事件处理函数和非周期任务截止期限事件处理函数。

[0099] 延迟服务器时间耗尽事件 $Event_i$ 的处理函数在延迟服务器的执行时间被消耗殆尽时调用，主要功能为：(1) 从事件队列 EventQueue 中删除该事件；(2) 终止当前执行的非周期任务并把该任务从非周期任务运行队列 $APRunQueue_i$ 转移到非周期任务等待队列 $APWaitQueue_i$ 中；(3) 把非周期任务就绪队列 $APRdyQueue_i$ 中的所有非周期任务转移到非周期任务等待队列 $APWaitQueue_i$ 中；(4) 把编号为 $TaskID_i$ 的延迟服务器从 PRunQueue 中删除，启动周期任务调度函数。

[0100] 非周期任务就绪事件 $Event_i$ 的处理函数在非周期任务就绪时刻调用，主要功能为：(1) 把任务编号为 $TaskID_i$ 的非周期任务从非周期任务等待队列 $APWaitQueue_i$ 转移到非周期任务就绪队列 $APRdyQueue_i$ 中；(2) 从事件队列 EventQueue 中删除该事件；(3) 调度非周期任务调度函数。

[0101] 非周期任务截止期限事件处理函数在非周期任务绝对截止期限时刻调用，主要功能为修改综合利用率和非周期任务集 S(t)：

[0102] $SU_i=SU_i-C_k/D_k$；

[0103] $S(t)=S(t)-\{J_k\}$；

[0104] 非周期任务调度函数在中断处理从内核态切换为用户态时刻、延迟服务器就绪时刻，非周期任务就绪时刻、非周期任务执行完毕时刻、以及非周期任务由于某种原因受到阻塞的时刻被调用，非周期任务调度函数描述如下：(1) 如果非周期任务就绪队列为空，则重置非周期任务集合 $S(t)=\Phi$ 和综合利用率 $SU_i=0$；(2) 计算所有就绪的非周期任务执行时间总和，如果执行时间总和大于延迟服务器的剩余执行时间，则创建延迟服务器时间耗尽事件，触发时间为当前时间加剩余执行时间，否则迟服务器剩余执行时间减去非周期任务执行时间总和；(3) 执行优先级最高的非周期任务；

[0105] 当非周期任务 J_k 到达时，判定非周期任务的可调度性，判定条件为：

[0106] $$SU_i+C_k/D_k\le C_i^s/P_i^s,\ (1\le i\le m)$$

[0107] 其中，SU_i 表示延迟服务器 S_i 的综合利用率，$SU_i=\sum_{J_x\in S(t)}\frac{C_x}{D_x}$，S(t) 表示前忙碌期内截止期限还没有过期的所有非周期任务。若服务器 S_i 接收非周期任务 J_k，则把任务 J_k 插入本服务器的非周期任务等待队列 $APWaitQueue_i$ 中，修改综合利用率和非周期任务集 S(t)：

[0108] $SU_i=SU_i+C_k/D_k$；

[0109] $S(t)=S(t)\cup\{J_k\}$；

[0110] 创建一个非周期任务截止期限事件，触发时间为任务 J_k 的绝对截止期限 d_k，插入到事件队列 EventQueue 中。

[0111] 由上可知，本发明的方法能够调度周期 / 非周期混合实时任务，保证满足所有周期实时任务和接收的非周期实时任务的截止期限，非周期任务可调度性判定方法时间复杂度低，时间复杂性为 O(1)，非周期任务可在每个处理器核心上执行，提高了非周期任务执行的并发性。

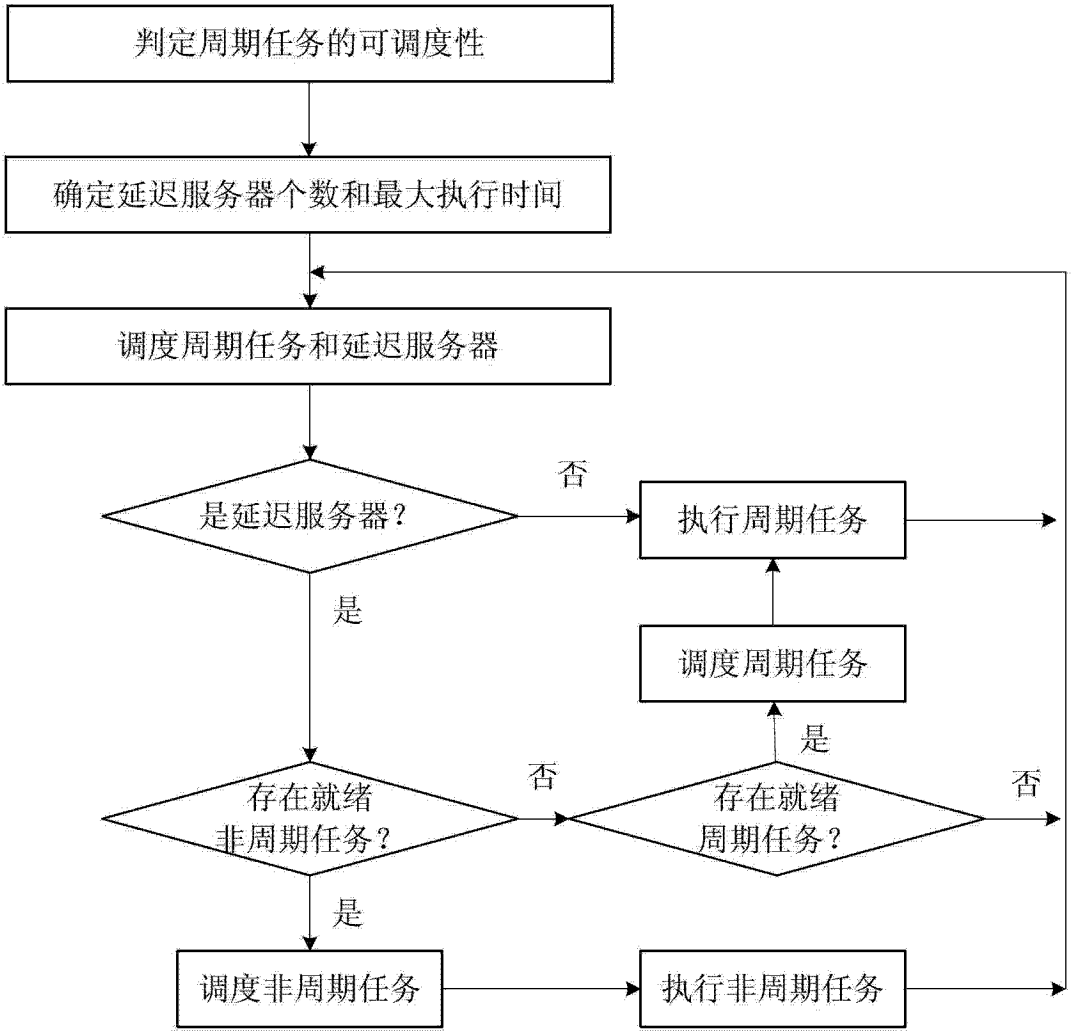


图 1

$$C_i^s = \sum \Delta t_i$$

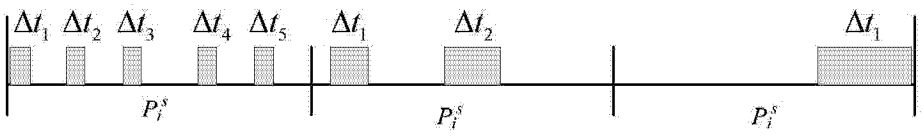


图 2

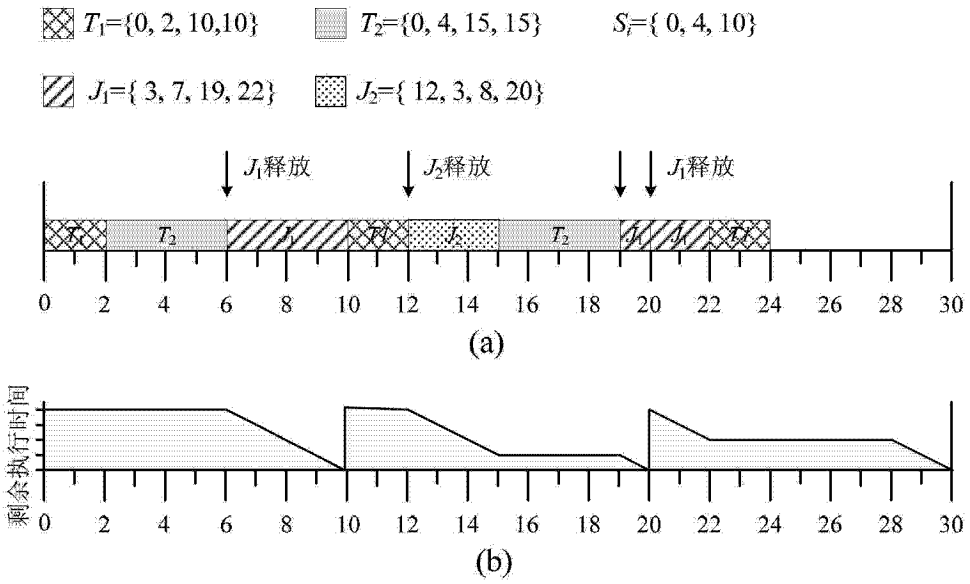

T_1={0, 2, 10,10}
T_2={0, 4, 15, 15}
S_i={ 0, 4, 10}
J_1={ 3, 7, 19, 22}
J_2={ 12, 3, 8, 20}
J_1释放
J_2释放
J_1释放
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
(a)
剩余执行时间
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
(b)

图 3

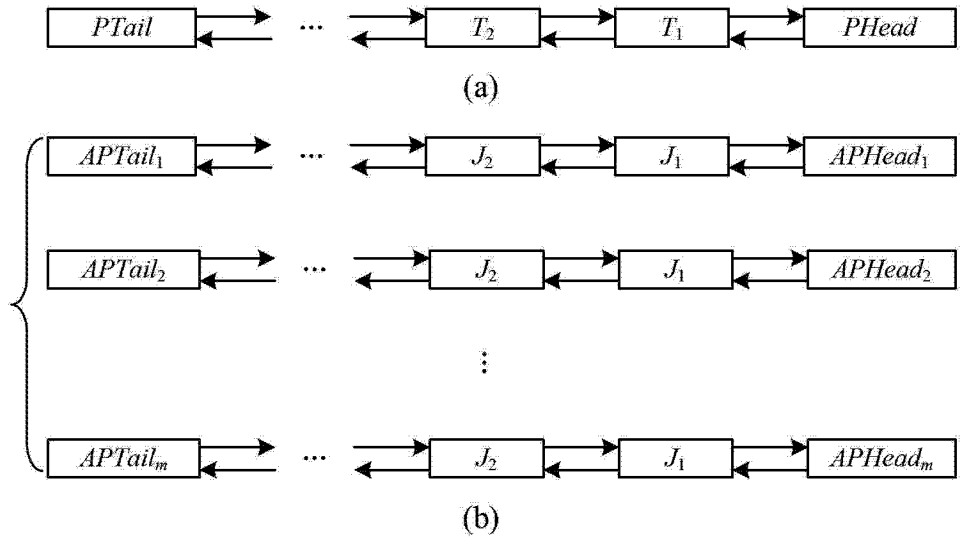

PTail
T_2
T_1
PHead
(a)
$APTail_1$
J_2
J_1
$APHead_1$
$APTail_2$
J_2
J_1
$APHead_2$
$APTail_m$
J_2
J_1
$APHead_m$
(b)

图 4