发明名称
实时交通信息处理的任务调度方法与系统

摘要
本发明涉及实时交通信息处理的任务调度方法与系统。本发明中的任务调度方法，主要依据子任务之间的依赖关系，确定任务优先级。周期性提取子任务的特点和任务服务器的属性，并计算子任务和各个任务服务器的综合效率函数。采用回溯算法将子任务分发给各个任务服务器，并根据各个任务服务器上的任务运行状况，更新存放相关记录的数据结构。本发明中的任务调度系统，包含：一个任务标记单元，能够根据子任务之间的依赖关系确定任务优先级；一个提交任务单元，能够按照优先级顺序发送子任务；一个任务分发单元，能够将子任务分发给各个任务服务器；一个任务运行监视单元，能够监控各个任务服务器上的任务执行情况，并将反馈信息发送给任务分发单元。
1. 实时交通信息处理的任务调度系统，其特征是：包含以下模块：
   一个任务标识器模块：按照子任务之间的依赖关系，确定任务优先级；
   一个提交任务池：暂时存放按照优先级顺序的子任务；
   一个任务分发器模块：按照综合效用函数，采用回溯算法，将子任务分发到各个任务服务器上；
   一个运行监视器模块：监控各个任务服务器上的任务运行状况，收集各个任务服务器的执行任务信息，把这些信息反馈给任务分发器模块。
2. 根据权利要求1所述的实时交通信息处理的任务调度系统，其特征是：所述综合效用函数确定过程如下：
   1) 在截止期限的约束下，建立预计任务完成时间最短的时间效用函数；
   2) 在预算的约束下，建立任务执行成本最小的时间效用函数；
   3) 在任务可靠性的约束下，建立安全性最高的可靠性效用函数；
   4) 在以上效用函数的基础上，建立使系统执行时间、费用和可靠性综合最优的效用函数。
3. 根据权利要求1所述的实时交通信息处理的任务调度系统，其特征是：所述运行监视器模块工作步骤是：
   步骤600：实时监控各个任务服务器上的任务执行情况；
   步骤601：不断更新当前任务已耗时时间表，将各个任务服务器上当前任务的运行时间发送给任务分发器；
   步骤602：检查是否还有其他任务服务器没有被检查过，如果没有则返回步骤600，如果有则进入步骤603；
   步骤603：检查该任务服务器上是否有任务刚刚完成，如果没有则返回步骤602，如果有则进入步骤604；
   步骤604：更新任务计数器，将已经完成任务发送给任务分发器；
   步骤605：更新任务队列表，把该任务服务器上等待处理任务发送给任务分发器；
   步骤606：更新设备故障率表，将该任务服务器的故障率发送给任务分发器。
4. 实时交通信息处理的任务调度方法，其特征是：包括以下步骤：
   步骤200，依据子任务的依赖关系，建立有向无环图；
   步骤201，确定任务优先级，并将子任务放入提交任务池；
   步骤202，周期性提取子任务和任务服务器的属性，并计算子任务和各个任务服务器的综合效用函数；
   步骤203，采用回溯算法将子任务分发到各个任务服务器；
   步骤204，根据各个任务服务器的执行任务运行状况，更新存放相关记录的数据结构。
5. 根据权利要求4所述的实时交通信息处理的任务调度方法，其特征是：所述步骤202中子任务的属性包括：任务长度、关联任务和优先级；任务服务器的属性包括：任务队列、执行速度、单位指令执行成本和故障率。
6. 根据权利要求4所述的实时交通信息处理的任务调度方法，其特征是：步骤203中所述回溯算法具体步骤是：
步骤 500，按照子任务之间的依赖关系，建立有向无环图，并确定任务优先级；
步骤 501，取出优先级最高的子任务；
步骤 502，提取该子任务和各个任务服务器的属性；
步骤 503，计算该子任务与各个任务服务器的综合效用函数；
步骤 504，检查是否有任务服务器满足时间、费用和可靠性效用函数的约束条件，如果有的则进入步骤 505，如果没有则进入步骤 509；
步骤 505，选择具有最大综合效用函数值的任务服务器作为该子任务的分发对象；
步骤 506，检查有向无环图中是否还有其他子任务，如果有则返回步骤 501，如果没有则继续下一步骤 507；
步骤 507，按照所有子任务的分发对象进行任务调度；
步骤 508，休眠等待进入下一个周期；
步骤 509，返回上一个子任务重新确定任务分发对象；
步骤 510，检查是否有任务服务器满足约束条件，如果有则继续进入下一步骤 511，如果没有则进入步骤 513；
步骤 511，检查满足约束条件的任务服务器的综合效用函数值是否小于选定任务服务器，如果是则进入步骤 512，如果否则进入步骤 513；
步骤 512，在满足条件的任务服务器中，选择具有最大综合效用函数值的任务服务器作为该子任务的分发对象；
步骤 513，检查该子任务优先级是否最高，如果是则进入步骤 514，如果否则返回步骤 509；
步骤 514，本周期的任务不能调度，返回步骤 508。
7. 根据权利要求 4 所述的实时交通信息处理的任务调度方法，其特征是：步骤 204 更新存放相关记录的数据结构的过程如下：
1) 根据每个任务服务器最新的当前任务已耗费时间，更新当前任务已耗费时间表；
2) 根据已完成任务情况，更新任务计数器；
3) 根据每个任务服务器的任务队列情况，更新任务队列表；
4) 根据每个任务服务器的故障率，更新设备故障率表。
实时交通信息处理的任务调度方法与系统

技术领域
[0001] 本发明属于智能交通系统 (ITS) 领域，特别是一种 ITS 领域中实时交通信息处理的任务调度方法与系统。

背景技术
[0002] 现有的实时交通信息处理系统主要采用主机式的信息处理系统，即系统的各个任务都由交通指挥中心的中央服务器完成。随着交通采集技术的发展，检测周期的缩短，需要处理的交通数据成倍增加。在这种情况下，交通指挥中心的中央服务器将不能满足实时处理的要求，需要充分利用交通系统内部的网络资源，将复杂繁重的信息处理任务分发给系统内的各个任务服务器，采用分布式处理方式。
[0003] 本发明提出的实时交通信息处理系统包括一个中央服务器和多个任务服务器。任务调度系统安装在交通指挥中心的中央服务器上。实时交通信息处理系统具有周期性处理海量数据的特点，且对任务的截止期限、预算和可靠性有严格的要求。本发明提出实时交通信息处理的任务调度方法与系统，综合考虑了时间、费用和可靠性约束，依据子任务和任务服务器的对应关系，将信息处理的每个子任务分发给合适的任务服务器，使得系统最后的执行时间、费用和可靠性最优。
[0004] 现有的任务调度方法很多，但大多数方法都是将任务分发给时间（响应时间或预计任务完成时间）最短的服务器上，未考虑系统对费用、可靠性等其他因素的要求。
[0005] 2007 年 11 月 7 日公开的中国专利 ZL200710652402.6（公告号 CN101068157），介绍了一种基于时间和费用约束的网格任务调度方法。该方法将资源按带宽 + 速度从大到小排序，任务按任务长度从大到小排序，将任务与资源从左到右一一映射。该发明考虑了时间和费用的综合最优，但未考虑资源与任务的对应关系。在这种情况下，资源与分发的任务可能不匹配，不能保证任务能够成功完成。

发明内容
[0006] 本发明就是鉴于上述技术问题而发明的，其目的在于提供一种实时交通信息处理的任务调度方法与系统，能够把信息处理的周期性任务合理地分发到实时交通信息处理系统的多个任务服务器上执行，在保证任务截止时间、预算和可靠性的前提下，完成周期性任务。
[0007] 为了实现本发明的目的，本发明采取的技术方案是：实时交通信息处理的任务调度方法，首先，依据子任务的依赖关系，建立有向无环图 (DAG 图)，确定任务的优先级，并将子任务放入提交任务池。然后，周期性提取子任务和任务服务器的属性，并计算子任务和各个任务服务器的综合效用函数。采用回溯算法将子任务分发给各个任务服务器。最后，根据各个任务服务器的任务运行状况，更新存放相关记录的数据结构。
[0008] 实时交通信息处理的任务调度系统，包括：一个任务描述器模块：按照子任务之间的依赖关系，确定任务优先级；一个提交任务池：暂时存放按照优先级排序的子任
务；一个任务分发器模块：按照任务权值函数。采用回溯算法，将子任务分发到各个任务服务器上；一个运行监视器模块：监控各个任务服务器上的任务执行状况，收集各任务服务器的运行信息，将这些信息反馈给任务分发器模块。

本发明的有益效果在于：通过计算权值函数，可以将各个子任务与任务服务器之间的配对关系，为任务调度提供依据。通过回溯算法，在保证任务截止时间、预算和可靠性的前提下，找到系统最优的调度方案。通过本发明所描述的系统，可以将实时交通信息处理系统的周期性子任务合理有效地分发到实时交通信息处理系统的各个任务服务器上。

附图说明

图 1 本发明所提及的实时交通信息处理系统的整体图；
图 2 实时交通信息处理的任务调度方法的步骤流程图；
图 3 实时交通信息处理的任务调度系统的组成框图；
图 4 子任务之间依赖关系示意图；
图 5 回溯算法流程图；
图 6 任务运行监视器的工作步骤流程图。

具体实施方式

下面结合附图对本发明作进一步的详细描述：

实时交通信息处理系统的任务可以分解成一组周期性子任务 (T1, T2, ......, Tm)，每个子任务都有一定的长度，用 MI(Million Instruct，百万指令) 表示。整个任务有截止期限 D、预算 B 和可靠性 R 的要求。其中，截止期限 D 用 S(Second，秒) 表示，预算 B 用 CU(Cost Unit，费用单位) 表示，可靠性用 % 表示。实时交通信息处理系统中有一组任务服务器 (S1, S2, ......, Sn)，负责执行从任务分发器得到的子任务。

图 1 是实时交通信息处理系统的整体图。系统由任务接合 100、中央服务器 101 和任务服务器集合 102 组成。任务集合 100 显示了实时交通信息处理系统中各个相互依赖的子任务。任务调度系统安装在中央服务器 101 上，负责将任务分配到各个任务服务器的各个子任务分发到各个任务服务器。图中的任务服务器集合 102，负责各个子任务的具体执行，并将任务的执行情况反馈给中央服务器。所有服务器之间通过物理的网络连接和某种网络协议连接。

如图 2 所示，实时交通信息处理的任务调度方法包括以下步骤：

步骤 200，依据子任务的依赖关系，建立有向无环图；
步骤 201，确定任务优先级，并将子任务放入提交任务池；
步骤 202，周期性提取子任务和任务服务器的属性，并计算子任务和各个任务服务器的综合权值函数；
步骤 203，采用回溯算法将子任务分发给各个任务服务器；
步骤 204，根据各个任务服务器的任务执行状况，更新存放相关记录的数据结构。
[0026] 本发明采用当前已经任务已耗时数据表、任务队列表、设备故障率表和任务计数器记录子任务和任务服务器的状态。以下对各种数据结构做详细解释。
[0027] 当前任务已耗时数据表用来保存每个任务服务器上当前正在运行任务已经耗时的运行时间。表头表示各个任务服务器。第一行数据表示的是每个任务服务器上当前正在运行任务已耗时的运行时间。例如，当前任务已经在任务服务器 S2 上运行了 5s，那么任务服务器 S2 在表中对应的已耗时时间值就是 5s。系统刚刚启动时，表中所有初始值都为 0。
[0028] 任务队列表用来保存各个任务服务器上等待处理任务数。表头表示各个任务服务器。第一行数据表示的是每个任务服务器上当前任务队列的长度。
[0029] 设备故障率表用来保存各个任务服务器的故障率。表头表示各个任务服务器。第一行数据表示的是每个任务服务器完成任务数 f，第二行数据表示的是每个任务服务器分配任务数 d，第三行数据表示任务服务器的故障率，由 (d-f)/d 计算得到。
[0030] 图 3 显示了实时交通信息处理的实时调度系统，主要包括：任务标识器模块 300、提交任务池 301、任务分配器模块 302 和任务运行监视器模块 303。图中的任务标识器模块 300 会根据子任务之间的依赖关系，创建 DAG 图，确立任务的优先级，并将任务放入提交任务池 301。提交任务池中的子任务按照优先级排序，优先级最高的子任务排在队首。任务分配器模块 302 是整个任务调度系统的核心。在每个周期，任务分配器计算每个子任务在各个任务服务器上时间、费用、链路效用函数。其中，时间效用函数的计算，是通过查询当前任务已耗时数据表 305 和任务队列表 306 来完成的。可链路效用函数，采用图测算法将每个子任务分发到合适的任务服务器上去。任务服务器集合 304 由多个任务服务器组成，负责各个子任务的运行。每个任务服务器都有一个任务队列，如任务服务器 1 的任务队列为 Task1。任务服务器接收任务分配器分配的任务，从任务队列的队首一个接一个地取出任务，并运行它们。任务运行监视器模块 305 监控每个任务服务器的任务执行情况，收集实时任务执行信息发送回任务分配器。当前任务已耗时数据表 305、任务队列表 306、设备故障率表 307 和任务计数器 308 分别依据任务运行监视器的反馈信息进行更新。
[0031] 接下来的段落将按照图 2 标明的调度方法的步骤，依次详细介绍任务调度的各个环节和所涉及到的模块。
[0032] 对应于步骤 200 依据子任务的依赖关系，建立有向无环图。图 4 给出一个具体的实施方案，周期性子任务的数目 m = 9。其中，T1 是线圈检测器采集数据的预处理子任务，T2 是微波检测器采集数据的预处理子任务，T3 是视频检测器采集数据的预处理子任务，T4 是数据融合子任务，T5 是交通参数提取子任务，T6 是交通流预测子任务，T7 是交通状态判断子任务，T8 是信息集成子任务，T9 是数据存储子任务。各个子任务的具体功能如下：子任务 T1，子任务 T2，子任务 T3 负责检测原始数据，剔除异常数据，修补残缺数据；子任务 T4 负责对不同采集方式得到的数据进行融合，为应用提供一个统一的数据源；子任务 T5 采用不同的交通模型，对处理后的交通数据进行统计计算，提取各种交通参数，如速度、流量、占有率、行程时间等；子任务 T6 依据子任务 T4 输出的动态交通流数据对未来时段的交通流数据进行预测；子任务 T7 根据子任务 T5 输出的交通参数
和子任务 T6 输出的预测交通数据，对相应路段的交通状态进行判断；子任务 T8 对子任务 T5 输出的实时交通数据进行集成；子任务 T9 将子任务 T8 输出的集成交通数据和子任务 T7 输出的交通状态存储进核心数据库。

对应于步骤 201，确定任务优先级，并将子任务放入提交任务池。任务标志器根据有向无环图，确定任务优先级，并发送到提交任务池。在提交任务池中，子任务按优先级从高到低排序，优先级最高的子任务位于队首。

对应于步骤 202 周期性提取子任务和任务服务器的属性，并计算子任务和各个任务服务器的综合效用函数。表 1 给出子任务和任务服务器的属性列表。

表 1 子任务和任务服务器的属性列表

<table>
<thead>
<tr>
<th>序号</th>
<th>子任务属性</th>
<th>示例</th>
<th>序号</th>
<th>任务服务器属性</th>
<th>示例</th>
</tr>
</thead>
<tbody>
<tr>
<td>X₁</td>
<td>任务特征</td>
<td>10MI</td>
<td>Y₁</td>
<td>任务队列</td>
<td>2</td>
</tr>
<tr>
<td>X₂</td>
<td>关联任务</td>
<td>2</td>
<td>Y₂</td>
<td>执行速度</td>
<td>5MIPC</td>
</tr>
<tr>
<td>X₃</td>
<td>优先级</td>
<td>1</td>
<td>Y₃</td>
<td>单位指令执行成本</td>
<td>2CUPI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Y₄</td>
<td>故障率</td>
<td>0.1%</td>
</tr>
</tbody>
</table>

提取任务的三个属性：任务长度 X₁，关联任务 X₂，优先级 X₃。任务服务器的四个属性：任务队列 Y₁，执行速度 Y₂，单位指令执行成本 Y₃ 和故障率 Y₄。其中，任务队列 Y₁ 用等待处理的任务数表示；执行速度 Y₂，用 MIPS(Million Instructions Per Second, 每秒百万条指令)表示；单位指令执行成本 Y₃，用 CUPI(Cost Unit Per Instruction, 每条指令执行单位)表示；故障率 Y₄，采用没有成功完成的任务数与总的分发任务数的比值表示。

任务调度模型由时间效用函数 O₁、费用效用函数 O₂、可靠性效用函数 O₃ 组成的综合效用函数 U 组成，以下分别介绍。

为了尽快完成任务，应该将任务分发给完成时间最短的任务服务器。因此，将子任务 Tᵢ 在任务服务器 Sᵢ 上的预计完成时间作为时间效用函数：

\[
Oᵢ = EFᵢ
\]  \( (1) \)

其中，EFᵢ 表示任务 Tᵢ 在任务服务器上 Sᵢ 的预计完成时间，它由任务服务器的响应时间 RTᵢ 和子任务 Tᵢ 的预计执行时间 ETᵢ 决定，EFᵢ 通过公式 \( EFᵢ = RTᵢ + ETᵢ \) 计算得到。

任务服务器响应时间 RTᵢ 是指需要等待多久，任务服务器 Sᵢ 才可以运行即将分发的任务。任务服务器 Sᵢ 的 RTᵢ 值的计算，是通过公式 \( RTᵢ = \sum \limits_{k=1}^{(Y₁)} ETᵢ + (ETᵢ − PTᵢ) \) 进行的。在这个公式中，(Y₁) 指任务服务器 Sᵢ 的任务队列长度，\( \sum \limits_{k=1}^{(Y₁)} ETᵢ \) 是任务服务器 Sᵢ 的任务队
列中 (Y1)i 个任务的预计执行时间之和，可以通过 \( \sum_{k=1}^{(Y1)_i} ET_{k} = \frac{\sum_{k=1}^{(Y1)_i} (X1)_k}{(Y2)_i} \) 计算得到。 其中， (X1)_k 表示子任务 T_k 的任务长度， (Y2)_i 是任务服务器 S_i 的执行速度。 T_c 是当前正在任务服务器 S_i 上运行的子任务， (ET_{c}-PT_{c}) 表示 T_c 全部完成预计还需要的时间。 其中， 
(ET_{c}) 是任务 T_c 的预计执行时间，由 (ET_{c}) = (X1)_c/(Y2)_i 计算得到；PT_{c} 是子任务 T_c 已经执行的时间，可以从前任务已耗时时间表中查到。

[0043] 待分发子任务 T_i 的预计执行时间 ET_{i}，可以通过 ET_{i} = (X1)_i/(Y2)_i 计算得到。 综上所述，时间效用函数可以表示为：

\[
O_{i} = \frac{\sum_{k=1}^{(Y1)_i} (X1)_k + (X1)_i + (X1)_c}{(Y2)_i} - PT_{c}
\]  

(2)

[0045] 为了保证任务在截止时间 D 约束条件下完成，未完成任务的平均执行速度 V 应满足：

\[
\bar{V} \geq (D - \sum_{k=1}^{m} T_k) / \sum_{k=1}^{m} (X1)_k, \quad k = 1, 2, \ldots, m;
\]  

(3)

[0047] 其中， 1 是已经完成任务数，可以查询任务计数器得到。 待分发任务 T_i 的预计执行时间 ET_{i} 应满足：

\[
ET_{i} \leq V \times (X1)_i
\]  

(4)

[0049] 综合 (1)、(2) 可以得到时间效用函数的约束条件：

\[
ET_{i} \leq (D - \sum_{k=1}^{m} T_k) \times (X1)_i / \sum_{k=1}^{m} (X1)_k, \quad k = 1, 2, \ldots, m;
\]  

(5)

[0051] ① 为了以最小的成本完成任务，子任务应该分发给费用最小的任务服务。 因此，将子任务 T_i 在任务服务器 S_i 上的执行成本作为费用效用函数：

\[
O_{c} = (X1)_i \times (Y3)_i
\]  

(6)

[0053] 为了保证任务在预算 B 内完成，任务服务器 S_i 的单位指令执行成本 (Y3)_i 应该不大于平均执行成本。 因此，费用效用函数的约束条件为：

\[
(Y3)_i \leq \left[ B - \sum_{k=1}^{m} (X1)_k \times (Y3)_k \right] / \sum_{k=1}^{m} (X1)_k, \quad k = 1, 2, \ldots, m;
\]  

(7)

[0055] 其中， (Y3)_k 表示执行任务 T_k 的相应任务服务器的单位指令执行成本。

[0056] ② 为了保证任务的可靠性，最重要的子任务应该分发给故障率最小的任务服务。 因此采用子任务 T_i 对任务服务器 S_i 的可靠性满意度作为可靠性效用函数：

\[
O_r = \partial_{o}
\]  

(8)

[0058] 其中， \( \partial_{o} \) 表示任务服务器 S_i 在可靠性方面与子任务 T_i 的匹配程度，通过 \( \partial_{o} = [1 - (Y4)_i] / I_i \) 计算得到。 I_i 表示子任务 T_i 的重要度，由 I_i = [m-(X3)_i + (X2)_i]/m 计算得到。 其中， m 表示子任务总数。

[0059] 所以，可靠性效用函数可以表示为：

\[
O_r = \frac{1 - (Y4)_i}{[m - (X3)_i + (X2)_i] / m}
\]  

(9)
为了保证任务在可靠性 R 的约束条件下完成，任务服务器 S_k 的故障率 Y_k 应该不大于未完成任务的平均故障率。因此，可靠性效用函数的约束条件为：

\[
(Y_k)_j \leq 1 - \sqrt{(1 - R) / \prod_{k=1}^{m} (1 - (Y_k)_c)}, \quad k = 1, 2, \cdots, m; \tag{10}
\]

其中，\((Y_k)_c\) 表示执行任务 \(T_k\) 的相应任务服务器的故障率。

对时间效用函数 \(O_1\)、费用效用函数 \(O_2\) 取倒数，可靠性效用函数 \(O_3\) 不变，得到：

\[
O'_1 = 1 / O_1; \tag{11}
\]

\[
O'_2 = 1 / O_2; \tag{12}
\]

\[
O'_3 = O_3; \tag{13}
\]

采用 \(F = (O'_1 - O'_\min) / (O'_\max - O'_\min)\) 对 \(O'_1\)、\(O'_2\) 和 \(O'_3\) 进行归一化，得到 \(F_1, F_2\) 和 \(F_3\)。综合①、②和③的约束条件，得到子任务 \(T_i\) 分发到任务服务器 \(S_k\) 的时间、费用和可靠性综合最优的效用函数，并建立任务调度的目标函数：

\[
\text{max } U = \omega_1 F_1 + \omega_2 F_2 + \omega_3 F_3 \tag{14}
\]

\[
s.t.
\]

\[
ET_{ij} \leq (D - \sum_{k=1}^{m} T_k) / \sum_{k=1}^{m} (X_1)_k; \tag{5}
\]

\[
(Y_k)_j \leq \left[ \frac{B - \sum_{k=1}^{m} (X_1)_k \times (Y_k)_c}{\sum_{k=1}^{m} (X_1)_k} \right] \tag{7}
\]

\[
(Y_k)_j \leq 1 - \sqrt{(1 - R) / \prod_{k=1}^{m} (1 - (Y_k)_c)}; \tag{10}
\]

其中，\(\omega_1, \omega_2, \omega_3\) 分别表示时间、费用和可靠性效用函数的权重。\(k = 1, 2, \cdots, m\)。

对应于步骤 203，采用回溯算法将子任务分发给各个任务服务器。图 5 给出回溯算法流程图。算法具体步骤如下：步骤 500，按照子任务之间的依赖关系，建立有序图，并确定任务优先级；步骤 501，取出优先级最高的子任务；步骤 502，提取该子任务和各个任务服务器的属性；步骤 503，计算该子任务与各个任务服务器的综合效用值；步骤 504，检查是否有任务服务器满足时间、费用和可靠性效用函数的约束条件，如果有则进入步骤 505，如果没有则进入步骤 509；步骤 505，选择具有最大综合效用函数值的任务服务器作为该任务的分发对象；步骤 506，检查有否一个无环图中是否还有其他子任务，如果有则返回步骤 501，如果没有则继续下一次步骤 507；步骤 507，按照所有子任务的分发对象进行任务调度；步骤 508，休眠等待进入下一个周期；步骤 509，返回上一个子任务重新确定任务分发对象；步骤 510，检查是否有任务服务器满足约束条件，如果有则继续下一次步骤 511，如果没有则进入步骤 513；步骤 511，检查满足约束条件的任务服务器的综合效用函数值是否小于选定任务服务器，如果是则进入步骤 512，如果否则进入步骤 513；步骤 512，在满足条件的任务服务器中，选择具有最大综合效用函数值的任务服务器作为该任务的分发对象；步骤 513，检查该任务优先级是否最高，如果是则进入步骤 514，如果不是则返回步骤 509；步骤 514，本期任务不
能调度，返回步骤 508。

[0076] 步骤 204 根据各个任务服务器的运行状况，更新存放相关记录的数据结构。这个步骤主要由任务运行监视器完成，图 6 描述了任务运行监视器的工作步骤。步骤 600：实时监控各个任务服务器上任务执行情况；步骤 601：不断更新当前任务已耗费时间表，将各个任务服务器上当前任务的运行时间发送给任务分发器；步骤 602：检查是否有其他任务服务器没有被检查过，如果没有则返回步骤 600，如果有则进入步骤 603；步骤 603：检查该任务服务器上是否有任务刚刚完成，如果没有则返回步骤 602，如果有则进入步骤 604；步骤 604：更新任务计数器，将已经完成任务数发送给任务分发器；步骤 605：更新任务队列表，把该任务服务器上等待处理任务数发送给任务分发器；步骤 606：更新设备故障率表，将该任务服务器的故障率发送给任务分发器。
图 2

200  依据子任务的依赖关系，建立有向无环图

201  确定任务优先级，并将子任务放入提交任务池

202  周期性提取子任务和任务服务器的属性，计算子任务和各个任务服务器的综合效用函数

203  采用回溯算法将子任务分发给各个任务服务器

204  根据各个任务服务器的的任务运行状况，更新存放相关记录的数据结构
图 4
按照子任务之间的依赖关系，建立DAO图，确定任务优先级

500

取出将处理的优先级最高的子任务

501

提取该子任务和各个任务服务器的属性

502

计算该子任务与各个任务服务器的综合效用函数

503

是否有任务服务器满足时间、费用和可靠性效用函数的约束条件？

504

是

505

选择具有最大综合效用函数值的任务服务器为该子任务的分发对象

否

506

在满足条件的任务服务器中，选择具有最大综合效用函数值的任务服务器作为该子任务的分发对象

507

按照所有子任务的分发对象进行任务调度

508

休眠等待进入下一期

509

否

510

是否

511

是

512

是

513

否

514

本周期任务不能再调度
图 6