

电网调度自动化的任务服务模型和调度算法研究

杨 峰, 刘心松, 邱元杰, 刘 丹

(电子科技大学 8010 研究室, 四川成都 610054)

摘 要: 电网调度自动化是确保电力系统安全、稳定、可靠运行的重要手段, 调度中心任务的处理效率将会直接影响到整个系统的性能. 本文针对电网调度中心的实时任务特点, 提出了一种高可靠的分布式动态镜像服务模型, 并在此模型基础上实现了一种基于预测的任务调度算法——动态模糊调度 DFS(Dynamic Fuzzy Scheduling)算法, 此算法的调度依据是经过模糊变换后的资源调度决策集. 对系统的模拟测试结果表明其具有良好的响应特性、服务能力和可靠性, 比基于 RT-SADS 算法进行调度的系统在性能上有较大的改善.

关键词: 电网调度自动化; 分布式动态镜像服务模型; 动态模糊调度

中图分类号: TP273 **文献标识码:** A **文章编号:** 0372-2112(2002)12-1811-05

Study on Task Service Model and Scheduling Algorithm in Electric Power Automatization

YANG Feng, LIU Xin-song, QIU Yuan-jie, LIU Dan

(8010 division, University of Electronic Science and Technology, Chengdu, Sichuan 610054, China)

Abstract: Electric power automatization is the most important approach that can ensure electric power system running safely, steadily and reliably and the task scheduling efficiency of which greatly affects on the performance of system. This paper analyzed the real-time task characteristics of electric power automatization scheduling center and presented a new distributed dynamic mirror service model. Furthermore, based on these work, we presented a new scheduling algorithm——dynamic fuzzy scheduling (DFS) algorithm whose base is the fuzzy decision-making set that comes from fuzzy transform. The emulation results show that the DFS system owns excellent response characteristic, service capability and reliability, and has a great improvement compared with RT-SADS system.

Key words: electric power automatization; distributed dynamic mirror service model; dynamic fuzzy scheduling

1 引言

电网调度自动化系统是 90 年代发展起来的多专业综合技术, 它以计算机为基础, 实现了电网调度的现代化管理, 是确保电力系统安全、稳定、可靠及经济运行的重要手段. 在电网调度自动化系统(EMS/SCADA)中, 调度中心的实时计算机系统起着重要的作用, 它包括信息总收发、处理和人机联系等几个重要子系统, 该系统的性能直接影响和决定整个电网调度自动化的实用水平. 目前的地区调度自动化系统通常在控制硬件上采用分层分布式结构, 而在调度和控制的软件上, 为了保证整个系统的效率和可靠性, 近年来也逐渐由集中式结构转向分布式结构, 因此电网调度自动化系统实质上是一个分布式实时控制系统^[1-6].

在分布式实时控制系统中, 当构成系统的硬件设备性能完全确定后, 影响系统性能的便是软件设计中所采取的各种策略和算法. 分布式任务调度研究最先出现在非实时系统中^[8-10], 并且被证明是一个 NP 完全问题. 而在实时系统中, 任务又被增加了一些限制, 如准备时间和死线(deadline)等,

这使得实时调度比非实时调度更难. 通常, 分布式实时调度算法采用面向分配(assignment-oriented)不是面向次序(sequence-oriented)的调度方式^[11], 即为满足任务的目标而选定某个节点处理机, 而不是任务执行的顺序. 文献[12]研究了实时分布式对象的调度问题, 其利用任务松弛程度(laxities)和对象的重要性来获得有效的调度决策, 利用本地计算就可产生整个系统的调度策略. 文献[13]中分析了如何调度分布式实时系统的非精确计算任务, 提出对关键系统进行必要的离线(off-line)保证并通过执行可选部分来增加系统的效率. 文献[14]则提出一种基于投标的容错调度算法(bidding-base fault-tolerant scheduling algorithm), 其使用三种任务分配策略对任务进行独立地调度.

以上这些调度算法主要是针对各种不同类型的实时任务进行调度, 对于像电网自动化这类包含周期性和偶发性任务的混合调度不太理想. 同时由于其都基于任务和系统的相关精确信息来进行调度, 因此需要大量的消息通信, 从而影响到系统的服务性能. 针对这种情况, 本文在电网调度任务分解的

收稿日期: 2002-01-23; 修回日期: 2002-07-16

基金项目: 四川省科学研究项目

基础上,提出一种高可靠的分布式动态镜像服务模型,此模型每次调度的执行服务镜像数可以自由设置,从而实现1到 $N-1$ 的备份执行,以满足不同系统的可靠性需求.同时在此模型上实现了一种基于预测的动态模糊调度算法 DFS,此算法共使用了三个模糊集:资源预测加权模糊集、需求上限资源加权模糊集,以及系统剩余资源模糊关系矩阵,将这三个模糊集进行模糊变换后得到调度决策集,再对决策集进行加权平均计算就得到调度结果.经过这样改进之后,整个系统的性能可得到较大的提高.本文以下部分将对这些内容进行详细阐述.

2 系统模型

2.1 任务模型

电网调度自动化系统中的实时任务主要由周期性任务(Periodic Task)和偶发性任务(Sporadic Task)^[15]组成,其中周期性实时任务包括状态检测、远动信息处理等,偶发性实时任务包括故障处理、紧急调度等.

定义1 周期性实时任务 T_i 定义为四元组 (A_i, P_i, D_i, R_i) , A_i 表示任务到达系统的时间, P_i 表示任务的周期, D_i 表示任务的死线(deadline), R_i 表示任务所用的系统资源(resource).任务 T_i 在第 k 周期内使用的系统资源为 $R_i(k)$, $R_i(k)$ 仅在第 k 周期结束时才成为已知.

定义2 偶发性实时任务 S_i 定义为四元组 (A_i, P_i, D_i, R_i) , A_i 表示任务到达系统的时间, P_i 表示两次请求的最小时间间隔, D_i 表示任务的死线, R_i 是任务执行需求的资源上限.这里的 R_i 根据悲观算法得到,以保证任务的实时性.

电网调度自动化中周期性实时任务的 P_i 和 D_i 和电网系统结构密切相关,一旦系统确定下来,这两个参数几乎不再变化.而偶发性实时任务的 P_i 和 D_i 通常是不确定的,其 P_i (两次请求的最小时间间隔)远比周期性实时任务的 P_i (任务的周期)大,一般在数百倍以上.

2.2 分布式动态镜像服务模型

在分布式动态镜像服务模型中,系统由若干高度自治的节点处理机组成,每个节点处理机上运行相同的管理程序和种服务程序,其中服务程序所提供的处理功能被称为“服务”,管理程序则被称为“服务管理器”,“服务管理器”负责管理节点上的所有“服务”,以保证系统中的节点能够协同工作.所谓的动态镜像服务是指每次任务都动态地(镜像节点不固定,随着服务节点的改变而变化)请求多个节点来镜像执行服务,假设系统中总节点数为 N ,任务执行要求的镜像节点数为 i ($0 < i < N-1$),则可根据需要使用不同的 i 值来提供不同的可靠性.当 $i=1$ 时,类似于双机热备份系统(不同之处在于本系统中镜像节点是动态的);而当 $i=N-1$ 时,系统中所有的节点都

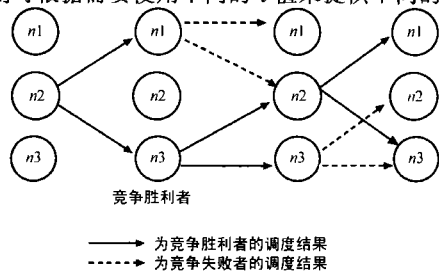


图1 使用互斥量的动态镜像执行过程

相互镜像,此时系统的可靠性达到最高.

在动态镜像服务执行中存在镜像调度竞争,即多个镜像执行节点如何协商由谁发起下一步任务,使用了互斥量来解决此问题.图1是使用互斥量的动态镜像执行过程,其中执行的镜像数 $i=1$,即每次调用两个节点来镜像执行任务.如图1所示,节点 n_2 首先调度两个节点 n_1 和 n_3 来镜像执行任务,在节点 n_2 将任务提交给节点 n_1 和节点 n_3 的同时,其随机地在节点 n_1 或节点 n_3 上建立一个互斥变量.当 n_1 和 n_3 完成服务进行下一步调度时,先完成调度任务的节点将改写互斥变量,并继续进行下一步的任务提交,如图中的 n_3 ,而后完成调度任务的节点检测到互斥变量已被改变,则将放弃进一步的操作,如 n_1 .这样一步步执行下去,直到任务结束.由于每次调度的执行服务镜像数可以自由设置,因此可以实现1到 $N-1$ 的备份执行,从而满足不同系统的可靠性需求.

3 DFS 算法设计

DFS算法的基本思想是根据实时任务的资源预测加权模糊集、需求上限资源加权模糊集、以及系统剩余资源模糊关系矩阵得到调度决策集,然后对调度决策集进行加权平均计算来得到调度结果.其中资源预测加权模糊集由(第 k 个周期后的)周期性实时任务根据其前 k 个周期的资源实际使用值递推得出;需求上限加权模糊集通过悲观算法得到;系统剩余资源模糊关系矩阵则根据资源预测值、需求资源上限或资源测量值得到.

3.1 周期性任务的资源预测

周期性实时任务资源预测是DFS算法的基础.电网调度自动化系统中的周期性实时任务具有处理内容确定、任务关系明确等特点,因此可在第 n 个周期结束的时候,根据前 k 个周期内任务运行时对资源的实际使用值来预测第 $n+1$ 个周期内任务对系统资源的使用情况(将任务的到达时间作为预测点),并且根据此预测值对调度后的剩余资源描述进行更新.本文采用一元线性回归方法来预测任务的第 $n+1$ 个周期使用的资源值.

设周期性实时任务 T_i 的周期为 P_i ,在第 n 个周期结束时,前 k ($k < n$) 个周期使用的资源值为 $R_i(j)$,其中 $j=1, 2, \dots, k$,则关于时间和资源的一元回归方程可表示为

$$\hat{R}_i = a_n + b_n t_i \quad (1)$$

其中 a_n, b_n 由最小二乘法应满足

$$\sum_{j=1}^k [R_i(j) - \hat{R}_i(j)]^2 = \sum_{j=1}^k [R_i(j) - (a_n + b_n t_i(j))]^2 = \text{最小值}$$

设

$$\bar{t}_i = \frac{1}{k} \sum_{j=1}^k t_i(j) = \frac{k+1}{2} p_i, \bar{R}_i = \frac{1}{k} \sum_{j=1}^k R_i(j)$$

则可得

$$b_n = \frac{\sum_{j=1}^k (t_i(j) - \bar{t}_i)(R_i(j) - \bar{R}_i)}{\sum_{j=1}^k (t_i(j) - \bar{t}_i)^2}, a_n = \bar{R}_i - b_n \bar{t}_i$$

那么,第 $n+1$ 个周期内的资源预测值为

$$R_i(k+1) = \frac{1}{k} \sum_{j=1}^k R_i(j) + \frac{\sum_{j=1}^k (t_i(j) - t_i)(R_i(j) - \bar{R}_i)}{\sum_{j=1}^k (t_i(j) - t_i)^2} \cdot \frac{k+1}{2} P_i \quad (2)$$

由于各周期的资源预测结果依赖于前 k 个周期内任务的资源使用值,因此对系统启动的最初 k 个周期无法进行预测,此时以需求上限资源来替代预测值。

3.2 模糊调度

分布式系统中的每台节点计算机有多种资源,可以分别完成多个任务,而完成某一任务通常只需要部分资源,在这里我们将系统的“剩余资源”用模糊矩阵 RR 来描述,任务的“资源预测”用加权模糊集 RP 来表示,“资源上限”用加权模糊集 RU 来表示。

定义 1 “剩余资源”模糊关系矩阵 RR 描述了当前可用资源在系统各节点机的分布情况,表示为 $RR = \{r_{i,j} | i = 1, 2, \dots, n; j = 1, 2, \dots, m\}$, 矩阵元素的下标 i 表示第 i 类资源,下标 j 表示第 j 个节点机, $r_{i,j}$ 表示第 j 台节点机的第 i 类资源的隶属函数值. 同一类资源要求其隶属函数值之和为 1, 即 $\sum_{j=1}^m r_{i,j} = 1$.

定义 2 “资源预测”加权模糊集 RP 描述了任务的资源预测权值,表示为 $RP = \{r_p | i = 1, 2, \dots, n\}$, 集合元素的下标 i 表示任务对第 i 类资源预测使用的权系数. 对同一任务要求权系数之和为 1, 即 $\sum_{i=1}^n r_p(i) = 1$.

定义 3 “资源上限”加权模糊集 RU 描述了任务的资源需求上限权值,表示为 $RU = \{r_u | i = 1, 2, \dots, n\}$, 集合元素的下标 i 表示任务对第 i 类资源需求上限权系数. 资源需求上限权集同样要求其权系数之和为 1, 即 $\sum_{i=1}^n r_u(i) = 1$.

下面通过模糊变换来得到调度决策集. 由定义 1 可得在某一调度时刻的“剩余资源”模糊关系矩阵为

$$RR = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1m} \\ r_{21} & r_{22} & \dots & r_{2m} \\ \dots & \dots & \dots & \dots \\ r_{n1} & r_{n2} & \dots & r_{nm} \end{bmatrix}$$

如果此时到达的是(第 k 个周期后的)周期性实时任务,并由定义 2 得到其加权模糊集为 $RP = \{r_{p1}, r_{p2}, \dots, r_{pn}\}$, 则调度决策集为

$$D_1 = RP \circ RR = (r_{p1}, r_{p2}, \dots, r_{pn}) \circ \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1m} \\ r_{21} & r_{22} & \dots & r_{2m} \\ \dots & \dots & \dots & \dots \\ r_{n1} & r_{n2} & \dots & r_{nm} \end{bmatrix} = (d_1, d_2, \dots, d_m) \quad (3)$$

如果到达的是偶发性任务或前 k 个周期内的周期性实时任务,则由定义 3 得到加权模糊集 $RU = \{r_{u1}, r_{u2}, \dots, r_{un}\}$, 此时可得调度决策集为

$$D_2 = RU \circ RR = (r_{u1}, r_{u2}, \dots, r_{un}) \circ \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1m} \\ r_{21} & r_{22} & \dots & r_{2m} \\ \dots & \dots & \dots & \dots \\ r_{n1} & r_{n2} & \dots & r_{nm} \end{bmatrix} = (d_1, d_2, \dots, d_m) \quad (4)$$

由式(3)或式(4),根据加权平均法就可得到被调度的节点机

$$u = \lfloor \frac{d_1 + 2 \times d_2 + \dots + m \times d_m}{d_1 + d_2 + \dots + d_m} + 0.5 \rfloor \quad (5)$$

在上述的模糊调度中,当任务调度完成时,需要根据“资源预测”值或“资源上限”值对“剩余资源”模糊关系矩阵进行修正;当任务执行结束时,需要对资源进行回收. 对资源的回收有以下两种情况:(1)若任务使用的是“资源上限”值,则任务结束时需对整个系统资源重新测定,以更新“剩余资源”模糊关系矩阵;(2)若任务使用的是“资源预测”值,则根据任务资源预测值和实测值的偏差来更新“剩余资源”模糊关系矩阵。

3.3 算法描述

3.3.1 第 k 个周期后的周期性实时任务调度

输入:周期性实时任务集 $T_i = \{(A_i, P_i, D_i, R_i) (i = 1, 2, \dots, n)\}$;

输出:被调度的节点机;

算法:

- (1)计算“资源预测”加权模糊集 $RP = \{r_{p1}, r_{p2}, \dots, r_{pn}\}$;
- (2)由“资源预测”加权模糊集 RP 和“剩余资源”模糊关系矩阵 RR 相积得到模糊决策集 D_1 ;
- (3)计算模糊决策集 D_1 的加权平均值,得到被调度的节点机 u ;
- (4)若节点机 u 的资源满足任务的 deadline 要求,则将任务分配给它;
- (5)根据“资源预测”值对“剩余资源”模糊关系矩阵 RR 进行修正;
- (6)等待下一个实时任务的到来。

3.3.2 偶发性任务和前 k 个周期内的周期性实时任务调度

输入:偶发性实时任务集 $S_i = \{(A_i, P_i, D_i, R_i) (i = 1, 2, \dots, n)\}$;

周期性实时任务集 $T_i = \{(A_i, P_i, D_i, R_i) (i = 1, 2, \dots, n)\}$;

输出:被调度的节点机;

算法:

- (1)计算“资源上限”加权模糊集 $RU = \{r_{u1}, r_{u2}, \dots, r_{un}\}$;
- (2)由“资源上限”加权模糊集 RU 和“剩余资源”模糊关系矩阵 RR 相积得到模糊决策集 D_2 ;
- (3)计算模糊决策集 D_2 的加权平均值,得到被调度的节点机 u ;
- (4)若节点机 u 的资源满足任务的 deadline 要求,则将任务分配给它;
- (5)根据“系统上限”值对“剩余资源”模糊关系矩阵 RR 进行修正;
- (6)等待下一个实时任务的到来。

4 实验分析

我们对上述模型和算法进行了仿真,为了比较性能,对基于 RT-SADS(Real-Time Self-Adjusting Dynamic Scheduling)算法^[11]的系统也进行了测试.实验的模拟环境节点机配置为 PIII 800MHz CPU, 128M 内存, 100M 网卡, Linux 平台, 内核版本 2.2.16, 网络环境为 1G, 16 端口交换机所构成的局域网.

为了满足大多数电网调度自动化任务处理的应用情况,模拟参数作如下设置:

(1) 处理机节点数设为 N ;

(2) 服务镜像数设为 M ;

(3) 预测周期数 $k = 100$;

(4) 输入多种周期性任务的截止时限 D_i 预先确定, 周期 P_i 约为 2 秒, 每次任务处理的检测数据量为 5000 Bytes;

(5) 输入偶发性任务的 P_i 在满足大于周期性任务最大 P_i 的 100 倍的原则下随机生成, 其截止时限 D_i 根据 P_i 来确定,

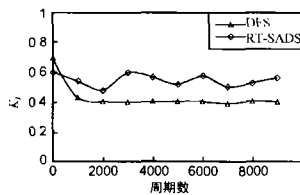


图 2 响应速度系数与系统运行时间的关系

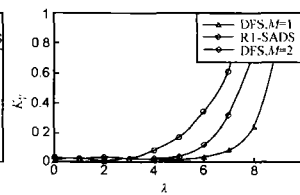


图 3 任务拒绝率与系统负载的关系

4.2 模拟实验 2

本实验研究 DFS 和 RT-SADS 的任务拒绝率 K_r (不能正常调度的任务数与总任务数之比) 与系统负载 λ 的关系, 基本参数 $N = 5$, DFS 的 M 为 1 和 2, 运行在预测稳定区内. 如图 3 所示, 当 λ 较小时 (在 3 以下), DFS 和 RT-SADS 的任务拒绝率几乎都为 0, 这是由于在系统负载较小的情况下, 各节点机拥有的剩余资源能够及时处理到达的任务, 因而不会拒绝新的任务. 但随着系统负载的增大, 节点机的剩余资源减少, 任务拒绝率将会随之增加. 图 3 中 RT-SADS 的 K_r 在 $\lambda = 5$ 时开始出现明显增加. 对于 DFS, 当镜像数为 1 时, K_r 在 $\lambda = 6$ 开始增加, 而镜像数为 2 时, K_r 在 $\lambda = 3$ 时就开始增加. 这表明 DFS 在镜像数为 2 时, 由于多镜像执行的所增加的任务数和通信开销, 其服务能力比 RT-SADS 要差一些. 但在镜像数为 1 的情况下, 其服务能力明显比 RT-SADS 强, 并且此时 DFS 别的性能和 RT-SADS 相差不多, 因此总的来说, DFS 服务能力比 RT-SADS 强.

4.3 模拟实验 3

本实验研究 DFS 的响应速度系数 K_r 与镜像数 M 和负载 λ 的关系, 基本参数 $N = 5$, λ 为 2、4、6, DFS 的镜像数 M 从 1 到 4 变化, 运行在预测稳定区内. 如图 4 所示, K_r 随着镜像数 M 的增加而增加, 但其变化是一个非线性关系. 在镜像数 $M = 0$ 的时候 (即不进行镜像执行时), 由于没有相应的通信开销和额外的镜像任务, 因而响应速度非常快. 但随着 M 的增大, 通信开销和执行的镜像任务数为 $M + 1$ 次抛物线增长, 使得响应时间增加得较快, 特别是在 M 较大的时候. 图 4 还表

每个任务进行一次控制逻辑处理.

(6) 系统负载 $\lambda = \ln$ 任务到达率;

4.1 模拟实验 1

本实验研究 DFS 和 RT-SADS 的响应速度系数 K_r 与系统运行时间的关系, 基本参数 $N = 5$, $\lambda = 3$, DFS 的 $M = 1$. 如图 2 所示, 时间横坐标以周期性任务中的最大 P_i 为单位, 纵坐标 K_r 表示响应速度系数, 定义为 $K_r = P_i / (D_i - A_i)$, 其中 P_i 为完成任务的节点机执行时间, A_i 是任务到达系统的时间, D_i 是任务 deadline. 从图中可以看出, DFS 的 K_r 值比 RT-SADS 的 K_r 低, 并且其变化幅度较小, 因此 DFS 比 RT-SADS 响应速度要快, 对任务的运行更有保障. 同时还可看出 DFS 的 K_r 值在系统运行的初期较高, 此后随着运行时间的增长而逐渐降低并趋于稳定. 这是由于在系统运行的初期, DFS 算法使用的是需求上限资源值, 调度结果不够合理, 而随着运行时间的增长, 调度算法主要使用资源预测值, 从而获得了较好的调度效果.

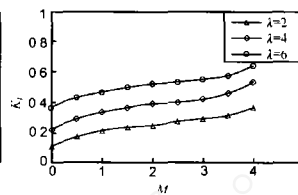


图 4 响应速度系数与镜像数和负载的关系

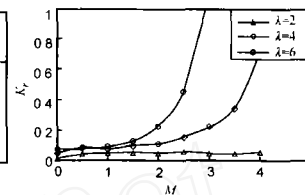


图 5 任务拒绝率与镜像数和负载的关系

明 K_r 随着负载 λ 的增加而增加, 这是由于负载增大使得分配给任务的单位时间使用资源减少, 从而影响了响应速度.

4.4 模拟实验 4

本实验研究任务拒绝率 K_r 与镜像数 M 和负载 λ 的关系, 基本参数 $N = 5$, λ 为 2、4、6, DFS 的 M 从 1 到 4 变化, 运行在预测稳定区内. 如图 5 所示, 在不同的负载下, 任务拒绝率与镜像数 M 的关系不一样. 在 $\lambda = 2$ 时, 由于负载较小, 因此即使 M 增大到 4, 其任务拒绝率也很小; 当 λ 为 4 时, M 增加到 3 任务拒绝率就开始增加了; 而在 λ 为 6 时, 任务拒绝率在 $M = 2$ 就迅速增加. 由此可知, 在任务量较小的时候可以追求高可靠性 (即使用大的 M 值), 而在任务量较大的时候, 必须在可靠性和任务拒绝率之间进行权衡.

此外, 我们还测试了分布式动态镜像服务模型的运行可靠性. 在设定服务镜像数 $M \geq 1$ 和合适系统负载的情况下, 随机的断开系统中的节点机, 然后测试各种任务的完成率. 测试结果表明任务的完成率几乎不发生变化, 说明基于这种模型的系统是相当可靠的, 即使某些节点机出现故障也不会影响到整个系统的运行.

5 结束语

电网自动化调度中心的任务处理效率会直接影响整个系统的性能. 本文建立了电网调度自动化的任务分解模型和分布式动态镜像服务模型, 并在此基础上提出一种基于预测的动态模糊调度算法. 对系统的模拟测试结果表明其具有良好的响应特性、服务能力和可靠性, 比基于 RT-SADS 算法进行

调度的系统在性能上有较大的改善.需要说明的是目前我们的工作主要基于实验室模拟环境,因此在实际应用还需要考虑别的一些因数,如广域网络中的通信延迟等.

参考文献:

- [1] Kuntz P A, Christie R D, Venkata S S. A reliability centered optimal visual inspection model for distribution feeders [J]. IEEE Transactions on Power Delivery, 2001, 16(4): 718 - 723.
- [2] Xiaohong Guan, Yu-Chi Ho, Fei Lai. An ordinal optimization based bidding strategy for electric power suppliers in the daily energy market [J]. IEEE Transactions on Power Systems, 2001, 16(4): 788 - 797.
- [3] Ching-Tzong Su, Chien-Tung Lin. Fuzzy-based voltage/reactive power scheduling for voltage security improvement and loss reduction [J]. IEEE Transactions on Power Delivery, 2001, 16(2): 319 - 323.
- [4] Apostolov A P, Richards S. Event driven maintenance based on protective relays data [A]. Seventh International Conference on Developments in Power System Protection [C]. Amsterdam, Netherlands: 2001. 25 - 28.
- [5] Manzak A, Chakrabarti C. Variable voltage task scheduling for minimizing energy or minimizing power [A]. ICASSP' 00 [C]. Istanbul, Turkey: 2000, (6). 3239 - 3242.
- [6] Katoh Y, Takada H, Iwamoto S. Transient stability preventive control using kinetic energy and critical clearing time [A]. International Conference on Power System Technology [C]. Perth, Australia: 2000, (1). 151 - 156.
- [7] Ramamritham K, Stankovic J A. Scheduling algorithms and operating system support for real-time systems [J]. Proceedings of the IEEE, 1994, 82(1): 55 - 67.
- [8] Belkhale K P, Banerjee P. An Approximate algorithms for the partitionable independent task scheduling problem [A]. Proc. Int'l. Conf. Parallel Processing [C]. St Charles, USA: 1990, (1). 72 - 75.
- [9] Krishnamoorthy R, Ma E. An approximate algorithm for scheduling tasks on varying partition sizes in partitionable multiprocessor systems [J]. IEEE Transaction on Computers, 1992, 41(12): 1572 - 1579.
- [10] Wang Q, Cheng K H. A Heuristic of scheduling parallel tasks and its analysis [J]. SIAMJ. Computing, 1992, 21(2): 281 - 294.
- [11] Atif Y, Hamidzadeh B. A scalable scheduling algorithm for real-time distributed systems [A]. Proceedings of 18th International Conference on Distributed Computing Systems [C]. Amsterdam, Netherlands: 1998. 352 - 359.
- [12] Kalogeraki V, Melliar-Smith P M, Moser L E. Dynamic scheduling for soft real-time distributed object systems [A]. Third IEEE International Symposium on Object-Oriented Real-Time Distributed Computing [C]. California, USA: 2000. 114 - 121.
- [13] de Oliveira R S, da Silva Fraga J, Farines J M. Scheduling imprecise tasks in real-time distributed systems [A]. Fourth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing [C]. Magdeburg, Germany: 2001. 319 - 326.
- [14] Satyanarayana N V, Mall R, Pal A. Fault-tolerant scheduling in distributed real-time systems [A]. 2001 International Conference on Computer Networks and Mobile Computing [C]. Beijing, China: 2001. 275 - 280.
- [15] 张怡, 等. 一种支持 QoS 的连续媒体 CPU 调度方法 [J]. 计算机研究与发展, 2000, 37(2): 233 - 237.

作者简介:



杨 峰 男, 1972 年 10 月生于四川渠县, 博士生, 研究方向为分布式操作系统、分布式并行服务器和宽带综合网络. 已发表学术论文多篇.

刘心松 男, 1940 年 12 月生于四川石柱, 教授, 博士生导师, 研究方向为宽带网络与通信、分布式计算机、并行处理、操作系统与数据库. 已在正式学术刊物上发表学术论文 100 余篇.