



湖南大学

HUNAN UNIVERSITY

同构多核处理器的实时混合 调度算法研究

姓 名：王 艳

学 号：S09102048

专 业：计算机科学与技术

导 师：李仁发 教授

目录

选题背景

论文主要工作

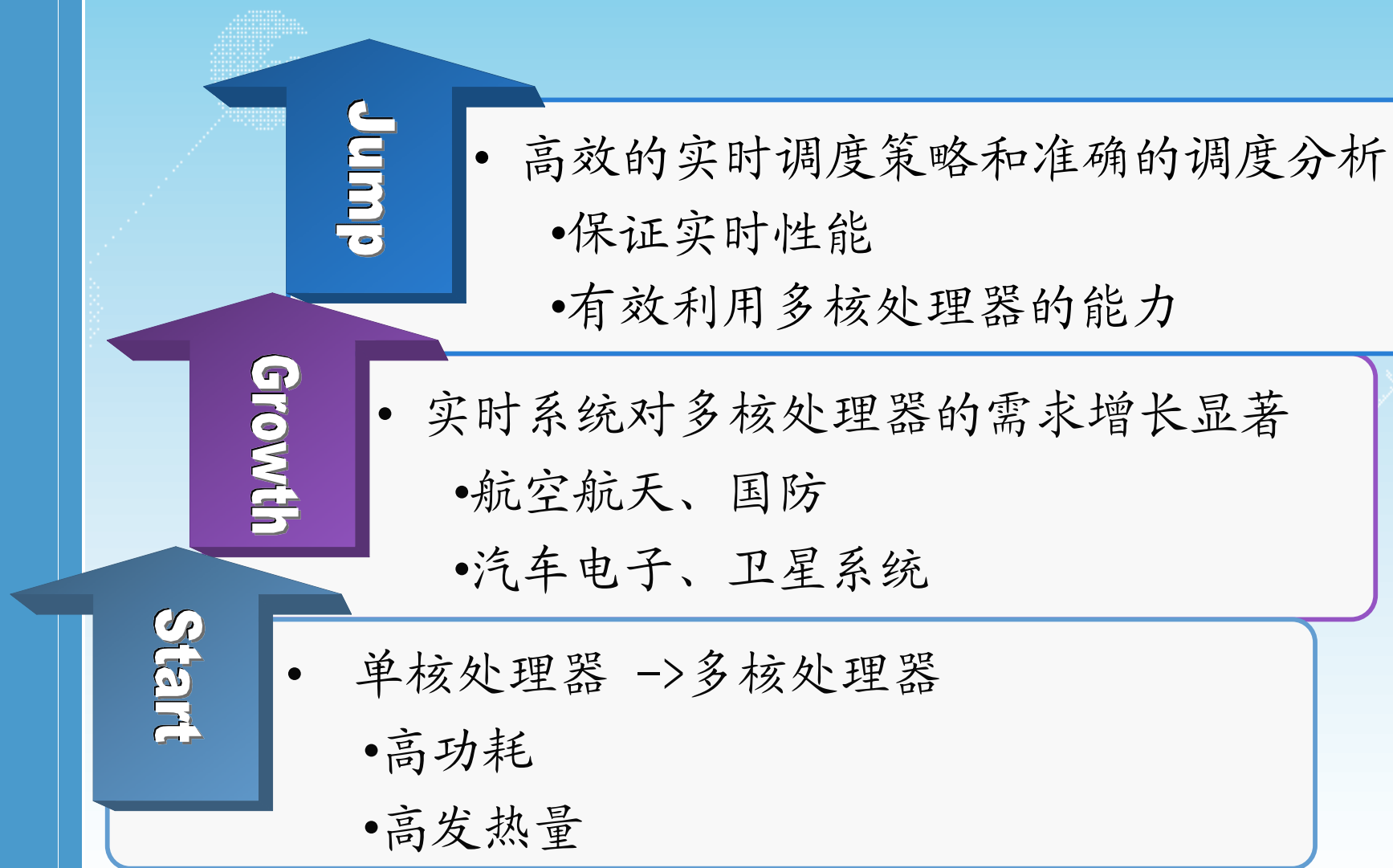
多核实时调度算法概述

基于分簇的混合调度算法

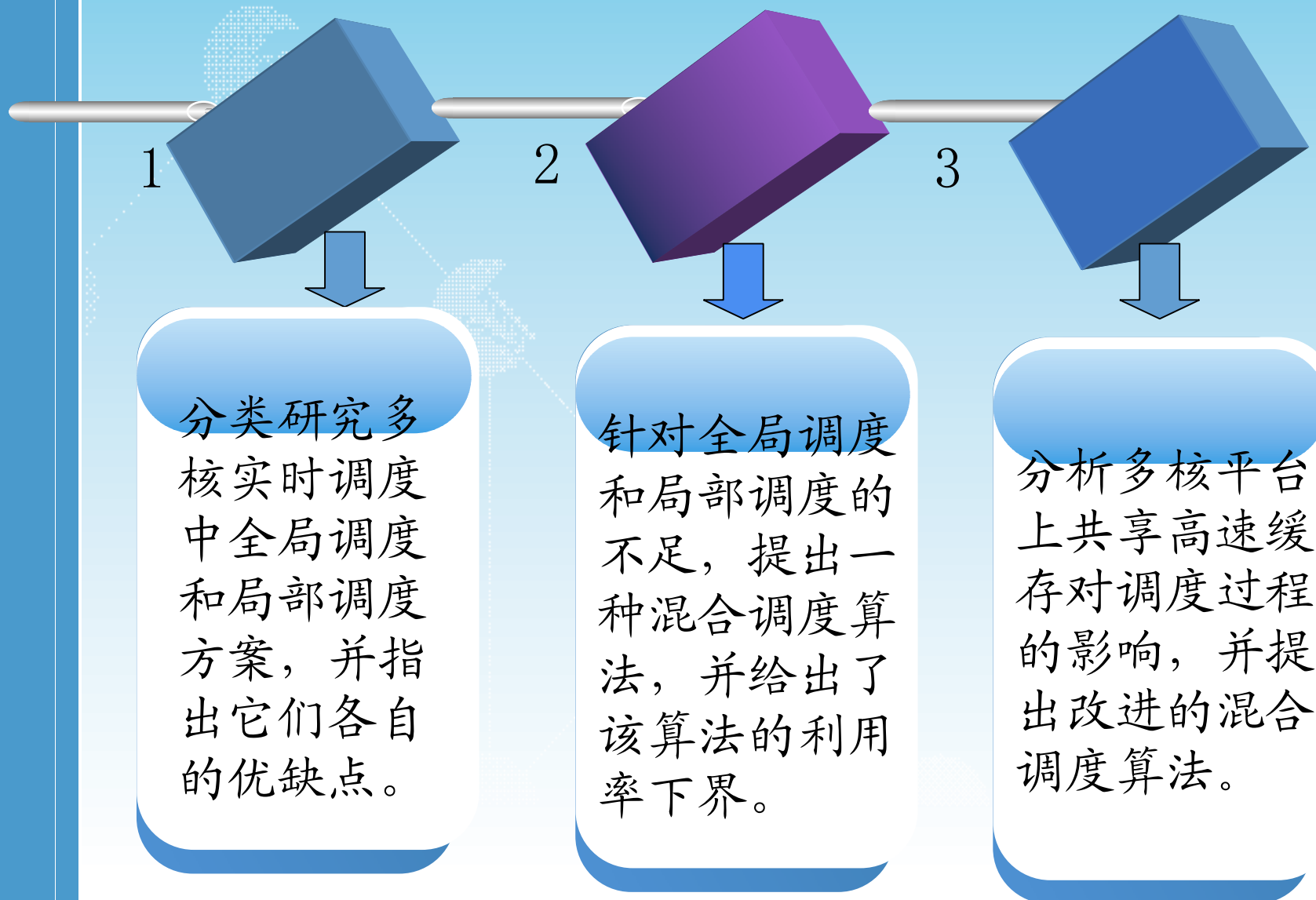
面向共享缓存改进的混合调度算法

总结和展望

选题背景



论文的主要工作



多核实时调度算法

- 多核实时调度分类
 - 局部调度
 - 全局调度

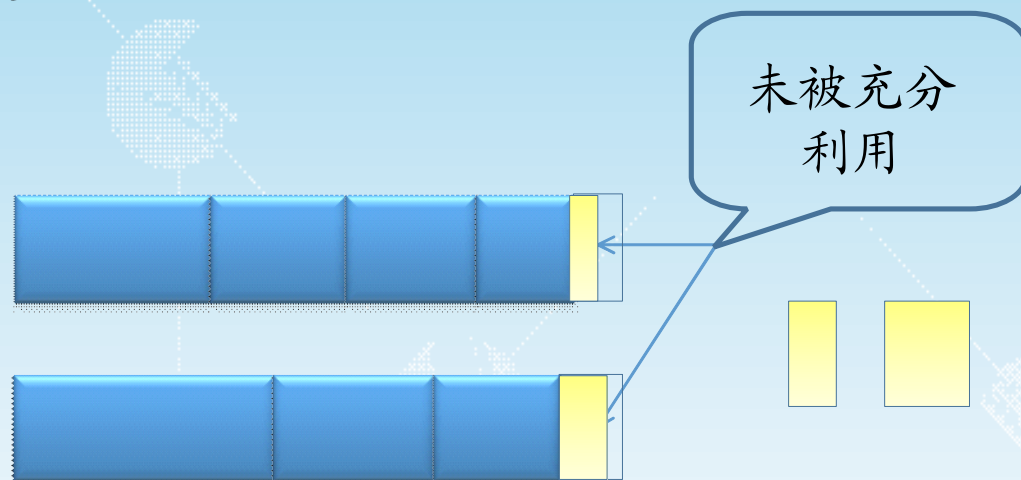
全局调度:

- Dhall效应;
- 迁移导致额外的通信开销和高速缓存失效。
- 处理相似的任务能力被分割了。



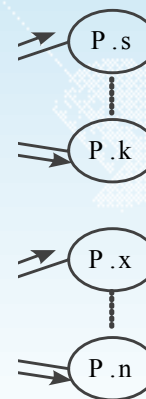
多核实时调度算法

- 混合调度算法
 - 半局部调度
 - 分簇调度



多核实时调度算法

- 混合调度算法
 - 半局部调度
 - 分簇调度



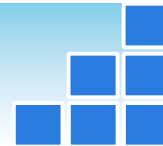
基于簇集的多核混合调度算法

- 分簇调度算法研究

考虑一个包含6个任务的零星任务集： $\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6$
 $\tau_3, \tau_4 = (3, 2, 3)$, $\tau_5 = (6, 4, 6)$ 和 $\tau_6 = (6, 3, 6)$ 。

任务 $\tau = (T, C, D)$

基于簇集的多核混合调度算法



- 算法描述
 - 处理器分组
 - 任务分配

基于簇集的多核混合调度算法

- 算法描述
 - 簇集内任务调度
 - 根据全局EDF-US[ζ]调度算法进行调度

基于簇集的多核混合调度算法

- 利用率分析
- 算法的可调度性下界为：
 - 其中 $\beta = \left\lfloor \frac{s}{\alpha} \right\rfloor$, s 为簇集内全局调度算法的利用率, $s = \frac{m/m'+1}{2}$, $\alpha = \max\{u_i, i=1,2,\dots,n\}$
 - 充分非必要条件
 - 证明方法:
 - 寻找一个最低界限, 当任务集满足时 $U \leq U_{\text{HEDF}}$, 是可调度的;
 - 当 $U > U_{\text{HEDF}}$ 时, 至少存在一个任务集是不能调度的

基于簇集的多核混合调度算法

- 与全局调度的比较分析

$$\frac{(m + m')(m'\beta + 1)}{(2\beta + 1)} > \frac{m + 1}{2}$$

- 与局部调度的比较分析

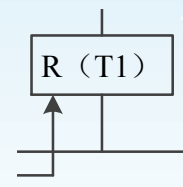
$$U_p - U_h = (m - m')(\frac{1}{2} - \alpha)$$

- $\alpha > 1/2$, 混合调度利用率高;
- $\alpha < 1/2$, 局部调度利用率高。

面向共享缓存改进的多核实时调度算法

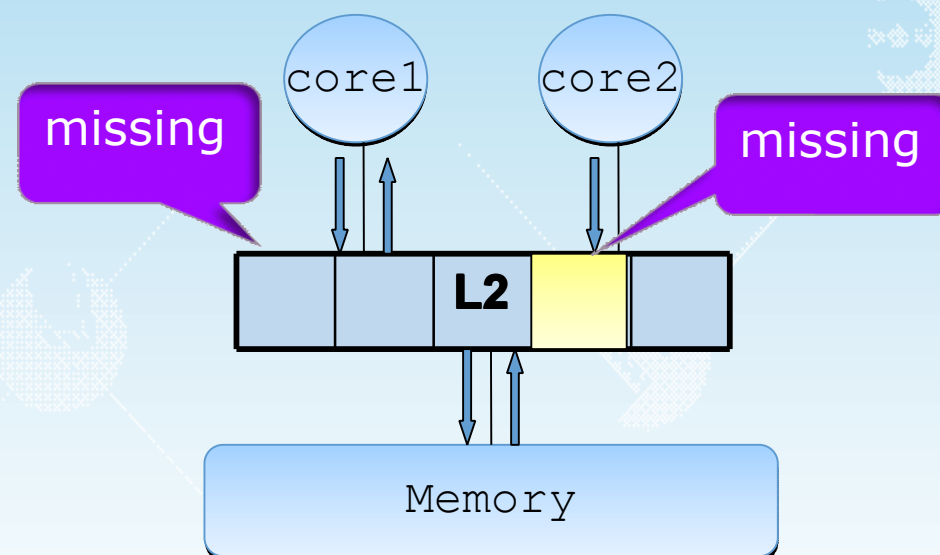
- 一、高速缓存对多核实时调度的影响
 - 对于私有的Cache结构

T2 T1



面向共享缓存改进的多核实时调度算法

- 一、高速缓存对多核实时调度的影响
 - 对于共享的Cache结构，多核架构中各个核竞争使用单一的共享Cache。



面向共享缓存改进的多核实时调度算法

• 二、系统模型

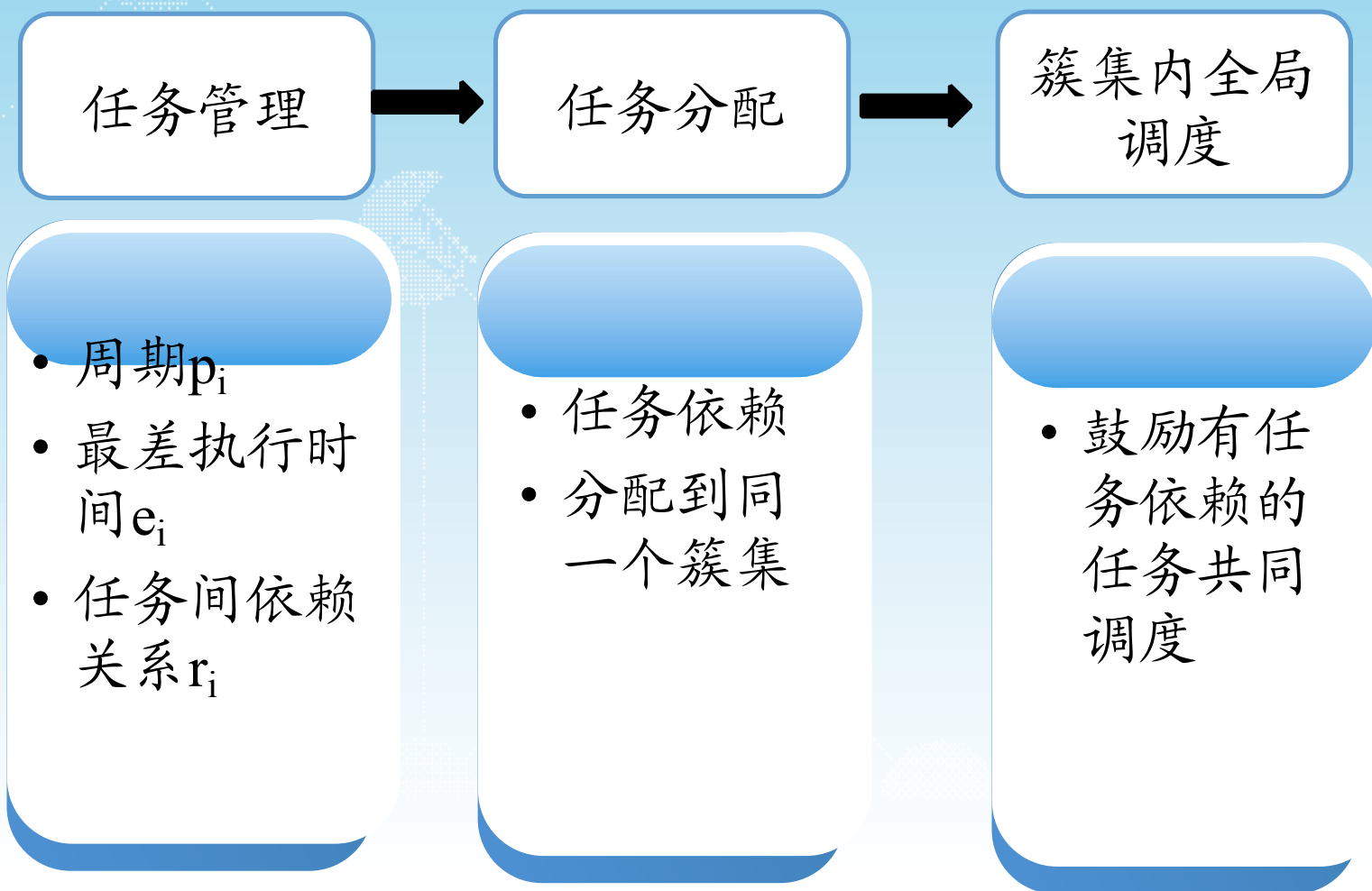
- 软实时应用.
- 处理器模型
 - 同构多核
 - 所有的核共享三级缓存，部分核共享二级缓存，单线程
- 任务模型
 - 周期任务 $T_i (e_i, p_i) (1 \leq i \leq n)$
 - 参数:
 - 截止时限
 - 最差执行时间
 - 可抢占的

符号 $r_i (dep_i, num_i)$:

- 当两个任务的 dep ($dep \neq 0$) 值相同时，表示两个任务存在数据依赖；
- num 值为任务之间数据依赖的程度。

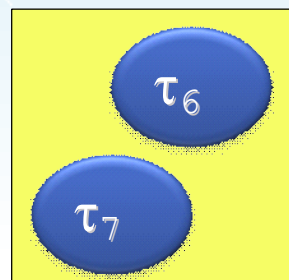
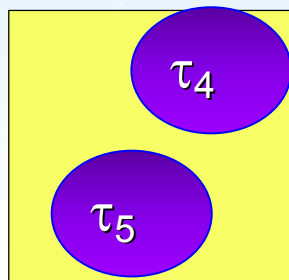
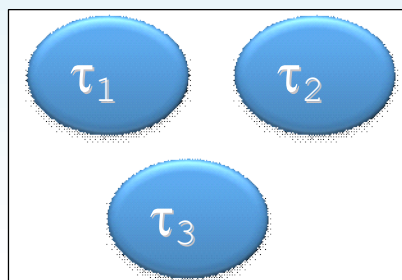
面向共享缓存改进的多核实时调度算法

• 调度过程



面向共享缓存改进的多核实时调度算法

- 三、算法实现
- 任务分配
 - 任务合并;
 - 将所有的任务和合并成的大任务按照CPU的利用率 $\beta = \sum e_i/p_i$ 降序排列
 - 依次将排序的任务分配给核
 - 找到负载最小的核
 - 将当前任务分配给这个核



WFD

面向共享缓存改进的多核实时调度算法

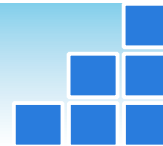
- 簇集内全局调度

- 紧急任务：当一个任务 T_i 被调度了，就绪队列中任务 T_j 的 dep_j 与任务 T_i 的 dep_i 值相等，则将 T_j 被标为紧急任务；
- 抢占任务：任务 $T_i (e_i, p_i, r_i)$ ，当前时间为 t ，若 $p_i - e_i \leq t + \varepsilon$ （ ε 为无穷小），则 T_i 是紧急任务
- 优先级
 - 抢占任务
 - 紧急任务
 - 就绪任务

鼓励共同调度

保证实时性

面向共享缓存改进的多核实时调度算法



- 四、仿真测试
- 处理器及任务集描述
 - 8个相同的核;
 - 每两个处理器核共享一个8路全相联512k二级高速缓存;
 - 高速缓存采用LRU替换策略。



面向共享缓存改进的多核实时调度算法

- 仿真实验分为三个部分:

- 生成任务集;

任务参数	生成情况
截止期 D_i	取值范围为 $[10, 100]$ 的随机整数
利用率 u_i	服从参数为 λ 的指数分布, λ 的取值范围为 $[8/n, 9/n]$
读取资源 c_i	取值范围为 $[1, 100]$ 的随机3个整数

- 任务分配算法, 算法将任务集中的任务分配到各个处理器簇集;

- 簇集内的全局调度算法的实现。

面向共享缓存改进的多核实时调度算法

实验结果及分析

表1 采用不同分配算法时Cache失效率测试结果

α	WFD (%)	HWFD (%)	命中率提高 (%)
80%	20.51	16.99	3.52
60%	22.55	18.71	3.84
40%	23.18	19.25	3.93
平均	22.08	18.32	3.76

表2 簇集内采用不同调度算法时Cache失效率测试结果

α	全局EDF (%)	caEDF (%)	命中率提高 (%)
80%	16.99	13.92	3.07
60%	18.71	15.69	3.02
40%	19.25	16.05	3.20
平均	18.32	15.22	3.10

总结与展望

• 总结

- 提出了一种基于EDF-US [1/2] 调度算法的混合调度算法，对算法的可调度性进行了分析并给出了算法利用率界限。
- 提出了一种面向共享缓存改进的混合调度算法，给出了模拟测试方案，实验证明了该算法能够有效提高共享高速缓存的命中率。

• 展望

- 针对基于共享缓存的混合调度算法，本文仅对算法进行了模拟实验，在今后的工作中，可以将算法应用到具体的多核实时操作系统中，从而在实践中对算法进行更进一步的分析。



湖南大学

HUNAN UNIVERSITY

感谢您的指导！！！！