

本节内容

多处理器 调度

知识总览

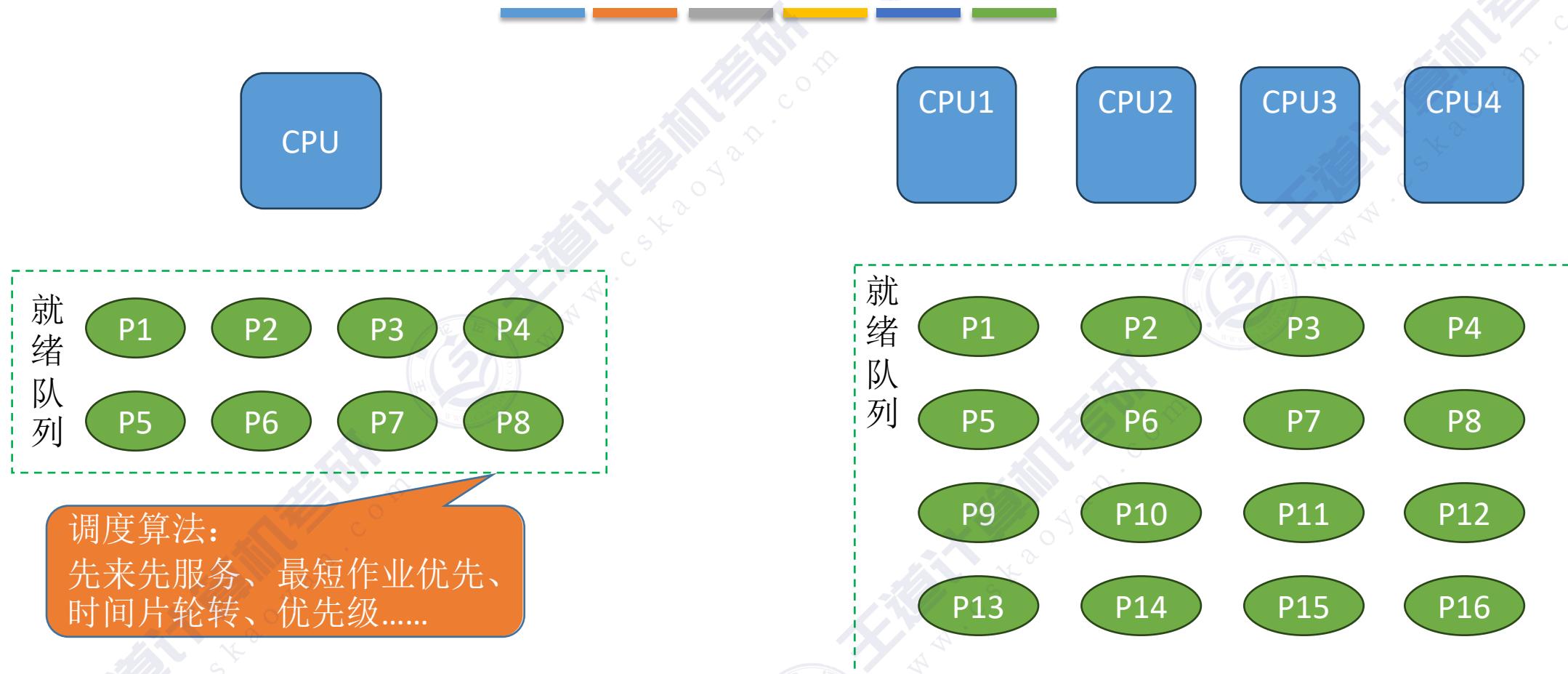
多处理机调度

与单处理机调度相比，需要面临哪些新问题？

方案一：公共就绪队列

方案二：私有就绪队列

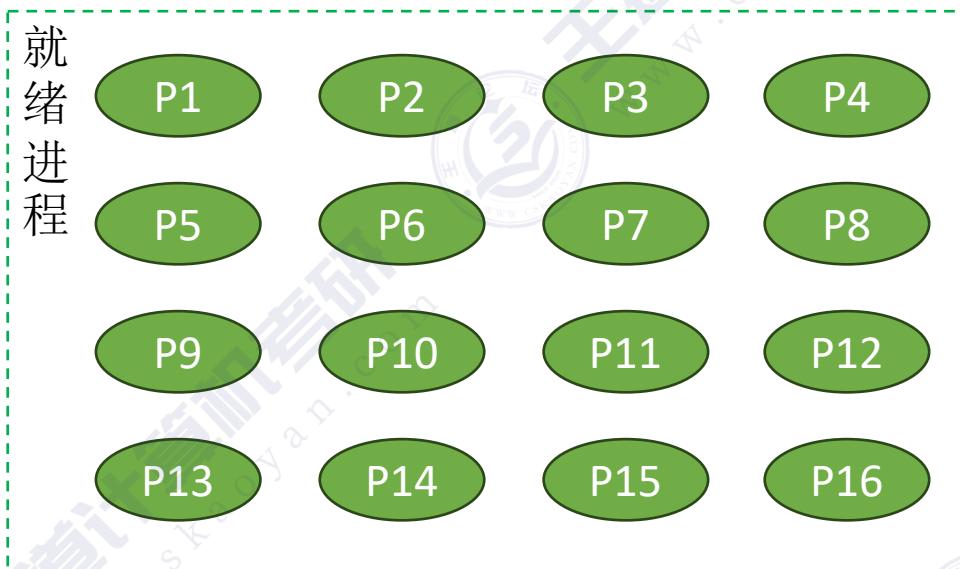
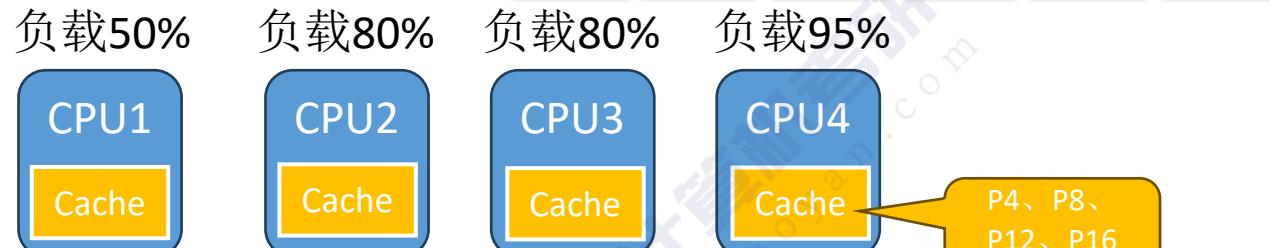
单处理机调度 vs 多处理机调度



单处理器调度：只需决定让哪个就绪进程优先上处理器即可。

多处理器调度：①用调度算法决定让哪个就绪进程优先上处理器；②还需决定被调度的进程到底上哪个处理器

负载均衡、处理机亲和性



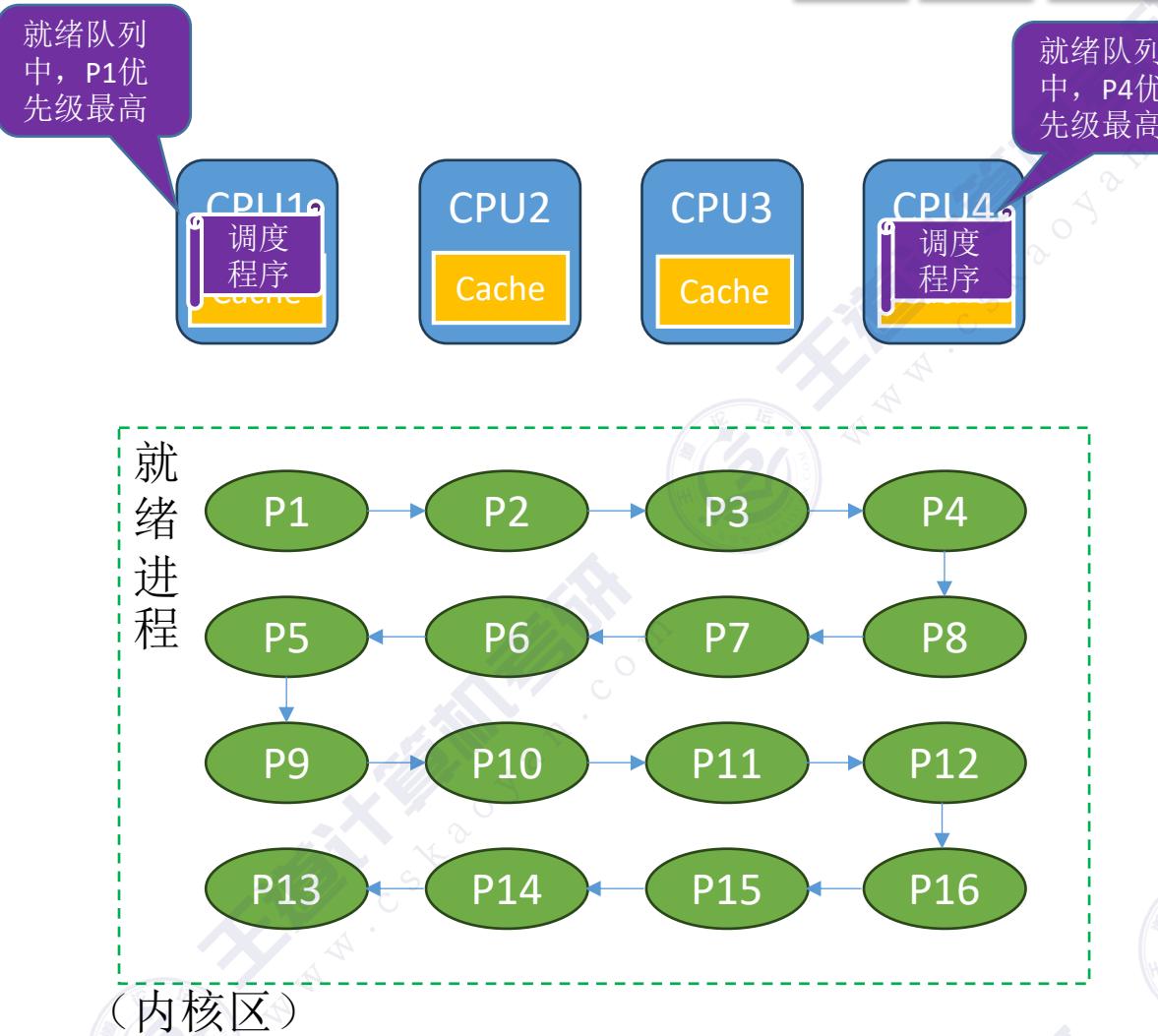
多处理机调度: ①考虑让哪个就绪进程优先上处理器；②还需考虑上哪个处理器运行

多处理机调度中，应追求的目标：

负载均衡——尽可能让每个CPU都同等忙碌

处理机亲和性——尽量让一个进程调度到同一个CPU上运行，以发挥CPU中缓存的作用（Cache）

方案一：公共就绪队列



公共就绪队列

- 所有CPU共享同一个就绪进程队列（位于内核区）
- 每个CPU时运行调度程序，从公共就绪队列中选择一个进程运行
- 每个CPU访问公共就绪队列时需要上锁（确保互斥）

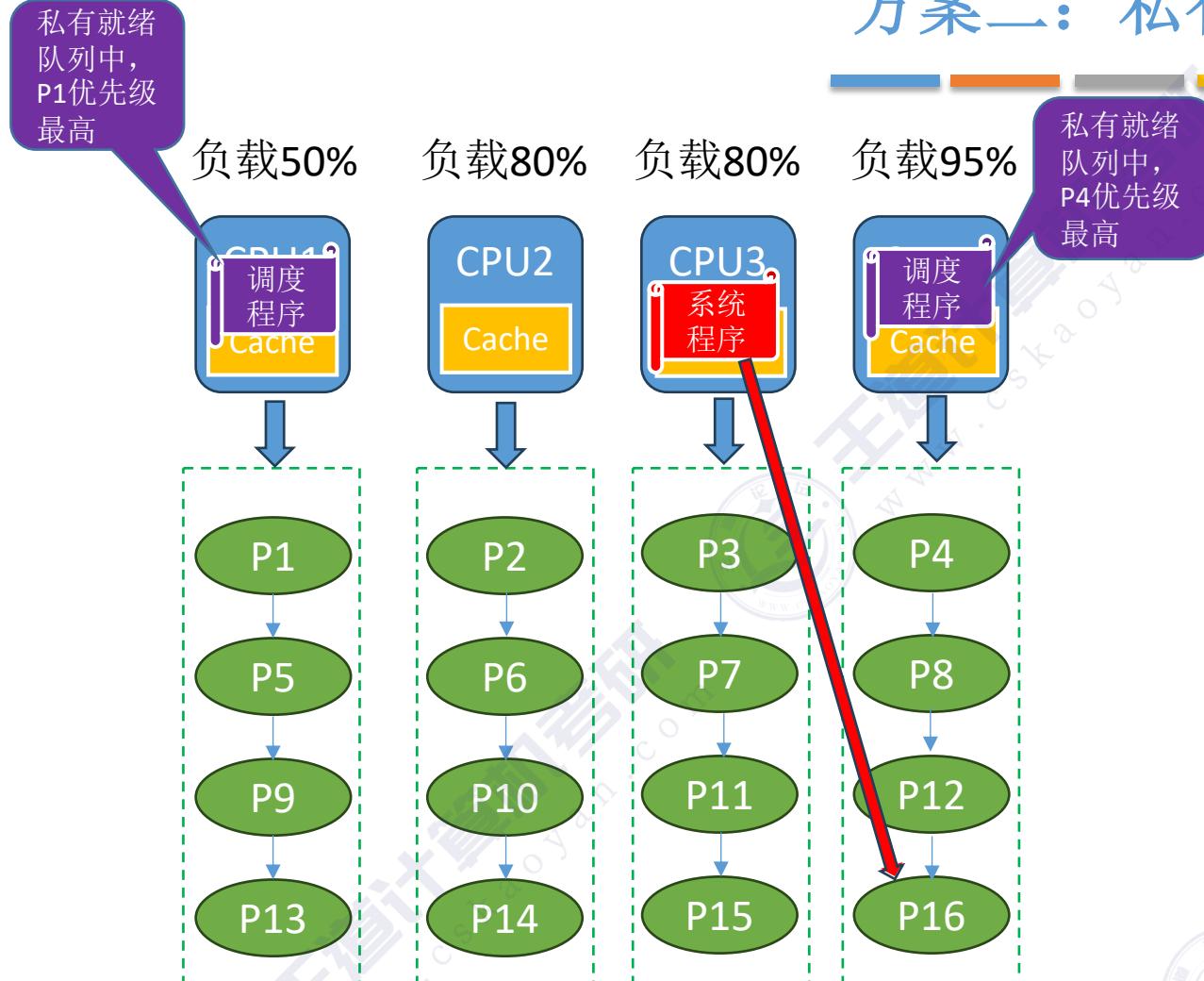
优点：可以天然地实现负载均衡

缺点：各个进程频繁地换CPU运行，“亲和性”不好

如何提升处理机亲和性？

- 软亲和：由进程调度程序尽量保证“亲和性”
- 硬亲和：由用户进程通过系统调用，主动要求操作系统分配固定的CPU，确保“亲和性”

方案二：私有就绪队列



私有就绪队列

- 每个CPU都有一个私有就绪队列
- CPU空闲时运行调度程序，从私有就绪队列中选择一个进程运行



目标一：如何实现负载均衡？

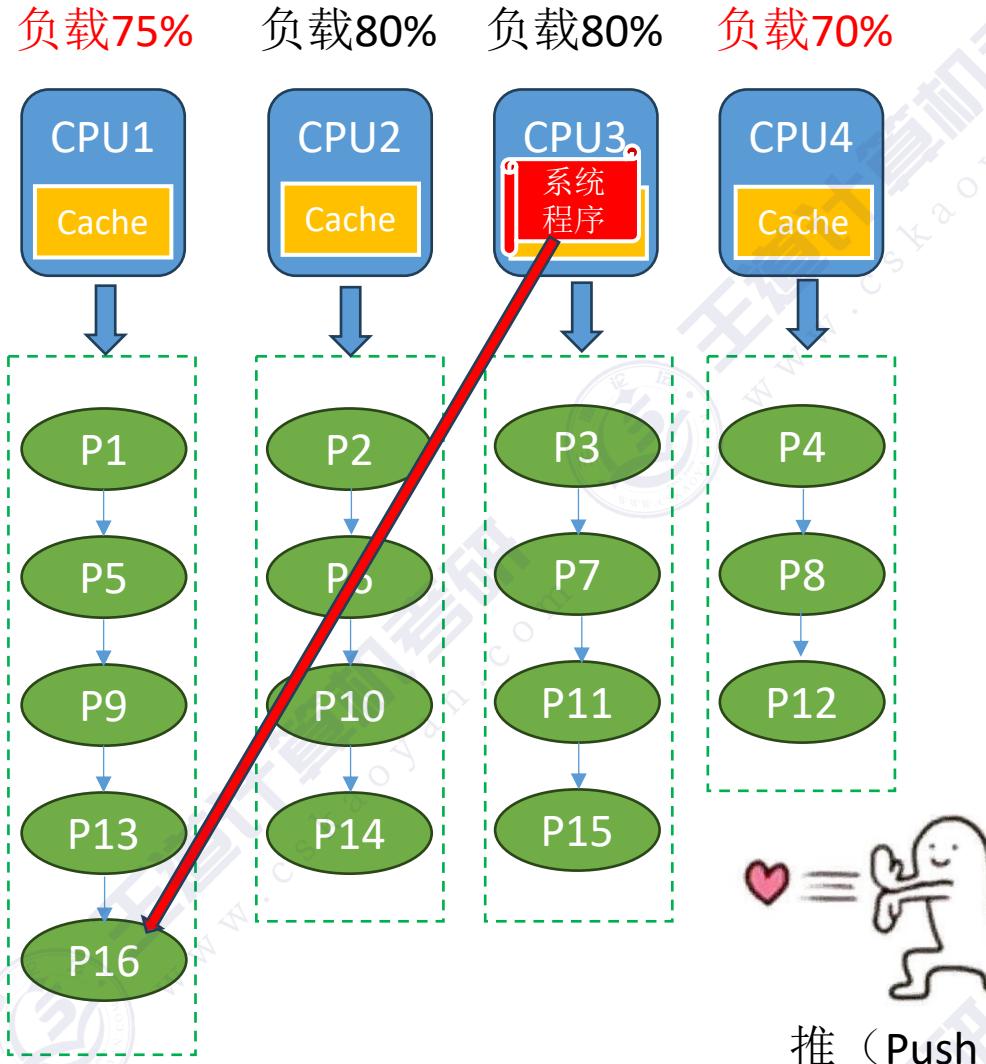
推迁移（Push）策略：

一个特定的系统程序周期性检查每个处理器的负载，如果负载不平衡，就从忙碌CPU的就绪队列中“推”一些就绪进程到空闲CPU的就绪队列



推（Push）

方案二：私有就绪队列



私有就绪队列

- 每个CPU都有一个私有就绪队列
- CPU空闲时运行调度程序，从私有就绪队列中选择一个进程运行



目标一：如何实现负载均衡？

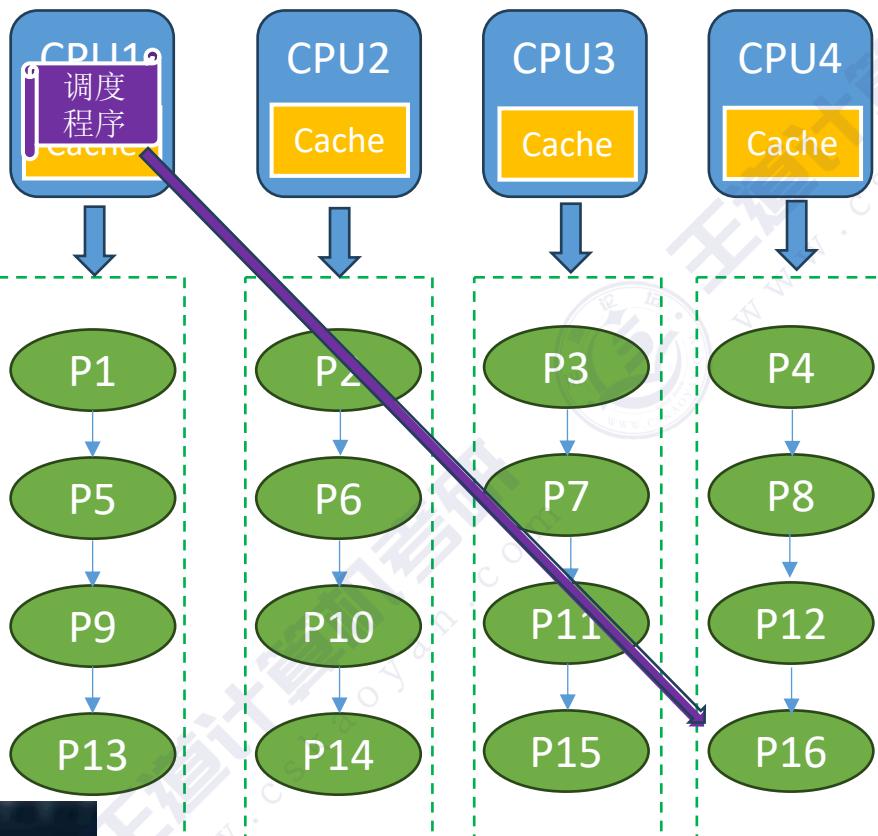
推迁移（Push）策略：

一个特定的系统程序周期性检查每个处理器的负载，如果负载不平衡，就从忙碌CPU的就绪队列中“推”一些就绪进程到空闲CPU的就绪队列

推迁移 = 有一个包工头专门负责派活

方案二：私有就绪队列

负载50% 负载80% 负载80% 负载95%



你过来呀
拉 (pull)

私有就绪队列

- 每个CPU都有一个私有就绪队列
- CPU空闲时运行调度程序，从私有就绪队列中选择一个进程运行



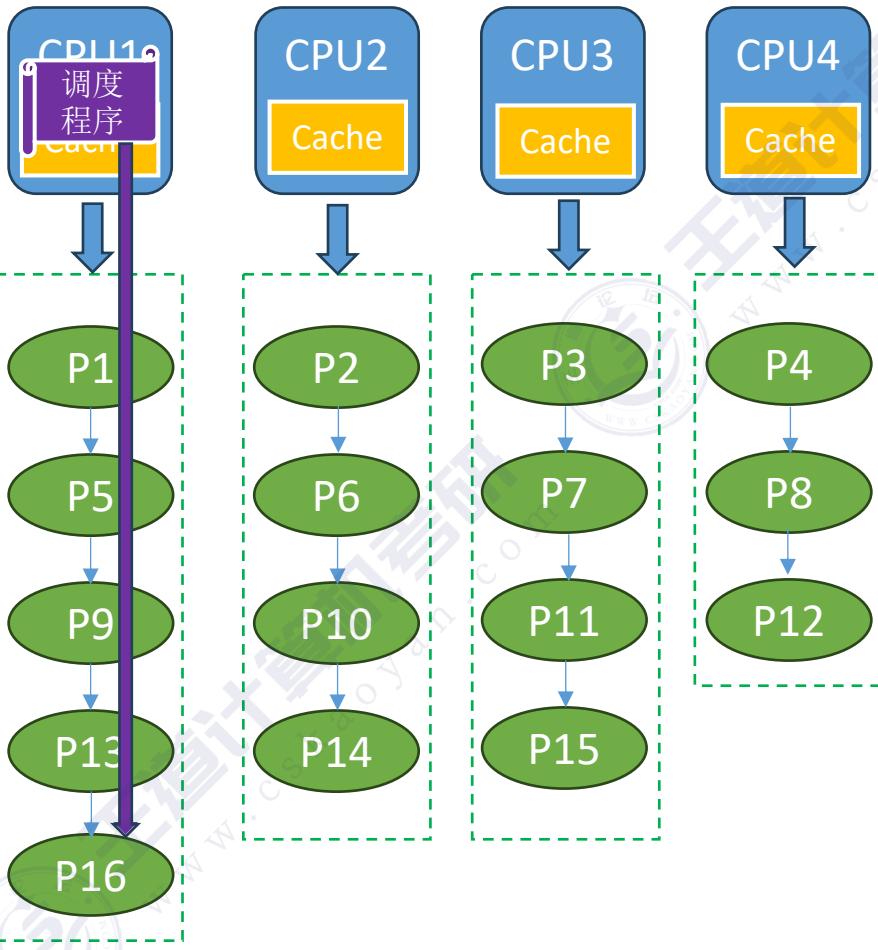
目标一：如何实现负载均衡？

拉迁移 (pull) 策略：

每个CPU运行调度程序时，周期性检查自身负载与其他CPU负载。如果一个CPU负载很低，就从其他高负载CPU的就绪队列中“拉”一些就绪进程到自己的就绪队列

方案二：私有就绪队列

负载75% 负载80% 负载80% 负载70%



私有就绪队列

- 每个CPU都有一个私有就绪队列
- CPU空闲时运行调度程序，从私有就绪队列中选择一个进程运行



目标一：如何实现负载均衡？

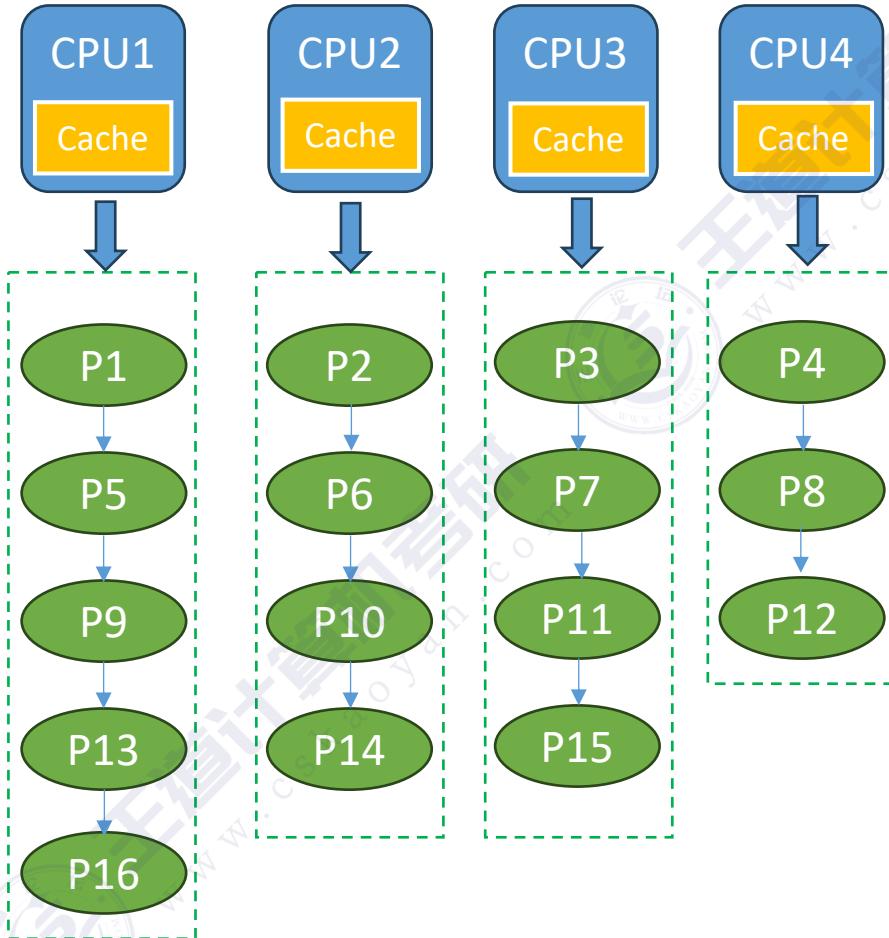
拉迁移（pull）策略：

每个CPU运行调度程序时，周期性检查自身负载与其他CPU负载。如果一个CPU负载很低，就从其他高负载CPU的就绪队列中“拉”一些就绪进程到自己的就绪队列

拉迁移 = 一群互帮互助的同事（看到其他同事很忙，主动揽活过来，分担任务）



方案二：私有就绪队列



私有就绪队列

- 每个CPU都有一个私有就绪队列
- CPU空闲时运行调度程序，从私有就绪队列中选择一个进程运行



目标一：如何实现负载均衡？

目标二：如何实现处理机亲和性？

如何提升处理机亲和性？

- 私有就绪队列天然地实现了“处理机亲和性”

知识回顾与重要考点

