作业二：论文阅读

2113301 朱霞洋

论文的主要贡献

这篇论文创新地构建了一种名为PARTIES的资源管理系统，用于在云计算中解决多个互动性、延迟关键（LC）的服务在同一节点上运行时的相互干扰和资源分配问题。其利用硬软件资源分区机制，不需要任何调度的先验信息，就可以在运行时动态调整资源分配，对延迟最高的LC服务增加分配，能让多个延迟关键的服务与多个尽力（Best Effort）服务同时工作，并且不违反服务质量保证（QoS），相较于现有的资源管理器，PARTIES平均提高了61%的服务质量下的吞吐量。

PARTIES方法存在的缺陷

尽管PARTIES创新地解决了多个LC服务和其他作业运行在同一服务器上的资源分配问题，但是这样的方法仍然存在一些缺点，个人认为有：

1. **PARTIES资源的开销**：PARTIES是一个实时的资源调度系统，也会在物理主机上与其他服务一同运行，因此PARTIES资源管理系统会占用一定的资源。在论文的5.4节中，作者提到PARTIES在0号内核上运行时，CPU占用率为15%，（检测和调整分别占10%与5%），这意味着在一些性能较弱的节点上（例如内存较小或CPU较弱），PARTIES或许会造成一定的负担；此外，PARTIES需要频繁的访问各个并发服务的延迟信息和资源分配信息（论文中是每500ms进行一次检测），这也会引发额外的开销；

2. **需要精细的监测和调整**：PARTIES系统通过在线监测和调整资源分配来实现QoS保证，这意味着系统需要对运行的应用程序进行实时监测，并进行细粒度的资源调整，这需要消耗一定的计算资源和运行时间，对设备也会有一定要求；

3. **极端场景下的缺陷**：当服务器总负载非常高，调整资源时只有少数几个可行的方案时，总是优先考虑latency slack最小的服务可能会导致乒乓效应（在两个不同的状态之间来回变化）；并且，当某个应用的资源非常紧张时，任何调整都需要 500 毫秒以上才能生效，尤其是在调整计算资源时，因为此时系统中已经积聚了很长的应用队列。这可能会让PARTIES 在计算轮中过度分配资源。增加监控间隔可以解决这一问题，但会增加收敛时间（5.2节）

对PARTIES改进空间的思考

在读完论文后，个人认为文章提出的PARTIES方法可以改进的主要点包括：

1. **针对其他类型服务的支持**：论文重点介绍了PARTIES对于多个交互式、延迟关键（LC）服务的资源分配的上的应用，但对于其他类型的服务（如批处理任务）的支持可能仍有改进空间，可以在这方面扩展PARTIES，以支持更多类型的服务将使其更加全面和灵活，达到更好的效率；

2. **性能的优化**：上文中提到，PARTIES在运行时也会产生资源开销，例如在5.4节中，PARTIES在0号内核会达到15%的利用率，可以对PARTIES进行一些优化，使其占用率降低以达到更好的性能；

3. **更快的收敛速度**：当前的PARTIES在运行时不需要任何有关调度的先验信息，会随机选择要调度的初始资源（4.2.3节），尽管PARTIES可以在若干次调度后较快收敛，但在一些情况下仍需要几十秒的时间（5.4节），个人认为可以加入一些关于调度的先验信息，帮助PARTIES更快的找到最佳决策；此外文章提到了

资源可交换性的概念，即某些资源可以相互交换以实现等效的应用程序性能，我认为进一步优化资源可交换性的算法和策略可能是一个改进点；

4. **算法的参数的优化**：PARTIES在对各应用的延迟检测，以及回收应用程序的资源时的标准目前是固定的，会每间隔500ms进行延迟检测，也只会在某服务的latency slack大于0.2时开始回收资源，这一点上个人认为还有优化空间。在论文的4.3节的`How are the controller parameters determined?`部分，作者提到更短的监控间隔能更快地检测出违反 QoS 的情况，但是也可能导致结果不稳定，因为没有积累足够的再查询次数来使尾延迟收敛；此外，在上文也提到，当只有少数几个可行的调整资源方案时，总是优先考虑latency slack最小的服务可能会导致乒乓效应。这些情况说明PARTIES的分配算法仍有优化的空间，例如使其latency slack参数和监控间隔参数动态的变化等等，让分配算法更加灵活。

综上所述，尽管PARTIES是一种资源管理系统的创新解决方案，但仍有一些可以改进的方面，如更广泛的资源管理机制、支持其他类型服务和进一步优化收敛速度等。