分　数：	
评卷人：	

華中科技大學

研究生（数据中心技术）课程论文（报告）

题　目：云计算集群调度策略

学　　号　M202173706

姓　　名　黄志颖

专　　业　电子信息

课程指导教师　施展 童薇

院（系、所）　计算机科学与技术学院

2022 年 1 月 3 日

云计算集群调度策略

黄志颖[1)]

[1)](华中科技大学计算机科学与技术学院，武汉 430074)

摘　要:　近年来，在云计算环境下，计算机集群的规模显著增长。今天，一个集群可能有 10 万台机器，每天执行数十亿个用户上传的任务，特别是短任务。因此，管理集群中资源利用率的调度器也需要升级，以便在更大范围内工作。然而，在大型生产集群中升级调度程序（一个中央系统组件）是一项艰巨的任务，因为我们需要确保集群的稳定性和健壮性，例如，应保证用户透明性，其他集群组件和现有调度策略需要保持不变。我们调查了现有的调度器设计，研究了三篇文献，Yihui Feng 等人提出了一种基于分区的同步策略的调度框架 ParSync，采用分区同步和自适应调度策略以减少调度延迟和提高调度值质量；Kshiteej Mahajan 等人介绍了一个新的 machine learning 训练工作负载调度框架 THEMIS，他们引入的一个新概念，GPU 分配策略强制 ML 工作负载以完成时间公平的方式完成；DamianFernández-Cerero 等人提出了一个新的云计算仿真工具 GAME-SCORE，它实现了一个基于 Stackelberg 博弈的调度模型。

关键词：云计算；大型集群；调度器；公平；效率

Distributed storage system optimization model

Huang ZhiYing[1)]

[1)](School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074)

Abstract　Data In recent years, in the cloud computing environment, the scale of computer clusters has increased significantly. Today, a cluster may have 100000 machines that perform tasks uploaded by billions of users every day, especially short tasks. Therefore, the scheduler that manages resource utilization in the cluster also needs to be upgraded to work on a larger scale. However, upgrading the scheduler (a central system component) in a large production cluster is a difficult task, because we need to ensure the stability and robustness of the cluster. For example, we should ensure user transparency, and other cluster components and existing scheduling strategies need to remain unchanged. We investigated the existing scheduler design and studied three literatures. Yihui Feng et al proposed a scheduling framework parsync based on partition synchronization strategy, which adopts partition synchronization and adaptive scheduling strategy to reduce scheduling delay and improve the quality of scheduling value; Kshiteej Mahajan et al. Introduced a new machine learning training workload scheduling framework Themis. They introduced a new concept, GPU allocation strategy, which forces ml workload to be completed in a time fair manner; Damian fern á ndez cerero et al. Proposed a new cloud computing simulation tool game-score, which implements a scheduling model based on Stackelberg game.

Keyword　Cloud computing; Large clusters; Scheduler; Fairness; efficiencyata center； distributed storage； distributed database； blockchain

一、引言

在当前互联网企业中，我们运营着大型集群，每个集群包含数万台机器。每天，集群中都有数十亿个任务被提交、调度和执行。集群调度器(或简称调度器)管理集群中的机器和任务。调度器根据任务的资源需求(如CPU、内存、网络带宽等)，通过各种调度算法将任务匹配到合适的资源，并在调度效率、调度质量、资源利用率、公平性、任务优先级等多个调度目标之间进行复杂的权衡。平衡这些目标的能力在很大程度上取决于技术和业务因素，因此因公司和集群而异。由于近年来我们业务的快速增长，我们在扩展调度程序方面面临着严峻的挑战，因为我们的集群中有更多的任务和机器。今天，我们的一些集群的规模接近100k台机器，平均任务提交率约为40k个任务/秒(在某些月份还会更高)。这种规模仅仅超过了单个调度器的容量，升级到分布式调度器体系结构是不可避免的。

然而，之前的单主架构无法处理当前集群的规模。首先，10倍以上的机器数量需要从一个workers到主机的两个连续的心跳消息之间有更大的时间间隔，这样位于所有决策的关键路径上的主机就不会超载。但是，更大的间隙更有可能使空闲的机器在间隙期间有更长的一段时间没有使用。其次，任务数量明显增多，超出了单个调度器的能力。具体来说，我们之前的调度器只能处理每秒几千个任务的任务提交速率。我们的架构应该要能够处理集群规模和任务提交率的规模，同时实现低延迟和高质量的调度。该设计还应满足两个硬约束:向后兼容性和无缝用户透明性。从这些约束中得到的调度程序的健壮性和稳定性对今天我们的生产环境至关重要。

在本综述中，我们将介绍几种分布式调度器架构的设计。如何对云计算环境下的大型生产集群进行更有效地调度，这一研究方向有很大的意义，在综述的三篇文章中，Yihui Feng等人提出了一种基于分区的同步策略的调度框架ParSync，其实验结果表明在其使用的分区同步手段和自适应策略可以有效降低调度延迟和提高调度质量。Kshiteej Mahajan等人介绍了一个新的machine learning训练工作负载调度框架THEMIS，他们引入的一个新概念，GPU分配策略强制ML工作负载以完成时间公平的方式完成。为了捕获放置敏感性并确保效率，THEMIS使用了两级调度体系结构，其中ML工作负载对由中央仲裁者运行的拍卖中提供的可用资源进行出价。他们的拍卖设计将GPU分配给中标人，在短期内以公平换取效率，但在长期内确保完成时间的公平性。我们对生产跟踪的评估表明，与最先进的调度程序相比，THEMIS可以将公平性提高225倍以上，集群效率提高约5%到250%。DamianFernández-Cerero等人提出了一个新的云计算仿真工具GAME-SCORE,它实现了一个基于Stackelberg博弈的调度模型。这个游戏展示了两个主要玩家:a)调度器和b)节能代理。他们使用GAME-SCORE模拟器来分析所提出的基于博弈的调度模型的效率。结果表明，Stackelberg云调度器的性能优于静态能量优化策略，能够在很短的时间内实现低能耗和短完工时间之间的平衡。

二、原理和优势

2.1 ParSync

他们发现调度延迟在G周期内不成比例地增加。当集群状态刚刚同步时，集群状态更新鲜，调度冲突更少。但是当状态在G的末尾变得更加过时时，调度决策就会导致更多的冲突。冲突导致重新调度，而重新调度又可能导致新的冲突，从而递归地重新调度。考虑同步间隙和Sextra = 15,000中他们的默认模拟设置，将G分为两个相等的间隔:0s-0.25s和0.25-0.5s。第一个间隔的平均冲突率为48%，第二个间隔的平均冲突率增加到64%。因此，第一个间隔的平均延迟只占总平均延迟的23%，而第二个间隔占总延迟的76%。换句话说，大部分延迟是由于在G的较晚时间间隔内集群状态的陈旧视图造成的。

他们的主要想法是减少地方状态的陈旧性，并在资源质量(即槽位分数)和调度效率之间找到一个良好的平衡，因为根据他们

的发现，这两者是冲突的主要贡献者。他们给出了解决方案，分区同步(ParSync)，如下所示。

ParSync 在逻辑上将主状态划分为 P 个部分。假设我们有 N 个调度器，其中 N≤p。每个调度器仍然保持一个本地集群状态，但不同的调度器每次同步状态的不同分区，而不是像讨论的那样同步整个状态(让我们将其表示为 StateSync)。具体来说，假设(为简单起见)P 是 N 的倍数，第 i 个调度程序启动的同步第$\left(\frac{P}{N}*(i-1)+1\right)$分区到第$\left(\frac{P}{N}*i\right)$分区在每一轮的同步，以及随后轮同步持续循环的方式。因此，在每一轮中，所有调度器同步集群状态的不同分区。这也是我们要求 P≥N 的原因，否则一些调度器将同步相同的分区。

首先，每个调度器都有集群资源的$\frac{P}{N}$个分区的新视图，因此调度器可以以较高的成功率将任务提交到这些分区中的可用槽中(因为它对这些分区的视图比任何其他调度器的视图都要新鲜)。这种设计特别适合在像他们们这样竞争激烈的集群中调度短任务。对于短期任务，更快更关键的获取资源的执行，而不是花费长时间找到最合适的插槽，因为更好的槽可能不补偿调度延迟(这本身可以比得上甚至超过一个简短的任务的执行时间)。注意，我们工作负载中的大多数任务都是短任务，如图 1 所示。其次，ParSync 还有效地降低了本地集群状态中条目的平均陈旧程度，我们将在下面解释这一点。由于同步的粒度从整个集群状态更改为$\frac{P}{N}$个分区，从而有效地将同步差距更改为$\frac{G}{N}$。与在每个 G 时间单位同步整个状态相反，使用 ParSync，每个调度器在每个$\frac{G}{N}$时间单位同步其本地状态的$\frac{P}{N}$个分区。现在，让我们将一个分区的陈旧状态定义为自其最近一次同步以来的一段时间，并让 as 为本地状态的所有分区的平均陈旧状态。无论是否使用 ParSync，“AS”在所有时间点上的平均值总是$\frac{G}{2}$。然而，ParSync 减少了“AS”的方差。

ParSync 通过采用自适应调度策略，允许我们更好地平衡资源质量和调度效率。例如，当集群不是处于密集使用状态时，冲突不太可能发生，因此调度器可能会承担更高的重新调度风险，并尝试将任务提交到其他陈旧分区(而不是其最新的$\frac{P}{N}$分区)的高质量槽中。在这种情况下，ParSync 可以采用乐观调度来优先考虑资源质量(即插槽分数)。相反，以成功提交为目标的调度策略可能采用悲观的方法(但所有调度器同时工作)。此外，每个调度器仍然具有集群的全局视图，因此也可以实现全局调度策略。

ParSync，它将生产集群的调度能力从在数千台机器上每秒几千个任务提高到在 100K 机器上每秒 40K 个任务。ParSync 有效地减少了资源竞争冲突，实现了低调度延迟和高调度质量。ParSync 的简单性允许我们保持用户透明性和向后兼容性，这对我们的生产集群来说是至关重要的。

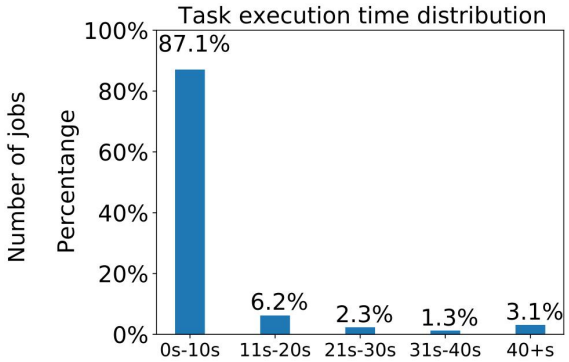

图 1 任务执行时长

2.2 THEMIS

在 THEMIS 的调度架构中，两级调度由多个应用程序调度程序和一个跨应用程序调度程序组成，我们称之为仲裁程序。仲裁者有我们的调度逻辑。每一个应用程序的顶级调度器被最小限度地修改，以与仲裁器交互。图 2 显示了 THEMIS 的体系结构。THEMIS 管理的集群中的每个 GPU 都有一个与之关联的租约。该租期决定了应用的 GPU 的所有权期限。当租期到期时，该资源被创建为可用并分配。

THEMIS 的仲裁池可用资源，并运行前面描述的一轮拍卖。在每一轮中，资源分配分为 5 个步骤，跨越 2 个阶段(如图 2 所示)：

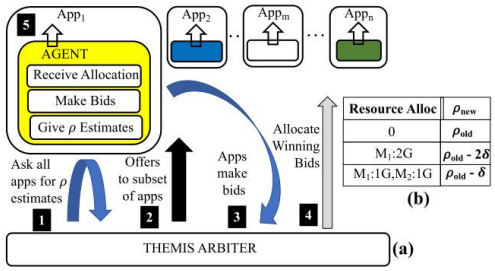

图 2 THEMIS 架构

第一个阶段，称为可见性阶段，跨越步骤 1-3。1.仲裁要求所有应用程序当前完成时间公平度量估计。2.仲裁发起拍卖，并将相同的非绑定资源提供给具有最差完成时间公平指标的 ML 应用程序的 $f \in [0,1]$(根据前面描述的逐轮过滤)。为了尽量减少参与拍卖的 ML 应用程序调度器的变化，THEMIS 引入了一个与每个 ML 应用程序调度器共存的 AGENT。AGENT 充当 ML 应用程序和仲裁程序之间的中介。3.应用程序并行检查资源提供。然后，每个应用程序的 AGENT 都会回复一个包含所需资源分配首选项的报价。

第二阶段，分配阶段，跨越步骤 4-5。4.仲裁者在收到本轮的所有投标后，根据前面所述的部分分配算法和剩余分配方案，选择中标者。然后它通知每个 AGENT 它的获胜分配(如果有的话)。5.AGENT 将分配信息传播给 ML 应用程序调度程序，然后由该程序决定组成作业之间的分配。

总之，两阶段资源分配意味着我们的调度器执行半乐观并发控制。类似于完全乐观的并发控制，多应用可见性，因为跨应用调度器同时为多个应用提供资源。与此同时，与悲观并发控制类似，资源分配是无冲突的，保证每个应用程序对资源的独占访问。为了在步骤 3 中准备竞价，THEMIS 实现了一个从 ML 应用程序调度器到 AGENT 的窄 API，该 API 支持应用程序特定信息的传播。AGENT 的投标包含一个估值函数($\rho(.)$)，该函数为每个资源子集提供应用程序将通过资源子集的分配实现的完成时间公平度量的估计。

实验表明，与现有的调度算法相比，THEMIS 算法能够提高调度的公平性和效率。

2.3 GAME-SCORE

GAME-SCORE 模拟器，它是 SCORE 模拟器的扩展，遵循相同的设计模式:离散事件模拟和多智能体模拟的混合方法。GAME-SCORE 的主要目标是模拟云的节能 IaaS。但该仿真工具存在一个限制，即能耗策略的应用是静态的，可能会对计算集群能耗的降低产生负面影响。因此，实验开始时只应用一个静态能量策略，运行时不能改变。这使得仿真工具对于现实的异构工作负载和不断变化的操作环境(如云计算场景中的操作环境)来说不是最优的。GAME-SCORE 这个模拟工具使我们能够在运行时关闭空闲机器的能效策略目录中进行动态选择。此外，GAME-SCORE 提出了一个基于 Stackelberg 博弈的算法，作为一个实际的用例。然而，任何其他旨在在一组能源效率策略和/或调度算法之间动态切换的策略都可以很容易地通过这个新的仿真工具实现。

GAME-SCORE 基础架构（如图 3）由两个主要模块组成:1 核心模拟器模块，负责执行实验，并由三个子模块组成:a 工作负载生成，负责生成基于统计分布或真实工作负载跟踪的合成工作负载。b 核心引擎，负责创建仿真环境、集群和多个调度代理。这个引擎是实际运行模拟的模块。c 调度模块，负责将任务分配给 worker 节点，以及多个调度框架模型的实现。2 能源效率模块，负责执行基于闲置机器停机的能源效率政策。

我们扩展了这个基础架构，以实现Stackelberg Game过程作为动态切换能源效率政策与能源效率的关系。为此，我们开发了以下模块，它们在调度模块和能效模块之间进行协商：

(a)管理能源效率政策目录的中央能源效率管理模块;(b) Stackelberg-Game管理器，实现基于并发的模型;(c)一个供领导玩家应用其决定的模块;(d)一个模块，用于Follower播放器应用其决策。

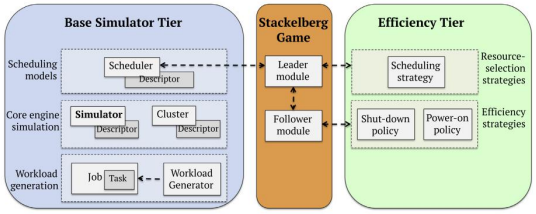

图3 GAME-SCORE的架构

三、研究进展

关于云计算环境下的大型生产集群调度问题，近期的研究热点仍然是工作负载和集群大小带来的可伸缩性挑战，此外，新的调度器还应该在各种调度目标之间实现良好的平衡。我们主要关注以下目标:(1)调度效率或延迟，即任务等待分配所需资源的时间;(2)调度质量，即是否满足任务的资源偏好(例如，存储数据的机器，拥有更大内存或更快cpu的机器);(3)公平性[19]，根据项目的可用资源配额分配一定数量的资源。(4)资源利用。

这些目标常常相互矛盾。例如，高调度质量可能会延长调度延迟;保持严格的公平性可能会使资源未被使用，从而损害利用率。多年来，平衡这些目标的复杂逻辑已经被编程成各种调度策略。此外，其他集群组件(如应用主代理和工作代理)一般由企业的其他团队维护，需要多年的协作努力，才能使它们健壮并按预期运行。因此，新的调度器设计应该尽可能少地改变现有的代码库，以确保整个系统的健壮性和向后兼容性(例如，不需要改变其他集群组件，现有的调度策略可以以相同的方式应用)。

现有的调度器大多采用共享状态调度器架构这个体系结构分别解决了单片、两层和静态分区方法中的限制。首先，共享状态体系结构中的每个调度器都可以针对不同类型的作业运行不同的调度策略，从而避免了软件工程中在一个代码库中维护所有策略和使用多线程解决单片体系结构中的首行阻塞的困难。其次，每个调度器都具有集群的全局视图，从而解决了在Mesos等两级调度器中可见性有限的问题。这使得全局政策(例如，公平和优先级)得以实施。第三，每个调度器可以将任务分配给集群中的任何机器，而不是一个固定的分区子集，这减少了静态分区集群中的资源碎片，实现了更高的利用率。

还有一些分布式调度器不共享集群状态。Sparrow采用批量采样和延迟绑定，提高调度效率和任务完成时间。Tarcil通过动态调整采样大小来扩展Sparrow。Hawk[将长批处理作业分配给集中式调度器(该调度器采用最小等待时间策略)，将短批处理作业分配给一组分布式调度器。Hawk还为短任务预留了一小部分集群，并使用任务窃取来避免短任务被长任务阻塞。Eagle更进一步，主动避免将短任务分配给运行或等待长任务的机器。它还使工作机器能够重复地从一个作业中取出剩余的任务，以模仿最少剩余时间优先的策略。这些调度器的设计也不是通用的，采用它们有以下几个关键问题。首先，它们也像共享状态方法一样依赖于精确的任务持续时间估计。其次，Sparrow没有实现全局策略的集群全局视图，它的策略是针对Spark作业的细粒度任务定制的，这些任务的持续时间在毫秒范围内。Hawk和Eagle专注于使用单一的调度算法来减少同构工作负载的JCT。相比之下，Omega的设计提供了全局集群视图，并允许多个调度器运行不同的调度算法。

调度器考虑了生产集群中的各种问题，如通用性、可扩展性、鲁棒性和资源利用率，而调度器关注的是高可伸缩性和低调度延迟。除了调度器架构外，许多调度算法已经被提出。关注作业内任务调度，优化作业完成时间。作业调度算法,目的是实现目标,如

公平，作业完成时间，和工作负载自动定量也可以采用我们的调度器根据应用程序的需要我们的集群。这些工作并不关注调度程序的架构设计。

四、总结和展望

4.1 总结

本文一共引入了三个最新的大型生产集群调度器：ParSync，一种基于分区的同步策略的调度框架 ParSync。其采用了分区同步的策略，减少了单个调度器的来自心跳消息的开销，此外，引入了自适应调度算法，将调度延迟和调度质量都纳入了考虑范围，将根据具体作业情况来选择侧重点，从而达到更好的效率。接着介绍了 THEMIS，一个公平的 ML 训练工作负荷调度框架。我们展示了现有的公平分配方案如何不足以处理长时间运行的任务和 ML 工作负载的放置首选项。为了解决这些问题，THEMIS 提出了一个新的、长期的公平目标——完成时间公平。然后，THEMIS 提出了一个两级半乐观调度架构，在这个架构中，ML 应用程序可以对拍卖中提供的资源进行竞价。最后介绍了 GAME-SCORE，它实现了一种方法，专注于平衡每个节能CC系统的两个相反需求:高性能吞吐量和低能耗。所提出的仿真工具和模型基于非零和 Stackelberg 博弈，其中，领导者调度经理试图通过其调度决策最小化最大完工时间，而追随者能效经理则通过其调度决策最小化最大完工时间。响应领先玩家的行动，应用节能政策，可能关闭闲置的机器。GAME-SCORE 模型可以根据当前和可预测的工作负载动态应用节能策略。该模型在能效和性能方面均优于单一能效政策的应用。

4.2 展望

我国云计算市场呈爆发式增长。2020 年，我国经济稳步回升， 云计算整体市场规模达 2091 亿元，增速 56.6%。其中，公有云市场 规模达 1277 亿元，相比 2019 年增长 85.2%；私有云市场规模达 814 亿元，较 2019 年增长 26.1%。

我国 SaaS 市场稳定增长，IaaS、PaaS 迎突破。2020 年，我国公有云 SaaS 市场规模达到 278 亿元，较 2019 年增长了 43.1%，受 新冠疫情对线上业务的刺激，SaaS 市场有望在未来几年迎来增长高 峰；公有云 PaaS 市场规模突破 100 亿元，与去年相比提升了 145.3%， 随着数据库、中间件、微服务等服务的日益成熟，PaaS 市场仍将保持较高的增速；公有云 IaaS 市场规模达到 895 亿元，比 2019 年增 长了 97.8%，随着云计算在企业数字化转型过程中扮演越来越重要的角色，预计短期内企业将继续加大基础设施投入，市场需求依然保持旺盛。

技术方面，云原生持续落地，行业应用加速。一是互联网和信息服 务业应用占比显著下降，垂直行业快速崛起。二是云原生技术价值进一步为用户所接受。三是采用云原生架构的生产 集群规模显著提升，但规模化应用带来的安全、性能和可靠性等问题仍需考虑。

架构方面，云网融合需求强，边缘侧潜力大。企业上云扩大云网融合需求。边缘计算需求潜力巨大。因此，在这种背景下，企业对于集群调度问题更是尤为关注，这关系到企业能否更公平、更有效地分配云端资源，也是该行业蓬勃发展的关键。

五、参考文献

[1] Feng Y, Liu Z, Zhao Y, et al. Scaling large production clusters with partitioned synchronization[C]//2021 USENIX Annual Technical Conference (USENIX ATC 21). 2021: 81-97.

[2] Mahajan K, Balasubramanian A, Singhvi A, et al. Themis: Fair and efficient {GPU} cluster scheduling[C]//17th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 20). 2020: 289-304.

[3]Damian Fernández-Cerero, Agnieszka Jakóbik, Alejandro Fernández-Montes, Joanna Kołodziej, GAME-SCORE: Game-based energy-aware cloud scheduler and simulator forcomputational clouds, Simulation Modelling Practice and Theory, Volume 93,2019,Pages 3-20,ISSN 1569-190X,