Stratus: cost-aware container scheduling in the public cloud

Stratus:公共云中成本敏感的容器调度

优化目标：成本

基于运行时间估计

ABSTRACT

Stratus是一个新的集群调度程序，专门用于编排虚拟集群上的批处理作业执行，在公共IaaS平台上动态分配虚拟机实例集合。与传统集群的调度程序不同，Stratus主要关注的是美元成本方面的考虑，因为公共云可以有效地提供无限的、按需分配的高度异构的资源。但是，由于资源在分配时是收费的，Stratus在作业运行时估计的指导下，积极地将任务打包到机器上，试图使分配的资源要么大部分是满的（高度利用），要么是空的（以便释放它们来节省资金）。基于谷歌和TwoSigma集群工作负载跟踪的仿真实验表明，与最先进的虚拟集群调度方法相比，Stratus降低了17-44%的成本。

1 INTRODUCTION

公有云计算已经成熟到许多组织依赖它从传统的内部集群(所谓的“云爆发cloud bursting”)中卸载工作负载，甚至完全取代内部集群。尽管传统的集群调度器可以用来管理公有云虚拟机(VM)实例的静态分配，但这样的安排无法利用公有云的弹性和按需特性，因此会带来不必要的开销。

一种常见的方法[15,36,38,54]是为每个提交的任务分配一个实例，然后在任务完成时释放该实例。尽管很简单，但是这种基于每个任务新实例的方法错过了通过将任务打包到更少甚至更大的实例来降低成本的重要机会。这样做可以提高租用资源的利用率，并能够利用实例类型之间的各种价格差异。

我们需要的是一个虚拟集群(VC)调度器，它将工作打包到实例上，就像传统调度器所做的那样，而不需要假设正在管理一个固定的资源池。这种调度器的关注点与传统集群不同，它增加了资源租用成本，并通过按需获取额外资源的能力消除了排队延迟，而不是强制某些作业等待其他作业完成。将成本最小化需要做出正确的决策，包括

1. 打包哪些任务到实例上
2. 什么时候添加实例
3. 添加什么类型的实例
4. 什么时候释放实例。

Stratus是一个专门针对公共IaaS平台上的虚拟集群的调度程序。Stratus自适应地增长和收缩它分配的实例集，精心挑选以最小化成本和适应高利用率的任务包。为了在一段时间内将成本降到最低，Stratus努力使每个实例的利用率达到100%，或者是0%，这样就可以立即释放它(停止为它付费)。通过积极地使用一种我们称为运行时绑定的新方法，Stratus根据预测的完成时间分组和打包任务。如果做得好，这些打包的任务将充分利用一个实例，几乎在同一时间完成，并允许释放当时处于空闲状态的实例，同时将利用率降到最低。为了避免由于错误地预测运行时间而延长对低利用率实例的保留，Stratus迁移了仍然在运行的任务来清除这些实例。

Stratus的扩展决策还旨在利用实例类型的多样性和实例价格的变化（静态和动态）。当虚拟集群中需要额外的实例来立即运行提交的任务时，Stratus会请求实例类型，这些实例类型能够有效地匹配一组预测完成时间类似的任务。我们发现，要实现良好的成本节约，需要考虑将待处理的任务打包，同时考虑任务所适用的实例的cost-per-resource-used；在另一种方法之前，单独考虑其中一种方法会导致更少的<packing, instance-type>组合，从而导致更高的成本。Stratus共同决定要将多少任务打包到实例中，以及使用哪些实例类型。

在AWS spot市场的虚拟集群仿真实验中，利用Google和TwoSigma的集群工作负载跟踪，验证了Stratus的有效性。与先进的task-per-VM不打包方法[47]相比，Stratus减少了25%的总成本(Google)和31%的总成本(TwoSigma)。与两个结合动态虚拟集群扩展和作业打包的最先进VC调度器相比，Stratus降低了17-44%的成本。即使使用静态实例定价，比如AWS的按需实例、Google计算引擎和微软Azure, Stratus也能降低10-29%的成本。

Stratus的优势归因于：

1. 根据运行时间的打包
2. 实例多样性感知
3. 低利用率驱动的迁移。

本文的主要贡献

1. 它标识了一些独特的混合特征，这些特征表示一个专门针对虚拟集群(VCs)的新作业调度程序的角色
2. 它描述了如何使用根据运行时间的打包来最大限度地最小化提高租用实例的低利用率和技术的不足，从而使其在实践中运行良好，包括不完善的运行时间估计
3. 揭示了打包决策与实例类型选择之间的相互依赖关系，展示了共同决策的成本效益
4. 描述了一个批作业调度器Stratus，使用新的打包和实例获取策略，并通过对两个大规模、真实集群工作负载的跟踪驱动仿真，证明了该策略的有效性

2 BACKGROUND AND RELATEDWORK

用户提交的作业，通常还包括每个任务的资源请求(例如，需要多少CPU和内存)。

出于资源隔离和安全目的，每个任务通常以某种形式的容器执行。

Stratus是一个集群调度器，旨在调度批处理工作负载。

Stratus通过利用公有云有效无界的虚拟集群弹性、实例类型多样性和租赁价格变化来降低成本。

IaaS实例类型和契约

CSP提供了不同的VM实例类型，主要由它们硬件资源（例如，核数和内存大小）和租赁契约模型来区分。

on-demand：不可抢占，价格基本固定

transient：更便宜，CSP可以在任何时候单方面撤销，价格经常变动。

AWS EC2中的transient实例称为spot实例，价格由spot市场决定并随时间波动，但通常比相应的on-demand实例价格低70-80%。

要租用spot实例，用户指定一个投标价格bid price,，即他愿意为该实例支付的最高价格。spot实例可以在任何时候被撤销，但在使用常见的竞价策略（例如bidding the on-demand price）时很少发生这种情况。

虚拟集群的自动伸缩

两方面：①确定扩展到的容量②选择适当的实例集来扩展到所述容量。

VC管理框架：根据用户指定的策略，选择和获取实例以扩展到目标容量，例如Amazon EC2 Spot Fleet [9]

Spot Fleet中可用的策略：①lowestPrice，总是添加当前spot价格最低的实例②diversified，添加新实例以增加spot VM池中的多样性。

确定VC的目标容量可以由VC调度器反应性地完成，当新作业的任务不能在现有资源上运行时，就可以进行扩展。web服务需要预测正确的目标容量，以防止SLOs被破坏，在SLOs中可容忍的延迟在几秒或更少[21、27、39、49]，而作业调度器所针对的集群工作负载则更为宽松。即使与总是能够立即启动每个新作业的理想情况相比，我们观察到反应性VC扩展提供了合理的作业延迟(参见第5节)。

将容器化的任务分配给实例

容器服务container services允许在公有云上运行容器化的用户应用程序任务。有两种主要的容器管理服务可用：server-based和container-based。

server-based模型：用户提供实例池(例如通过Spot Fleet)，容器服务根据配置的放置策略将任务调度到可用的VM。例如，Amazon ECS[2]中可用的任务放置策略包括binPack(任务放到最少可用资源的实例上)、random和spread(round-robin)。在VC调度上下文中，server-based容器服务负责将容器化的任务打包到VM实例。

container-based模型：容器服务自动管理容器的放置、执行和所有底层基础设施。根据容器消耗的资源对用户进行计费。对于大规模集群工作负载，这种方法比spot实例VC的server-based模型贵（详见Section 4）。

2.2 Related work

Private cluster schedulers

私有集群通常具有一组固定的机器组成，无论在部署时存在何种硬件异构性。现有先进的调度器[17、19、22、24、29、31、32、41]经常根据现有的实例集优化调度决策。但是，公有云提供了许多类型和大小的实例，允许虚拟集群不仅在大小上而且在组成上随时间变化，因此通常可以在需要时获得给定作业的最佳匹配实例类型。不同的实例类型有不同的租赁价格，必须考虑这一点。更复杂的是，特定实例类型的租赁价格可能随时间变化，尤其是在AWS spot市场。这种差异要求VC调度器关注与传统集群调度器不同的问题。

Task-per-instance virtual cluster schedulers

以前在公有云资源上调度作业的大多数工作都是将每个作业的每个任务映射到一个实例，该实例仅在任务执行期间获得。这种方法既适用于云爆发配置cloud bursting [15、25、54]，在这种配置中，私有集群的多余负载被转移到公有云资源上，也适用于full virtual cluster configurations。

已经探索了task-per-instance调度器的各种策略增强。Mao等人[36,38]提出了框架感知的VC调度技术，该技术平衡了作业deadlines和预算约束。Niu等人讨论了调度启发法，通过对新任务重用实例来解决AWS以前的基于小时的计费模型。HotSpot[47]利用了现货市场的动态特性和实例类型的多样性，总是为新任务分配最便宜的实例，并随着现货市场价格的波动将任务从昂贵的实例迁移到便宜的实例。

Packing VC schedulers.

与现有VC调度文献中为每个实例分配单个任务的常见方法相比，将任务（来自相同或不同的作业）打包到实例的调度程序可能会降低总体成本，因为它们降低了由于不完美匹配而导致的低利用率风险。

一种合理的方法[10]是使用CSP提供的服务在弹性VC上打包容器化的任务。具体来说，可以使用server-based容器服务(例如ECS)将容器化的任务放在(spot)实例上，同时使用实例管理框架(例如Spot Fleet)维护正在运行的实例池。这种组合实质上导致了一个packing VC调度器，这是我们在Section 5中比较Stratus的方法之一。

SuperCloud是一个支持跨不同云迁移应用程序的系统[48]，它包括一个用于获取和打包spot实例的子系统（SuperCloud-Spot）[30]。SuperCloud-Spot似乎主要是为一组固定的长时间运行的作业（例如服务）设计的，因为没有讨论用于处理动态任务到达/完成和各种任务CPU/内存需求的在线打包方法。但是，它代表了有效VC调度的一个重要步骤，我们将它包含在我们的评估中。我们也评估它的自然扩展，作为理解Stratus特性的增量效益的一部分。

Energy-conscious scheduling.

energy-conscious调度器试图通过主动地使一些机器处于空闲状态并关闭它们来降低集群的能源消耗。为了做到这一点，他们试图将任务尽可能紧密地打包到机器上，以最小化必须保持的数量[13、14、35]。这个目标与VC调度器的目标类似，后者的主要目标是通过使用更少的实例时间和更有效地打包实例来最小化集群的开销。获取和释放云中的VM实例类似于打开和关闭物理机器。

尽管energy-conscious调度器和VC调度器都有一个最大限度地利用active machines的目标，但energy-conscious调度器通常不会解决VC调度的实例异构性或价格变化方面所带来的机会。然而，最接近Stratus的方案是Knauth等人提出的调度器，它将VM打包到基于预先确定的运行时间（租用时间）的物理机器上。与调度器不同的是，Stratus不知道运行时间，但它利用运行时间预测来打包预计在同一时间完成的任务。

6 CONCLUSION

Stratus集群调度程序利用云属性和运行时间估计来降低在公有云上执行的集群作业的成本。通过打包应该在同一时间完成的作业，同时考虑可能的打包和可用的实例类型/价格，以及明智地使用任务迁移来清除未充分利用的实例，Stratus积极地避免租用利用率不高的机器。我们期望Stratus的方法成为未来公有云虚拟集群管理的核心元素。