大规模异构云数据中心排队等待成本感知控制模型研究报告

摘要

本文提出一种数据中心的队列等待成本感知控制模型，考虑维护成本与系统性能间的权衡，将数据中心服务器配置为可变服务率的 M/M/1 队列系统，制定并解决最优负载分配和服务率控制问题，通过 DFC（动态反馈控制器）和 SRC（服务率控制器）实现动态反馈控制，经数值模拟验证模型在多服务器动态配置和任务分配中的有效性，在数据中心实现负载均衡，并且我们将其 DFC 和 SRC 控制器应用到分布式算力设备的 LLM 并行计算技术中。

关键词：DFC；SRC；负载均衡

1 引言

云计算作为一种重要的计算范式，通过基础设施即服务（IaaS）、平台即服务（PaaS）和软件即服务（SaaS）为用户提供服务。云数据中心由大量异构服务器组成，随着用户对大数据处理服务需求的日益复杂多样，高效分配计算资源成为关键问题。然而，现有的许多负载分配和平衡算法仅使用服务器的静态信息进行任务调度，未考虑队列等待时间对系统成本和性能的影响、服务器服务率的动态变化以及资源池消耗成本受异构架构的影响。通过对实际工作负载跟踪分析发现，任务的响应时间与执行时间之比平均较高，且内存资源请求比 CPU 资源更密集，因此需要一种新的控制模型来优化资源分配。

研究贡献 1. 将队列等待成本视为任务最优调度的目标成本函数，提出基于将每个执行服务器视为具有可变服务率的 $M/M/1/\infty$ 排队系统的高效新颖负载分配策略。2. 利用速度缩放技术为每个执行服务器配置顺序服务率以实现动态配置，并通过服务率控制算法考虑性能与缩放消耗之间的权衡，该算法使服务率与平均排队长度成比例，并使服务器动态向调度器返回服务率信息。3. 进行了广泛的数值模拟以评估和验证模型，结果表明所提出的负载分配优化策略可优化任务调度，在每个执行服务器中实现最小平均队列等待成本 [1]。

2 相关研究

负载分配和多服务器控制是云计算环境中的关键研究问题。许多研究将分布式计算机系统建模为排队模型来解决负载平衡问题，并提出了静态和动态优化策略。在速度缩放策略方面，也有大量研究探讨了其在优化能源成本和系统性能方面的应用。此外，排队论被广泛用

于建模和分析大规模异构服务器数据中心架构，但大多数现有研究未考虑由虚拟机实例增加、内存大小、数据存储、网络链接等引起的持有成本（与队列长度成正比）。

3 动态控制模型

（一）动态反馈控制器（DFC）设大规模云计算数据中心有一组异构服务器，每个服务器具有不同的服务率和成本系数。在 DFC 中，将每个服务器视为 $M/M/1$ 排队系统，以等待成本函数为目标函数。通过计算任务到达率、当前平均服务率等参数，可得到每个服务器的平均等待时间和等待成本，DFC 的目标是将任务流分配到各个服务器，使数据中心的平均等待时间成本最小化。

任务到达率为 λ，分配到服务器 S_i 的任务到达率为 λ_i。根据服务器 S_i 的当前平均服务率 $\overline{\mu_c^i}$，可以计算出其平均等待时间 $W_q^i = \frac{\lambda_i}{\mu_c^i(\mu_c^i - \lambda_i)}$ $(\lambda_i < \overline{\mu_c^i})$。进而得出每个计算节点的等待成本为 $\frac{\lambda_i}{\lambda}(W_q^i + C_i\overline{\mu_c^i})$，整个数据中心的平均等待时间成本 $\overline{C_w} = \sum_{i=1}^{n} \frac{\lambda_i}{\lambda}(W_q^i + C_i\overline{\mu_c^i})$。DFC 与调度器协同工作，其目标是通过合理分配任务流，使 C_w 最小化。

（二）服务率控制器（SRC）利用速度缩放技术，每个服务器被视为具有可变服务率的 $M/M/1/\infty$ 排队系统。SRC 根据服务器的平均队列长度调整服务率，通过控制变量 k_i 来控制调整过程，并考虑服务器的功率消耗、调整消耗和持有消耗三种成本。其目标是选择最优的 k_i 值，使服务器的平均总成本最小化。

服务器 S_i 具有 M_i 个可变服务率 μ_j^i。根据服务器 S_i 的平均队列长度 L_q^i 来调整服务率，通过控制变量 $k_i(k_i \in N_+)$ 实现。在调整过程中，考虑服务器 S_i 的三种成本：与平均服务率成正比的功耗、与单位时间平均调整次数成正比的调整消耗以及与平均队列长度成正比的持有消耗。设这三种成本系数分别为 c_p^i、c_a^i 和 c_h^i，则服务器 S_i 的平均总成本 $F_i(k_i) = c_p^i F_\mu^i(k_i) + c_a^i F_t^i(k_i) + c_h^i F_l^i(k_i)$，SRC 的目的是选择最优的 k_i 值使 $F_i(k_i)$ 最小化。

（三）动态控制模型该模型包括负载分配、服务器最优控制和动态反馈三个步骤。DFC 根据从服务器监控器获取的参数控制调度器分配任务，SRC 计算服务器的平均服务率并调整服务率，同时向服务器监控器提供反馈信息，服务器监控器更新信息并发送给 DFC，形成动态循环控制。该模型在包含 n 个异构服务器的数据中心中，通过以下三个步骤实现对多服务器的优化控制：

1. 负载分配：DFC 从服务器监控动态获取 $\overline{\mu_c^i}$、C_i 和 λ 等参数后，依据负载分配和平衡算法控制调度器将任务 λ_i 分配到服务器 S_i，以最小化成本度量 C_w。

2. 服务器最优控制：服务器 S_i 的 SRC 首先通过最小化 $F_i(k_i)$ 选择 k_i 的最优值，然后根据任务到达率 λ_i 计算服务器 S_i 的平均服务率 $\overline{\mu_c^i}$，并利用该值控制服务器的服务率，以优化性能指标（平均队列等待时间、平均队列长度和平均响应时间）。最后，SRC 向服务器监控动态提供反馈信息，用于基于队列等待成本感知的最优负载分配。

3. 动态反馈：服务器监控更新参数信息 $(\overline{\mu_{c'}^i}, C_i, n, \lambda_i)$ 并动态发送给 DFC。

4 多服务器最优控制模型

（一）负载分配策略

在云计算资源分配和管理中，优化队列等待时间可提高系统吞吐量。将最优负载分配问

题建模为排队等待成本最小化问题，通过求解该问题得到最优调度策略 λ，并证明该优化问题是凸优化问题，可使用二分搜索算法求解。

在云计算资源分配和管理中，优化队列等待时间有助于减少资源消耗和提高系统吞吐量。由于每个服务器的成本系数取决于先验信息，且队列等待时间和任务持有成本之间存在权衡，因此最优负载分配问题被建模为排队等待成本最小化问题。其优化目标为 $\underset{\{\lambda,\overline{\mu_c^i},C_i\}}{min}\ (F(\lambda) = \overline{C_w} = \sum_{i=1}^{n}(\frac{\lambda_i^2}{\overline{\mu_c^i}(\overline{\mu_c^i}-\lambda_i)\lambda} + \frac{\lambda_i}{\lambda}C_i\overline{\mu_c^i}))$，同时满足约束条件 $\sum_{i=1}^{n}\lambda_i = \lambda$，$\lambda_i \geq 0$，$\overline{\mu_c^i} - \lambda_i > 0$。该问题是凸优化问题，可通过二分搜索算法求解，相应算法为 Algorithm Calculatelds$(C_i[], \overline{\mu_c^i}[], \lambda, n)$。

（二）最优控制动态配置

计算 $F_\mu^i(k_i)$：设服务器 S_i 在服务率 μ_j^i 下的利用率为 $\rho_{ij} = \frac{\lambda_i}{\mu_j^i} < 1$，$p(i,j,l)$ 表示在该服务率下队列系统处于状态 l（任务队列长度为 l）的稳态时间百分比。通过分析可知 $M/M/1/\infty$ 队列可变服务率问题是一个生灭过程，由此可得

$F_\mu^i(k_i) = \left(\mu_1^i + \sum_{j=1}^{M_i-1}(\mu_j^i\frac{\rho_{ij}(1-\rho_{ij}^{k_i})}{1-\rho_{ij}}(\prod_{t=1}^{j-1}\rho_{it}^{k_i})) + \mu_{M_i}^i\frac{\rho_{iM_i}}{1-\rho_{iM_i}}(\prod_{t=1}^{M_i-1}\rho_{it}^{k_i})\right)p(i,0,0)$。

计算 $F_t^i(k_i)$：根据状态转移概率图，当稳态从 $p(i,j,jk_i)$ 变为 $p(i,j+1,jk_{i+1})$ 时，服务率从 μ_j^i 变为 μ_{j+1}^i，其概率等于稳态 $p(i,j+1,jk_{i+1})$。因此，单位时间内服务率的变化次数为 $\mu_{j+1}^i p(i,j+1,jk_i+1)$，进而可得服务器单位时间内平均调整次数为 $F_t^i(k_i) = 2\sum_{j=0}^{M_i-1}\mu_{j+1}^i p(i,j+1,jk_i+1) = 2\sum_{j=1}^{M_i}(\lambda_i(\prod_{t=1}^{j-1}\rho_{it}^{k_i}))p(i,0,0)$。

计算 $F_l^i(k_i)$：由状态转移概率图可知每个稳态表示队列长度及此时的概率，根据相关公式可得 $F_l^i(k_i) = \left(\sum_{j=1}^{M_i}\sum_{l=(j-1)k_i+1}^{jk_i} l\rho_{ij}^{l-(j-1)k_i}(\prod_{t=1}^{j-1}\rho_{it}^{k_i}) + \frac{\rho_{iM_i}(1+M_ik_i-M_ik_i\rho_{iM_i})}{(1-\rho_{iM_i})^2}(\prod_{t=1}^{M_i}\rho_{it}^{k_i})\right)p(i,0,0)$。

计算 k_i：为了获得 k_i，需要证明 $F_i(k_i)$ 存在最小值 $F_i(k_i) = C_{fmin}$，且 $k_i' \in (0,N)$。通过分析可知 $F_i(k_i)$ 在 $(0,N)$ 是凹函数且存在最小值，可利用线搜索算法（Algorithm 1 Calculate $k_i(c_i^p, c_i^a, c_i^h, m_i, M_i)$）在小范围内计算最优 k_i。

（三）基于动态控制策略的动态反馈

根据服务器的状态信息，DFC 和 SRC 调整服务器配置。在不同条件下，负载分布不平衡或服务器利用率超出阈值时，DFC 会触发 SRC 调整服务率或选择合适的服务率序列，同时设置了开关阈值和利用率阈值来控制调整过程，避免系统振荡。

负载分配调整：当 $\sum_{i=1}^{n}\mu_c^i \leq \lambda$ 且 $\lambda < \sum_{i=1}^{n}\mu_{M_i}^i$ 不满足时，若 Algorithm Calculate lds$(C_i, \overline{\mu_c^i})$ 无法优化负载分配，DFC 应启动 SRC 提高最小服务率；当 $\lambda_i\sqrt{\mu_c^i} < \Theta_l^i$ 或 $\lambda_i\sqrt{\mu_c^i} > \Theta_h^i$ 时，DFC 会向相关 SRC 发送信息以调整服务器配置，SRC 将更新服务配置率并在下一个时间片将 $F_\mu^i(k_i)$ 的值返回给 DFC 用于负载分配，且 SRC 应根据任务到达率的斜率反向调整最小服务率。同时，设置开关阈值 $\xi = \lambda/\sum_{i=1}^{n}\mu_{M_i}^i$（通常设为 0.7）来控制是否使用相关算法调整服务率范围，以避免系统振荡。

M_i 选择影响：M_i 的选择会影响系统成本、敏感度和振荡幅度。当 DFC 启动 SRC 提高最小服务率时，M_i 较大的设置会使系统消耗增加更快；在响应任务到达率变化时，M_i 较大的服务器对任务到达率的微小变化更敏感，且振荡幅度更小。云提供商可根据服务器状态调整 M_i，以实现合理成本、系统稳定性和能量消耗，并保证 DVFS 调整策略的可实施性。

5 数值验证

（一）实验设置

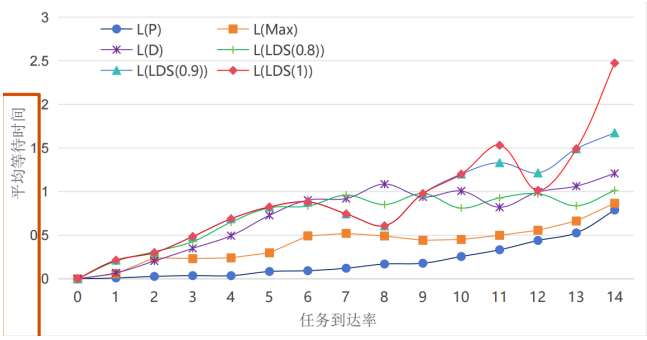

图 1. 平均等待时间方案对比图

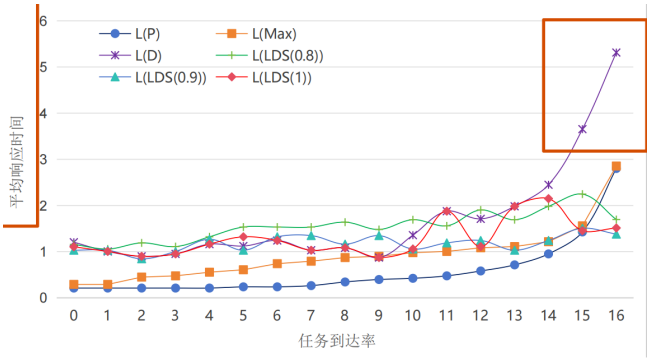

图 2. 平均响应时间方案对比图

实验中的数据中心有 3 代服务器，设置了不同的任务到达率和利用率阈值，并使用 MultiRECloudSim 和 Matlab 进行模拟实验。考虑了多种负载分布策略：比例负载分布 L(P)、最大服务率负载分布 L(MAX) 和基于本文模型 L(LDS) 的不同策略，并记录了任务的响应时间和等待时间等指标。

（二）性能模拟结果

1. 通过模拟实验和分析结果对比，发现本文模型在不同任务到达率下，平均队列等待时间（TW_q）和平均响应时间（TW_s）的分析结果与 CloudSim 模拟结果一致，验证了模型指标的可靠性。方案对比结果如图 1，图 2，图 3 所示。

2. 在不考虑运营和维护成本（$C_i = 0$）的情况下，比较 $L(P)$ 和 $L(Max)$ 负载分布策略，结果表明 $L(Max)$ 在平均队列等待时间和平均响应时间方面优于 $L(P)$，改进幅度在 2.15

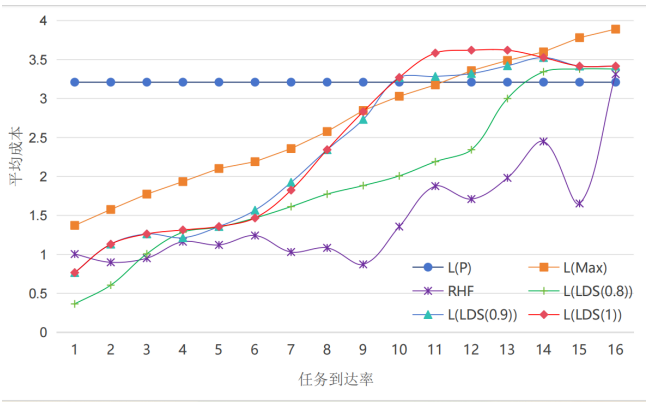

图 3. 平均成本方案对比图

3. 考虑运营和维护成本及平均队列等待时间时，比较不同策略的性能指标，发现本文模型中基于可变服务率的策略（$L(LD)$、$L(LDS_{(1)})$ 和 $L(LDS_{(0.7)})$）在平均队列等待时间和平均响应时间上略大于 $L(Max)$，但通过设置合适的开关阈值（$\xi = 0.7$）可避免振荡。同时，这些策略在成本方面也有所不同，$L(LDS_{(1)})$ 的运营和维护成本增长较快。

（三）M_i 的影响

通过改变 M_i 和 $\mu^i[]$ 的参数设置进行模拟，结果表明不同的 M_i 选择会影响系统对任务到达率的敏感度、振荡幅度和即时成本。较大的 M_i 使系统对任务到达率变化更敏感，但也可能导致振荡，同时即时成本较低，但调整策略实现难度增加。云提供商可根据服务器状态合理选择 M_i。

6 结论

本文提出了一种用于异构数据中心云计算的动态反馈和队列等待成本感知的最优负载分配和多服务器控制模型。通过将服务器配置为具有可变服务率的排队系统，并利用凸优化和排队论理论，研究了最优负载和成本分布。通过数值和模拟实验验证了模型的有效性，并表明在不同情况下应选择不同的负载分布策略。未来研究可进一步优化模型参数设置和算法效率，以适应更复杂的云计算环境。

在算力共享模型中，我们的任务等待成本感知控制模型能够动态调整资源分配和服务速率，以适应不断变化的任务需求和系统状态。通过实时监测任务队列长度和服务速率，模型能够优化负载分配，降低任务等待时间，并提高系统整体性能。

此外，该模型还能够考虑成本因素，包括能源成本、频率扩展成本和排队延迟成本等。在算力共享环境中，这些成本因素对于云服务提供商和用户都具有重要意义。因此，我们的模型能够在保证系统性能的同时，实现成本的最小化。

参考文献

[1] Shaowei Huang Weihua Bai, Jiaxian Zhu and Huibing Zhang. A queue waiting cost-aware control model for large scale heterogeneous cloud datacenter. *IEEE Transactions on Cloud Computing*, PP(99):1–17, 2020.