

HPC 和云计算中工作流的调度机制文献综述



摘要: 传统的工作流系统可以让用户建立可视化的应用程序, 而且不需要复杂和耗时的编程。但是工作流系统的缺点是不能对软件组件进行方便的访问, 这样使得工作流系统不能得到广泛的应用, 刚好云计算为其提供了一个访问资源池和硬件的服务, 使得工作流系统得到了很好的发展。其中云工作流任务调度算法是云工作流系统的核心技术之一, 其调度策略的好坏将直接影响到云工作流系统性能的表现。本文总结了一些云工作流调度的方法。

关键词: 云工作流, 调度, 云计算, 高性能计算, HPC

1 云工作流概述

1.1 云工作流

工作流是依据一系列已经定义好的规则自动执行的业务过程, 在不同的执行者间传递文档和任务, 并由多位参与者分别协同完成。在工作流执行的过程中, 工作流管理系统通过计算机技术对工作流进行执行和管理, 并协调好各成员或工作之间的信息交互。[19]

随着云计算的不断发展, 分布式工作流系统也逐渐向云计算的工流系统方向演变。这是工作流管理系统在云计算环境下产生的一种全新的应用模式, 简称为云工作流。云工作流管理系统对云计算数据中心的计算或存储资源统一管理, 实现云工作流任务调度有序高效执行, 从而实现业务流程自动化。工作流中的任务根据其计算依赖性链接, 这些任务被表示为有向无环图 (DAG), 云工作流具有以下特征:

- (1) 透明性。在云计算环境下, 所有的资源都是虚拟化的, 在云中的所有服务所需的运行环境, 操作系统, 实现语言既可以是相同的, 也可以是不同的。用户无须知道所用服务的内部实现结构, 更不需要知道所用资源具体在哪个位置。
- (2) 可伸缩性。云资源具有按需分配的特点, 用户可以随时随地的通过网络购买所需的资源。用户也可对多余的资源进行释放, 减少租用资源的费用。这种动态的资源管理方式

使得工作流任务得以高效执行。此外, 这种可伸缩性不但实现用户对计算资源的自适应, 也使得云服务商最大程度利用资源, 实现自身利益最大化。

- (3) 实时监控。在云计算环境下, 监控管理模块通过监测云工作流任务的运行情况, 可以实现资源的负载均衡、故障监控以及节点规模控制。总而言之, 云工作流是针对云计算系统需要配置灵活、任务的自动调度、资源的优化管理及计算流程性等特点提出的一种优化解决方案。通过云工作流不仅能够压缩云计算成本, 而且能够提高云服务质量。因此云工作流必定会得到越来越多的关注, 也会快速的发展起来, 从而推动整个云计算产业的发展。

1.2 工作流的执行

云环境下工作流的执行包含两个主要的阶段:

- (1) 资源提供 (2) 资源映射到任务上。为工作流提供资源有两种方式: (1) 静态调度 (2) 动态调度。

在静态调度环境 (集群和网格) 中, 资源的配置是事先已知的。因此, 根据用户需求来配置资源是困难的。云环境的动态可扩展性为克服这种静态调度提供了有效的解决方案。

在云环境中, 资源是动态配置的。由于在单个 CSP, 云服务代理, 私有云和公共云之间映射适当的资源, 这被认为是具有挑战性的研究问题。这个有趣的问题为近年来各种资源调度算法的出

现铺平了道路。

1.3 workflow 建模

workflow 应用被表示为有向无环图 (DAG)，表示为 $G(T,E)$ ，其中 G 是图， T 表示任务集， E 表示任务之间依赖关系的有向边集合。在一个应用程序中，每个任务代表以百万条指令 (MI) 衡量的单个任务。没有优先级任务的任务称为入口任务，没有后代任务的的任务称为出口任务。 workflow 大小由任务总数决定。图 1 显示了十个任务的示例 workflow。

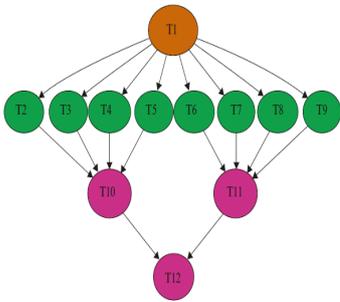


图 1: workflow 模型

workflow 的结构描述 workflow 中任务之间的依赖关系。该结构可以分类为序列，并行和在基于 DAG 的结构中的选择。顺序结构 workflow 将以串行方式执行；并行结构同时执行 workflow 任务。在选择结构中， workflow 以串行和并行方式执行。在图 1 中，任务 T1，T4 和 T10 代表序列 workflow 结构，而 T5 和 T6 代表并行 workflow 结构。

1.4 云 workflow 任务调度目标

云 workflow 任务调度就是通过云 workflow 系统中的任务调度器实现对用户提交的 workflow 任务实施综合管理，而云环境下的用户群体各异，不同的用户对云 workflow 任务调度的目标优化也各不相同，这里可以将云 workflow 任务调度的目标优化问题分为两大类：单目标优化问题和多目标优化问题，如图 2 所示。

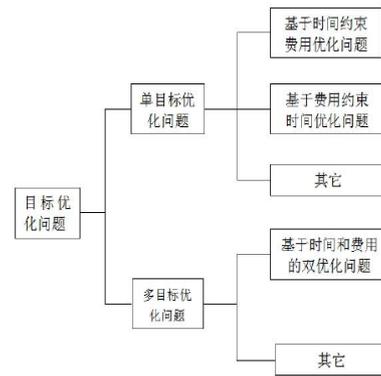


图 2: 目标优化问题

其中相对用户而言，对于云 workflow 任务调度的目标优化问题，大多数关注以下几个 QOS 指标：

执行时间：它表示云 workflow 任务完成需要的时间，是调度策略中最重要的目标。对用户来说，如果所提交的云 workflow 任务对完成时间有着迫切的需求，只需将其映射到性能好的计算资源上，这样可以使得云 workflow 任务的执行时间大大的减少。

执行费用：即一个云 workflow 任务执行完成后，对服务资源的使用而产生的费用，其中包括带宽传输费用和计算费用等。云 workflow 任务的执行费用与它所使用资源的性能有关，也就是说如果用户希望提交的工作流任务得到快速的完成，则可将任务分配到处理能力强的虚拟资源上执行，与此同时，用户则需要为其支付更多的费用。

可靠性：在某些领域诸如航天航空对 workflow 任务执行的可靠性要求非常高，因为一个数据的不确定可能会带来意想不到的灾难。但是在云计算环境下如何的去保持数据的可靠性是有一定的难度的，因为任务在处理过程中可能出现一些异常，又或者由于底层软硬件出现了问题而导致任务不能执行。

安全性：云计算是跨领域的，用户所使用的虚拟资源可能分布在各个不同的地区，当用户提交的云 workflow 任务在不同地区的虚拟机资源上执行时，具有依赖关系的任务则会在网络上进行数据的传输，此时，可能一些网络黑客会窃取用户的数据，造成用户一些重要数据的泄露。因此云 workflow 任务在执行过程中需要确保数据的安全，满足云计算环境下用户对安全性的需求。

1.5 workflow 调度分类

为了区分众多的 workflow 调度方法, Fuhai 等将其分为三类: 静态调度, S-plan-D-Sched, 以及基于 workflow, 资源, 任务分配等资源的动态调度 [23]。分类如图 3 workflow 调度分类所示。

比较研究表明, 静态调度算法在几种情况下优于动态调度算法。原因在于静态调度在 workflow 级别和任务级别全局搜索解决方案空间。它假定任务执行, 通信时间可以提前获得, 这在实际系统中并不总是真实的。资源和工作流信息都决定了任务执行和通信时间。工作流信息包括工作流的结构, 任务执行工作量和通信数据大小。资源信息包括可用性, 处理能力, 通信带宽等。多租户资源共享, 硬件和软件故障的性能波动由于其动态资源提供特性而在云计算环境中也被考虑到。动态调度能够处理这些不确定性。综合的方法被称为 S-Plan-D-Sched, 它利用了静态和动态调度。它基于任务执行和通信时间的估计近似地为所有任务做出静态的计划。在运行时, 任务分配是自适应调整的, 如果需要的话, 任务被重新安排。

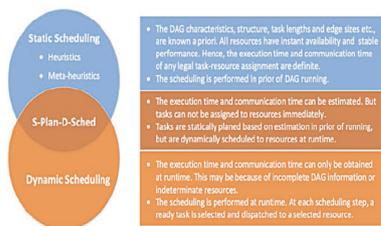


图 3: workflow 调度分类

2 静态 workflow 调度

静态调度会精确地假定任务计时信息, 但会导致运行时间较少。这是包含基于元启发式和启发式的方法。一个静态调度算法的例子是机会负载均衡 (OLB)。

2.1 列表调度启发式

它通过分配优先级和根据优先级对任务进行排序来创建一个调度列表, 然后重复选择任务和资源, 直到 DAG 中的所有任务都被调度。需要优先化属性和资源选择策略来确定每个任务的优先级别和最优资源。

一些列表调度启发式是修改关键路径 (MCP), 映射启发式 (MH), 插入调度启发式和最早时间优先 (ETF), 异构最早完成时间

(HEFT), 处理器关键路径 (CPOP), 动态关键路径 (DCP) 以及预测最早结束时间 (PEFT)。13 年, [42] 这篇文章针对云环境采用 PCP 算法 (作者们之前设计和分析了一个实用的用于公共事业网格的两阶段调度算法, 称为部分关键路径, 旨在最大限度地降低 workflow 执行的成本, 同时满足用户定义的期限), 提出了两种 workflow 调度算法: 一种是 IaaS 云局部关键路径 (IC-PCP) 算法, 另一种是两阶段算法, 称为 IaaS 云具有截止时间分布的部分关键路径 (IC-PCPD2)。两种算法都具有多项式时间复杂度, 使其成为调度大型工作流的合适选择。仿真结果表明, 这两种算法都具有良好的性能, 在大多数情况下, IC-PCP 的性能优于 IC-PCPD2。

早期提出的这些算法主要关注 workflow 任务的执行时间。其中 HEFT 算法 [1] 是最被广泛使用的云 workflow 启发式算法。Cui Lin 在 HEFT 算法的基础上进行了改进, 提出了 SHEFT 调度算法 [2], 该算法能够在运行时对资源弹性地扩展, 并通过实验表明, SHEFT 算法在优化 workflow 的执行时间方面明显优于其它几种代表性的 workflow 任务调度算法。Dudin 等人 [3] 提出两个多项式时间复杂度的算法, 并与 IC-Loss 算法相对比, 该算法更能有效地减少任务完成时间。Delavar 等人 [4] 对传统遗传算法进化速度慢的缺点进行改进, 提出一种 HSGA 算法, 并通过实验表明, 该算法能获得任务完成时间的最优解。

2.2 聚类启发式

聚类启发式旨在优化数据相关任务之间的传输时间。聚类启发式算法的两个主要部分是聚类和排序。聚类将任务映射到集群, 而在同一集群中排序任务。

2.3 复制启发式

复制技术通常与列表调度或聚类调度一起用作优化过程或作为新算法。设计有效的任务重复算法时有两个问题需要解决:

1. 要复制哪个任务? 通过选择哪个父任务被复制, 可以最小化子任务的开始时间。
2. 在哪里复制任务? 在资源上分配合适的时间段以复制父任务。

根据重复任务的选择, 重复算法分为全重复调度 (SFD) 和部分重复调度 (SPD)。SFD 中的高优先级或高级别任务被认为是重复的。

2.4 面向网格计算的启发式算法

在网格环境下所研究出的 workflow 调度算法许多已经被应用于现实中的各种项目工程中了。文献 [33] 根据调度时对任务的侧重点的不同将 workflow 调度算法分为了两种启发式 workflow 调度算法, 即 workflow 级和任务级。workflow 级的启发式算法是根据整个 workflow 的所任务包含的信息来进行调度的。而任务级的启发式算法则是根据单个任务或局部几个独立任务的信息来进行调度的。在针对网格的 workflow 调度算法的研究中最小化 - 最小化算法、Sufferance 算法和最大化 - 最小化位法目前应用较多的针对任务方面的启发式 workflow 调度算法, 文献 [34] 就将这三种算法应用于生物成像中。文献 [35] 改进了任务级调度算法, 将影响 workflow 调度的关键任务重新调度到执行成本更低的资源上从而降低了整个 workflow 的执行成本。

文献 [36] 改进了 GRASP 算法, 设计出了一种任务级的调度算法, 并且在数据密集和实例密集两种特殊的任务情况下将其与 Min-Min 调度算法进行分析和对比。Blythe 等人 [28] 研究贪婪 GRAP (贪婪随机自适应搜索) 算法为网格上的 workflow 调度, 执行比 Min-Min 启发式更好的数据密集型应用程序。Young 等人 [29] 研究了在网格环境中调度 workflow 应用的模拟退火 (SA) 算法的性能。

上述算法虽然都在一定程度上能对 workflow 的调度有所优化, 但是它们都仅仅关注与如何减少 workflow 整体的执行时间, 而没有考虑到 workflow 执行的成本问题。在网格计算环境下, 其向用户所提供的服务大多是免费的, 所以在网格环境下的 workflow 调度算法可以不必考虑执行成本的问题, 而仅仅只需要关心任务的执行时间是否满足用户的需求即可。

2.5 元启发式

为了获得更好的优化解决方案, 使用元启发式方法。使用这种方法解决了大量复杂的问题。

DAG 调度是一个 NP 完全问题。因此, 与精确的方法相比, 开发一个近似算法是一个很好的选择。一些元启发式提出了解决方案, 因为它们提供了一个有效的解决方案。遗传算法是任务调度问题的最佳解决方案之一。一些 DAG 调度的例子包括 [24, 25, 26, 27]。这些遗传算法在搜索空间中的时间表的字符串表示, 用于时间表的适应度函数评估, 用于生成新时间表的遗传算子以及

用于控制遗传算子的随机指派方面不同。

很多算法只考虑到如何尽可能地减少 workflow 任务的执行时间, 比如说上文所介绍到的针对网格计算的 workflow 调度算法, 没有考虑到云 workflow 任务执行时所花的费用。然而云计算有着以市场为导向的经营模式, 用户需要为他们使用的资源付费, 这促使了任务成本花费也成了云 workflow 任务调度的研究重点。最初执行费用优化算法只考虑最小化任务费用, M.A.Rodriguez 等 [5] 针对科学 workflow 调度问题, 以整个云 workflow 任务的执行费用最小为目标, 提出了基于 PSO 的优化调度方法, 并通过实验证明了该方法的优越性。王宗江等人 [6] 提出了一种在混合云环境中最小化总费用的任务调度方案。还有文献 [7][8] 都提出一种费用优化的调度算法, 都只考虑最小化任务费用。Pandey 等人 [9] 提出了一种基于启发式的粒子群优化算法用来最大限度地减少调度 workflow 到云计算资源的执行费用。

3 动态调度

动态调度在运行期间不知道任务到达信息, 尽管它能更好地适应执行期间的变化。动态调度的一些例子是早期最早优先 (EDF) 和最低松弛优先 (LLF)。开发动态调度是为了处理与其他 workflow 或非 workflow 系统负载不可用的调度信息和资源争用。Sonmez 等人 [30] 提出了基于三资源信息状态, 处理速度和链路速度以及两任务信息任务长度和通信数据量的动态调度分类。动态调度算法在可用资源队列之间平衡负载。Xie 和 Qin [31, 32] 讨论了一组用于同构群集和异构分布式系统的动态安全感知调度算法。

4 其他算法

下面介绍的算法从优化本身角度出发, 从约束条件和优化目标角度来归类这些算法。并单独介绍几个使用机器学习方法的算法, 这几篇也是在最近机器学习深度学习人工智能比较红的时候发出来的, 比较会蹭热点。

4.1 在某些约束条件下实现某目标的最优化

但是在很多启发式的算法的研究单单考虑费用或时间一个因素, 一般情况下, 用户要求云 workflow 任务在某些约束条件下实现某目标的最优化, 因此许多学者对此又进行了一些研究。如有些学

者提出了有时间约束下费用最小化的工作流任务调度算法 [10]。Tordsson 等 [11] 利用二进制对不同虚拟资源做出划分, 然后提出一种新的粒子群优化算法用于任务分配, 在满足用户时间约束情况下, 使得费用能够最小。Liu 等 [12] 人提出了一种权衡时间和执行费用的调度算法, 该算法在满足用户费用约束的条件下减少了总执行时间。Abrishami 等人 [13] 将部分关键路径概念引入所提的 QOS 工作流任务调度算法, 该算法以截止时间为约束, 使得费用最小。然后将他们将部分关键路径方法应用到 IaaS 云平台, 以截止时间为约束, 形成 ICPCP 调度算法, 通过确定每个任务的最迟完成时间来督促整个工作流在截止时间内完成任务 [14]。Ming Mao 等人 [15] 提出了一种自动伸缩的工作流任务调度策略, 该算法能够动态的对云计算资源进行调度, 使得任务在截止时间内完成, 并且最小化服务费用。Suraj [16] 提出了一个基于粒子群 (PSO) 的算法去优化云工作流任务, 考虑截止时间约束, 最小化执行费用。

4.2 双目标或多目标优化的方法

目前有很多的云工作流任务调度算法主要研究任务完成时间或执行费用的单目标优化问题。因此一些研究者又对多目标优化问题进行了研究, 如 F. Zhang 等人 [17] 考虑了时间和费用的双目标优化。徐立等 [18] 提出一种基于克隆选择算法的任务完成时间与执行费用多目标优化算法, 并通过实验证明该算法是有效的。

大多数现有方法通常将所有目标集中在一个单一的功能中, 先验定义不需要任何有关解决问题的知识, 这对解决方案的质量产生了负面影响。相比之下, 基于 Pareto 的方法有一组 (几乎) 最佳的解决方案, 代表不同的目标之间的权衡, 已经很少研究。[21] 分析了基于 Pareto 的列表调度启发式 MOHEFT, 它为用户提供了一组折衷的最优解决方案, 从中可以手动选择更适合用户需求的解决方案。[21] 展示了 MOHEFT 在商业 Amazon EC2 Cloud 上进行多目标工作流调度的潜力。[21] 使用不同的合成和现实世界的工作流程比较了 MOHEFT 折衷解决方案的和两种最先进的方法 (用于单目标调度的经典 HEFT 算法和用于多目标优化问题的 SPEA2 * 遗传算法) 的质量。结果表明 Durillo J J 等人方法能够计算比 SPEA2 * 更高质量的解决方案。另外, 由于基于遗传的方法无法处理由这些系统施加的一些约束, 所以 Durillo J J 等人证明 MOHEFT 比 SPEA2 *

更适合于商业云环境中的工作流调度。

[21] 这篇 paper 的贡献总结如下: (1) 为异构的分布式计算系统设计了一个新的基于 Pareto 的多目标工作流调度算法 (2) 扩展 MOHEFT 以处理商业云计算系统, 提供有限的实例和实例类型的灵活组合; (3) 在 Amazon EC2 的环境下, 将 MOHEFT 的结果与适应商业云新情况的两种最先进的算法进行了比较。 (4) 分析了 MOHEFT 针对不同的合成和现实世界的工作流应用程序计算的权衡解决方案。 (5) 分析使用不同数量的资源对计算折中解决方案的影响。

一般来说, 云工作流调度是一个复杂的优化问题, 需要考虑不同的准则, 以满足大量的 QoS 要求。传统的工作流调度研究主要集中在时间或成本约束下的优化, 而不关注能耗。[41] 这个研究的主要贡献在于提出了一种云中多目标工作流调度的新方法, 并提出了混合 PSO 算法来优化调度性能。[41] 的方法是基于动态电压和频率标定 (DVFS) 技术来最小化能耗。这种技术允许处理器通过牺牲时钟频率在不同的电压电平下工作。这种多重电压涉及时间表和能量之间的妥协。对合成和现实世界的科学应用的仿真结果突出显示了这种方法的强大性能。

云计算和移动互联网的不断融合, 促进了移动云计算的产生与发展。在移动云计算环境下, 用户可将工作流的任务迁移到云端执行, 这样不但能够提升移动设备的计算能力, 而且可以减少电池能源消耗。但是不合理的任务迁移会引起大量的数据传输, 这不仅损害工作流的服务质量, 而且会增加移动设备的能耗。基于此, [40] 提出了基于延时传输机制的多目标工作流调度算法 MOWS-DTM。该算法基于遗传算法, 结合工作流的调度过程, 在编码策略中考虑了工作流任务的调度位置和执行排序。由于用户在不断移动的过程中, 移动设备的无线网络信号也在不断变化。当传输一定大小的数据时, 网络信号越强则需要的时间越少, 从而移动设备的能耗也越少。而且工作流结构中存在许多非关键任务, 延长非关键任务的执行时间并不会对工作流的完工时间造成影响。因此, [40] 在工作流调度过程中融入了延时传输机制 DTM, 该机制能够同时有效地优化移动设备的能耗和工作流的完工时间。仿真结果表明, 相比 MOHEFT 算法和 RANDOM 算法, MOWS-DTM 算法在多目标性能上更优。

4.3 结合机器学习和数据挖掘等大数据处理的方法

[20] 对云计算环境下 workflow 任务调度的现有方案进行分析, 针对存在运行时间长、资源利用率低等不足, [20] 提出一种结合改进型布谷鸟搜索算法和决策树的 workflow 任务调度方案。首先, 根据 workflow 任务属性分配截止时间; 其次, 利用改进型布谷鸟搜索算法将 workflow 分割成多个子 workflow, 最小化数据依赖性, 再利用决策树选择出满足任务 QoS 约束的资源; 最后, 根据任务的计算时间、排队时间和通信延迟的总和来判断是否满足截止时间约束, 以此配置相应的资源。实验结果表明, 该方案具有较短的总运行时间和较高的任务完成率。

[20] 的主要创新在于: 1) 采用 CS 搜索算法, 将 workflow 划分成子 workflow, 最小化数据依赖性, 以此提高后续任务调度的效率。2) 对 CS 搜索算法进行改进, 提出了一种适用于 workflow 划分的变异和交叉机制, 避免算法陷入局部最优, 提高划分能力。3) 采用决策树来分类和选择候选资源, 根据任务截止时间约束和排队时间来分配相应的资源。

今天, 几乎每个人都连接到互联网, 并使用不同的云解决方案来存储, 交付和处理数据。云计算组装了大型虚拟化服务网络, 如硬件和软件资源。ICT 渗透到几乎所有领域 (医疗保健, 老年护理, 社会救助, 监视, 教育等) 的新时代都需要新的多媒体内容驱动的应用程序。这些应用程序生成大量的数据, 需要收集, 处理, 然后聚合在由混合云系统 (公共/私人), 移动设备网络和基于桌面的群集等等产生的容错, 可靠和安全的异构分布式系统中。大数据应用程序调度的动态资源调配成为现代系统中的一个挑战。[22] 针对不同类型的应用程序提出了资源感知混合调度算法: 批处理作业和 workflow。所提出的算法考虑在分配阶段将可用资源层次聚类成组。任务执行分两个阶段执行: 第一阶段任务分配给资源组, 第二阶段为每组资源使用经典的调度算法。所提出的算法适用于异构分布式计算, 尤其适用于现代高性能计算 (HPC) 系统, 在这些系统中, 应用程序以各种需求 (包括 IO 和计算密集型) 建模, 并强调多媒体应用程序的数据。我们在实际负载平衡, 成本节约, workflow 依赖性保证和计算效率方面借助 CloudSim 工具评估了性能, 并在运行时调查了这些性能指标的计算方法。

针对异构环境, [22] 的主要工作总结如下:

(1) 首先, 提出了一种考虑任务和资源集群的 HD-C 任务调度混合方法。我们承认没有适合各种任务模型的解决方案。因此, 我们的调度策略是基于使用不同的调度策略, 通过考虑计算资源的异构性和应用任务和/或流程来选择。计算资源利用的有效性取决于资源分配策略的效率。(2) 考虑到 HDC 资源的抽象建模, 提供了一个适应性和动态聚类的模型。在这个模型中计算的“距离”突出了资源之间的相似性, 因此我们可以将资源分组在簇中, 然后在 HySARC2 算法中使用形成的簇。(3) Vasile M A 等人扩展 CloudSim 来考量提出的方法, 并整合了四个调度策略: 两个独立的任务 (最早截止时间优先和最短作业优先) 和两个 DAG 调度 (修改关键路径和最早时间优先)。

针对云计算环境中资源具有规模庞大、异构性、多样性等特点, [39] 提出了一种对资源进行模糊聚类的工作流任务调度算法。经过对网络资源属性进行量化、规范化, 以预先构建的任务模型和资源模型为基础, 结合模糊数学理论划分资源, 使得在任务调度时能够较准确地优先选择综合性能较好的资源类簇, 缩短了任务资源相匹配的时间, 提高了调度性能。通过仿真实验将此算法与 HEFT、DLS 进行比较, 实验结果表明, 当任务在 $[0, 100]$ 范围增加时, 该算法平均 SLR 比 HEFT 小 3.4%, 比 DLS 小 9.9%, 其平均 Speedup 比 HEFT 大 5.9%, 比 DLS 大 10.2%; 当资源在 $[0, 100]$ 范围增加时, 该算法平均 SLR 比 HEFT 小 3.6%, 比 DLS 小 9.7%, 其平均 Speedup 比 HEFT 大 4.5%, 比 DLS 大 10.8%。所提算法实现了对资源的合理划分, 且在执行跨度方面具有优越性。

4.4 感知方法

高效数据感知方法作业调度对于分布式存储管理和数据管理平台中成功执行数据密集型应用程序是必需的。然而, 在大规模分布式云计算环境和集群计算环境中, 对数据密集型科学应用方法的研究还不够充分, 数据感知方法也变得越来越复杂。在论文 [37] 中, 提出了一种数据-位置感知 workflow 调度 (D-LAWS) 技术和一种用于 HPC 云环境中的数据密集型科学 workflow 的局部感知资源管理方法。D-LAWS 将基于网络带宽的数据局部性和数据传输时间应用于科学的工作流任务调度, 并平衡节点级任务的资源利用率和并行性。这种方法整合了虚拟机, 并在数据密集型科学 workflow 的任务执行计划期间考虑数据流的任务并行

性。另外考虑更复杂的工作流模型和与任务执行前数据的放置和传输有关的数据局部性。作者在云环境中实现和验证这种方法。实验结果表明，作者所提出的方法可以提高云环境下数据密集型工作流程的性能和数据局部性。

[37] 的主要工作如下：(1) 提出了一个数据局部感知工作流调度技术，称为 D-LAWS，它包括一个考虑数据局部性的数据感知工作流调度算法和资源整合方法。(2) D-LAWS 不仅考虑了工作流程的关键路径，还考虑了数据的局部性数据传输时间基于网络带宽，这在具有异构资源的分布式环境中至关重要。(3) 所提出的方法在任务执行计划期间也考虑数据流的任务并行性，以便使用 VM 资源整合算法来最大化资源利用率。(4) 在私有云计算环境中评估了提出的方法，并将结果与其他两种工作流调度技术进行比较

工作流调度 (WFS) 主要侧重于任务分配，通过追求可用资源的最佳利用来实现期望的工作负载平衡。同时，为了解决云计算中特定的 WFS 问题，必须考虑相关的性能标准和系统分布结构，通过按用户付费和按需提供不同的服务给用户。在文献中，讨论了影响 WFS 执行成本的各种 WFS 的挑战，但是，以前的工作并没有集体考虑这些挑战。[38] 的主要目标是帮助研究人员从可用的备选方案中选择合适的成本感知型 WFS 方法。为了实现这一目标，作者们进行了广泛的调研，以调查和分析相关方法的基本概念。WFS 在云计算方面的成本意识相关挑战是基于服务质量 (QoS) 性能，系统功能和系统架构进行分类的，这些最终导致分类集合的不同。还讨论了一些研究机会，有助于确定未来在云计算领域的研究方向。这次调研的结果为开发成本意识模型提供了路线图，这将激励研究人员为服务消费者和/或云计算公用事业提供商提出更好的成本感知方法。

5 云中工作流调度的问题

IaaS 和 PaaS 组合云形成了完整的工作流调度架构，并带来新的挑战：

- 针对网格和群集的调度工作专注于满足期限或最小化完工时间不考虑成本。为了避免不必要的成本，需要考虑现收现付模式来开发新的调度算法。
- 适应性：云是一个动态的环境，在工作流程

执行过程中有不同的表现；因此需要自适应调度解决方案。

- 迁移：为进行负载平衡而迁移作业也会影响工作流程调度的性能
- 虚拟机更换：工作流调度算法应该在发生故障的情况下有效地替换虚拟机。工作流程执行前，云环境中有两个主要阶段：资源调配和任务——资源映射。网格和集群环境只关注任务——资源映射阶段，因为它们是静态环境，其配置事先已知。资源配置的一个主要问题是确定工作流应用程序要求的数量和资源类型，这会影响工作流程的成本和完工时间。
- 当前云基础设施提供的虚拟机不具有稳定的性能。这对调度策略有重大影响。VM 启动时间和停止时间是云调度应该考虑的其他重要因素。
- 集成架构：第一个挑战是整合云中的工作流管理系统。工作流引擎应该被设计为具有强大的功能以便能够处理大规模的任务。
- 大规模数据管理和工作流调度：科学的工作流应用程序是数据密集型的，存储和计算资源之间的数据资源管理和数据传输是主要的瓶颈。找到一种有效的方法来管理工作流程所需的数据是非常重要的。
- 服务组合和协调：为了适应进一步增长和复杂的工作流应用程序需求，应该组合几个云提供商的服务，以提供统一的 QoS。为了达到消费者的要求，服务的组成和编排应该以自动和动态的方式进行。在 Intercloud 中寻找高效的组合解决方案是一个重大的挑战，因为它涉及在成本和期限约束下的服务组合和编排优化。

6 总结

云环境为工作流系统提供了前所未有的可扩展性，并改变了我们感知和进行实验的方式。针对不同目标的工作流调度进行了研究，以获得优化的解决方案。本文介绍了云环境中各种现有的工作流调度算法，调度工作流的问题和挑战。算法根据目标进行分类并进行比较。预算是最多数工作中遇到的关键问题，但其他问题（如安全性，能源，可扩展性和可靠性）却没有得到解决；这些

目标应该有一个考虑,以便制定一个有效的时间表。而且随着工作流程的复杂性越来越高,单一的云不能满足要求,对多云环境的工作量也就越来越少。目前大多数算法只适用于单一云,需要在多云上运行 workflow 管理系统;因此,开发满足多云特性的调度算法的机会很大。

参考文献

- [1] Dogan A, Fuè Sun Oè Zguè Ner. Matching and Scheduling Algorithms for Minimizing Execution Time and Failure Probability of Applications in Heterogeneous Computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2002, 13(3):308-323.
- [2] CUI LIN. Scheduling scientific workflows elastically for cloud computing[C]//Cloud Computing (CLOUD), 2011 IEEE International Conference on. IEEE, 2011:746-747.
- [3] Dudin E B, Smetanin Y G. A review of cloud computing[J]. Scientific and Technical Information Processing, 2011, 38(4):280-284.
- [4] Delavar A G, Aryan Y. HSGA: a hybrid heuristic algorithm for workflow scheduling in cloud systems[J]. Cluster Computing, 2014, 17(1):129-137.
- [5] M. A. Rodriguez and R. Buyya. Deadline Based Resource Provisioning and Scheduling Algorithm for Scientific Workflows on Clouds[J]. IEEE Transactions on Cloud Computing, 2014(2):222-235.
- [6] Jiang W Z, Sheng Z Q. A new task scheduling algorithm in hybrid cloud environment[C]//Cloud and Service Computing(CSC), 2012 International Conference on. IEEE, 2012:45-49.
- [7] Pereira W F, Bittencourt L F, da Fonseca N L S. Scheduler for data-intensive workflows in public clouds[C]//Cloud Computing and Communications(Latin Cloud), 2nd IEEE Latin American Conference on. IEEE, 2013: 41-46.
- [8] Selvarani S, Sadhasivam G S. Improved cost-based algorithm for task scheduling in cloud computing[C]//Computational Intelligence and Computing Research(ICCIC), 2010 IEEE International Conference on. IEEE, 2010:1-5.
- [9] Pandey S, Wu L, Guru S M, et al. A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments[C]//Advanced Information Networking and Applications(AINA), 2010 24th IEEE International Conference on. IEEE, 2010:400-407.
- [10] Verma A, Kaushal S. Deadline and budget distribution based cost-time optimization workflow scheduling algorithm for cloud[C]//Proceedings of the IJCA on International Conference on Recent Advances and Future Trends in Information Technology(IRAFIT' 12). 2012:1-4.
- [11] Tordsson J, Montero RS, Moreno-Vozmediano R, Llorente IM. Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers[J]. Future Generation Computer Systems, 2012, 28(2):358-67.
- [12] Liu K, Jin H, Chen J, et al. A compromised-time-cost scheduling algorithm in Swin De W-C for instance-intensive cost-constrained workflows on cloud computing platform[J]. International Journal of High Performance Computing Applications, 2010, 24(4):445-456.
- [13] Abrishami S, Naghibzadeh M. Deadline-constrained workflow scheduling in software as a service cloud[J]. Scientia Iranica, 2012, 19(3):680-689.
- [14] Abrishami S, Naghibzadeh M, Epema D H J. Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds[J]. Future Generation Computer Systems, 2013, 29(1):158-169.
- [15] Mao M, Humphrey M. Auto-scaling to minimize cost and meet application deadlines in cloud workflows[C]//High Performance Computing, Networking, Storage and Analysis(SC), 2011 International Conference for. IEEE, 2011:1-12.
- [16] Pandey S, Wu L, Guru S M, et al. A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments[C]//Advanced Information Networking and Applications(AINA), 2010 24th IEEE International Conference on. IEEE, 2010:400-407.
- [17] Zhang F, Cao J, Li K, et al. Multi-objective scheduling of many tasks in cloud platforms[J]. Future Generation Computer Systems, 2014, 37:309-320.
- [18] 徐立, 梁意文. 云平台下 workflow 任务调度的克隆选择改进算法 [J]. 计算机应用研究, 2015, 32(12):3742-3745.
- [19] 李光智. 云计算环境下 workflow 任务调度策略研究 [D]. 合肥工业大学, 2017.
- [20] 陈超. 改进 CS 算法结合决策树的云 workflow 调度 [J]. 电子科技大学学报, 2016, 46(6):974-980.
- [21] Durillo J J, Prodan R. Multi-objective workflow scheduling in Amazon EC2[J]. Cluster Computing, 2014, 17(2):169-189.
- [22] Vasile M A, Pop F, Tutueanu R I, et al. Resource-aware hybrid scheduling algorithm in heteroge-

- neous distributed computing[J]. *Future Generation Computer Systems*, 2015, 51(C):61-71.
- [23] Kousalya G, Pethuru Raj C, Balakrishnan P. *Automated Workflow Scheduling in Self-Adaptive Clouds*[M]. Springer International Publishing, 2017.
- [24] Ahmad I, Dhodhi MK (1995) Task assignment using a problem genetic algorithm. *Concurr Pract Exp* 7(5):411-428
- [25] Ali S, Sait SM, Benten MS (1994) Gsa: scheduling and allocation using genetic algorithm. In: *Proceedings of the conference on European design automation*, IEEE, pp 84-89
- [26] Hou ESH, Ansari N, Ren H (1994) A genetic algorithm for multiprocessor scheduling. *IEEE Trans Parallel Distrib Syst* 5(2):113-120
- [27] Wu AS, Yu H, Jin S, Lin KC, Schiavone G (2004) An incremental genetic algorithm approach to multiprocessor scheduling. *IEEE Trans Parallel Distrib Syst* 15(9):824-834
- [28] Blythe J, Jain S, Deelman E, Gil Y, Vahi K, Mandal A, Kennedy K (2005) Task scheduling strategies for workflow-based applications in grids. In: *Proceedings of cluster computing and the grid, C-Grid 2005*, vol 2, IEEE International Symposium on 2005, pp 759-767
- [29] Young L, McGough S, Newhouse S, Darlington J (2003) Scheduling architecture and algorithms within the iceni grid middleware. In: *Proceedings of UK e-science all hands meeting*, Citeseer, pp 5-12
- [30] Sonmez O, Yigitbasi N, Abrishami S, Iosup A, Epema D (2010) Performance analysis of dynamic workflow scheduling in multicluster grids. In: *Proceedings of the 19th ACM international symposium on high performance distributed computing*, ACM, pp 49-60
- [31] Xie T, Qin X (2006) Scheduling security-critical real-time applications on clusters. *IEEE Trans Comput* 55(7):864-879
- [32] Xie X, Qin (2007) Performance evaluation of a new scheduling algorithm for distributed systems with security heterogeneity. *J Parallel Distrib Comput* 67(10):1067-1081
- [33] Yu J, Buyya R. A budget constrained scheduling of workflow applications on utility Grids using genetic algorithms[C]// *Workflows in Support of Large-Scale Science*, 2006. WORKS '06. Workshop on. IEEE, 2006:1-10.
- [34] Mandal A, Kennedy K, Koelbel C, et al. Scheduling strategies for mapping application workflows onto the grid[C]// *High PERFORMANCE Distributed Computing*, 2005. Hpd-14. Proceedings. IEEE International Symposium. IEEE Computer Society, 2005:125-134.
- [35] Sakellariou R, Zhao H. A low-cost rescheduling policy for efficient mapping of workflows on grid systems[M]. IOS Press, 2004.
- [36] Blythe J, Jain S, Deelman E, et al. Task scheduling strategies for workflow-based applications in grids[C]// *IEEE International Symposium on CLUSTER Computing and the Grid*. IEEE Computer Society, 2005:759-767.
- [37] Choi J, Adufu T, Kim Y. Data-Locality Aware Scientific Workflow Scheduling Methods in HPC Cloud Environments[J]. *International Journal of Parallel Programming*, 2016:1-14.
- [38] Alkhanak E N, Lee S P, Khan S U R. Cost-aware challenges for workflow scheduling approaches in cloud computing environments: Taxonomy and opportunities[J]. *Future Generation Computer Systems*, 2015, 50(C):3-21.
- [39] 郭凤羽, 禹龙, 田生伟, 等. 云计算环境下对资源聚类的工作流任务调度算法 [J]. *计算机应用*, 2013, 33(8):2154-2157.
- [40] 周业茂, 李忠金, 葛季栋, 李传艺, 周筱羽, 骆斌. 移动云计算环境下基于延时传输机制的多目标工作流调度方法 [J/OL]. *软件学报*:1-20[2018-01-22].<https://doi.org/10.13328/j.cnki.jos.005479>.
- [41] Yassa S, Chelouah R, Kadima H, et al. Multi-objective approach for energy-aware workflow scheduling in cloud computing environments.[J]. *TheScientificWorldJournal*, 2013, 2013(3-4):350934.
- [42] Abrishami S, Naghibzadeh M, Epema D H J. Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds[J]. *Future Generation Computer Systems*, 2013, 29(1): 158-169.