arXiv:2304.03424v1 [cs.DC] 7 Apr 2023

大数据分析中的运行时间变化

YIWEN ZHU,Microsoft，美国RATHIJIT
SEN, Microsoft，美国ROBERT
HORTON,Microsoft，美国JOHN MARK
AGOSTA,Microsoft，美国

云上的资源分配和运行时条件具有动态性，这可能导致作业在多次迭代中的运行时间出现较大变化，从而带来糟糕的体验。识别此类变化的来源，并能够预测和应对这些变化，对于云服务提供商设计可靠的数据处理流水线、配置和分配资源、调整服务定价、满足服务级别目标以及调试性能隐患至关重要。

本文分析了 Cosmos 上数百万个生产环境中的 SCOPE 作业的运行时间变化。Cosmos 是微软内部一个 EB 级分析平台。我们提出了一种创新的两步方法，通过刻画典型的分布形状，并结合分类模型来预测作业运行时分布，平均准确率为 >96%，其表现优于传统回归模型，也能更好地捕捉长尾。我们考察了作业计划特征和输入、资源分配、物理集群异构性与利用率，以及调度策略等因素。

据我们所知，这是首项针对大规模企业分析工作负载预测运行时分布类别的研究。此外，我们还探讨如何运用这些方法分析假设情景，重点关注资源分配、调度和物理集群配置决策对作业运行时间一致性和可预测性的影响。

CCS 概念： • **计算机系统组织** → **云计算**； • **计算方法**→ **因果推理与诊断**； • **信息系统** → 数据分析。

补充关键词和短语：大数据、变化、预测、可解释性、聚类

ACM 引用格式：
Yiwen Zhu、Rathijit Sen、Robert Horton 和 John Mark Agosta。2023。大数据分析中的运行时变化。*ACM* 数据管理会议录 1，1，第 67 篇文章（2023年5月），20 页。https://doi.org/10.1145/3588921

1 引言

过去十年间，大数据平台已无处不在，能够以高效率、高安全性和高易用性实现可扩展的数据处理 [3, 11, 17, 52, 55, 68, 69, 74]。然而，资源配置、调度以及与其他作业共置的动态特性，可能偶尔导致作业减速。此外，作业本身的内在属性（如参数值和输入数据大小）可能会在重复运行之间发生变化，进而导致运行时间出现差异。图 1 展示了 Microsoft 的大数据分析平台 Cosmos [49], 中的一组重复作业，这些作业以不同频率提交。我们可以看到，有些作业的运行时间较为稳定，而有些则会偶尔出现不规律的减速。但目前尚不清楚为何会发生这些变化，以及如何

作者地址：Yiwen Zhu，Microsoft，美国，yiwzh@microsoft.com；Rathijit Sen，Microsoft，美国，rathijit.sen@microsoft. com；Robert Horton，Microsoft，美国，rhorton@microsoft.com；John Mark Agosta，Microsoft，美国，john-mark.agosta@ microsoft.com。

允许为个人或课堂使用而免费制作本作品的全部或部分内容的数字副本或纸质副本，前提是这些副本不得用于营利或商业利益，且每份副本的首页均须附上本声明和完整引文。对于本作品中由作者以外的其他方拥有版权的部分，必须尊重其版权。注明出处后，允许摘录。若要以其他方式复制、再版、发布到服务器或分发至列表，则须事先获得具体许可和/或支付费用。请向 permissions@acm.org 申请许可。© 2023 版权所有归所有者/作者所有。出版权已授权给 ACM。2836-6573/2023/5-ART67 $15.00 https://doi.org/10.1145/3588921

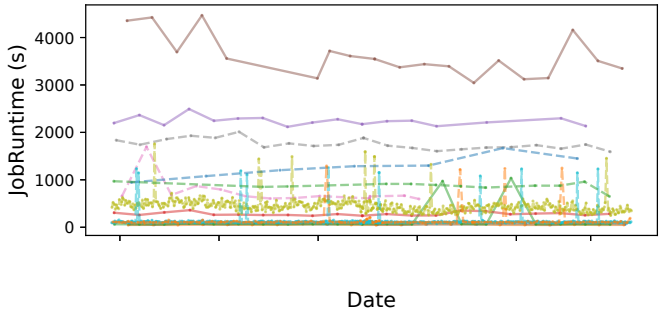


图 1. 运行时间发生变化的重复作业。

可以缓解，或者下一次作业运行相较于历史运行成为异常值的可能性有多大。

在生产系统中，作业通常会基于强数据依赖关系进行调度或流水线处理（作业将其他作业的输出数据作为输入） [13]。作业运行时间的稳定性和可预测性是影响数据处理流水线基础设计与架构的重要因素。然而，尽管作业减速不可避免 [65]，算子往往因评估困难而忽视这些因素。即使拥有海量的遥测数据，云服务提供商仍然默认采用人工分类处理流程，因为难以捕捉影响作业运行时间及其稳定性的复合因素，而这种流程既无法扩展，又容易出错。

尽管先验工作 [20, 56, 85] 已经通过实证刻画了运行时变化，但它们并未提出预测这种变化的方法，也未预测一次新运行相较于平均运行时间或中位运行时间成为离群值的可能性。其他工作（例如 Griffon [65] ）使用机器学习模型来预测数量有限的作业模板的运行时间轻微变慢。它们无法预测表现为离群值的显著变慢。由于机器学习模型在处理离群值方面一向表现不佳，尤其是在离群值出现概率较低的情况下，基于先验时间序列的方法 [43, 67] 并不适用。在本文中，我们旨在通过开发一种新颖且系统化的方法，对作业运行时间变化进行建模、预测和解释，从而实现对特征进行更细粒度的区分，以弥补生产数据分析系统中的这一空白。

在本研究中，我们全面考察了 Cosmos [49]，（微软支持众多 Microsoft 产品 [49]、规模达 EB 级的分析平台）上数百万个生产环境 SCOPE [11] 作业的运行时间变化。我们的主要贡献是提出了一个系统分析、预测和解释运行时间变化的框架，其中包括：

（1）描述性分析：通过考察包括作业固有属性、资源分配和物理集群状况在内的历史数据，我们更好地理解了影响每个作业运行时间变化的因素。具体而言：

（a）我们表明，先前工作{v1}所考察的常用标量指标（例如变异系数（COV））
不足以在存在离群值的情况下刻画变化。
相反，我们提出了一种利用作业
归一化运行时间分布的属性来刻画变化的新方案，该方案能够提供细粒度的信息，例如
作为离群值的概率、分位数以及分布形状。 [章节 4]
(b) 我们创新性地使用 Shapley 值 [66] 来解释变化和
定量分析不同特征的贡献。 [章节 6]
(2) 预测性分析：我们开发了一种基于似然度的创新方法，用于识别具有显著差异的运行时间分布，并以 >96%的准确率预测该分布，优于以往方法 [65]。 [部分 5]。

(3) 规范性 分析：基于预测器，我们对假设情景进行定量分析，并通过限制空闲令牌、安排在更新一代的机器上运行以及更好地进行负载均衡，确定减少变化的潜在机会。 [部分 7]

本文其余部分的组织结构如下。第2节讨论估算和预测运行时间变化所面临的挑战，以及本研究的目标和方法。第3节简要介绍 Cosmos 中的 SCOPE作业、潜在的变化来源，以及我们研究的数据集。第4至7节将介绍上文概述的描述性、预测性和规范性分析。第8节讨论这一领域的相关工作，第9节总结全文。

2 目标、挑战与方法

性能变化通常由经验丰富的工程师基于强假设进行人工分析，而这些假设可能导致结果产生偏差。近年来，云中海量遥测数据的可用性（其中既包含作业特征信息，也包含物理集群的状态信息），以及数据分析方法的兴起，使人们期待能够通过更系统、更严谨的方法改进这一过程。

我们的目标是在单个 作业层面评估和预测运行时间变化，并采用定制化且针对特定用例的度量方式，使客户在监控和规划方面获得更有价值的信息。正如与项目经理（PMs）的多次交流所证实的那样，这也是客户非常希望获得的一项指标。我们还希望向用户提供有关变化的丰富信息，例如作业运行时间超过某个极端值的概率，或运行时间分布的各种定量属性，如分位数和离群值；使用传统的机器学习方法很难捕捉这些信息。

此外，分布式系统中计算作业的性能建模很困难，尤其是在关注可靠性时，原因在于以下挑战。

• **复杂的环境因素 (C1)**：云计算平台中的资源共享会增加作业运行时建模的复杂性，这是由于存在吵闹邻居和其他不断变化的环境因素。手动方法无法传递各个计算节点的动态状况，也无法揭示导致性能下降的潜在问题。

• **罕见事件的存在（C2）**：对于导致离群值和运行时间分布长尾的罕见事件（例如偶发的服务中断），很难为重复执行的作业收集足够的离群值观测数据，以便准确估计其分布。因此，能够利用其他拥有充足观测样本的作业组中的作业实例所积累的学习成果至关重要。

• **缺乏适当的度量指标（C3）**：对于运行时间所具有的典型长尾分布，如何衡量变化仍然是一项挑战，因为传统的基于方差的度量无法捕捉所关注的极端值。通常用于评估运行时变化的指标（如 COV）不足以捕捉各种运行时分布的详细特征。

• **缺乏带标签数据（C4）**：尽管大多数用于预测性分析的机器学习方法都需要带标签数据，但没有记录运行时间分布或作业减速的原因标签。对于出现次数较少的新作业，手动评估运行时间可靠性以确定其所属的分布类别同样不可行。

我们在这项工作中的两步方法是，基于重复作业的归一化运行时间分布来刻画并预测变化（见图 2）。利用作业级别和机器级别的信息，我们:

• **刻画运行时间分布**：我们的聚类分析 采用一种新颖的特征化方案，对具有明显多样化运行时间分布的 [5] 历史作业进行聚类。我们将

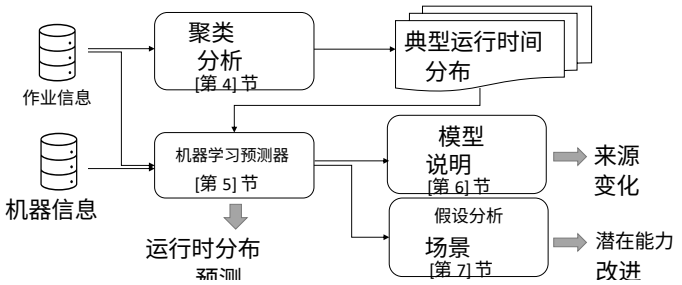


图 2. 框架。

作业与其所属的某个分布相关联，采用创新且自适应的后验似然方法。对于每种类型（单峰、多峰）的分布，我们定义了关键指标来描述该分布，并以用户易于理解的数值形式量化其变化 [挑战**C3**，第 4] 节。

• **开发** 机器学习预测器：我们利用机器学习（ML）分类技术来预测作业最有可能属于哪种运行时间分布，同时考虑作业属性、资源分配以及系统负载等可能导致吵闹邻居的环境条件 [**C1**]。为克服挑战 **C2** 和**C4**，我们使用在较长时间间隔内对分布的观测值以及具有更多重复执行的作业来开发模型（数据集**D1**，表**1**），同时该模型可应用于任何新作业。 [章节 5]

• **解释预测结果**：我们使用基于特征贡献算法的模型解释器，以更好地理解与运行时变化 [挑战**C1**相关的各种因素，见第 6] 节。
• **分析** 假设情景：基于预测模型，我们提出假设情景，并定量评估运行时间性能的潜在提升 [章节 7]。

虽然作业运行时间的点预测本身就是一个重要且具有挑战性的问题 [19, 28, 29, 46–48, 51, 82], ，但我们希望预测重复作业运行时间的潜在变化，而不是绝对运行时间。因此，直接预测作业运行时间并非本研究的目标。

3 平台和数据集

Cosmos [49] 是微软自2002年起开发的艾字节级大数据平台，在全球多个数据中心拥有超过300k台机器 [83]。该系统使用基于 YARN 的 [73] 资源管理器，每天处理来自数万名微软内部用户的 >600k 个作业。它是一个大型内部共享集群，效率至关重要。过去几十年来，众多研究项目和工程工作提升了其效率、安全性、可扩展性和可靠性 [6–9, 11, 14, 16, 21, 32–34, 50, 52, 64, 76, 80–83, 83]。

Cosmos 作业可以使用一种类似 SQL 的方言 SCOPE [6] 编写，并大量使用 C# 和用户定义函数（UDF）。提交后，作业会被编译为一个经过优化的执行计划，即算子有向无环图，并分布到不同的机器上。每个作业由多个顶点组成，也就是将在分配给某台物理机器的容器上执行的独立进程。

3.1 作业组

我们的工作重点是理解并预测我们归入 作业组 的作业在重复运行时运行时间的变化。只有作业会重复执行时，变化才有意义（即样本量 > 1）。先前研究 [14, 33, 82]表明，Cosmos 上有 40–60% 的作业会重复执行。

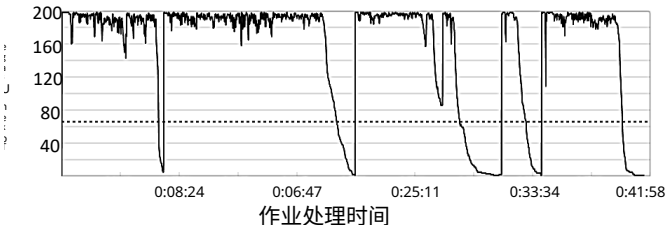


图 3. 作业运行期间的令牌用量示例。

其他研究 [20] 也报告称，其系统中有相当一部分作业会重复执行。我们通过匹配一个由以下内容组合而成的键来识别重复执行。

- 规范化作业名称，其中已移除提交时间和输入数据集等特定信息 [32, 82]。
- 作业签名 [32], ，这是在编译计划中的算子 DAG 上递归计算得出的哈希值。该签名不包含作业输入参数。

因此，我们得到作业组，每个组中包含作业实例，它们对应于作业的重复执行。同一作业组中的作业实例具有相同的键值。

3.2 变化来源

在每个作业组中，作业实例的运行时间可能会因以下任何原因而有所不同：

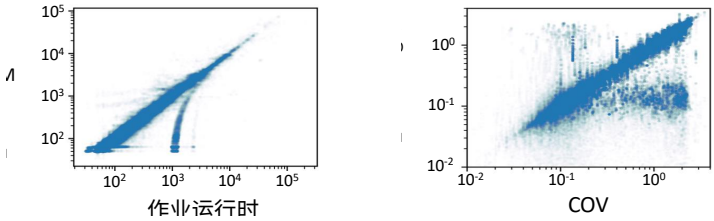
内在特征。用于对作业分组的键包含执行计划的信息（例如算子类型、估计基数、算子之间的依赖关系），但不包含作业输入参数（例如过滤谓词的参数）或输入数据集。不同实例的这些参数、数据集及其大小可能各不相同。如果参数发生变化时编译计划没有随之改变，就可能导致组内运行时间不同。在我们的数据集中，我们观察到，同一作业组内的输入数据大小最多可相差 50 倍。

资源分配。在 Cosmos 中，资源分配的单位是令牌 [49], ，其概念类似于容器。作业可获得的令牌数量由用户在提交作业时指定，也可以由系统推荐 [63]。为了进一步提高现有基础设施的利用率，未使用的资源会被重新用作可抢占备用令牌 [7]，供任何作业自由使用[1]。这些空闲令牌的可用性难以预测，且会对运行时间产生显著影响。图 3 展示了一个分配了 66 个令牌的 Cosmos 作业的令牌使用天际线（虚线）。加上空闲令牌，该作业在整个处理期间最多总共使用了 198 个令牌。

作业使用的令牌最大数量取决于其可利用的并行性，同时受分配的令牌数量限制。在观察 Cosmos 上各种工作负载的执行情况时，我们发现同一作业组中的最大令牌数量可能相差 10 倍。同样，所分配资源的特征也存在差异。令牌映射到计算节点上的计算资源，而这些计算节点使用不同的库存单位（SKU）。经过十多年的演进，Cosmos集群由 10–20 种具有不同处理速度的 SKU组成 [83]。在我们的数据集中，我们观察到同一作业组中的不同作业实例会同时在一至九种不同的 SKU 上运行。

物理集群环境。最后，物理集群环境也会导致运行时间出现变化。这既包括空闲令牌的可用性（如上所述），也包括

[1]空闲令牌的使用量受用户指定的分配上限限制。



(a) 中位数与实例运行时间 (b) 历史 COV 与 COV

图 4. 历史中位数与作业运行时间之间的相关性 (a)，以及历史 COV 与所有观测值的 COV 之间的相关性 (b)。

各台机器上的负载。利用率（负载）越高，共享资源的争用可能越严重；负载范围越大，运行时间变化也可能越明显。

3.3 数据集

为了深入了解上述讨论的变化来源，我们通过以下方式收集数据：i) 使用 Peregrine 框架 [32] 提取 SCOPE 优化器估算的内在特征信息，例如计划中的算子数量、输入数据大小、基数、成本等；ii) 使用 KEA 框架 [83] 从作业执行日志中获取令牌使用信息，以及 SKU 和机器负载信息；iii) 通过匹配作业 ID、执行各顶点的机器名称以及相应的顶点开始/结束时间，将所有这些信息关联起来。

表1. 本研究使用的数据集。

数据集	时间间隔	作业组	作业实例	支持度
D1	6个月	>9K	>3M	20
D2	15天	>11K	>700K	3
D3	5天	>11K	>200K	3

表1总结了我们研究中使用的数据集。这些数据集由相应时间间隔内运行的部分作业组成；当每组的实例数（支持度）超过最低阈值时，便将其纳入。以最少3次出现为支持度，纳入了53%的作业。在本研究中，我们只关注批处理作业，而不关注 Cosmos 还支持的流式作业或交互式作业。我们使用数据集D1来识别并分组出现次数较多的作业的运行时间分布（见第4.2节，>20）。我们使用D2训练运行时变化预测器，并使用D3测试其准确性（第5节）。

4 运行时间变化特征分析

现在，我们将讨论如何刻画和量化重复作业的运行时变化。这将构成我们在第5节讨论的预测策略的基础。

4.1 标量指标

如前文（第 1 节）所述，作业的平均运行时间无法充分揭示重复运行之间的差异，也无法说明该作业下一次运行需要多长时间。因此，它对于刻画、预测或解释这些差异并不十分有用。

接下来，我们研究作业的中位运行时间与该作业多次重复运行时的运行时间之间的相关性。图4a展示了使用数据集D2、在对数尺度下，作业各次重复运行的运行时间与其历史中位数的对比。

我们在图4a中观察到两种截然不同的模式：一组点聚集在对角线附近，表明各个作业实例的运行时间与中位数具有良好的相关性；另一组点则以类似钟乳石[2]的形态单独聚集。这些钟乳石状点所对应的运行次数远慢于中位运行时间，并形成运行时间分布的（长）尾部。这类运行很少见（占所有运行次数的比例不到5%），且中位数越大，发生概率越低。然而，我们发现很难预先判断一次新的运行最终会落在对角线上还是钟乳石状区域。即使中位运行时间稳定且已知，这两种模式的存在，以及难以预测新的运行会属于哪一种模式，也使中位数不适合用于预测运行时间或描述其变化。对于历史运行时间的平均值和 95th 百分位数，也能发现类似趋势。

变异系数（COV）是另一种常用的变化度量指标，定义为标准差与平均值的比率（无单位）。COV 易于计算和解释，先前的研究 [56] 已使用 COV 来描述作业运行时间的变化。但 COV 存在若干局限：

- **偏差**：我们研究的 SCOPE 作业运行时间从几秒到几天不等，平均值差异显著。这可能导致 COV 产生偏差，而且对于运行时间较短的作业，始终可能观察到非常大的 COV。
- **不稳定性**：离群值的存在会使平均运行时间增加（在如此大规模的分布式系统中，有些作业偶尔运行缓慢在所难免）。因此，随着数据集中加入更多作业，COV 可能会变得不稳定。与平均值不同，COV 不会随着样本量增大而收敛，因此不是一致估计量 [22]。
- **粒度较粗**：COV 无法捕捉分布的许多特征，例如其形状（如单峰、双峰以及是否存在离群值）。因此，它难以细致地解释变化。

图 4b 展示了对于数据集D2中的每个作业实例，基于历史运行计算得出的 COV（y 轴）与根据 D3 中的观测结果、基于所有运行计算得出的 COV（x 轴）之间的吻合程度。与上文对中位数的讨论类似，我们看到多个点群：相同的历史 COV 对应着新运行中的不同 COV 值，因此很难预测某次新运行会属于哪个点群。此外，COV 指标也存在上述局限。

总体而言，我们发现，平均值、中位数、分位数和 COV 等标量指标本身不足以理解或预测运行时变化。

4.2 分布

我们用每个重复作业组的运行时分布来表示其运行时变化。尽管 SCOPE作业的运行时间差异很大，我们发现许多不同作业的运行时间具有相似的概率分布。我们将这些称为形状。只要知道某个作业的分布，就足以确定其变化的任何特征，包括其运行时间超过指定阈值的风险。

为计算形状，我们首先对作业运行时间进行归一化，然后计算其经验概率质量函数（PMF，即直方图）。作业会根据其运行时分布的相似性进行聚类，对于任何新作业，我们都可以预测它所属的簇。我们将作业的概率质量函数（PMF）视为其所属簇的概率质量函数。这种方法使我们能够将分析推广到不同作业，并转而使用少量用户易于理解的簇。

[2]石笋是从洞穴地面向上生长的岩石结构，有时会延伸至洞顶 [1]。

我们使用以下两种归一化策略转换作业运行时间，并采用表1所示、基于数据集D1中的“历史”数据计算出的中位数：

定义*4.1*。**比率归一化** 定义为作业运行时间与其历史中位数的比率，即作业运行时间/中位运行时间。**Delta 归一化**定义为差值，即作业运行时间 - 中位运行时间。

比率归一化分布衡量运行时间的相对变化，而 Delta- 归一化分布衡量相对于中位数的绝对偏差，以秒为单位进行测量。请注意，具有不同范围的运行时间可以更有效地使用比率归一化进行归一化。例如，长时间运行的作业的绝对变化通常更高（1小时±10分钟），而短时间运行的作业的绝对变化可能低得多。在这方面，比率归一化提高了一致性，并将运行时间范围不同但具有可比性的分布归并在一起。另一方面，对于非常短或非常长的作业，以百分比衡量方差可能不太有意义。对于短时间运行的作业，百分比可能非常大（例如，5秒±300%）。对于长时间运行的作业，百分比可能非常小，从而导致通过比率归一化测得的分布非常“窄”。因此，在本文中，我们结合使用 Delta 归一化和比率归一化，以绝对值的形式捕捉变化。

为了恢复各作业间“典型”的分布形状，我们使用数据集D1进行聚类分析，其中仅纳入包含 >20 次出现的作业，以便更准确地估计其运行时分布。对于每个作业组，我们推导出其归一化运行时间分布的直方图，然后使用无监督机器学习算法对其进行聚类。请注意，聚类分析的输入是直方图中每个区间的概率质量函数概率，而不是作业特征（例如输入大小等）。

我们的运行时分布聚类方法的主要设计选择如下：

• **分箱大小和范围**：范围应覆盖大多数值，具有相对精细的粒度，但也不能过小，以免捕捉到由噪声引起的波动。我们将离群值合并到一个分箱中（根据是否 ≤ 或 ≥ 某些阈值）[3]。我们评估了50、100、200和500个分箱，并选择了200个分箱，此时概率质量函数曲线相对平滑，且可以观察到不同形状的分布。

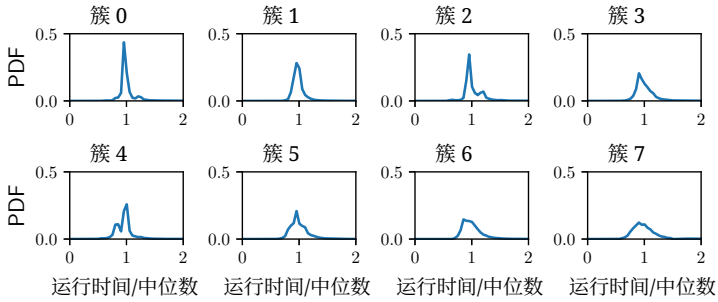
• **聚类算法**：基于树状图的层次聚类 [44] 和凝聚聚类 [54] 采用不同的距离度量、链接方法以及用户指定的簇数量。然而，它们会产生不均衡的簇（某些情况下，单个簇包含 >90 % 的数据）。K 均值聚类 [62] 产生的簇更加均衡，因此选择该方法进行后续分析。

• **簇数量**：簇数量根据以下因素确定：(i) 对惯性进行数值分析。惯性定义为每个样本与其所在簇质心之间距离的平方和（我们选择一个肘部点，在该点增加更多簇不会显著降低惯性）；(ii) 通过目视检查聚类结果，确认各簇之间具有足够差异并具备独特特征。

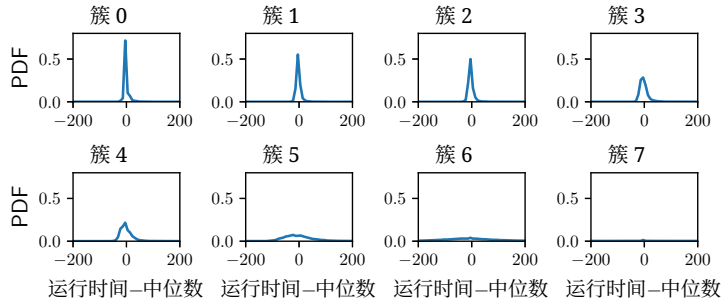
• **直方图平滑**：标准聚类算法将概率质量函数（PMF）概率作为输入向量，并假设各维度相互独立。实际上，相邻分箱的密度值（例如运行时间落在第 4 或第 5 个分箱中的概率）彼此相关。然而，使用任何距离度量（例如点积）时，都不会考虑相邻分箱之间的相关性。因此，我们在得出概率质量函数后引入平滑步骤，以共同考虑相邻分箱的值，从而提高上述两个平滑后向量之间的亲和度。

[3]对于 Delta 归一化，我们使用 [-900, 900]（其中 1% 的作业比中位数慢 1066 秒，我们向下取整至 900 秒，即 15 分钟）；对于比率归一化，我们使用 [0, 10]（其中 1% 的作业速度比中位数慢 10.6 倍，我们向下取整至 10 倍）。比中位数慢 >900 秒或 10 倍的作业被定义为离群值。

图 5 展示了采用比率归一化和 Delta 归一化策略时 8 个簇的分布。我们发现，某些分布具有两个众数（例如，采用比率归一化的簇 0、2 和 4），且方差各不相同。表 2 汇总了每个簇的重要统计量。例如，采用比率归一化的簇 0 的异常值概率为 1.63%（定义为比率归一化运行时间比中位数慢 ≥ 10 倍）；第 25 百分位数与 75th 百分位数之间的差值为 0.06；该分布的第 95th 百分位数为 1.41，标准差为 2.46。采用比率归一化的簇 7 的异常值概率降至 0.06%。各簇根据第 25th 百分位数与第 75th 百分位数之间的差值进行排序。



(a) 比率归一化



(b) Delta 归一化

图 5. 归一化运行时间的典型分布。

表 2. 运行时间分布簇的统计量。

	比率					增量				
cid	异常值 (%)	25−75th	95th	std		cid	离群值 (%)	25 − 75th (s)	95th (s)	标准差 (s)
0	1.63	0.06	1.41	**2.46**		0	1.93	4	28	155
1	0.42	0.11	1.2	0.93		1	0.49	11	19	140
2	**1.66**	0.16	1.37	2.18		2	0.53	11	23	148
3	0.25	0.17	1.29	1.45		3	0.55	16	33	140
4	1.46	0.17	1.35	1.94		4	0.98	31	63	153
5	0.25	0.19	1.34	0.82		5	0.73	69	128	179
6	0.26	0.20	1.37	0.97		6	2.43	199	408	296
7	0.06	**0.29**	**1.46**	0.55		7	**24.23**	**936**	**1359**	**2548**

5 预测运行时间变化

我们现在基于可解释机器学习开发一个预测模型，用于预测作业的运行时分布的形状（如图5所示）。我们使用数据集D2作为训练集，D3作为测试集。

5.1 特征选择

我们考虑三类在作业编译时可用的预测特征：从作业执行计划中派生的特征（“内在”特征）、表示作业过去资源使用情况统计量的特征，以及描述作业将要运行的物理集群负载的特征。下面介绍这些类别。

内在特征。 我们利用查询优化器在编译时获得的作业执行计划信息 [32] 作为输入，该信息可以表明查询类型、数据模式及其潜在的计算复杂度。其中包括每种类型的算子数量（例如 Extract、Filter）、估计基数等。对于新提交的作业，其详细的输入数据大小是未知的，估计基数可能会存在较大偏差 [82]。因此，利用同一作业组的历史作业实例，我们提取总数据读取量、临时数据读取量的统计量，以及与执行计划相关的统计量，作为能够反映作业规模的额外输入特征。

我们还将运行在每个 SKU上的顶点占比作为输入特征，该占比反映了每个 SKU的资源消耗。此前的一项研究 [83] 表明，在 Cosmos 中，某些较新的 SKU 处理数据的速度可能快于其他 SKU；因此，我们认为，在不同 SKU 上执行的顶点占比会影响运行时分布。

资源分配。 令牌分配是衡量特定作业所利用资源的良好指标，因此会影响运行时间。然而，[63] 发现，用户经常会过度分配（例如，用户选择分配1000个令牌，但实际峰值用量仅为600个）。在这项工作中，我们将历史令牌利用率与令牌分配相结合，作为输入。对于同一作业组的历史作业实例，我们提取资源利用率（根据图3中的天际线计算令牌用量的最小值、最大值和平均值），并将历史统计量作为特征（历史平均值和标准差）。我们还为闲置令牌创建了一个新变量（历史平均值）。与实际利用率相比，模型学会了降低令牌分配这一特征的重要性；我们也通过第5.2节中的沙普利分数证实了这一点。

物理集群环境。 作业运行时可能会受到执行其顶点的机器利用率的影响——更高的利用率水平表示机器负载更高，也更可能出现与吵闹邻居和资源争用相关的严重问题。因此，我们提取作业提交时间每个 SKU中相应机器的CPU利用率水平作为输入。与现有方法相比，例如 [65] 和 [85], , 纳入新的实时机器状态信息可以提高模型准确率（见第5节）。

目前正在记录每个容器的 CPU 和内存使用情况，其中包含有关特定作业的更多针对性信息，相较于机器级计数器，也能更准确地反映资源消耗 [25]。这些信息一旦可用，就能轻松替换或添加到我们的模型中。我们预计它们将成为作业运行时间的有力指标，因为它们能够捕捉更多作业级特征。它们还可能揭示 CPU 密集型或内存密集型作业是否更容易出现较大的运行时间方差。

5.2 簇归属预测

对于即将提交的新作业，我们希望根据编译时可用的信息预测 其运行时分布形状。这自然对应于一个分类问题，其中预测

目标是将每个作业映射到特定的分布形状（例如，图5中的8种不同形状之一）。

基于后验似然的簇归属。对于每个作业实例，为确定其所属的分布形状，我们利用分析期间具有相同作业组（即相同作业名称和执行计划）的一组相似作业实例，推导出该作业组的经验概率质量函数（PMF），即运行时分布的直方图。基于少量运行时观测值，我们可以推导出这些观测值来自图 5 所示任一预定义分布形状的后验似然。

基于贝叶斯定理 [5], ，具有 N 个运行时观测值 $x_{n=1\cdots N}$的作业组属于簇 $z_{i=1\cdots K}$的后验对数似然，可以根据这些 N观测值的 PMF $\phi_{h=1\cdots H}$以及 $K = 8$ 个预定义簇的 PMF $\theta^{i=1\cdots K}_{h=1\cdots H}$推导得出，并且能够适应更大的样本量：

$$p(x_1, x_2 \cdots x_N|z_i) = \prod_{n=1\cdots N} F(x_n|\theta^i_{h=1\cdots H}) \tag{1}$$

$$= \prod_{n=1\cdots N} \theta^i_{h(x_n)} \tag{2}$$

$$\log p(x_1, x_2 \cdots x_N|z_i) = \sum_{n=1\cdots N} \log(\theta^i_{h(x_n)}) \tag{3}$$

$$p(z_i|x_1, x_2 \cdots x_N) = \frac{p(x_1, x_2 \cdots x_N|z_i)p(z_i)}{\sum_{i=1\cdots K} p(x_1, x_2 \cdots x_N|z_i)p(z_i)} \tag{4}$$

$$= \frac{\prod_{n=1\cdots N} \theta^i_{h(x_n)}}{\sum_{i=1\cdots K} \prod_{n=1\cdots N} \theta^i_{h(x_n)}} \tag{5}$$

$$\sim \prod_{n=1\cdots N} \theta^i_{h(x_n)} \tag{6}$$

$$\log p(z_i|x_1, x_2 \cdots x_N) = \sum_{n=1\cdots N} \log\left(\theta^i_{h(x_n)}\right) - \text{constant} \tag{7}$$

$$= \sum_{h=1\cdots H} n_h \log\left(\theta^i_h\right) - \text{constant} \tag{8}$$

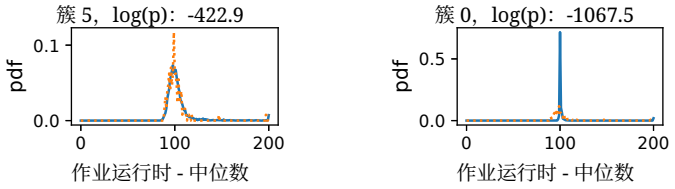
$$\sim \sum_{h=1\cdots H} \phi_h \log\left(\theta^i_h\right) \tag{9}$$

其中，

H：为每个分布推导概率质量函数时使用的离散区间数，该数值对所有分布均为常数。

$\theta^{i=1\cdots K}_{h=1\quad H\cdots}$：簇 i 的归一化运行时分布参数，具体而言，是区间 h的概率质量函数值。

$\phi_{h=1\cdots H}$：基于特定作业组观测值的分布参数（即$x_{n=1,2\cdots N}$），具体而言，是概率质量函数中区间 h 的概率。$h(x_n)$：观测值 x_n 所属的区间索引。n_h：属于区间 h的作业组运行时间观测值（即 $x_{i=1\cdots N}$）的数量。$x_{n=1\cdots N}$：运行时间观测值 n，其中 $x_{n=1\cdots N}|z_{i=1\cdots K} \sim F(\theta^{i=1\cdots K}_{h=1\quad H\cdots})$。$p(z_i)$：对每个簇的概率所设的先验，假定所有簇的先验均为常数（非信息先验 [5]）。



（a）最可能的簇 （b）最不可能的簇

图 6. 似然值示例（值越高表示越可能）。

值得指出的是，对数后验似然与特定作业组的观测值概率质量函数（即 ϕ_h）和预定义的 8 个簇之一的概率质量函数（取对数后，即θ_h^i）的点积成正比（见公式 9）。

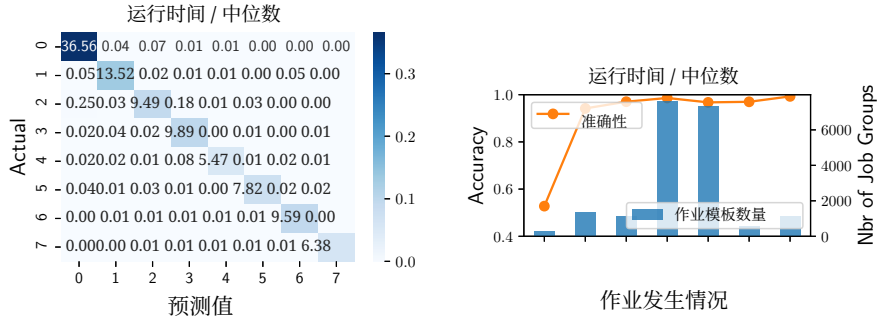
图 6 展示了一个作业的归一化运行时分布（经过 Delta 归一化）的示例，该作业出现了 10 次，并与 2 个簇进行比较。虚线表示该作业组观测值的概率质量函数，即 ϕ_h；实线表示预定义簇的概率质量函数，即 θ_h^i。我们可以看到，对数似然最高（-422.9）的簇 5 具有最相似的形状（见图 6a），而簇 0 的形状最不相似（见图 6b）。随后，将每个作业实例及其所属作业组关联到似然最高的簇标签，并将该标签作为预测目标（标签）。这种簇关联算法总会将作业归入最可能的簇。我们观察到，观测值较少的作业可能与任何现有簇的运行时分布都不相似。因此，我们只关注样本充足的作业组。按照第4.2节所述，我们采用惯性曲线来调整簇的数量，以避免过拟合。

分类模型。根据输入，我们进行（1）基于特征重要性的被动—激进式特征选择 [78]，以避免使用相关特征；（2）参数扫描，为分类算法选择最佳超参数，例如基于树的算法中的树数量；以及（3）使用 RandomForestClassifier [61], 、LightGBMClassifier [41] 和 EnsembledClassifier [57] 进行拟合，通过结合一组常用的分类算法，例如使用软投票的 RandomForestClassifier、LightGBMClassifier、GradientBoostingClassifier [59], 、GaussianNB [58],和 XGBClassifier [60], 。需要注意的是，众所周知，RandomForestClassifier 和 LightGBMClassifier在使用表格数据执行机器学习任务时具有较高的准确率，尤其是在样本外测试中。在本研究中，在各分类器中，LightGBMClassifier 的准确率最高，因此本文其余部分报告其结果。通过分析预测结果，我们注意到：

通过考察输入特征的基尼重要性 [38] ，我们发现，与计算复杂度和输入数据大小相关的特征（例如顶点数量和读取的数据量）十分重要，与历史运行时观测值相关的特征也同样重要。令牌利用率（例如最大值）和编译时间信息（例如基数估计）也很重要。机器的 CPU 利用率也会影响预测，这与我们的观点一致：物理集群环境会影响作业运行时间的变化。我们还注意到，许多算子的重要性较低。顶点总数不如数据大小（读取数据总量或与基数相关的指标）重要。这可能是因为每个顶点处理的数据量存在巨大差异。第 6 节将进一步详细探讨一些特征的贡献。

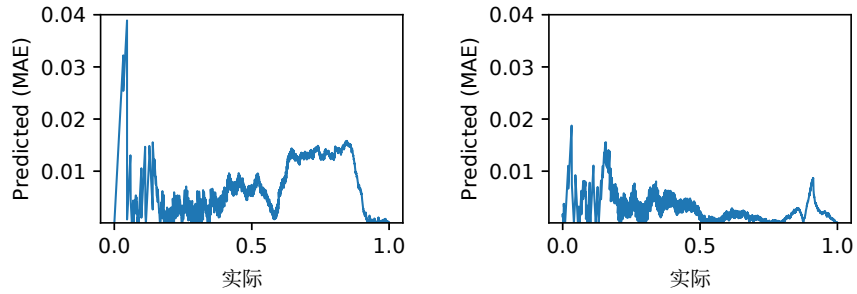
启示：从模型中学到的特征重要性大体上与我们的预期一致。

图 7a 展示了测试数据上的混淆矩阵，对比预测标签（横轴）与真实标签（纵轴），每个单元格表示各类别作业所占的比例。预测结果



(a) 混淆矩阵 (b) 按出现次数统计的准确率

图 7. 比率归一化的预测准确率。



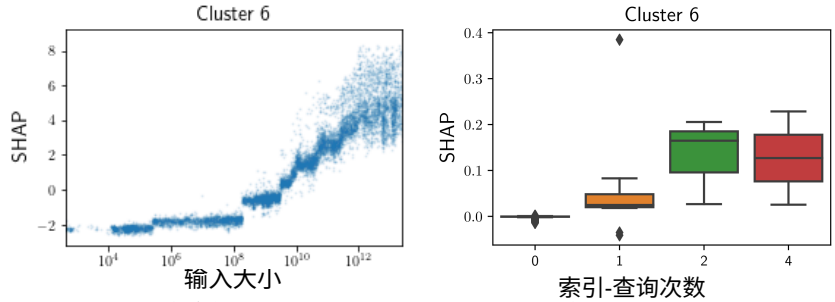
(a) 回归 (b) 所提出的方法

Fig. 8. Delta 归一化与传统回归模型的预测准确率比较 .

同时采用比率和 Delta 归一化时，整体准确率达到 >96%。图 7b（橙色线）展示了具有不同历史出现次数的作业的准确率。可以看出，历史出现次数越多，预测准确率越高，这表明通过添加来自同一作业组的更多观测值，可以进一步优化模型预测。蓝色柱形表示按作业出现次数（1-5 次、6-10 次等）划分的作业组数量。可以看出，在分析期间，大多数作业有 16-50 个历史观测值。Delta 归一化也呈现出类似趋势。

洞见： 使用比率和 Delta 归一化的模型预测均具有很高的准确性。

我们扩展了 [65] 提出的传统随机森林回归模型，将更多查询优化器信息和近实时机器状态信息作为特征，并以作业运行时间作为标签进行预测。图 8 使用分位数-分位数图 [24], 比较了所提出方法（使用分类模型预测分布形状）和回归模型对所有作业运行时间分布的预测结果与实际作业运行时间分布，并在 y 轴上绘制平均绝对误差（MAE）。如果两个分布相同，绘制的分位数应当对齐，且 MAE=0。我们可以看到，与传统回归模型（图 8a）相比，所提出的分类模型（图 8b）准确度更高，尤其是在较高百分位数处，因为它能更好地捕捉作业簇分布中离群值的存在（见图 5）。Kolmogorov–Smirnov 距离 [35] 也降低了 9.2%，表明准确度有所提高。



(a) 按输入大小 (b) 按算子数量

图 9. SHAP 值分布。

洞察： 所提出的方法在预测离群值方面优于现有模型。

6 解释运行时变化

本节将开展描述性分析，以便更好地理解导致不同运行时分布的作业特征。我们从上一节的分类模型出发，使用机器学习解释工具来更好地了解运行时变化的来源。

6.1 沙普利值

Shapley 值 [39, 66] 用于解释博弈论框架中每个“参与者”的贡献，已被用于解释机器学习模型中特征的贡献。在我们的语境中，基于第5节中开发的预测变量，它们通过随机置换其他特征值并评估预测结果的边际变化 [42]，来解释每个特征的定量贡献。

图 9a 展示了相对于读取的输入数据总量的沙普利值分布，其中每个点对应一个作业实例。我们可以看到，使用 Delta 归一化时，输入大小较大的作业更有可能属于簇 6（因为它们的特征值会带来更高的沙普利值，因此属于簇 6 的可能性也更高）。请注意，簇 6 的方差相对较大，且出现离群值的概率较高。对于令牌较少的作业，也可以发现类似趋势。

见解： 输入规模较大且使用令牌较少的作业，更有可能出现较大的变化。更多令牌可能会将同一机器上的其他作业挤出，从而有望减少干扰以及吵闹邻居的影响。

同样，算子数量等作业特征也会显著影响预测，这表明某些算子的存在更有可能导致不同的运行时间分布（参见图 9b，其中展示了采用 Delta 归一化的簇 6 的 Shapley 值）。

见解： 某些算子的数量，例如索引查找、窗口和范围，会增加变化。

通过比率归一化，簇 0 的方差和离群值概率均低于簇 2，而两者都有两个众数。聚焦于一组重要性较高的特征，我们比较了用于预测这两个簇的 Shapley 值，发现与簇 2 相比，CPU 利用率较低、标准差较小且空闲令牌使用量较低的作业更可能属于簇 0（其性能更可靠）。总体而言，我们预计利用率较高或标准差较大的机器性能会较不稳定，这与

我们在此处的观测结果一致。空闲令牌的使用（其可用性较难预测）也可能导致运行时间更加不稳定。

> **洞见：** 降低 CPU 利用率（负载）、降低标准差并减少空闲令牌的使用，可以提高运行时间的可靠性。

对于比率归一化，在 Gen5 和 Gen6（较新一代）上增加顶点数量，往往会使预测倾向于簇 0 和簇 1，这表明在这些机器 SKU 上运行顶点，可能有助于减少运行时变化。与 Gen3 和 Gen4 机器相比，这些机器通常速度更快，资源容量也更大 [83]。

> **洞见：** 模型识别出某些 SKU，在这些机器上增加顶点数量会提高归入簇 0 和簇 1 的可能性，而这两个簇的方差较小。

7 控制运行时变化

利用预测模型（第5节）并结合沙普利值（第6节）所揭示的洞见，我们现在确定了若干用于调度和资源分配的假设情景，并评估其表现。根据作业运行时间的变化，可以定量评估具体的经济影响。

7.1 情景1：空闲令牌分配

空闲令牌是否可用取决于集群的物理条件，而这些条件会受到其他作业执行的影响，因此构成一种变异来源。这里，我们使用模型估算不分配空闲令牌对运行时变化的影响。

借助预测器，我们在测试集（如表1所示的数据集D3）中为所有作业禁用空闲令牌。使用比率归一化后，原先预测为簇 2 的作业中有 15% 现在被归入簇 1； 25th 与 75th 百分位数之间的离群概率差距，以及归一化运行时间的 95th 百分位数均有所降低（另见表2）。第二个显著变化是从簇 3 转为簇 5：第 25 百分位数与 75th 百分位数之间的差距略微增加，从 0.17 升至 0.19，而标准差则大幅下降（从 1.45 降至 0.82 ）。Delta 归一化也能观察到类似变化。根据预测器的结果，对于希望降低方差的作业集合，我们建议禁用其空闲令牌，以最大限度地提升性能。在生产系统中，目前正在开展相关工作，以减少空闲令牌数量上限相对于已分配令牌数量的倍数。我们观察到，空闲令牌较少的作业运行速度较慢，但方差较小，这与模型的预测一致。

7.2 场景2：在后续一代机器上进行调度

一个作业的顶点可以由多台机器以分布式方式执行，正如第3.2节所述，同一组中的不同作业实例可以分配到许多不同的SKU。在这里，如果我们在后续代的机器上执行更多顶点，就可以估计其对运行时变化的影响。

通过将所有顶点（包括比例和计数）从 Gen3.5 转移到 Gen5.2，20.95% 的作业最有可能从簇 2 变为 0，且比率归一化的第 25 至 75th 百分位数之间的差距显著缩小。对于 Delta 归一化，最有可能的预测变化是从簇 1 变为 0，第 25 至 75th 百分位数之间的差距从 11 秒缩小至 4 秒。

因此，在新一代 SKU 上运行更多顶点会更好。不过，我们的模型无法捕捉工作负载重新平衡带来的变化累积效应，例如 CPU 利用率水平的变化。可以轻松集成能够预测不同工作负载分布下利用率的模型，例如 KEA [83], ，以定量捕捉这种动态影响。

7.3 场景 3：改善负载均衡

如第 3.2 节所述，物理集群的状况（例如各机器之间的负载差异）是运行时间变化的一个来源。这里我们估算负载分布更加均匀所带来的影响。

如果 CPU 利用率的标准差能够降至 0（即所有机器在任何时候的负载都相等），采用比率归一化时，最可能的变化是从簇 2 转为簇 0（占 29.78% 的作业）；异常值概率从 1.66% 降至 1.63%，而通过 25th 和 75th 百分位数之差衡量的变化幅度则从 0.16 降至 0.06。Delta 归一化也能看到类似的改善。因此，更完善的调度系统能够带来显著的经济价值，这也进一步推动了 Cosmos 的其他研究项目。

8 相关工作

许多研究探讨了在不同资源分配和平台参数设置下预测作业运行时间的问题 [19, 26, 28, 28, 29, 46–48, 51]。Morpheus [33] 检查重复作业的资源使用情况，并找出最匹配的模式，用于更好的资源分配和调度，但不会预测给定作业的异常值概率。运行时间估计也被用于推断作业调度，以支持策略，例如最短处理时间优先（SPF） [23, 36, 70, 84]。Zrigui 等人 [85] 根据运行时间将作业实例聚类为小型和大型，并使用分类算法以高准确率为其作业调度器提供信息。在我们的工作中，我们针对具有更丰富信息（百分位数、方差等）的 delta 和比率归一化运行时分布，得出了更多的簇，从而全面了解客户体验。

Schad 等人 [56] 通过运行 CPU 和内存性能、磁盘I/O 以及网络带宽基准测试，研究了 AmazonEC2 [2] 上的性能变化。Feitelson等人 [20] 考察了 NASAAmes一个 128 节点集群上的并行科学工作负载，并展示了作业提交率、系统利用率以及作业特征分布的变化，例如作业类型、运行时间和并行程度。然而，他们没有对作业运行时间变化进行建模或预测。

先前的研究提出了自动化方法，包括用于分析系统故障、性能下降和潜在异常的机器学习技术 [4, 12, 18, 65]。因果推断和依赖关系/图学习也可用于这些应用 [27, 37, 40, 45, 53, 77, 79] 。然而，在我们的工作中，特征维度数量庞大，且特征组之间存在复杂的相关性，这类方法需要人工输入来调整依赖结构，例如添加或删除检测到的依赖关系，并且仍可能存在偏差。在我们的工作中，我们不会手动构建变化原因之间的依赖关系，而是使用 Shapley 分数进行推断。

Huang 等人 [30, 31] 研究了事务型数据库中查询延迟变化的原因。他们使用方差（和协方差）作为量化变化的主要指标，并构建了与调用图对应的方差树，以深入了解方差的影响因素。相比之下，我们使用运行时间 分布 来预测变化，而不使用标量指标，因为后者不足以全面表征或解释变化。此外，我们使用特征重要性和 Shapley 分数来识别对预测的主要贡献因素，从而确定作业的新一次运行可能属于哪种分布。

先前的研究也探索了如何通过并发查询减少性能波动。Cres-cando [71] 提出了一种关系表实现，通过采用仅扫描架构（不使用索引）以及新型协同扫描和更新连接算法等设计选择，将可预测的查询性能置于最优性能之上。CJOIN [10] 引入了一种新的连接算子，并在并发查询之间共享计算和资源，以提升吞吐量和性能稳定性。为我们的规范性分析增添以下能力：

评估计算共享及其他并发查询优化措施的影响，是一个值得未来探索的方向。

为了调优作业性能，贝叶斯优化 [5, 15],梯度下降 [72, 75], 等黑盒优化方法需要多次运行实验。在本研究中，我们提出了一种一次性方法，可以直接确定最佳行动方案。

9 结论

在本研究中，我们通过系统地刻画、建模、预测和解释作业运行时间变化，对 Microsoft Cosmos 上反复运行的生产作业的运行时间变化进行了广泛分析。我们提出的两步方法会为每个作业计算后验似然，将其与预定义的概率分布相关联；该分布的形状取决于以下因素：（1）固有的作业特征；（2）资源分配；（3）提交作业时的集群状态。我们以 >96% 的准确率推断作业运行时间的分布，其表现优于传统回归模型，并能更好地捕捉该分布的长尾。通过使用可解释的机器学习算法，我们分析了变化来源，例如空闲令牌的使用、计算节点上的负载偏斜，以及在不同 SKU 上执行的顶点比例。我们通过调整这些控制变量，量化了性能改进幅度。结合能够通过工作负载再平衡捕捉其对系统利用率影响的模型，我们的技术可以动态优化单个作业的性能。

参考文献

[1] 2022. 石笋。2022年检索自 https://en.wikipedia.org/wiki/Stalagmite

[2] 2022 *2* 15, 2022 2 Amazon. . *Amazon EC*。

[3] 2020 4, 2020 2月检索自 https://aws.amazon.com/aws/ec Amazon.com, Inc. . *Amazon Athena*。7月检索自 https://aws.amazon.com/athena/ [4] Kanishka Bhaduri、Kamalika Das 和 Bryan L Matthews。2011年。检测云基础设施中的异常机器特征。载于*2011 IEEE* 第*11*届数据挖掘国际会议研讨会。IEEE，137–144。[5] Christopher M Bishop 和 Nasser M Nasrabadi。2006年。模式识别与机器学习。第4卷。Springer。[6] Eric Boutin、Paul Brett、Xiaoyu Chen、Jaliya Ekanayake、Tao Guan、Anna Korsun、Zhicheng Yin、Nan Zhang 和 Jingren Zhou。2015年8月12日，1680–1691。Jetscope：云规模下可靠且交互式的分析。*VLDB Endowment* 会议录（），–。[7] Eric Boutin、Jaliya Ekanayake、Wei Lin、Bing Shi、Jingren Zhou、Zhengping Qian、Ming Wu 和 Lidong Zhou。2014年。Apollo：面向云规模计算的可扩展协同调度。载于第*11*届 *USENIX* 操作系统设计与实现研讨会（*OSDI 14*）。285–300。[8] Nicolas Bruno、Sameer Agarwal、Srikanth Kandula、Bing Shi、Ming-Chuan Wu 和 Jingren Zhou。2012年。SCOPE 中的重复作业优化。载于*2012*年 *ACM* 数据管理国际会议（*SIGMOD*）会议录。805–806。[9] Nicolas Bruno、Sapna Jain 和 Jingren Zhou。2013年。持续的云规模查询优化与处理。*VLDB Endowment* 会议录6，11（2013），961–972。[10] George Candea、Neoklis Polyzotis 和 Radek Vingralek。2011年。面向数据分析的可预测性能与高查询并发性。*VLDB Journal*20，2（2011），227–248。[11] Ronnie Chaiken、Bob Jenkins、Per-Åke Larson、Bill Ramsey、Darren Shakib、Simon Weaver 和 Jingren Zhou。2008年。SCOPE：大规模数据集并行处理的简便高效方案。*VLDB Endowment* 会议录1，2（2008），1265–1276。[12] Roshan Chitrakar 和 Chuanhe Huang。2012年。结合 k-medoids 聚类与朴素贝叶斯分类的混合学习方法，用于基于异常的入侵检测。载于*2012*年第*8*届无线通信、网络与移动计算国际会议。IEEE，1–5。[13] Andrew Chung、Subru Krishnan、Konstantinos Karanasos、Carlo Curino 和 Gregory R Ganger。2020年。发掘作业间依赖关系以改进集群调度。载于第*14*届{*USENIX*} 操作系统设计与实现研讨会（{*OSDI*} *20*）。1205–1223。[14] Andrew Chung、Subru Krishnan、Konstantinos Karanasos、Carlo Curino 和 Gregory R. Ganger。2020年。发掘作业间依赖关系以改进集群调度。载于第*14*届 *USENIX* 操作系统设计与实现研讨会（*OSDI 20*）。1205–1223。

[15] Carlo Curino、Neha Godwal、Brian Kroth、Sergiy Kuryata、Greg Lapinski、Siqi Liu、Slava Oks、Olga Poppe、Adam Smiechowski、Ed Thayer 等。2020年。MLOS：自动化软件性能工程基础设施。载于第四届端到端机器学习数据管理国际研讨会会议录。1–5。[16] Carlo Curino、Subru Krishnan、Konstantinos Karanasos、Sriram Rao、Giovanni M Fumarola、Botong Huang、Kishore Chaliparambil、Arun Suresh、Young Chen、Solom Heddaya 等。2019年。Hydra：面向数据中心规模分析的联邦式资源管理器。载于第16届 USENIX 网络系统设计与实现研讨会（NSDI 19）。177–192。[17] Francesco Diaz 和 Roberto Freato。2018年。Azure Data Lake Store 和 Azure Data Lake Analytics。载于《使用 Microsoft Azure 进行云数据设计、编排与管理》。Springer，327–392。[18] Songyun Duan、Shivnath Babu 和 Kamesh Munagala。2009年。Fa：自动化故障诊断系统。载于2009年 IEEE 第25届数据工程国际会议。IEEE，1012–1023。[19] Zhiwei Fan、Rathijit Sen、Paraschos Koutris 和 Aws Albarghouthi。2020年。通过机器学习自动调优查询并行度。载于第三届利用人工智能技术进行数据管理国际研讨会会议录。文章2，共4页。[20] Dror G Feitelson 和 Bill Nitzberg。1995年。NASA Ames iPSC/860 上生产环境并行科学工作负载的作业特征。载于并行处理作业调度策略研讨会。Springer，337–360。[21] Andrew D Ferguson、Peter Bodik、Srikanth Kandula、Eric Boutin 和 Rodrigo Fonseca。2012年。Jockey：保障数据并行集群中的作业延迟。载于第7届 ACM 欧洲计算机系统会议会议录。99–112。[22] Hans Fischer。2011年。中心极限定理史：从古典概率论到现代概率论。Springer。[23] Eric Gaussier、David Glesser、Valentin Reis 和 Denis Trystram。2015年。利用机器学习预测运行时间以改进回填。载于 SC'15：高性能计算、网络、存储与分析国际会议会议录。IEEE，1–10。[24] Ramanathan Gnanadesikan 和 Martin B Wilk。1968年。用于数据分析的概率图绘制方法。Biometrika 55，1（1968），1–17。[25] Gregg，Brendan。2022年。CPU 利用率是错误的。2022年10月4日检索自 https://www.brendangregg.com/blog/2017- 05-09/cpu-utilization-is-wrong.html [26] Jian Guo、Akihiro Nomura、Ryan Barton、Haoyu Zhang 和 Satoshi Matsuoka。2018年。高性能计算系统中作业运行时间低估的机器学习预测。载于亚洲超级计算前沿会议。Springer，Cham，179–198。[27] David Heckerman 和 John S Breese。1996年。使用贝叶斯网络进行概率评估与推断时的因果独立性。IEEE《系统、人类与控制论汇刊：A辑：系统与人类》26，6（1996），826–831。[28] Herodotos Herodotou、Fei Dong 和 Shivnath Babu。2011年。没有一种集群规模适用于所有场景：面向数据密集型分析的自动集群规模配置（SOCC '11）。计算机协会，纽约，美国，文章18，共14页。https://doi.org/ 10 1145 2038916 2038934 。/。Zhiyao Hu、Dongsheng Li、Dongxiang Zhang 和 Yixin Chen。[29] 2020年。ReLoca：利用深度学习优化数据并行作业的资源分配。载于 IEEE INFOCOM 2020 - IEEE 计算机通信会议。1163–1171。[30] Jiamin Huang、Barzan Mozafari、Grant Schoenebeck 和 Thomas Wenisch。2016年。识别事务延迟方差的主要来源：迈向更可预测的数据库。arXiv 预印本 arXiv:1602.01871（2016）。[31] Jiamin Huang、Barzan Mozafari、Grant Schoenebeck 和 Thomas F Wenisch。2017年。自上而下实现数据库系统性能可预测性的方法。载于2017年 ACM 数据管理国际会议会议录。745–758。[32] Alekh Jindal、Hiren Patel、Abhishek Roy、Shi Qiao、Zhicheng Yin、Rathijit Sen 和 Subru Krishnan。2019年。Peregrine：云查询引擎的工作负载优化。载于 ACM 云计算研讨会会议录。416–427。[33] Sangeetha Abdu Jyothi、Carlo Curino、Ishai Menache、Shravan Matthur Narayanamurthy、Alexey Tumanov、Jonathan Yaniv、Ruslan Mavlyutov、Íñigo Goiri、Subru Krishnan、Janardhan Kulkarni 等。2016年。Morpheus：迈向企业集群的自动化服务级别目标。载于第12届 USENIX 操作系统设计与实现研讨会（OSDI 16）。117–134。[34] Konstantinos Karanasos、Sriram Rao、Carlo Curino、Chris Douglas、Kishore Chaliparambil、Giovanni Matteo Fumarola、Solom Heddaya、Raghu Ramakrishnan 和 Sarvesh Sakalanaga。2015年。Mercury：大型共享集群中的混合式集中与分布式调度。载于2015年 USENIX 年度技术会议（USENIX ATC 15）。485–497。[35] Marvin Karson。1968年。《应用统计方法手册》。第一卷：计算技术、描述性方法与统计推断。第二卷：调查与实验规划。IM Chakravarti、RG Laha 和 J. Roy，纽约，John Wiley；1967年，9.00美元。[36] Michael Kuchnik、Jun Woo Park、Chuck Cranor、Elisabeth Moore、Nathan DeBardeleben 和 George Amvrosiadis。2019年。这就是为什么机器学习驱动的集群调度仍然普遍不实用。技术报告（2019年）。

[37] Po-Ling Loh 和 Peter Bühlmann。2014。通过逆协方差估计进行线性因果网络的高维学习。《机器学习研究期刊》15，1（2014），3065–3105。[38] Gilles Louppe、Louis Wehenkel、Antonio Sutera 和 Pierre Geurts。2013。理解随机树森林中的变量重要性。《神经信息处理系统进展》26（2013）。[39] Scott M Lundberg 和 Su-In Lee。2017。解释模型预测的统一方法。《神经信息处理系统进展》30（2017）。[40] Panagiotis Mandros、Mario Boley 和 Jilles Vreeken。2017。发现可靠的近似函数依赖。载于《第 23 届 ACM SIGKDD 知识发现与数据挖掘国际会议论文集》。355–363。[41] Microsoft Corporation。2022。LGBMClassifier。检索于 2022 年 2 月 4 日，网址：https://lightgbm.readthedocs.io/en/latest/ pythonapi/lightgbm.LGBMClassifier.html [42] Christoph Molnar。2020。可解释的机器学习。Lulu.com。[43] Mina Naghshnejad 和 Mukesh Singhal。2018。云环境中用于改善 HPC 应用程序延迟的自适应在线运行时间预测。载于《2018 年 IEEE 第 11 届云计算国际会议（CLOUD）》。IEEE，762–769。[44] Frank Nielsen。2016。面向数据科学的 MPI 高性能计算导论。Springer。[45] Harsha Nori、Samuel Jenkins、Paul Koch 和 Rich Caruana。2019。InterpretML：用于机器学习可解释性的统一框架。arXiv 预印本 arXiv:1909.09223（2019）。[46] Ilia Pietri、Gideon Juve、Ewa Deelman 和 Rizos Sakellariou。2014。用于估算云上科学工作流执行时间的性能模型。载于《2014 年第 9 届支持大规模科学的工作流研讨会》。IEEE，11–19。[47] Anish Pimpley 、 Shuo Li 、 Rathijit Sen 、 Soundararajan Srinivasan 和 Alekh Jindal。2022。迈向大数据分析的最优资源分配。载于《第 25 届数据库技术扩展国际会议（EDBT）》。338–350。[48] Anish Pimpley、Shuo Li、Anubha Srivastava、Vishal Rohra、Yi Zhu、Soundararajan Srinivasan、Alekh Jindal、Hiren Patel、Shi Qiao 和 Rathijit Sen。2021。无服务器查询的最优资源分配。arXiv 预印本 arXiv:2107.08594（2021）。[49] Conor Power、Hiren Patel、Alekh Jindal、Jyoti Leeka、Bob Jenkins、Michael Rys、Ed Triou、Dexin Zhu、Lucky Katahanas、Chakrapani Bhat Talapady 等。2021。Microsoft Cosmos 大数据平台：十余年的进展与未来十年的展望。《VLDB Endowment 会议录》14，12（2021），3148–3161。[50] Shi Qiao、Adrian Nicoara、Jin Sun、Marc Friedman、Hiren Patel 和 Jaliya Ekanayake。2019。Hyper dimension shuffle：Scope 中 PB 级数据的高效重新分区。《VLDB Endowment 会议录》12，10（2019），1113–1125。[51] Kaushik Rajan、Dharmesh Kakadia、Carlo Curino 和 Subru Krishnan。2016。PerfOrator：用于资源优化的精确性能模型。载于《第七届 ACM 云计算研讨会论文集》。415–427。[52] Raghu Ramakrishnan、Baskar Sridharan、John R Douceur、Pavan Kasturi、Balaji Krishnamachari-Sampath、Karthick Krishnamoorthy、Peng Li、Mitica Manu、Spiro Michaylov、Rogério Ramos 等。2017。Azure Data Lake Store：面向大数据分析的超大规模分布式文件服务。载于《2017 年 ACM 数据管理国际会议论文集》。51–63。[53] Garvesh Raskutti 和 Caroline Uhler。2018。基于最稀疏排列学习有向无环图模型。《Stat》7，1（2018），e183。[54] Lior Rokach 和 Oded Maimon。2005。聚类方法。载于《数据挖掘与知识发现手册》。Springer，321–352。[55] Aurobindo Sarkar 和 Amit Shah。2018。学习 AWS：使用 AWS 云组件设计、构建和部署响应式应用程序。Packt Publishing Ltd。[56] Jörg Schad、Jens Dittrich 和 Jorge-Arnulfo Quiané-Ruiz。2010。云中的运行时间测量：观察、分析与降低方差。PVLDB 3, 1 2 2010 460 471 10 14778 1920841 1920902 –（），–。[57] 2022 4, 2022 https://doi.org/ . / . Scikit-Learn。 . EnsembledClassifier。检索于 2 月，网址：https://scikit-learn.org/stable/modules/ensemble. html[58] Scikit-Learn。2022。GaussianNB。检索于 2022 年 2 月 4 日，网址：https://scikit-learn.org/stable/modules/generated/sklearn. naive_bayes.GaussianNB.html [59] Scikit-Learn。2022。GradientBoostingClassifier。检索于 2022 年 2 月 4 日，网址：https://scikit-learn.org/stable/modules/ generated/sklearn.ensemble.GradientBoostingClassifier.html [60] Scikit-Learn。2022。GradientBoostingClassifier。检索于 2022 年 2 月 4 日，网址：https://scikit-learn.org/stable/modules/ generated/sklearn.ensemble.GradientBoostingClassifier.html [61] Scikit-Learn。2022。RandomForestClassifier。检索于 2022 年 2 月 4 日，网址：https://scikit-learn.org/stable/modules/ generated/sklearn.ensemble.RandomForestClassifier.html [62] David Sculley。2010。Web 规模的 k-means 聚类。载于《第 19 届万维网国际会议论文集》。1177–1178。[63] Rathijit Sen、Alekh Jindal、Hiren Patel 和 Shi Qiao。2020。AutoToken：预测 Microsoft 大数据分析的峰值并行性。PVLDB 13，12（2020），3326–3339。

[64] Rathijit Sen、Abhishek Roy、Alekh Jindal、Rui Fang、Jeff Zheng、Xiaolei Liu 和 Ruiping Li。2021。AutoExecutor：面向 Spark SQL 查询的预测性并行。VLDB Endowment 论文集 14，12（2021），2855–2858。[65] Liqun Shao、Yiwen Zhu、Siqi Liu、Abhiram Eswaran、Kristin Lieber、Janhavi Mahajan、Minsoo Thigpen、Sudhir Darbha、Subru Krishnan、Soundar Srinivasan 等。2019。Griffon：在基于云的平台中利用无标签数据推理作业异常。载于 ACM 云计算研讨会论文集。441–452。[66] S Shapley Ll. 1953。n 人博弈的一个值。《博弈论贡献》第二卷，数学研究年鉴 28（1953）。[67] Ozan Sonmez、Nezih Yigitbasi、Alexandru Iosup 和 Dick Epema。2009。对网格中的作业运行时间和队列等待时间预测进行基于轨迹的评估。载于第 18 届 ACM 高性能分布式计算国际研讨会论文集。111–120。[68] Ashish Thusoo、Joydeep Sen Sarma、Namit Jain、Zheng Shao、Prasad Chakka、Suresh Anthony、Hao Liu、Pete Wyckoff 和 Raghotham Murthy。2009。Hive：基于 MapReduce 框架的数据仓储解决方案。VLDB Endowment 论文集 2，2（2009），1626–1629。[69] Jordan Tigani 和 Siddartha Naidu。2014。Google BigQuery Analytics。John Wiley & Sons。[70] Dan Tsafrir、Yoav Etsion 和 Dror G Feitelson。2007。使用系统生成的预测而非用户运行时间估计进行回填。IEEE并行与分布式系统汇刊 18，6（2007），789–803。[71] Philipp Unterbrunner、Georgios Giannikis、Gustavo Alonso、Dietmar Fauser 和 Donald Kossmann。2009。为不可预测的工作负载提供可预测的性能。VLDB Endowment 论文集 2，1（2009），706–717。[72] Dana Van Aken、Andrew Pavlo、Geoffrey J Gordon 和 Bohan Zhang。2017。通过大规模机器学习实现自动数据库管理系统调优。载于 2017 年 ACM 数据管理国际会议论文集。1009–1024。[73] Vinod Kumar Vavilapalli、Arun C Murthy、Chris Douglas、Sharad Agarwal、Mahadev Konar、Robert Evans、Thomas Graves、Jason Lowe、Hitesh Shah、Siddharth Seth 等。2013。Apache Hadoop YARN：又一个资源协商器。载于第 4 届年度 ACM 云计算研讨会论文集。ACM，5。[74] Matei Zaharia、Mosharaf Chowdhury、Michael J Franklin、Scott Shenker、Ion Stoica 等。2010。Spark：利用工作集进行集群计算。HotCloud 10，10-10（2010），95。[75] Ji Zhang、Yu Liu、Ke Zhou、Guoliang Li、Zhili Xiao、Bin Cheng、Jiashu Xing、Yangtao Wang、Tianheng Cheng、Li Liu 等。2019。基于深度强化学习的端到端自动云数据库调优系统。载于 2019 年 ACM 数据管理国际会议论文集。415–432。[76] Jiaxing Zhang、Hucheng Zhou、Rishan Chen、Xuepeng Fan、Zhenyu Guo、Haoxiang Lin、Jack Y Li、Wei Lin、Jingren Zhou 和 Lidong Zhou。2012。通过理解用户定义函数优化数据并行计算中的数据重分区。载于第 9 届 USENIX 网络系统设计与实现研讨会（NSDI 12）论文集。295–308。[77] Yunjia Zhang、Zhihan Guo 和 Theodoros Rekatsinas。2020。从统计学角度发现噪声数据中的函数依赖。载于 2020 年 ACM SIGMOD 数据管理国际会议论文集。861–876。[78] Hai-Tao Zheng 和 Haiyang Zhang。2015。基于保留特征的被动攻击算法进行在线特征选择。载于亚太网络会议。Springer，707–719。[79] Pengfei Zheng 和 Benjamin C Lee。2018。Hound：用于数据中心规模拖尾任务诊断的因果学习。ACM计算系统测量与分析 2，1（2018），1–36。[80] Jingren Zhou、Nicolas Bruno 和 Wei Lin。2012。大规模分布式计算的高级分区技术。载于 2012 年 ACM SIGMOD 数据管理国际会议论文集。13–24。[81] Jingren Zhou、Per-Ake Larson 和 Ronnie Chaiken。2010。将分区和并行计划纳入 SCOPE 优化器。载于 2010 年 IEEE 第 26 届数据工程国际会议（ICDE 2010）。IEEE，1060–1071。[82] Yiwen Zhu、Matteo Interlandi、Abhishek Roy、Krishnadhan Das、Hiren Patel、Malay Bag、Hitesh Sharma 和 Alekh Jindal。2021。Phoebe：一种基于学习的检查点优化器。Proc. VLDB Endow. 14，11（2021 年 7 月），2505–2518。https://doi.org/10.14778/3476249.3476298[83] Yiwen Zhu、Subru Krishnan、Konstantinos Karanasos、Isha Tarte、Conor Power、Abhishek Modi、Manoj Kumar、Deli Zhang、Kartheek Muthyala、Nick Jurgens 等。2021。KEA：调优百亿亿字节级数据基础设施。载于 2021 年 ACM 数据管理国际会议论文集。2667–2680。[84] Dmitry Zotkin 和 Peter J Keleher。1999。回填调度器中的作业长度估计与性能。载于《论文集：第八届高性能分布式计算国际研讨会》（目录编号 99TH8469）。IEEE，236–243。[85] Salah Zrigui、Raphael Y de Camargo、Arnaud Legrand 和 Denis Trystram。2022。利用在线作业运行时间分类提升批处理调度器的性能。J. Parallel and Distrib. Comput. 164（2022），83–95。

收到：2022年7月；修订：2022年10月；接受：2022年11月