

# 一种考虑局部性的作业执行时间预测算法

闫家晨, 肖永浩, 王凌锋, 熊敏

(中国工程物理研究院计算机应用研究所, 四川 绵阳 621900)

**摘要:** 针对高性能计算系统中作业执行时间预测不能充分利用作业的局部性、预测精度低的问题, 提出一种考虑局部性的作业执行时间预测算法(JRPL). 该算法综合利用作业日志数据的总体特征与局部特征, 通过投票机制结合机器学习预测和基于局部性的时序预测, 提高了预测准确性. 实验结果表明, 在 Unliu Gaia 和 PIK IPLEX 等实际调度日志数据集上, JRPL 算法在平均绝对误差、平均预测精度和命中率 3 个指标上均优于或不劣于作为基底的机器学习算法. 该研究结果为高性能计算系统中作业调度提供了预测模型的改良方法, 有助于进行更准确的执行时间预测, 提高系统资源利用率, 降低计算成本.

**关键词:** 高性能计算系统; 作业执行时间预测; 机器学习; 局部性; 预测精度

**中图分类号:** TP39 **文献标志码:** A **文章编号:** 1671-5489(2025)06-1685-09

## A Job Runtime Prediction Algorithm Considering Locality

YAN Jiachen, XIAO Yonghao, WANG Lingfeng, XIONG Min

(*Institute of Computer Application, China Academy of Engineering Physics,  
Mianyang 621900, Sichuan Province, China*)

**Abstract:** Aiming at the problem of underutilization of job locality and low prediction accuracy in high-performance computing systems, we proposed a job runtime prediction algorithm considering locality. The algorithm comprehensively utilized the global and local features of job log data, and improved prediction accuracy through a voting mechanism that combined with machine learning prediction and locality-based time-series prediction. Experimental results show that on actual scheduling log datasets such as Unliu Gaia and PIK IPLEX, the JRPL algorithm outperforms or is not inferior to the machine learning algorithms as the baseline in all three metrics: average absolute error, average prediction accuracy, and hit rate. This research result provides an improved prediction model for job scheduling in high-performance computing systems, which helps to make more accurate execution time forecasting, improve system resource utilization, and reduce computational costs.

**Keywords:** high-performance computing system; job runtime prediction; machine learning; localized; prediction accuracy

高性能计算在工程模拟、人工智能、工业制造等领域应用广泛. 在高性能计算系统中, 常用的调度算法<sup>[1]</sup>有先来先服务(first come first service, FCFS)和短作业优先(short job first, SJF)等. Tang 等<sup>[2]</sup>提出高效的调度策略需要对作业执行时间有一个大概估计, 估计越精确, 调度越高效<sup>[3]</sup>. 以 SJF

收稿日期: 2025-04-16.

**第一作者简介:** 闫家晨(1999—), 男, 汉族, 硕士研究生, 从事高性能计算的研究, E-mail: 1713217184@qq.com. **通信作者简介:** 熊敏(1980—), 女, 汉族, 博士, 副研究员, 从事高性能计算的研究, E-mail: katheleen\_1980@163.com.

**基金项目:** 国家自然科学基金(批准号: 62302468).

为例, 如果没有对作业执行时间的简单估计和预测, 则无法识别短作业. 在多个节点的负载平衡问题上, 作业的估计执行时间也同样是一个重要信息. 一般调度系统中作业预估执行时间由用户提供. 文献[4-5]研究表明, 用户提供的执行时间通常是一个不准确的估计. Cirne 等<sup>[6]</sup>研究表明, 50%~60% 的作业实际运行时间不到系统用户给出估计时间的 20%.

另一种获得作业预测时间的方法是由系统自动生成预估时间. 文献[7-9]通过系统分析历史数据进行了执行时间预测. 文献[10-12]通过对用户模式进行识别聚类后进行作业时间预测. 文献[13-14]则通过组合多种预测算法, 对面向网格和集群计算的任务进行了针对性的预测. Tsafirir 等<sup>[11]</sup>研究表明, 由系统根据执行历史预测任务的执行时间比用户提供的估计准确得多, 同时提出了以同一个用户最近两次提交作业的实际执行时间的平均值作为下一次作业执行时间的预测结果, 该方法很好地利用了高性能计算作业潜在的局部性, 且操作简单. 这里的局部性是指高性能计算作业执行时间的常见规律: 作业执行时间通常与历史执行记录有相关性, 特别是与最近提交的、相似的作业相关. 文献[8]则通过结合用户提交的信息进行进一步分析进行预测, 以获得更精确的作业执行时间. Nissimov<sup>[15]</sup>利用日志中作业执行时间的局部特征, 即时序相关性给出了一个隐 Markov 模型预测作业执行时间. 文献[11]和文献[15]的方法都从作业的时序相关性出发试图对作业执行时间进行预测, 但都侧重于作业的局部特征而忽视了作业的全局特征, 因而导致最终预测结果不理想.

Amaris 等<sup>[16]</sup>使用机器学习方法预测高性能计算系统中程序的执行时间, 其使用线性回归、支持向量机和随机森林方法对来自 9 个 GPU 上执行的 9 个不同应用程序的执行时间进行了预测, 并取得了较好结果. Gao 等<sup>[17]</sup>创造了一种新的基于机器学习的工具 DNNPerf, 用于预测基于图神经网络的深度学习模型的训练时间, 最终能达到总体平均相对误差 7.4% 的优秀结果. 实践证明机器学习模型能很好地捕捉复杂标签中的潜在关系, 在信息丰富、程序类别单一时能取得良好的预测效果, 但基于机器学习的预测方法对数据局部特性的识别存在不足, 如文献[18]提出随机森林等算法缺乏对局部性的利用, 因此本文提出一种作业执行时间预测算法 JRPL(job runtime prediction based on locality), 其在应用传统机器学习算法对作业进行建模预测的同时综合考虑作业的局部性, 从而得到一个相比传统机器学习算法更好的预测结果.

### 1 作业执行时间预测算法 JRPL

JRPL 算法在传统机器学习算法的基础上, 通过综合考虑历史日志数据的局部性, 最终通过投票机制生成作业执行时间预测结果, 其算法流程如图 1 所示. 首先是机器学习预测, 即应用随机森林、支持向量机、线性回归、决策树等传统机器学习算法, 对历史作业数据进行训练, 得到一个预测结果. 其次考虑数据局部性预测, 即将作业按照特征进行分组, 对每组 S 单独维护一个窗口队列 S' 及其预测参数, 然后根据组内信息得到一个基于局部信息的预测结果. 最后投票选择, 即利用投票策略在基于局部特征预测结果与基于机器学习模型预测结果中做选择, 将投票结果作为最终的作业执行时间预测结果.

在当次预测结束后, 相关参数将会根据当前预测结果数据进行动态调整. 其中, 机器学习算法可解决一些很难用局部特征进行预测的作业预测问题, 如某用户首次提交作业等. 该预测可在分组前, 也可以在分组后. 一般地, 对于如随机森林算法或决策树算法等有较强分类能力的算法, 可对全

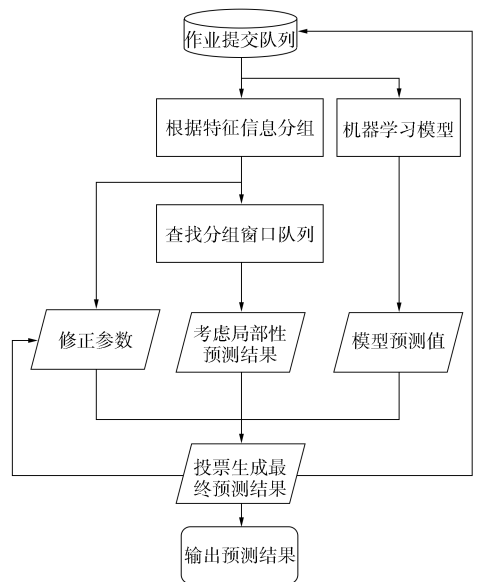


图 1 JRPL 算法流程

Fig. 1 Flow chart of JRPL algorithm

部数据统一建模加快 JRPL 算法的执行速度,否则在分组后对每组单独建立机器学习模型进行预测,提高预测准确度.

将 JRPL 算法作用于高性能计算系统作业日志时,在利用作业日志局部特征进行预测的同时对一些很难根据局部特征预测的作业使用传统机器学习算法的预测结果作为补充,通常可得到比传统机器学习算法更好的结果.而在一些数据标签丰富、机器学习算法发挥更出色的数据集中, JRPL 的投票机制也能确保其不会得到比作为补充的机器学习算法更差的结果.

### 1.1 作业执行时间预测算法

考虑局部信息,提出基于局部特征的预测算法,该预测算法分以下 3 个步骤:

- 1) 基于特征信息的作业分组,即利用用户、队列等相似性特征信息对作业进行分组;
- 2) 动态窗口序列构建,即对分组建立窗口队列,并在作业过多时引入滑动窗口;
- 3) 获取预测时间,即根据滑动窗口中的作业实际执行时间获得当前作业的预测时间.

#### 1.1.1 基于特征信息的分组

作业的某些特征信息有很强的标签作用,因此本文可选择一些这样的特征信息对作业进行分组.当两个作业的这些特征完全相同时,认为它们在同一组  $S$  中,在同一组中的作业共用一套窗口队列  $S'$  和相关参数,将  $S'$  中作业数记为  $|S'|$ .文献[19]提出在预测作业执行时间过程中,用户名是判断作业相似性的重要属性,因此本文实验中选取用户名(user\_id)这一特征进行分组,即按用户、队列等特征分组后,同一组内作业的执行时间模式相似.同一用户提交的作业,其执行时间通常有相似的模式,这种分组后的相似性是一种空间上的局部性.在一些特殊数据集中,组名(group\_id)和队列名(queue\_id)也可作为分组的依据.

#### 1.1.2 基于长时等待建立动态窗口队列

在同一组  $S$  中,如果新提交作业的提交时间与上一个作业的提交时间有较长间隔,则称该分组发生了长时等待.在两个长时等待事件间隔中,作业数据提交时间间隔较短,常具有更好的局部性.即同一用户(或分组)连续提交的作业在时间序列上有执行时间的相关性.当新作业提交时间与上一个作业提交时间间隔较短时(即没有长时等待),这些作业的执行时间通常相似,可利用这种时序相关性进行预测.窗口队列的建立和销毁随长时等待事件的发生而发生,因此,窗口队列大小通常是不固定的.

为判断是否发生长时等待,本文实时统计分组中每两个作业提交时间间隔的平均值.如果某个作业提交与上次提交的时间大于该平均间隔时间的某倍数时,则可断定发生了长时等待事件,此时应删除已有窗口队列并重新建立一个新的窗口队列.特别地,发生了长时等待事件作业的时间间隔不纳入时间间隔统计.本文实验中该倍数设为 10.

在绝大多数分组中,与长时等待事件绑定的分组建立-销毁机制可很好地动态维护一个恰当长度的窗口队列.但在一些极端情况下,也可能出现分组很大的情况,为避免窗口过大而使其维护耗时耗力,引入参数  $K$ ,当窗口队列内作业个数超过  $K$  时,窗口变为滑动窗口,即只关注最近  $K$  个作业的执行情况.经数值实验验证,参数  $K$  值超过某个阈值后其大小对预测准确度无明显影响.本文实验中  $K=20$ .合适的  $K$  值根据具体数据集有变化,经过多次实验对比,本文选取的  $K$  值较适合本文选取的实验数据集.

#### 1.1.3 基于误差检查和加权平均的作业执行时间预测

在建立窗口的长时等待机制基础上,为提高对作业数多、无明显等待事件分组的预测准确度,在分组内加入误差检查机制.误差检查机制是指当某次作业的考虑局部性预测结果与真实值的相对误差达到比率阈值  $p$  以上时,在新预测中将不使用窗口中该作业及其之前的作业数据.

在对超算作业队列中的作业进行分组后,在每组中建立一个长度不固定但有最大长度的动态窗口队列  $S'$ .得到窗口队列  $S'$  后,可使用这  $|S'|$  个作业的数值特征及它们的真实执行时间预测新作业的执行时间. $S'$  中的作业有相同的关键特征,且相邻作业提交时间间隔被长时等待机制控制,所以相比整个超算作业队列有良好的局部性特征,即相比超算作业队列, $S'$  中距离较近的作业其执行时间相似程度更高.实验结果表明,针对局部性特征良好、大小有限的窗口队列  $S'$ ,加权平均算法具有预测速度

快、资源占用小、对作业的局部特征利用率高的优点。所以本文选用其计算考虑局部信息的作业执行时间预测结果  $Pred_2(\text{Job\_new})$ ，计算公式为

$$Pred_2(\text{Job\_new}) = \sum_{0 < n < m} \frac{T_n}{2^n} + \frac{T_m}{2^{m-1}}, \tag{1}$$

其中  $T_n$  为当前窗口中距离当前需预测作业  $n$  个位置前的作业执行时间的真实值， $m$  为当前窗口中最早一次作业到本次作业前的作业总数。

误差检查和加权平均的共同作用，构成使用分组动态窗口中历史作业数据预测新作业的考虑局部性预测算法。

### 1.2 投票选择

本文分别得到两个作业执行时间预测结果：基于机器学习算法的预测结果  $Pred_1(\text{Job\_new})$  和考虑局部性的预测结果  $Pred_2(\text{Job\_new})$ 。其中， $Pred_1(\text{Job\_new})$  是使用如随机森林、决策树算法等机器学习算法模型得到的预测结果。为得到最终的作业执行时间预测结果，本文引入投票机制，该机制先基于动态窗口队列中历史作业的两种预测结果准确度进行投票，然后根据最终投票结果决定选取哪一个预测结果作为最终的预测结果。投票方法如下：首先，对某个作业投票结果  $K$ ，其计算公式为

$$K = \begin{cases} 0, & PA_1 < PA_2, \\ 1, & PA_1 \geq PA_2, \end{cases} \tag{2}$$

其中  $PA_1$  为当前作业传统机器学习预测的预测精度， $PA_2$  为当前作业考虑局部性预测的预测精度。其次，计算投票权重  $\alpha$  值，其初始值设为 0，计算公式为

$$\alpha = \sum_{0 < n \leq M} \frac{K_n}{2^n}, \tag{3}$$

其中  $M$  为动态窗口中的历史作业总数， $K_n$  为距离当前作业  $n$  次前的作业投票结果。投票权重  $\alpha$  的值在  $0 \sim 1$  之间。最后选定阈值  $\alpha_0$ ，若  $\alpha$  大于阈值  $\alpha_0$ ，则选择考虑局部性的预测结果作为最终预测结果，否则选择机器学习算法的预测结果作为最终结果。本文选取  $\alpha_0 = 0.6$ ，约等价于要求在最近一次预测以及距离为 2,3 的两次预测中的其中一次，此时考虑局部性预测结果更准确，于是当前预测选择考虑局部性预测结果。调整  $\alpha$  值可使要求适当放宽或收紧。期间， $\alpha$  会在每次触发长时等待或误差检查未通过时重置为初始值 0。当遇到难以只靠作业局部特征进行预测的作业时，应使用机器学习算法的预测结果作为最终结果。同时，可将机器学习算法的整体预测平均预测精度作为比率阈值  $p$ 。当考虑局部性的预测结果较机器学习算法预测结果无明显优势时，会频繁触发误差检查使  $\alpha$  维持在一个较低水平，从而确保了 JRPL 算法的预测精度下限不会低于补充的传统机器学习算法的预测结果。

## 2 数值实验结果与分析

本文首次数值实验选取的传统机器学习算法是随机森林算法，即首先应用随机森林算法及对应扩展的 JRPL 算法在 Unliu Gaia 和 PIK IPLEX 两个真实的作业日志数据进行实验，得到实验结果后进行分析；其次，进一步对决策树算法和 SVR 算法以及分别对应扩展的 JRPL 算法在数据集 Unliu Gaia 和 PIK IPLEX 上进行数值实验，将实验结果进行分析后，得到相一致的结论；最后，将 3 种机器学习算法及其对应的 JRPL 算法在 RICC 等 5 个真实的作业日志数据上进行数值实验，以进一步验证相关结论。

### 2.1 数据集

本文实验选择文献[20]中提供的高性能计算系统的真实日志数据集，数据集信息列于表 1。

表 1 实验使用的高性能计算日志

Table 1 High-performance computational logs used in experiments

日志名	提交时间	作业数量/个
Unliu Gaia	2014-05—2014-08	51 987
PIK IPLEX	2009-04—2012-07	742 964(取前 100 000)

表1中:Unliu Gaia是德国卢森堡大学的Gaia集群在2014年5月到8月的作业日志,Gaia集群最初于2011年发布,现在是一个经过多次升级的异构集群,目前有151个节点,2004个核,其中20个节点配备了NVidia Tesla级GPU加速器,主要供处理大数据问题的生物学家和进行物理模拟的工程人员使用;PIK IPLEX日志包含来自德国波茨坦气候影响研究所(PIK)的320节点IBM iDataPlex集群超过3年的日志记录,IBM iDataPlex集群共有2560个核,共享32GB内存。

## 2.2 预测评价准则

本文采用平均绝对误差(mean absolute error, MAE)和平均预测精度(average predictive accuracy, APA)两个评价指标,并考虑命中率(hit rate, HR)。

### 2.2.1 平均绝对误差

平均绝对误差是一个衡量预测模型绝对误差的指标,它表示预测值与实际值之间差的绝对值的平均值,其值越小表明预测效果越好。在执行时间预测时,长作业的预测准确度对平均绝对误差的影响显著高于短作业,其计算公式为

$$MAE = \frac{1}{N} \sum_{0 < i \leq N} |P_i - R_i|, \quad (4)$$

其中 $P_i$ 和 $R_i$ 分别为第 $i$ 个作业的预测执行时间和真实执行时间, $N$ 为被预测作业的总数量。

### 2.2.2 平均预测精度

平均预测精度是一个评判相对误差的指标,它是所有作业预测精度的平均值。对 $N$ 个作业平均预测精度定义为

$$APA = \frac{1}{N} \sum_{0 < i \leq N} PA_i, \quad (5)$$

其中 $PA_i$ 是第 $i$ 个作业的预测精度,其计算公式为

$$PA_i = \begin{cases} P_i/R_i, & P_i < R_i, \\ R_i/P_i, & P_i \geq R_i, \end{cases} \quad (6)$$

$P_i$ 和 $R_i$ 分别为第 $i$ 个作业的预测执行时间和真实执行时间。APA在 $[0,1]$ 内取值,其值越接近1说明预测效果越好。作为相对误差的评判指标,APA对长作业和短作业同等对待,是对MAE的有效补充。

### 2.2.3 命中率

命中率表示预测执行时间与实际执行时间的偏差在 $\pm 5\%$ 内作业数占作业总数的比率,记为HR。特别地,对真实执行时间在1min内或5000h以上的特殊作业,只要预测时间同样在1min内或5000h以上即视为命中。命中率是一个相对误差的评判指标,它去除了极端情况干扰,并重点关注预测精准命中的概率,对实际工作有更强的参考性。命中率的计算公式为

$$HR = \frac{1}{N} \sum_{0 < i \leq N} H_i, \quad (7)$$

其中 $H_i$ 为在第 $i$ 个作业的命中率,当满足上述命中条件时取值为1,否则为0。HR在 $[0,1]$ 内取值,其值越接近1说明预测效果越好。

## 2.3 实验步骤

在进行实验前,首先删除日志中缺失真实执行时间的作业和全部空置作业,为减少异常数据的影响,删除日志中运行时间小于180s的作业数据和运行时间为最高的2%的数据。然后使用不同算法针对数据集进行建模,训练模型并预测作业执行时间。最后计算3种算法及其对应扩展的JRPL算法对7个数据集预测的平均绝对误差和平均预测精度,根据预测结果对算法进行评价。

实验中使用的随机森林算法、决策树算法和SVR算法由Python的机器学习算法库scikit-learn实现。训练集和预测集的分组方式列于表2。实验基于真实高性能计算中心作业日志数据集进行,但并非在实际应用环境中进行部署。本文忽略作业执行延时时间,假设作业执行环境为理想情况,即能在作业提交后立即得到相关日志数据。

表 2 训练集和测试集的分组方式

Table 2 Grouping method for training sets and test sets

日志名	在数据集中的位置	日志名	在数据集中的位置
训练集	0~60	测试集	60~100

对于 Unliu Gaia,PIK IPLEX,RICC 等 7 个日志数据, 都有 18 个标签, 本文根据数据完整性及重要性进行筛选, 最终选取其中的作业编号、请求处理器数量、请求时间、请求的内存和用户 ID 等 5 个标签进行机器学习模型训练.

### 2.4 实验结果分析

随机森林算法和对应的 JRPL 算法在两个数据集 Unliu Gaia 和 PIK IPLEX 上的实验结果如图 2 所示. 由图 2 可见, JRPL 算法在两个数据集上的 MAE,APA,HR 相比随机森林算法都有同等或更高水平.

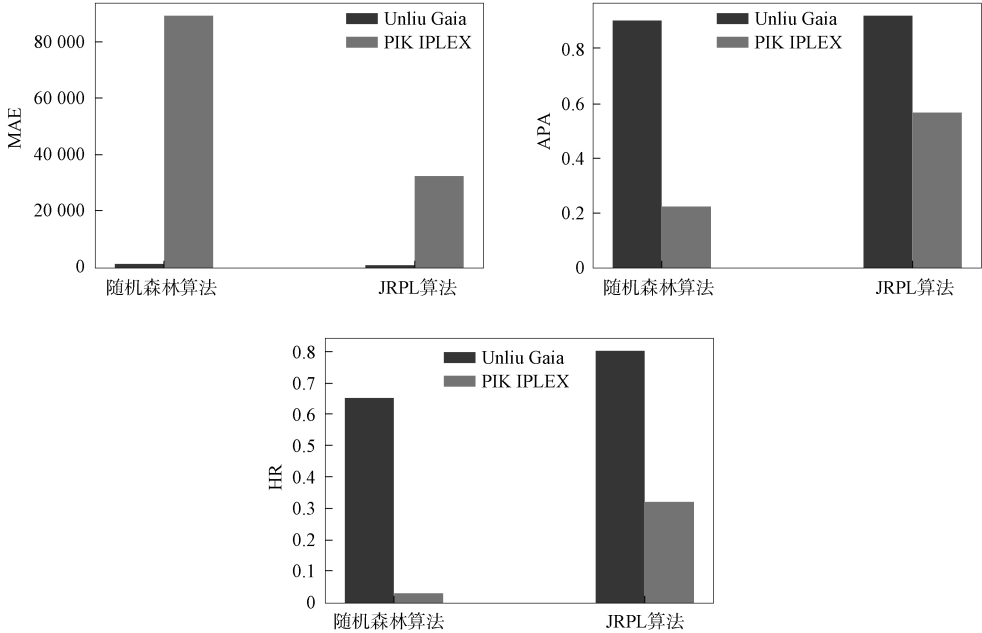


图 2 两种算法在数据集 Unliu Gaia 和 PIK IPLEX 上的 MAE,APA 和 HR  
Fig. 2 MAE,APA and HR of two algorithms on Unliu Gaia and PIK IPLEX datasets

在数据集 Unliu Gaia 上, 随机森林算法有很高的预测准确率, 在 MAE,APA,HR 3 个指标上的结果都很好; 而 JRPL 算法在 APA 和 HR 两个指标上虽然没有更好的表现, 但也不低于随机森林算法, 在 MAE 指标上, JRPL 算法则显著优于随机森林算法. 在数据集 PIK IPLEX 上, JRPL 算法在 3 个指标上都体现了优于随机森林算法的能力. 表明在单纯依靠传统机器学习算法预测作业执行时间时, JRPL 算法来自数据局部特征的考量可显著改进预测准确度. 因此, 无论是在随机森林算法表现较好的数据集 Unliu Gaia 上, 还是在其表现较差的数据集 PIK IPLEX 上, 综合考虑了局部性的 JRPL 算法都能在 MAE,APA,HR 3 上指标上取得优势. 在数据集 PIK IPLEX 上, 随机森林和 JRPL 算法的 MAE 都在  $10^5$  的数量级上, 分析认为这是两方面原因导致的: 一方面是随机森林算法在预测该数据集时表现不佳, 另一方面是预测时出现的个别极大误差点大幅度拉升了整体 MAE.

本文先将测试集的 40 000 个作业按实际执行时间长短由短到长排序, 再按执行时间长短排列将作业平均分成 20 组, 分别计算其在每组作业的 MAE,APA,HR 指标, 结果如图 3 和图 4 所示. 由图 3 和图 4 可见, JRPL 算法在每组作业的各指标上都体现了优于随机森林算法的预测能力, 且稳定性良好.

图 3 和图 4 对数据进行了分段展示, 进一步体现了 JRPL 算法在任何时刻都不弱于基础机器学习算法的特点: 当两种算法的指标重合时, 表明该范围内绝大多数投票都偏向于机器学习算法, 机器学

习算法可成为 JRPL 算法的保底策略; 当两种算法指标相差较大时, 表明该范围内采用时序性分析结果的投票更多, 加入时序性考量能弥补机器学习算法的不足部分. 综合可见, JRPL 算法的性能优于作为基础的机器学习算法.

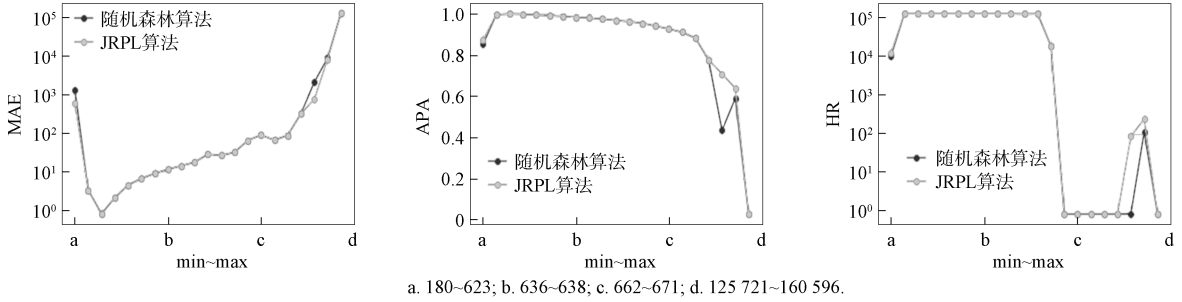


图 3 两种算法在 Unliu Gaia 分组作业上的 MAE, APA 和 HR

Fig. 3 MAE, APA and HR of two algorithms on Unliu Gaia grouping tasks

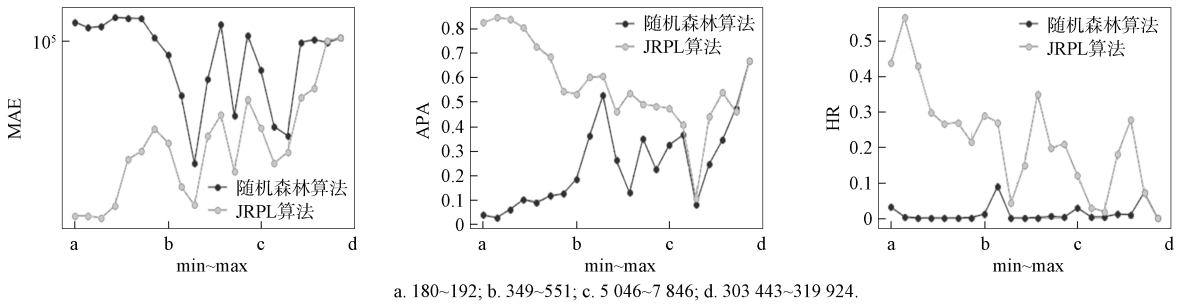


图 4 两种算法在 PIK IPLEX 分组作业上的 MAE, APA 和 HR

Fig. 4 MAE, APA and HR of two algorithms on PIK IPLEX grouping tasks

分析图 4 中 MAE 和 APA 在最后一段的折线图, 发现 APA 参数较好, 但 MAE 却异常大, 达到  $10^5$  数量级. 这种矛盾现象存在的原因在于在很多预测良好的作业外, 存在极少数误差极大的特异点. 由于 APA 的计算方式将其值限定在  $0 \sim 1$  之间, 因此这些特异点产生的影响能被 APA 很快中和, 但会极大拉升 MAE 参数.

本文对决策树算法、SVR 算法及分别对应扩展的 JRPL 算法在数据集 Unliu Gaia 和 PIK IPLEX 上进行相关实验, 实验结果如图 5 所示. 由图 5 可见, SVR 算法及 JRPL(SVR)算法与随机森林算法及 JRPL(随机森林)算法的预测结果相近, 但在 APA 上略逊, 在 HR 上更高, 表明在一定程度上, 在这两个数据集中, 决策树算法预测相对误差的方差更大. 而 SVR 算法则在 APA 和 HR 上表现不佳, 在 MAE 上较优秀, 表明相比决策树算法, SVR 算法预测的绝对误差更低, 相对误差更高, 即 SVR 算法能更好地预测长任务, 但在预测短任务的执行时间上表现不佳. 而以 SVR 算法为补充的 JRPL(SVR)算法则在一定程度上弥补了其在 APA 和 HR 上的表现, 并在 MAE 上取得了更好的结果.

最后, 将随机森林算法、决策树算法和 SVR 算法及对应扩展的 JRPL 算法在文献[7]中数据集 RICC, SDSC Par, LANL CMS, ANL, KIT FH2 上进行实验, 结果如图 6 所示. 由图 6 可见, JRPL 算法预测结果各评测指标在各方面都优于原机器学习算法, 表明 JRPL 算法利用数据的局部特征并结合机器学习算法进行程序执行时间预测相比单独使用机器学习算法在预测效果上的提升具有普适性.

综上所述, 针对高性能计算系统中作业执行时间预测不能充分利用作业的局部性、预测精度低的问题, 本文提出了一种考虑局部特征的高性能计算作业执行时间预测算法, 是对传统机器学习算法的一种改进, 本文实现并分析了其在 7 个典型实际数据集中的数值实验结果. 实验结果表明, 在不同环境下, 加入来自数据局部特征的 JRPL 算法在各评估指标上都能取得比传统机器学习算法同等或更好的性能. 在对高性能计算中心的作业进行执行时间预测时, JRPL 算法相较于传统机器学习算法更显著、稳定且普适.

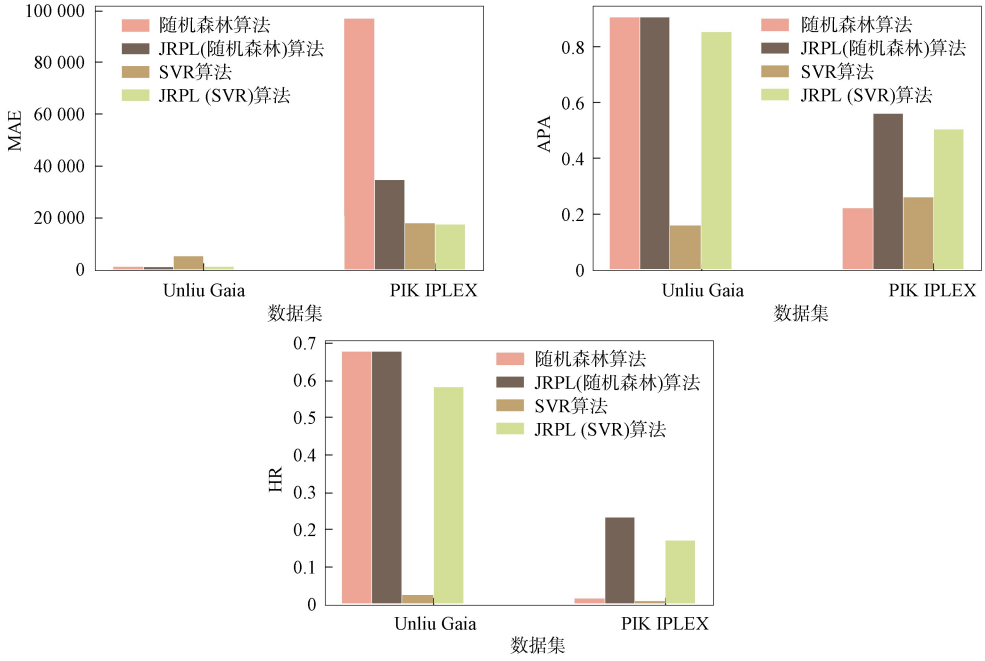


图 5 不同算法在数据集 Unliu Gaia 和 PIK IPLEX 上的 MAE, APA 和 HR  
 Fig. 5 MAE, APA and HR of different algorithms on Unliu Gaia and PIK IPLEX datasets

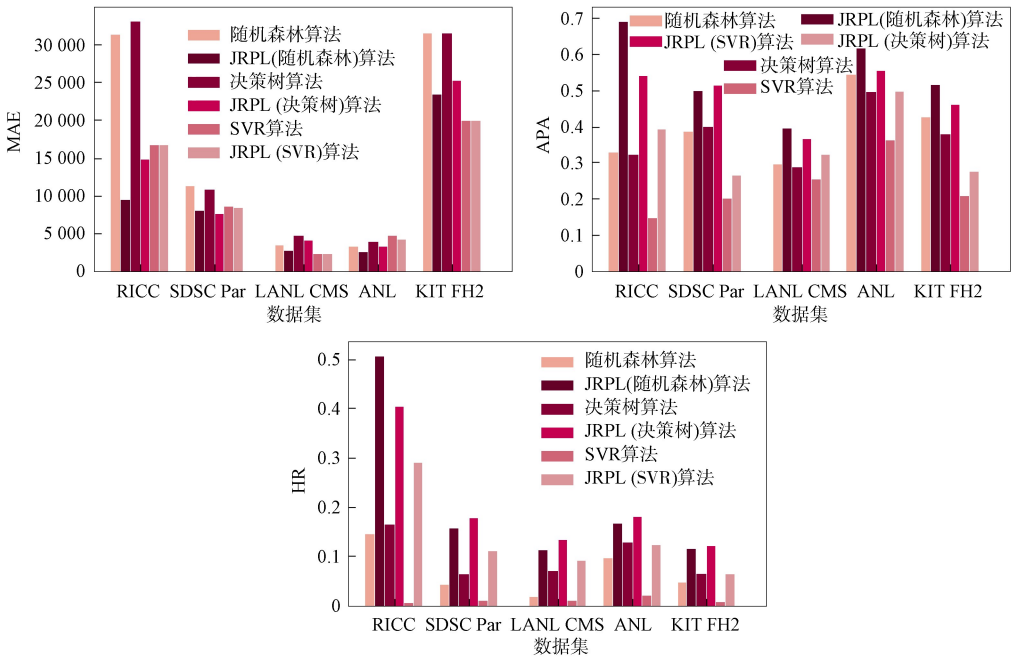


图 6 不同算法在 5 个数据集上的 MAE, APA 和 HR  
 Fig. 6 MAE, APA and HR of different algorithms on five datasets

参 考 文 献

[ 1 ] 李于锋. 超算 workflow 调度算法研究及运行时系统研制 [D]. 北京: 中国工程物理研究院, 2021. (LI Y F. Research on Scheduling Algorithms and Runtime System Development of Supercomputing Workflow [D]. Beijing: China Academy of Engineering Physics, 2021.)

[ 2 ] TANG W, LAN Z L, DESAI N, et al. Fault-Aware, Utility-Based Job Scheduling on Blue, Gene/P Systems [C]//IEEE International Conference on CLUSTER Computing and Workshops. Piscataway, NJ: IEEE, 2009: 1-10.

- [ 3 ] CHIANG S H, ARPACI-DUSSEAU A, VERNON M K. The Impact of More Accurate Requested Runtimes on Production Job Scheduling Performance [J]. *Job Scheduling Strategies for Parallel Processing, Lecture Notes in Computer Science*, 2002, 2537: 103-127.
- [ 4 ] MU'ALEM A W, FEITELSON D G. Utilization, Predictability, Workloads, and User Runtime Estimates in Scheduling the IBM SP2 with Backfilling [J]. *IEEE Transactions on Parallel and Distributed Systems*, 2001, 12(6): 529-543.
- [ 5 ] FEITELSON D G, MU'ALEM A W. Utilization and Predictability in Scheduling the IBM SP2 with Backfilling [C]//*Proceedings of 12th International Parallel Processing Symposium*. Piscataway, NJ: IEEE, 1998: 542-546.
- [ 6 ] CIRNE W, BERMAN F. A Comprehensive Model of the Supercomputer Workload [C] //*IEEE International Workshop on Workload Characterization*. Piscataway, NJ: IEEE, 2001: 140-148.
- [ 7 ] TRAN N M, LEX W. Using Historical Data to Predict Application Runtimes on Backfilling Parallel System [C]//*18th Euromicro Conference on Parallel, Distributed and Network-Based Processing*. Piscataway, NJ: IEEE, 2010: 246-252.
- [ 8 ] GIBBONS R. A Historical Application Profiler for Use by Parallel Schedulers [J]. *Job Scheduling Strategies for Parallel Processing, Lecture Notes in Computer Science*, 1997, 1291: 58-77.
- [ 9 ] SMITH W, FOSTER I, TAYLOR V. Predicting Application Run Times Using Historical Information [J]. *Job Scheduling Strategies for Parallel Processing, Lecture Notes in Computer Science*, 1998, 1459: 122-142.
- [10] DOWNEY A B. Using Queue Time Predictions for Processor Allocation [J]. *Job Scheduling Strategies for Parallel Processing, Lecture Notes in Computer Science*, 1997, 1291: 35-57.
- [11] TSAFRIR D, ETSION Y, FEITELSON D G. Backfilling Using System-Generated Predictions Rather than User Runtime Estimates [J]. *IEEE Transactions on Parallel and Distributed Systems*, 2007, 18: 789-803.
- [12] LIANG F, LIU Y Z, LIU H, et al. A Parallel Job Execution Time Estimation Approach Based on User Submission Patterns within Computational Grids [J]. *International Journal of Parallel Programming*, 2015, 43(3): 440-454.
- [13] 余莹,李肯立,徐雨明. 计算集群中一种基于任务执行时间的组合预测方案 [J]. *计算机应用*, 2015, 35(8): 2153-2157. (YU Y, LI K L, XU Y M. Combined Prediction Scheme for Runtime of Tasks in Computing Cluster [J]. *Journal of Computer Applications*, 2015, 35(8): 2153-2157.)
- [14] 蒋炎华. 网格环境下任务的执行时间预测技术研究 [J]. *计算机工程与设计*, 2011, 32(10): 3428-3430. (JIANG Y H. Research of Task Execution Time Prediction Technology in Grid Computing Environments [J]. *Computer Engineering and Design*, 2011, 32(10): 3428-3430.)
- [15] NISSIMOV A. Locality and Its Usage in Parallel Job Runtime: Distribution Modeling Using HMM [M]. Jerusalem: Hebrew University of Jerusalem, 2006: 1-55.
- [16] AMARÍS M, DE CAMARGO R Y, DYAB M, et al. A Comparison of GPU Execution Time Prediction Using Machine Learning and Analytical Modeling [C]//*2016 IEEE 15th International Symposium on Network Computing and Applications (NCA)*. Piscataway, NJ: IEEE, 2016: 326-333.
- [17] GAO Y J, GU X Y, ZHANG H Y, et al. Runtime Performance Prediction for Deep Learning Models with Graph Neural Network [C]//*2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. Piscataway, NJ: IEEE, 2023: 368-380.
- [18] LOUPPE G. Understanding Random Forests: From Theory to Practice [EB/OL]. (2014-07-26)[2025-01-01]. <https://arxiv.org/abs/1407.7502>.
- [19] 许伦凡,熊敏,肖永浩. 基于调度历史数据在线预测作业执行时间 [J]. *计算机应用研究*, 2020, 37(3): 763-767. (XU L F, XIONG M, XIAO Y H. On-Line Prediction of Application Using Schedule Historical Data [J]. *Application Research of Computers*, 2020, 37(3): 763-767.)
- [20] FEITELSON D G, TSAFRIR D, KRAKOV D. Experience with Using the Parallel Workloads Archive [J]. *Journal of Parallel and Distributed Computing*, 2014, 74(10): 2967-2982.