引言

本文针对大型计算平台作业中断和提前终止问题，分三步解决。首先研究多个大型平台的跟踪记录数据，以确定不同集群下作业终止的特征和根本原因。其次设计了一个基于机器学习的框架来预测作业的终止。最后通过实例展示了上述框架如何基于任务克隆策略来减轻任务失败的影响。

跟踪记录说明

本研究使用以下三个组织的跟踪记录：

（a）Google：一个月内提交给Google的由12000个节点组成的多功能计算集群的所有34.9万个作业的跟踪记录。

（b）CMU OpenCloud：OpenCloud是卡内基梅隆大学的一个64节点Hadoop集群，本研究所使用的跟踪记录包含了30个月内的7.8万个作业。

（c）LANL HPC集群：LANL为256节点HPC系统，本研究使用了其为期三年的作业跟踪记录。

失败作业特征分析

首先研究不同集群的失败作业的特征。

（a）作业持续时间：持续时间定义为该作业在退出之前在集群中花费的总任务分钟数。表2显示，所有三个集群中成功作业持续时间最短，被杀死的作业花费的时间最长。

（b）并行度：图1显示了数据集中所有作业的结果。当按系统时间进行标准化时，与单个作业相比，并行作业每单位时间的作业中断率较低（LANL集群中唯一被杀死的作业除外）。

（c）群集资源使用：图2分别绘制了完成任务，失败任务或被杀死任务的任务平均请求资源的分配情况（对于LANL任务，仅按请求数目的处理器）。数据显示，资源请求在所有集群中的成功和不成功作业之间明显不同，这表明对作业请求的集群资源数量的了解可以潜在地用于预测作业的最终状态。图3研究了Google和OpenCloud集群中作业的集群资源平均利用率分布。数据显示，成功和不成功的工作之间存在不同的资源使用行为；在OpenCloud中已完成和中断的作业之间还存在不同的I / O使用行为。

执行失败的根本原因

下面基于跟踪记录给出导致执行失败的根本原因的假设。

（a）违反资源限制。图4-（a）绘制了该比例在Google任务中的分布情况，并按任务退出状态进行了细分；图4-（b）绘制了任务执行期间报告的最大CPU /内存量与任务请求的CPU /内存之间的比率。数据显示，内存管理问题与任务失败相关，尽管失败的任务比其他任务需要更多的内存，但它们仍然更有可能超过其内存限制，并且随着时间的推移会经历内存使用的急剧增长。

（b）作业容错配置。这里主要研究用于配置Hadoop的“推测执行”功能，处理不良数据记录和管理任务重试的容错旋钮。结果表明，对于具有配置数据的CMU Hadoop集群，不成功的作业更有可能具有与默认框架设置不同的容错配置。

（c）任务失败导致作业失败。图6绘制了针对Google和OpenCloud的任务下一次尝试成功的概率与过去失败尝试次数的函数。结果表明，对于以任务重新执行为恢复机制的集群，尝试重新执行一次或多次以上的任务是徒劳的。

（d）机器中断。表III列出了与机器拆卸相关的作业/任务事件的类型细分，还显示了每种事件类型，此类型事件总数中有多少比例与机器故障相关。结果表明，对于数据集中有机器停机日志的集群，机器移除相关的大多数任务/作业事件是逐出或杀死。

预测不成功的执行

这里主要关注Google跟踪以设计和评估提出的预测框架。

（a）预测框架设计：设计目标是设计一个能够预测集群中正在运行的作业或任务未成功终止的框架。表IV总结了用作预测变量的输入变量。预测技术是使用基于决策树的机器学习技术随机森林（RF）对工作和任务进行预测，将RF算法配置为在每次迭代中使用50棵树，并对数据进行拆分，以使跟踪中Google的工作中有70％用于培训，而30％用于测试。预测旋钮使用默认的分类（决策）阈值0.5来预测类成员身份，但在的某些实验中将阈值从0.1更改为最大0.9；对于训练集，采用“数据超采样”以校正原始数据偏差。对于预测指标，用来量化预测质量的指标是精度和召回率。精度是指真实肯定的数量除以正确肯定和错误肯定的数量。召回是真实肯定的数量除以真实肯定和错误否定的数量。预测场景包含四个场景。预测场景一是仅使用作业提交时可用的数据来预测作业的不成功终止，预测场景二是假设了解该作业的早期资源使用指标，尤其是在执行的前5分钟内（除了配置参数之外）该作业任务的CPU，内存和磁盘消耗的平均和标准偏差。预测场景三添加了一个运行时标志，一旦在作业中的所有任务中检测到单个任务失败，就将其设置为TRUE。预测场景四用到了上述讨论提到的全部参数‘Config + 5min Usage + TaskFailFlag’。

（b）作业被杀死预测：图7显示了在不同预测变量下获得的精度和召回率，左图显示的是将分类阈值从[0.1-0.9]更改时的结果，右图显示的是在固定阈值0.5下更改训练集中的采样率时的结果。数据显示，即使在仅使用作业配置参数的最简单情况下，也可以非常准确地预测作业终止。

（c）作业失败预测：图8显示了使用相同的失败预测配置时，工作失败预测器的精度与召回率。结果展示了在不同设置下获得的预测质量的高得多的可变性，以及更广泛的精确调用折衷范围。图9绘制了第一个任务失败发生后并行作业中的剩余时间（左）和并行作业寿命中的剩余部分（右）的分布。该结果表明，可以及早准确地预测工作失败。

（d）任务失败预测：本框架首先在第V-D1部分中预测任务的最终状态，即任务退出时（可能在多次任务重试之后）的状态，然后在第V-D2部分中预测预测中间重试尝试的状态。为预测任务最终状态，还还添加了一个AttemptFailFlag预测器，如果过去至少一次对该任务的尝试失败，则将其设置为true。图10（a）显示了任务级别的结果。结果显示，与作业相比，仅使用配置数据会产生较不精确的预测，而使用AttemptFailFlag可以最大程度地提高预测效果，特别是召回率。对于预测任务尝试失败，图10（b）显示，与任务级和作业级失败相比，本框架可以更精确地预测单个任务尝试的失败。

减轻执行失败的影响

（a）策略建议：限制重试次数；启用其他监视器；增加检查点的频率；调整计划优先级或分配的资源；计划冗余任务（推测执行）。

（b）用例：主动克隆。克隆算法在算法1中进行了描述。这里使用基于Google跟踪的模拟来评估使用主动克隆策略可以加速多少任务完成时间。图11（a）和（b）显示了算法1下完成时间的累积分布函数（CDF），是追踪记录中原始完成时间的一部分，图11（c）显示了对克隆C的不同值进行实验的结果。对于开销评估，表V报告了算法1的不同配置带来的开销（系统上的总工作量的增加）。结果表明，为了使开销保持合理，需要保守地选择阈值，范围在0.9或更高。

结论

本文使用来自不同生产系统的跟踪数据集来研究并行集群中的工作终止，确定了可以区分在现场如何配置和执行不成功作业的模式。研究发现，在多个群集中，不成功的作业会运行更长的时间，需要更多的群集资源，报告的I / O活动要比已完成的作业重，并且其参数配置通常会偏离框架的默认参数。本文设计和评估了作业和任务终止的有效预测器，并确定了具备工作或任务终止预测能力的参数。最后，本文提出了多种方法来利用本文的研究结果来减轻集群中的失败，并提供对其中之一的详细评估。