

基于作业记账日志的并行作业特征分析工具 JobCAT*

田鸿运¹, 刘旭¹, 武林平¹, 罗红兵¹, 莫则尧²

(1. 北京应用物理与计算数学研究所, 北京 100094; 2. 中物院高性能数值模拟软件中心, 北京 100088)

摘要:并行作业特征分析是负载分析的重要基础。作业记账日志是开展作业特征分析的重要数据源。由于作业记账日志中没有记录应用名称, 现有工具无法按应用名称开展作业特征分析。提出基于关键字模糊匹配的作业记账日志标记方法, 设计通用的作业数据模型和柔性可扩展软件架构, 集成实现并行作业特征分析工具 JobCAT。通过某超级计算机系统百万量级作业记账日志数据测试验证, JobCAT 的作业记账日志标记率大于 95%。JobCAT 支持 7 个插件、29 项统计报表, 可一键生成应用的作业特征分析报告, 对负载分析研究具有实用价值。

关键词:负载分析; 作业记账日志; 作业特征分析工具; 超级计算机

中图分类号: TP315 **文献标志码:** A **文章编号:** 1001-2486(2022)05-063-08

Parallel job characteristic analysis toolkit based on job accounting logs: JobCAT

TIAN Hongyun¹, LIU Xu¹, WU Linping¹, LUO Hongbing¹, MO Zeyao²

(1. Institute of Applied Physics and Computational Mathematics, Beijing 100094, China;

2. CAEP Software Center for High Performance Numerical Simulation, Beijing 100088, China)

Abstract: Characteristic analysis of parallel job is the basics of workload analysis research. Job accounting log is an important data source of job characteristic analysis, however, existing tools cannot do statistics analysis with application name, because of application name not recorded in job accounting log. To solve this problem, a novel marking method for job accounting log was proposed, which based on keyword fuzzy matching. Combined with a general job data model and a flexible extensible software architecture, a parallel job feature analysis tool JobCAT was implemented. According to verification test by millions of job accounting log data from a supercomputer system, the log marking rate of JobCAT was greater than 95%. JobCAT supports 7 plugins and 29 statistical reports, and can easily make analysis report classified by application name, which has practical value to workload analysis research.

Keywords: workload analysis; job accounting log; job characteristic analysis toolkit; super computer

并行作业特征分析是负载分析的重要基础, 对于超算系统装备建设单位具有实际应用价值。通过并行作业统计特征分析, 可以定量给出并行应用的机时使用情况、作业并行规模、作业运行成功率、运行波动率等作业特征, 量化分析高性能计算应用的能力发展状态及其对于超级计算机算力的需求, 从而为系统容量配置、系统架构选型以及应用特征导向的系统能力评测、系统资源调度优化等研究提供参考^[1-2]。例如, 美国国家能源研究科学计算中心(national energy research scientific computing center, NERSC)针对其装备的高性能计算机(Hopper-2011、Edison-2013、Cori-2016)定期发布作业数据分析报告, 从应用科学

领域、应用并行能力、编程语言、数学库、内存使用、IO使用等多个维度详细分析系统上的负载特征及其演化规律, 进而形成相应的数值模拟应用-算法特征矩阵, 指导系统评测基准程序的研制与选型和系统容量配置优化^[3-6]。Michael等对美国国家超算应用中心(national center for super-computing applications, NCSA)的Blue Waters系统负载进行了分析研究^[7-8], 不仅从科学领域、应用、算法等方面开展了宏观的负载统计分析, 还从并行方式、内存使用、IO行为特征、异构资源使用特征等方面深入研究了该系统上的高性能计算应用特征, 指导系统优化研究, 引导下一代系统设计。橡树岭国家实验室(Oak Ridge

* 收稿日期: 2021-12-29

基金项目: 国家重点研发计划资助项目(2018YFB0204003)

作者简介: 田鸿运(1986—), 男, 重庆人, 博士研究生, E-mail: tian_hongyun@iapcm.ac.cn;

莫则尧(通信作者), 男, 研究员, 博士, 博士生导师, E-mail: mo_zeyao@iapcm.ac.cn

national laboratory, ORNL) 的 You 等对该实验室的 Kraken 计算机开展了作业特征统计分析研究^[9], 他们的研究着重于从作业并行规模、作业排队时间、队列资源利用率等角度研究资源的使用情况进而为调度优化提供参考。黄瑞芳等基于气象海洋应用的计算特征分析, 设计了面向气象海洋应用的高性能计算机评测体系, 他们的分析着重于定性分析^[10]。

并行作业特征的量化分析主要基于作业记账日志数据开展。Baer 等开发了针对 PBS 作业管理系统的作业数据统计工具 pbsacct^[11], 该工具不支持按照用户需求自定义数据分析插件。Nielsen 等开发的 Slurm_tools^[12], 提供对于作业数据分析的统计脚本, 但主要是对 sacct 和 sreport 的封装, 没有提供应用维度的统计分析功能。LLNL 实验室开发的 XDMoD, 是一款针对并行作业管理系统的负载分析工具, 但不支持用户自定义数据统计分析插件^[13]。

当前, 基于作业记账日志开展并行作业特征分析存在三个挑战。首先, 并行作业记账日志数据中没有记录应用名称, 现有工具无法按应用名称开展作业特征分析。其次, 不同作业管理系统的作业记账日志数据格式可能存在差别, 对分析工具的跨平台兼容性也提出了挑战。此外, 不同业务场景对于并行作业特征分析的侧重点各有不同, 多目标场景对于分析工具的功能可扩展性也提出了要求。

针对上述挑战, 提出了一种基于关键字模糊匹配的作业记账日志应用名称标记方法, 为作业记账日志增加应用名称索引标签, 使作业记账日志具备了从应用维度开展统计分析的能力。同时, 设计了一种通用的作业数据描述模型, 通过将作业记账日志数据格式转换为标准格式, 屏蔽了不同作业管理系统记账日志数据格式的差异, 实现了作业数据统计分析与作业数据格式的解耦。针对多目标场景对软件功能灵活性的挑战, 设计了支持插件定制的柔性可扩展软件架构, 支撑按需求定制开发作业数据分析插件。基于前述三项关键技术, 集成实现了并行作业特征统计分析工具 JobCAT(job characteristic analysis toolkit)。

1 并行作业记账日志数据

1.1 资源管理系统

资源管理系统是超级计算机中的重要基础软件, 负责并行作业的资源分配、调度、管理、回收、记录、查询。目前, 最为主流的并行作业管理系统

有: SLURM (simple linux utility for resource management)、LSF (load share facility)、PBSpro (portable batch system professional) 以及 TORQUE (Terascale open-source resource and queue manager) 等^[14]。其中, SLURM 是由美国劳伦斯·利弗莫尔国家实验室 (Lawrence Livermore national laboratory, LLNL) 开发的一种可用于大型计算集群的资源管理器和作业调度系统, 被全球范围内的超级计算机广泛使用^[15]。下面以 SLURM 为例, 介绍并行作业管理系统。

1.2 SLURM 并行作业记账日志数据格式

SLURM 会为每个作业生成作业记账日志数据, 用户可通过 sacct 查询作业记账数据。SLURM 作业记账日志字段如表 1 所示, sacct 支持按起止日期、用户名、作业 ID 等方式查询, 但不支持按应用名称进行查询^[15]。

表 1 SLURM 作业记账日志字段示例
Tab. 1 Example of SLURM job accounting log fields

字段	说明
AllocCPUS	作业申请分配的 CPU 核数
AllocNodes	作业申请分配的计算结点数
CPUSTime	作业使用的 CPU 时间, 单位: 核 · s
ElapsedRaw	作业运行时长, 单位: s
JobID	作业 ID 号
JobName	作业名称
WorkDir	作业执行路径
Start	作业开始运行时间
End	作业结束运行时间
State	作业退出状态
Submit	作业提交时间

SLURM 作业记账日志包括作业号记账日志和作业步记账日志两部分。其中, 作业号记账日志的 JobID 为一个整数, 作业步记账日志的 JobID 在对应作业号记账日志的基础上增加作业步编号后缀。实际分析时需要将作业号记账日志数据和作业步记账日志数据合并才能得到一条完整的记账日志数据信息。

SLURM 支持用户使用交互式、预分配式、批处理式三种方式提交并行作业。实际中, 用户常使用批处理方式。对于批处理方式提交的作业, 作业号记账日志的 JobName 字段对应于批处理脚本名称, WorkDir 字段对应于批处理脚本执行目

录,而作业步记账日志中的 JobName 字段,则对应于作业名称,WorkDir 字段对应于作业执行目录。

实际使用中,用户为了区分不同版本的可执行程序文件,常会在可执行程序名称或者作业执行脚本上附带版本号、日期、用户名等信息,或者将可执行程序按照领域内涵进行命名,将可执行程序名称放入批处理作业脚本文件名中,导致同一个应用程序在作业记账日志中的命名并不唯一。

此外,为了加快大规模作业的加载速度,用户常常会组合使用多条作业执行命令,例如将作业投递命令(如:srun)与数据分发命令(如:sbcast)组合使用,导致一个并行作业产生多条作业日志数据,而这些日志记录中可能存在诸如 mkdir、rm 等辅助功能的命令,导致应用名称易被错误标记。

上述用户行为,导致实际的并行应用名称在作业记账日志中的位置、顺序、大小写没有固定特征。应用名称既可能存在于批处理脚本名称中,也可能存在于某一个作业步的作业名称中。同时,应用名称大小写也可能因为用户命名习惯的差异而各不相同。用户行为的差异,给作业记账日志的应用名称识别标记带来挑战。

2 作业记账日志分析工具 JobCAT

2.1 基于关键字模糊匹配的作业日志标记方法

为了应对并行作业记账日志中作业名称多样性给应用名称标记带来的困难,提出一种基于关键字模糊匹配的作业记账日志标记方法,如图 1 所示。该方法的实现步骤如下:

1) 建立应用关键字列表。将系统上的常见应用程序名称和可忽略的作业名称各建立一个关键字列表。以 python 为例,可以使用类似{"applist": ["app1", "app2", "app3"], "ignore_list": ["mkdir", "rm"]} 的字典格式存储系统上的应用程序名称和忽略作业名称列表,并将其存入 json 格式的配置文件中,在工具启动时动态加载解析。如果两个程序的关键字存在包含关系,则字段长的关键字应列在前面。例如:lared-s 和 lared 存在包含关系,则将 lared-s 写在前面。对于应用名称关键字字典和可忽略作业名称字典,在后续测试中不断补充完善。

2) 作业日志自动分组。遍历系统在指定时间段内的作业日志,按照 jobid 进行日志分组。设计一个临时日志池 logpool, 然后开始遍历作业日志。如果 jobid 不包含 '.' 分隔符并且 logpool 为空,则表明是某作业日志的第一条日志记录信息,

将日志添加到 logpool 中;如果 jobid 不包含 '.' 分隔符并且 logpool 非空,则表明是一条新的日志,先进行已分组日志的解析工作;如果 jobid 包含 '.' 分隔符并且作业 ID 与当前解析的作业 ID 一致,则表明是当前解析作业的作业步日志,将其添加到当前的日志分组中;如果 jobid 包含 '.' 分隔符并且其作业 ID 与当前解析的作业 ID 不一致,则表明是一条新的作业日志,先进行已分组日志的解析工作。

3) 关键字模糊匹配遍历。遍历已分组日志的 JobName 和 WorkDir 字段,将其与应用名称关键字字典进行遍历匹配,匹配时不区分大小写,如果匹配成功则返回。对于未识别的作业日志,表明日志中不包含应用名称关键字,可能是由于用户根据领域内涵对应用名称进行了重命名,此时尝试通过关键字重命名映射表找出作业的应用程序名。关键字重命名映射表是根据常见用户习惯设置的作业名称到实际可执行应用的映射表,通常这类用户在不同系统上的命名习惯相同,从而可以使用相同的“关键字重命名映射表”。如果发现用户在不同系统上的命名习惯不相同,也可以为每一台超算系统设置不同的关键字重命名映射表。如果遍历完关键字重命名映射表依然未能成功匹配,则将应用名称标记为“unknown”。

4) 数据合并。对于成功识别出应用程序名称的作业日志,合并相应的作业步日志和该作业 ID 对应的首条日志信息,并将合并后的作业日志标记为指定应用的作业日志数据。

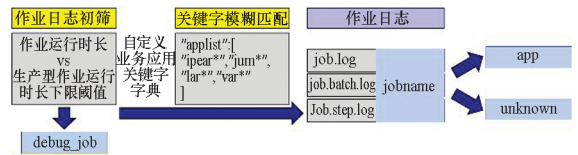


图 1 基于关键字模糊匹配的作业日志识别方法

Fig. 1 Job log recognition method based on keyword fuzzy matching

2.2 支持大纲检查的通用作业数据描述抽象模型

为了应对不同作业管理系统记账日志格式的差异,设计了一种支持大纲格式检查的通用作业数据描述模型,如图 2 所示。

该模型将作业日志描述信息划分为四个维度,分别是基本信息维度、时间信息维度、资源消耗信息维度以及其他信息维度。其中:①基本信息维度主要包含与作业描述最基本的相关要素,目前包含作业 ID (JOBID)、作业名称 (JOBNAME)、应用名称 (AAPPNAME)、程序版本

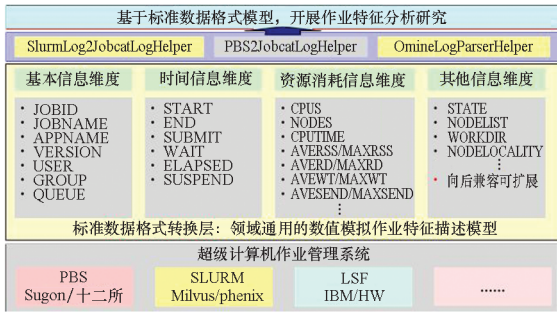


图 2 统一作业数据格式

Fig. 2 Uniformed job account log

(VERSION)等7个字段;②时间信息维度包括与作业信息描述相关的时间要素,目前包含作业提交时间(SUBMIT)、作业开始运行时间(START)、作业结束运行时间(END)等6个字段;③资源消耗信息维度主要包括与CPU、内存、IO相关的要素,目前包含CPU核数(CPUS)、计算节点数(NODES)、机时(CPUTime,单位为核·s)等13个字段;④其他信息维度包括基本信息、时间信息、资源消耗信息之外的共性要素,目前包含作业返回状态(STATE)、节点列表(NODELIST)等字段。

通过 schema 库提供的大纲格式检查功能, JobCAT 实现了作业记账日志数据的格式检查。后续只需实现标准数据格式向各作业管理系统数据格式的转换层,即可实现对相应作业管理系统的平滑移植,从而保证 JobCAT 的跨平台兼容性。目前, JobCAT 已经实现了对 SLURM 作业管理系统的数据库格式转换层支持。

2.3 支持插件定制的柔性可扩展软件架构

为了灵活应对不同场景的作业数据分析需求, JobCAT 实现了一套支持插件定制的柔性可扩展架构,如图 3 所示。主要通过插件模板、插件管理器、运行时管理器三部分协同实现。

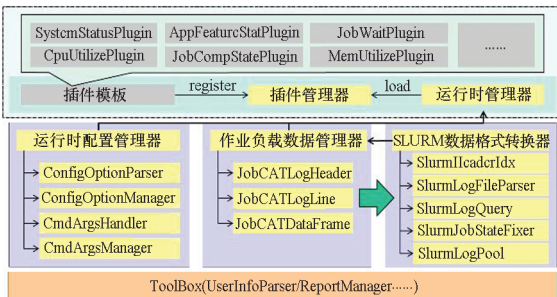


图 3 JobCAT 总体架构

Fig. 3 Architecture of JobCAT

插件模板包含 doanalysis()、drawdatas()、

help()等共性接口, doanalysis()和 drawdatas()根据用户需求定制实现,分别负责数据的分析与数据分析结果的输出展示, help()提供了缺省实现,用户可以选择自行实现也可以选择使用模板自带的缺省实现。根据用户需求定制实现的分析插件,通过注册器注册到插件管理器中即可被插件管理器自动识别并加载运行。

插件管理器负责搜索、加载、运行、管理插件,同时将运行时状态信息传递给分析插件。

运行时管理器负责运行时配置参数的管理以及原始作业数据的查询与转换,通过运行时配置管理器, JobCAT 可以选择与当前系统平台适配的作业数据格式转换器进行数据查询与格式转换。运行时管理器根据用户运行时输入参数,动态查询并转换作业数据格式,并将其作为数据分析插件的输入。运行时管理器由运行时配置管理器、作业负载数据管理器、SLURM 数据格式转换器三部分组成。其中,运行时配置管理器包含配置选项解析器(ConfigOptionParser)、配置选项管理器(ConfigOptionManager)、命令行参数管理器(CmdArgsManager)、命令行参数解析句柄(CmdArgsHandler)。作业负载数据管理器包含 JobCAT 日志格式模板(JobCATLogHeader)、JobCAT 日志数据行抽象类(JobCATLogLine)以及 JobCAT 数据集(JobCATDataFrame)。SLURM 数据格式转换器负责 SLURM 作业记账日志格式向 JobCAT 通用日志模型的转换,包含作业记账日志格式模板(SlurmHeaderIdx)、作业记账日志解析器(SlurmLogFileParser)、作业记账日志查询器(SlurmLogQuery)、作业状态修复工具(SlurmJobStateFixer)、作业记账日志解析缓存池(SlurmLogPool)。

通过上述三项关键技术, JobCAT 实现了作业记账日志的应用识别标记,满足了工具的跨平台兼容性需求,支持根据业务分析场景开展定制分析。

2.4 JobCAT 主要功能介绍

JobCAT 的插件清单见表 2。目前 JobCAT 支持 7 个分析插件,共包含 29 项统计报表。这些功能插件可以为用户提供超算资源使用分析、内存使用分析、应用特征分析等并行作业数据统计分析支持,进而支撑超算资源调度效率优化、计算节点内存容量配置优化设计、典型业务应用程序筛选等研究,为超算硬件能力建设服务。

表 2 JobCAT 的插件清单
Tab.2 Plugin list of JobCAT

名称	说明
systatus	集群资源利用率总体情况分析插件包
cpuutilz	作业机时与作业数量统计(计算资源利用率)
memutilz	作业内存使用情况分析
fsutilz	作业 IO 使用情况分析
jobstate	作业完成情况分析
jobwait	作业排队情况分析(计算需求覆盖率)
appfeature	应用特征分析

JobCAT 支持通过命令行参数选项,可以动态设置数据分析的起止日期、指定特定的目标用户群体、应用程序集等,并支持根据需要动态加载分析插件。

JobCAT 支持命令行与配置文件相结合的配置管理方式,对于系统平台运行环境相关的配置参数,可通过配置文件设置。用户还可在配置文件中灵活指定应用名称关键字列表、作业并行规模分类器、作业运行时长分类器、文档数据格式转换器、用户信息查询缓存表等配置信息。与分析插件、命令行参数一样,配置文件选项也可以按需灵活扩展。

3 实验与分析

3.1 实验环境简介

实验集群包含 3 584 个计算节点,共 114 688 个处理器核,系统峰值性能为 2 803.2 TFlops,采用胖树网络拓扑,SLURM 版本为 v18.08.04,集群上共 232 位用户。分别采用 2018-05-01 至 2018-11-01、2018-05-01 至 2019-05-01、2018-05-01 至 2020-05-01 三个时间段的作业记账日志开展识别率验证,并使用最后一时间段的作业记账日志开展作业特征统计分析研究。

3.2 作业记账日志识别率验证

实际生产环境下,面向真实物理过程研究的科学计算数值模拟应用往往需要大规模、长时间运行,运行时间短的作业常常是调试作业。调试作业具有运行时间短、占用机时资源少、作业命名随意性强的特点,因此,本文的并行作业特征统计分析主要针对生产型作业。结合实际经验,将调试型作业的运行时长上限设为 30 min。

$$R = \left(1 - \frac{T_{\text{unknown}}}{T_{\text{total_jobs}}} \right) \times 100\% \quad (1)$$

式中:R 为作业记账日志识别率;T_{unknown} 为未识别标记的作业记账日志数;T_{total_jobs} 为总日志数,由 JobCAT 自动统计。

通过对目标集群上的 SLURM 作业记账日志数据库进行分析处理,三个时段的作业记账日志数据识别率如图 4 所示。在三个统计时间段内,对于 183 987、384 680、1 145 393 条并行作业记账日志,日志识别率 R 分别为 98.97%、98.90%、97.11%。这表明 JobCAT 对于百万量级作业记账日志的识别率可达到 95% 以上。同时,从系统中随机选择 20 位用户(约占总用户数的 10%)验证这些用户下的作业记账日志识别正确性,实验随机选择 300 条日志进行人工分析校验,所有日志-应用的关联都是正确的。通过日志-应用关联识别,共有效标记了系统上的 44 个数值模拟业务应用并行程序。在三个统计时段内,未识别作业和调试作业占总机时的总和比例均小于 5%,对于后续基于作业特征统计分析的典型应用筛选影响较小。

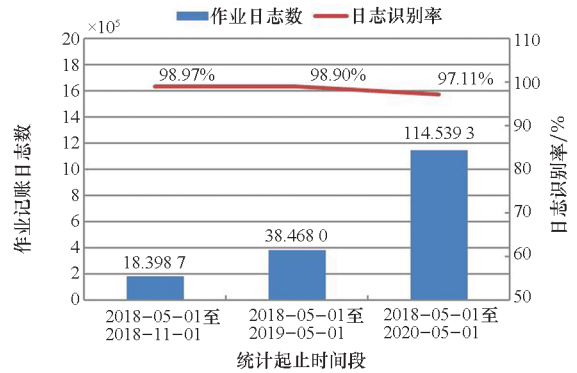


图 4 作业记账日志数据识别率

Fig.4 Recognition rate of job account log

对于未识别的作业,进一步分析发现,这些作业主要由两类组成:一类是基于 JASMIN 框架编写的程序,其可执行程序缺省命名为 main2d 或者 main3d,开发者没有将其根据实际程序进行命名,故而导致无法进行有效区分;另一类是用户对可执行程序进行了匿名化处理,导致无法将其与具体的应用名称建立关联。

上述实验数据表明,基于关键字模糊匹配的并行作业日志标记方法正确有效。根据文献调研,LLNL 的 XDMoD 对于作业记账日志的识别率大于 90%^[7,13]。相比于 XDMoD,JobCAT 增加了结合领域知识的应用名称关键字重命名映射表,这可能是 JobCAT 作业日志识别率更高的原因。

3.3 作业数据统计分析报告

JobCAT 可以支持一键生成作业数据统计分析报告。报告生成的时间开销与作业记账日志数据量成正比。实验数据表明,对于数十万量级的作业记账日志数据,分析报告生成的时间开销小于 5 min。

作业数据统计分析报告包含总体情况分析、计算资源使用情况分析、应用特征分析、作业数据分析。

总体情况分析。包括资源可用率曲线、资源利用率曲线、作业排队情况,它们可以作为超算资源运维管理效率和资源调度效率的衡量指标。资源可用率是衡量系统故障修复能力的重要指标;资源利用率是衡量资源利用率的重要指标;作业排队情况可作为资源调度效率的参考。

计算资源使用情况分析。JobCAT 按照用户、团队、科研业务方向分析统计区间内的计算使用情况,给出按照用户、团队、应用程序、并行规则、运行时长等多个维度的计算资源使用明细情况,从而帮助用户服务更有针对性和指向性。

应用特征分析。JobCAT 从 9 个维度开展应用程序作业特征分析,定量刻画了实际系统上的应用程序业务特征,进而为研制系统评测程序提供参考依据。

作业数据分析。JobCAT 给出了对内存、IO 的资源使用情况的统计分析结果,可以反映实际业务应用对内存、IO 资源的真实使用情况,从而为系统容量配置规划提供参考依据。

3.4 并行作业特征分析举例

JobCAT 的设计遵循开放可扩展原则。按照作业特征分析场景的需求,开发了 7 个分析插件。本节列举几项最常用的数据分析功能。

3.4.1 系统运维效率指标统计

基于 JobCAT,可以实现系统可用率、资源利用率、计算需求覆盖率的自动统计。三项指标的定义分别如下:

$$R_{avail} = \frac{T_{avail}}{T_{total}} \times 100\% \quad (2)$$

$$R_{util} = \frac{T_{used}}{T_{avail}} \times 100\% \quad (3)$$

$$R_{cov} = \frac{T_{used}}{T_{used} + T_{queue}} \times 100\% \quad (4)$$

$$T_{queue} = \sum t_{queue} \cdot N_{cpu} \quad (5)$$

其中: T_{avail} 为系统实际可用机时, T_{total} 为系统理论可用机时, T_{used} 为作业实际使用机时, T_{queue} 为作业排队等效核时。 T_{avail} 、 T_{total} 、 T_{used} 由作业管理系统

提供, T_{queue} 由 JobCAT 计算生成。 T_{queue} 为计算队列上排队作业的等待时间与作业并行规模的乘积累加和。 R_{avail} 代表系统可用率,用于衡量系统故障运维能力; R_{util} 代表资源利用率可用于衡量资源使用情况,反映用户上机意愿; R_{cov} 代表作业排队率(计算需求覆盖率),可用于衡量资源调度能力。

如果统计区间内的系统可用率低,表明系统故障率高,此时应该加强系统检修;如果统计区间内的资源利用率低,表明此时有较多的闲置计算资源,此时需要加强用户沟通,促进计算资源的充分使用;如果统计区间内的作业排队率高,计算需求覆盖率低,并且资源利用率低,则表明资源调度策略可能存在优化空间。最为理想的状态是:系统可用率、资源利用率、计算需求覆盖率三者均维持在较高的水平。图 5 为某千万亿次系统在一个月内的系统可用率与资源利用率统计。从图 5 可以看出,在统计区间内,系统可用率一直维持在 95% 以上的水平,表明系统运维效率高。但是在 7 月 5 日和 7 月 26 日的资源利用率下降到了 80% 以下,表明在上述区间内用户上机意愿不强,存在优化空间。

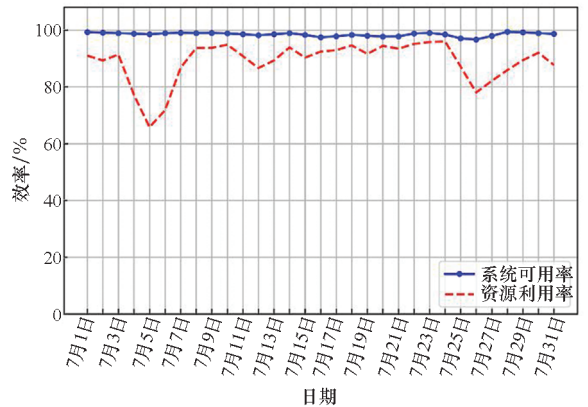


图 5 系统可用率与资源利用率统计
Fig. 5 Statistics of system availability and resource utilization

表 3 为队列计算需求覆盖率统计。从表 3 可见,统计区间内,主要计算队列的计算需求覆盖率均大于 85%,表明系统在统计区间内的资源调度运行良好。

3.4.2 并行应用活跃度统计

依托 JobCAT 对作业记账日志数据从应用维度开展统计分析的能力,JobCAT 可以给出系统上的并行应用程序活跃度统计分析。图 6 和图 7 所示分别是目标系统上的数值模拟应用程序用户数量统计和数值模拟应用程序机时及作业数统计。

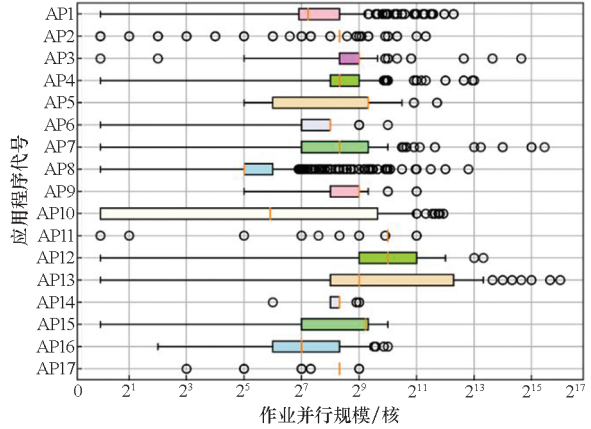
通过上述统计图表,可以量化分析系统上各个应用程序的机时使用情况,并根据程序的用户数量、作业数量、机时数量来综合评估应用活跃度。此外,JobCAT 还可进一步给出应用程序的各个用户对该程序的详细使用情况(作业数、机时数、成功率),从而确定各个应用程序的主要用户,为应用程序的用户支撑服务提供参考。

表3 队列计算需求覆盖率

Tab.3 Coverage ratio of compute demands

队列名称	用户使用机时/(万核·h)	计算需求机时/(万核·h)	计算需求覆盖率/%
ALL	14 681.86	16 843.97	87.16
small	7 858.99	8 940.95	87.90
big	4 704.23	5 365.00	87.68
nice9	1 184.99	1 363.86	86.89
nice4	595.53	698.65	85.24
work	161.32	262.13	61.54
nice1	82.04	94.33	86.96
nice2	69.95	71.64	97.63
interact	20.25	37.57	53.91
nice10	4.45	9.53	46.72
vis	0.11	0.31	34.68

运行特征,图8给出的是某系统上的主力应用程序的并行规模分布情况和各应用程序的单进程平均内存使用情况统计。图8使用箱体图分别展示了应用程序的并行规模分布图 and 应用程序的平均内存使用量。箱体图内的橙色实体代表中位数,箱体图的实体部分代表有50%的样本处于实体区间内,圆圈代表异常样本。上述统计图可以反映各应用程序的作业主要并行规模分布区间、作业的最大并行规模、作业的单进程平均内存使用量等信息。



(a) 作业并行规模分布

(a) Job's parallel processes distribution

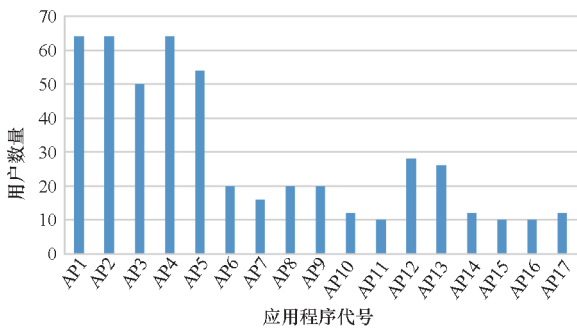


图6 应用程序用户数量统计

Fig.6 Statistics of application users

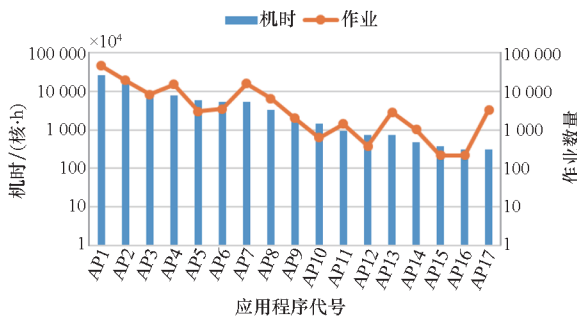
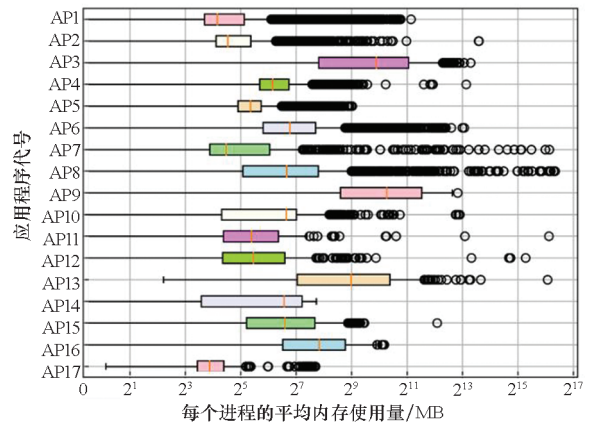


图7 应用程序作业机时与作业数量统计

Fig.7 Statistics of job node hour and quantity



(b) 单进程平均内存使用量

(b) Average memory usage per process

图8 并行应用特征统计分析

Fig.8 Statistical analysis of parallel application characteristics

除此之外,JobCAT 还可给出各程序的单进程最大内存使用量、作业级的内存使用量、IO 读写数据量、作业的运行时长分布、作业运行时长与作业并行规模的关系图等统计数据。这些数据可以从不同角度准确刻画各应用程序在生产环境下对计算资源的使用需求,从而为系统容量规划提供科学参考依据。同时,也可以为并行程序的性能优化提供辅助参考。

3.4.3 应用宏观特征统计分析

JobCAT 可以统计分析并行应用在系统上的

3.4.4 应用负载特征雷达图

基于 JobCAT 对并行应用作业特征的统计分析,可以进一步对并行应用特征开展多维度的量化建模分析研究。图 9 为应用负载特征统计雷达图示例。该图从应用活跃度、作业成功率排名、性能波动率、典型并行规模排名等 13 个维度刻画了并行应用的负载特征,可为典型应用筛选提供依据。关于应用负载特征建模的量化分析已经超出了本文的范畴,这里不做展开。

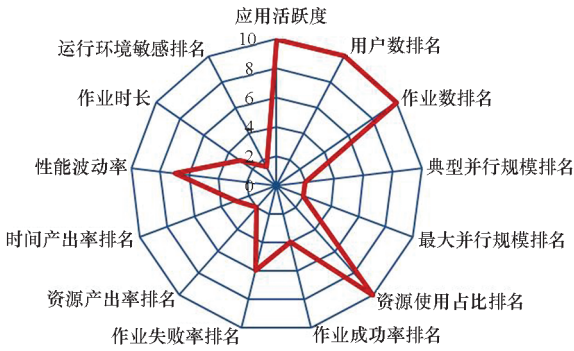


图 9 应用负载特征雷达

Fig. 9 Radar chart of application's workload characteristics

4 结论

基于应用的作业特征分析是高性能计算负载分析研究的重要基础。作业记账日志数据是开展并行作业特征分析的数据源头。针对并行作业特征分析中存在的作业数据识别困难、数据格式不统一、目标场景多样化等问题,本文分别提出了三项关键技术,并集成实现了作业特征分析工具 JobCAT。

通过百万量级作业记账日志数据的测试验证,JobCAT 的作业记账日志标记率大于 95%,这一能力优于 LLNL 的同类工具 XDMoD。JobCAT 支持 7 个插件、29 项统计报表,可一键生成应用的作业特征分析报告,对应用负载分析研究具有实用价值。

参考文献 (References)

[1] DI MARTINO C, KRAMER W, KALBARCZYK Z, et al. Measuring and understanding extreme-scale application resilience: a field study of 5 000 000 HPC application runs[C]// Proceedings of 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 2015: 25 - 36.

[2] SHVETS P, VOEVODIN V, NIKITENKO D. Approach to workload analysis of large HPC centers[C]//Proceedings of International Conference on Parallel Computational Technologies, 2020: 16 - 30.

[3] ANTPAS K, SHALF J, WASSERMAN H. NERSC - 6 workload analysis and benchmark selection process [R]. USA: Department of Energy's Office of Science, 2008.

[4] ANTPAS K, AUSTIN B, BUTLER T, et al. NERSC workload analysis on hopper [R]. USA: Department of Energy's Office of Science, 2013.

[5] AUSTIN B, BHIMJI W, BUTLER T, et al. 2014 NERSC workload analysis[R]. USA: Department of Energy's Office of Science, 2015.

[6] AUSTIN B, BHIMJI W, GERBER R, et al. NERSC - 10 workload analysis on edison and cori (data from 2018) [R]. USA: Department of Energy's Office of Science, 2020.

[7] JONES M D, WHITE J P, INNUS M, et al. Workload analysis of Blue Waters[EB/OL]. (2017 - 03 - 02) [2021 - 10 - 14]. <https://arxiv.org/abs/1703.00924>.

[8] WHITE J P, INNUS M, JONES M D, et al. Challenges of workload analysis on large HPC systems: a case study on NCSA Blue Waters [C]//Proceedings of the Practice and Experience in Advanced Research Computing on Sustainability, Success and Impact, 2017: 1 - 8.

[9] YOU H H, ZHANG H. Comprehensive workload analysis and modeling of a petascale supercomputer[C]//Proceedings of Workshop on Job Scheduling Strategies for Parallel Processing, 2013: 253 - 271.

[10] 黄瑞芳, 刘俊铖, 鲍振华. 气象海洋高性能计算机测评系统研究与实现[J]. 成都信息工程学院学报, 2012, 27(2): 157 - 161.

HUANG R F, LIU J C, BAO Z H. An evaluation system research and implementation for high-performance computer in meteorology and ocean field [J]. Journal of Chengdu University of Information Technology, 2012, 27(2): 157 - 161. (in Chinese)

[11] BAER T, JOHNSON D. Pbsacct: a workload analysis system for PBS-based HPC systems[C]//Proceedings of the Annual Conference on Extreme Science and Engineering Discovery Environment, 2014: 1 - 6.

[12] OLEHOLMNIENSEN. Slurm_tools[EB/OL]. [2021 - 11 - 15]. https://github.com/OleHolmNielsen/Slurm_tools.

[13] PALMER J T, GALLO S M, FURLANI T R, et al. Open XDMoD: a tool for the comprehensive management of high-performance computing resources[J]. Computing in Science & Engineering, 2015, 17(4): 52 - 62.

[14] HPC WIKI. Batch-scheduler. [EB/OL]. [2021 - 11 - 15]. <https://hpc-wiki.info/hpc/Scheduler#General>.

[15] SchedMD. SLURM workload manager [EB/OL]. [2021 - 11 - 15]. <https://slurm.schedmd.com>.