

大型高能物理计算集群资源管理方法的评测

孙震宇 石京燕 姜晓巍 邹佳恒 杜然

(中国科学院高能物理研究所 北京 100049)

摘要 高能物理数据由物理事例组成,事例之间没有相关性。可以通过大量作业同时处理大量不同的数据文件,从而实现高能物理计算任务的并行化,因此高能物理计算是典型的高吞吐量计算场景。高能所计算集群使用开源的TORQUE/Maui进行资源管理及作业调度,并通过将集群资源划分成不同队列以及限制用户最大运行作业数来保证公平性,然而这也导致了集群整体资源利用率非常低下。SLURM和HTCondor都是近年来流行的开源资源管理系统,前者拥有丰富的作业调度策略,后者非常适合高吞吐量计算,二者都能够替代老旧、缺乏维护的TORQUE/Maui,都是管理计算集群资源的可行方案。在SLURM和HTCondor测试集群上模拟大亚湾实验用户的作业提交行为,对SLURM和HTCondor的资源分配行为和效率进行了测试,并与相同作业在高能物理研究所TORQUE/Maui集群上的实际调度结果进行了对比,分析了SLURM及HTCondor的优势和不足,探讨了使用SLURM或HTCondor管理高能物理研究所计算集群的可行性。

关键词 资源管理系统,作业调度器,计算集群,高吞吐量计算,高能物理计算

中图分类号 TP319 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.10.016

Evaluation of Resource Management Methods for Large High Energy Physics Computer Cluster

SUN Zhen-yu SHI Jing-yan JIANG Xiao-wei ZOU Jia-heng DU Ran

(Institute of High Energy Physics, Chinese Academy of Sciences, Beijing 100049, China)

Abstract High energy physics data consist of multiple events, among which there is no relativity. A high energy physics computing mission is parallelized by running multiple jobs processing multiple different data files simultaneously. Therefore, high energy physics computing is a typical high throughput computing scenario. The computer cluster running at the institute of high energy physics (IHEP) uses the open-source TORQUE/Maui for resource management and job scheduling. IHEP keeps a fair-use policy by dividing the computing resources of this cluster into multiple queues, and limiting the maximum number of running jobs of each user. However, this leads up to a low overall resource usage of the cluster. SLURM and HTCondor are both popular open-source resource management system. SLURM has plenty of job scheduling policy, while HTCondor well suits high throughput computing. Both of them are the possible solutions of resource management for computer clusters, replacing old, lack-of-service TORQUE/Maui. In this paper, job submission behavior of users from Daya Bay experiment was simulated at SLURM and HTCondor testing cluster, testing the resource allocation behaviors and efficiencies of SLURM and HTCondor. Their scheduling results were then compared with the actual scheduling result of the same jobs on IHEP TORQUE/Maui cluster. Finally the strengths and weaknesses of SLURM and HTCondor were analyzed, and the practicability of using SLURM or HTCondor to manage the IHEP computer cluster was discussed.

Keywords Resource management system, Job scheduler, Computer cluster, High throughput computing, High energy physics computing

1 背景

计算集群,即互相连接、协同工作的大量计算机组成的整体,在许多方面都可以被视为一台计算机^[1]。计算集群通常由提供计算能力的计算节点、高速互连网络、高性能共享存储

系统、与用户直接交互的节点及资源管理和作业调度系统等部分组成。与单台超级计算机相比,计算集群通常能够提供更高的可靠性、可扩展性以及性价比,因此计算集群是数据中心向用户提供科学计算服务的一种常见解决方案。

资源管理及作业调度系统负责为用户提交的作业分配最

到稿日期:2016-09-14 返修日期:2016-11-24 本文受国家自然科学基金项目(11475210)资助。

孙震宇(1990—),男,博士生,主要研究方向为集群计算,E-mail:sunzy@ihep.ac.cn;石京燕(1972—),女,博士,副研究员,主要研究方向为高性能计算;姜晓巍(1987—),男,硕士,工程师,主要研究方向为集群计算、分布式计算、高性能计算;邹佳恒(1982—),男,博士,副研究员,主要研究方向为高性能计算、调度系统;杜然(1986—),女,博士,助理研究员,主要研究方向为存储系统、调度系统。

合适的计算资源,使作业在最合适的时间运行。资源管理及作业调度系统是计算集群中的核心组件,其性能直接影响到计算集群可扩展的规模及其可管理的作业数量。

高能物理(也称粒子物理),是研究基本粒子及其相互作用的学科。高能物理数据处理的主要过程包括以下3个步骤:

(1)实验数据获取以及模拟数据生成。在线数据获取系统在实验过程中实时采集、筛选并记录探测器产生的实验原始数据(Raw Data),包括探测器中每个粒子的飞行方向、速度、动量等物理量;同时,蒙特卡罗模拟计算,根据物理模型模拟粒子在探测器中的碰撞过程,生成等量或数倍于实验原始数据的模拟数据。

(2)刻度和重建计算。将实验原始数据和模拟数据分别进行拟合,生成重建数据(DST Data)。

(3)物理数据分析计算。利用物理分析工具筛选和分析重建数据,得到物理研究结果。

现代的大型高能物理实验通常会产生海量的实验数据,例如欧洲的大型强子对撞机(LHC)每年会产生大约30PB的数据^[2]。如果没有高效的计算平台,就无法对海量的实验数据进行及时且准确的分析处理,高能物理研究也就无从谈起。高能物理数据由大量物理事例(Event)组成,事例之间没有相关性。物理学家面对海量的数据,可以在事例的粒度上对其任意拆分,并根据需要组成不同尺寸的数据文件。目前,绝大多数高能物理模拟、重建和分析计算都是串行计算,如果将这些计算以作业形式提交到集群运行,则每个作业只需使用一个CPU核心。通过将多个作业分配到多个不同CPU核心上同时运行,处理不同的高能物理数据文件,可实现高能物理数据处理的并行化。因此,高能物理计算符合EGI对高吞吐量计算(High Throughput Computing, HTC)的定义,即更重视大量松耦合作业高效执行的计算模式^[3]。目前在高能物理计算领域,除网格计算、云计算等计算模式外,批处理计算方面仍以单一集群和单一集中式资源管理及作业调度系统为主,其中常用的集中式开源调度系统包括PBS^[4], SLURM^[5], HTCondor^[6]等。

中国科学院高能物理研究所(IHEP)计算中心运行并管理着一套大型本地计算集群。该集群由大约2000台服务器组成,可以提供约1.2万个CPU核心的计算规模;拥有3PB的Lustre共享文件存储系统,用于高能物理数据存储;另外还提供5PB的CASTOR磁带存储系统,用于高能物理数据的长期保存。高能所计算集群为北京谱仪(BES)^[7]、大亚湾中微子实验^[8]、羊八井宇宙线实验^[9]以及在建的江门中微子实验^[10]、高海拔宇宙线观测站(LHAASO)^[11]等大型高能物理实验提供计算和存储服务。

多年来,高能所计算中心使用开源软件TORQUE/Maui管理集群资源,并进行作业调度。集群中的计算节点由各个实验组分别出资购买,并交由计算中心统一管理。根据各实验组的要求,每个实验组购买的服务器只能用来运行本组的作业。为此,计算中心为每个实验组设置了一个或多个专用作业队列。同时,为了保证每个实验组内各个用户之间的公平性,计算中心还对单个用户可以同时运行的最大作业数进

行了限制,以避免出现实验组内的少量用户占用本组全部计算资源的情况。

然而,这些基于计算资源购买经费以及用户公平性的作业队列与调度策略设定经常会降低计算集群的整体资源利用率。我们发现尽管集群有时排队作业总量较大,但整体资源利用率仍然只有50%左右,经常会出现某一实验组的资源比较空闲时其他实验组有大量作业排队的情况,这是由各个实验组的计算任务不同和忙闲时段各异导致的。此外,随着集群队列数量的增加以及计算集群规模的扩大, TORQUE/Maui的作业调度性能大幅下降。而随着使用需求的增加,我们又逐渐发现TORQUE/Maui软件本身也存在一些问题,例如对GPU计算的支持不好、作业队列不能无限增加等,更为严重的是当前TORQUE/Maui社区已经不够活跃,上述问题始终得不到良好解决,从而影响了实际使用效果。

基于此,亟需寻求一套更佳的资源管理及作业调度解决方案,该方案不仅要消除计算资源之间的隔阂,提高整个计算集群的资源利用率,还要保证不增加每个用户的作业平均排队时间。

2 3种资源管理和作业调度器

PBS (Portable Batch System)^[4]是早期较为流行的一种集群资源管理和作业调度系统,最初由NASA开发,并被用于该机构内部的并行计算机。TORQUE^[12]是开源的OpenPBS的一个分支,现由Adaptive Computing公司进行维护,具有良好的规模扩展性和较强的系统容错性,可以支持一定规模的Linux集群系统。TORQUE通常搭配开源的Maui^[13]或其对应的商业版Moab外部调度器实现作业调度功能。开源的TORQUE/Maui组合曾在高能物理计算领域得到广泛应用,但近年来TORQUE/Maui社区活跃度逐渐降低,用户也逐渐转向其他的资源管理和作业调度系统。

PBS搭配Maui使用时,其整体架构如图1所示。用户通过PBS提供的命令访问PBS作业服务器(pbs_server),完成作业提交、作业中止等操作的请求。用户提交的作业会被记录在PBS队列中,由作业服务器进行管理。作业服务器需要调度作业时,会发送请求给Maui调度器;Maui向各个计算节点(MOM)上的pbs_mom进程查询其资源信息,同时向PBS作业服务器查询排队的作业信息;之后根据预设的调度算法,为作业选择合适的节点,并将选择结果返回给PBS作业服务器;PBS作业服务器将作业发送给Maui指定的计算节点,并由该节点上的pbs_mom启动作业。至此,PBS服务器完成一次作业调度。

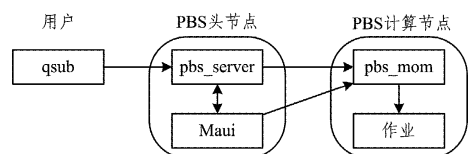


图1 PBS/Maui架构图

SLURM (Simple Linux Utility for Resource Management)^[5]是近年来比较流行的一款开源的Linux集群资源管理系统。SLURM的主要优势同样在于其良好的扩展性和高

度容错性,例如天河-2 超级计算机的 312 万个 CPU 核心就是由 SLURM 管理的。SLURM 除作为资源管理系统外,还具备丰富的作业调度功能,包括简单的先进先出(FIFO)、作业优先级计算、回填算法、群调度(Gang Scheduling)、资源抢占等功能。同时,SLURM 也支持使用类似 Maui/Moab 的外部调度器进行作业调度。SLURM 不仅对 MPICH, Open MPI, BlueGene MPI 等多种不同实现的 MPI 提供良好支持,还支持对图形处理器(GPU)、集成众核(Many Integrated Core, MIC)、网络带宽甚至内存带宽等“通用计算资源”的分配。

SLURM 的整体架构如图 2 所示。与 PBS 类似,SLURM 由一个中央管理器(slurmctld)和分布在各个计算节点上的 slurmd 进程组成。SLURM 使用自己内置的调度器进行作业调度时,通常需要搭配记账服务(slurmdbd)来计算每个用户的 Fair-share 值,并结合作业长度、排队时间等因素计算每个作业的优先级,合适的高优先级作业会被调度到空闲的计算资源上。

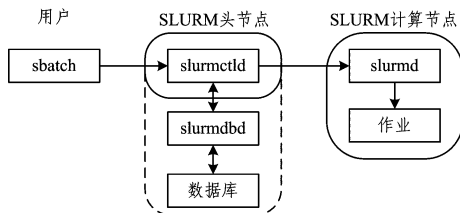


图 2 SLURM 架构图

PBS 的每个计算节点都需要直接与中央管理节点通信,以汇报节点状态、获取作业信息,当集群中的计算节点较多时,PBS 中央管理节点会承受极大的压力。因此,PBS 的架构限制了其支持的最大集群规模。与之不同,SLURM 的计算节点会自行组合成高容错性的树形结构,并以此结构连接其他计算节点进行通信,从而有助于减少中央管理节点同一时刻的连接数,但是在大规模的计算集群中也会带来一定的访问延迟。

HTCondor^[6]是一款非常适合高吞吐量计算的开源资源管理系统。HTCondor 将空闲机器提供的计算资源以及作业对计算资源的需求都抽象为“分类广告”(ClassAds),而 HTCondor 本身的工作便是在资源供给和资源需求这两类广告之间进行匹配^[14]。因此,HTCondor 不仅支持大规模的专用计算集群的管理,还可以利用空闲的非专用计算资源,例如办公用台式机,来处理更多的作业。HTCondor 也支持对用户和作业优先级的计算,使得多个用户能够较为公平地使用同一个 HTCondor 环境。

HTCondor 的整体架构如图 3 所示,其与 PBS/SLURM 完全不同。用户提交到 HTCondor 的作业会被暂存于 condor_schedd 进程所维护的作业队列中,该进程会为排队作业生成描述资源需求的 ClassAd。与此同时,空闲计算节点上运行的 condor_startd 进程也会生成描述计算资源的 ClassAd。这两种 ClassAd 都会发送给头节点上的 condor_collector 进程。condor_negotiator 进程会定期访问 condor_collector,查找计算集群中的空闲资源,并计算每个用户的 Fair-share 值作为优先级,将空闲资源的 ClassAd 优先匹配到高优先级用户的

高优先级作业,并将匹配结果发送给 condor_schedd 进程。之后,condor_schedd 根据匹配结果通知相应计算节点上的 condor_startd 进程,并由其启动相应的作业。同时,condor_schedd 会为每个开始运行的作业启动一个 condor_shadow 进程,该进程负责响应该作业的远程系统调用(Remote System Call)请求,例如文件 I/O 操作。

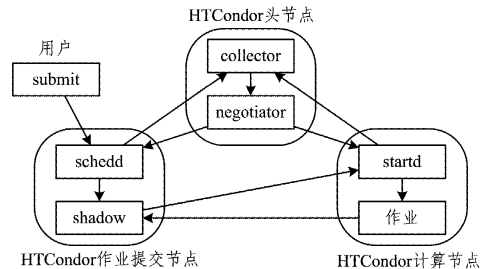


图 3 HTCondor 架构图

不同于 PBS 的划分队列以及 SLURM 的资源分区,HTCondor 将计算集群内所有的计算资源视为一个资源池,其作业队列可以访问这一资源池内的所有可用资源,可以运行在有足够空闲资源的任意节点上。HTCondor 的 ClassAd 机制可以对只使用一个 CPU 核心的作业提供更快速的调度服务。Lahiff 等人的测试表明,HTCondor 在没有调优的情况下也可以轻松地支持 16000 个作业并行运行^[15]。因此,如果一个计算集群专门为以串行单核心作业为主的高能物理作业服务,那么它就非常适合由 HTCondor 进行资源管理。

PBS 和 SLURM 都能够提供从中央控制节点集中管理整个计算集群的功能,便于大规模集群的日常运行和维护管理。而 HTCondor 集群中,各计算节点和作业提交节点自主向中心服务器汇报自身状态,并不听从于中心服务器的统一管理。与 PBS 和 SLURM 相比,HTCondor 的易管理性仍然有所欠缺。根据 HTCondor 的设计思想,中心节点汇报给中心服务器的是分布式的、不可靠的计算资源,中心服务器很可能没有权力“管理”这些资源;计算资源的属主对自己的机器享有完全的控制权,而 HTCondor 则全力减小自己对这一属主的影响^[16]。HTCondor 头节点对其他节点的管理功能仅限于修改部分 ClassAd 的基本属性值,而无法任意调配节点信息。

3 3 种调度系统的对比测试

3.1 测试的背景和目的

高能物理研究所计算中心采用 TORQUE/Maui 调度系统管理集群已达十年之久。近几年,随着集群规模的不断扩大,TORQUE/Maui 的性能和功能问题逐渐显露出来,需要寻求一套更佳的资源管理及作业调度解决方案。根据前期调研结果,我们决定将对同一批真实高能物理作业的调度效果作为评判标准,测试 SLURM 和 HTCondor 对同样批次作业的调度情况,并与其在真实 TORQUE/Maui 环境上的调度结果进行比较,研究使用 SLURM 或者 HTCondor 替代 TORQUE/Maui,管理高能所计算集群的可行性。

如前文所述,各物理实验组的资源由本实验组独享。本文分别设置了 SLURM 和 HTCondor 单队列的两个测试集群,以作业在这两个环境中的运行状况作为本次测试的实验

组,而以真实的 TORQUE/Maui 单个实验组的计算资源及作业运行状况作为对照组。

3.2 测试作业的准备

本文选取计算中心 TORQUE/Maui 计算集群中的一部分作业作为测试样例,其是 2015 年 6 月 22 日至 6 月 28 日期间,大亚湾实验资源的繁忙时段内的全部 50163 个真实高能物理作业,共由 19 个用户提交、运行。我们收集并统计了这些作业在真实 TORQUE/Maui 集群上的提交时间、排队及运行时间等调度结果作为另外两个测试集群的对照结果,从而可以保证 TORQUE/Maui 的调度结果的真实性,同时简化整个测试的流程。所有作业均为串行作业,每个作业只使用一个 CPU 核心。分析作业提交时间发现,这批作业有 47549 个作业是分 344 批提交到计算集群上的,每批内的相邻作业的提交时间间隔不超过 1s;其余 2614 个作业是互相独立地提交到集群上的,每个作业与其他作业之间有至少 2s 的明显间隔。因此,我们认为这一批作业可以反映出高能物理用户普遍的作业提交习惯。

设定每项测试的开始时间为起始零点,计算上述每个作业的提交相对时间和实际运行时长,并据此编写程序,负责在不同的指定时刻向测试集群提交作业,作业内容为睡眠(sleep)指定的时间长度。通过这一脚本,分别向 SLURM 和 HTCondor 测试集群提交全部 50163 个 sleep 作业,并收集两个测试集群上的作业调度结果用于后续的分析。选用 sleep 作业作为测试作业,可以保证每个测试环境中相同作业的运行时长一致。另外,sleep 作业不要求计算节点拥有完全相同的计算能力,在整体计算资源紧张的情况下,可以只使用少量专用资源来完成测试。

使用一个特权用户账号从测试零点开始运行程序,向 SLURM 和 HTCondor 测试集群提交每个 sleep 作业。在提交每个作业前,该特权用户会先自动切换到作业对应的提交者身份,再提交对应的作业,以保证每个 sleep 作业仍然属于相应的真实用户,从而模拟出与真实多用户的作业提交完全一致的运行场景,保证了本次 SLURM 和 HTCondor 内置调度器的调度行为和调度效率测试的有效性。

3.3 测试平台的准备

使用真实 TORQUE/Maui 计算集群作为对照测试平台,该集群包含 64 台 8 核或 16 核的刀片服务器,共有 888 个 CPU 核心,最多可供 888 个单核作业同时运行。SLURM 测试集群的计算节点为 28 台服务器,可见的计算资源为 888 个 CPU 线程。HTCondor 测试集群的计算节点为 148 台虚拟机,可见的计算资源为 888 个作业槽(slot)。选用 sleep 作业作为测试作业,因此单位计算资源的计算能力差异对测试结果不会产生影响。

本次测试使用的 SLURM 软件包为 15.08.1 版本,HTCondor 软件包为 8.2.5 版本。测试启用了 SLURM 内置的多元优先级算法,以及 HTCondor 自有的基于用户和作业优先级的调度策略,对每个用户进行记账和优先级计算。除此以外的设置均为 SLURM 和 HTCondor 软件包的默认值。

表 1 汇总、整理了以上对测试的准备工作。

表 1 各个测试环境的对比

角色	对照组	实验组 1	实验组 2
资源管理器	TORQUE	SLURM	HTCondor
作业调度器	Maui	内置优先级 计算工具	内置优先级 计算工具
计算节点	64 台服务器	28 台服务器	148 台虚拟机
资源分配单位	CPU(核心)	CPU(线程)	作业槽
可用资源数量	888	888	888
提交的作业	真实高能物理作业	sleep 作业	sleep 作业

4 测试结果及分析

4.1 资源管理和作业调度的整体情况

根据测试结果绘制统计图,以便直观地描述各个调度器的运行及排队作业数量随时间的变化情况。图 4 给出 50163 个真实作业在 TORQUE/Maui 集群中的实际调度结果;图 5 和图 6 分别给出 SLURM 和 HTCondor 测试集群的作业调度结果。图 4—图 6 中,横轴表示时间,纵轴表示作业数量;深色曲线表示运行作业数,浅色曲线表示排队作业数。

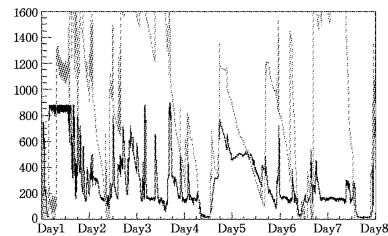


图 4 TORQUE/Maui 集群中的作业运行及排队状况

由图 4 可以很明显地看出,在出现突发大量提交作业的情况时,TORQUE/Maui 只会为一定数量的作业分配相应的资源,之后便不再将空闲资源分配给仍在排队的作业。这是因为在实际的 TORQUE/Maui 集群中,为了保证前文所述的组内和组间公平性,设置了多个队列,并限制了单个用户同时运行的最大作业数,从而造成了资源隔离。具体表现是,某个用户提交大量作业,其同时运行的作业数达到上限后,该用户的剩余作业就只能处于排队状态。

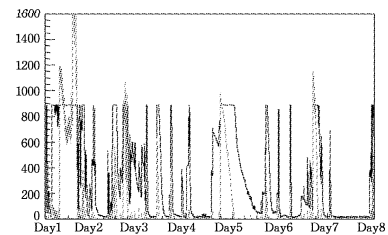


图 5 SLURM 测试集群中的作业运行及排队状况

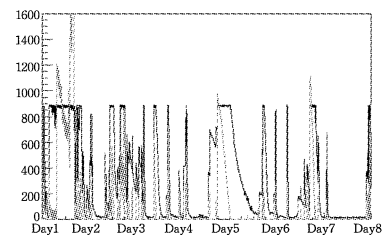


图 6 HTCondor 测试集群中的作业运行及排队状况

图 5 和图 6 表明,在同样出现突发大量提交作业的情况时,SLURM 和 HTCondor 调度器都会把尽量多的计算资源分配给排队的作业,以减少排队作业数量。本次测试中的

SLURM 只有一个队列,且没有对队列可以使用的资源数和用户可以运行的作业数设定限制。而 HTCondor 队列总是可以访问整个资源池,队列中的串行作业可以匹配到资源池内 888 个作业槽中的任何一个空闲作业槽。因此,SLURM 和 HTCondor 都能够充分利用空闲资源。

根据长期运行 TORQUE/Maui 集群积累的经验, TORQUE/Maui 还存在一定的执行效率问题。TORQUE/Maui 在为突发的大量作业分配资源时,同时运行的作业数并不能在第一时间提升到队列和用户允许的最大值,而是在几十分钟到数小时内逐渐平稳地增加到最大值,或是完成所有排队作业的调度,这一现象在图 4 中也可以观察到。而 SLURM 和 HTCondor 在执行效率方面都明显更优,图 5 和图 6 表明,两者为同样的突发大量作业分配资源时,基本都能在几分钟内完成。

4.2 作业排队时间的统计

下面对 3 个不同集群的作业排队时间做进一步分析,并统计出各个集群中所有作业的平均排队时间 $\overline{t_{job}}$ 、每个用户作业平均排队时间的平均值 $\overline{t_{user}}$ 和标准差 $\sigma(t_{user})$,统计结果如表 2 所列。另外,还统计了每个用户的作业最大排队时间。

表 2 作业排队时间的统计

环境	$\overline{t_{job}}$	$\overline{t_{user}}$	$\sigma(t_{user})$
TORQUE/Maui	10754.4	6357.59	10492.1
SLURM	1298.01	722.746	891.645
HTCondor	1286.14	407.766	856.722

从表 2 可以看出,SLURM 和 HTCondor 测试集群中所有作业的平均排队时间、每个用户作业平均排队时间的平均值及标准差都要明显优于 TORQUE/Maui 环境下的相应值。对系统管理员而言,这意味着 SLURM 和 HTCondor 能够提高计算集群的整体作业调度效率以及用户对作业排队时间的整体满意度。

TORQUE/Maui 计算集群中设置了多队列和资源数量使用限制,增加了某些大作业量用户的排队时间,但是可以有效减少小作业量用户的排队时间。由于 SLURM 和 HTCondor 测试集群没有用户作业数量限制,只能依靠降低大作业量用户的优先级来减少小作业量用户的排队时间,因此就每个用户的平均及最大作业排队时间而言,虽然大部分用户在 SLURM 和 HTCondor 测试集群中更优,但仍有个别用户在这两个测试集群中的作业排队时间会比在 TORQUE/Maui 集群中更长。

以用户 xujl 为例,该用户共提交了 9180 个作业,占有用户提交作业总量的近 20%。这些作业在 SLURM 环境下的平均和最大排队时间均约为在 TORQUE/Maui 环境下的 1.6 倍,而在 HTCondor 环境下的平均和最大排队时间则分别达到了 TORQUE/Maui 环境下的 2 倍和 3 倍。进一步分析可知,该用户在测试时段的第 1 天内提交并运行了大量作业,使计算集群整体都处于繁忙状态,这种情况下,SLURM 和 HTCondor 都降低了该用户的优先级,并于其后的几个小时内减少了分配给该用户的资源数,使得其他用户的作业可以尽快开始执行;而 TORQUE/Maui 则允许该用户占用相对更多的计算资源,这便导致了在第 1 天的后半段时间内,集群整体的排队作业数居高不下。

再以用户 yuzy 为例,通过分析发现该用户只在第 4 天晚

上提交了少量的作业,且这些作业在 TORQUE/Maui 和 HTCondor 环境中只排队 10min 左右就开始运行,而在 SLURM 环境中则排队 3h 以上,于整个 SLURM 队列的末尾接受调度。SLURM 的这一行为使得该用户的最大排队时间增加了近 23 倍,与我们优化计算集群的初衷背道而驰。

从集群作业的整体运行效果来看,在个别用户提交大量作业时,将这些用户的排队时间少量延长,以换取其他大多数用户的排队时间的大幅下降是值得的,SLURM 和 HTCondor 在公平性方面并不比 TORQUE/Maui 差,但是处理个别用户的少量作业时,默认配置的 SLURM 仍然存在可以改善的空间。

结束语 SLURM 和 HTCondor 都是近年来流行的开源资源管理系统,都是在高吞吐量的高能物理计算集群中替代 TORQUE/Maui 的可行方案。我们根据大亚湾高能物理实验的真实作业,生成运行时间与之相等的 sleep 作业,利用这些 sleep 作业对 SLURM 和 HTCondor 的集群资源分配、作业调度行为和效率进行了测试,并与高能所 TORQUE/Maui 集群的实际作业调度结果进行了对比。在资源管理器本身的执行效率和大多数用户的排队时间方面,默认配置的 SLURM 和 HTCondor 都明显比 TORQUE/Maui 更优,而在公平性方面 HTCondor 则更胜一筹。

参考上述研究测试结果,高能所已经建立了一个 HTCondor 高吞吐量计算集群,目前还处于试运行阶段。该集群包含约 1300 个 CPU 核心,支持大亚湾和 CMS 两个物理实验。我们将每个实验组的一部分计算节点放入 HTCondor 的公共资源池,这些资源除了优先运行本组的作业以外,空闲时还可以运行其他实验组的排队作业,提高集群的整体资源利用率。对于除共享节点以外的计算节点,我们通过设置其 ClassAd,使之只向相应的实验组提供计算服务,而不能运行其他实验组的作业,以此来满足各个实验组少量但快速的作业运行需要。这一集群已经稳定运行半年,使用效果良好,集群规模将进一步扩大。

本文的测试结果显示,在对传统的高能物理计算作业进行调度时,SLURM 并不是最佳的选择。然而,SLURM 的优势在于其丰富的调度策略,以及对 MPI 和 GPU、MIC 等高性能计算框架和设备的良好支持。高能物理研究所目前也正在建设高性能计算平台,以期看到 SLURM 的优势在此得以发挥。

参考文献

- [1] Building a Beowulf Cluster in just 13 steps [EB/OL]. (2009-05-13)[2016-07-30]. <https://www.linux.com/blog/building-beowulf-cluster-just-13-steps>.
- [2] CERN. Computing [EB/OL]. [2016-07-29]. <https://home.cern/about/computing>.
- [3] European Grid Initiative. Glossary V1 [EB/OL]. (2016-07-19)[2016-07-22]. https://wiki.egi.eu/wiki/Glossary_V1.
- [4] HENDERSON R L. Job scheduling under the Portable Batch System [M]// Job Scheduling Strategies for Parallel Processing. Springer, Berlin, Heidelberg, 1995:279-294.
- [5] YOO A B, JETTE M A, GRONDONA M. SLURM: Simple Linux Utility for Resource Management [M]// Job Scheduling Strategies for Parallel Processing. Springer, Berlin, Heidelberg, 2003:44-60.

- [6] LITZKOW M, LIVNY M, MUTKA M. Condor-A Hunter of Idle Workstations [C]// Proceedings of the 8th International Conference of Distributed Computing Systems. IEEE, 1988; 104-111.
- [7] XU R S, LANG P F, CHEN Y Q, et al. BES Offline Data Processing [J]. High Energy Physics and Nuclear Physics, 1991, 15(7): 577-583. (in Chinese)
许榕生, 郎鹏飞, 陈雅青, 等. 北京谱仪数据的离线处理[J]. 高能物理与核物理, 1991, 15(7): 577-583.
- [8] WANG Y F. A Neutrino Experiment Using the Daya Bay Reactor [J]. Physics, 2007, 36(3): 207-214. (in Chinese)
王贻芳. 大亚湾反应堆中微子实验[J]. 物理, 2007, 36(3): 207-214.
- [9] NIE S M, ZHANG J L, TAN Y H, et al. Real Time Transmission and Analysis of the Yangbajing Cosmic Rays Observation Data [J]. Nuclear Electronics and Detection Technology, 2007, 27(1): 14-17. (in Chinese)
聂思敏, 张吉龙, 谭有恒, 等. 羊八井宇宙线观测数据实时传输及处理系统[J]. 核电子学与探测技术, 2007, 27(1): 14-17.
- [10] 江门中微子实验[EB/OL]. <http://www.ihep.cas.cn/dkxzz/juno>.
- [11] 高海拔宇宙线观测站[EB/OL]. <http://www.ihep.cas.cn/dkxzz/lhaaso>.
- [12] TORQUE Resource Manager-Adaptive Computing[EB/OL]. <http://www.adaptivecomputing.com/products/open-source/torque>.
- [13] Maui-Adaptive Computing [EB/OL]. <http://www.adaptivecomputing.com/products/open-source/maui>.
- [14] RAMAN R, LIVNY M, SOLOMON M. Matchmaking; Distributed Resource Management for High Throughput Computing [C]// Proceedings of the Seventh IEEE International Symposium on High Performance Distributed Computing. Chicago, 1998.
- [15] LAHIFF A, DEWHURST A, KELLY J, et al. HTCondor at the RAL Tier-1 [OL]. <https://indico.cern.ch/event/272785/contributions/1612799>.
- [16] Center for High Throughput Computing, University of Wisconsin-Madison. HTCondor Manual [EB/OL]. http://research.cs.wisc.edu/htcondor/manual/v8.5/3_1Introduction.html.

(上接第 79 页)

小,已成为其不能替代硬盘的主要问题^[14]。Flashcache 等分层混合存储技术应运而生,并且表现出较好的性能。目前随着工艺的逐渐精进,纯固态硬盘设备已经逐渐普及,纯硬盘设备开始慢慢被取代。相信在不久之后,纯固态硬盘设备会拥有较大的存储空间和较低的价格,进而可能完全替代硬盘设备,但就目前而言,Flashcache 等分层混合存储技术仍起着十分重要的作用。

结束语 海量的高能物理实验数据、高并发的大文件访问对文件存储系统有着极高的要求,传统的硬盘设备已经不能很好地满足系统需求。混合存储系统有着较高的性价比,可以为高能物理计算环境提供廉价的高性能存储设备,虽然它有着一定的缺陷,但是针对高能物理计算环境,部署混合存储系统可以带来很高的性能收益。本文对混合存储技术进行了详细的原理分析和数据测试,并对其性能影响因素进行了细致化分析,总结出了混合存储的优化配置公式,对高能物理以外的其他大数据系统也有着借鉴意义。随着存储设备制造工艺的不断改进,纯固态硬盘设备逐渐普及,未来的存储系统架构、存储技术也会不断革新进步,但是目前分层混合存储技术仍有着重要作用。

参 考 文 献

- [1] WLCG-Worldwide LHC Computing Grid[OL]. <http://lcg.web.cern.ch/LCG>.
- [2] CABRERA L, LONG D D E. Swift; Using Distributed Disk Striping to Provide High I/O Data Rates[J]. Computing Systems, 1991, 4(4): 402-441.
- [3] CHENG Y D, SHI J Y, CHEN G. A survey of High Energy Physics Computing System[J]. e-Science Technology & Application, 2014, 5(3): 3-10. (in Chinese)
程耀东, 石京燕, 陈刚. 高能物理计算环境概述[J]. 科研信息化技术与应用, 2014, 5(3): 3-10.
- [4] CHENG Y D, WANG L, HUANG Q L, et al. Design and Optimization of Storage System in HEP Computing Environment [J]. Computer Science, 2015, 42(1): 54-58. (in Chinese)
程耀东, 汪璐, 黄秋兰, 等. 高能物理计算环境中存储系统的设计与优化[J]. 计算机科学, 2015, 42(1): 54-58.
- [5] MITUZAS D. Flashcache at Facebook From 2010 to 2013 and beyond[EB/OL]. [2013-10-9]. <https://www.facebook.com/notes/facebook-engineering>.
- [6] Facebook/Flashcache[OL]. <https://github.com/facebook/flashcache>.
- [7] MORE A, GANJEWAR P. Dynamic Cache Resizing in Flashcache[J]. Advances in Intelligent Systems and Computing, 2015, 327: 537-544.
- [8] 敖青云. 存储技术原理分析: 基于 Linux 2.6 内核源代码[M]. 北京: 电子工业出版社, 2011: 363-476.
- [9] LEE D, CHOI J, KIM J H, et al. On the Existence of a Spectrum of Policies That Subsumes the Least Recently Used (LRU) and Least Frequently Used (LFU) Policies[J]. SIGMETRICS Performance Evaluation Review, 1999, 27(1): 134-143.
- [10] RAMOS L E, GORBATOV E, BIANCHINI R. Page placement in hybrid memory systems. [C]// Proceedings of the international conference on Supercomputing. ACM, 2011: 85-95.
- [11] YANG Z Y. Design and Implementation of Hybrid Storage Scheme Based on Flashcache[D]. Wuhan: Huazhong University of Science and Technology, 2013. (in Chinese)
杨昭宇. 基于 Flashcache 的混合存储方案设计与实现[D]. 武汉: 华中科技大学, 2013.
- [12] Freecode/fio[OL]. <http://freecode.com/projects/fio>.
- [13] FRÜHWIRT P, HUBER M, MULAZZANI M, et al. InnoDB Database Forensics[C]// 24th IEEE International Conference on Advanced Information Networking and Applications (AINA). IEEE, 2010: 1028-1036.
- [14] LIN T J. Principle, Composition and Application of Several Typical Solid State Drives[J]. Computer and External Equipment, 1999(1): 16-21. (in Chinese)
林天静. 几种典型固态盘的原理、组成形式及应用[J]. 电子计算机与外部设备, 1999(1): 16-21.