高性能计算集群实验室的构建与优化研究

孙海松

（南京科技职业学院 信息工程学院，江苏 南京 210048）

摘 要：本文构建了面向计算密集型任务的高性能计算集群实验室（High Performance Computing Cluster Lab，HPCCL），介绍了该集群的主要硬件和软件配置。为了提高集群实验室的管理效率，结合深度强化学习技术，提出了计算作业管理优化方法，即 JoSHH（Job Scheduling for HPCCL）。基于所构建的集群实验室，进行实验评估该方法的性能。实验结果表明，与现有方法相比，该方法减少了计算任务的运行时间。

关键词：高性能计算集群；深度学习；作业管理优化

中图分类号：TP391 **文献标识码：**A **文章编号：**1003-9767（2022）10-008-03

Research on the Construction and Optimization of High-Performance Computing Cluster Laboratory

SUN Haisong

(Department of Information Engineering, Nanjing Polytechnic Institute, Nanjing Jiangsu 210048, China)

Abstract: This paper constructs a High Performance Computing Cluster Lab (HPCCL) for computing intensive tasks, and introduces the main hardware and software configurations of the cluster. In order to improve the management efficiency of the cluster laboratory, combined with the deep reinforcement learning technology, a computational job management optimization method, namely JoSHH (Job Scheduling For HPCCL), is proposed. Based on the cluster laboratory, the performance of this method is evaluated by experiments. Experimental results show that compared with existing methods, this method reduces the running time of computing tasks.

Keywords: high performance computing cluster; deep learning; job management optimization

0 引言

高性能计算（High Performance Computing，HPC）通过耦合计算能力，提供了强大的计算性能，已逐渐成为科学和工程等领域的重要工具[1-2]。高性能计算集群实验室（HPC Cluster Lab，HPCCL）配置有强大算力集群的重要基础设施，能满足用户的 HPC 需求。HPCCL 连接科学计算用户、数据、应用软件、中间件以及集群中心，将它们集成到一个单一的研究环境中。通过提供集成、可用性、效率、共享和协作能力，将超级计算资源转变为一个全国性甚至是全球性的生产环境。本文构建了 HPCCL，介绍了所需的硬件及其相应的型号和配置。为了避免网络通信成为 HPC 的瓶颈，HPCCL 的计算节点之间使用了 InfiniBand 技术进行连接[3]。为提高计算作业的管理效率，本文基本深度学习技术提出了计算作业管理优化方法，即 JoSHH。

1 集群实验室构建

该高性能计算集群实验室能满足来自大学、研究所等机构计算密集型任务的计算需求。该集群包含 256 个计算节点、3 个 I/O 节点和两个管理节点。

管理节点使用了戴尔 Power Edge R450 服务器，是 1U 双插槽服务器。该节点搭载了两个第 3 代英特尔至强 Gold 5318Y 中央处理器，每个处理器的内核数量是 24，基本频率是 2.10 GHz，缓存为 36 MB。节点总内存为 1 TB，由 4 条 256 GB DDR4-3200 内存条组成。节点拥有 8 个 2.5 英寸的固态硬盘，每个硬盘的容量为 2 TB。

I/O 节点使用了戴尔 EMC PowerEdge R550 服务器，是 2U 双路服务器。该节点搭载了两个第 3 代英特尔至强 Platinum 8358P 中央处理器，每个处理器的内核数量是 32，基本频率是 2.60 GHz，缓存为 48 MB。节点总内存为 1 TB，由 4 条

作者简介：孙海松（1977—），男，江苏赣榆人，硕士研究生，实验师。研究方向：计算机软、硬件、云计算技术。

256 GB DDR4–3200 内存条组成。节点拥有 8 个 2.5 英寸的固态硬盘，每个硬盘的容量为 4 TB。

计算节点使用了戴尔 EMC PowerEdge R750xa 服务器，是双插槽 2U 服务器。该节点搭载了两个第 3 代英特尔至强 Platinum 8380 中央处理器，每个处理器的内核数量是 40，基本频率是 2.30 GHz，缓存为 60 MB。节点总内存为 2 TB，由 8 条 256 GB DDR4–3200 内存条组成。节点拥有 6 个 2.5 英寸的固态硬盘，每个硬盘的容量为 8 TB。

所有计算节点都通过 InfiniBand 通信技术连接，交换机为 Mellanox SB7800 交换机，计算节点均配置了 Mellanox ConnectX–4 网卡，以支持高带宽和低时延的应用。其他节点的连接使用了 CloudEngine 8800 系列数据中心交换机。为了提高集群实验室的可用性，为实验配置了不间断电源，其型号为 UPS5000–S–(50–800k)–SM/FM。

节点所使用的操作系统为 CentOS 8.3.2011，集群的管理系统为 Rocks Cluster 7.0。作业管理使用了 OpenPBS Torque 软件以及本文提出的作业调度优化方法。

2 作业管理优化

假设 HPCCL 由 N 个计算节点组成，每个节点具有 R 种资源类型，每个节点具有固定数量的可用资源 S^r，有一个包含了 Q 个等待调度作业的队列。用户将作业提交到等待队列。作业提交包括有关所需硬件资源（中央处理器（Central Processing Unit，CPU）、图形处理器（Graphics Processing Unit，GPU）、内存等）和预期执行时间（最大的实际完成时间）。每个作业 j 的特征包含了资源需求向量 $r_j=(r_{j,1},\cdots,r_{j,R})$、持续时间 t_j 和工作负载类型 w_j。JoSHH 代理负责选择队列中的下一个作业并为其分配资源。作业选择器模块的输入是集群当前状态和等待作业的表示。作业选择器选择作业后，策略选择器将根据作业类型决定如何将资源分配给作业。

2.1 作业选择器

作业选择器使用当前状态的表示作为输入，其中包括集群中调度的作业和队列中等待的作业。为减少状态表示的大小，作业的最大长度不能超过 T。每个长于 T 的作业都由长度为 T 的等效作业表示，当此类作业被调度时，仍然按照原始持续时间运行。新作业临时存储在积压队列中，一旦观察队列中有可用的空闲槽，就会移动到观察队列。为了神经网络创建输入，合并先前定义的集群和队列表示。集群节点沿横轴方向连接在一起，形成大小为 $\boldsymbol{N}\times\boldsymbol{S}^r\times\boldsymbol{T}$ 的 R 个矩阵，与等待作业的表示大小相同。集群和队列在深度维度上合并，创建了大小为 Q+1 的第 3 个维度，其中第一个通道是集群状态，其他通道是队列的槽。为合并不同的资源，本文沿横轴连接资源。此时，作业调度器模块的输入是一个大小为 $(\sum_R \boldsymbol{NS}^r)\times\boldsymbol{T}\times(\boldsymbol{Q}+1)$ 的三维矩阵。

作业选择器的动作空间用 $\{\ ,1,\cdots,i,\cdots,Q\}$ 表示，其中 $a=i$ 表示在第 i 个时隙调度作业，$a=$ 表示代理不在当前时间步中安排任何作业。JoSHH 的代理可以在每个时间步调度多个作业，同时保持动作空间固定。

在作业选择器中，回报 v_t 是未来局部奖励 r_t 和全局奖励 r_g 的组合，即：

$$v_t=\left(\sum_{i=t}^{L}\gamma^{i-t}r_t\right)r_g \tag{1}$$

每个调度动作的局部奖励 r_t 是所有等待作业的等待时间与请求持续时间之比的总和，即：

$$r_t=-\sum_{j=1}^{Q}\left(wt_j/l_j\right)-\sum_{j=1}^{B}\left(wt_j/l_j\right) \tag{2}$$

目标是在调度等待作业时最小化奖励。全局奖励 r_g 可以使用系统调度程序的任何评估指标来计算，对此使用平均归一化周转时间指标 [4]。选择 γ=0.9 作为折扣因子。作业选择器是一个具有卷积层的神经网络，使用 16 个 2×2 过滤器，步幅设置为 1。卷积层之后是 ReLU 激活函数和批量归一化。最后 Softmax 层输出每个动作的预测概率。

2.2 策略选择器模块

策略选择器的输入是工作负载的类型，策略选择器由具有 10 个神经元的隐藏层和一个 ReLU 激活函数组成的浅层全连接神经网络，最后一层的 Softmax 输出两种分配策略（深度优先和广度优先策略）的概率分布作为输出。该模块使用策略梯度进行训练，回报 v_t 仅基于局部动作 a_t 及其奖励值 r_t。

2.3 模型训练

JoSHH 由两个神经网络模块构成。用 θ^1 表示作业选择器的网络参数，用 θ^2 表示策略选择器的网络参数，分别用 Π_{θ^1} 和 Π_{θ^2} 表示作业选择器和策略选择器网络。每个 episode 由固定数量的作业组成，当所有作业完成执行时，一个 episode 便结束了。在每个时间步中，新提交的作业到达并在一个固定长度的队列中等待被调度。无法进入队列时，作业会被保存在积压的先输入先出（First Input First Out，FIFO）队列中。当队列中的一个插槽变为可用时，使用先到先服务策略将积压的作业移动到队列中。所有请求的资源必须从作业开始执行时间到完成连续分配，分配给作业的资源便不再改变。

在每个训练 epoch，为每个作业集运行 M 次蒙特卡罗模拟，以探索不同可能动作的概率空间，并针对每个作业集更新一次神经网络。对于每个时间步 i，收集作业选择器和策略选择器状态、动作和局部奖励 $\{s_i^1,a_i^1,r_i^1\}$ 和 $\{s_i^2,a_i^2,r_i^2\}$。一次 episode 产生全局奖励 r_g，局部和全局奖励分别用于计算作业选择器和策略选择器累积回报 v_t^1 和 v_t^2。最后，每个步骤更新两个网络的梯度信息。

3 实验评估

实验环境有 256 个计算节点，资源类型有 CPU 资源和资源两种，集群的节点拥有两个 CPU 和 4 个 GPU。GPU 实验考虑了计算密集型工作负载。作业选择器记录 T=5 个时间步内的资源使用情况，等待作业的队列大小是 Q=5。

为了评估 JoSHH，将其与两种传统的基于启发式的策略

进行比较：最短作业优先（Shortest Job First，SJF）策略，调度队列中最短的作业，如果多个作业的长度相同，则调度队列中第一个到达的作业；主导资源公平（Dominant Resource Fairness，DRF）策略，选择主导资源请求较小的作业 [5]。

使用平均归一化周转时间作为性能指标。作业的周转时间是其等待的时间，即在队列中花费的时间加上其服务时间。标准化周转时间是周转时间与作业服务时间之间的比率。通过作业的服务时间对周转时间（Turn–Around Time，TAT) 进行规范化可以减少大型作业的影响。

在 5000 个 epoch 内 JoSHH 训练的标准化周转时间、等待时间和奖励对比如表 1 所示。由结果可知，JoSHH 取得最好的标准化周转时间，其值为 3.3。

表 1　标准化周转时间、等待时间和奖励对比

方法名称	标准化周转时间	等待时间 /s	奖励
JoSHH	3.3	11.1	−377.2
SJF	4.5	15.3	−953.3
DRF	6.5	16.2	−1945.2

JoSHH的作业选择器模块的全局奖励是标准化周转时间，根据这个指标，作业选择器将优先安排短工作。不同长度作业集合的平均标准化周转时间和等待时间如图 1 所示。其中，SJF 在处理长作业时会引入较长的等待时间，由于长作业所需的运行时间较长，因此不会对标准化周转时间指标造成较大影响。

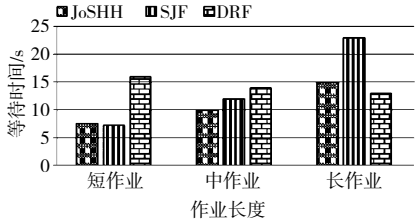

图 1a　等待时间

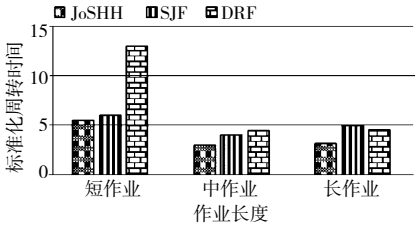

图 1b　标准化周转时间

图 1　不同长度作业的标准化周转时间和等待时间

JoSHH 减少了集群和作业表示的时间范围，因此减少了作业选择器的状态空间，提高了集群优化的可扩展性。JoSHH 和对比策略之间的标准化周转时间比率如图 2 所示。随着集群规模的增加，JoSHH 仍然具有较好的性能，其表现总是优于其他策略。

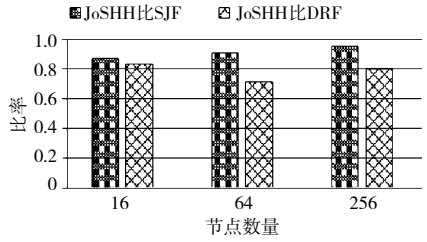

图 2　标准化周转时间比率

4　结　语

本文构建了高性能计算集群实验室，并提出了 JoSHH 方法。利用 JoSHH 观察调度决策的结果，通过使用深度神经网络和强化学习来学习最优的调度策略。在集群中进行实验，实验结果表明，与其他方法相比，JoSHH 能实现最好的性能。未来的工作将结合通信优化，进一步提高集群管理效率。

参考文献

[1]BALAPRAKASH P, DONGARRA J, GAMBLIN T, et al. Autotuning in high-performance computing applications[J]. Proceedings of the IEEE,2018,106(11):2068-2083.

[2]GULO C A S J, SEMENTILLE A C, TAVARES J M R S. Techniques of medical image processing and analysis accelerated by high-performance computing: a systematic literature review[J].Journal of Real-Time Image Processing, 2019,16(6):1891-1908.

[3]YANG S, SON S, CHOI M J, et al.Performance improvement of apache storm using InfiniBand RDMA[J].The Journal of Supercomputing,2019,75(10):6804-6830.

[4]RAHEJA S, ALSHEHRI M, MOHAMED A A, et al.A smart intuitionistic fuzzy-based framework for round-robin short-term scheduler[J].The Journal of Supercomputing, 2022,78(4):4655-4679.

[5]GU J, CHOWDHURY M, SHIN K G, et al.Tiresias: a GPU cluster manager for distributed deep learning[C]//16th USENIX Symposium on Networked Systems Design and Implementation,2019:485-500.