公开的集群负载[24],评估队列算法 Random、非增量算法 Cost scaling(CS)、增量算法 ICS 的运行时间.

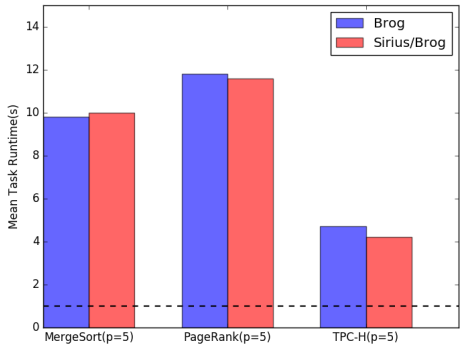

Fig.10 Mean task runtime of different scheduling in the same priority workload

图 10 不同调度器、相同优先级负载下的任务执行时间

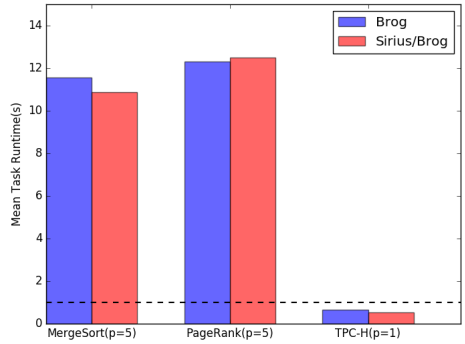

Fig.11 Mean task runtime of different scheduling in different priority workload

图 11 不同调度器不同优先级负载下的平均任务执行时间

实验结果如图 12 和图 13 所示:3 种算法的运行时间随任务数目增加而增加,近似线性相关,其中,CS 算法执延迟远大于增量式算法和队列算法,且随着规模增加,延迟快速增长;队列算法增长速度其次;ICS 算法增长速度最慢.当物理服务器规模达到 10 000 台时,ICS 算法性能优于队列算法.此外,ICS 算法调度延迟相对于 CS 算法最多降低 10 倍.

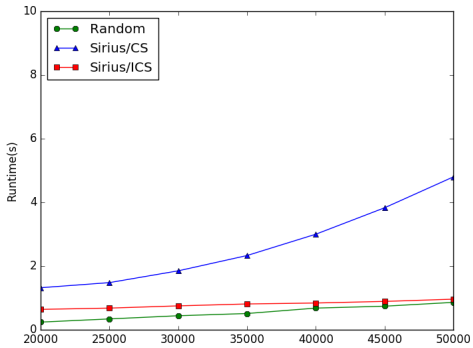

Fig.12 Runtime of different algorithm in 5 000 simulation node

图 12 5 000 个仿真节点下不同算法运行时间

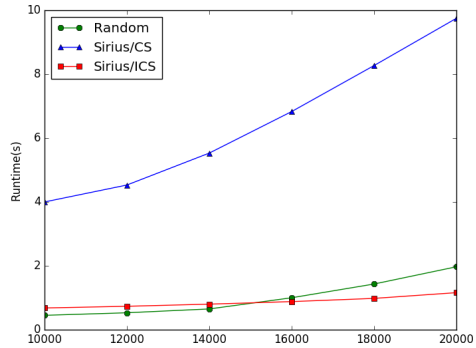

Fig.13 Runtime of different algorithm in 10 000 simulation node

图 13 10 000 个仿真节点下不同算法运行时间

4.3.2 资源开销

本节评估基于队列的调度系统 Random 和基于最小费用最大流的调度系统 Sirius 在不同规模集群下的资源开销,评估指标为 CPU 使用率和内存使用率.采用 Cgroup 把 Sirius 和 Random 限制到单个 CPU 核执行,内存大小限制为 4GB.

实验结果如图 14 和图 15 所示:Sirius CPU 使用率约为 Random 的 1.5 倍,内存使用率约为 Random 的 2 倍,且内存使用率随物理资源规模增长不断增大.网络流图的求解本身是 CPU 敏感型,且增量式求解算法需要缓存上次求解状态,是典型的空间换时间的求解方法,因此,Sirius 在 CPU 和内存方面的资源开销会高于队列模型.我们将在未来的工作中对 Sirius 进行优化,以减少 Sirius 的资源消耗.

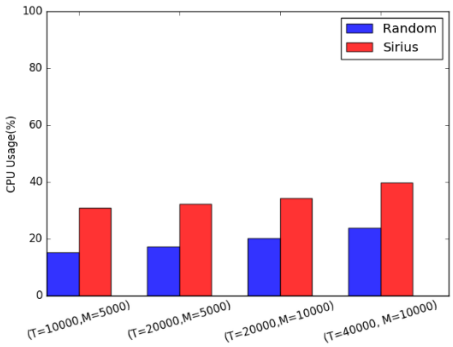

Fig.14 CPU usage on different size cluster

图 14 不同集群规模下的 CPU 使用率

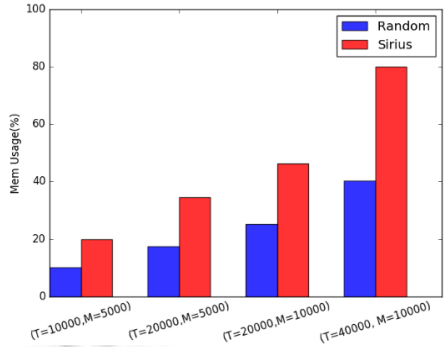

Fig.15 Memory usage on different size cluster

图 15 不同规模下的内存使用率

5 总结与展望

并行作业是大规模资源调度的研究热点.本文提出一种基于最小费用最大流网络的大规模资源调度建模方法,将任务的资源需求和物理资源供给问题转换成最小费用最大流图的构造和求解问题.首先总结相关工作,归纳出公平性、优先级和约束性这 3 种典型的调度目标;接着从资源的视角,将典型调度目标进行描述,并映射到图的构造问题,使用具备适应性调整能力;然后实现了一种最小费用最大流的增量式求解算法,针对图的求解问题进行优化.目前,该方法还存在如下待改进的问题:首先,费用参数赋值主要依赖人工经验,其取值合理性将严重影响方法的效果,如何实现参数的自动化赋值是后续的主要工作之一;其次,图的求解复杂度高,采用合并、过滤等机制简化图的复杂度,加快资源调度决策时效性,是该方法在大规模环境中实用的重要前提.我们将继续围绕参数自动化配置和图的求解优化机制展开研究.

References:

[1] Black DL. Scheduling support for concurrency and parallelism in the Mach operating system. Computer, 1990,23(5):35−43. [doi: 10.1109/2.53353]

[2] Karanasos K, Rao S, Curino C, Douglas C, Chaliparambil K, Fumarola GM, Heddaya S, Ramakrishnan R, Sakalanaga S. Mercury: Hybrid centralized and distributed scheduling in large shared clusters. In: Proc. of the 2015 USENIX Annual Technical Conf. (USENIX ATC 2015). 2015. 485−497.

[3] Verma A, Pedrosa L, Korupolu M, Oppenheimer D, Tune E, Wilkes J. Large-Scale cluster management at Google with Borg. In: Proc. of the 10th European Conf. on Computer Systems. ACM Press, 2015. 18. [doi: 10.1145/2741948.2741964]

[4] Mars J, Tang L. Whare-Map: Heterogeneity in homogeneous warehouse-scale computers. ACM SIGARCH Computer Architecture News, 2013,41(3):619−630. [doi: 10.1145/2485922.2485975]

[5] Tumanov A, Zhu T, Kozuch MA, Harchol-Balter M, Ganger GR. Tetrisched: Space-Time scheduling for heterogeneous datacenters. Technical Report, CMU-PDL- 13-112, Carnegie Mellon University, 2013.

[6] Isard M, Prabhakaran V, Currey J, Wieder U, Talwar K, Goldberg A. Quincy: Fair scheduling for distributed computing clusters. In: Proc. of the ACM SIGOPS 22nd Symp. on Operating Systems Principles. ACM Press, 2009. 261−276. [doi: 10.1145/1629575.1629601]

[7] Vavilapalli VK, Murthy AC, Douglas C, Agarwal S, Konar M, Evans R, Graves T, Lowe J, Shah H, Seth S, Saha B, Curino C, O'Malley O, Radia S, Reed B, Baldeschwieler E. Apache hadoop yarn: Yet another resource negotiator. In: Proc. of the 4th Annual Symp. on Cloud Computing. ACM Press, 2013. [doi: 10.1145/2523616.2523633]

[8] Hindman B, Konwinski A, Zaharia M, Ghodsi A, Joseph AD, Katz R, Shenker S, Stoica I. Mesos: A platform for fine-grained resource sharing in the data center. NSDI, 2011,11: 22−22.

[9] Schwarzkopf M, Konwinski A, Abd-El-Malek M, Wilkes J. Omega: Flexible, scalable schedulers for large compute clusters. In: Proc. of the 8th ACM European Conf. on Computer Systems. ACM Press, 2013. 351−364. [doi: 10.1145/2465351.2465386]

[10] Tumanov A, Cipar J, Kozuch MA, Ganger GR. Alsched: Algebraic scheduling of mixed workloads in heterogeneous clouds. In: Proc. of the 3rd ACM Symp. on Cloud Computing. ACM Press, 2012. [doi: 10.1145/2391229.2391254]

[11] Delimitrou C, Kozyrakis C. Quasar: Resource-Efficient and QoS-aware cluster management. ACM SIGPLAN Notices, 2014,49(4): 127−144. [doi: 10.1145/2541940.2541941]

[12] Zaharia M, Borthakur D, Sarma SJ, Elmeleegy K, Shenker S, Stoica I. Delay scheduling: A simple technique for achieving locality and fairness in cluster scheduling. In: Proc. of the 5th European Conf. on Computer Systems. ACM Press, 2010. 265−278. [doi: 10.1145/1755913.1755940]

[13] Ousterhout K, Wendell P, Zaharia M, Stoica I. Sparrow: Distributed, low latency scheduling. In: Proc. of the 24th ACM Symp. on Operating Systems Principles. ACM Press, 2013. 69−84. [doi: 10.1145/2517349.2522716]

[14] Ghodsi A, Zaharia M, Shenker S, Stoica I. Choosy: Max-Min fair sharing for datacenter jobs with constraints. In: Proc. of the 8th ACM European Conf. on Computer Systems. ACM Press, 2013. 365−378. [doi: 10.1145/2465351.2465387]

[15] Venkataraman S, Panda A, Ananthanarayanan G, Franklin MJ, Stoica I. The power of choice in data-aware cluster scheduling. In: Proc. of the 11th USENIX Symp. on Operating Systems Design and Implementation (OSDI 2014). 2014. 301−316.

[16] Boutin E, Ekanayake J, Lin W, Shi B, Zhou J, Qian ZP, Wu M, Zhou LD. Apollo: Scalable and coordinated scheduling for cloud-scale computing. In: Proc. of the 11th USENIX Symp. on Operating Systems Design and Implementation (OSDI 2014). 2014. 285−300.

[17] Tumanov A, Zhu T, Kozuch MA, Harchol-Balter M, Ganger GR. Tetrisched: Space-Time scheduling for heterogeneous datacenters. Technical Report, CMU-PDL-13-112, Carnegie Mellon University, 2013.

[18] Goder A, Spiridonov A, Wang Y. Bistro: Scheduling data-parallel jobs against live production systems. In: Proc. of the 2015 USENIX Annual Technical Conf. (USENIX ATC 2015). 2015. 459−471.

[19] Delimitrou C, Kozyrakis C. Paragon: QoS-Aware scheduling for heterogeneous datacenters. ACM SIGPLAN Notices, 2013,48(4): 77−88. [doi: 10.1145/2451116.2451125]

[20] Huang XL, Bensaou B. On max-min fairness and scheduling in wireless ad-hoc networks: Analytical framework and implementation. In: Proc. of the 2nd ACM Int'l Symp. on Mobile Ad Hoc Networking & Computing. ACM Press, 2001. 221−231. [doi: 10.1145/501445.501447]

[21] Goldberg AV. An efficient implementation of a scaling minimum-cost flow algorithm. Journal of Algorithms, 1997,22(1):1−29. [doi: 10.1006/jagm.1995.0805]

[22] Dantzig GB. Linear Programming and Extensions. Princeton: Princeton University Press, 1963.

[23] Seref O, Ahuja RK, Orlin JB. Incremental network optimization: Theory and algorithms. Operations Research, 2009,57(3): 586−594. [doi: 10.1287/opre.1080.0607]

[24] Cherkassky BV, Goldberg AV. On implementing the push—Relabel method for the maximum flow problem. Algorithmica, 1997, 19(4):390−410. [doi: 10.1007/PL00009180]

[25] Reiss C, Wilkes J, Hellerstein JL. Google cluster-usage traces: Format + schema. Technical Report, Google Inc., 2011. http://code.google.com/p/googleclusterdata/wiki/TraceVersion2

[26] Firmament. Firmament quincy scheduler. 2015. http://firmament.io

[27] Docker Swarm. Docker swarm filters. 2014. http://github.com/docker/swarm

[28] Kubernetes. Kubernetes kube-schduler. 2014. http://kubernetes.io

陈晓旭(1992－),男,安徽宿州人,硕士,主要研究领域为分布式计算.

陆志刚(1979－),男,高级工程师,主要研究领域为分布式计算.

吴恒(1983－),男,博士,副研究员,CCF 会员,主要研究领域为分布式计算.

张文博(1976－),男,博士,研究员,博士生导师,CCF 高级会员,主要研究领域为分布式计算.

吴悦文(1988－),男,工程师,主要研究领域为分布式计算.