Journal of Computer Applications
计算机应用, 2020, 40(2): 347 - 351
ISSN 1001-9081
CODEN JYIIDU
2020-02-10
http://www.joca.cn

文章编号:1001-9081(2020)02-0347-05
DOI:10.11772/j.issn.1001-9081.2019081366

基于随机森林和遗传算法的Ceph参数自动调优

陈　禹,毛莺池*
(河海大学 计算机与信息学院,南京 211100)
(*通信作者电子邮箱 yingchimao@hhu.edu.cn)

摘　要:Ceph系统性能受Ceph配置参数的显著影响,在Ceph集群的配置优化中,配置参数种类繁多、含义复杂,导致难以实现快速准确寻优。针对以上问题,提出一种基于随机森林(RF)和遗传算法(GA)的参数调优方法,用于自动调整Ceph参数配置以优化Ceph系统性能。该方法使用RF算法为Ceph系统构建性能预测模型,并将预测模型的输出作为GA的输入,通过GA对参数配置方案进行自动迭代优化。仿真结果表明,调优后的参数配置较默认的参数配置相比,使Ceph文件系统的读写性能提高了约1.4倍,并且寻优耗时远低于黑盒参数调优方法。

关键词:Ceph;参数配置;随机森林;遗传算法;自动调优
中图分类号:TP393.06　　**文献标志码**:A

Automatic tuning of Ceph parameters based on random forest and genetic algorithm

CHEN Yu, MAO Yingchi*
(*College of Computer and Information*, *Hohai University*, *Nanjing Jiangsu* 211100, *China*)

Abstract: The performance of Ceph system is significantly affected by the configuration parameters. In the optimization of configuration of Ceph cluster, there are many kinds of configuration parameters with complex meanings, which makes it difficult to achieve fast and accurate optimization. To solve the above problems, a parameter tuning method based on Random Forest (RF) and Genetic Algorithm (GA) was proposed to automatically adjust the Ceph parameter configuration in order to optimize the Ceph system performance. RF algorithm was used to construct a performance prediction model for the Ceph system, and the output of the prediction model was used as the input of GA, then the parameter configuration scheme was automatically and iteratively optimized by using GA. Simulation results show that compared with the system with default parameter configuration, the Ceph file system with optimized parameter configuration has the read and write performance improved by about 1.4 times, and the optimization time is much lower than that of the black box parameter tuning method.

Key words: Ceph; parameter configuration; Random Forest (RF); Genetic Algorithm (GA); automatic tuning

0　引言

Ceph是一个集可靠性、可扩展性、统一性的分布式存储系统,提供对象(Object)、块(Block)及文件系统(File System)三种访问接口,它们都通过底层的LIBRADOS库与后端的对象存储单元(Object Storage Device,OSD)交互,实现数据的存储功能[1]。Ceph内部包含众多模块,模块之间通过队列交换消息,相互协作共同完成IO的处理,典型模块有:网络模块(Messenger)、数据处理模块(FileStore)、日志处理模块(FileJournal)等[2]。面对众多的模块,Ceph提供了丰富的参数配置选项,然而由于Ceph集群的负载情况每时每刻都在变化,加上Ceph作业的类型繁多,不同用户不同作业的需求也千差万别,默认的配置参数往往不能保证集群资源的充分利用和系统的高吞吐率,因此需要调整参数配置从而提高系统在吞吐量、能耗、运行时间等方面的性能[3]。

找到最佳Ceph参数配置的简单方法是尝试配置参数值的每个组合并选择最佳配置参数值,然而在没有深入了解Ceph内部系统和给定应用程序的情况下手动调整这么多的参数非常繁琐且耗时,甚至可能导致性能严重下降;并且Ceph参数配置的最佳性能是应用于特定Ceph应用程序的,因此将针对特定应用程序优化的一组配置应用于不同的Ceph系统会导致性能欠佳;另外大量普通Ceph用户缺乏对Ceph的底层实现原理的理解,对参数优化没有经验,导致运维人员的工作量显著增加。现有的针对存储系统参数调优问题研究已有一定进展,Cao等[4]提出一种黑盒自动调整存储系统参数的方法,迭代尝试不同的配置,测量系统的性能,并根据之前的信息,选择下一次的参数配置。但该方法每次需要运行具有大量输入数据集的应用程序,会耗费大量时间并且会占用大量系统资源。

针对以上问题,本文提出一种基于随机森林(Random Forest, RF)和遗传算法(Genetic Algorithm, GA)参数自动调优的方法,用于自动调整Ceph参数配置,优化Ceph系统性能。该方法通过随机森林算法建立性能预测模型,与需要执行应用程序的方法相比,建立性能模型能够更快地预测应用程序的性能,从而减少系统资源的占用,节省大量测试时间;

收稿日期:2019-07-31;**修回日期**:2019-09-17;**录用日期**:2019-09-23。
基金项目:国家重点研发计划项目(2018YFC0407905);华能集团重点研发课题资助项目(HNKJ17-21)。
作者简介:陈禹(1994—),男,江苏南通人,硕士研究生,主要研究方向:分布式计算、并行处理;　毛莺池(1976—),女,上海人,教授,博士,CCF高级会员,主要研究方向:分布式计算、并行处理、分布式数据管理。

之后将性能预测模型的输出作为遗传算法的输入对Ceph参数配置方案进行自动迭代优化，以找到最佳参数配置。实验结果表明，与默认的Ceph参数配置相比，调优后的参数配置使Ceph文件系统读写性能平均提高了1.4倍，并且寻优耗时远低于黑盒参数调优方法。

1 相关工作

统计推理和机器学习技术[5]已经用于存储系统参数调优的研究，如模拟退火（Simulated Annealing，SA）、贝叶斯优化（Bayesian Optimization，BO）、遗传算法、支持向量机（Support Vector Machine，SVM）、深度Q网络（Deep Q Network，DQN）、随机搜索（Random Search，RS）等被应用到参数自动调优算法中，形成了相应的参数自动调优模型。目前常用的存储系统参数自动调优方法包括基于策略选择的抽样算法、基于HCOpt系统的参数调优、TaskConfigure服务器方法、DAC（Datasize Adjustment of Configuration，一种数据量感知自动调整方法）、自适应调优框架MrEtalon、基于遗传算法的参数调优和黑盒参数自动调优等。黑盒自动调整存储系统参数方法的基本机制是迭代尝试不同的参数配置，测量出目标函数的值，并根据以前学习的信息，选择出下一个需要尝试的参数配置，指导搜索出最优参数配置；但是该方法优化参数配置时每次都需要运行具有大量输入数据集的应用程序，会耗费大量时间并且会占用大量系统资源。文献[6]提出了一种基于HCOpt系统的参数调优，该系统采用了基于遗传算法的参数调优算法，无需分析Hadoop各参数间的制约关系，能够缩短求解时间。HCOpt系统目前针对的是用户手动提交的Map Reduce作业，但实际的应用环境中，有很多的Map Reduce作业是一些高级的系统框架根据用户命令自动生成的，该方法并没有考虑到此问题。文献[7]提出了一个自适应调优框架MrEtalon，可以在短时间内为新工作推荐接近最优的配置。MrEtalon设置了一个配置存储库来提供候选配置，以及一个基于协作过滤的推荐引擎来加速参数的优化；但该方法在生成推荐的过程会延迟作业执行，并导致资源空闲。文献[8]提出了基于作业资源消耗签名的任务分类和基于遗传算法框架的调优系统，系统依据任务的资源消耗模式为不同类型的任务分配适合的配置表，将每一次任务运行作为一次集群测试，将测试结果反馈后用以调优。文献[9]提出的TaskConfigure服务器通过构建Hadoop集群参数信息库系统实现对集群参数的自动调优配置，通过对集群节点及任务的分类，提出集群按类分配配置参数及采用节点资源利用效率生成集群系统参数的优化配置值。文献[10]提出了基于策略选择的抽样算法，在Hadoop中加入了策略感知层，实验结果表明改进的Hadoop框架可以自动优化设置这些复杂的参数，从而提高整个系统的运行效率。这里策略感知层的思想是用一种抽样算法来实现样本感知总体，从而实现根据集群的运行状况和作业的数据特点来动态选择配置参数。但该方法在数据量比较小时，框架的运行效率反而不如以前的框架；而且海量数据的抽样算法属于一种随机抽样，所以样本对总体的估计有很大的随机性。

上述方法能在一定程度上解决存储系统繁琐的参数配置调整问题，推动存储系统参数自动调优的研究工作，但未能考虑存储系统参数的空间巨大性及非线性关系，不能很好地解决Ceph系统的参数调优问题。

2 Ceph配置参数

Ceph数据读写的基本流程：msg模块接收到读写请求，通过ms_fast_dispatch等函数的处理，添加到OSD（Object Storage Device）的op_sharedwq函数中。在op_sharedwq函数的请求下通过线程池的一系列复杂处理，最后调用FileJournal的submit_entry函数同时添加到FileJournal的writeq队列里和completions队列里。FileJournal的write_thread线程以aio的形式写日志数据到磁盘，FileJournal的write_finish_thread以线程检查IO请求是否完成。Finisher的线程把请求添加到FileStore的op_wq队列里，FileStore调用write系统，先将对象数据写入系统的PageCache里，然后再写入XFS文件系统里。当数据内容不足时，也有可能直接刷磁盘，然后向上层返回ack应答读写成功[11]。

上述模块中包含大量的Ceph参数配置，本文选取了这些模块中能够显著影响Ceph文件系统读写性能的一些参数进行研究，具体的参数及其取值范围如表1所示。

表1 Ceph配置参数说明

Tab. 1 Description of Ceph configuration parameters

参数名	数据类型	默认值	范围
filestore max sync interval	integer	5	5~16
filestore min sync interval	float	0.01	0.01~5
filestore op threads	integer	2	1~32
filestore queue max ops	integer	50	16~25 000
osd max write size	integer	90	30~2 000
osd op threads	integer	2	1~32
osd map cache size	integer	200	64~1 024

其中不同的参数取值会对Ceph系统性能产生不同的影响，如将osd_op_threads（处理peering等请求的线程数）、filestore_op_threads（IO线程数）参数设置一个较大值，能够加快IO处理速度，但是如果线程太多，频繁的线程切换也会影响系统性能。为了更直观地了解不同参数配置对Ceph文件系统读写性能产生的影响，对表1中的7个参数取值进行随机分配，形成300组不同的Ceph参数配置，以Ceph文件系统每秒的读写次数（Input/Output Operations Per Second，IOPS）作为性能指标，通过运行相应的Ceph集群，观察系统的性能如何随参数配置而变化。图1显示了当Ceph配置参数发生变化时，系统的读写性能发生显著变化，并且参数配置与性能呈现出非线性关系，有些参数配置在经过调整后反而降低了系统性能。由此可见，手动调整Ceph配置参数非常繁琐且耗时，甚至可能导致性能严重下降。

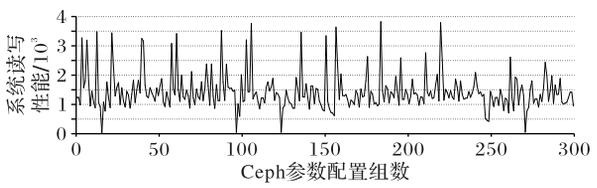


图1 不同参数配置下性能变化

Fig. 1 Performance changes with different parameter configurations

3 模型架构

针对上述问题，本文提出基于随机森林和遗传算法的参数自动调优方法，可以自动调整配置参数的值，以优化给定集群上运行的Ceph应用程序的性能。图2显示了该方法的总体

架构,主要由生成数据、建立模型和参数寻优三个部分组成。

图2　模型总体架构

Fig. 2　Overall architecture of the proposed model

主要步骤如下:

1)在Ceph参数配置范围内,对参数进行随机采样,生成一个参数集$\{conf_1, conf_2, \cdots, onf_n\}$;

2)使用生成的参数集运行相应的Ceph集群,以Ceph文件系统每秒的读写次数作为性能指标,收集相应的性能$\{IOPS_1, IOPS_2, \cdots, IOPS_n\}$;

3)使用步骤1)和2)中的数据集,通过随机森林算法建立性能预测模型;

4)输入一组配置参数的初始值到性能预测模型中,性能预测模型输出相应的IOPS值;

5)将默认的参数配置和相应的IOPS值作为遗传算法输入,执行交叉、变异操作,生成一组新的配置参数,将这些参数再次传入性能预测模型,直到找到最优参数配置。

3.1　生成数据

生成数据模块主要目的是收集不同的参数配置所对应的Ceph文件系统的读写性能,然后,将收集的数据集用于建立性能模型。本文使用随机采样的方法对表1中的参数进行采集,生成参数集$conf_i = \{c_{i1}, c_{i2}, c_{i3}, c_{i4}, c_{i5}, c_{i6}, c_{i7}\}$。$conf_i$是一组参数配置,$c_{in}$是单个的参数取值。将采样的参数集$conf_i$放入Ceph集群中运行,以Ceph文件系统每秒的读写次数作为性能指标,得到相应的性能$IOPS_i$,构造一个向量$S = \{(conf_1, IOPS_1); (conf_2, IOPS_2); \cdots; (conf_n, IOPS_n)\}$存储参数配置集和相应的性能。

3.2　建立模型

考虑到Ceph配置参数以复杂的非线性关系相互作用,本文选取随机森林来为Ceph文件系统构建性能预测模型。随机森林是一种强大的集成模型,是bagging算法的一种扩展,该方法结合了统计推理和机器学习方法的优势[12],基于一组回归或分类树而不是单一树进行预测,并组合各个树的输出以得到最终输出,这使得性能预测不仅准确而且建立的模型更加稳定。此外,该方法对过度拟合具有很强的鲁棒性,并且它没有对预测变量作出任何假设,算法1展示了建立随机森林模型的具体过程。

算法1　随机森林建模过程。

输入　训练集S;训练样本B;

1)　For i = 1 to B {

2)　　T_i=bootstrap sample from S

3)　　$C_x = \{T_i\}_1^B$

4)　}

5)　$Pre(x) = \frac{1}{B}\sum_{i=1}^{B} T_i(x)$

输出　预测性能Pre。

使用向量S作为随机森林的输入,从全部样本中选取大小为B的bootstrap样本,并将它们存储在T_i中;样本特征数目为7,对B个bootstrap样本选择7个特征数目中的k个特征,用建立决策树的方式获得最佳分割点,重复m次,产生m棵决策树,将它们存储到C_x中;通过聚合B个bootstrap样本树的预测来预测新数据Pre_i。

3.3　参数寻优

在建立完性能模型后,仍然不知道最优的参数配置,本文选取遗传算法来搜寻最优参数配置。遗传算法是一种通过模拟自然进化过程搜索最优解的方法[13],其主要特点是:直接对结构对象进行操作,不存在求导和函数连续性的限定;具有内在的隐并行性和更好的全局寻优能力;采用概率化的寻优方法,不需要确定的规则就能自动获取和指导优化的搜索空间,自适应地调整搜索方向。图3描述了参数寻优的基本框架。

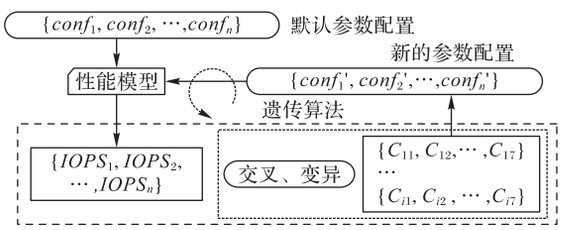


图3　参数寻优框架

Fig. 3　Framework of parameter optimization

首先将默认的参数配置输入到性能模型中,得到对应的读写性能;将默认配置参数及相应的读写性能作为GA的输入,读写性能作为GA的适应度值;GA将默认参数进行交叉、变异,得到一组新的参数配置;将新的参数配置传入性能模型中。重复以上步骤,直到找到最优参数配置。

参数寻优的算法如算法2所示:

算法2　参数寻优。

输入　初始种群$p(0)$,迭代计数器t,交叉发生的概率P_c,变异发生的概率P_m,种群规模M,终止进化的代数T;

1)　初始化P_c、P_m、M、T参数,随机产生第一代种群$p(0)$

2)　do

3)　　计算种群$p(0)$中每一个个体的适应度$f(t)$。

4)　　初始化空种群$p(t)$

5)　　for t = 0 to M

6)　　根据适应度以比例选择算法从种群$p(0)$中选出2个个体

7)　　if (random(0,1)<P_c)

8)　　对2个个体按交叉概率P_c执行交叉操作

9)　　if (random(0,1)<P_m)

10)　　对两个个体按变异概率P_m执行变异操作

11)　　将2个新个体加入种群$p(t)$中

12)　end for

13)　用$p(t)$取代$p(0)$

14)　while t<T

输出　$p(t)$。

算法具体流程如下:

1)初始种群生成。初始种群采用随机的方式生成,设为$p(0)$,设定迭代计数器t = 0,$p(t)$表示第t代种群,$p(0)$中每个个体由表1中7个参数值组合。

2)计算个体适应度值。适应度值用于评价个体的优劣程度,适应度越大个体越好,反之适应度越小则个体越差。这里以Ceph文件系统每秒的读写次数作为遗传算法的适应度值。

3)选择操作。根据适应度的大小对个体进行选择,以保证适应性能好的个体有更多的机会繁殖后代,使优良特性得

以遗传。这里采用的选择算法为“轮盘赌算法”，即个体被选中的概率与其适应度函数值成正比。首先计算所有个体的适应度之和：

$$F = \sum_{t=1}^{M} f(t) \tag{1}$$

然后计算每个染色体被选择的概率：

$$P_i = f(i)/F \tag{2}$$

4）交叉操作。对配置方案参数进行二进制编码，并采取单点交叉方式，随机设置交叉点互换参数信息。

5）变异操作。采用基本位变异的方法，对个体编码串以变异概率 P_m 随机指定某一位或某几位基因进行变异操作。对每一个变异点，将其按位取反或用其他等位基因来替换。操作完成后，对参数编码进行解码，生成新的种群 $p(t)$。

6）更新种群。用交叉、变异后得到的新的种群 $p(t)$ 取代初始种群 $p(0)$。

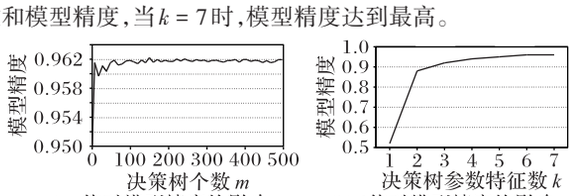
7）终止条件判断。当进化代数达到规定迭代次数 T 时，输出结果；否则转到步骤3）。

本文将一组参数配置 $conf_i = \{c_{i1}, c_{i2}, c_{i3}, c_{i4}, c_{i5}, c_{i6}, c_{i7}\}$ 作为遗传算法中的一条染色体，其中的每一个参数值代表一个基因，Ceph文件系统每秒的读写次数作为遗传算法的适应度值。在遗传算法寻优前，首先要确定 P_c、P_m、M、T 的取值：变异概率 P_m 变异实质上是对参数配置取值空间的深度搜索，变异概率取值太大则会使遗传算法成为随机搜索算法，并且由于随机性太大，GA在搜索上会花费更多的时间[14]，故 P_m 取值为0.01；交叉概率 P_c 影响了配置方案的交替速度，选取较高的交叉概率使算法效率更高，这里取为0.8；种群规模 M 与迭代次数 T 越大，可以增加搜索规模，提高搜索精度，但是太大会增加时间开销，降低搜索的效率，本文将 M 和 T 设置为100。

4 实验结果与分析

4.1 性能模型对比分析

为了利用随机森林算法构造性能模型来自动优化Ceph程序，需要确定3.2节中的两个参数 m 和 k，其中 m 代表决策树的个数，k 是决策树参数的最大特征数。较高的 m 值能使随机森林模型获得较高的精度，但也会导致较长的模型评估时间[15]。本文选取网格搜索的方法来确定 m 和 k 的最优值，图4(a)中横坐标代表决策树的个数，纵坐标代表模型的精度。从整体上看，随着 m 值的增大，模型精度随之上升，当 m = 141时，对模型精度的提高趋于平缓，综合模型的运行成本，选取 m 值为141。图4(b)中横纵坐标分别代表决策数的最大特征数和模型精度，当 k = 7时，模型精度达到最高。



(a) m值对模型精度的影响 (b) k值对模型精度的影响

图4 m 和 k 值对模型精度的影响

Fig. 4 Influence of m and k values on model accuracy

在确定了随机森林的参数之后，从表1的配置参数中随机采样出500组数据，在4台服务器上部署Ceph文件系统读写性能测试集群测试它们对应的读写性能，构成实验数据集 S，其中80%作为训练集，20%作为测试集。图5显示了随机森林建立的性能模型的预测效果，其中横纵坐标分别代表不同的参数配置和Ceph系统性能。观察图5可以发现，随机森林算法建立的性能模型能够很好地预测Ceph文件系统的读写性能，并能及时反映实测值的变化趋势。

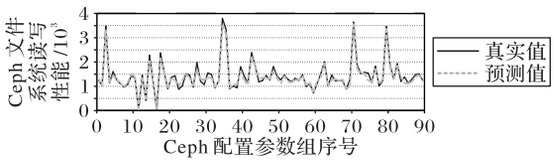


图5 随机森林预测效果图

Fig. 5 Random forest prediction effect map

为了验证随机森林性能模型的优劣性，本文采用了随机森林(RF)、支持向量机(SVM)、人工神经网络(Artificial Neural Network，ANN)和K最近邻(K-Nearest Neighbors，KNN)算法几种机器学习算法分别为Ceph文件系统构建了读写性能模型，并分析比较了这几种性能模型的精度。几种机器学习算法建立的性能预测对比模型如图6所示，其中Real为实际测试值，SVM、KNN、ANN、RF分别代表支持向量机、K最近邻、人工神经网络和随机森林算法建立的性能模型预测值。从整体趋势上看，采用随机森林算法得到的预测值能够及时反映实测值Real的变化趋势，而采用其他算法得到的预测值曲线则与实测值Real曲线存在较明显差异；并且当实测值Real发生骤然改变时，随机森林模型的预测值能够及时跟随Real上升或下降，而其他模型的预测值则存在明显振荡，且严重滞后于实测值的变化趋势。

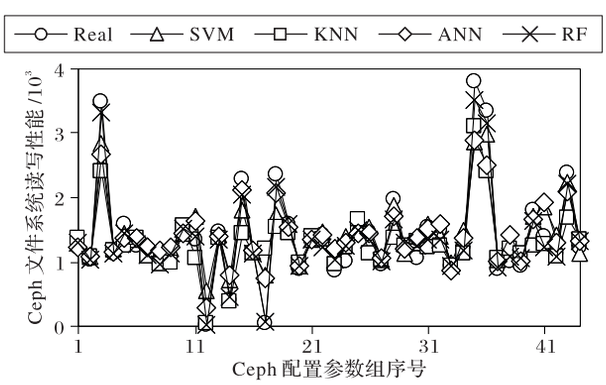


图6 几种模型的综合性能对比

Fig. 6 Performance comparison of several models

为了更直观地比较各模型精度，本文定义预测误差公式为：

$$err = \left(\frac{1}{n} \sum_{i=1}^{n} \left(\left| Pre_i - Real_i \right| / Real_i \right) \right) * 100\% \tag{3}$$

其中：Pre_i 是Ceph文件系统读写性能的预测值，$Real_i$ 是测试的实际性能值。err 反映了性能模型的预测值与实际测试值之间的相对差异，并且越低越好。RF、SVM、ANN、KNN的误差分别为8.6%、20%、27%和49%，由此可见，随机森林算法建立的模型性能要明显优于其他机器学习算法。

4.2 寻优性能对比分析

使用遗传算法迭代寻优的趋势图如图7所示，图中横坐标代表遗传算法的迭代次数，纵坐标代表Ceph文件系统的读写性能。为了确保实验结果的有效性，取5次遗传算法寻优程序运行的平均值作为最终实验结果。观察图7可以看出，遗传算法经过约70次迭代之后，能够达到平稳状态。经过参数优化后的Ceph文件系统读写性能约为3 144，而默认参数配置的性能只能达到1 300，性能提高了约1.4倍。

为了评估本文方法对Ceph参数自动调优的效率，将黑盒

参数调优法与随机森林和遗传算法参数调优法进行对比,图8显示了在不同大小文件下,两种方法搜索出最优参数的耗时对比。观察图8可以看出,使用随机森林和遗传算法优化参数的耗时要明显低于黑盒参数调优的方法;并且随着输入文件的变大,黑盒参数调优法的耗时显著增加,而加入性能模型调优的方法耗时能够稳定在一个较小的值。因为遗传算法每次的迭代只需要对性能模型进行评估,并不需要重新运行Ceph应用程序,而黑盒参数调优法每次需要运行具有大量输入数据集的应用程序,会耗费相当长的时间。

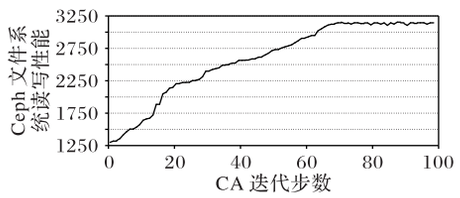


图7　读写性能与GA迭代步数关系

Fig. 7　Relationship between read and write performance and GA iteration step

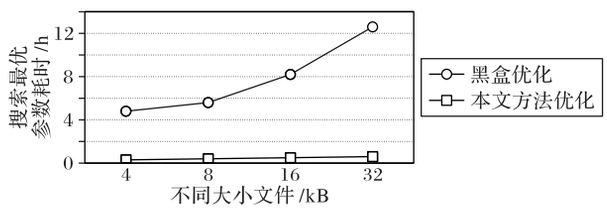


图8　不同大小文件下寻优效率对比

Fig. 8　Comparison of optimization efficiency under different file sizes

5　结语

本文针对Ceph配置参数难以手动调优的问题,提出一种基于随机森林和遗传算法的参数调优方法,用于自动调整Ceph配置参数,以优化Ceph系统性能。利用随机森林建立准确和强大的性能预测模型,将模型的输出作为遗传算法输入以自动搜索Ceph参数配置空间,产生新的参数配置,优化应用程序性能。实验结果表明,经过参数调优后的参数配置与默认参数配置相比,Ceph文件系统读写性能提高了约1.4倍,并且寻优耗时远低于黑盒参数调优方法。在以后的工作中,将继续深入对集群参数的研究,进一步了解参数之间及参数值的设定与系统资源的相关性,并增加参数的研究数量。

参考文献 (References)

[1] HUANG M, LUO L, LI Y, et al. Research on data migration optimization of Ceph[C]// Proceedings of the 14th International Computer Conference on Wavelet Active Media Technology and Information Processing. Piscataway: IEEE, 2017: 83-88.

[2] 李翔. Ceph分布式文件系统的研究及性能测试[D]. 西安:西安电子科技大学, 2014:15-20. (LI X. Research and performance testing of the Ceph distributed file system[D]. Xi'an: Xidian University, 2014:15-20.)

[3] ZHANG X, GADDAM S, CHRONOPOULOS A T. Ceph distributed file system benchmarks on an Openstack cloud[C]// Proceedings of the 2015 IEEE International Conference on Cloud Computing in Emerging Markets. Piscataway: IEEE, 2015: 113-120.

[4] CAO Z, TARASOV V, TIWARI S. Towards better understanding of black-box auto-tuning: a comparative analysis for storage systems[C]// Proceedings of the 2018 Annual USENIX Technical Conference. Berkeley: USENIX Association, 2018: 893-907.

[5] YIGITBASI N, WILLKE T L, LIAO G, et al. Towards machine learning-based auto-tuning of MapReduce[C]// Proceedings of the IEEE 21st International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems. Piscataway: IEEE, 2013:11-20.

[6] 曾林西. 基于性能预估的Hadoop参数自动调优系统[D]. 武汉:华中科技大学, 2013: 34-37. (ZENG X L. A cost-based optimizer for configuration parameters of Hadoop candidate[D]. Wuhan: Huazhong University of Science and Technology, 2013: 34-37.)

[7] CAI L, QI Y, LI J. A Recommendation-based parameter tuning approach for Hadoop[C]// Proceedings of the 2017 IEEE International Symposium on Cloud and Service Computing. Piscataway: IEEE, 2017:223-230.

[8] 马跃,余骋远,于碧辉. 基于资源签名与遗传算法的Hadoop参数自动调优系统[J]. 计算机应用研究, 2017, 34(11):3219-3222, 3228. (MA Y, YU C Y, YU B H. Hadoop parameter automatic tuning system based on resource signature and genetic algorithm[J]. Application Research of Computers, 2017, 34(11): 3219-3222, 3228.)

[9] WU D, WANG Y, FENG H, et al. Optimization design and realization of Ceph storage system based on software defined network[C]// Proceedings of the 13th International Conference on Computational Intelligence and Security. Piscataway: IEEE, 2017:277-281.

[10] 王皎,刘闫锋. Hadoop集群参数的自动调优[J]. 电脑知识与技术, 2012, 8(12):2768-2772. (WANG J, LIU Y F. Parameter auto-tuning of Hadoop clusters[J]. Computer Knowledge and Technology, 2012, 8(12): 2768-2772.)

[11] 刘辉勇,王勇,俸皓. Ceph云存储中基于混合文件系统的读写性能优化方法[J]. 微电子学与计算机, 2018, 35(5):27-34. (LIU H Y, WANG Y, FENG H. Optimization of Ceph reads and writes performance based on hybrid file system[J]. Microelectronics and Computer, 2018, 35(5): 27-34.)

[12] BEI Z, YU Z, ZHANG H, et al. RFHOC: a random-forest approach to auto-tuning Hadoop's configuration[J]. IEEE Transactions on Parallel and Distributed Systems, 2016, 27(5): 1470-1483.

[13] YU Z, BEI Z, QIAN X. Datasize-aware high dimensional configurations auto-tuning of in-memory cluster computing[C]// Proceedings of the 23rd International Conference on Architectural Support for Programming Languages and Operating Systems. New York: ACM, 2018:564-577.

[14] YILDIRIM G, HALLAC İ R, AYDIM G, et al. Running genetic algorithms on Hadoop for solving high dimensional optimization problems[C]// Proceedings of the IEEE 9th International Conference on Application of Information and Communication Technologies. Piscataway: IEEE, 2015: 12-16.

[15] BEI Z, YU Z, LUO N, et al. Configuring in-memory cluster computing using random forest[J]. Future Generation Computer Systems, 2018, 79(1):1-15.

This work is partially supported by the National Key Research and Development Program of China (2018YFC0407905), the Key Research and Development Project of China Huaneng Group (HNKJ17-21).

CHEN Yu, born in 1994, M. S. candidate. His research interests include distributed computing, parallel processing.

MAO Yingchi, born in 1976, Ph. D., professor. Her research interests include distributed computing, parallel processing, distributed data management.