云计算基础设施中的性能瓶颈的识别和优化①

相方莉

(浙江长征职业技术学院 计算机与信息技术系, 杭州 310023)

摘 要: 本文主要介绍了基于云计算平台的性能采样, 分析, 集成工具和方法. 主要针对云计算平台的分布式以及高性能计算特性, 提供了一些性能分析定位的工具, 可以帮助云计算平台开发者和维护人员快速定位性能瓶颈. 通过对瓶颈的分析, 本文还给出一些关于消除瓶颈和优化云计算平台性能的建议.

关键词: 云计算; 性能; 热点; 优化

Performance Bottleneck Identification and Optimization in Cloud Computing Infrastructure

XIANG Fang-Li

(Department of Computer and Information Technology, Zhengjiang Changzheng Technical and Vocational College, Hangzhou 310023, China)

Abstract: In this paper we introduce a method that helps us to identify the bottleneck of a work load based on Cloud Computing Infrastructure. And we also provide some useful tools which could help us to find the problem quickly. After the bottleneck finding, we also give some advices about how to remove the bottleneck and optimize the work-load run.

Key words: performance; hotspots; cloud computing; optimization

1 概述

云计算主要提供一种分布式的计算和存储能力[1], 对于终端用户而言是作为一种服务. 对于消费者而言主要有三种类型的云计算(参见图 1): 软件即服务(SaaS), 平台即服务(PAAS)和基础设施即服务(IaaS)[2].

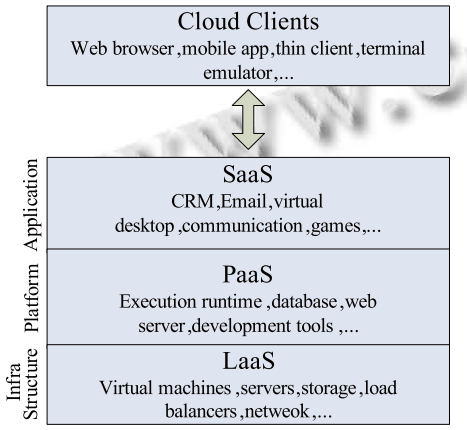

图 1 云计算的三种主要类型

SaaS 提供一种针对最终用户的软件服务, 主要包括传统的应用程序服务, 如会计和电子邮件等. 云计算平台提供相应的软件设施以及基础设施服务. 用户只需掌握通常的应用软件配置设置, 不需要了解底层平台[3].

IaaS 也称为硬件即服务(HaaS), 是由云端的硬件基础设施构成的. 它包括存储, 服务器和网络组件. 基础设施可用于运行软件或简单地存储数据. 服务提供者拥有设备并负责相应的运行和维护, 为最终用户提供简单的租赁服务.

PaaS 称为云计算平台, 出租位于云中的计算等能力, 并提供包括软件的开发和平台部署服务. 这是一种在互联网上出租硬件, 操作系统, 存储和网络容量的方法. 在这个模式中, 消费者通过租用供应商的工具和软件库中的软件来创造软件平台. 消费者可通过工具控制软件的部署和配置设置[4]. Hadoop 就是一个 PaaS.

Hadoop 是一个基于 MapReduce [5]的运行框架,

① 收稿时间:2013-05-01;收到修改稿时间:2013-06-13

Hadoop 通过一种简单的编程模型提供在云计算基础设施中对大型数据集进行分布式处理的能力. Hadoop 可提供从单一服务器到数千台计算机的本地计算和存储能力[6,7].

Hadoop 平台有两个主要部分: Hadoop MapReduce 和 HDFS, 如图 2 所示. Hadoop MapReduce 是一个基于大型集群上的计算节点, 编写并行应用程序的编程模型和软件框架. Hadoop 分布式文件系统是云计算系统的存储层. HDFS 在多个副本计算节点上进行数据块的分配和创建.

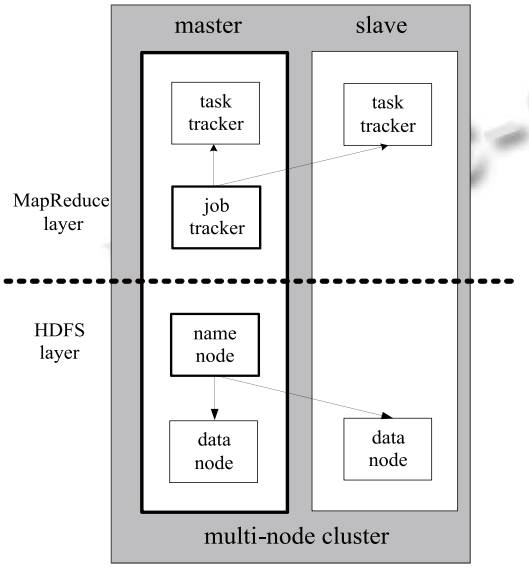

图 2 一个多节点的 Hadoop 集群

在一个 MapReduce 框架中, 一个主节点将输入分成更小的子问题, 并将他们作为工作的节点(Mapper). Mapper 处理问题, 并将答案返回 Reducer. 然后 Reducer 收集所有子节点的处理结果并生成相应的输出. Mapper 和 Reducer 是 MapReduce 框架中的两个主要角色. 在该框架中, 不同角色的性能有不同的衡量标准.

云计算平台的是否合格, 依赖于多种指标: 可用性, 稳定性, 性能等. 不管何种云计算平台, 性能是平台的一个重要指标. 而一个云计算平台的性能可以由完成大量的、有用的、基于云计算基础设施的系统所需使用的时间和资源来比较, 如响应时间, 吞吐量和资源利用. 本文将主要描述云计算平台性能问题的定位和分析,介绍一些方法来识别云计算平台中不同角色和节点的性能瓶颈, 并给出一些建议和方法, 以提高整个平台的性能.

在云计算平台中现在并不存在被广泛采用的性能分析方法和工具. 云计算性能分析工程师往往凭借经验对问题进行定位和优化, 利用单机工具对问题进行剖析. 本文针对不同的角色, 系统化的提供分析和优化思路, 为云计算性能优化提供参考.

2 瓶颈识别

云计算平台存在多种角色, 以上节图 2 中所描述的 Hadoop 平台为例, Master 主要管理和控制整个集群的资源, 而 Slave 主要管理单台机器的资源, 并提供针对本机的服务. Master 角色需要对资源进行动态规划, 并保持和各个 Slave 之间的通讯, 因此对于 Master 来说他是一个 CPU 和网络密集型的应用. 而 Slave 主要是控制单台机器的资源, 根据提供的服务类型可区分为计算型服务和存储型服务.

因此在开始优化工作之前, 首先要做的就是根据不同的角色, 确定角色的工作负载类型, 分而治之是云计算性能优化领域常用的方法.

2.1 工作负载分类

为了找到瓶颈, 首先我们需要进行性能分析. 从系统的角度来看, 性能是软件、数据平台和体系结构三者之间协同的结果. 一个应用程序反映到体系结构上就是一个用于访问 CPU, 内存或 I/O 特定行为. 对一个应用程序来说, 它的性能体现在对上述三个设备的利用上, 当然也会受到上述三个设备的制约, 因此可以根据不同设备对性能的制约, 将性能问题划分为: CPU 型, 内存型和 I/O 型. 具体可以反映到不同的性能指标, 如 CPI, IOPS 等.

为了识别一个瓶颈, 首先我们需要对某个特定的平台有所了解, 必须根据平台的特点进行优化, 才可以提供最大的性能. 类似 Unix 系统, 都会提供的一些工具用来对系统的特定部分进行性能测试, 已提供平台性能标准. 如测试文件系统性能的 IOzone, 测试网络吞吐量的 Netperf, 以及测试 UNIX 平台的 CPU 和测试存储器访问带宽的 Bandwidth 等.

IOzone 是一个测试文件系统性能的工具, 能进行以下 I/O 性能操作: read, write, re-read, re-write, read backwards, read strided, fread, fwrite, random read, pread ,mmap, aio_read, aio_write[8]. Netperf 可用于测量许多不同类型的网络的性能. 它能够提供不定向的吞

吐量测试以及终端到终端的延迟的测试. 目前的 Netperf 能提供以下功能: 通过 BSD 套接字的 TCP 和 UDP 的 IPv4 和 IPv6 测试, DLPI, UNIX 域套接字, IPv4 和 IPv6 的 SCTP[9]. Unixbench 提供一个类似 Unix 系统性能的基本指标[10]. Bandwidth 是主要用于测试 X86 和 X86_64 计算机的内存带宽, 用于识别计算机的内存子系统, 总线架构, Cache 体系结构和处理器本身中可能存在的性能瓶颈[11]. 使用上面提到的工具, 我们就构成一张平台完整的性能图.

当然, 性能优化工作是在某个特定的平台上针对某个特定的应用进行的. 因此我们需要一个有代表性的测试用例. 这个用例必须能够代表真实世界中平台的工作方式, 并具有一定的稳定性, 如 HDFS 的 Terasort. Terasort 本质上是一个顺序的 I/O 测试, 是在多台服务器集群上评估大规模 I/O 处理能力的最好的办法.

当我们有一个代表性的测试案例, 我们需要把它归类为我们之前提到的三种类型(CPU 型, 内存型和 I/O 型). Unix 类系统提供了一些工具来帮助我们快速分类. 顶 Top 命令提供了一个动态实时查看的运行系统. 它可以显示系统的概要信息以及任务的 CPU 和内存使用情况. 通过 TOP 工具帮助, 我们可以发现, 当前的工作负载是受到 CPU 约束还是 I/O 约束. 如果 CPU 使用时间是非常高的, 我们可以认为倾向 CPU 型的工作负载. 否则, 如果 I/O 等待非常高, 我们可以认为这项工作负载比较符合 I/O 绑定标签. 除此之外, 我们还可以使用 iostat 和 netstat 来识别工作负载是属于磁盘 I/O 还是网络 I/O.

2.2 CPU 绑定的工作负载瓶颈识别

一个 CPU 型的工作负载, 首先我们必须确定哪一段应用程序代码区需要消耗最大量的 CPU 时间, 我们称这些代码区域为热点. 为了找到热点, 需要对应用程序的执行过程进行采样分析. 在分析应用程序的执行时, 我们需要对在系统中应用程序如何利用处理器进行采样, 采样方式一般分为 user space sampling 和 event based sampling. 通过采样可以提供应用程序的并行度以及应用程序如何利用处理器资源的相关信息.

user space sampling 和 event based sampling 通过操作系统定时器来中断进程, 收集当前应用程序的上下文(包括指令地址, 调用堆栈)等. 这些采样结果存储在收集的数据文件里. 分析程序可对采样结果进行统计分析, 以帮用户理解程序的执行流程和发现热点.

内存型工作负载和 CPU 型很难区分. 为了找到内存访问模式, 我们可以使用基于事件的硬件采样. 基于硬件事件的采样模式和用户模式的不同在于中断源的不同. 用户模式采样中断源是 OS 定时器, 而基于事件的硬件采样中断源是硬件事件计数器. 例如在 IA32, 我们可以使用跟踪事件来识别远程节点的内存访问比例[12]. OFFCORE_RESPONSE_0.DATA_IN.LOCAL_DRAM and OFFCORE_RESPONSE_0.DATA_IN.REMOTE_DRAM.

远程存储访问率可以计算如下:

Remote Access Ratio %=(OFFCORE_RESPONSE_0.DATA_IN.REMOTE_DRAM/(OFFCORE_RESPONSE_0.DATA_IN.REMOTE_DRAM+OFFCORE_RESPONSE_0.DATA_IN.LOCAL_DRAM))* 100%

不同的工作负载可能有不同的远程访问率. 20-30%的远程访问在某些特定的应用中可以被认为是一种正常的工作负载. 同样 20%远程访问可能在矩阵计算这类高性能计算的应用中却可以认为是不正常的.

VTune 是 intel 开发的一款基于用户空间采样和 EBS 的性能分析工具. 英特尔 VTune 利用硬件性能计数器和 OS 定时器实现采样机制, 并通过 Drawf 信息将代码和采样结果进行可视化展示[13]. Googleperf 也提供了一种基于定时器的 CPU 采样工具. 通过上述工具, 我们可以找到工作负载的热点. 然后根据特定情况进行相应的优化.

2.3 I/O 型工作负载的瓶颈识别

如果用户程序的 I/O 等待时间在整体运行时间中占有非常大的比例, 我们就可以简单的认为此应用程序的行为是 I/O 型. 对这类应用, 首先我们需要确认应用的时间是消耗在等待网络还是磁盘 I/O 上. iostat 和 netstat 可以分别为我们提供磁盘和网络的使用情况. iostat 是用于监测系统的输入/输出设备, 通过观察应用程序执行期间, 设备的利用率变化可以判断应用是否是磁盘型. Netstat 可以对不同的套接字接口进行统计.

对于一个 I/O 型的工作负载, 我们首先需要找到磁盘设备实际吞吐量和最大吞吐量的差距. 这就是为什么我们首先需要确定系统的基准性能. 如果实际吞吐量非常接近基准吞吐量, 那么我们面对的问题将是如何提高磁盘的吞吐量. 例如, 我们的实验平台是 RH2285, 该平台采用六块 1TB 的 Seagate Barracuda ES 2 磁盘作为存储. 该装置通过 IOzone 产生的吞吐量可以在表 1 中找到.

表 1 希捷 1TB 的 Seagate Barracuda ES.2 基准吞吐量

随机读	
Blok Size/KB	吞吐量
4	bw=882KB/s, iops=220
8	bw=1,732KB/s, iops=216
16	bw=3,296KB/s, iops=205
32	bw=6,238KB/s, iops=194
64	bw=11,668KB/s, iops=182
128	bw=20,843KB/s, iops=162
256	bw=34,140KB/s, iops=133
512	bw=54,664KB/s, iops=106
1024	bw=66,285KB/s, iops=64
2048	bw=81,089KB/s, iops=39
4096	bw=88,286KB/s, iops=21
8192	bw=98,626KB/s, iops=12
Sequence Write/2m	bw=104MB/s, iops=52
Sequence Read/2m	bw=152MB/s, iops=76
4k 混合 read/Write	
0/10	bw=983KB/s, iops=245
1/9	bw=103KB/s, iops=25
3/7	bw=271KB/s, iops=67
5/5	bw=447KB/s, iops=111
7/3	bw=576KB/s, iops=144
9/1	bw=758KB/s, iops=189
10/0	bw=851KB/s, iops=212

如果工作负载的 I/O 吞吐量非常接近于表 1 中的数据，我们需要使用一些技术来提高系统的最大吞吐量或减少系统的 I/O 量. 例如，我们有六个磁盘，我们可以做一个 RAID 阵列. 让工作负载通过 RAID 读或写. 不同的 RAID 阵列具有不同的性能和特点. 我们可以选择一个适合工作负载的基于 RAID 阵列的磁盘访问模式.

如果有设备的实际吞吐量和吞吐量基准差距比较大，我们需要找到为什么的 CPU 时间被浪费在等待 I/O 上. 不同的工作负载具有不同的 I/O 访问模式，通过手段平衡 I/O 和计算可以加快应用.

3 性能优化

性能优化是一个重复的工作. 图 3 显示了一个基本性能分析和优化工作流程. 在找到并识别出瓶颈后，我们可以根据以下三个主要的策略，提高应用程序的性能.

3.1 平衡 I/O 与计算

当处理器时间利用率低，I/O 等待时间高时，这是因为应用程序在等待 I/O 完成，那么平衡 I/O 和计算可以加快应用程序的执行. 平衡 I/O 和计算通常是在系统级和应用级调整.

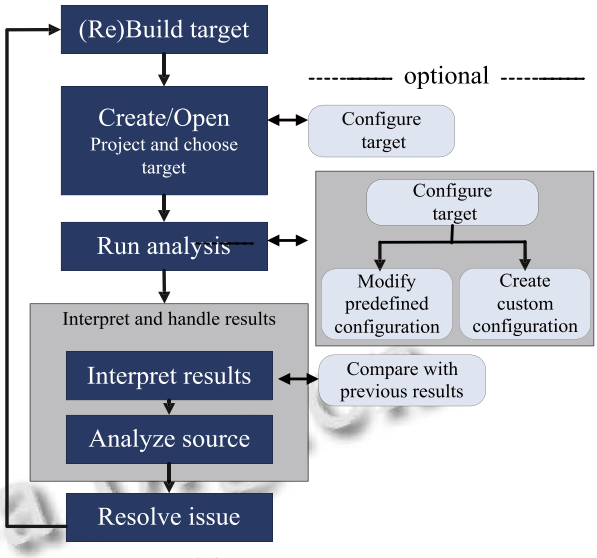

图 3 Perf opt loop

例如，如果 I/O 吞吐量非常接近设备的最大吞吐量，我们可以用压缩的方式压缩我们想要的数据. 采用压缩可以借用 CPU 时间减少 I/O.

特别是现在在云计算平台中，普遍采用的存储设备是普通硬盘. 而普通硬盘的年故障率在 3-5%. 同时硬盘数据存在万分之 5 的错误率，因此往往需要各种校验手段，而校验可以通过处理器计算，也可以用专用校验设备完成，这种平衡之处在云计算平台中很多地方存在.

甚至于为了提高处理器利用率降低由于 IO 延时带来的处理器空转，越来越多的云计算平台开始采用 SSD 作为二级缓存，以加速 IO 的访问速率. 以 Intel s3700 为例，在不同压力下可提供如下 rand write 的 IOPS 性能.

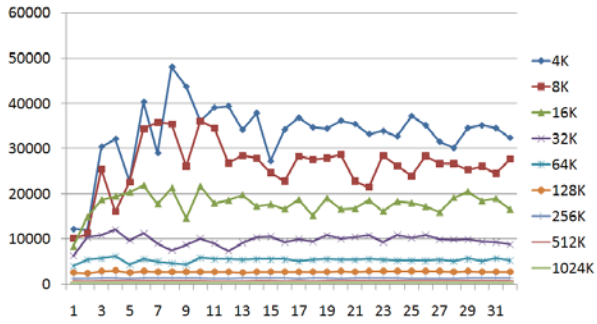

图 4 write, ioengine=psync, iodepth=1 IOPS

3.2 提高线程模型

在应用层面上改变单线程为多线程可以通过更有效地利用可用的处理器资源来提高应用程序的执行速度. 这通常会提高处理器的利用率. 在线程模型，我

们需要处理好同步. 否则, 多线程将是一场噩梦.

线程越多需要的同步也越多, 而云计算平台上还涉及到不同机器之间的同步, 因此如何降低同步代价, 将是提高处理器利用率, 必须考虑的一个问题. 现在一些新的技术比如无锁队列, 事务内存等将会对云计算线程模型性能的提升带来很大的帮助.

3.3 提高计算效率

使用更有效的算法也可以加快你的应用程序. 这种调整包括语言水平的技术如新算法, 栈分配等. 但对一个 I/O 型的工作负载, 如果 I/O 仍然是相同的, 那么我们将看到处理器利用率下降.

处理器利用率的提升, 需要根据计算机的流水线进行优化, 不同的指令在不同的处理器上将会有不同的表现, 特别是对现在广泛采用的云计算平台, 很可能存在异构的机器, 因此必须考虑到不同架构带来的副作用. 比如 Intel Nehalem 和 Sandybridge 的同一条指令 LEA, 却存在不同的 throughput 和 latency, 因此需要根据处理器架构进行优化. 以云计算中存在的 MD5 校验为例, 下面的表格给出了 Sandybridge 平台上优化前后的性能数据对比.

表 2 基于 Sandybrige 的 MD5 指令级优化

ORG	Lea Stall Remove	指令重排	Shld 替换 Roll
252.2 MiB/s	294.9 MiB/s	344.8 MiB/s	367.5 MiB/s

通过 MD5 优化数据对比可见针对特定平台的指令级优化, 可明显提高计算效率.

特别是在云计算平台中普遍采用的编译器版本都是以稳定为主, 而不会去追逐最新的技术, 往往在编译器编译优化过程中不能很好的发挥处理器的性能, 因此也需要手工介入进行处理器层面的优化工作.

4 结论

本文主要介绍了基于云计算平台的性能采样, 分析, 集成工具和方法. 在云计算平台中现在并不存在被广泛采用的性能分析方法和工具. 云计算性能分析工程师主要凭借经验对问题进行定位和优化, 本文针对不同的角色, 系统化的提供分析和优化思路, 为云计算性能优化提供参考.

本文主要针对云计算平台的分布式以及高性能计算特性, 提供了一些性能分析定位的工具, 可以帮助云计算平台开发者和维护人员快速定位性能瓶颈. 通过对瓶颈的分析, 本文还给出一些关于消除瓶颈和优化云计算平台性能的建议. 在云计算中的关键工作都在节点上完成(存储或计算节点). 云工作的工作量将被分到不同的节点上. 提高节点效率是性能改善的根本. 这就是为什么我们专注于节点性能优化的原因. 任何情况下没有总是正确的法则. 在优化过程中, 我们需要根据不同的 CPU 和 I/O 访问模式, 做出判断并选择不同的策略.

参考文献

1 Cloud Computing. Academic Room. http://www.academicroom.com/topics/cloud-computing. Retrieved. 2012-06-16.

2 Peter M, Timothy Grance. The NIST definition of cloud computing. National Institute of Standards and Technology, U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf. Retrieved 28 February 2012.

3 de Haaff B. Cloud computing - the jargon is back! Cloud Computing Journal. http://cloudcomputing.sys-con.com/node/613070. Retrieved 28 February 2012.

4 The NIST definition of cloud computing. National Institute of Science and Technology. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf. Retrieved 24 July 2011.

5 Luo Y, Guo ZH, Sun YM, Plale B, Qiu J, Li W. A hierarchical framework for cross-domain map reduce execution. Indiana Universityfrom University of California, San Diego.

6 The hadoop distributed file system: Architecture and design. D Borthakur-Hadoop Project Website, 2007- cloudcomputing. www.googlecode.com.

7 Shvachko K, Kuang HR, Radia S, Chansler R. The hadoop distributed file system. 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST). 3-7 May 2010

8 http://www.iozone.org/.

9 http://www.netperf.org/netperf/.

10 http://code.google.com/p/byte-unixbench/.

11 http://zsmith.co/bandwidth.html.

12 Intel®64 and IA-32 Architectures Optimization Reference Manual

13 Intel®VTune™Amplifier XE 2011. Document number: 323435-003US