云计算环境下虚拟机部署策略的优化*

郭 涛[a], 刘菲军[b], 杜 垚[b], 陈俊杰[b]
(太原理工大学 a. 信息中心; b. 计算机科学与技术学院, 太原 030024)

摘 要: 针对虚拟机群在云计算平台动态部署过程中现有的部署机制所存在的局限性,提出一种优化的部署模型。首先建立预分块存储的分布式预定制虚拟机全镜像模板库;然后由调度中心依据调度策略选择适当的宿主机,并通过多源并行传输的方式提高部署效率;最后由对应的虚拟机监视器完成实例化任务。实验结果显示,在不同的虚拟机群部署规模下,单台虚拟机的平均部署时间较以往缩短 3% ~45%。实验结果验证了该模型的有效性。

关键词: 云计算; 虚拟机群; 调度优化; 快速部署
中图分类号: TP393.07 **文献标志码**: A **文章编号**: 1001-3695(2012)09-3425-03
doi:10.3969/j.issn.1001-3695.2012.09.060

Virtual machine deployment optimization in cloud

GUO Tao[a], LIU Fei-jun[b], DU Yao[b], CHEN Jun-jie[b]
(*a. Center of Information, b. School of Computer Science & Technology, Taiyuan University of Technology, Taiyuan* 030024, *China*)

Abstract: In order to overcome limitations of deploying virtual cluster in the cloud, this paper designed a optimized deployment model. In the model a distributed full customized VM image library was built in blocks. The VM images were selected, deployed and instantiated according the deployment policy and transmitted from multi-source in parallel, and finally a hypervisor in host completed the process. Experimental results show the single virtual machine's average deployment time is less than 3% ~45% in different VM cluster deployment scales. The method is effective.

Key words: cloud computing; virtual cluster; scheduling optimization; rapid deployment

云计算是一种利用互联网实现随时随地、按需、便捷地访问共享资源池的计算模式[1]。云计算把管理一个计算中心的费用和维护硬件的费用转换到第三方,如亚马逊 EC2[2]。公司或者个人根据其业务负载快速申请或释放资源,只需支付实际资源的使用费用就可以在世界范围内部署所需服务。

美国国家标准与技术研究院将按需自助、快速弹性等定义为云计算的必备特征[3]。虚拟化及其相关技术是目前实现这些特征的主要技术之一[4]。云计算平台需要通过灵活的部署和搭建不同规模与类型的虚拟机群来应对各异的用户需求,从而实现其弹性服务。虚拟机调度是云计算环境下的一个重要机制[5~7]。随着用户数量及业务增长的迅速增长,对大规模、大容量的虚拟机的快速部署提出了更高的要求

1 背景与相关研究

1.1 传统的虚拟机部署策略

传统的调度平台主要由调度中心、宿主机和虚拟机镜像模板库三部分组成,如图 1 所示。

1)调度中心 接收部署请求,根据该请求在机群中选择满足特定条件的目标宿主机。

2)宿主机 承载虚拟机的实例化及运行功能。主要包含宿主机管理、模板传输和实例化等模块。其中虚拟机管理模块用于提供虚拟机的运行及管理服务;模板传输模块用于从虚拟机镜像模板库中获取模板文件至宿主机;实例化模块用于对虚拟机镜像模板进行实例化。

3)虚拟机镜像模板库 用来存储与管理云计算平台预定制的各类型虚拟机模板。

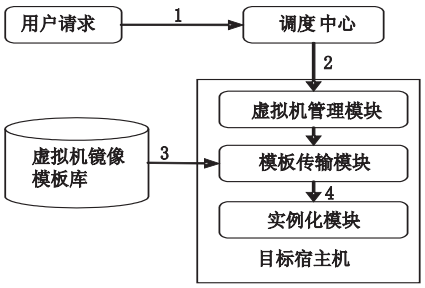

图1 传统的部署模型

其部署流程如下:

a)调度中心接收部署请求;

b)调度中心根据部署请求及约束条件在集群中选择能够满足需求的一台宿主机作为目标主机,并将该请求转给该宿主机的虚拟机管理模块;

c)由宿主机上的模板传输模块从虚拟机镜像模板库中获取虚拟机镜像模板文件;

收稿日期: 2011-12-08; **修回日期**: 2012-02-07 **基金项目**: 山西省科技基础条件平台建设项目(2010091103-0101);山西省研究生优秀创新项目(201003031);山西省国际科技合作项目(2009081022)

作者简介: 郭涛(1978-),男,山西祁县人,博士研究生,主要研究方向为分布式计算、下一代互联网;刘菲军(1986-),男,山西洪洞人,硕士研究生,主要研究方向为分布式计算;杜垚(1984-),男,山西太原人,硕士研究生,主要研究方向为分布式计算;陈俊杰(1956-),男,河北定州人,教授,博导,博士,主要研究方向为智能信息处理、搜索引擎(chenjj@tyut.edu.cn).

d)通过宿主机上的实例化模块将虚拟机镜像模板文件加载并实例化。

1.2　传统的虚拟机部署策略的局限性

传统部署方法存在着如下三个方面的局限性:a)通常一个安装有操作系统和各种应用业务软件的虚拟机模板映像的容量在几个甚至几十个 GB 以上,传输这种大尺寸的映像文件往往需要很长的时间,降低了整体的部署效率;b)在部署含有大量虚拟机的集群时,会导致较高的网络传输开销;c)大多数虚拟机模板库依赖于集中式存储,其往往成为数据吞吐的瓶颈,而且其单点故障也会对平台的整体运行带来关键影响。

1.3　相关研究

针对上述问题,国内外一些学者在如何快速部署虚拟机、优化虚拟机调度效率等方面做了相关研究。文献[8,9]参考 fork 函数的思想,利用父虚拟机迅速克隆出大量子虚拟机,这样可以通过并行创建子虚拟机的方式来提高虚拟机调度效率。文献[6] 给出了一种虚拟机快速克隆方法,利用写时拷贝技术来创建虚拟磁盘和内存状态的快照,然后用按需分配内存技术和多点传送技术来请求和传输这些状态信息。但上述机制依然依赖于集中存储式的架构。文献[10]提出聚合各宿主机的存储空间为一个通用存储池,并将虚拟机镜像模板分别存储在存储池中,通过并行数据传输来提高部署效率。其缺点在于镜像模板缺少冗余存储,当大量部署同一类型的虚拟机时,由于频繁的数据传输会大大增加该存储节点的负载。

2　优化的虚拟机模板部署策略

2.1　部署模型

本文在上述文献的基础上提出的云计算虚拟机调度由五部分组成,如图 2 所示。

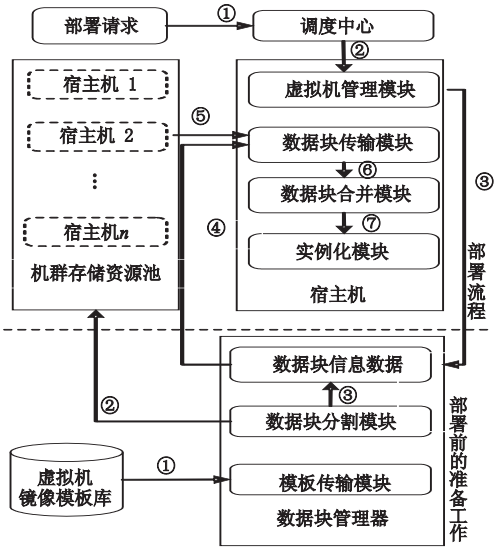

图2　优化的虚拟机调度模型

1)调度中心　接收用户服务请求,根据该请求在服务器集群中选择满足用户需求的一台服务器作为宿主机,并将服务请求转给该宿主机。

2)宿主机　包含虚拟机管理模块、数据块传输模块、数据块合并模块、存储节点负载表和实例化模块。各模块功能如下:硬件配置模块,根据调度中心命令配置宿主机硬件环境;数据块传输模块,从数据块管理服务器上获取模板与数据块的信息,以并发方式从客户机存储资源处存储各数据块;数据块合并模块,将获取的各数据块合并为镜像模板;存储节点负载表,主要包含 Data_path(分块存储位置)、Load_size(该存储节点负载量)、Down_speed(该节点的传输速度)。

3)数据块服务器　包含模板传输模块、数据块分割模块和数据块信息数据库。各模块功能如下:模板传输模块,从虚拟机镜像模板库中传输模板文件;数据块分割模块,将模板文件分割为大小相等的数据块(除最后一个数据块);数据块信息数据库,用于存储模板和数据块的相关信息,主要包括模板分块表和数据块存储表。

4)虚拟机镜像模板库　用来存储云平台所提供的虚拟机镜像模板和用户自己的虚拟机镜像模板。

5)机群存储资源池　将用各宿主机的部分可用存储介质组成共享的通用存储池,用于分布式存储虚拟机模板数据块,作为模板传输源。这种方法有两方面的好处:a)潜在的高扩展性,随着用户数目的增加自动增加虚拟机镜像模板的存储空间;b)它释放了大量的集中存储空间,促进 I/O 性能提高和提供了数据吞吐量。

2.2　部署前的准备工作

部署前的准备工作是数据块管理服务器从虚拟机镜像模板库中读取虚拟机镜像模板,将其分割为若干数据块,并分别部署到各宿主机组成的存储资源池中,同时将模板分割表和数据块存储表等相关信息存储在数据块信息数据库中。

先在数据块管理服务器上安装模板传输模块、数据块分割模块和数据块信息数据库。图 2 中流程如下:

a)模板传输模块从虚拟机镜像模板库中读取虚拟机镜像模板文件。

b)数据块分割模块将虚拟机镜像模板文件分割为若干数据块,同时记录各数据块的起始位置。完成分块任务后,将数据块分布存储在客户机资源池中。例如将一镜像文件 VM1.vd 分割为 VM1_1.vd、VM1_2.vd、VM1_3.vd、VM1_4.vd 和 VM1_5.vd 五个数据块。为服务的稳定性,这里采用冗余存储的方式,例如一台服务器上存储有 VM1_1.vd,同时在其他服务器上存储 VM1_1.vd 的副本文件 VM1_1.vd'。其存储示意如图 3 所示。

$$\begin{pmatrix} \text{server1:} & \text{VM1.1.vd} & \text{VM1.3.vd'} \\ \text{server2:} & \text{VM1.2.vd} & \text{VM1.2.vd'} \\ \text{server3:} & \text{VM1.3.vd} & \text{VM1.4.vd'} \\ \text{server4:} & \text{VM1.4.vd} & \text{VM1.5.vd'} \\ \text{server5:} & \text{VM1.5.vd} & \text{VM1.1.vd'} \end{pmatrix}$$

图 3　数据块分布存储示意图

c)对每一个虚拟机镜像模板分块信息存储在模板分块表中包含 VM_name(模板名称)、Data_num(数据块数目);将数据块的存储信息存储在数据块存储表中,包含 VM_name、Data_sequence(数据块序号)、Data_path(网络存储位置)、Start_pos(起始位置)、End_pos(结束位置)、Storage_num(存有该数据块的节点数目)。模板分块表和数据块存储表均存储在数据块信息数据库中。

2.3　部署流程

a)调度中心收到调度请求。

b)调度中心根据用户请求在服务器集群中选择能够满足用户需求的一台服务器作为宿主机,并将此请求转给该宿主机

的虚拟机管理模块。

c)由宿主机上虚拟机管理模块将用户所需的虚拟机镜像模板信息发送至数据块管理服务器。

d)由数据块服务器根据用户需求首先在模块分块表中查看该模板分块数目 Data_num,然后在数据块存储表中筛选各数据块的存储信息发送至宿主机。

e)由宿主机上的数据块传输模块实现传输任务。其步骤如下:

(a)获取所需模板文件的分块数目 Data_num。建立 Data_num 个线程来并行传输这 Data_num 个数据块。

(b)从数据块存储表中依次读取模板文件的各数据块的 Data_path[i]、Storage_num[i]($i \in [1, \text{Data_num}]$)。同时,应用 HttpURLConnection. getHeaderField(sHeader)函数获得该数据块大小 Data[i]. Length。将该数据块分为 Storage_num[i]个部分。为区别之前提到的数据块,本文将这 Storage_num[i]个部分叫做传输数据段。这些数据段的大小为 Data_splitter,计算方法为

$$\text{Data}[i]_\text{splitter} = \frac{\text{Data}[i].\text{Length}}{\text{Storage_num}[i]} \tag{1}$$

(c)第 i 个数据段的起始位置分别为 Start_pos[i],End_pos[i],其计算式分别为

$$\text{Start_pos}[i][j] = \text{Start_pos}[i][j] + \text{Date}[i]_\text{splitter} + 1 \tag{2}$$

$$\text{End_pos}[i][j] = \text{Start_pos}[i][j] + \text{Date}[i]_\text{splitter} \tag{3}$$

其中:$j \in [1, \text{Data}[i].\text{Length}]$。以上公式中的 i 表示第 i 个数据块,j 表示第 i 个数据块的第 j 个传输数据段。

(d)调用 DownFileSplitter(String dlURL, long Start_pos[i][j],long End_pos[i][j])类来实现这个任务。该类的三个主要参数意义分别如下:dlURL 为传输的网络路径;long Start_pos[i][j]为传输的起始位置;long End_pos[i][j]为传输的结束位置。long Start_pos[i][j]和 long End_pos[i][j]可由(b)中方法获得。dlURL 可由数据块存储表中的 Data_path 得到,在分配时需通过计算表 1 的存储节点负载表中各存储节点负载情况来优化分配。具体优化策略是通过比较各节点 Load_time(负载时间),其计算式为

$$\text{Load_time} = \frac{\text{Load_size}}{\text{DownSpeed}} \tag{4}$$

(e)在确定线程的 Data_path 时,取 Load_time 值最小的存储节点。为避免并行的线程数目过多,严重消耗系统资源,这里采用线程池技术来控制线程的总个数,减少线程的创建次数。

f)当完成各个数据块的传输后,由数据块合并模板将各模板数据块合并为完成的虚拟机镜像模板文件。

g)虚拟机镜像模板实例化,应用实例化指令来完成虚拟机镜像模板文件的实例化。

3　实验结果与分析

为验证上述研究的可用性和有效性,本文进行了相关实验。实验环境由 5 台机架式服务器组成,配置均为 2 颗 AMD Opteron 2378 CPU,共 8 核心,32 GB 内存,512 GB SATA 硬盘。所有服务器通过千兆交换机相连组成一个千兆网络。其中 1 台服务器作为调度中心,1 台服务器作为数据块管理服务器,3 台服务器作为服务器集群宿主机。采用 VMware Server 2 作为虚拟机监视器。

实验中,本文分别对同时部署 1、6、15、30 台规模虚拟机群的情况进行测试,观察部署过程中平均单台虚拟机所花费的部署时间消耗等情况。实验中采用了平均调度策略来部署虚拟机群,即将指定数量的虚拟机平均部署在每台宿主机上。

首先选择该实验环境中的分块策略,将预部署的模板文件(3 GB)按一定大小分割为 N 个数据块。将该 N 个数据块命名为 WT_N_i($i \in [1, N]$)。N 值依次取 3、6、8、10,来测量不同分块策略的传输效率。测试结果如表 2 所示。

表 1　$N = 3$ 时的数据块存储表

模板名	分块名称	存储位置	副本量
WT	WT_3_1	\\HOST1\DS\WT3_3_1	3
		\\HOST2\DS\WT3_3_1	
		\\HOST3\DS\ WT3_3_1	
	WT_3_2	\\HOST1\DS\WT3_3_1	3
		\\HOST2\DS\WT3_3_1	
		\\HOST3\DS\ WT3_3_1	
	WT_3_3	\\HOST1\DS\WT3_3_1	3
		\\HOST2\DS\WT3_3_1	
		\\HOST3\DS\ WT3_3_1	

表 2　不同分块数目传输耗时

分块策略/块	传输耗时/s	分块策略/块	传输耗时/s
3	132	6	160
8	169	10	203

按照传统方法部署不同规模的虚拟机集群,其实验数据如表 3 所示。

表 3　传统方法部署虚拟机群的实验结果

项目	结果			
虚拟机数量/台	1	6	15	30
平均单台传输速度/Mbps	18.6	5.23	1.56	0.85
平均单台传输时间/s	164	583	1 955	3 588
实例化时间/s	72	72	72	72
平均单台调度时间/s	236	655	2 027	4 660

按照优化后的方法部署不同规模的虚拟机集群,其实验数据如表 4 所示。

表 4　优化后方法部署虚拟机群的实验结果

项目	结果			
虚拟机数量/台	1	6	15	30
平均单台传输速度/Mbps	23	6	2.3	1.3
平均单台传输时间/s	132	508	1 326	2 346
平均单台合并时间/s	55	55	55	55
实例化时间/s	72	72	72	72
平均单台调度时间/s	259	635	1 453	2 545

实验结果显示,当部署的虚拟机台数较少时,传统方法由于不需要进行数据块合并的过程,因此调度时间略少于本文所述的方法。但是随着虚拟机部署规模的不断扩大,采用优化后的方法后,单台虚拟机的平均部署时间会出现明显低于以往,节省了 3% ~45% 的部署时间,提高了 14% ~52% 的单台传输速度,如图 4、5 所示。

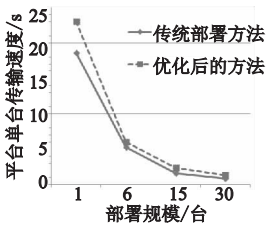

图4　传统方法与优化方法传输速度的对比

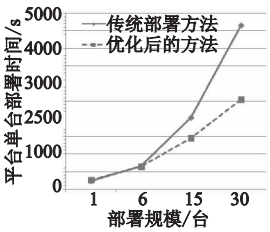

图5　传统方法与优化方法部署时间的对比

(下转第 3431 页)

(上接第 3427 页)

4 结束语

云计算平台的快速弹性特征使其区别是传统的数据中心平台。本文提出了一种优化的虚拟机群快速部署机制,避免了集中式存储架构面临的局限,提高了虚拟机群的弹性部署能力。相关实验验证了其可用性和有效性。目前的实践中,本文针对 VMware Server 虚拟机监视器实现了相关功能,下一步的工作是逐步扩展到 XEN 等更多类型虚拟机监视器,使该模型适用于不同的虚拟环境。

参考文献:

[1] MELL P, GRANCE T. The NIST definition of cloud computing[R]. [S. l.]: National Institute of Standards and Technology,2011.

[2] EC2:amazon elastic compute cloud[EB/OL]. http://aws. amazon. com/ec2/.

[3] MELL J, GRANCE T. The NIST definition of cloud computing(draft) [EB/OL]. [2011-01]. http://csrc. nist. gov/publications/drafts/800-145/Draft-SP-800-145_cloud-definition. pdf.

[4] BUYYA R, YEO C S. Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility[J]. Future Generation Computer Systems,2009,25(2):599-616.

[5] 罗军舟,金嘉晖,宋爱波,等. 云计算:体系架构与关键技术[J]. 通信学报,2011,32(7):31-32.

[6] 郑婷婷,武延军,贺也平. 云计算环境下的虚拟机快速克隆技术[J]. 计算机工程与应用,2011,13(2):101-102.

[7] 殷康. IaaS 公有云平台调度模型研究[J]. 计算机工程与设计,2011,12(6):49-50.

[8] VRABLE M, MA J, CHEN J, *et al*. Scalability, fidelity, and containment in the potemkin virtual honeyfarm[C]//Proc of SOSP. New York: ACM Press,2005:148-162.

[9] ANDR H, LAGAR-CAVILLA S, WHITNEY J A, *et al*. SnowFlock: virtual machine cloning as a first-class cloud primitive[J]. ACM Trans on Computer System,2011,29(1):1-45.

[10] NICOLAE B, BRESNAHAN J, KEAHEY K, *et al*. Going back and forth: efficient multideployment and multisnapshotting on clouds [C]//Proc of HPDC. 2011.