

大数据云中心虚拟机资源高效分配应用研究

余国清, 周兰蓉

(湖南信息职业技术学院 计算机工程学院, 长沙 410200)

摘要: 为降低大数据云中心的能量消耗和实现资源的优化配置, 提出一种虚拟机资源高效分配策略; 提出的策略对选定的特征上具备相似性任务分组的聚类进行定义, 将各组任务映射到定制化的高效虚拟机类型; 其高效指的是以最低限度的资源损耗成功执行任务; 虚拟机的相关参数为核数量、内存量和存储量; 虚拟机分配基于日志中提取的历史数据, 并以任务的使用模式为基础; 提出的资源分配策略以任务的实际资源使用量为基础, 实现了能源消耗的降低; 实验结果表明: 不同聚类任务下, 提出的虚拟机资源分配策略可以大幅节约能源消耗, 具有较低的平均任务拒绝次数。

关键词: 大数据; 资源分配; 虚拟机; 能量消耗; 聚类

Application Research on Efficient Allocation of Virtual Machine Resources in Large Data Cloud Center

Yu Guoqing, Zhou Lanrong

(College of computer engineering, Hunan Information College, Changsha 410200, China)

Abstract: To reduce the energy consumption and optimize the allocation of resources in big data cloud center, a virtual machine resource allocation strategy is proposed. The proposed method defines the clustering of the selected features with similar task grouping, and maps the tasks of each group to the customized efficient virtual machine type. And this efficiency is the successful implementation of tasks with minimal resource depletion. The parameters of virtual machine are the number of cores, memory and storage. The virtual machine is based on the historical data extracted from the log trace, and it is based on the usage pattern of the task. The proposed resource allocation strategy is based on the actual resource usage of the task, and the energy consumption is reduced. The experimental results show that the proposed virtual machine resource allocation strategy can save energy consumption and reduce the average number of tasks in different cases of clustering.

Keywords: big data; resource allocation; virtual machine; energy consumption; clustering

0 引言

大数据云计算^[1]让很大一部分的IT业发生转变, 随着云服务的采用和普及率急速上升, 能量消耗成为云数据中心的主要关注点之一。越来越多的研究着眼于降低云数据中心的能源消耗, 同时满足高效的服务水平要求。虚拟化技术^[2]是能够降低数据中心的能源消耗的一项关键技术。这一技术通过工作负载的迁移与合并, 能够实现高效的资源利用和负载均衡^[3]。由此, 通过将虚拟机迁移出低负载的服务器, 并使这些服务器进入更低的能耗状态, 可以实现相当可观的能源节约。

大量的文献通过基于硬件或基于软件的各种解决方案, 对虚拟化或非虚拟的数据中心中的能量管理进行了研究^[4]。该领域中的大部分研究没有利用对现实云后端的trace的分析中得出的知识, 但没有考虑到云计算中工作负载的变化情况。

如文献^[5]研究了谷歌集群数据中的服务器特征和资源利用情况。还探索了在不同的时间周期中, 对于每个架构类型, 失败、被终止和被驱逐的任务所导致的资源损耗。文献^[6]研究了云数据中心和其他Grid/HPC系统之间的差异, 考

虑到谷歌数据中心中的工作负载和主机负载。该研究证明了较高的作业提交率和较短的作业长度, 会导致云环境中的主机负载的差异变大。由此确定了云与网络之间在工作负载上的主要差别。对于任务聚类, 文献^[7-8]中提出的方法使用K均值算法, 并在一个有限的范围(1~10)内改变类的数量。然后, 通过考虑推导出的聚类中的变化程度和类内平方和推导出最优值的k。其最大缺点是聚类数量必须手动调整。

本文利用从工作负载特征中得出的知识来确定高效的虚拟机配置。对数据中心上的请求进行高效地分配, 降低了基础设施的能源消耗。并考虑到每个集群的使用模式, 以识别虚拟机在CPU、内存和磁盘容量方面的配置。

1 虚拟机类型识别

首先, 对选定的特征上具备相似性任务分组的聚类进行定义; 下一步, 分配能够高效执行属于该聚类的任务的虚拟机类型。此处的高效指的是以最低限度的资源损耗成功执行任务。虚拟机的相关参数为核心数量、内存量和存储量。由于本文中任务所需的存储量较小, 假定分配给虚拟机的磁盘容量为10 GB, 该容量完全能够满足虚拟机上的操作系统安装和任务磁盘使用需求。

1.1 虚拟机的任务数量估计

针对每个虚拟机类型进行任务数量估计所采取的步骤如算法1所示。为避免虚拟机过载^[9], 每个虚拟机中最大任务数量

收稿日期: 2017-04-20; 修回日期: 2017-05-11。

基金项目: 湖南省科学技术厅科技计划项目(2011FJ3086)。

作者简介: 余国清(1971-), 男, 湖南常德人, 硕士, 副教授, 主要从事数据分析, 模式识别等方向的研究。

被设定为 150。相对于虚拟机容量, 资源要求较小的情况下, 允许对这一数量进行提升。接下来, 对于允许的每个任务数量 i (i 在 1 至 150 之间), 从任务聚类中随机选择出 i 个任务, 并计算出平均 CPU 利用率, 报告 CPU 错误并将其存储在 tmp_{error} 中。

然后, 算法根据 tmp_{error} 寻找具有最低 CPU 使用估计错误的 i 并将其作为虚拟机的任务数量。将这一过程重复 500 次, 这样就可以收集到足够的数出来得出结论。随后将每次迭代中虚拟机的任务数量存储到 Min_{error} 中。根据 Min_{error} , 每个虚拟机类型的任务数量为在大部分迭代中表现出最少的估计错误的数量。即该算法选择出最有可能导致较少的估计错误的任务数量。

算法 1: 每个虚拟机类型的最优任务数量估计

输入: 任务聚类

输出: 每个聚类的任务数量

- 1) 对于每个任务聚类 do
- 2) 平均 CPU \leftarrow 任务聚类的平均 CPU 使用率
- 3) for 从 1 至 500 的 k do
- 4) for 从 1 至 500 的 i do
- 5) 聚类样本 \leftarrow 任务聚类的 i 个不放回抽样
- 6) $avgCPU_s \leftarrow$ 聚类样本的平均 CPU 使用率
- 7) $CPU_{error} \leftarrow \frac{avgCPU - avgCPU_s}{avgCPU}$
- 8) $tmp_{error}[k] ? CPU_{error}$
- 9) 最小值的索引号
- 10) 每个聚类的任务数量 \leftarrow 模式

1.2 聚类中任务资源的使用情况估计

对每个虚拟机中的最大任务数量进行估计的目的是降低估计误差, 在此之后, 需要对虚拟机的类型进行定义。为此, 需要对虚拟机中运行的典型任务的资源使用情况进行估计。这里使用本文选定的数据集平均资源使用量和每个任务聚类的方差, 对一个聚类中的每个任务的资源使用情况进行估计。即:

第一步是计算出 trace 的第二天里每个任务的平均资源使用情况;

第二步是对于每个聚类, 将该组中任务的平均资源利用情况的置信区间设为 98%。

第三步是使用计算出的置信区间的上界作为指定聚类中一个典型任务的资源需求估计。

1.3 虚拟机配置估计

在得出需求 (RD) 和一个虚拟机类型 (nT) 中的任务数量之后, 使用下式推导出虚拟机的规格。

$$cap = \lceil nT * RD \rceil \quad (1)$$

式中, cap 表示容量, $\lceil \rceil$ 表示向上取整。因为虚拟机运行的任务已经对资源进行了共享, 所以每个虚拟机至少分配一个 CPU 核。同时, 上式的舍入取整也是对每个虚拟机任务数量的二次估计。

通常, 应用上述操作, 可以确定每个聚类的虚拟机类型。然后将得到的虚拟机类型存储在虚拟机类型仓库中, 由任务映射器完成分配。应用该操作得出的虚拟机类型如表 1 所示。在本文提出的基于使用量的资源分配策略中, 任务数量 nT 被用作虚拟机的任务容量。下一节将对本文提出的策略进行简短

讨论。

表 1 虚拟机配置

虚拟机类型	任务数量	虚拟 CPU	内存/GB
类型 1	136	3	4.5
类型 2	125	1	0.5
类型 3	500	1	0.8
类型 4	38	6	11
类型 5	139	5	3.4
类型 6	250	1	0.9
类型 7	143	14	20.6
类型 8	150	3	2.4
类型 9	154	8	4.3
类型 10	250	1	0.4
类型 11	188	3	1.6
类型 12	1250	1	1.1
类型 13	118	4	10.3
类型 14	126	25	14.2
类型 15	100	2	1.9
类型 16	136	3	6.8
类型 17	143	2	1.1
类型 18	500	1	3.8

2 资源分配策略

在不同的聚类中, 驻留在虚拟机中的任务数量也不相同。本文根据驻留在虚拟机中的任务使用模式来定制虚拟机配置。提出的所有策略均使用相同的虚拟机配置。然而, 每个任务聚类的虚拟机任务容量是不同的。具体资源分配策略详述如下:

1) 使用量的资源分配 (Utilization-based Resource Allocation, URA): 在该策略中, 根据映射到虚拟机中的任务资源的平均使用量, 以 98% 的置信区间计算出分配到每个虚拟机的任务数量。举例来说, 如果历史数据显示: 一个聚类中的任务平均使用 1 GB 的内存, 而该聚类中的任务将被分配到具有 RAM 为 4 GB 的虚拟机上, URA 将分配 4 个此类任务, 而不会考虑用户在提交相应的作业时所声明的估计内存容量。且每个虚拟机类型的任务容量与式 (1) 中的 nT 相等。

2) 请求量的资源分配 (Requested Resource Allocation, RRA): 在该策略中, 所考虑的虚拟机类型与 URA 相同; 但是, 分配到虚拟机的任务数量则是以提交任务的平均请求量为基础。如前文所述, 资源请求量与任务一起提交。本文将 RRA 作为基准, 在数据中心能源消耗和服务器利用率方面做进一步比较。

URA 的评价结果中推导出其他 4 个策略。在这方面, 本文研究了虚拟机的使用情况, 以更好的理解任务拒绝的原因 (CPU、内存或磁盘), 以及在发生拒绝情况时每个虚拟机中的运行任务数量。

对于每个虚拟机, 算法 2 给出了确定每个虚拟机运行任务的最小数量, 这使得虚拟机资源利用率超过其容量的 90%, 且不会造成任务被拒绝的情况。算法 2 中 $vmtol$ 表示虚拟机数量, $clindex$ 表示聚类索引, $sour$ 表示资源。

算法 2:

输入：虚拟机聚类列表 = $\{VMlist_1, \dots, VMlist_{18}\}$;
 虚拟机列表联合索引 = $\{VMID_1, \dots, VMID_{\text{vmtot}}\}_{\text{clindex}}$;
 sour 列表 = {CPU, 内存, 磁盘};
 输出: $nT_{\text{clindex}, \text{sour}} = \{nt_{VMID_1}, \dots, nt_{VMID_{\text{vmtot}}}\}$

- 1) for $clindex \leftarrow 1$ 至 18 do
- 2) 虚拟机 ID 列表 \leftarrow 虚拟机聚类列表 (clindex)
- 3) for 虚拟机 ID 列表中的虚拟机 ID do
- 4) 对于每个 资源列表中的资源 do
- 5) 找到运行任务的最小数量, 使得虚拟机资源利用率为其容量的 90% 至 100%。

对于由算法 2 聚类索引确定的聚类, 可以得出每个虚拟机类型的 nt 。然后, 对于每个被考虑的资源 (包括 CPU、内存和磁盘), 在每个聚类中将虚拟机的 nt 聚集到 $nT_{(clindex, sour)}$ 集合中。本文提出的 4 种策略可确定驻留在每个虚拟机中的任务数量。这些策略是基于每个聚类 $nT_{(clindex, sour)}$ 中推导出的估计值, 即平均、中位数、第一个分位数和第三个分位数。

1) 平均资源分配策略 (avgRA): 对于每个任务聚类, 设 m 为集合 $nT_{(clindex, sour)}$ 的长度, 对于平均任务数量, 有:

$$nT_{(avg, sour)} = \frac{\sum_{i=1}^m nt_{(i, sour)}}{m} \quad (2)$$

单独估计出每种资源的 nT_{avg} 。在该策略中, 驻留在每种虚拟机类型中的任务数量等于每种资源所得到的最小值的 nT , 即:

$$nT_{\min} = \min(nT_{avg, CPU}, nT_{avg, 内存}, nT_{avg, 磁盘}) \quad (3)$$

2) 中位数资源分配策略 (MeRA): 在该策略中, $nT_{clindex, sour}$ 集合的第二个分位数 (中位数) 被用于确定分配到每个虚拟机类型的任务数量。与之前的策略一样, 使用每种资源得出的最小数量确定虚拟机的任务容量。

3) 第一个分位数资源分配策略 (FqRA): 在该策略中, $nT_{clindex, sour}$ 集合的首个分位数 (一个有序集的 k 分位数即截至第一个 $25 * k\%$ 的数据, 对于第一个、第二个和第三个分位数, k 分别等于 1、2 和 3) 被用于确定分配到每个虚拟机类型的任务数量。该策略使用了每种资源得出的最小数量。此处资源指的依然是虚拟机的 CPU、内存或磁盘容量。

4) 第三个分位数资源分配策略 (ThqRA): 在该策略中, $nT_{clindex, sour}$ 集合的第三个分位数被用于确定分配到每个虚拟机的任务数量。与之前的策略一样, 使用每种资源得出的最小数量来确定虚拟机的任务容量。

3 实验及结果分析

本文在可用 trace 中, 对资源利用度量和请求进行归一化, 并为每列单独执行归一化操作。如文献 [10] 所述, 根据任何一个机器上所发现的某种资源的最高数量进行归一化操作。在这一背景下, 为了更好地理解数据, 本文假定每列的资源数量最高。为了消除放置约束, 仅对被调度到三个可用平台中的一个平台任务进行考虑。

提出的系统针对每个聚类进行模拟, 在每个处理窗口中 (实验中设置该过程为 1 分钟), 将任务分配到相对应的虚拟机类型。模拟运行时间设为 24 小时。单独报告每个任务聚类的资源使用量和被拒绝任务数量。具体实验及结果分析如下:

3.1 数据中心服务器配置及虚拟机放置

本文使用表 2 中列举的 3 个服务器配置定义一个数据中

心。设计理念来自于在本文研究的 trace 周期内的数据中心及其主机配置。一般集群中的主机在 CPU、内存和磁盘容量方面是异构的^[11-12]。然而, 相同平台 ID 的主机有着相同的架构。

表 2 平台中的可用服务器配置

服务器类型	核心数量	核心速度/GHz	内存/GB	磁盘/GB
类型 1	32	1.6	8	1000
类型 2	32	1.6	16	1000
类型 3	32	1.6	24	1000

本文应用第一个拟合算法, 作为寻找对新实例化的虚拟机进行托管的第一个可用机器的放置策略。该算法首先在运行主机中进行搜索, 以查找是否有足够的虚拟机可用资源。然后报告能够提供虚拟机所需资源的第一个运行主机。如果没有发现放置虚拟机的运行主机, 那么将启动一个新的主机。新主机从可用主机列表中选出, 该列表中包含主机 ID 及其配置, 从 trace 日志中得出。本文提出的所有算法使用同一个主机列表, 以确保放置策略不会影响到模拟结果。

3.2 实验结果综合性能分析

如表 3 所示, 长度较长的任务聚类表现出的使用差异相对较低。这使得资源分配策略可以对资源的使用估计和预测更加准确和高效, 因为所需的采样数据较少, 且可以对预测窗口进行加宽。长度较短的任务聚类则与之相反: 在这些聚类中可以观察到更大的差异, 因此, 在预测中需要更频繁的采样和更窄的时间窗口。

表 3 给出了每个算法的虚拟机容量。在 URA 策略中, 基于实际使用量对任务进行分配。由于任务的请求资源和实际用量之间存在差异, URA 中的虚拟机任务容量要高于其他算法。因此, 在大多数的聚类中, RRA 在一个虚拟机中容纳的任务数量最少。而在 URA 策略之外, ThqRA 有着最高的虚拟机任务容量。

表 3 不同分配策略下各聚类的虚拟机任务容量

聚类索引	RRA	URA	ThqRA	MeRA
聚类-1	15	136	38	32
聚类-2	20	125	43	40
聚类-3	20	500	16 500	16 500
聚类-4	41	38	56	38
聚类-5	28	139	56	46
聚类-6	8	250	117	77
聚类-7	80	143	62	53
聚类-8	73	150	48	47
聚类-9	41	154	107	104
聚类-10	6	250	8	7
聚类-11	12	188	105	99
聚类-12	11	1250	71	46
聚类-13	28	118	13	13
聚类-14	83	126	67	67
聚类-15	18	100	52	33
聚类-16	48	136	101	94
聚类-17	7	143	23	22
聚类-18	73	500	478	478

参考文献:

[1] Wilson W C, Coffey N C, Madaras E I. Leak detection and Location Technology Assessment for Aerospace Applications [R]. Langley Research Center, Hampton, Virginia, Sep. 2008.

[2] Dunlap P H, Jr. Christopher C. Daniels, et al. Pressure decay testing methodology for quantifying leak rates of full-scale docking system seals [R]. Glenn Research Center, Cleveland, Ohio. NASA/TM-2010-216244, 2010, April.

[3] 孟冬辉, 闫荣鑫, 喻新发, 等. 航天器舱体差压检漏系统研究 [J]. 真空, 2008, 45 (4): 62-65.

[4] 钟 君, 蔡黎明, 于 涌. 基于 MLX90614 的无线温度采集系统设计 [J]. 传感器与微系统, 2015, 34 (3): 87-89.

[5] 程建华, 齐 兵, 屈传波. 一种高精度低自热多通道测温系统设计与实现 [J]. 传感器与微系统, 2014, 33 (1): 56-63.

[6] 金海龙, 赵军卯. 基于恒流源的多路高精度测温仪的研究 [J]. 仪器仪表学报, 2006, 27 (6): 242-244.

(上接第 274 页)

在任务拒绝率方面, 如果算法的虚拟机任务容量较大, 则任务拒绝率也比较高。因此, 在大部分聚类中, 文献 [8] 的任务拒绝率最高。但是本文 RRA 和 ThqRA 在任务拒绝率上的差异几乎可以忽略不计。因此, 每个虚拟机中任务数量最低的 RRA 策略在模拟实验中任务被拒绝的情况最少。

表 4 不同方法每分钟的平均拒绝次数

聚类索引	本文 RRA	文献[8]	本文 ThqRA	文献[5]	文献[6]
聚类-1	0.2	1	0.3	0.2	0.4
聚类-2	0	5	1	2	2.2
聚类-3	0	0	0	0	0
聚类-4	2	8	6	3	4
聚类-5	1.8	6	3	2	1
聚类-6	3	11	8	7	3.7
聚类-7	0	1	0	0.3	0.6
聚类-8	0.1	0.8	0.2	0.4	0.5
聚类-9	0.1	3	1	0.4	0.7
聚类-10	2	9	2	1	1
聚类-11	1.2	3	1.5	1.9	1.9
聚类-12	1	4	1	4	5
聚类-13	0.2	0.8	0.3	0.1	0.3
聚类-14	1.3	0.6	1.7	1.7	1.9
聚类-15	1	4.1	2.2	3.3	3.7
聚类-16	0.4	1	1	0.4	0.8
聚类-17	1	2	1.3	2.2	2.5
聚类-18	0.3	0.3	0.4	0.2	0.3

由于各算法的主机使用量均比较少, 本文策略在能耗方面表现更优。本文使用如下公式计算机每个活动节点的能耗:

$$EC_i = \frac{(\sum_{j=1}^v \sum_{k=1}^u cpu_{ijk}) (\sum_{j=1}^v \sum_{k=1}^u mu_{ijk})}{M} \quad (4)$$

式中, v 表示第个节点上运行的虚拟机数量, 存储器单元数为 M , 虚拟机的任务数量用 u 表示, cpu_{ijk} 表示虚拟机中的 k 个任务的处理器情况, mu_{ijk} 分别表示存储器的使用情况。

与其他方法的性能比较如表 5 所示。可以发现, 本文方法的能耗明显减少, 其在云环境下的性能评价更优。因此可扩展性也更强。在大部分聚类中, 文献 [5-8] 的平均任务拒绝率较高, 使得任务执行的延时也很高。因此, 这些策略适用于低优先级且执行中延时敏感度较低的任务。而本文方法可以执行较高的延时敏感度任务。

表 5 各方法框架的性能比较

方法框架	迁移减少量	能量节约量	主机减少量
与文献[5]方法相比	60.1	30.7	20.5
与文献[8]方法相比	70.1	40.2	36.1
与文献[6]方法相比	68.4	38.0	29.8

4 结论

本文提出了一个端到端的架构, 并根据工作负载定制虚拟机配置的方法。在对任务进行聚类并映射到虚拟机时, 考虑的是每个聚类的实际资源使用量, 而非用户请求的资源量。在模拟实验中, 本文在每个任务聚类中对提出的算法进行了性能比较, 结果表明本文提出的策略具有一定优势。

未来将研究能量感知的虚拟机放置算法, 考虑每组任务的特征和约束。此外, 还将根据每组任务的规格选择正确的策略。

参考文献:

[1] 曹 洁, 曾国荪, 匡桂娟, 等. 支持随机服务请求的云虚拟机按需物理资源分配方法 [J]. 软件学报, 2017, 28 (2): 457-472.

[2] 罗 杰, 张之明, 高志强, 等. 基于效益最大化的云虚拟机资源分配研究 [J]. 计算机应用研究, 2016, 33 (10): 2963-2966.

[3] 郑顾平, 王秋萍. 基于负载预测的虚拟机资源优化分配方案 [J]. 科学技术与工程, 2016, 16 (23): 223-228.

[4] 吴伟美, 陈 勋, 段班祥, 等. 虚拟机资源概率配置的云计算 SEF-FD 算法 [J]. 计算机与现代化, 2016 (10): 15-20.

[5] 高任飞, 武继刚, 周 莹, 等. 极小通讯延迟的虚拟机分配算法 [J]. 计算机科学与探索, 2016, 10 (7): 924-935.

[6] 王煜烽, 刘 敏, 房秉毅, 等. 面向网络性能优化的虚拟计算资源调度机制研究 [J]. 通信学报, 2016, 37 (8): 105-118.

[7] 常春雷, 杨大伟, 陈 斌, 等. 基于虚拟机智能动态资源调度策略技术的研究 [J]. 电气时代, 2016 (11): 68-71.

[8] 李明阳, 严 华. 改进粒子群算法在云计算负载均衡中的应用研究 [J]. 计算机测量与控制, 2016, 24 (10): 219-221.

[9] 邓 莉, 姚 力, 金 瑜. 云计算中基于多目标优化的动态资源配置方法 [J]. 计算机应用, 2016, 36 (9): 2396-2401.

[10] 刘 盛, 虞红芳, 郑少平. 面向业务动态变化的虚拟机迁移技术研究 [J]. 计算机应用研究, 2016, 33 (2): 534-539.

[11] 梁 鑫, 桂小林, 戴慧珺, 等. 云环境中跨虚拟机的 Cache 侧信道攻击技术研究 [J]. 计算机学报, 2017 (2): 317-336.

[12] 栗少萍, 秦召广. 异构网络下车联网的注水资源分配算法研究 [J]. 电子设计工程, 2016, 24 (9): 139-143.