

基于Openstack的KVM调优实战

调优背景

2015年11月上旬，CRMAPP系统所使用的KVM虚拟机的CPU使用率过高异常，达到70%以上，相同业务压力下同性能配置的Vmware虚拟机的负载却非常低，只在10%以内波动，并且发现KVM宿主机的CPU使用率也异常高。

```
od01crmapp05: ~ # sar -u 1
Linux 3.0.76-0.11-default (od01crmapp05)      11/09/15      _x86_64_

10:48:34      CPU      %user      %nice      %system      %iowait      %steal      %idle
10:48:35      all       20.63      51.44         5.93         0.03         0.00      21.96
10:48:36      all       21.58      51.12         4.49         0.17         0.00      22.64
10:48:37      all       30.28      48.72         5.95         0.23         0.00      14.83
10:48:38      all       55.99      34.31         9.60         0.03         0.00         0.07
10:48:39      all       30.00      50.13         7.57         0.65         0.00      11.65
10:48:40      all       26.78      48.95         5.48         0.29         0.00      18.50
10:48:41      all       38.77      46.51         7.43         0.16         0.00         7.14
10:48:42      all       32.31      48.76         5.50         0.29         0.00      13.13
10:48:43      all       55.78      33.78         7.53         0.00         0.00         2.91
10:48:44      all       47.54      42.22         6.24         0.00         0.00         4.01
10:48:45      all       40.90      43.88         8.15         0.10         0.00         6.97
10:48:46      all       31.57      47.71         6.20         0.13         0.00      14.39
10:48:47      all       34.93      47.53         7.07         0.10         0.00      10.38
10:48:48      all       20.62      52.03         4.67         0.26         0.00      22.43
10:48:49      all       29.94      47.31         6.73         0.00         0.00      16.01
10:48:50      all       38.28      44.89         7.10         0.13         0.00         9.59
10:48:51      all       29.74      22.45         6.37         0.22         0.00      41.22
10:48:52      all       20.67         7.96         5.01         0.38         0.00      65.99
10:48:53      all       16.75         6.21         4.99         1.03         0.00      71.03
10:48:54      all       17.55         2.71         3.79         2.87         0.00      73.09
10:48:55      all       30.66         5.27         5.17         0.36         0.00      58.55
```

分析与解决

分别从以下几个方面逐一分析排查问题原因：

1、KVM的CPU虚拟模式

首先查看KVM虚拟机CPU信息如下：

```
od01crnapp05:~ # lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                 32
On-line CPU(s) list:   0-31
Thread(s) per core:    1
Core(s) per socket:    1
Socket(s):              32
NUMA node(s):          1
Vendor ID:              GenuineIntel
CPU family:             6
Model:                  42
Stepping:               1
CPU MHz:                2199.998
BogoMIPS:               4399.99
Virtualization:         VT-x
Hypervisor vendor:     KVM
Virtualization type:    full
L1d cache:              32K
L1i cache:              32K
L2 cache:               4096K
NUMA node0 CPU(s):     0-31
od01crnapp05:~ #
```

与Vmware虚拟机CPU相比较如下：

```

Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                32
On-line CPU(s) list:   0-31
Thread(s) per core:    1
Core(s) per socket:    8
Socket(s):             4
NUMA node(s):          4
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 45
Stepping:               7
CPU MHz:               2200.000
BogoMIPS:               4400.00
Hypervisor vendor:     VMware
Virtualization type:   full
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              12288K
NUMA node0 CPU(s):     0-7
NUMA node1 CPU(s):     8-15
NUMA node2 CPU(s):     16-23
NUMA node3 CPU(s):     24-31

```

通过比较Vmware与KVM的虚拟机CPU信息发现KVM虚拟机CPU模式存在如下几个问题：

[] 缺少L3 Cache：初步分析是虚拟机CPU虚拟化模式选择不合理所导致。[/] [] CPU不是NUMA架构并且CPU Topology不合理：KVM宿主机物理CPU属于NUMA多node的结构，虚拟机的CPU只有一个NUMA node，所有CPU Core都在这一个node中，且虚拟机CPU的Topology是多Socket单Core的形式。[/]

处理措施

>>>> 缺少L3 Cache

针对该问题，检查了Openstack的nova.conf配置文件libvirt部分的cpu_mode的参数配置是host-model，该参数含义是根据物理CPU的特性，选择一个最靠近的标准CPU型号进行虚拟化模拟。除了host-model外还可以有host-passthrough模式，该模式直接将物理CPU暴露给虚拟机使用，在虚拟机上完全可以看到的就是物理CPU的型号。

因Openstack 仍承载业务，选择小范围的修改cpu_mode的参数，通过将Openstack的代码文件 driver.py中cpu_mode的取值修改为host-passthrough并重启宿主机上Nova-computer服务与KVM虚拟机，将自动重新生成虚拟机的 libvirt.xml文件。

>>>> CPU不是NUMA架构并且CPU Topology不合理

经过梳理Openstack虚拟机的创建流程，并查阅Openstack官方文档与代码，发现在JUNO版的Openstack中，KVM的CPU的拓扑可以通过image或者flavor进行元数据传递来定义，如果没有特别的定义此类元数据，则模拟的CPU将是多Socket单Core单NUMA节点的CPU，这样的CPU与物理CPU完全不同。

通过nova命令对flavor增加了hw:numa_cpu、hw:numa_nodes、hw:cpu_sockets等属性。

处理结果

经过上面两个方面的修改后，新建KVM虚拟机的CPU的信息发生如下改变，CPU的使用率有了明显下降，在10%到20%之间波动。

[]从单个NUMA节点变成4个NUMA节点[/][]具备了L3 cache[/][]CPU的Topology从32个Socket 每个Socket 1个 core, 变成了4个Socket 每个Socket 8个Core[/]

```
Architecture:      x86_64
CPU op-mode(s):    32-bit, 64-bit
Byte Order:        Little Endian
CPU(s):            32
On-line CPU(s) list: 0-31
Thread(s) per core: 1
Core(s) per socket: 8
Socket(s):          4
NUMA node(s):       4
Vendor ID:          GenuineIntel
CPU family:          6
Model:              45
Model name:         Intel(R) Xeon(R) CPU E5-4607 0 @ 2.20GHz
stepping:           7
CPU MHz:            2199.998
BogoMIPS:           4399.99
virtualization:     VT-x
Hypervisor vendor:  KVM
virtualization type: full
L1d cache:          32K
L1i cache:          32K
L2 cache:           256K
L3 cache:           12288K
NUMA node0 CPU(s):  0-7
NUMA node1 CPU(s):  8-15
NUMA node2 CPU(s):  16-23
NUMA node3 CPU(s):  24-31
```

2、KVM宿主机与NUMA运行状况

在NUMA的CPU内存架构下，无论是物理主机还是虚拟机，如果NUMA的配置不合理对应用程序的性能都有较大的影响，并且不同类型的應用都有不同的配置需求。

Vmware ESX 5.0及之后的版本支持一种叫做vNUMA的特性，它将Host的NUMA特征暴露给了Guest OS，从而使得Guest OS可以根据NUMA特征进行更高性能的调度。SUSE与Redhat作为原生的操作系统在NUMA调度上需要人为的根据应用程序的类型的做特殊配置，对云平台来说，这部分的工作是难以做到的。

在优化之前，CRM APP的KVM虚拟机的NUMA调度状态非常不理想，表现为所有NUMA节点的numa_miss统计数值大于numa_hit，这意味着CPU访问内存的路径不是优化的，存在大量CPU访问remote memory的情况。因为宿主机Ubuntu 12.02自身带有Automatic NUMA balancing，所以物理主机NUMA调度运行状态良好。

处理措施

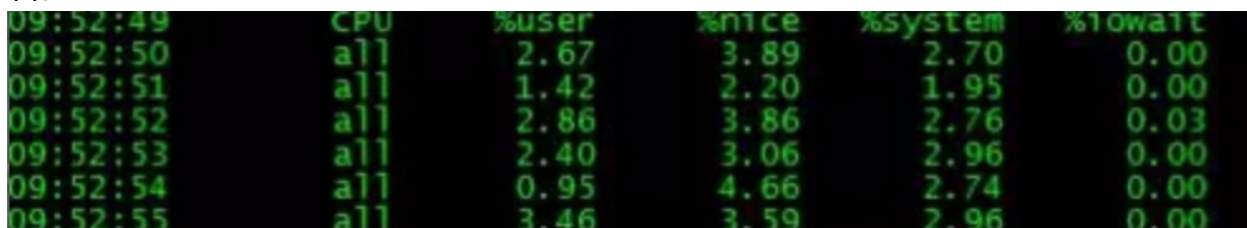
升级KVM虚拟机的操作系统到SUSE 12，因为新版本的SUSE支持Automatic NUMA balancing，并且操作系统检测到硬件属于NUMA架构时将自动开启。

处理结果

在新建的KVM虚拟机的 dmesg日志中可以看到如下信息：

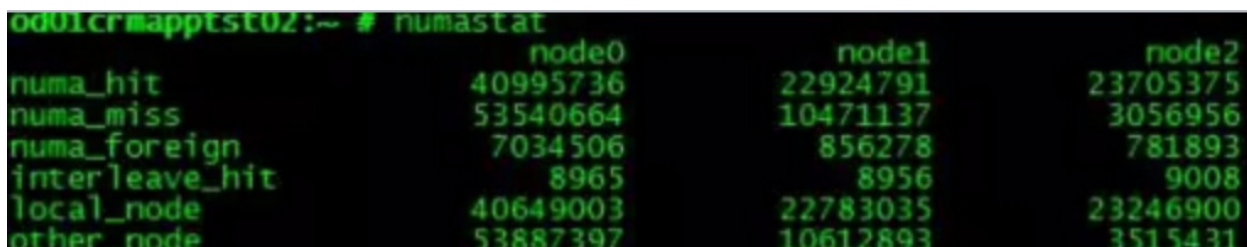
```
Enabling automatic NUMA balancing. Configure with numa_balancing=
or sysctl
```

经过24小时的运行之后，CRMAPP的KVM虚拟机运行状态良好，CPU用率可以稳定在10%左右。



	CPU	%user	%nice	%system	%iowait
09:52:49					
09:52:50	all	2.67	3.89	2.70	0.00
09:52:51	all	1.42	2.20	1.95	0.00
09:52:52	all	2.86	3.86	2.76	0.03
09:52:53	all	2.40	3.06	2.96	0.00
09:52:54	all	0.95	4.66	2.74	0.00
09:52:55	all	3.46	3.59	2.96	0.00

同时NUMA调度也有了较大程度的改善



	node0	node1	node2
numa_hit	40995736	22924791	23705375
numa_miss	53540664	10471137	3056956
numa_foreign	7034506	856278	781893
interleave_hit	8965	8956	9008
local_node	40649003	22783035	23246900
other_node	53887397	10612893	3515431

总结

[]通过各环节的优化，目前KVM虚拟机的CPU利用率过高问题不再发生，整体运行达到Vmware虚拟机水平。[/]不同类型的应用程序对于NUMA适应性不同，需要进行针对性优化。JAVA

在NUMA方面也可尝试进行优化。参考Oracle官方文档，JAVA7针对并行扫描垃圾回收站（Parallel Scavenger garbage collector）加入了对NUMA体系结构的支持，实现了NUMA感知的内存分配器，由它为Java应用提供自动的内存分配优化。[/][/]目前RedHat7与SUSE 12都加入了NUMA自动负载均衡的特性，可以尽量采用较新版本的操作系统，无论是物理机还是虚拟机，对于NUMA调度的优化不仅与KVM虚拟机有关，其实物理主机也应该关注NUMA调度是否是最优的。[/]