Aula 10 - Gerenciamento de memória

Sobre

- Objetivos:
  - Entender como o Linux gerencia memória e como o KVM tira vantagem disso.
  - Compreender e implementar ajustes finos de memória para maior performance.
  - Redução e aumento de memória durante a execução.

Aspectos sobre Linux e memória

- A maioria dos sistemas operacionais e aplicativos não usam 100% da memória RAM disponível o tempo todo.
- O kernel Linux aloca memória para cada processo quando o processo solicita mais memória.
- O kernel Linux faz swap de memória raramente usada da memória física para a área de swap.
- Quando a memória física é totalmente utilizada ou um processo fica inativo por algum tempo, o Linux move a memória de um processo para o swap.
- O swap é normalmente uma partição/volume que o Linux usa para aumentar a memória virtual.
- Swap é significativamente mais lento que RAM, devido à transferência e tempos de resposta dos discos rígidos e drives de estado sólido.

Gerenciamento de memória do Linux

- Comandos básicos
  - ps
  - top
  - vmstat
- Parâmetros do /proc/meminfo
-  The Linux Page Cache and pdflush: Theory of Operation and Tuning for Write-Heavy Loads

Aspectos sobre KVM e memória no Linux

- KVM pode alocar mais memória para os clientes virtuais do que fisicamente disponível no host.
- Os guests são apenas processos para o Linux.
- Os guests não tem blocos de RAM física dedicados a eles.

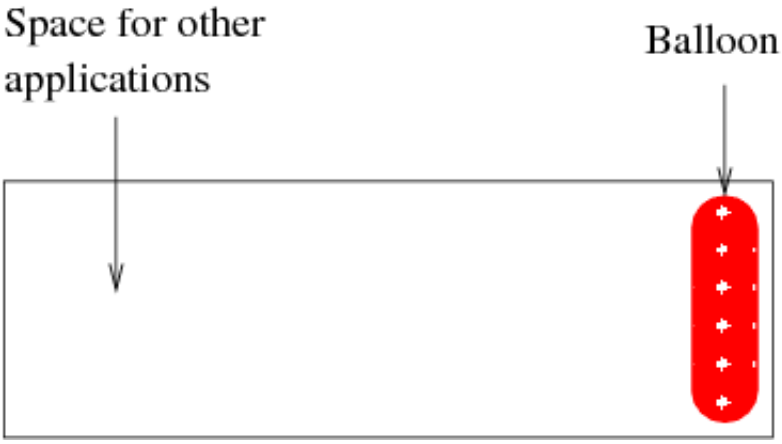
- KVM alocada memória quando solicitado pelo sistema operacional do guest.
- O cliente requer apenas um pouco mais de memória física do sistema operacional virtualizado relatórios como usados.
- Usando swap diminui a quantidade de memória real necessária pelos guests.
- Como as máquinas virtuais KVM são processos, memória subutilizadas ou ociosas de guests virtualizados é movida por padrão para swap. (swappiness)
- O total de memória usada pelos clientes pode ser maior que a memória física disponível (overcommit).
- O overcommitting requer espaço de swap suficiente para todos os guests e todos os processos.
- Sem espaço de swap suficiente para todos os processos na memória virtual do processo pdflush, o processo de limpeza, começa.

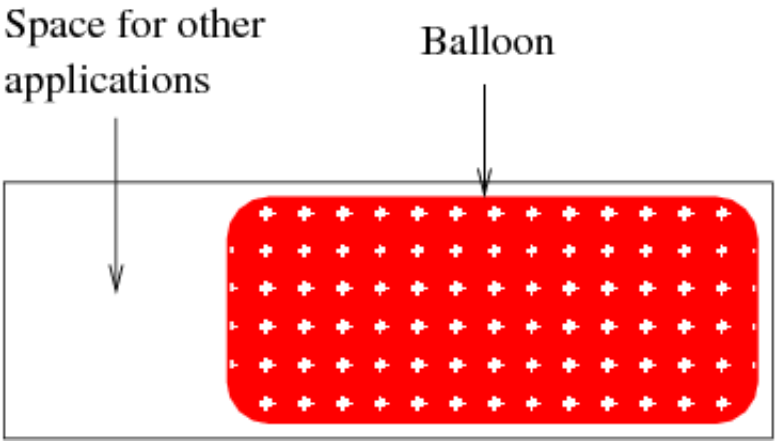
Tipos de overcommit

- Swap
  - Esta é a forma clássica de apoio ao overcommit, o host escolhe algumas páginas de memória de um dos guests e as envia para o disco. Se um hóspede exige memória que tenha ido para a swap, o host trás de volta do disco.
- Balões (ballooning)
  - Com o balão, o host e o guest cooperam em que páginas serão liberadas. É de responsabilidade do guest escolher a página e liberá-la, se necessário.
- Compartilhamento de páginas
  - O hypervisor olha para páginas de memória que possuem dados idênticos, estas páginas são fundidas em uma única página, que é marcada apenas para leitura. Se um cliente escreve em uma página compartilhada, ela é "descompartilhada" antes de conceder ao guest a gravação.
- Live-migration
  - O hypervisor move um ou mais guests para um host diferente, liberando a memória.

Balloning

- Balonismo é bastante eficiente, já que depende do guest para liberar a memória. Muitas vezes, os clientes podem simplesmente encolher seu cache para liberar memória, que pode ter um impacto muito baixo no guest. O problema com o balonismo é que ele depende da cooperação dos guests, o que reduz a sua confiabilidade.





- Como funciona:

```
# kvm -balloon virtio

ou

# kvm -device virtio-balloon-pci
```

- No guest:

```
# lspci | grep RAM
00:04.0 RAM memory: Red Hat, Inc Virtio memory balloon
modprobe virtio_balloon
```

- Acessar o monitor da VM:

```
(qemu) info balloon
balloon: actual 1024
(qemu) balloon 512
(qemu) info balloon
balloon: actual 512
```

- Muito bom: http://rwmj.wordpress.com/2010/07/17/virtio-balloon/

- Agente de gerenciamento de balão: http://aglitke.wordpress.com/2011/03/03/automatic-memory-ballooning-with-mom/

Swapping

Swap não depende do cliente, por isso é mais confiável do ponto de vista do host. No entanto, o host tem menos conhecimento do que o guest sobre a memória do guest, assim que swap é menos eficaz que o balão.

- Parâmetros de ajuste de swap:

```
# sysctl -w vm.swapiness=100 (quanto maior, mais agressivamente o
kernel fará swap)
# sysctl -w vm.overcommit_memory=2 (0 heurístico, 1 overcommit sem
limit, 2 não fazer overcommit)
# sysctl -w vm.overcommit_ratio=50 (quando overcommit_memory=2)
```

Quando overcommit_memory=2: **memória_total** = **swap** + **(memória_física * (overcommit_ratio / 100))**

Compartilhamento de páginas

Compartilhamento de páginas depende do comportamento dos hosts indiretamente. Quando os guests executam aplicativos semelhantes, o host vai atingir uma proporção elevada de compartilhamento.

- KSM - Kernel Samepage Merging
- Increasing memory density by using KSM
- Usando no host:

```
# echo 1 > /sys/kernel/mm/ksm/run (demora alguns segundos para os
primeiros resultados)
# cat /sys/kernel/mm/ksm/pages_sharing (multiplicar por 4KB para total
compartilhado)
5487
```

Estratégia do KVM

Então kvm usa uma estratégia mista: compartilhamento de página e balões são usados como métodos preferenciais para a overcommit de memória uma vez que são eficientes. Livre migration é usada para equilibrar a longo prazo dos requisitos de memória e recursos. Swap é usada como último recurso.

Hugepages

Hugepages permite uso de páginas de 4MB em sistemas x86 de 32 bit e 2MB de 64bit. Assim, há um uso mais eficiente da memória quando é necessário um grande volume de memória. Outra característica também é que esse tipo de páginas não pode ser movida para a área swap.

- Como configurar hugepages: http://www.freedominterface.org/2010/09/27/hugepages/
- KVM e hugepages: http://www.linux-kvm.com/content/get-performance-boost-backing-your-kvm-guest-hugetlbfs
- Detalhes técnicos de hugepages e TLB: http://publib.boulder.ibm.com/infocenter/lnxinfo/v3r0m0/index.jsp?topic=/liaat/liaattunhp.htm
- Não dá pra misturar KSM e hugepages: http://us.generation-nt.com/answer/ksm-hugepages-help-196557231.html

Novidades

- KVM: Add host swap event notifications for PV guest: http://lwn.net/Articles/409961/

Para estudo aprofundado

- Kernel Virtualization Optimizations for KVM (muita coisa!): http://www.redhat.com/promo/summit/2010/presentations/summit/decoding-the-code/thurs/jshaksho-310-420-performance/larry_shak_perf_summit2010_v2.pdf
- Paginação de memória em hardware (NPT/EPT): http://avikivity.blogspot.com/2008/04/paravirtualization-is-dead.html
- Documentação definitiva de hugepages: http://lwn.net/Articles/374424/
- What every programmer should know about memory
  - Parte 1: http://lwn.net/Articles/250967/
  - Parte 2: http://lwn.net/Articles/252125/
  - Parte 3: http://lwn.net/Articles/253361/
  - Parte 4: http://lwn.net/Articles/254445/
  - Parte 5: http://lwn.net/Articles/255364/
  - Parte 6: http://lwn.net/Articles/256433/
  - Parte 7: http://lwn.net/Articles/257209/