

Gerenciador Condor KVM
Roberto Kendy Sawamura
Abril/2013

Sistema Gerenciador de Máquinas Virtuais Trabalhadoras de Tarefas Condor

Objetivo: otimizar o consumo de recursos de uma máquina física, ativando suas máquinas virtuais trabalhadoras apenas quando necessário.

1. Visão Geral

O sistema monitora a fila de Jobs do Condor, ativando ou desativando máquinas virtuais (VM), visando a melhor utilização dos recursos disponíveis na máquina física (host).

Principais funções:

- Mantém ativa a máquina virtual responsável pela fila de Jobs do Condor;
- Ativa VMs trabalhadoras caso haja Jobs aguardando recursos;
- Desativa VMs trabalhadoras caso a fila de Jobs esteja vazia.

O diagrama exhibe os principais componentes e os passos executados pelo Gerenciador.

Componentes:

- VM Condor: máquina virtual responsável pela fila de Jobs do Condor;
- VM 1 à n: máquinas virtuais trabalhadoras. São máquinas que executam os Jobs da fila;
- Libvirt: biblioteca/api de virtualização. É através dela que o Gerenciador envia comandos para as máquinas virtuais;
- Gerenciador: componente central. É responsável pela leitura da fila de Jobs e ativação/desativação das VMs. O Gerenciador é ativado a cada x segundos (definido por uma constante) e executa os passos 1 à 5, descritos detalhadamente na seção 2.

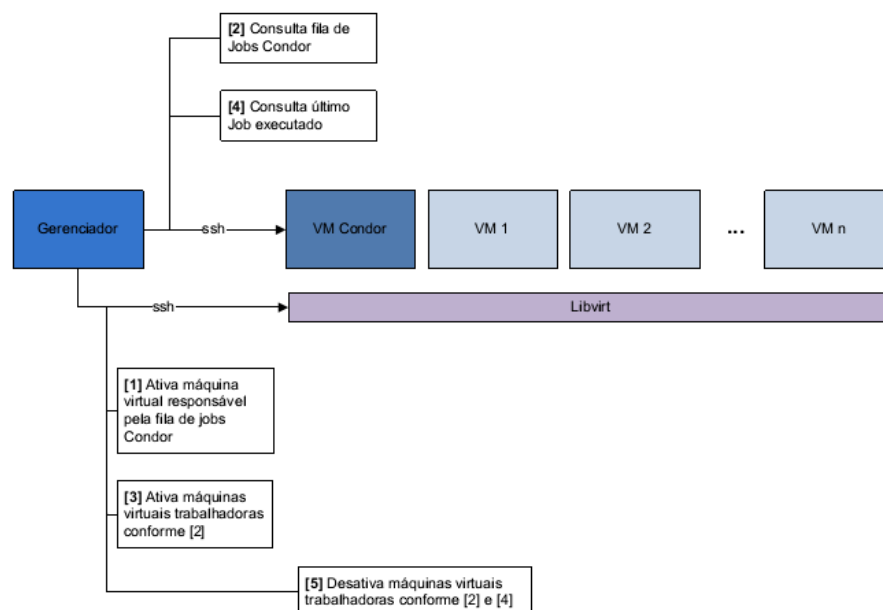


Figura 1- Visão Geral

2. Ações do Gerenciador

As ações executadas pelo Gerenciador seguem a ordem numérica de 1 a 5. Algumas ações podem não ser executadas dependendo do estado do ambiente.

As ações são descritas abaixo:

1. Ativa máquina virtual responsável pela fila de Jobs Condor

A máquina virtual responsável pela fila de Jobs Condor é essencial ao processo de monitoramento pois permite a consulta dos status das tarefas e assim a tomada de decisão em relação à ativação/desativação das VMs. Ela deve estar sempre ativa.

Seu nome (definido no momento de sua criação como máquina KVM) é especificado na constante “VM_SERVIDOR_ESSENCIAL”, no arquivo `acessoLibVirt.h`.

Funcionamento: o monitor consulta o status da máquina virtual pelo `libvirt`. Caso o status seja `VIR_DOMAIN_SHUTOFF` ou `VIR_DOMAIN_CRASHED`, a VM é ligada (`virDomainCreate`). Caso o status seja `VIR_DOMAIN_PAUSED`, a VM é ativada (`virDomainResume`).

2. Consulta fila de Jobs Condor

Com a consulta da fila de Jobs Condor, é possível verificar quantas tarefas estão aguardando recursos e quantas tarefas estão em execução. Essas informações são utilizadas para ativação ou desativação das VMs trabalhadoras.

A consulta da fila de Jobs é realizada via SSH. O comando do condor responsável pela leitura da fila de jobs é o “`condor_q`”, definido na constante `CONDOR_QUEUE_COMMAND`, no arquivo `acessoCondorQueue.h`.

As informações lidas pelo comando são:

- `GlobalJobId`: Id único do job
- `Owner`: Dono do job (não utilizado)
- `Qdate`: Data de quando o job foi submetido (não utilizado)
- `JobStatus`: Status do job

Funcionamento: a consulta é executada via SSH, e a partir do seu resultado o gerenciador monta duas listas: uma contendo informações das tarefas aguardando execução e outra com informações das tarefas em execução. O registro de uma dada tarefa é inserido na sua respectiva lista com seu *contador* zerado. Se na próxima execução do gerenciador (próxima leitura da fila do condor) a mesma tarefa for encontrada (mesmo id e mesmo status), seu *contador* é incrementado. Caso contrário o registro é removido da lista.

Os códigos dos status estão definidos no arquivo “`acessoCondorQueue.h`”, em `JOB_STATUS_IDLE` (valor 1) e `JOB_STATUS_RUNNING` (valor 2) e são dados pelo Condor. Tarefas com outros status são ignorados.

3. Ativa máquinas virtuais trabalhadoras

Tarefas que permanecem com status “aguardando execução” durante algumas iterações

indicam falta de recurso para que sejam executadas. Neste contexto VMs devem ser ativadas.

Funcionamento: se houver alguma tarefa na lista de status "aguardando execução" com seu contador maior que o valor mínimo definido na constante `MAX_COUNT_READER_JOB_WAITING` (arquivo `acessoCondorQueue.h`), uma máquina virtual é ativada.

O gerenciador procura ativar primeiramente VMs em pausa. Caso não encontre, o gerenciador procura VMs desligadas. Caso não encontre, nada é executado.

Apenas VMs com nome iniciando com "Condor-wn" (definido em `VM_NOME_INICIO_PADRAO` no arquivo `acessoLibVirt.h`) são consideradas.

O registro da tarefa que gerou a ativação da VM não é modificado. Em teoria ele seria excluído pois seu status seria alterado para "em execução", causando a sua remoção na próxima iteração (próxima leitura da fila do Condor).

Apenas uma VM é ativada por iteração, mesmo se a fila possuir mais de uma tarefa aguardando execução e com contador maior que o mínimo.

4. Consulta último Job executado

A ausência de tarefas nas listas de "aguardando execução" e "em execução" pode indicar ausência de tarefas no sistema. Para certificar que não há tarefas sendo executadas entre as consultas ao Condor, a última tarefa executada é lida do histórico do Condor. Caso o cenário permaneça constante, VMs podem ser desativadas.

Funcionamento: se as duas filas de registros de tarefas ("aguardando execução" e "em execução") estiverem vazias, o monitor consulta o registro da última tarefa executada. Esta consulta é realizada pelo comando "condor_history", definido em `CONDOR_HISTORY_COMMAND`. O monitor armazena a última tarefa executada e um contador, que é zerado inicialmente e incrementado a cada iteração.

5. Desativa máquinas virtuais trabalhadoras

Se o cenário definido no item 4 permanecer constante, VMs são desativadas, garantindo o melhor uso da máquina física.

Funcionamento: quando o contador do último job executado ultrapassar o valor definido em `MAX_COUNT_READER_NO_RUNNING_JOBS`, o monitor desativa uma VM. Primeiramente o monitor procura por VMs em execução, e executa o comando de pausa. Se todas as VMs já estiverem em pausa, então uma vm é desligada.

Apenas VMs com nome iniciando com "Condor-wn" (definido em `VM_NOME_INICIO_PADRAO` no arquivo `acessoLibVirt.h`) são consideradas.

O contador é utilizado para ter certeza que o sistema não possui job sendo concluído entre as iterações do monitor. Ele define também o tempo necessário para considerar o sistema como parado.

A constante `PAUSED_VM_COUNT_MIN` indica o mínimo de VMs em Pausa na ausência de tarefas no sistema. Assim, mesmo se não houver nenhuma tarefa sendo executada, haverá um número mínimo de VMs prontas para serem ativadas, melhorando a resposta do ambiente para a chegada de uma tarefa.

Observação: caso a constante `GERENCIA_MAQUINAS_VIRTUAIS` (definido em `main.c`) estiver com valor diferente de "1", nenhuma máquina virtual será ativada ou desativada.

3. Monitor em estado inativo

Após as verificações, o monitor entra em estado inativo por `n` segundos. O tempo inativo depende de qual operação foi executada.

Se houve ativação de VM, o tempo é definido em:

`SLEEP_TIME_COM_ATIVACAO_DE_DOMINIO.`

Se houve desativação de VM, o tempo é definido em:

`SLEEP_TIME_COM_DESATIVACAO_DE_DOMINIO.`

Se não houve nenhuma ação, o tempo é definido em:

`SLEEP_TIME_SEM_ACAO.`

ToDo: separar ainda mais, caso a ativação/desativação seja de uma vm desligada/ligada ou pausada/interrompida.

4. Logs

Um arquivo de log é criado no mesmo diretório onde é executado o gerenciador. O nome do arquivo segue o padrão "log-YYYYMMDD-HHmm.log". Na mudança do dia, um novo arquivo é gerado.

5. Compilação

É preciso instalar o pacote "libvirt-dev".

Rodar o comando "make" no diretório do projeto. Caso o comando "pkg-config --libs libvirt" não retorne os parâmetros corretos, ajuste a variável de ambiente conforme o caso:

`export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig/`

A compilação manual pode ser feita utilizando as opções "-I" e "-L" conforme o exemplo:

`gcc -I/usr/local/include/libvirt -L/usr/local/lib -lvirt *.c`

Pode ser necessário ajustar a variável de ambiente conforme o comando:

`export LD_LIBRARY_PATH=/usr/local/lib`

6. Implantação

– SSH:

Como os comandos para leitura do condor (fila e histórico) e acesso ao libvirt são via ssh, é necessário permitir login sem o prompt de senha. É preciso verificar qual usuário executará o serviço, e incluir sua chave na lista de autorizados do ambiente destino. Os comandos executados na origem devem ser o suficiente:

cd

ssh-keygen -t rsa (enter para todas as perguntas)

cat .ssh/id_rsa.pub | ssh *usuario@servidor* 'cat >> .ssh/authorized_keys'

“usuario” deve ser substituído pelo usuário que irá rodar o serviço.

“servidor” deve ser substituído pelo endereço onde o comando será executado (condor_q, condor_history ou ambiente libvirt).

– Serviço

Para executar como serviço, utilize o script “*monitorKvmCondor*”. Copie o script para o diretório */etc/init.d*, dando as permissões necessárias para a sua execução.

Inclua o script para execução automática conforme o comando:

chkconfig --level 345 monitorKvmCondor on

Ou desabilite conforme:

chkconfig monitorKvmCondor off

A chamada manual segue o padrão.

/etc/init.d/monitorKvmCondor start

/etc/init.d/monitorKvmCondor stop

/etc/init.d/monitorKvmCondor status

/etc/init.d/monitorKvmCondor restart

7. ToDo

- 1) Permitir gerenciar mais de um host libvirt. Seria necessário uma lista de hosts, com o padrão dos nomes das VMs para cada host;
- 2) Permitir ativar/desativar o processo de monitoramento da VM especial (fila de Jobs Condor) caso a máquina não seja virtual e sim física;
- 3) Permitir especificar o endereço do host onde a VM especial está implantada;