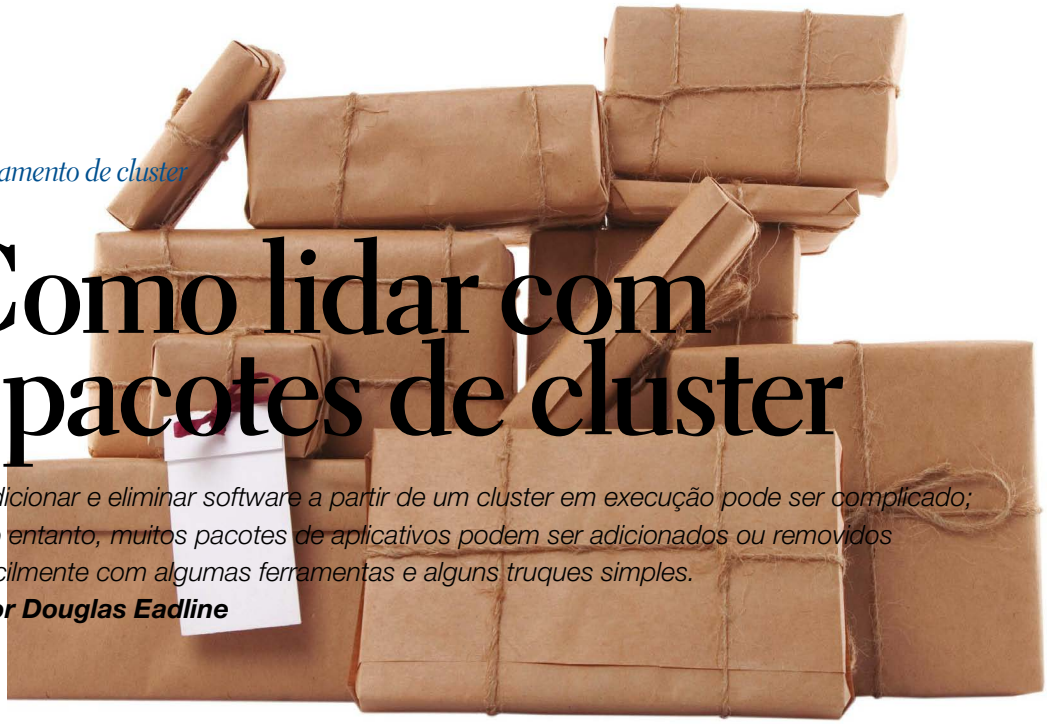
Como lidar com pacotes de cluster



Adicionar e eliminar software a partir de um cluster em execução pode ser complicado; no entanto, muitos pacotes de aplicativos podem ser adicionados ou removidos facilmente com algumas ferramentas e alguns truques simples.

por Douglas Eadline

Instalar e configurar um cluster HPC não é tão difícil como costumava ser; algumas ferramentas fornecem recursos agradáveis que permitem a praticamente qualquer pessoa obter sucesso em pouco tempo. Uma questão que vale a pena considerar, no entanto, é a facilidade com que podemos mudar as coisas, uma vez que o cluster esteja em funcionamento. Por exemplo, se um usuário receber um cluster configurado e, em seguida, aparecer alguém e dizer: "preciso de um pacote XYZ construído com a biblioteca EFG versão 1.23", existirá a possibilidade de reprovisionar as coisas a fim de atender tal necessidade, ou haveria uma maneira fácil de adicionar e remover software de um cluster em execução que seja minimamente invasiva?

A resposta é "sim". Antes de descrevermos como podemos organizar um cluster para ser mais maleável, alguma menção sobre provisionamento de pacotes será útil. Três métodos básicos são oferecidos por vários conjuntos de ferramentas:

- *Image Based* – Uma imagem de nó de disco é propagada para os nós na inicialização. Diferentes "rolls" (imagens) podem ser construídos para diferentes pacotes. Um exemplo são os *Rocks Clusters* [1].
- *NFS Root* – Cada nó faz o boot e instala tudo como `NFS root`, exceto para coisas que mudam para cada nó (por exemplo, `/etc`, `/var`). Este sistema pode ser executado com menos disco ou com o disco cheio. Um exemplo é o *oneSIS* [2].
- *RAM Disk* – Um disco RAM será criado em cada nó que mantém uma imagem do sistema em execução. O sistema de disco RAM pode ser criado em modo híbrido, no qual alguns arquivos estão disponíveis através do NFS, e ele pode ser executado com menos disco ou com o disco cheio. Um exemplo é o *Warewulf* [3]. (Uma boa descrição do Warewulf pode ser encontrada na série HPC Admin do Warewulf [4]).

Independentemente do sistema de provisionamento, o objetivo é fazer alterações sem a necessidade de reiniciar os nós. Nem todas as mudanças podem ser feitas sem reiniciar nós (ou seja, mudando o provisionamento subjacente); no entanto, muitos pacotes de aplicativos podem ser adicionados ou removidos sem muita dificuldade se algumas medidas simples forem tomadas.

Dump em /opt

Em quase todos os clusters HPC, os usuários possuem um `/home` compartilhado globalmente, e um `/opt path` global compartilhado também é possível. O NFS é utilizado em pequenos e médios clusters para compartilhar esses diretórios. Em clusters maiores, alguns tipos de sistemas de arquivos paralelos podem ser necessários. Em qualquer caso, sempre existe um mecanismo para compartilhar arquivos em todo o cluster.

O método mais simples é instalar os pacotes em `/opt`. Esta abordagem possui a vantagem de "instalar uma vez e estar disponível em todos os lugares", embora tenhamos que enfrentar alguns problemas com os arquivos de log; no entanto, em geral, este método irá funcionar com a maioria dos aplicativos de software.

A questão principal com a qual os administradores devem lidar é a biblioteca de vínculo dinâmico. Como os pacotes não estão instalados no caminho padrão `/usr/lib` e não desejamos copiar entradas de pacotes em `/etc/ld.so.conf/` nos nós, precisamos de uma forma de gerenciar o local para as bibliotecas. Claro, fazer a vinculação estática completa é uma possibilidade, e usar o `LD_LIBRARY_PATH` é outra, mas ambas as soluções impõem algumas exigências extras sobre os usuários, o que, no final das contas, volta para o administrador de sistemas para suportar quaisquer problemas. O

método preferido é instalar pacotes que "simplesmente funcionam".

A solução é muito simples. Primeiro, crie `/opt/etc/ld.so.conf.d/` para que todos os pacotes posicionem seus caminhos de biblioteca em arquivos `conf`, como fariam em `/etc/ld.so.conf.d/`. Em seguida, é necessário fazer uma pequena inclusão em `/opt/etc/ld.so.conf` para todos os nós (ou seja, precisa ser parte da etapa de provisionamento do nó, por isso está lá após o boot do nó). A linha adicional é:

```
include /opt/etc/ld.so.conf.d/*.conf
```

Esta nova linha informa ao `ldconfig` que deve procurar em `/opt/etc/ld.so.conf.d/` por caminhos de biblioteca adicionais. Se um pacote é adicionado ou removido, tudo o que precisa acontecer é um `ldconfig global` em todos os nós para atualizar os caminhos da biblioteca. Esta etapa pode ser facilmente concluída com uma ferramenta como a `pdsh`. Assim, instalar um pacote global no cluster é tão simples como instalá-lo em `/opt`, com uma entrada em `/opt/etc/ld.so.conf.d/` e execução em um `ldconfig` global.

Se, por exemplo, possuirmos a versão atual do Open MPI instalada e um usuário quiser experimentar as bibliotecas `PetSc` com uma nova versão, poderá facilmente instalar e compilar tudo em `/opt` e terá o usuário executando o novo código sem reiniciar nós ou instruindo-os sobre as nuances da `LD_LIBRARY_PATH`. Agora que obtemos uma maneira de adicionar e remover pacotes facilmente do cluster, precisamos informar aos usuários sobre como usá-los.

Listagem 1: Carregamento de módulos

```
01 if [ -z $NOMODULES ] ; then
02 LOADED=`echo -n
   ➥ $LOADEDMODULES|sed 's/:/ /g'`
03 for I in $LOADED
04 do
05 if [ $I != "" ] ; then
06 module load $I
07 fi
08 done
09 else export LOADEDMODULES=""
10 fi
```

Módulos de ambiente global

Em um artigo da Admin HPC, descrevemos o pacote de módulos de ambiente [5]. (Recentemente observamos que alguns outros autores da Admin HPC também têm coberto o mesmo tema [6]). O uso dos módulos de ambiente [7] fornece fácil gerenciamento de várias versões e pacotes em um ambiente HPC dinâmico. Um dos problemas, no entanto, é como manter os módulos de ambiente quando utilizamos outros nós. Se usarmos o `SSH` para fazer login nos nós, então teremos uma maneira fácil de manter (ou não manter) o módulo de ambiente.

Com um pouco de configuração, o protocolo `SSH` permite a passagem de variáveis de ambiente. Além disso, os módulos atualmente carregados são armazenados em uma variável de ambiente chamada `LOADEDMODULES`. Por exemplo, se carregarmos dois módulos (`ftw` e `mpich2`) e, em seguida, olharmos para o nosso ambiente, encontraremos:

```
LOADEDMODULES=
fftw/3.3.2/gnu4:mpich2/1.4.1p1/gnu4
```

Neste ponto, tudo o que precisamos fazer é incluí-lo em todas as sessões de cluster SSH, e então podemos recarregar o módulo de ambiente. Para passar uma variável de ambiente via `ssh`, tanto o arquivo `/etc/ssh/ssh_config` como o `/etc/ssh/sshd_config` precisam ser alterados.

Para começar, o arquivo `/etc/ssh/ssh_config` precisa ter a seguinte linha adicionada a ele:

```
AcceptEnv LOADEDMODULES NOMODULES
```

Tenha em mente que poderá usar a opção de host no arquivo `ssh_config` para restringir os hosts de receber esta variável. Da mesma forma, o arquivo `sshd_conf` precisa da seguinte linha adicionada:

```
SendEnv LOADEDMODULES NOMODULES
```

Uma vez que o serviço SSHD seja reiniciado, as futuras sessões SSH irão transmitir as duas variáveis para logins SSH remotos. Antes que o login remoto possa usar os módulos, ele deve ser carregado. Este passo pode ser realizado pela inclusão de um pequeno pedaço de código ao script `.bashrc` do usuário, como mostrado na **listagem 1**.

Como pode ser visto a partir deste código, se `NOMODULES` for definido, nada é feito, e nenhum módulo é carregado. Se não for definido, cada módulo listado em `LOADEDMODULES` é carregado. Note que esta configuração assume o pacote do módulo e os arquivos do módulo ficam disponíveis para o nó. Considere o exemplo da **listagem 2**, na qual os dois módulos são carregados (`fftw` e `mpich2`) antes de efetuar login em outro nó (`n0`, neste caso). No primeiro login, os módulos são carregados no nó remoto. No segundo login, com `NOMODULES` configurado, nenhum módulo está disponível: conforme observamos, um pressuposto importante é a disponibilidade dos arquivos de módulo para todos

Listagem 2: Definição de NOMODULES

```
01 $ module list
01 Currently Loaded Modulefiles:
01 1) fftw/3.3.2/gnu4 2) 01
   ➥ mpich2/1.4.1p1/gnu4
01 $ ssh n0
01 $ module list
01 Currently Loaded Modulefiles:
01 1) fftw/3.3.2/gnu4 2) 01
   ➥ mpich2/1.4.1p1/gnu4
01 $ exit
01 $ export NOMODULES=1
01 $ ssh n0
01 $ module list
01 No Modulefiles Currently
   ➥ Loaded.
```

os nós. Ao colocar os arquivos de módulo no NFS compartilhado `/opt`, todos os nós podem encontrar os arquivos de módulo em um só lugar, e eles podem ser adicionados ou removidos sem alterar a imagem em execução no nó.

Em direção ao cluster RPM

O ingrediente final para esta receita é encapsular ambas as ideias em pacotes RPM; ou seja, um RPM vai instalar um pacote em `/opt`, fazer a entrada em `/opt/ld.so.conf.d`, e instalar um arquivo de módulo. Dessa forma, com exceção de um `ldconfig` global, todo o pacote poderia ser instalado no cluster inteiro em uma única etapa. Se o `pdsh` (ou similar) forem necessários como parte do processo de instalação do RPM, o `ldconfig` global poderia ser feito pelos RPMs (da mesma forma que um `ldconfig` local é feito por quase todos os RPMs).

Claro, construir boas RPMs leva algum tempo, mas uma vez que tenhamos o "esqueleto" básico, não será tão difícil arrastá-las para os passos *configure/make/install* para vários pacotes. Uma vez que o usuário possua boas RPMs de cluster para seus aplivativos, no entanto, a instalação e reinstalação é simples, conveniente, e compreende todo o cluster. ■

Gostou do artigo?

Queremos ouvir sua opinião.
Fale conosco em:
cartas@linuxmagazine.com.br
Este artigo no nosso site:
http://lnm.com.br/article/8579

Mais informações

[1] Rocks Clusters: http://www.rocksclusters.org/wordpress/

[2] oneSIS: http://onesis.org/

[3] Warewulf: http://warewulf.lbl.gov/trac/

[4] Gerenciamento de cluster Warewulf: http://www.admin-magazine.com/HPC/Articles/Warewulf-Cluster-Manager-Master-and-Compute-Nodes/

[5] Gerenciamento do ambiente de módulos : http://www.admin-magazine.com/HPC/Articles/Managing-the-Build-Environmentwith-Environment-Modules/

[6] Módulos de ambiente Lmod: http://www.admin-magazine.com/HPC/Articles/Lmod-Alternative-Environment-Modules/

[7] Módulos de ambiente: http://modules.sourceforge.net/