FreeBSD 操作系统在无远程控制台下的远程安装

Daniel Gerzo <danger@FreeBSD.org>

修订：43126

2013-11-07 由 gabor.

摘要

本文归档了当远程控制台不可用的情况下 FreeBSD 操作系统的远程安装。 文章背后的主要灵感归功于和 Martin Matuska 还有由 Pawel Jakub Dawidek 提供的宝贵输入合作的结果。

目录

1. 背景

世界上有很多的服务器主机供应商， 但是他们中只有很少的一部分正式支持 FreeBSD， 他们通常为他们提供的服务器上安装 Linux® 发行版提供支持。

在某些情况下，如果你请求这些公司他们会安装一个你首选的 Linux® 发行版。有了这个选择，我们将试图安装 FreeBSD。 在其他情况下，他们可能提供一个急救系统用于紧急情况。 使用这个可能将有利于我们的目的更好的实现。

本文涵盖了引导一个包含 RAID-1 及 ZFS 性能的 FreeBSD 系统的远程安装的基本安装配置所必须的步骤。

2. 简介

这一节会摘要本文的目的以及更好阐述这里所概括的东西。 本文中的这些指令将有益于那些使用不支持 FreeBSD 的托管设施提供的服务的人。

1. 如我们提到过的 背景 的那一节，许多的有声望的服务器主机托管公司提供了各种的急救系统。 可以从他们自己的 局域网 启动并可以通过 SSH 访问。 他们通常提供这种支持目的用于帮助他们的顾客修正损坏的操作系统。 如文章将说明的，我们将能够通过这些急救系统的帮助来安装 FreeBSD。

2. 文章的下一小节会描述如何配置，并在本地机器上构建最小限度的 FreeBSD。该版本最终会从随机存储盘运行到远程机器上面去。 这将允许我们使用 sysinstall 实用程序从一个 FTP 镜像安装一套完整的 FreeBSD 操作系统。

3. 文章的剩余内容除了描述 ZFS 文件系统的配置还将描述系统本身的安装步骤。

2.1. 需求

想要成功地做下去，你必须：

- 拥有一个可通过 SSH 网络访问的操作系统。
- 理解 FreeBSD 的安装过程
- 熟悉 sysinstall(8) 实用程序
- 拥有 FreeBSD 安张的 ISO 镜像文件或者易于使用的 CD

3. 准备工作 - mfsBSD

在 FreeBSD 可能安装到目标系统上之前， 需要先构建一个最小化的从磁盘启动的 FreeBSD 操作系统映像文件。 此方法中新系统必须能够从网络访问， 并且安装的其他过程能够在没有远程访问到系统控制台的情况下完成。

mfsBSD 设置工具能够被用来构建一个微小的 FreeBSD 映像。如 mfsBSD 名字的含义 （“mfs” 的意思是 “memory file system” 内存文件系统）， 最后的映像全部从随机存储器运行。多亏了这个特性， 磁盘的操作将不会有任何限制，因此它能够被用来安装一个完整的 FreeBSD 操作系统。 mfsBSD 的主页在 http://people.freebsd.org/~mm/mfsbsd/， 包含了指向最新释出的设置工具。

请注意关于 mfsBSD 内幕以及它所有的适用都超出了本文的内容， 感兴趣的读者应该去查阅 mfs 的原始文档得到更多详细内容。

下载并解压出最新的 mfsBSD 版本，并改变自己的当前工作目录到存在 mfsBSD 脚本文件的目录：

```
# fetch http://people.freebsd.org/~mm/mfsbsd/mfsbsd-latest.tar.gz
# tar xvzf mfsbsd-1.0-beta1.tar.gz
# cd mfsbsd-1.0-beta1/
```

3.1. mfsBSD 的配置

引导 mfsBSD 之前， 必须设置一些重要的配置选项。 最重要的是我们必须有正确地，自然地，网络配置。 最适合的方法配置网络选项取决于我们是否事先知道我们会用到的网络接口， 而且网络接口驱动程序应被系统为我们的硬件载入。 我们将看到 mfsBSD 如何能够在任一种情况下被配置。

另外一件重要的事情是设置 root 的密码。 这将通过编辑 conf/rootpw.conf 文件来完成。 请记住该文件将把你的密码保存在简单的文本中， 所以在此我们不推荐你使用真实的密码。然而， 这只是一个临时使用一次的密码，你可以在随后安装好的系统中更改它。

3.1.1. 编辑 conf/interfaces.conf 的方法

如果我们安装好的网卡是未知类型的， 我们可以使用 mfsBSD 的自动探测功能。 mfsBSD 启动脚本能够探测到正确的驱动来使用， 基于网络接口的 MAC 地址，我们假设在 conf/interfaces.conf 文件中设置如下选项:

```
initconf_interfaces="ext1"
initconf_mac_ext1="00:00:00:00:00:00"
initconf_ip_ext1="192.168.0.2"
initconf_netmask_ext1="255.255.255.0"
```

别忘了添加 defaultrouter 信息到 conf/rc.conf 文件中:

```
defaultrouter="192.168.0.1"
```

3.1.2. 编辑 conf/rc.conf 的方法

当网络接口的驱动是已知类型的，使用 conf/rc.conf 文件添加联网选项会更加方便。 该文件的语法跟 FreeBSD 中标准的 rc.conf(5) 文件的语法相同。

例如，当你知道被使用的将是一个 re(4) 网络接口设备， 你可以在 conf/rc.conf 文件中设置如下选项:

```
defaultrouter="192.168.0.1"
ifconfig_re0="inet 192.168.0.2 netmask 255.255.255.0"
```

3.2. 构建一个 mfsBSD 映像

构建一个 mfsBSD 映像文件的过程是非常简单明了的。

第一步是挂载 FreeBSD 的安装 CD， 或者挂载安装 ISO 文件到 /cdrom。 因为例子的缘故，在文章中我们将假定你下载的是 FreeBSD 7.0-RELEASE ISO 文件。使用 mdconfig(8) 实用程序挂载 ISO 映像文件到 /cdrom 目录非常简单:

```
# mdconfig --a --t vnode --u 10 --f 7.0-RELEASE-amd64-disc1.iso
# mount_cd9660 -/dev/md10 -/cdrom
```

紧接着，构建可启动的 mfsBSD 映像：

```
# make BASE=/cdrom/7.0-RELEASE
```

> **注意**
>
> 上面的 make 命令必须在 mfsBSD 目录树的最高一层运行，也就是： ~/mfsbsd-1.0-beta1/。

3.3. 启动 mfsBSD

现在 mfsBSD 映像已经准备好了， 必须把它上传到远程的一个正在运行的急救系统上或者一个预安装了 Linux® 发行版的系统上。最适合做这个工作的工具是 scp：

```
# scp disk.img root@192.168.0.2:.
```

想要正确的引导 mfsBSD 映像， 必须把它安放在机器的第一块（可启动）设备上。这可能会和使用的例子我们假定的一样，第一块可启动磁盘设备是 sda：

```
# dd if=/root/disk.img of=/dev/sda bs=1m
```

如果一切正常，该映像现在应该存在于第一块设备的 MBR（主引导区）而机器也应该能够被启动了。 使用工具 ping(8) 来查看机器是否被正确启动。 一旦它回复在线状态，就应该能够使用 root 用户和配置好的密码通过 ssh(1) 来访问它了。

4. FreeBSD 操作系统的安装

mfsBSD 成功被引导后它就应该能够通过 ssh(1) 登入了。这一节会描述如何创建 slices 并标记 slices 的 label， 为 RAID-1 配置 gmirror， 还有如何使用 sysinstall 来安装一个最小的FreeBSD操作系统版本。

4.1. 准备磁盘

首要的任务是为 FreeBSD 分配磁盘空间，也就是， 创建 slices 和 partitions。很显然， 当前运行的系统是全部被载入到系统内存中的因此操作磁盘将没有任何问题。 要完成这个任务，可以是使用 sysinstall 或者 fdisk(8) 中的二者任一并结合工具 bsdlabel(8)。

在开始时，将所有磁盘都标记成空的， 在每个磁盘上重复如下命令：

```
# dd if=/dev/zero of=/dev/ad0 count=2
```

下面，使用你喜欢的工具创建 slices 并标记磁盘 label。 比较简单的方法是使用 sysinstall， 强大也可能几乎没有漏洞方法是使用标准的基于文本的 UNIX® 工具， 类似于 fdisk(8) 和 bsdlabel(8) 这些工具的使用也会在这一节中包括。前者已经被包括在 FreeBSD 手册的 安装FreeBSD 一章中了。如本节中刚提到的，这篇文章会展示如何设置一个带有 RAID-1 和 ZFS 性能的系统。我们的设置由一个小工具 gmirror(8) 镜像为 / (root)， /usr 和 /var 文件系统， 并把剩余的磁盘空间被分配为 zpool(8) 镜像出的 ZFS 文件系统。请注意， ZFS 文件系统将在 FreeBSD 操作系统成功安装并启动后才会被配置。

下面的例子会描述如何去创建 slices 和 labels， 在每个 partition 上初始化 gmirror(8) 并如何在每个被镜像过的 partition 上创建 UFS2 文件系统：

```
# fdisk --BI -/dev/ad0 ❶
# fdisk --BI -/dev/ad1
# bsdlabel --wB -/dev/ad0s1 ❷
# bsdlabel --wB -/dev/ad1s1
# bsdlabel --e -/dev/ad0s1 ❸
# bsdlabel -/dev/ad0s1 > -/tmp/bsdlabel.txt && bsdlabel --R -/dev/ad1s1 -/tmp/
bsdlabel.txt ❹
# gmirror label root -/dev/ad[01]s1a ❺
# gmirror label var -/dev/ad[01]s1d
# gmirror label usr -/dev/ad[01]s1e
# gmirror label --F swap -/dev/ad[01]s1b ❻
# newfs -/dev/mirror/root ❼
# newfs -/dev/mirror/var
# newfs -/dev/mirror/usr
```

❶ 在整个磁盘上创建一个 slice 并初始化包含在磁盘第一个扇区启动代码。 重复在系统上全部的磁盘上执行此命令。

❷ 为每块磁盘写入一个包括启动代码的内容的标准 label。

❸ 现在，手动去编辑磁盘的 label。可以查阅 bsdlabel(8) 的联机手册来找到如何建立 partitions 的方法。创建如下 partions，a 为 / (root) 文件系统， b 为 swap 交换空间， d 为 /usr 还有最后 f 被用于 ZFS。

❹ 引入你刚才创建的 label 到第二块磁盘， 所以两块磁盘会使用同样的 label。

❺ 在每个 partition 上初始化 gmirror(8)。

❻ 注意 -F 选项被用在 swap 交换分区的 partition。 gmirror(8) 这个指令认为设备处于可靠的状态除非电源系统故障。

❼ 在每个被镜像的分区上创建一个 UFS2 的文件系统。

4.2. 系统安装

这是最重要的一部分。 此节将描述如何在我们上一小节已经准备好的磁盘上安装一个最小的 FreeBSD 版本。要达成这个目的，所有的文件安系统需要被挂载乃至于 sysinstall 可以把 FreeBSD 系统的内容写到磁盘上：

```
# mount -/dev/mirror/root -/mnt
# mkdir -/mnt/var -/mnt/usr
# mount -/dev/mirror/var -/mnt/var
# mount -/dev/mirror/usr -/mnt/usr
```

当你做完这些时，打开 sysinstall(8)。 从主菜单选择自定义 Custom 安装。 选中 Options 选项然后按回车确认。 使用方向键获取帮助，移动鼠标指针到 Install Root 选项，按 空格 更改为 /mnt。 按 回车 提交你的更改并使用 q 退出 Options （选项） 菜单。

> **警告**
>
> 注意这一步骤非常重要，如果被跳过了， sysinstall 将不能安装 FreeBSD。

到 Distributions （发行版） 菜单选项， 使用方向键移动鼠标指针到 Minimal （最小化） 选项， 并使用 空格键 选中该选项。 本文使用了最小版本来保存网络联通信息，因为系统本身会通过 ftp 来安装。使用 Exit （退出） 选项退出这个菜单。

> **注意**
>
> Partition 和 Label 菜单将被跳过， 这些没有多少价值了。

Media （媒介） 菜单， 选择 FTP 选项。 选择一个距离你最近的镜像站点并交给 sysinstall 假定网络已经配置完好。你将再回到 Custom （自定义） 菜单。

最后，选择最后的选项来执行系统的安装过程， Commit， 当安装完成后退出 sysinstall 即可。

4.3. 后期安装步骤

FreeBSD 操作系统现在应该安装完毕了；通常情况下， 安装过程还没有结束。还需要进行一些安装后期的步骤使得容许 FreeBSD 在将来启动并能够登入系统。

你现在必须 chroot(8) 到刚安装的全新的系统中来完成安装。 使用如下命令：

```
# chroot -/mnt
```

要达到我们的目的，进行如下步骤：

- 拷贝 GENERIC （通用） 内核到 /boot/kernel 目录：

```
# cp -Rp /boot/GENERIC/* /boot/kernel
```

- 创建 /etc/rc.conf， /etc/resolv.conf 还有 /etc/fstab 文件。 不要忘记正确地设置网络信息并在 /etc/rc.conf 文件中启用 sshd。 /etc/fstab 文件内容类似于下面的内容：

```
# Device          Mountpoint    FStype  Options      Dump    Pass#
/dev/mirror/swap     none         swap    sw           0       0
/dev/mirror/root     /            ufs     rw           1       1
/dev/mirror/usr      /usr         ufs     rw           2       2
/dev/mirror/var      /var         ufs     rw           2       2
/dev/cd0             /cdrom       cd9660  ro,noauto    0       0
```

- 创建 /boot/loader.conf 文件，并写入如下内容：

```
geom_mirror_load="YES"
zfs_load="YES"
```

- 执行下面的命令，使得 ZFS 在下次启动后可用：

```
# echo 'zfs_enable="YES"' >> /etc/rc.conf
```

- 可以用 adduser(8) 工具来添加额外的用户。 不要忘记添加一个用户到 wheel 组，这样你可以在重新启动后获得 root 权限。
- 反复检验你的设置是否正确。

现在你的系统在下次启动后应该可用了。使用 reboot(8) 命令重新启动你的系统。

5. ZFS

如果你的系统重新启动后还完好，现在应该能够登入了。 欢迎来到崭新的 FreeBSD 安装，进行远程的不使用远程控制台的安装。

最后还剩下的步骤是配置 zpool(8) 并创建一些 zfs(8) 文件系统。建立并管理 ZFS 非常简单。 首先，创建一个镜像的pool：

```
# zpool create tank mirror /dev/ad[01]s1f
```

再接着,创建一些文件系统：

```
# zfs create tank/ports
# zfs create tank/src
# zfs set compression=gzip tank/ports
# zfs set compression=on tank/src
# zfs set mountpoint=/usr/ports tank/ports
```

```
# zfs set mountpoint=/usr/src tank/src
```

这就是全部步骤了。如果你对 FreeBSD 上的 ZFS 感兴趣，请查阅 FreeBSD WIKI 中的 ZFS 一节。