Enabling Serial Port Console Redirection on Linux-based Dell Servers

Console redirection allows administrators to monitor and manage servers from a remote location by redirecting keyboard input and display output through the serial port. This article introduces the concept of serial port console redirection for servers running the Linux® operating system (OS), and then details how to implement this functionality for the BIOS, boot loader, kernel, and OS. In addition, this article examines the unique features in version 8.0 of the Red Hat® Linux OS that help simplify the configuration of serial port redirection.

BY CHRISTOPHER STANTON

Many companies and IT departments migrating from or considering migration from proprietary UNIX® platforms to the Linux® operating system (OS) contemplate more than just the potential cost savings. They look for features that enable a Linux-based server to operate seamlessly within the existing and well-established UNIX-based data center. One feature these organizations often require is serial port console redirection, a technique that redirects the main console (keyboard input and display output) to a server's serial port.

UNIX servers are designed to send console, POST, and BIOS information through the serial port. However, most x86-based platforms are not set up to fully support serial consoles. UNIX systems are designed as multi-user systems, and the main console on the UNIX server shows all system messages, independent of the users; x86-based hardware was originally intended for single-user systems and does not have a separate main console. Because Linux runs on x86-based hardware, administrators must use serial port console redirection to make a Linux server behave like a UNIX server.

Using console redirection, administrators can remotely manage servers. Console redirection also helps eliminate the need for individual monitors and keyboards for each server. This benefit is particularly useful in high-density computing installations, such as high-performance computing clusters, because administrators can monitor multiple remote consoles from one location. This article explores the procedure for enabling serial port console redirection on Linux-based servers and then examines the new features of Red Hat® Linux version 8.0 that help make this redirection easier to implement.

Using serial port console redirection for monitoring and managing servers

Using the serial port to access the console is considered out-of-band because it occurs outside the normal network access point of the server. The serial port can either be

connected directly to another machine's serial port using a null modem cable or, more commonly, can be attached to a serial port concentrator. A serial port concentrator acts as a network switch for serial ports. The serial ports of the servers being monitored are connected to the concentrator, and the concentrator can be connected directly to the monitoring station or can be uplinked to a standard network switch. In the latter case, IT administrators can use multiple machines to monitor and manage the redirected servers.

If network connectivity to the server is lost, the server can still be accessed through the serial port console. Thus, console redirection allows administrators to diagnose and debug problems, even if those problems limit the ability to reach the server directly. In the rare occurrence of a hung system or kernel panic, the error messages are printed to the console, allowing administrators to copy the error messages or kernel dump text and save them remotely from the serial console screen. This capability aids further analysis and bug reporting. And because this information is in plain-text form, administrators can include it directly in the body of an e-mail or feed it to ksymoops, a utility that decodes the Linux kernel oops file, once the machine has been rebooted.

Redirecting the BIOS output to the serial port

Most recent Dell™ PowerEdge™ servers support BIOS-level serial port redirection. This redirection capability allows the BIOS to send its output stream to the VGA terminal as well as through the serial port as a pure text-mode display. Earlier PowerEdge models required that the administrator hard-code the speed at which to send the stream. Many of today's PowerEdge servers auto-detect the speed at which the receiving connection is configured and automatically match that line speed. This capability allows administrators to focus only on the connection speed for the boot loader, kernel, and OS and eliminates the need to modify the BIOS settings in the future.

Besides enabling and disabling the connection and possibly setting the speed of the connection, administrators can configure the formatting of the display itself and whether the BIOS should redirect the display after the BIOS has finished loading. Enabling the BIOS to continue redirecting the display after the BIOS has loaded will not redirect the OS through the serial port. At that point, the BIOS has relinquished control of the hardware and the OS is operating independently.

Serial port console redirection helps eliminate the need for individual monitors and keyboards for each server.

Key sequence	Key mapping for redirected console
<F10>	<ESC><0>
<Ctrl><M>	<ESC><Ctrl><M>
<Ctrl><H>	<ESC><Ctrl><H>
<Ctrl><I>	<ESC><Ctrl><I>
<Ctrl><J>	<ESC><Ctrl><J>
<Alt><*x*> (where *x* is any letter key, and *X* is the upper case of that key)	<ESC><*X*><*X*>
<Ctrl><Alt><Del>	<ESC><R><ESC><r><ESC><R>

Figure 1. Key mapping for serial port console redirection on the PowerEdge 2650 server

Choosing to redirect the display allows the BIOS to redirect the output of programs such as the Preboot Execution Environment (PXE) loader distributed with the built-in Gigabit Ethernet[1] network cards of certain PowerEdge servers. If this option is left disabled, the loading of the PXE loader and its attempts to contact a Dynamic Host Configuration Protocol (DHCP) server will not be redirected through the serial port.

The BIOS console displayed through the serial port is the text-mode boot display, and the terminal emulation choices are "ANSI" or "VT-100/VT-220." Administrators can watch as the BIOS initializes various devices such as SCSI, RAID, and network interface cards (NICs), and they can access the corresponding configuration programs.

Access to the embedded BIOS configuration menu is also possible through serial port redirection. During normal operation, administrators can gain access to this menu by pressing the F2 function key while the BIOS is booting. However, the ANSI, VT-100, and VT-220 terminal specifications do not support function keys, so the BIOS supports an alternative key combination for access to the BIOS configuration menu. Figure 1 shows the available alternative key combinations for the PowerEdge 2650 server.

Redirecting output from the boot loader and kernel

The next program that will need its output redirected through the serial port is the boot loader. The boot loader determines which OS, and in the case of Linux, which configuration of the OS, to load. GRUB is the default boot loader installed on Red Hat Linux 7.2 and later releases. It supports many advanced features, including serial port redirection of its output. Figure 2 shows a sample GRUB configuration file.

Administrators must configure the `serial` and `terminal` lines so that GRUB's full functionality will be available through the serial port (see Figure 3). The `unit` option of the `serial` command specifies which serial port to use. Unit 0 corresponds to COM1, unit 1

[1] Gigabit Ethernet indicates compliance with IEEE® 802.3ab and does not connote speeds of 1 Gbps.

```
default=0
timeout=10
splashimage=(hd0,2)/grub/splash.xpm.gz
title Red Hat Linux (2.4.18-4smp)
     root (hd0,2)
     kernel /vmlinuz-2.4.18-4smp ro root=/dev/sda9
     initrd /initrd-2.4.18-4smp.img
```

Figure 2. Sample GRUB configuration file for Red Hat Linux version 7.3

```
#ADD SERIAL PORT REDIRECTION FOR GRUB
serial –unit=0 –speed=115200
terminal –timeout=10 serial console
default=0
timeout=10
#COMMENTING OUT THE GRAPHICAL DISPLAY LINE
#splashimage=(hd0,2)/grub/splash.xpm.gz
title Red Hat Linux (2.4.18-4smp)
     root (hd0,2)
     kernel /vmlinuz-2.4.18-4smp ro root=/dev/sda9
     initrd /initrd-2.4.18-4smp.img
```

Figure 3. Configuring GRUB for serial port redirection

```
serial –unit=0 –speed=115200
terminal –timeout=10 serial console
default=0
timeout=10
#COMMENTING OUT THE GRAPHICAL DISPLAY LINE
#splashimage=(hd0,2)/grub/splash.xpm.gz
title Red Hat Linux (2.4.18-4smp)
     root (hd0,2)
     kernel /vmlinuz-2.4.18-4smp ro root=/dev/sda9
        console=ttyS0,115200
     initrd /initrd-2.4.18-4smp.img
```

Figure 4. Overriding the use of the default console

to COM2, and so on. The `speed` option of the `serial` command sets the speed that should be used when communicating through the port in bits per second (bps). The `terminal` command specifies how many seconds to wait for user input before displaying the GRUB boot menu through the default interface. The `serial` and `console` options of the `terminal` command specify available interfaces. Because the `serial` option is listed first, it is treated as the default interface. The boot menu is only available through the main VGA console if input is received from the local keyboard before the timeout expires.

GRUB has two display modes: text and graphical. The graphical display mode is the default mode for GRUB when it is installed during the Red Hat Linux OS installation process. For serial port redirection, administrators should configure GRUB to use its text mode because the graphical display cannot be viewed through a serial connection and will cause the boot selection menu to be unusable. Commenting out the `splashimage` graphical display line in /etc/grub.conf will cause GRUB to default to text mode (see Figure 3).

The GRUB configuration file is also the location where kernel command-line parameters are specified. In Figure 2 and Figure 3, the root partition location (`root=/dev/sda9`) and read-only (`ro`) kernel parameters were passed to the kernel. The Linux kernel also supports, among many other parameters, the `console` argument, which allows overriding the usage of the default console tty0 (the first VGA terminal). The console parameter is in the form of either `console=`*`device`* or `console=`*`device`*`,`*`speed`* where *speed* is in bits per second. To create a console on ttyS0 (or COM1 in DOS/Microsoft® Windows® operating systems) with a line speed of 115200 bps, a `console=ttyS0,115200` parameter must be passed on the kernel append line (see Figure 4).

```
serial –unit=0 –speed=115200
terminal –timeout=10 serial console
default=0
timeout=10
#COMMENTING OUT THE GRAPHICAL DISPLAY LINE
#splashimage=(hd0,2)/grub/splash.xpm.gz
title Red Hat Linux (2.4.18-4smp)
     root (hd0,2)
     kernel /vmlinuz-2.4.18-4smp ro root=/dev/sda9
        console=tty0 console=ttyS0,115200
     initrd /initrd-2.4.18-4smp.img
```

Figure 5. Enabling the Linux kernel to use multiple consoles

The kernel is not restricted to using one console for logging messages. All that is required to enable kernel messages on the main video screen as well as through the serial port is an additional console argument (see Figure 5). Note that the tty0 console was declared before the ttyS0 console—the last console declared is treated as the master console.

Although the kernel can multiplex its output to multiple display devices, user-space programs write only to standard output (stdout). The kernel maps stdout to the master console, restricting user-space programs, such as init (the process that brings up the network, starts services such as the Apache and Sendmail® programs, and prepares the machine's state for the users), to a single display console. If the consoles are declared in the order shown in Figure 5, kernel messages displayed while the kernel loads will appear on both tty0 and ttyS0. When the kernel starts init, however, messages and errors produced in user space will appear only on the master console (ttyS0 in Figure 5).

If network connectivity to the server is lost, the server can still be accessed through the serial port console. Thus, console redirection allows administrators to diagnose and debug problems, even if those problems limit the ability to reach the server directly.

Creating a Linux console accessible through the serial port

Up to this point in the configuration procedure, the BIOS, kernel, and init program have redirected their displays through the serial port, allowing an administrator to monitor and interact with the full boot process. The only step remaining after Linux has fully booted is the creation of a login console accessible through the serial port. Creating this console requires a straightforward and relatively easy modification to two Linux configuration files: inittab and securetty.

The inittab configuration file is used by the init daemon and describes the particular services to start at each runlevel. Service or process declarations are in the form *ID:runlevels:action:process*. *ID* is a unique sequence of no more than four characters and is used to identify an entry. The *runlevels* field is a string of runlevels on which to start the process. For example, the string "234" would start the process on levels 2, 3, and 4. Linux has seven user runlevels that correspond to states such as single-user mode, multiuser mode, and graphical mode.

The *action* field describes how the process should be handled. Two example values for this field are `once` and `respawn`. The `once` action tells init to start the process once when one of the selected runlevels is entered. If `respawn` is chosen, init will restart the process whenever the process terminates. The *process* field provides the command to run and the parameters or settings that should be passed to the process.

To create a console on the first serial port, ttyS0, a new service description should be added to /etc/inittab. Although mingetty is used for the standard display consoles (see Figure 6), this process is not intended or suited for managing a console routed through the serial port. The process that should be used to manage the serial port console is either agetty or mgetty. Adding a `0:12345:respawn:/sbin/agetty ttyS0 115200` line to inittab will cause agetty to be started on all the standard runlevels as well as in single-user mode (see Figure 7). The console will be broadcast through ttyS0 at 115200 bps. If the agetty exits or is terminated for some reason, init will restart the process with the same parameters. The zero character (0) was chosen as the serial port's ID because the inittab man page states that, for getty processes or other login processes, the *ID* field should be the tty suffix of the corresponding TTY. For example, the ID should be 1 for tty1. This method for determining the ID should enable login accounting to work correctly.

```
# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

Figure 6. Inittab file used for the standard display consoles

```
# Run gettys in standard runlevels
0:12345:respawn:/sbin/agetty ttyS0 115200
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

Figure 7. Inittab file used for a console routed through the serial port

After the inittab file has been modified, users can log in to the server through the serial port. However, root login is still not available because it is restricted to consoles listed in the /etc/securetty security file. To enable root access, administrators can simply append ttyS0 to this file. Once this last step is completed, full serial port redirection will be enabled.

Exploring the new features of Red Hat Linux version 8.0

Version 8.0 of the Red Hat Linux distribution adds extra intelligence to the installation system. If a serial console is defined as one of the display units on the installation kernel's command line, the installation program will automatically set up the server very similarly to the serial port redirection procedure detailed in this article. This automatic setup usually occurs during installation of the OS over the network using PXE. Figure 8 shows an example pxelinux config file. In the first line of the file, the pxelinux loader is set to display its output through the serial port.

The installation program detects the console parameter `console=ttyS0,115200` being passed to the kernel on the kernel append line and then uses that parameter to configure the installed machine for console redirection. A console parameter can be in the form of `console=`*`device`* or `console=`*`device`*`,`*`speed`*. The first form is generally used for TTY devices while the second is used for serial ports.

One difference in Red Hat Linux version 8.0, compared to the serial port redirection procedure discussed earlier in this article, is the /etc/inittab file. The installation program in version 8.0 will set the serial console to use VT-100 terminal emulation by adding a VT-100 terminal parameter to agetty. This setting overrides the default terminal settings of init, which starts a "linux" terminal by default. The "linux" terminal has greater functionality than a VT-100 terminal; it supports special keys such as the arrow keys, Home key, and End key. If the display terminal or emulator being used to access the output of the server's serial port is not restricted to only VT-100, removing the VT-100 terminal type passed to agetty in /etc/inittab may be beneficial.

```
serial 0 115200
default localboot
prompt 1
timeout 200

label localboot
    localboot 0

label rh80
    kernel vmlinuz
    append ks=nfs:10.0.0.1:/home/nfs/ks.cfg
       ksdevice=eth0 ramdisk_size=16384
       initrd=initrd.img lang= devfs=nomount
       console=ttyS0,115200
```

Figure 8. Example of a pxelinux configuration file

Configuring a Linux-based Dell PowerEdge server to provide full functionality and access through one of its serial ports is a straightforward and not significantly time-consuming procedure.

The other difference in Red Hat Linux version 8.0 is that even if multiple console arguments are passed on the kernel command line, only the master console—the console listed last—is enabled on the installed server. If the kernel messages must be displayed on the video console, administrators must manually add that console's argument to the kernel append line in the GRUB configuration file.

Simplifying serial port console redirection

Configuring a Linux-based Dell PowerEdge server to provide full functionality and access through one of its serial ports is a straightforward and not significantly time-consuming procedure. Red Hat Linux 8.0 makes the process even easier by providing seamless auto-configuration of the deployed servers if the environment is configured to use serial port console redirection. Furthermore, serial port console redirection can help organizations more easily integrate Linux-based servers into their UNIX-based environments.

Christopher Stanton (christopher_stanton@dell.com) is a senior systems engineer in the Scalable Systems Group at Dell. His interests include Linux as well as high-performance computing cluster (HPCC) installation, management, and performance benchmarking. Christopher has a B.S. with special honors in Computer Science from The University of Texas at Austin.

FOR MORE INFORMATION

Dell and Linux: http://www.dell.com/linux