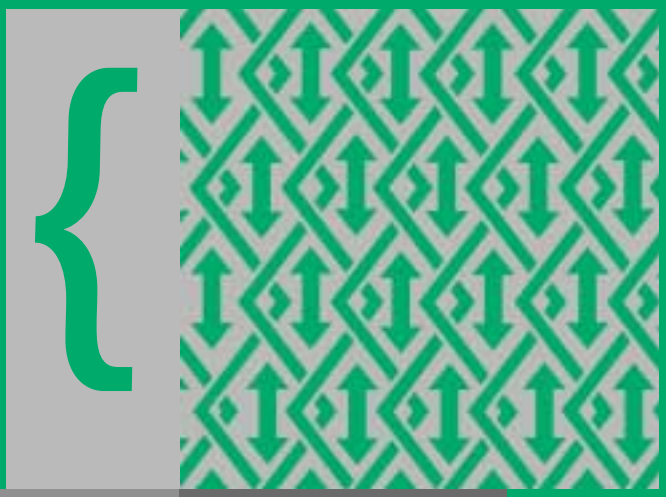
;login:

THE MAGAZINE OF USENIX & SAGE
October 2000 • volume 25 • number 6



inside:

SYSADMIN
Finding Time to do it All

USENIX & SAGE
The Advanced Computing Systems Association &
The System Administrators Guild

finding time to do it all

by David K. Z. Harris

David K. Z. Harris has been a network plumber "for more than a decade." He's been a member of the technical staff at Certainty Solutions for nearly three years.

<zonker@certaintysolutions.com>

and Bryan Stansell

Bryan Stansell is one of the earliest members of Certainty Solutions staff, and is the current keeper of the Conserver code.

<bryan@certaintysolutions.com>

David Stuit, and Michael Batchelder, both of GNAC, also provided substantial material for this article. (In August 2000, Global Networking and Computing (GNAC) became Certainty Solutions.)

During the past few years, we have seen an explosive growth in the number of companies that depend on the Internet for the success of their business. For those of us who work to support the systems and networks, we can look forward to stable employment. But the success of our companies means more work to do, and it is becoming harder to find additional qualified technical people to hire. As a result, we need to learn to do more with less. This article will tell you about one tool that saves me time, reduces downtime, speeds troubleshooting, and aids in the training of new staff. (Oh yeah, it's also free!)

Thomas A. Fine and Steven Romig presented a client-server application called Conserver at LISA IV (Fall 1990[1]). Conserver allowed administrators around the network to have access to serial-console ports using the Telnet protocol between the client and server, and the server would log all of the session activity coming from each console. This application saved Ohio State University space, power, and cooling by removing many of the monitors and keyboards that used to be attached to the servers in their data center. It also saved them a great deal of running to the server room every time they needed to make a change on the physical console! I'm happy to report that Conserver has been getting better over time.

During the past decade, the Conserver application has evolved[2], based on the needs and feedback of the users. Terminal servers, as a product class, have also evolved during this decade, and many now allow Telnet-like (socket-based) and secure-shell (ssh) access from networked hosts to individual serial ports. This extends the reach of the administrators to the far corners of their network, including control of devices that do not have network connections (such as CSU/DSU equipment, test and diagnostic devices, to name only a few).

Conserver is distributed freely[3], but the serial ports you use with it, unfortunately, are not. While a terminal server can provide remote access to serial-console ports, the cost-per-port to deploy them has been high, and is currently still higher than the cost of a 10/100 switch port. There are many advantages to be gained by deploying terminal servers in your network, and I will outline some of the best reasons in this article. I have found the benefits well worth the cost of deployment, especially in terms of the speed of recovery from outages. And in today's e-commerce world, some outages can be downright costly, in terms of lost sales, not to mention customer perception.

When your hosts won't boot, access to the serial-console port becomes invaluable, but the serial console can also be useful in your change-control procedures. Once your hosts are up and running, the administrator(s) will have many options for remote access to them to control their functions. But those avenues of access can mean that multiple administrators (or users with administrative access) will have simultaneous ability to the change settings on the host. This suggests that you also need to consider good change-control practices, to prevent many administrators from making changes on top of each other. If the administrators all share access to a single console of a host, they will know whether they have exclusive access to make their changes. Conserver can help whether the host is up or down.

Considering the Alternatives

You may have a bunch of hosts connected to monitors and keyboards. The monitors all take up space, use power, and create extra heat when they are turned on. If you aren't watching all the time, you probably rely on the scrollback in some of the GUI windows

on each host. The cost of each 15" screen and keyboard approaches the cost for a terminal-server port. In a large data center, add in the cost of air conditioning capacity for all those monitors. (You probably don't want more than a few display devices in a large data center.)

Keyboard/video/mouse (K/V/M) switch systems allow you to connect many hosts to a single display device. You'll find the cost of K/V/M ports are close to the cost of a terminal-server port if you are attaching PCs, and the cost is significantly higher to attach UNIX workstations (Sun, SGI, etc.) to a K/V/M switch. Don't forget the complication of screen resolutions, and scan rates! You can't get more than 12 to 18 devices on a big switch. While you can cascade the switches, it still doesn't scale for large data centers. You can find more information about K/V/M switches from the Celeste Stokely System Administration Web site.[4, 5]

VT-type terminals (or a terminal emulation window on a host) connected to a multiport switchbox give you text-only (CLI) support for much less money than a K/V/M, but you don't get GUI access when the host is up. The average cost for switchboxes with eight ports will cost you about the same as one terminal server port. Your limitations here include the physical distance for the serial console lines between hosts and the switchbox, as well as the number of devices you can attach to the switchbox. You can cascade these devices to gain more ports per terminal, but the length limitation is still there, plus you need to know which switch combinations will connect you to the port that you want.

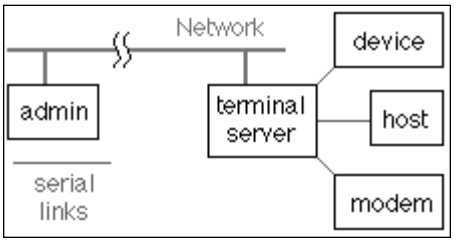
Terminal servers generate less heat than your average monitor, take up one network port, and allow you to access any serial port that you connect from anywhere on your network via a Telnet session. Most units take only one to rack units of vertical space to mount. Many sessions can be open to different ports at the same time, unlike most of the other alternatives above. (There are many Web sites that show how to connect various devices to various vendors' terminal servers.[5, 6]) The disadvantages to using terminal servers include the price, and the fact that there is no logging inherent in the terminal servers. If you were not connected to a console and watching, the data coming from each connected host is lost.

Adding the (free) Conserver application to a deployment of terminal servers adds the logging option, as well as adding a mentoring capacity, provides access control (and auditing of access) to the console ports, thus increasing the overall value of the deployment to the administrators.

The Value of Remote Console Access

If you are one of a few (or the only one) who carries a pager to respond to outages around your network, then remote access to your consoles should be one of the tools at your disposal. This gives you the ability to sit at any workstation (even dialed in from home) and be on the physical console of any connected host. This gives you faster visibility into your problem, and faster time to resolution, than if you needed to return to the data center to get to a terminal or keyboard. Do you remember a time when you were less than ten minutes from home, your pager went off, and you had to fight traffic back to the office to fix some problem? What would it be worth to you to keep driving home and get on the console remotely, with your dinner at your side?

Let's consider the benefits of deploying terminal servers for remote access to consoles, then we'll discuss the extra benefits of adding the Conserver software.



terminal server (no conserver)

Terminal Server-Only Communications Overview

- Device console ports are attached to terminal servers on the network.
- The administrator uses a Telnet session to connect to a specific serial port.
  - Telnet session is from admin's system to the terminal server.
  - Only one session at a time can be connected to each serial port.
  - If no session is connected to a port, data from the attached device is lost.

Imagine that you get home and access the console on the downed host. You hit return, and there is no response. If there is someone at the office whom you can call, you can ask them to cycle the power on your host, while you stay at home and watch the boot cycle. (Even if you don't have after-hours staff in the office there are power strips with serial ports[7], and/or Ethernet ports with Telnet and/or HTTP interfaces[8], to let you control power to individual outlets. You can cycle the power yourself, from home!)

Do you need to escort administrators into restricted areas, just for simple machine reset? We use console access to allow system administrators to control their hosts that need to live in a lab with restricted physical access. The ability to provide them with remote access to power-cycle the device as well as control the console has eliminated the need to allow nonessential staff to have physical access to security-sensitive areas.

While most UNIX hosts provide a method to use serial consoles instead of a video display and keyboard, your average PC platform does not have this built into the BIOS. While a UNIX kernel running on a PC will eventually allow you to use a serial port for a console, it isn't active until the kernel has booted, so you normally cannot see the Power-On Self Test information. But PC platform users do have some options!

Some PC server makers are modifying the BIOS to provide the ability to redirect the results of the Power-On Self Test (POST) to a serial port. Network Engines has recently added this, while Compaq has had this ability for about a year. Hewlett-Packard has an optional system management card that you can add to their servers. There are no standards for this type of console access, so the features for each vendor are different, and you should ask your vendor about the features they offer.

Tip: If your server vendor includes console redirection in BIOS, you should understand the limitations of their implementation. For example, the Network Engines BIOS hands off the startup to intelligent hard-disk controllers and Ethernet NICs, and the output from those interfaces is *not* redirected. If there are problems posted by those controllers, you will not see them through BIOS redirection in the current implementation.

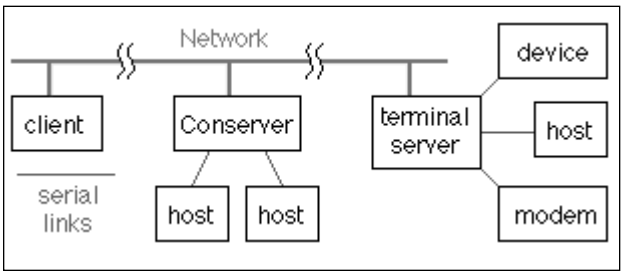
If you use a PC that doesn't have console support in the BIOS, but it does have a spare ISA slot, you can consider using the PC Weasel 2000 add-in card[9]. This appears to the server as a monochrome display adapter, but it translates the characters sent to the pseudo-screen into characters on a serial port. The PC Weasel also includes hardware that senses when your OS tries to use the serial port for its console, and it connects the serial port on the card to the operating system. The card also monitors the serial stream for a special sequence coming into the console, which allows the administrators to talk to the PC Weasel again to restart the PC (or the PC Weasel). Their Web site has an online demo.

Added Value from Free Software

Let's visit that downed host again. You connected to the console port, hit return, and nothing came back. What caused it to stop? If nobody was connected and watching, anything the host had sent to the console was lost. The logging capability of a console server application is one of the best reasons to combine it with your terminal servers.

Terminal Servers with Conserver Overview

- Terminal servers are deployed near the devices you want to monitor.
- The Conserver host is configured for the devices you want to monitor.
  - Hosts can attach to serial ports on the Conserver host.
  - Hosts can attach to serial ports on the terminal servers.
  - Conserver opens a socket-based session to each terminal server port.
- The client application connects to the Conserver via Telnet today (optionally by SSL soon).
  - The client session requests to be connected to a host logging session.
  - Many clients can connect to the same device session simultaneously.
  - Clients can connect to multiple distributed Conserver hosts.



conserver overview

With Conserver in this equation, you connect to the downed host, press return, and get nothing back. Then you use a few meta-keystokes to replay the last 20 or 60 lines of the log for that device to your session. This will usually tell you why it stopped (bad memory, full disk, other problems), and this may affect your decision to just cycle the power. You may decide that something needs to be fixed before you bring it up again.

You can deploy terminal servers, and distributed console servers, across the country, and even internationally. This can be an important tool if you have to support smaller, remote offices without administrators onsite. Sure, you can preconfigure a new host for a remote office, and even set it up in a test lab. You can include diagrams and documentation for the remote-office staff, directing them how to unpack it and plug it in. But when they follow your instructions, and you can't connect to it, what do you do? Before you can talk someone through fixing the problem, you need to troubleshoot it.

- Is it powered?
- Is the network connection plugged into the correct interface?
- Is there a duplicate network address in the office?
- Did the network switch auto-negotiate to the wrong speed?
- Was the disk damaged in shipment?
- What will the remote office folks use for a console, in order to be your eyes?

If you had it plugged into a terminal-server port in that office, you could get most of those answers yourself, rather than talking the office staff through the troubleshooting steps. (Now how much is that terminal server worth to you?)

You can find links to a number of console server applications on the Console Connection Guide Web pages[6], but my favorite application is currently Conserver for a number of reasons:

1. Conserver allows you to control devices attached to serial ports connected to the Conserver host, as well as to serial ports on terminal servers scattered throughout your network.

2. Everything that comes out of a device's console port goes into the Conserver log file for that device.

3. Single-user control, but multi-user viewing, makes Conserver an excellent mentoring tool. Control can be easily switched between users, allowing collaboration between administrators in different locations.

We have used Conserver during maintenance downtimes, along with a conference call, to allow a standby administrator to hear and watch the progress of an upgrade

session. This was done because the person making the changes was across the country, while the standby person was at his desk in the same building as the equipment, in case the hardware failed during the changes.

4. The log files for various devices can also be used as a teaching tool, as a junior administrator can look through sessions to see how a senior administrator performed some tasks, providing that your security policies and the file permissions allow this. (Conserver logs the data coming in from the attached host, not the data going from clients to the host.)

You can also use Conserver logs for training after a failure. If a junior-level administrator started the troubleshooting, and a senior had to come in later and follow up, both administrators can benefit. The senior can look at the logs and see what was already done (including looking to see what happened before the failure). The junior can watch the senior and perhaps learn new methods to use the next time.

5. Scripting tools can also sift through the logs, looking for problems, as a backup to your other automated device managers. These includes SWATCH10, presented at LISA in 1993.

6. The logs can also provide additional logging, as a backup to your syslog files. (A cracker can find out where you are sending syslog messages, but the host doesn't require any special pointers to the Conserver host in order to be connected and logged, so a cracker would not know there was another log to clean up after an intrusion.)

We have found syslog settings misconfigured on some hosts more than once using the Conserver logs. While one machine kept quietly rebooting at random intervals, the administrators found nothing in the syslog files, yet the conserver logs told two stories: first, the cause of the rebooting was a failure to write to a full file system; second, the fact that syslog wasn't logging the errors led us to check the syslog configurations.

7. If a device attached to your Conserver doesn't timestamp its output, Conserver can be set up to put an hourly timestamp in the log for that device.

8. Multiple distributed Conserver hosts can be controlled by a single configuration file, which specifies which servers are connected to the various hosts and devices. This provides easy configuration, even when you deploy Conservers in remote offices.

Tip: Deploying a Conserver host at each remote office means that console data won't be lost if a WAN link fails. If you allow remote access to the remote office, you can look at the logs during the link failure to see what activity may have happened and keep an eye on your link providers work during the repair.

If the WAN link to a remote office goes down, you can still dial into a modem attached to the terminal server in that office, pass the authentication challenge, allowing the network administrator to look at the CSU/DSU and router on both ends of the failing link at once.

9. You can't use grep(1) on a pile of paper on the back end of a DECWriter! But you can use it on the log files from Conserver.

Tip: As with syslog and backups, keeping the clocks between all of your Conserver-connected systems in sync is very important. We recommend a stable NTP infra-

structure be in place when you deploy a console server. During large problem events (denial of service attacks, network outages, cascading failures across multiple servers), your troubleshooting will be faster if all of the clocks are in sync. This allows you to correlate time stamps between various hosts and network devices, and to understand what happened first and what happened after that. (Consider using one time zone for all hosts if making the time-zone conversions for troubleshooting is a concern for you.)

The maintainer of Conserver and I will present a half-day tutorial about deploying remote serial console access at the LISA conference[11] in December. We invite anyone with an interest in the topic to sign up. We will be covering several models for deploying Conserver, and we'll try to take the mystery out of connecting various serial devices to your terminal servers.

There are also options that you can use for added security, but architecture also plays a part in that discussion. Deployment models vary, depending on the security needs. Bring your questions to LISA, and look for the Conserver BoF session!

What about the BREAK Problem?

If you have Sun hosts, you may have been warned away from attaching terminal servers for remote access. Let me offer a brief explanation and some information that may help.

There is a serial-port equivalent to the Telnet BREAK signal, where the data lead signal is inverted from its normal state for a brief period of time (more than the duration of a single character, including start and stop bits)[12]. This is not the same as sending control characters. Most terminal servers send a serial BREAK to every port when you turn the power off, and some even do it when you turn power on, or during their boot sequence. This problem also exists on most multi-port add-in cards for PCs.

When Sun machines receive a serial BREAK, they will drop down to the "ok>" prompt. This stops the operating system, and all the useful services that the machine is doing at the time, until someone gets on the console and types "go." This is actually quite useful for getting the machine out of a hung state, so you normally don't want to block this signal. Newer versions of SunOS either allow a patch[13], or include in the OS the ability to ignore the serial BREAK but listen for a specific character sequence instead. Sun customers with SunSolve[14] support access can check SunSolve for more information.

The problem occurs when you have a bunch of Sun hosts connected to a terminal server, and then the terminal server sends a BREAK to all of the attached hosts – and everything stops until you get on each of the consoles and type "go" for each host. (You can infer why it would be a bad idea to plug the console of your Conserver host into a terminal server that you access through that Conserver.)

Cisco Systems has also posted a Field Notice related to the BREAK problem to a Web page[15] in April 1998. The basic idea behind the notice was trying to keep the BREAK signal from getting to the Sun host, or make the Sun host ignore the signal. Unfortunately, the signal is too useful to administrators, so we can't advocate methods that block the BREAK signal from getting to the host.

Our search for a good answer has led us to try to find terminal servers that don't send BREAK at inopportune times. In support of Mark Burgess's series of articles (Systems Administration Research) in recent issues of *;login:*, I have started a series of tests on a variety of terminal-server models. We're posting our test methods as well as the results,

and encouraging other sites with an interest to perform similar tests and share their results with us. We're trying to determine whether the failures for various vendors are related to hardware or software; trying to determine when you can expect the failure to occur; and trying various recommended settings on the terminal servers to try to eliminate the problem. You can find more information on our results page[16].

In Summary

If you have always worked at a site with remote access to the serial consoles, consider yourself lucky. If you don't have some type of access now, I hope I've given you some reasons to consider adding it soon. If I've managed to interest you in the subject, but you want to learn more, I hope to see you in the tutorial in December.

I believe it was Elizabeth Zwicky who wrote a series of articles about software that should have been included in our favorite operating systems. I think Conserver belongs in that category.

References

1. Thomas A. Fine and Steven M.Romig, The Ohio State University: "A Console Server." Found in hardcopy proceedings of LISA IV, October 17-19, 1990. It is not currently online at the USENIX site.

2. Historical notes: Ohio State distributed the original terminal-server code. Conserver was the name of the server component. Folks at Purdue University added a "pipe to shell" feature, which some folks used to send sessions through Telnet sessions to high TCP ports in order to reach console servers. Later, Robert Olsen at Argonne National Labs (ANL) hacked the OIS code to include the socket-based support. The Certainty Solutions (formerly GNAC) version derives from a Purdue release, with the ANL additions, plus contributions from Arnold de Leon while at Synopsys, Inc.

Package	Maintainer	URL
Conserver	Thomas A Fine	<http://hea-www.harvard.edu/~fine/Tech/console-server.html>
Purdue version	Kevin S. Braunsdorf	<ftp://ftp.physics.purdue.edu/pub/pundits/>
Certainty Solutions (formerly GNAC) version	Bryan Stansell	<http://www.conserver.com/>

3. Conserver application <http://www.conserver.com/>

4. Celeste Stokely Sysadmin Web site <http://www.stokely.com/>

5. Celeste Stokely UNIX Serial Port page <http://www.stokely.com/unix.serial.port.resources/tutorials.html>

6. David K. Z. Harris's Console Connection Guide Web pages <http://www.certaintysolutions.com/consoles/>

7. BayTech Data Communications Division, Bay Saint Louis, Missouri, USA. RPC series power control devices. <http://www.baytechdcd.com/products/rpcseries.shtml>

8. American Power Conversion Corp., West Kingston, Rhode Island, USA. Master Switch, and Master Switch Plus units. <http://www.apcc.com/products/masterswitch/index.cfm>

9. Calgary Connect Corporation, Calgary, Alberta, Canada. PC Weasel 2000. <http://www.realweasel.com/>

10. SWATCH <http://www.stanford.edu/~atkins/swatch/>

11. USENIX LISA 2000 conference. <http://www.usenix.org/events/lisa2000/>)

12. "Communications Concepts, An Introduction to Data Communications", Communications Research Group, and Telebit Corp., pp. 3–12

13. Sun part number for the Consulting Special to address the serial BREAK problem: "CONSULT-ZSBRK"

14. SunSolve contract support site <http://sunsolve.sun.com/>

15. Cisco Field Notice about fixing the BREAK problem on Suns <http://www.cisco.com/warp/public/770/fn-tsbreak.html>

16. Certainty Solutions (formerly GNAC) Terminal Server BREAK-off pages <http://www. certaintysolutions.com/consoles/breakoff.html>