Console Management

Ian Durkacz, School of Informatics, April 2007

Table of Contents

Remote access to the serial consoles of Informatics servers is currently handled using a combination of locally-configured software and hardware, some of which is no longer obtainable. This report is intended to be an overview which summarises the current approach, discusses the pros and cons of several possible alternative approaches, and makes some suggestions for future provision.

The general conclusion is that, although there is no 'one-size-fits-all' solution, it appears viable to continue the current approach for our existing server hardware, and to move to an IPMI-based solution as future purchases allow. KVM over IP may have some niche application but does not appear to be of general use in this context, not least because it is currently too expensive.

1 Introduction

1.1 Requirements

The general requirement for a console management scheme here is for a simple and inexpensive solution (say, < £70 per node, based on the cost of the current set-up) which allows us remotely to:

- do machine installations;
- look at serial console outputs, even for a dead/locked/unresponsive boxes;
- power cycle machines; and
- examine and set BIOS/bootprom values.

And we would like to be able to do all this for multiple target machines simultaneously.

Whilst it is not clear exactly how many – and which – Informatics server machines need to be accessible in this way via a console management scheme (though ideally it might be all such), the general requirement is, ideally, one console server solution per 'bank' of racks. Since it is expected that each bank will be composed of four or five racks, with a total of perhaps 80 to 90 machines in each such bank, the ideal outcome would be a console management scheme that could handle 80 to 90 machines 'per unit'.

1.2 Current approach

Currently, access to the serial consoles[1] of various Informatics Linux and Solaris servers is handled by six console server machines. Each such console server is fitted with either a 16- or a 32-way serial card which is used to concentrate the serial ports of up to 32 target machines; each console server runs the `conserver` application [1] both to buffer the output of each target's console, and to arrange orderly access to these consoles.

Of the 32-way serial cards currently in use, five are Cyclades Cyclom-Y cards, and the other is a Perle SX card. Cyclades no longer exists as a separate company – it was taken over by Avocent – and the Cyclom-Y cards themselves are no longer available. Analogous Avocent serial cards *are* still produced (see [2]), but are not available in the UK, and are available in the US to OEM purchasers only. However, Perle multi-port serial cards as currently used here *do* remain available for purchase in the UK (see [3]).

The issues therefore are:

- Some of the hardware (namely, the Cyclades cards) we are using is no longer available: we need to ensure that we can support whatever approach we take.
- The current approach requires many serial cables to be run around machine rooms: there is a

1 In the case of our Linux servers running on Dell hardware, 'console access' also includes access to the BIOS screen by virtue of suitable console redirection settings in the BIOS.

desire to tidy this up if possible.

- It may be that approaches other than the current one are simply better and/or cheaper.

2 Possible future options

Considered here are five possible options: KVM over IP; IPMI SOL; Dell DRAC; Serial concentrator cards (i.e. the current approach); and commodity boxed solutions.

2.1 KVM over IP

KVM over IP allows conventional-style KVM access to server machines over the LAN. In general, the KVMoIP box itself will have a single network connection, will require the allocation of a single IP address, and will either be directly connected to a single target machine, or to several such machines via a separate KVM switch. Initial configuration of the KVMoIP box is done via a directly-attached keyboard and monitor; thereafter (in particular, after networking has been set up), configuration of the KVMoIP box proceeds over the LAN.

Where the KVMoIP box is connected to multiple target machines via a KVM switch, only one such target can usefully be addressed at any one time irrespective of how many remote user sessions the box might support.

Authentication mechanisms available to KVMoIP units will vary from manufacturer to manufacturer: a point for us would be the integration of any such device into our authentication infrastructure.

In the course of this report, only one such box – the AdderLink IP – has actually been tested, but there will be many similar products available: some notes are given in section 2.2.2 about one such alternative.

2.1.2 KVMoIP – AdderLink IP

On its own, the AdderLink IP ([4]) provides remote access to *one* target machine which has been directly connected to the AdderLink via a KVM cable; linked to a suitable KVM switch (or a cascade of such switches), it can provide remote access to 128 target machines.

The AdderLink IP box is completely self-contained and is accessed in practice via a Java-enabled web browser; interaction with it is via a VNC client implemented as a Java applet which is downloaded from the box on connection. It is possible to configure the unit so that it rejects incoming connection attempts from IP addresses outside a specified set.

In this evaluation, the AdderLink IP box has only been tested when directly connected to one target machine – and in this mode it appears to works as advertised, providing full and seamless console access. It would be useful, however, to test it in conjunction with an appropriate KVM switch, in order that its usefulness when connected to multiple target machines can be assessed.

Appendix A contains some configuration notes regarding the AdderLink IP box.

Pros:

- Easy to configure; after initial set-up, all configuration can be done remotely.
- Appears to work well and provides a fully-functional console.
- Requires the allocation of a *singl*e IP address only; this could be on a separate management network.

Cons:

- Expensive – £700 if used to target a single server; £100 per target server when used with a KVM switch.
- There is no way to buffer console output – so less 'post-mortem' information is available.
- The AdderLink IP unit only supports up to four remote connections at any time. (However, the unit can be configured so that a new remote connection from the 'admin' user is *always* accepted even if there are four remote connections existing at the time: in such cases, one of the existing connections is dropped.)
- When the AdderLink IP unit is connected to multiple target machines via a KVM switch, there appears to be no clean way of arbitrating access to these various targets when more than remote user is connected to the unit.

  In other words: despite up to four remote connections being available simultaneously, these can only usefully be to the same target server.
- There appears to be no way of integrating this device into our existing authentication infrastructure: the usernames and passwords associated with the AdderLink IP unit are stored within the unit itself in a local database; there is no support for distributed authentication via RADIUS, Kerberos, or similar.
- Cabling multiple servers to a KVM switch box would create similar (or worse) cabling problems to the existing serial card solution; in addition, maximum cable lengths need investigation.
- KVM is perhaps overkill anyway if we simply want text consoles.
- [Minor issue] Mouse calibration for this unit seems consistently to fail – though this is not really a problem for a pure text console.

Unit cost:

AdderLink IP unit	~£700
16-way KVM switch AdderView Matrix MP AVM216MP	~£900

Cost per target server:

~£700 (when used to target a single machine)

~£1600 / 16 = ~£100 (when used with a KVM switch)

2.1.2 KVMoIP – Lantronix SecureLinx Spider

This product (see [5]) has only recently become available and it has *not been tested in the course of this projec*t but, on paper, has several advantages over the AdderLink IP. In particular, it has a small footprint, it supports RADIUS, and it is easily scalable. The manufacturer's intention is to deploy one such KVMoIP box per target server; however one such unit could also service multiple target servers via a KVM switch in the same way described above for the AdderLink IP unit, and with the same advantages and disadvantages.

Pros:

- Intrinsically scalable.
- Does not require a separate power supply.
- Supports up to 8 remote connections at any time.
- On paper, at least, could be integrated into our existing authentication infrastructure via RADIUS (but not Kerberos.)

Cons:

- Expensive – £270 per target server. (But a single unit *could* be connected to multiple servers via a KVM switch.)
- There is no way to buffer console output – so less 'post-mortem' information is available.
- Requires the allocation of an additional IP address per target machine when used as the manufacturer intends. (All such addresses could be on a separate management network.)

Unit cost:

Lantronix SecureLinx Spider unit ~£270

Cost per target server:

£270 (but cheaper if used with a KVM switch)

2.2 IPMI – Intelligent Platform Management Interface

The Intelligent Platform Interface (IPMI, [6]) has been developed by Intel, Dell, HP and NEC as a specification for providing systems management capability in hardware. The Baseboard Management Controller (BMC) is the heart of an IPMI-based system; it is responsible for monitoring, controlling and reporting on all the manageable devices in the system.

The original version of IPMI – version 1.0 – allowed access to the BMC via system buses only.

IPMI v1.5 added support for accessing the BMC through either a serial port or via the network. (The physical serial and network connectors used can be either dedicated to the BMC, or multiplexed with the system's own connectors.) The network transport employs the Remote Management Control Protocol (RMCP) running over UDP, and this allows, for example, remote querying of machine status, and remote power up and/or power down of the machine. Such requests can be issued using appropriate client software: the `ipmitool` command [7] which is installed on DICE machines is one such client, and, for IPMI v1.5, the correct channel to use is `lan`.

IPMI v2.0 – the current specification – adds, among other things, support for encrypted network traffic, and formal support for Serial-Over-Lan (SOL) sessions: these allow the input and output of the serial port of the managed system to be redirected over the network. IPMI v2.0 SOL uses the RMCP+ protocol (again, this runs over UDP), and its use is directly supported by `ipmitool`. RMCP+ uses the `lanplus` channel.

Note that there is no formal support for SOL sessions in IPMI v1.5: various SOL implementations for IPMI v1.5 *do* exist, but these are all necessarily proprietary, and all require the use of additional proprietary software (the SOL proxy daemon) on the client side.

In the context of this report, the key IPMI feature is SOL: since it allows the redirection of the target machine's serial console (including the initial BIOS screen where this has been suitably enabled) over the network, it implements a remote console.

Despite the fact that the various implementations of IPMI SOL appear to be somewhat immature (various Usenet and web postings discuss various glitches), it appears that it is now an increasingly popular approach for console management; in particular, for compute clusters.

2.2.1 IPMI v1.5

IPMI v1.5 is supported by various 8th generation Dell servers: of interest here, it is supported by the PowerEdge 850, 860, and SC1425. (See Appendices B & C.)

IPMI v1.5 SOL has been successfully used in the course of this work to remotely access the consoles of both Dell PowerEdge 860 and SC1425 machines (`prague` and `split` respectively) – see Appendix D for further configuration notes on this.

The exact machine configuration necessary to get IPMI v1.5 and SOL working on any particular machine will vary depending on the details of that machine, its manufacturer, and its BIOS: Dell's Baseboard Management Controller Utilities User's Guide ([8]) gives details for current Dell machines.

Pros:

- Comes 'for free' with suitable servers – no additional cost per machine. (An aggregating console server machine *would* still be desirable however; that is, we would still need to

provide a distinct console server box per bank of racks. See the next point.)

- SOL sessions from many target machines s*hould* be able to be integrated (via the `conserver` application running on a console server host) into a single point-of-contact: this would allow easy integration with the existing DICE infrastructure (authentication etc.), provide buffering of the console output, and permit multiple simultaneous reader sessions. (But note: *this has not been tested.*)

Cons:

- Requires the allocation of an additional IP address per target machine; it is not clear whether this can be on a different network to that of the machine itself. (Note: *VLAN issues need to be investigated.*)
- Requires a proprietary SOL proxy daemon program: this is only available as a binary download, and it cannot be *guaranteed* to run on any particular version of Linux.
- SOL interaction is tediously slow – perhaps unusably slow – owing to the limitations of the underlying protocol.
- It does not seem possible to send a 'Break' to the target – presumably the SOL proxy doesn't forward this correctly?

Caveat:

- The machine/BIOS set-up necessary to support IPMI v1.5 SOL seems highly vendor- and machine-specific. Of the two machines accessed in this report, only the SC1425 (`split`) was available as a true test machine which could be brought down to the BIOS level, rebooted, etc., in order to investigate some of these configuration aspects.

Unit cost:

The cost is for the aggregating console server machine only. If an older machine can be redeployed for this, £0; otherwise, ~£1000.

Cost per target server:

£1000 / 48 = ~£20 (for 48 machines served by each aggregating server)

2.2.2 IPMI v2.0

IPMI v2.0 is supported by various 9th generation Dell servers: of interest here, it is supported by the PowerEdge 1950 and 2950 machines. (See Appendices B & C.)

IPMI v2.0 SOL has been successfully used in the course of this work to remotely access the consoles of both Dell PowerEdge 1950 and 2950 machines (`pasta` and `franklin` respectively) – see Appendix E for further configuration notes.

As for IPMI v1.5, exact configuration details necessary to set up IPMI and SOL v2.0 will vary

between machines and manufacturers.

Pros:

- Appears to work well and provides a fully-functional console.
- Comes 'for free' with suitable servers – no additional cost per machine. (An aggregating console server machine *would* still be necessary however; that is, we would still need to provide a distinct console server box per bank of racks. See the next point.)
- SOL sessions from many target machines s*hould* be able to be integrated (via the `conserver` application running on a console server host) into a single point-of-contact: this would allow easy integration with the existing DICE infrastructure (authentication etc.), provide buffering of the console output, and permit multiple simultaneous reader sessions. (But note: *this has not been tested.*)
- Supports encrypted network traffic.

Cons:

- Requires the allocation of an additional IP address per target machine; it is not clear whether this can be on a different network to that of the machine itself. (Note: *VLAN issues need to be investigated.*)
- Supported by very few of our current machines.

Caveats:

- The test machine `franklin` only became available late in the writing of this report, so the usability of SOL in all stages of the target machine's boot cycle has not yet been exhaustively tested. In addition, some networking issues remain to be fully investigated.[2]
- As for IPMI SOL v1.5, the machine/BIOS set-up necessary to support IPMI v2.0 SOL seems highly vendor- and machine-specific.

Unit cost:

The cost is for the aggregating console server machine only. If an older machine can be redeployed for this, £0; otherwise, ~£1000.

Cost per target server:

£1000 / 48 = ~£20 (for 48 machines served by each aggregating server)

2.3 Dell DRAC cards

Dell manufacture and sell proprietary 'Dell Remote Assistant Cards' (DRAC cards): these are add-

2 Specifically, as currently connected to our wires, `franklin`'s BMC's NIC does *not* receive network input unless configured as tagged-VLAN aware – yet the upstream switch is configured *not* to send tagged packets. This matter is under investigation. In any case, the VLAN capabilities of the BMC as a whole need further consideration.

on PCI cards implementing proprietary BMC functionality which are intended to be used with Dell-supplied software in order to provide a remote monitoring capability, including the provision of a remote console. DRAC cards thus functionally provide a similar facility to that provided by IPMI. There is a range of such cards, and it is necessary to use the appropriate one with any particular target Dell server type: the cards are *not* freely interchangeable between the various Dell servers.

DRAC cards in fact pre-date the IPMI initiative, so should now perhaps be considered overtaken by events. In any case, their proprietary nature makes them an unattractive proposition, at the least because they do not offer a solution for anything other than Dell hardware. There is no history of using them here, and to do so would require retrofitting of all machines. They are mentioned here only for completeness.

2.4 Serial concentrator cards and bespoke configuration

This is the current approach: a standard DICE server is fitted with a serial port concentrator card, the serial ports of machines of interest are connected via serial cables, and the whole is managed by the `conserver` application.

The current set-up uses serial concentrator cards manufactured by both Cyclades and Perle: the former are no longer available in the UK, but the latter *do* remain available here.

To continue using this approach we need to ensure that:

1. Serial port concentrator cards are available, at a reasonable price, and with an interface (PCI, PCI-X, PCI-Express, 3.3V, 5V, …) that suits our intended console server machine(s).
2. Drivers for such cards are available for the version of Linux we want to run on our console server machine(s).

Pros:

- Requires the allocation of a *singl*e IP address only for the console server; this can be on a separate management network.
- Easy to integrate into our existing authentication infrastructure, in the same way as any other DICE machines.

Cons:

- Availability of multi-way serial cards can't be guaranteed, (although suitable Perle cards *do* currently remain available at a reasonable cost.)
- The availability of Linux drivers for such cards can't be guaranteed.
- Requires the current extensive serial cabling.

Unit cost:

Perle 32-way serial port concentrator + break-out boxes £1356

A suitable (i.e. one with an appropriate PCI slot, among other things) aggregating console server machine is also needed. If an older machine can be redeployed for this, £0; otherwise, ~£1000.

Cost per target server:

(£1356 + £1000)/ 32 = ~£73

2.5 Commodity solutions

Several manufacturers produce rackable commodity 'console server' boxes. Generally, these boxes are fitted with serial concentrator cards (up to 48-way), run some version of Linux, and provide buffering of each serial input – so in practice they provide a very similar, but 'canned', solution to the current Informatics console servers. Avocent – the company which took over Cyclades – appears to have moved to the supply of such boxes only, rather than the serial concentrator cards which they use as an internal component.

At least three manufacturers – Avocent, Lantronix, and Perle – produce equipment which is readily available in the UK.

Pros:

- *If* they work as advertised *and* can be integrated into our environment, then such boxes offer an easy (and relatively inexpensive) drop-in solution for our requirements. But *no testing of this has been done in the current project.*

Cons:

- These boxes appear limited to a maximum of 48 serial ports per unit.
- These boxes would be practically identical to the current approach; in particular, they would have exactly the same serial cabling requirements.
- The boxes provide a canned solution which may or may not be easy either to alter or to update (the latter, for example, in response to security issues) – though in this regard some manufacturers do make software development kits available.
- The details of integrating such any such device into our existing authentication infrastructure need investigation. At least one (Lantronix SecureLinx SLC, [9]) claims support for Kerberos and RADIUS – but *no testing of this has been done in the current project.*

Comment:

- We would need to obtain one or more of these boxes for testing purposes in order properly to evaluate their potential use here.

Unit cost:

Lantronix SecureLinx SLC 48-way console server £2620

Cost per target server:

Lantronix: £2620 / 48 = ~£55

3 Summary

3.1 General conclusions

- KVM over IP works nicely and provides seamless access to remote consoles, but is currently expensive, and can't provide buffering of console output. The AdderLink unit reviewed cannot be integrated in to our existing authentication infrastructure, does not scale very well, and does not appear to offer a solution which would allow several users simultaneous access to distinct remote machines. Units from other manufacturers may be better in some of these respects.
- IPMI v1.5 SOL does not seem a viable option: it is not standardized; it requires additional software (the telnet proxy) which is only available as a binary download; it is too slow to be comfortably usable; it doesn't transmit serial breaks.
- IPMI v2.0 SOL is an attractive option: it seems to work well, and is supported for free (and with no additional software requirements) by conforming machines. Unfortunately, we don't currently have many such machines, but this situation should change as new and replacement equipment is purchased: any new Dell PowerEdge server should support IPMI v2.0. The issue of integrating IPMI SOL with a consolidating server in order to provide console buffering remains to be explored.
- The current approach of using serial concentrator cards works well and is – despite initial impressions – maintainable: the necessary hardware *can* still be sourced in the UK, and at similar prices to those we have paid in the past.
- The commodity console server boxes appear to offer a drop-in replacement for the current approach, provided they can be integrated into our infrastructure. They do not, however, address any of the cable management issues; in this regard they are not an advance on the current approach.

In summary:

Unless there are alternative approaches which have been completely overlooked in this review, it would seem reasonable to take the combined approach of continuing (and/or extending as necessary) the current arrangements, and introducing a solution based on IPMI v2.0 as we acquire machines that can support it.

The major issue with deploying KVM over IP is the trade-off between cost and convenience: the approach of using one KVMoIP box per target is attractive, but costly; introducing KVM switches lessens the cost per target machine but compromises overall usability. KVMoIP may therefore have use in certain controlled circumstances where we want to provide remote console access either to a small set of machines, or to a small set of users, but otherwise it does not appear to offer a general solution for us.

For reference, the estimated cost-per-target-server figures for the various approaches are repeated below from section 2:

Option	***Type***	***Approximate Cost per Target Server***
KVM over IP	*AdderLink IP*	£700 (~£100 when used with a 16-way KVM switch)
	Lantronix SecureLinx Spider	£270 (less when used with a KVM switch)
IPMI SOL	*v1.5*	£20
	v2.0	£20
Serial Concentrator Cards	*Perle SX card 32-way*	£73
Commodity Solutions	*Lantronix SecureLinx SLC 48-way*	£55

3.2 Unresolved questions

- How many of our machines require remote console access? (That is: what is the size of the problem we are trying to solve?)
- Of these, how many (currently, or will) support IPMI v2.0?
- How important is the buffering of console output?
- Can we successfully integrate any of the commodity boxes into our existing authentication infrastructure? (The only way to know for sure will be to test such boxes.)
- How successfully can `conserver` be integrated with IPMI v2.0 SOL?
- *... etc. ...*

Appendix A – AdderLink IP configuration

The AdderLink IP unit is initially configured via a directly attached keyboard and monitor; configuration thereafter proceeds via the network – refer to the product manual at http://www.adder.com/:

1. Allocate an appropriate IP address for the AdderLink. (Here: `129.215.46.132` = `kbadder1.inf.ed.ac.uk`.)
2. Set up IP address, netmask, and gateway on the AdderLink via a keyboard and monitor directly attached to the unit.

   (Aside: This unit *can* use DHCP, but there is a general question of autonomy here: a 'console server' should presumably be as independent of the rest of the infrastructure as possible.)
3. Finalise configuration remotely via the VNC applet embedded on the unit itself: point a web browser at `http://kbadder1/`.

Notes:

1. When using the AdderLink in KVM mode via the directly-attached connection, and logged in (to the AdderLink) as 'admin', Ctrl-Alt-C brings up the configuration screen.
2. To hard reset the unit (if ever necessary) :
   - Power off.
   - Set DIP switch 1 to ON.
   - Power on. You should see a maintenance screen: select 'Reset configuration'.
   - Power off; return DIP switch 1 to OFF; power on. You should see the initial configuration screen.

Appendix B – Dell naming conventions & IPMI support

The 'generation' of any Dell PowerEdge server is specified by the third digit from the right in the model number: a PE x9xx is 9th generation (eg PowerEdge 1950, 2950); a PE x8xx is 8th generation (eg PowerEdge 860); etc. (Aside: the leading digit in the model number is a key to the physical size of the server, in U's.)

Note that Dell 'SC' servers follow a different naming convention.

Generally: 9th generation Dell servers support IPMI v2.0; 8th Generation Dell servers support IPMI v1.5; 7th and 6th generation Dell servers *may* support IPMI v1.0 if suitably equipped; earlier generations offer no support for IPMI. Specifically, in our case for the types of machines we currently have:

Machine type	*Generation*	*IPMI version*
PowerEdge x9xx SC1435	9	2.0
PowerEdge x8xx PowerEdge 830 PowerEdge 850 SC1425	8	1.5
PowerEdge 750 (supports IPMI with optional ERA/O card)	7	1.0

See also `http://linux.dell.com/ipmi.shtml`

Appendix C – Infrastructure servers & IPMI support

Specifically, IPMI provision on the current principle KB Infrastructure Machines is as follows:

Hostname	*Machine type*	*IPMI version*
berlin	PE850	1.5
boulez	PE750	None (but 1.0 available via DRAC III card – aka ERA/O)
exeter	PE750	None (but 1.0 available via DRAC III card – aka ERA/O)
linnaeus	PE650	None
nautilus	SC1425	1.5
roujan	PE750	None (but 1.0 available via DRAC III card – aka ERA/O)
solti	PE750	None (but 1.0 available via DRAC III card – aka ERA/O)

Appendix D – IPMI v1.5 SOL on a 8th generation Dell

Configure the target machine:

Using `ipmitool` directly on an installed target machine, configuration of the IPMI LAN channel proceeds exactly as for a 9th generation machine, so see Appendix E below.

(*Comment:* Refer to [8] for details on configuring the BMC on a new machine pre-install via the BIOS, but note in particular that `Integrated Devices> Serial Port 1` *must* be set to `BMC NIC` in order for SOL to work correctly. Other settings (e.g. `COM1`) will result in `Error(0xa9)` when attempting to initiate SOL.)

On the target machine, put a serial console on COM1 set to 19200 baud:

```
#include <dice/options/serialconsole.h>
```

```
!init.entry_gettyS0        mREPLACE(9600, 19200)
```

(*Comments:*

- A baud rate of 9600 does not work for this version of SOL, even though it is offered as an option by the SOL proxy. Why?
- IPMI v1.5 SOL sessions appear always to be configured on `COM1`; cf. IPMI v2.0 sessions which appear to be configured on `COM2` – see Appendix E.)

Configure the client machine:

Alter the LCFG profile of any standard DICE machine thus:

```
!profile.packages mADD(-OpenIPMI-devel-*-* \
                       -OpenIPMI-*-*)
!profile.packages mADD(osabmcutil9g-2.0-36/i386)
```

(*Comment:* The `osabmcutil9g-2.0-36/i386` package is a download from Dell (go to http://support.dell.com/; keyword search for 'linux remote management') and it is needed to supply the telnet proxy daemon necessary for IPMI v1.5 SOL. It also installs other binaries, however, amongst which is `/usr/sbin/ipmish`. That is not needed here, but it conflicts with the binary of the same name installed by the `OpenIPMI-*-*` package – hence the latter's removal. The `osabmcutil9g-2.0-36/i386` package has been uploaded to the RPM repository in unmodified form; were it ever to be used seriously here the binary conflict should be resolved.)

Initiate a IPMI SOL session from the client machine*:*

```
[sandilands]idurkacz: telnet localhost 623
Trying 127.0.0.1...
Connected to localhost.inf.ed.ac.uk (127.0.0.1).
Escape character is '^]'.

...[snip]...

1:Connect to the Remote Server's BMC
2:Configure the Serial-Over-LAN for the Remote Server
3:Activate Console Redirection
4:Reboot and Activate Console Redirection
5:Help
6:Exit

Please select the item(press 1, 2, 3, 4, 5, 6):1
   1. Server Address:129.215.32.58 ← split's BMC's IP address
Username:root
Password:
Key:

SOLProxy Status:Connected.

...[snip]...

Current connection:129.215.32.58:root

...[snip]...

Please select the item(press 1, 2, 3, 4, 5, 6):2
```

```
Status: Serial-Over-LAN Enabled.
Current settings:
                Baud Rate:19.2K  ← must be 19.2K
                Minimum required privilege:admin

1.      Disable Serial-Over-LAN.
2.      Change Serial-Over-LAN settings.
3.      Cancel

Please select the item(press 1, 2, 3):3

...[snip]...

Please select the item(press 1, 2, 3, 4, 5, 6):3
Activating remote console now.
Remote console is now active and ready for user input.

Fedora Core release 5 (Bordeaux)
Kernel 2.6.18-1.2257_FC5_dice_1.2 on an i686

split.inf.ed.ac.uk login: idurkacz
Password:
Last login: Wed Apr 11 12:41:32 from sandilands.inf.ed.ac.uk
[split]idurkacz: exit
logout

Fedora Core release 5 (Bordeaux)
Kernel 2.6.18-1.2257_FC5_dice_1.2 on an i686

split.inf.ed.ac.uk login: ~
Console redirection is deactivated by user.
Deactivating ............

Console deactived.

...[snip]...

Please select the item(press 1, 2, 3, 4, 5, 6):6

Disconnected from 129.215.32.58:root
Remote console session terminated
Connection closed by foreign host.
[sandilands]idurkacz:
```

Appendix E – IPMI v2.0 SOL on a 9th generation Dell

Configure the target machine:

`pasta` is a Dell PowerEdge 1950, already installed and operational. The BMC on a new machine can be completely configured pre-install via the BIOS (refer to [8] for details); here, the BMC configuration was done via the command line on the running system.

First allocate a unique IP address to the BMC. (Here: `129.215.32.42` = `pastabmc.inf.ed.ac.uk`.) Then configure the BMC so that IPMI is functional over the network:

Load the IPMI kernel modules so that the IPMI `open` channel can be used:

```
[pasta]root: /sbin/modprobe ipmi_msghandler
```

```
[pasta]root: /sbin/modprobe ipmi_devintf
[pasta]root: /sbin/modprobe ipmi_si

[pasta]root: ipmitool -I open bmc info
  ...[snip]...
  IPMI Version              : 2.0
  ...[snip]...
```

Discover the LAN channel:

(*Comment:* there appears to be no standard number for the IPMI LAN channel – it's found by looking at all possible channels – but on all Dell implementations tried here, the LAN channel turns out to be channel 1.)

```
[pasta]root: ipmitool channel info 1
Channel 0x1 info:
  Channel Medium Type   : 802.3 LAN
...[snip]...
```

Configure the LAN channel (having previously allocated the BMC a unique IP address):

```
[pasta]root: ipmitool lan print 1
...[snip]...
MAC Address             : 00:15:c5:e8:fc:60
...[snip]...

[pasta]root: ipmitool lan set 1 ipaddr 129.215.32.42
[pasta]root: ipmitool lan set 1 netmask 255.255.255.0
[pasta]root: ipmitool lan set 1 auth ADMIN MD5,PASSWORD
[pasta]root: ipmitool lan set 1 defgw ipaddr 129.215.32.354
[pasta]root: ipmitool lan set 1 arp respond on
[pasta]root: ipmitool lan set 1 access on
```

Configure the IPMI root user for channel 1:

```
[pasta]root: ipmitool user list 1
ID  Name   Callin   Link Auth   IPMI Msg   Channel Priv Limit
2   root   true     true        true       ADMINISTRATOR

[pasta]root: ipmitool user set password 2 <IPMI root password>
```

At this stage, normal IPMI commands should be functional over the network, so:

Test IPMI over the LAN from any other machine, logged in as any user:

```
[sandilands]idurkacz: export IPMI_PASSWORD=<IPMI root password>

[sandilands]idurkacz: ipmitool -I lan -H pastabmc -U root -E chassis status
System Power         : on
Power Overload       : false
...[snip]...
```

On the target machine, put a serial console on COM2:

```
/* SOL additions to pasta's profile */

#include <dice/options/serialconsole.h>

!grub.kernelargs_defaultboot_disk1      mREPLACEQ("console=ttyS0,9600", \
                                                  "console=ttyS1,57600")

!init.entries           mADD(gettyS1)
init.entry_gettyS1      T2:2345:respawn:/sbin/agetty -L 57600 ttyS1 vt100
```

```
!auth.securetty         mADD(ttyS1)
```

(*Comment:* The above assumes that machine has been configured to provide serial console output; that is , that the BIOS has previously been configured thus:

Set `Serial Communication> Serial Communication` to `On with Console Redirection via COM2`

Set `Serial Communication> External Serial Connector` to `COM2`)

Initiate a IPMI SOL session from any other DICE machine:

```
[sandilands]idurkacz: export IPMI_PASSWORD=<IPMI root password>

[sandilands]idurkacz: ipmitool -I lanplus -H pastabmc -U root -E sol activate
[SOL Session operational.  Use ~? for help]

Fedora Core release 5 (Bordeaux)
Kernel 2.6.17-1.2174_FC5_dice_1.1smp on an i686

pasta.inf.ed.ac.uk login: idurkacz
Password:
Last login: Mon Apr 23 19:01:26 on ttyS1
[pasta]idurkacz: exit
logout

Fedora Core release 5 (Bordeaux)
Kernel 2.6.17-1.2174_FC5_dice_1.1smp on an i686

pasta.inf.ed.ac.uk login: ~. [terminated ipmitool]
[sandilands]idurkacz:
```

References

1. Conserver home page:
   http://www.conserver.com/
2. Avocent serial cards:
   http://www.connectivity.avocent.com/products/bus-based/
3. Perle serial cards:
   http://www.perle.com/products/serial-cards.shtml
4. AdderLink IP KVMoIP:
   http://www.adder.com/main.asp?id=508_2074_23622
5. Lantronix SecureLinx Spider KVMoIP:
   http://www.lantronix.com/data-center-management/kvm-solutions/securelinx-spider.html
   http://www.lantronix.com/pdf/Spider_PB.pdf
6. IPMI specifications:
   http://www.intel.com/design/servers/ipmi/
7. `ipmitool` home page:
   http://ipmitool.sourceforge.net/
8. Dell OpenManage Baseboard Manager Controller Utilities User's Guide:
   http://support.dell.com/support/edocs/software/smbmcmu/
9. Lantronix SecureLinx SLC Console Server:
   http://www.lantronix.com/data-center-management/console-servers/securelinx-slc.html

---

[END]