PowerCockpit™ 2.0

Technology Overview

PowerCockpit Technology Overview
Version 2.0, Spring 2003

Contact Information:

Mountain View Data, Inc.
1668 Hollingsworth Drive
Mountain View Data, CA 94040 USA
Phone: 650.988-9345
http://www.powercockpit.com/

Notice:

Contents

PowerCockpit Technology Overview

1 Introduction

PowerCockpit™ software is a powerful system for setting up and managing large groups of servers with dramatically less effort and greater reliability than other solutions. It integrates many new and existing technologies in innovative ways to enable administrators to move from managing single servers to collections of servers using the same interface. This report describes some of the key obstacles to managing racks of servers and the technologies PowerCockpit employs to solve these problems.

1.1 PowerCockpit Markets

PowerCockpit is designed to improve productivity and enable new services and capabilities for server-related hardware and software businesses. Users can apply PowerCockpit to many different markets, including:

- Hardware manufacturer and OEM companies: To more efficiently and flexibly pre-load servers with operating system (OS) and application software stacks.
- Build-to-order and customize-to-order channels: To construct customized servers quickly and flexibly, including customer-supplied images or network parameters.
- VARs: To create, manage and deploy enhanced solution stacks.
- ISPs and remote hosting facilities: To provision new servers or reassign computing assets quickly.
- IT departments: To maintain consistent system configuration for servers on worldwide networks from centralized management hubs.
- Developers and end users: To backup, replicate and test system configurations for server and workstation machines.

1.2 Managing Servers

Servers are managed differently from workstations because they are used differently. There can be many servers in an organization, and they often do not have monitors and keyboards attached and are housed in racks in a computer room, instead of beside someone's desk.

More important from a management perspective is that servers are typically critical to the operation of the business, since they are often used in direct interaction with customers. Because of this:

- Uptimes are critical.
- They are often reprovisioned to meet changing business needs.
- Software updates must be fast, painless, verifiable and reliable.
- Applications, more than users, are the focus.

Racks of low cost servers can provide greater reliability and customizability at a far lower equipment and software cost than vertically integrated high end solutions, but these benefits can often necessitate high system management costs.

1.3 The PowerCockpit Solution

PowerCockpit dramatically simplifies system administration, reducing costs and increasing reliablity and agility. It has many powerful capabilities that revolutionize managing multiple servers. With it, system administrators can apply their expertise more effectively to large numbers of servers and

capture that expertise to avoid repetitive and error prone tasks. The PowerCockpit software can:

- *Collect* validated images from reference servers in a form that can be quickly deployed to other servers.
- Manage a *repository* of collected images for later deployment.
- *Deploy* complete images to multiple servers simultaneously to install complete validated OS and application stacks.
- *Customize* the image with its own personality and software layers during the deployment process.
- Keep a record of previous deployments and *redeploy* using those records to replicate the previous deployment.
- Deploy using *IP multicast* to large numbers of servers efficiently.
- Build a *Restore CD* which can restore the image on a server from a CD-ROM or DVD-ROM.
- Collect and deploy Linux and Microsoft® Windows® 2000 and XP, using the same interface.
- Maintain a hierarchical record of all of the *nodes*, logical groupings of them and ways of manipulating them.
- Allow administrators to perform commands on sets of nodes and collate the output to help find and fix problems easily.
- Automatically collect an extensible set of hardware and software properties from nodes on user-specified schedules.
- Employ a sophisticated and extensible *trigger* mechanism to take actions when designated events occur.
- Run from both a sophisticated graphical interface and from Perl scripts.
- Support modular licensed extensions created by the team, customers or third-party vendors, using a clean and lightweight component model.

PowerCockpit is a sophisticated, yet easy to use, application that can transform how you manage your servers.

2 Deployment

Installing and configuring new servers can require hours or often days of work, even with expert administrators. The operating system must be installed and configured, as well as additional application packages, generally by hand or using ad-hoc scripts. In addition to being error prone, this process is simply too slow to use in a dynamic environment. The result is that server farms do not respond well to fluctuations in demand or business needs, candidate configurations are not properly tested and the services available on a network become inflexible and difficult to scale.

The software on an OS distribution CD-ROM is not a runnable version of the operating system – it is a collection of files and scripts used to *build and configure* an operating system. Even when scripted, this process is lengthy and prone to error.

PowerCockpit takes a much different approach to this problem of *provisioning* a new server, illustrated in Figure 1. It collects an image of an *already installed* operating system, including installed software and packages, so that it can be *deployed* to new servers quickly and easily. These images are automatically customized by adding further layers of software and configuration.

The collected image is saved in a repository to be deployed over the network, or it can be written to a CD or DVD for direct deployment onto the hard disk.

The expertise of the system administrator is used to install the OS and software a single time. PowerCockpit records that expertise, saves it in a repository and deploys it whenever and wherever needed. This is the most important capability of PowerCockpit.

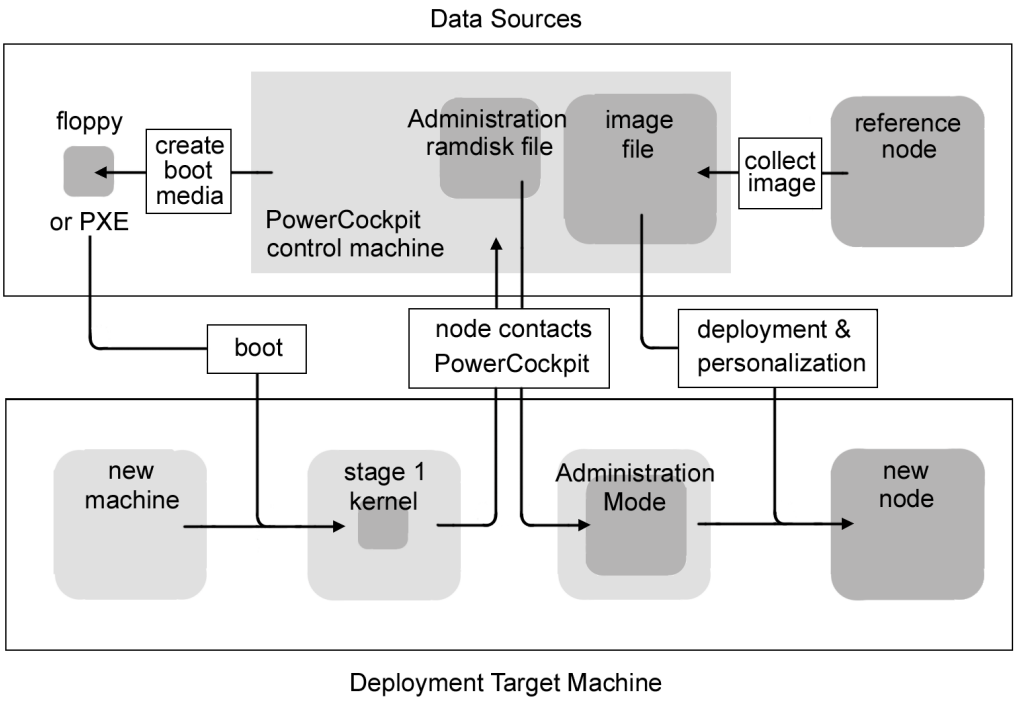

Figure 1: An overview of the image collection and deployment process.

2.1 Image Repository

PowerCockpit creates images by *collecting* them from existing, operational systems. Once the system administrator has decided what software is needed and how it is to be configured on one system, PowerCockpit collects the image and saves it in an image repository, ready to be quickly deployed and customized on any number of other nodes.

An image repository becomes a record of past configurations, of trial images for testing and of administrator expertise in the form of certified, fully configured images. Easier deployment means better tested systems, wider and easier adoption of certified images, and better disaster recovery.

Images can be in one of two forms, named after the Unix commands for recording them:

- A **tar** image, which is a collection the files, directories and their metadata. A tar image is combined with an abstract disk layout to build a working file system on the destination disk.
- A **dd** image, which is a direct transcription of the bits on the disk device and includes the source disk's partitioning and file system. A dd image is laid down directly on the destination disk without interpretation.

2.2 Deployment History

Each time a deployment is done, a record of it is saved, and that record can be used to repeat that deployment or provide detailed accounting of the history of each server and of the deployments done on the network.

Over time, a history is built of the deployments that have been completed in a form that allows auditing and rolling back to any point, either by direct user command or automatically from a script.

2.3 Customization

Because the target machine is often different in some way from the source, each deployment includes *customizing* the image for each destination.

The disk that the image is deployed onto may be a different size from the source, thus PowerCockpit has *abstract disk layouts* that adapt partition sizes between the source and destination before laying

down the new image.

After a base image is laid down, *layers* may be added to it with software and data to specialize a generic image for a particular usage.

Software in the image often needs to be configured after being deployed, such as with activation keys, or to be configured for the network parameters of the new machine. Devices like hard disks and network cards on the destination may be different from the source, requiring that different drivers be enabled and that adjustments be made to standard configuration files. PowerCockpit deployments include arbitrary personalization steps and comes packaged ready to handle these adjustments for many popular Linux distributions and for Microsoft® Windows® 2000 and XP.

By combining these techniques, PowerCockpit users can create stable core images that are automatically customized for a wide range of uses.

3 Clusters of Nodes

Each of the computers on a network generally assume multiple functional roles and the operations done on them depend on the roles. There are file, print, web and compute servers, databases, caching proxies etc. Machines are also grouped based on location, network subnet, CPU architecture and other attributes.

PowerCockpit handles this complexity by maintaining a database of all of the information about the managed nodes and allowing the administrator to assemble the network *nodes* in arbitrary overlapping and hierarchical *clusters*. This cluster data base is known as the *Clusterbase*. Nodes can belong to any number of clusters, and clusters may have subclusters. Thus, administrators are free to represent a wide variety of overlapping roles for each node.

In the user interface and implementation of PowerCockpit, you continually operate on entire sets of nodes. Other existing system administration GUIs that are capable of managing a remote machine at all can generally administer only one machine at a time. The administrator must set up one machine, then the next and iterate through the remaining machines. This is inefficient, tedious and error prone. With PowerCockpit, all the machines can be set up and managed simultaneously.

3.1 Node Properties

PowerCockpit can manage hundreds or even thousands of nodes, and each one can record complex properties about its hardware, software and current runing state. These properties may be simple strings (like its name) or complex hierarchies of data (like the details of all of the hardware installed on it).

PowerCockpit provides a graphical interface that allows node properties to be browsed and edited (including cut/copy/paste) for one or many nodes simultaneously.

PowerCockpit allows properties to be set by having the nodes send reports to the console automatically on predetermined schedules or in response to events. PowerCockpit has an up to the minute accurate record of the current state of your servers.

Nodes may be placed in clusters automatically or selected in the GUI for further operations using simple queries on these properties.

Changing node properties (like fluctuations in web pages being served or rising errors rates from hardware devices) can trigger scripts on the console and back on the nodes to respond to events as they unfold, so your servers can be self-healing.

4 Global Commands

Once an image has been deployed and the nodes recorded, the state of the machines will not generally remain static. Between deployments, software may be installed and configured, problems diagnosed and solved and many other maintenance tasks performed. With conventional tools, this is often done by hand or through complex scripts. These methods do not have access to the unified and up to date database or to the sophisticated algorithms that PowerCockpit provides to make managing sets of machines as easy and secure as one.

4.1 PowerCockpit Daemon

When you run the PowerCockpit control console, it can establish connections to the nodes to run the

PowerCockpit daemon. Multiple users from multiple PowerCockpit consoles can connect to the same set of nodes simultaneously (as shown in Figure 2) without interfering with each other, because each is communicating with a separate daemon on the node.

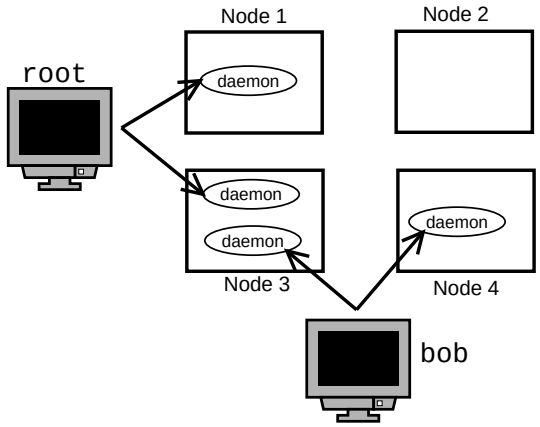

Figure 2: Daemons are run on nodes only when consoles connect to them. Each daemon runs as the user that requested the connection.

PowerCockpit does *not* introduce an additional set of user names, passwords and protocols for its authentication. All of the connections PowerCockpit makes use of SSH, which provides secure authentication and a single sign-on. Users may authenticate as any user, and the daemon has only the privileges of that user.

Once a connection is authenticated with SSH, the persistent socket connection to the node is set up and can be encrypted with SSL, if the user decides the performance penalty is warranted.

The node and console exchange *heartbeats*, so the console can keep track of the state of the connection to the node. Using the heartbeat, PowerCockpit can detect problems in seconds and display them in the GUI, inform a script through a callback, or shut down the daemon if the console was shut down in an uncontrolled way.

4.2 Global Operations

A fundamental part of system administration amounts to executing commands on a machine, examining the output and taking action based on that output. This is how problems are detected, diagnosed and solved. PowerCockpit provides a sophisticated interface (shown in Figure 3) for executing commands on a set of nodes and collating the output so that nodes with identical output are grouped. The administrator can then easily find which nodes are anomalous and focus on them, exclude them from further analysis, repeat previous operations, etc. This interface can be used efficiently on any number of nodes without the user needing to type in node names or select groups of nodes directly.

This same interface is used for:

- Sending files to lists of nodes and monitoring the transfer. The files may be transferred from the computer running the control console or from proxy servers. Files are sent directly from the proxy server to the destinations, not routed through the console, so the console can be connected through a slow or firewalled channel and large files transferred entirely behind the firewall.
- Installing software using the RPM system using simplified interface that combines RPM command execution and automatic file transfers if needed.
- Running property report scripts that record data about the nodes into the Clusterbase.

For example, suppose a security advisory reports that a particular version of `sendmail` is insecure and should be upgraded. From PowerCockpit, you use a global command to find the `sendmail` version on all the nodes, and PowerCockpit automatically collates the nodes with the same versions. On the nodes with the insecure version, you can transfer the RPM from a proxy server and install it in one operation. You can then set up an automatic property reporter that looks for the insecure version and notifies the operator if it somehow gets reinstalled, or even does the upgrade automatically in the future.

5 Multiple Interfaces

System administrators have two kinds of requirements for a tool like PowerCockpit. They need

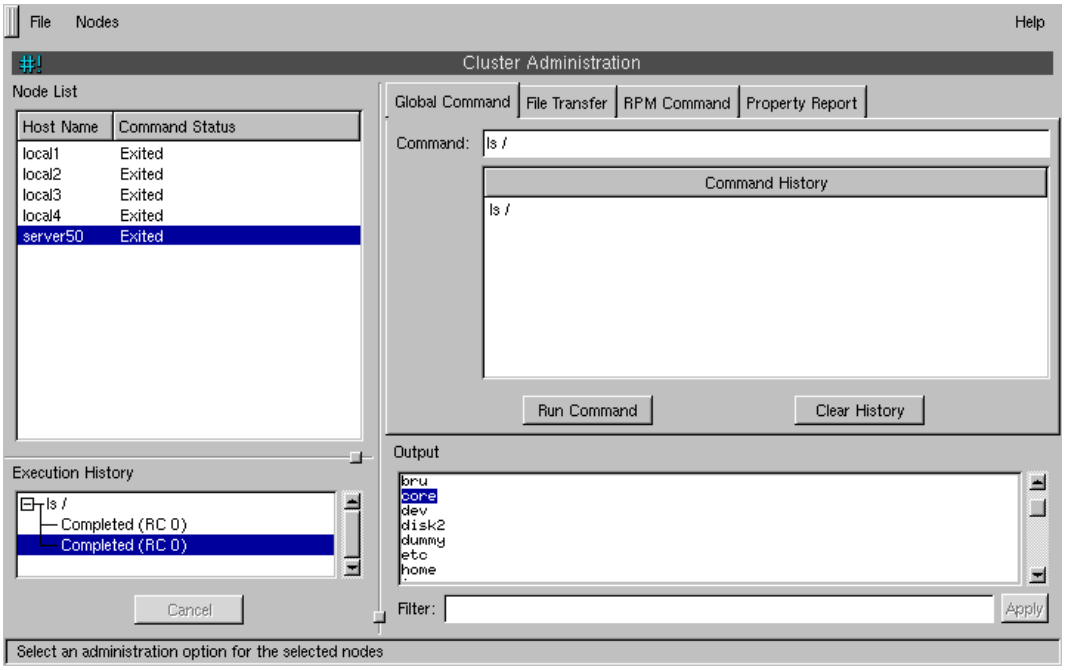

Figure 3: The Global Command window, with the list of nodes at the upper left, the collated command history at the lower left, the command entry at the upper right and the command output below.

to be able to sit down in front of a single system and debug problems, and they need automated control over routine operations. For the former, PowerCockpit provides a graphical user interface and for the latter PowerCockpit provides a Perl scripting interface.

5.1 GUI Design

GUIs that are sumultaneously powerful, extensible and easy to use are few and far between. PowerCockpit pulls together successful elements from many existing sources to build a uniquely powerful GUI.

- The Workspace window (shown in Figure 4) is modeled on a file browser: cluster tree on the left, details about the selected cluster on the right and actions to perform across the top.
- The Deployment window leads the user through the steps to be performed in sequential order.
- The Node Properties window allows editing of many nodes at once, avoiding the common problem of a point-click-type-repeat cycle.
- The Global Command window handles the tedious aspects of collating data and selecting sets of nodes.
- On-line context sensitive help is provided through an embedded browser.

PowerCockpit uses the Gtk+ GUI library to provide a smooth, reliable, portable and familiar interface. It is layered on top of X11, which natively provides remote display capabilities to any X-window server available for all popular desktop operating environments and web browsers.

5.2 Scripting and Triggers

PowerCockpit is scriptable using the Perl language, and scripts can be run either together with the GUI or purely from the command line.

The scripting interface provides Perl objects that directly reflect the internal objects. The scripting system therefore sees the same interfaces that the C++ code sees, within the limits of the differences in the languages.

Both the GUI and the scripts use the same internal objects. The objects for clusters, nodes, connections etc. emit notifications when they change,

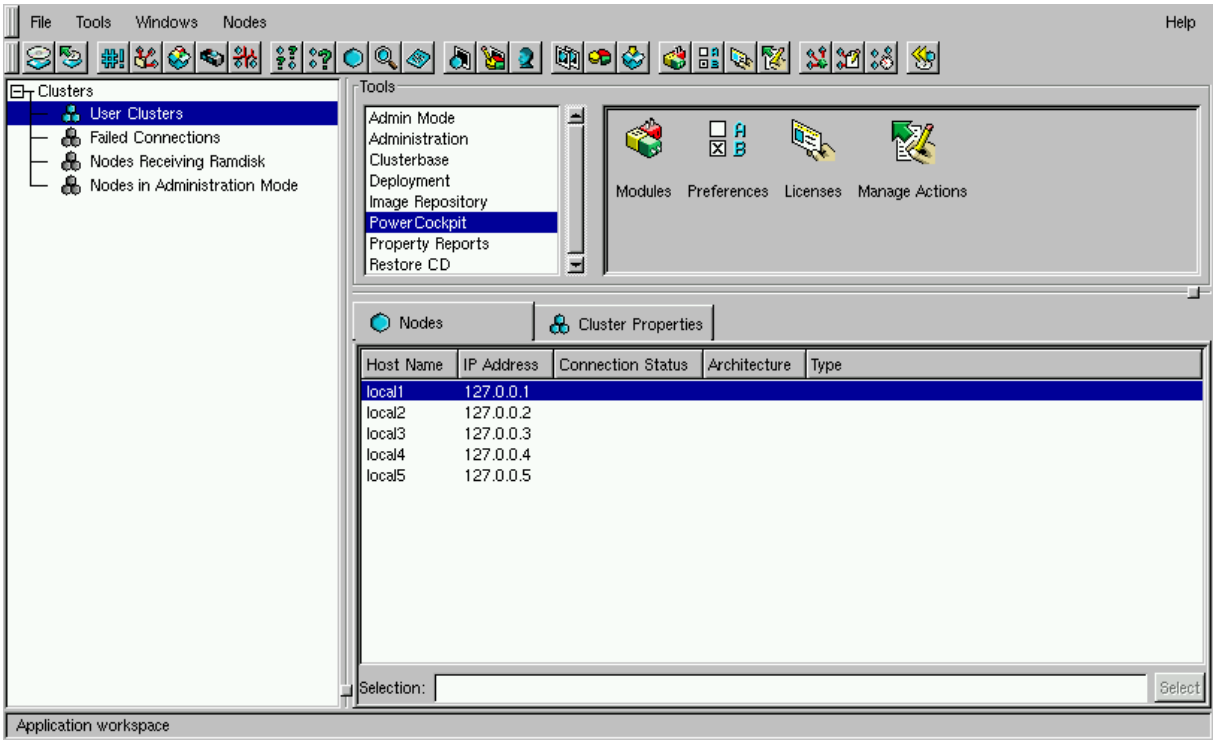

Figure 4: The Workspace, the control console of PowerCockpit.

and both the GUI and scripts listen for those messages and take actions. When the user initiates a change through one part of the GUI, the objects are changed and notifications are sent that update the rest of the GUI and any scripts watching those objects.

This is the foundation of the PowerCockpit *trigger* system. Events trigger scripts to run, which can change the data in the Clusterbase, change the data visible in the console, and run scripts that take actions on the nodes. Those actions can in turn generate events that trigger new actions, so sophisticated system management can be built from simple pieces.

5.3 Component Model

PowerCockpit is built using a dynamic component model, and the console and daemon applications are in fact the same core with different components loaded. In addition to extending the functionality of PowerCockpit with scripts, the PowerCockpit SDK can be used to develop new modules for both the console and daemon to extend the functionality of PowerCockpit as far as you like, taking advantage of the sophisticated infrastructure that PowerCockpit provides.

The licensing system in PowerCockpit is sufficiently general that third parties and OEM's can develop and sell modules as additions to PowerCockpit and distribute them with securely signed licenses, which will be validated before the module can be loaded. PowerCockpit is not just a system administration tool, it is a platform for building focused products to solve particular needs.

6 Summary

PowerCockpit combines advanced technologies of solid design and a high quality implementation to build a uniquely powerful tool for system administrators in data centers, enterprise environments or hardware manufacturers. It provides the security necessary for data centers through its use of tools like SSH and SSL. It has the power and ease of use necessary to run racks of servers through its unmatched GUI and the flexibility and automation capabilities necessary for unattended operation through its powerful scripting and triggered action

interface. Through its component model architecture, it provides a platform for OEM, VAR and other third parties to develop custom extensions and for future PowerCockpit products already under development.

For more information on PowerCockpit and related products, see the PowerCockpit web page *www.powercockpit.com*, or contact us directly at *sales@powercockpit.com*.