Proceedings of the Linux Symposium

June 27th–30th, 2007
Ottawa, Ontario
Canada

Conference Organizers

Andrew J. Hutton, *Steamballoon, Inc.*
C. Craig Ross, *Linux Symposium*

Review Committee

Andrew J. Hutton, *Steamballoon, Inc.*
Dirk Hohndel, *Intel*
Martin Bligh, *Google*
Gerrit Huizenga, *IBM*
Dave Jones, *Red Hat, Inc.*
C. Craig Ross, *Linux Symposium*

Proceedings Formatting Team

John W. Lockhart, *Red Hat, Inc.*
Gurhan Ozen, *Red Hat, Inc.*
John Feeney, *Red Hat, Inc.*
Len DiMaggio, *Red Hat, Inc.*
John Poelstra, *Red Hat, Inc.*

Linux Rollout at Nortel

Ernest Szeideman
Nortel Networks Ltd.
eszeidem@nortel.com

Abstract

At Nortel, we have focused on delivering a "Standard Operating Environment" for our design systems whereby we maintain a common set of tools and processes in the rollout of Linux and other operating system images. There are a number of opportunities, challenges, and pitfalls with bringing this about at an enterprise level.

1 Introduction

The Linux version of the Standard Operating Environment (SOE) was born a few years ago out of an initiative to introduce a standard image configuration that would address the needs of groups who were increasingly looking to Linux for product development and testing. Since that initial SOE release, Linux has become common for desktop and server computing solutions across the corporation. The goal of every Linux SOE release is to introduce a certified and supported Enterprise Linux distribution into Nortel.

A survey of the literature reveals many articles detailing the specific implementation details and related challenges faced in creating a standardized image. Fewer articles speak of the high level design and engineering process driving the implementation, and fewer still speak specifically about lessons learned which would assist others in overcoming assumptions and processes counterproductive to such an endeavor. Ubiquitous throughout the IT industry is the concept of a Standard Operating Environment referring to a standard image configuration. However, to be useful and accepted in an enterprise, an SOE requires that the design must solve real business problems for a company, problems that vary over time, across different industries, business environments, and even different business cultures. As such, although the design and development of an SOE may vary, the lessons learned by one company should prove of benefit to others as well.

This paper will discuss what an SOE is, briefly describe how we did the design and why, and most importantly speak of the lessons we have learned and how they relate to Linux and the open source model from an enterprise perspective.

2 What is an SOE and Why

A Standard Operating Environment for Linux means a standard image configuration for both desktops and servers. The intention is that unless there is sufficient justification, all supported Linux installs within the company will use the SOE. This greatly simplifies management by ensuring consistency in the deployed image regardless of location which provides high levels of reliability and supportability.

The SOE should support a limited set of hardware that has been chosen for company-wide use for both servers and workstations. Although Linux provides perhaps the most hardware support of any operating system ever, minimizing the set of hardware reduces the matrix of testing required. This reduces hardware support costs, and reduces hardware acquisition costs due to volume purchasing.

The inclusion of a common set of Linux vendor packages on all machines, a common set of third-party packages, as well as a common set of company-developed packages, ensures consistency in the deployed image regardless of location. It also provides a vehicle whereby software can be deployed company-wide to meet ever changing business needs.

Security and network certification of the image implies security and network configuration changes (such as ensuring limited world access to init scripts or checks to ensure that IP forwarding is turned off). This helps the company in minimizing risk, taking advantage of security and network expertise, providing confidence in the

SOE, as well as ensuring that the image plays nicely within a company's environment.

Providing a standardized installation process, including appropriate storage locations for the image, reduces the net cost per Linux install by reducing complexity, ensuring standardization, and maximizing the ability to support the install via documentation and support help lines.

Lastly, having a formalized process to gather requirements, design, implement, test, and trial an SOE ensures that tasks can be adequately resourced, timelines meet business priorities, and consumers of the image can plan for the deployment and use of the image to accomplish business goals.

3 The Nortel SOE

At Nortel Networks Inc., there are over 291,000 nodes on a network with over 350 locations throughout the world containing 8,000 subnets housing a myriad of servers and desktops running many different operating systems and providing access to a number of different network services (as of May, 2006). Any opportunities at standardization will result in substantial savings to the corporation.

Level		Explanation
5	Patching	Post-SOE maintenance
4	Group specific	UML, Clearcase
3	Location specific	Postinstall, cfengine
2	Global config	Packages, security, network
1	Vendor OS	Consistent set of packages
0	Hardware	Hardware catalogue

The table above denotes the high-level design of the Linux SOE.

Level 0, or the Hardware layer, represents all activities in achieving a standard catalogue of hardware including hardware comparisons, benchmarking, vendor negotiation, and the like. Any SOE that is released will have, as a minimum, the requirement to support the catalogue hardware.

Level 1, or the Vendor operating system layer, is the inclusion of a consistent set of packages from the vendor that is supposed to achieve three things:

1. It must include a reasonable set of packages required to support the environment.
2. It must include packages that are deemed as required by the internal customers.
3. It must attempt to be consistent with previous SOE releases (i.e., it must attempt to match the functionality that was included in previous SOE releases at this layer).

In this layer, one should capitalize on and make use of the installation tools or mechanisms provided by the vendor (e.g., Anaconda/kickstart used in RHEL (Red Hat Enterprise Linux) or YaST (Yet Another Setup Tool) used in SuSE (Software und System-Entwicklung)). As Nortel is using RHEL in its SOE, this layer is accomplished with the use of kickstart where the required package groupings and packages are specified in the %packages section.

Level 2, or the Global configuration layer, is where other packages not necessarily provided by the vendor are installed. This includes security, network, and company-provided packages. A special design consideration for this layer is to keep absolute separation between whatever installation mechanisms the vendor provides and the one relied upon at this layer. At Nortel, the automated kickstart installation mechanism has a post install section that is used to automatically kick off an install script, which completes all aspects of this layer. The benefits of this are threefold:

1. It allows easy determination of where problems may exist in an install.
2. It insulates the SOE engineers from changes made to the vendor's installation tools or mechanisms.
3. It allows for changes to the underlying Linux distribution without major impact to this level or the levels above it.

Level 3, or the location-specific layer, is designed to answer the requirement of how to maintain standardization across a multitude of locations where specific infrastructure services, service names, and configuration processes differ. This is a key challenge for any large corporation. The methodology employed requires that the target node take advantage of locally dependent services

while maintaining the standardization of the SOE. For Nortel, an init (initialization) script, which allows for complete automation, handles this layer which includes support for NIS, NTP, LDAP, and cfengine as well as other services. Automation of all of these tasks is not only desirable, but is also required if one wants to ensure consistency in deployment. An additional requirement of this layer is to be able to re-implement these services in the event that a machine changes locations (for example, if a node is redeployed to another site, the employee changes locations, etc.). Making use of an init script allows for this requirement. Configuration-specific parameters are sourced from location-named files containing all data relevant to this layer for each major location at the company.

Level 4, or the group-specific layer, contains that which does not need to be installed everywhere, yet which is required by specific groups. Examples of this are the use of virtualization such as UML (User-Mode Linux) as well as Clearcase. Interestingly, group-specific software, such as Clearcase, also has location-specific dependencies (for example, which VOB (Versioned Object Base) servers to connect to may be dependent on which site you work at). The ownership of each of the capabilities relied upon at this level is provided at Nortel by specific individuals or groups who may have formal vendor relationships as required. As these people or groups have the requirement to have their code work with the SOE, they form a special community who has access to pre-releases of all SOEs. References are made to their documentation from within that provided for each SOE. In some cases, there is an automatic reinstallation of these pieces in the event a reimage is performed.

Level 5, or the patching layer, concerns itself with post-SOE maintenance. Once an image is deployed, it must still be maintained and/or kept track of for licensing. There must also be the facility to account for changing business requirements, which may include the deployment of new or updated products (for example, DST (Daylight Saving Time) fixes). At Nortel, we are currently using RHN (Red Hat Network) Satellite. We take a snapshot of the base channel minus kernel (3rd party applications are tied to kernel versions). The snapshot of the channel is tested before the patch bundle is released to ensure that the patches don't break anything in the Nortel environment. This patch bundle is produced quarterly. This allows time to deploy the bundle to all of the systems in an orderly and systematic way. Within Nortel, rhnsd is not used. The patch window for each system is scheduled ahead of time and controlled by configuration files on the system. A generic scheduling method is employed that can be used across all the UNIX and UNIX-like operating systems. As everything is packaged as a requirement of being included in the SOE, this enables all aspects of the standardized image across all levels to be patched.

4 Lessons Learned

A number of lessons have been learned since a fully supported Linux was introduced in Nortel a few years ago. These are lessons taken from an enterprise perspective and may not apply everywhere.

1. Remote cloning of machines in an enterprise is a deployment concern that must be taken into account. Hewlett Packard's iLO (Integrated Lights-Out) or other similar remote console mechanisms are highly desirable, particularly when the system administrator or installer is located remotely from the machine being imaged. One should not assume that the installer is able to sit in front of the machine being deployed.

2. Windows interoperability solutions contained within Linux have really enhanced its value in the enterprise, particularly when compared with other proprietary UNIX operating systems. However, it has and continues to cause many challenges. VMware with a Windows guest is currently being used in Nortel with workstations to provide standardized Windows images to those running Linux. One may wish to refresh one's Linux image to the latest General Availability (GA) release, but this does not assume that one wishes to have their Windows environment upgraded as well. To accommodate this requirement, a localdisk partition is created on all default Linux installs which holds, among other things, the VMware image files. An upgrade clone is utilized which wipes all partitions, except localdisk, and thereby allows the user to get a new Linux build while keeping any VMware images they may have had.

3. One cannot assume access to an enterprise's DHCP (Dynamic Host Configuration Protocol) infrastructure to make use of PXE (Preboot eXecution En-

vironment) installs or Red Hat Network provisioning. In a large corporation, different groups are responsible for different aspects of the infrastructure. As such, it takes time to get consensus on how best to implement change. One such example is the use of PXE as it relates to the DHCP infrastructure. If this is the case, as it is at Nortel, one may need to find alternative means to accomplish remote upgrades of machines. A script, which integrates with cron (a time-based scheduling service in Linux) is currently being used to provide this functionality.

4. For security reasons, patching of one's infrastructure is necessary using internal repositories. As well, no information about the nodes being patched should leave the company network (cannot use RHN or RHN proxy; must use RHN Satellite).

5. An additional comment with regards to patching involves the potential for divergence when patching versus re-rolling an SOE using a newer update. This is particularly true if one is limited in updating kernels due to third-party reliance on the kernel (e.g. Clearcase or UML). For example, if one starts with a RHEL 4.1 machine and patches it with a patch bundle to 4.3, one may not end up with the same system as if one started with RHEL 4.2 because not all patches are included in the patch bundle. In Nortel's case, the kernel is not included in the patch bundle meaning that although both RHEL 4.1 and 4.2 machines were patched to a RHEL 4.3 level, they are not identical.

6. Users in a corporation typically do not have root access. For example, a user without root access cannot add a printer so printer configuration must be managed on a global basis. When users do not have root access, there are significant management and support implications. On the other hand, if the users do have root access, there are a whole different set of support and management implications.

7. Being able to identify an SOE machine remotely is not only desirable, but is also required from a systems management, licensing, lifecycle, and maintenance perspective.

8. Packaging all components of an SOE including in-house and third-party software in the same format as your Linux vendor's packages is important. Being able to easily upgrade if required (such as in the event of a security vulnerability), easily determining versions of software, being able to validate the authenticity of software (via digital signing), and being able to understand where files on a system came from are some of the benefits of this.

9. Communication to your deployment people as well as to those making use of the SOE is paramount to achieving success. Whether by use of WebPages, blogs, user groups, or other forms of documentation, communication gets more challenging as the size of your company grows.

10. ISV (Independent software vendor) support will probably be the single most important factor in determining which distribution your SOE is based on.

11. No matter how much you simplify an install or install process, deployment using installers without Linux experience will be a problem.

12. It is difficult to get patches from your Linux vendor fast enough (e.g., when you find a problem while producing the SOE or a patch bundle, both of which have deadlines to meet).

13. If a vendor says hardware is certified, what does that mean? Read the fine-print!

14. Despite every effort to the contrary, using third-party proprietary applications/code is a requirement that should be assumed in an enterprise (e.g. Clearcase).

15. A variety of products to choose from (KDE vs. Gnome, for example) makes standardization difficult when one must appease many palates.

16. Keep Global configuration (Level 2) and higher separate from the vendor install at all costs or you *will* be sorry!

17. Digitally sign *all* of your in-house packages. Being able to ascertain the authenticity of the packages contained within the SOE is important from a corporate and a security standpoint.

18. Have backup copies of *all* GA'd images. This will save future time and aggravation. At some point somebody will need to install an old image, either for testing or other purposes.

19. Change control is a critical component of SOE development (e.g. CVS, Clearcase, etc.).

20. Testing is important. Sadly, it will be the first thing to go when schedules are tight; having a testing matrix and test plan is paramount. A corollary to this: If someone tells you they have tested their product, but does not have a test plan, they are not telling the truth.

21. Fixes upstream are useless unless they are backported to the current SOE environment(s). (The Fix is Upstream BOF with Matthew Tippett)

22. One cannot move from proprietary UNIX's (Solaris/HP-UX) to Linux in one step, although new projects can start out quite well.

23. There have been and continue to be issues with Linux interoperating in a heterogeneous enterprise environment (e.g. assuming print servers are Linux as opposed to the non-CUPS-aware HP-UX). Do not assume your Linux vendor does any extensive testing using other operating systems your company uses.

24. Each package which you include in an SOE that does not come from the vendor needs to have someone who is responsible for it (a provider).

25. There is a large amount of proprietary thinking on the part of management that needs to be modified when using Linux and/or other open source software. An example is assuming that the Linux vendor that you are paying for support can fork some code to meet the corporation's requirements instead of the vendor waiting for the fix to be available from upstream. The Linux vendor's preferred method is to wait for the fixes to come from upstream (e.g. they would rather wait for the Evolution folks to fix the code than having to fix it themselves and then maintain the fix in subsequent upstream versions if the Evolution folks don't take the fix).

26. When creating an SOE, use of a vendor-supported Linux offering is recommended over an unsupported version. An example is the use of RHEL vs. Fedora. A supported Linux offering will have more and better ISV support. Updates tend to focus on customer problems and compatibility tends to be of higher importance than new features. And lastly, in an enterprise environment where downtime may mean the loss of substantial amounts of money, the ability to get support from a company is important from a business perspective.

5 Conclusion

The engineering and design of an SOE for an enterprise requires participation from throughout the corporation. For any large corporation, developing an SOE is worth the cost due to the many benefits it offers. The design is critical to an SOE's success and must reflect and solve real world business problems.

It is hoped that the many lessons learned in our SOE voyage at Nortel will help guide others pursuing the many benefits a Linux SOE has to offer.

6 References

[1] Office of Government Commerce, Information Technology Infrastructure Library. Retrieved April, 2007 from `http://www.itil.co.uk/`

[2] Aupek, Andrew, Architectural Design of Enterprise Wide Standard Operating Environments. Retrieved April, 2007 from `http://www.lib.mq.edu.au/about/conferences/Architectural%20Design%20of%20Enterprise%20wide%20Standard%20Operating%20Environments.pdf`

[3] Carnegie Mellon Software Engineering Institute, Capability Maturity Model Integration (CMMI). Retrieved April, 2007 from `http://www.sei.cmu.edu/cmm/`

[4] Edith Cowan University, IT Services Standard Operating Environment. Retrieved April, 2007 from `http://soe.ecu.edu.au/about/`

[5] Griffith University, Standard Operating Environment for Staff Desktop Computers. Retrieved April, 2007 from `http://www62.gu.edu.au/policylibrary.nsf/0/1e7a27d0e03ceed44a256ee00063ed6b?opendocument`

[6] Queensland University of Technology, Standard Operating Environment. Retrieved April, 2007 from `http://www.its.qut.edu.au/soe/`