Top 5 Strategies for Managing Linux

Practical Advice from the Field

White Paper
INTRODUCTION .................................................................................................................. 3

STRATEGY #1: AVOID DISTRIBUTION OVER-DIVERSIFICATION ........................................ 3
   Extreme Customization - The Double-Edged Sword ............................................................ 3
   Too Many Distributions, of Too Many Kinds .................................................................... 4

STRATEGY #2: AVOID DISTRIBUTION UNDER-DIVERSIFICATION .................................... 5

STRATEGY #3: LEVERAGE COMMODITY HARDWARE ......................................................... 6
   Price / Performance ........................................................................................................ 6

STRATEGY #4: SCALE-OUT WITH LINUX ........................................................................... 7

STRATEGY #5: USE LINUX-SPECIFIC MANAGEMENT TOOLS ....................................... 8
   Multi-OS System Management Tools .............................................................................. 8
   Simple Linux Management Tools .................................................................................. 8
   Sophisticated Linux Management Tools ...................................................................... 9

IN CLOSING .......................................................................................................................... 9
Introduction

Despite continuous and rapid evolution in the manageability of Linux, a 2006 survey\(^1\) cited manageability concerns as one of the top three reasons why organizations are hesitating to adopt Linux.

Having been in the business of Linux management and support since 1998, Levanta strongly feels that Linux can be as manageable, if not more so, than other operating systems by following a few key strategies. These strategic recommendations were developed from experiences in numerous customer environments, both large and small.

**Strategy #1: Avoid Distribution Over-Diversification**

The Linux universe is blessed with a flexibility and diversity that is practically unparalleled in the technology world. While this blessing has a number of benefits, too much of a good thing can lead to additional Linux management headaches.

The issue of over-diversification in a managed Linux environment takes two general flavors: customization of the Linux distribution(s) themselves and too many different installed distributions.

**Extreme Customization - The Double-Edged Sword**

While other *nix operating systems, such as Solaris, certainly offer the ability to customize configurations, Linux defines whole new frontiers of flexibility. The combination of custom configurations, vendor-supplied packages, third-party packages, downloaded software compiled from source, and modified open source projects allows Linux to be tweaked to fit very specific scenarios (or even just personal tastes). This is true even if you limit yourself to the major enterprise distributions like Red Hat or SUSE. Linux’s ‘tweakability’ can be a boon for squeezing out the best possible performance from a specific server, but if done ad hoc and in large numbers, is not without cost.

The net result can be a data center that has a bewildering diversity of different OS and application configurations requiring a huge amount of expensive administrative labor to maintain.

Defining standard configurations for various types of servers (web, application servers, database, etc.) is nothing new to IT management, but with the wide range of possible Linux customizations, it becomes much more critical. Using Linux management tools can assist in this task and turn it into an easy-to-follow operational habit.

Too Many Distributions, of Too Many Kinds

A quick glance at DistroWatch (www.distrowatch.com) gives you the Top 100 Linux distributions sorted by popularity…and this is only a fraction of the Linux universe. It’s likely that the number of Linux distributions available today dwarfs that of all versions of other major operating systems combined, over their entire histories.

When faced with this bountiful array of choices, it’s easy to go overboard with a desire to try all the choices, especially in situations where people are new to Linux. Often this experimentation extends beyond personal workstations to servers and even production deployments, creating management headaches.

The Linux Diversity Lifecycle follows the pattern below:

![Diagram of the Linux Diversity Lifecycle]

**Discovery Phase**
- First Linux deployments within an organization
- Often driven by one individual and a specific need (e.g. spam filtering)
- Proves viability of Linux within the organization
Proliferation Phase

Linux starts to take hold:

- Chosen to solve more point problems
- Often the distribution “the admin knows best” is adopted due to time pressure
- End result: A large number of different distributions being used for different tasks

Consolidation Phase

Limitations of the proliferation phase become apparent:

- Maintaining and patching different environments starts to consume more effort / cost
- Deep knowledge of distributions wanes as the original champions change roles or leave the organization
- Management mandates a standard Linux distribution to address these issues

Pragmatic Phase

Limitations of the 'single corporate standard' approach become apparent:

- Support costs increase as non-mission critical applications are standardized via the corporate mandate
- The 'best of breed' advantage of Linux for some applications is lost
- Organization feels pain of lock-in and blesses carefully considered alternative distribution(s)

Strategy #1: Conclusions

While in the midst of this roller coaster, it is important to keep a few points in mind if you’re going to avoid near-unmanageable chaos:

- Avoid personal favorites. You may be stuck with another person’s “favorite” long after they’ve left.
- Avoid mixing packaging formats. Supporting both .RPM-based and .DEB-based distributions in the same environment means a lot more work.
- Know what you have before standardization. Not knowing can lead to severely underestimating transition costs.

Strategy #2: Avoid Distribution Under-Diversification

After reaching the peak of unsupportable levels of diversification, the typical initial over-reaction is to adopt one distribution as the preferred choice across the enterprise. This immediately leads us to the next problem: under-diversification.

As noted in Figure #1, the downside of under-diversification is vendor lock-in. Vendor lock-in reduces your negotiating power with distribution vendors, which may lead to increases in licensing and support costs.

Under-diversification can also sometimes prevent you from selecting a Linux distribution that may be particularly well-suited to the task at hand for technological reasons.
To avoid under-diversification:

- Think twice before committing to using a single distribution for all purposes in your organization. You may not need the uptime assurance or guaranteed response times provided by enterprise distribution vendors for less-critical servers such as print servers and workstations.
- Evaluate using genetically-related distributions (e.g. RHEL, CentOS, and Fedora or SUSE and OpenSUSE) in a mixture that simultaneously avoids vendor lock-in, balances costs, and yet uses common packages and skill sets.

<table>
<thead>
<tr>
<th>Strategy #2: Conclusions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage distribution diversity to get the most value from your Linux investment.</td>
</tr>
<tr>
<td>- Smooth out the peaks and valleys of the Linux Diversity Lifecycle by knowing how Linux is being deployed and used in your organization.</td>
</tr>
<tr>
<td>- Be realistic when attempting to consolidate Linux deployments.</td>
</tr>
</tbody>
</table>

**Strategy #3: Leverage Commodity Hardware**

**Price/Performance**

The Intel-compatible platform is offering unprecedented levels of price/performance due to extreme competition between Intel and AMD and innovations from a number of smaller competitors. The rapid improvements in raw performance, power consumption, and specialized machine instructions are beginning to challenge the capabilities of single-vendor architectures. For many general purpose applications and most specialized technical and scientific applications Linux is the key to unlocking the power of commodity Intel-compatible hardware.

The predominant deployment of Linux is on 1 to 4-processor systems based on Intel or AMD CPUs. These systems are so inexpensive that they are almost disposable. Commodity Linux boxes are typically optimized for single use, and they are usually deployed as dedicated servers with multiple instances to increase redundancy and provide load balancing. Consequently, Linux server farms tend to have 5 to 20 times the number of systems of comparable UNIX SMP operations.

The popularity of large Linux-based server farms is growing for a number of reasons. Because they can be grown incrementally, scaling horizontally avoids difficult sizing and costing models required to justify the large up-front price of a 'Big Iron' installation. In addition, large single system machines are often tied closely to the vendor OS, reducing choice for the system purchaser and requiring less common (more expensive) staff to support the system.

<table>
<thead>
<tr>
<th>Strategy #3: Conclusions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux allows you to capitalize on the unprecedented price / performance of Intel-compatible hardware by adopting a horizontal scale-out model. Following the distribution diversification strategy described earlier ensures extra OS licensing fees don’t eat hardware savings every time the environment grows.</td>
</tr>
</tbody>
</table>
Strategy #4: Scale-Out with Linux

On commodity hardware, performance is increasing faster than reliability, especially when cost is factored in. Scaling horizontally reduces the dependencies on individual machines and reduces the impact of hardware failures on overall system availability.

In addition to deploying redundant scale-out systems, many organizations are further increasing the reliability and flexibility of their systems by reducing the number of high failure-rate components (e.g. disk drives, fans, etc.). Combining diskless servers with robust shared storage reduces hardware failure risks, while increasing system portability and hardware independence. Linux's strong plug-n-play abilities make it particularly well-suited for this setup.

For some applications, virtualization technologies provide additional insulation from hardware failures by offering features such as live migration of running of entire OS stacks. While this solution is not appropriate in all situations, many are finding a hybrid physical / virtual hardware platform is desirable. Where virtualization is appropriate, hardware scale-out can be achieved rapidly and with minimal disruption to servers.

**Strategy #4: Conclusions**

By combining Linux with reliable scale-out hardware, shared storage, and virtualization, reliability and scalability go up and management costs go down.
Strategy #5: Use Linux-Specific Management Tools

Tools that can be used to manage Linux environments generally fall into three buckets:

- Multi-OS system management frameworks (HP OpenView, IBM Tivoli, CA Unicenter, etc.)
- Simple Linux management tools (roll-your own scripting or basic patch management)
- Sophisticated Linux management tools (Novell Zenworks Linux Management, Levanta, etc.)

Let’s review some of the pros and cons of these options.

Multi-OS System Management Tools

When considering the various systems management solutions available today one inevitably thinks of the mature and established systems management products offered by the 'big dogs' in the industry. A cursory examination of these products suggests they are a good fit for the challenges of managing Linux environments.

However, a deeper examination reveals some common flaws with these tools. Linux support is often a recent addition to these product suites. As a result, Linux features can be incomplete or poorly integrated into the rest of the offering, undermining the value of a unified multi-OS tool. The unique challenges of managing Linux (including extreme heterogeneity, the plethora of patch/update sources and the popularity of customized packages built from source code) mean that even when Linux support is well executed, the challenges faced by Linux systems administrators are only partially addressed by the tool.

For example:

- Multi-OS tools usually have significant lag on how quickly they add support for updated distributions.
- Typically they only support the largest Linux distributions (e.g. RHEL and SUSE) while ignoring support for other distributions that may be critical for your vendor lock-in avoidance strategy.
- These tools typically don’t understand Linux software packaging. This means they rely on disk imaging; the images themselves become more numerous and difficult to manage over time.

One frequently overlooked challenge when adopting a multi-OS systems management solution is the cultural bias of system administrators from different technology backgrounds. The Windows system administration approach is very different from the 'hands-on' approach of Linux system administrators. Attempting to unify both camps under a single tool creates exceptionally difficult management challenges.

Simple Linux Management Tools

Almost all Linux distributions include some facilities for systems management, including provisioning and patch management. These offerings are worth considering because of the low price-point and the ease of evaluation.
Tools of this type typically focus on base OS provisioning and OS patch management as they’re the most
generalized and widely encountered problem. Few, if any, provide features such as application and
configuration deployment or strict change tracking. Many emphasize their ability to be extended using
scripting.

However, making extensive use of scripting can quickly cause problems. Reliance on a large body of in-house
scripts for deployment and configuration management creates a very undesirable situation where few people, if
any, truly understand how the scripts work. Any documentation that might exist is perpetually out-of-date,
placing the organization in a position of high operational risk.

Sophisticated Linux Management Tools

Given the limitations of both of the approaches above, “best of breed”, sophisticated Linux management tools
are worth considering. These focus specifically on solving the problems presented by managing Linux systems.

Your short-list of possible tools should offer enterprise-class management features such as:

- Fine-grained change tracking and control
- Controlled OS and application patch roll-out
- Support for diskless and reduced-disk environments
- Support for mixed virtualized and physical hardware
- Application and configuration management
- Enable OS and application stack portability independent of the underlying hardware

These capabilities should be achieved while:

- Maintaining desirable aspects of Linux itself, such as structured software packaging
- Preserving proven mechanisms of making ad hoc changes on Linux systems (e.g. ssh’ing into a system
  and making changes using vi)
- Enhance, rather than disrupt, existing trouble-shooting best practices

In Closing

While this paper does not attempt to address specific tactical concerns of Linux management, these strategies
should guide you in creating scalable and productive Linux management practices.

If you’d like to learn more about how Levanta technology can assist with your Linux management needs, feel
free to send your questions to info@levanta.com.