

Desktop Linux Feasibility Study Overview

Friday, August 20, 2004 – Mark Lachniet (mark@lachniet.com)

This document is a “sanitized” version of a document I wrote for another organization in the summer of 2004. There are some fairly obvious search and replacements of sensitive names and phrases, but some portions of the document could be of value to individuals or organizations investigating desktop Linux, or Linux in general. Unfortunately, the sanitization process has crippled the flow of the document, and makes it somewhat unwieldy in places. I apologize for this discontinuity, but hope that it will be of value anyway. I have tried to provide extensive documentation to support my assertions, wherever possible, with approximately 60+ citations in the document as it currently stands.

I put this document out for public consumption in hopes of promoting a better understanding of the issues surrounding Open Source Software in the enterprise. *This document and associated tools are to be used for non-commercial purposes only.* To obtain rights to the commercial use of this information, please contact the author for permission. If you think the paper is useful, I’d love to hear from you. If you have a valid argument that might challenge my thinking, I’d also love to hear from you. I have done my best to minimize my personal bias, and provide a fair and impartial treatment of the subject, but as with anything, this is impossible to do completely, so this document should be considered the subjective opinion of one individual.

The opinions expressed in this document are those of Mark Lachniet, and Mark Lachniet alone. I welcome any comments, questions or clarifications you may wish to offer. If there is sufficient interest in this document, I would consider revising it, in particular in areas that are especially weak such as the Vendors section. If you are an Open Source Software vendor, and would like to craft a section, in the format given, I would be glad to include your text in future versions of this document.

This document can be found at <http://lachniet.com/desktoplinux>

Thank you,

Mark Lachniet, CISSP, CISA, LPIC-1, MCSE, MCNE, CCSE, TICS
mailto: mark@lachniet.com

Table of Contents

DESKTOP LINUX FEASIBILITY STUDY OVERVIEW	1
TABLE OF CONTENTS	2
1 PROJECT OVERVIEW	3
1.1 EXECUTIVE SUMMARY	3
2 PROJECT REQUIREMENTS.....	5
2.1 PROJECT OVERVIEW.....	5
2.2 PROJECT SCOPE.....	5
2.3 EXPLANATION OF PROJECT METHODOLOGY	6
2.4 SHORTCOMINGS OF PROJECT METHODOLOGY	6
3 OPEN SOURCE SOFTWARE OVERVIEW	7
3.1 DEFINITION OF LINUX / OPEN SOURCE SOFTWARE.....	7
3.2 LINUX HISTORY AND TRENDS.....	9
3.2.1 <i>Linux as a “Disruptive Technology”</i>	9
3.2.2 <i>Success Story - Apache Web Server</i>	10
3.2.3 <i>Linux Adoption Overall</i>	11
3.2.4 <i>The City of Munich</i>	13
3.3 LINUX CAPABILITIES AND ISSUES.....	13
3.3.1 <i>Perceived Advantages to Linux and Open Source</i>	13
3.3.2 <i>Perceived Advantages to Microsoft Windows</i>	15
4 RESEARCH SUMMARY	17
4.1 GARTNER STUDY	17
4.1.1 <i>Findings of Gartner Study</i>	17
4.1.2 <i>Criticisms of Gartner Study</i>	18
4.2 STATSKONTORET STUDY	18
4.2.1 <i>Findings of the Statskontoret Study</i>	18
4.2.2 <i>Criticisms of the Statskontoret Study</i>	19
4.3 THE SOUTH AFRICAN STUDY	19
4.3.1 <i>Findings of the South African Study</i>	19
4.3.2 <i>Criticisms of the South African Study</i>	20
4.4 CYBERINSECURITY REPORT	20
4.4.1 <i>Findings of the CyberInsecurity Report</i>	20
4.4.2 <i>Criticisms of the CyberInsecurity Report</i>	21
5 LINUX VENDOR SUMMARY	22
5.1 VENDOR SUPPORT IS CRITICAL.....	22
5.2 KEY LDC LINUX VENDORS	22
5.2.1 <i>Vendor Overview: Novell</i>	22
5.2.2 <i>Vendor Overview: IBM</i>	22
5.2.3 <i>LDC Specific Vendors</i>	23
6 CRITICAL SOFTWARE.....	24
6.1 CRITICAL SOFTWARE OVERVIEW	24
6.2 NON-WINDOWS SOFTWARE OPTIONS	24
6.2.1 <i>Linux Native Binaries</i>	24
6.2.2 <i>Web Access</i>	24
6.2.3 <i>Remote Desktop</i>	24

6.2.4 Windows Emulation.....	25
6.2.5 Redevelopment and Replacement	25
6.2.6 No Migration.....	25
6.3 CRITICAL SOFTWARE DATA COLLECTION	26
6.3.1 Example Analysis: Office XP / Outlook / Microsoft Access.....	26
6.4 NON-CRITICAL BUT FISCALLY RELEVANT SOFTWARE.....	28
6.4.1 ZEN Works	28
6.4.2 HP OpenView.....	28
6.5 SOFTWARE REPLACEMENT MATRIX.....	29
7 USER COMMUNITY / CONVERSION COSTS.....	30
7.1 USER COMMUNITY / CONVERSION COSTS SUMMARY	30
7.2 USER COMMUNITIES	30
7.3 USER SOFTWARE BY COMMUNITY	31
7.4 TRAINING OPTIONS	32
7.5 SECURITY MANAGEMENT COSTS	33
8 SUGGESTED IMPLEMENTATION STEPS.....	34
8.1 IMPLEMENTATION OVERVIEW	34
8.1.1 Administrative Tasks.....	34
8.1.2 Application Development Tasks.....	35
8.1.3 Information Technology Tasks.....	35
9 APPENDIX ITEMS.....	37
9.1 DATA COLLECTION TEMPLATES	37
9.1.1 User Communities Data Collection Worksheet.....	37
9.1.2 Critical Applications Data Collection Form	37
9.2 COLLECTED DATA.....	37
9.2.8 Software Replacement Matrix	37
9.3 PAPERS REFERENCED	37
9.3.1 Dutch (Statskontoret) study	37
9.3.2 South African OSS Strategy v3.....	37
9.3.3 CyberInsecurity Report.....	37

I.1 Executive Summary

In the last five years, Open Source Software and the Linux operating system has had growing acceptance by enterprises similar to Lachniet Dot Com (LDC). Combined with increased support and partnership options, and bolstered by the recent acquisition of SuSE Linux by the Novell Corporation, the perception of Linux as a “risky” operating system has diminished significantly in recent years. This trend, combined with the cost issues associated with Microsoft software, has lead many organizations to analyze the benefits of Linux in the enterprise. In order to exercise due diligence and fiscally responsible I.T. planning, LDC has requested that Mark Lachniet (MJL) conduct a feasibility study to analyze the issues, benefits and costs of implementing Linux on the desktop.

Based on the data collected and research on the future of desktop Linux in the enterprise, Mark Lachniet believes that there are significant potential benefits to embracing Open Source Software at LDC. The recommendation to proceed with examining Open Source Software alternatives such as Linux is predicated on the following findings:

- **Vendor support.** Many commercial and non-commercial sources of support for Linux and Open Source Software now exist. Based on the continued participation of key industry players such as IBM and Novell, it is probable that this support will not only maintained at the current level, but increased significantly over time.
- **Software maturity.** While arguably not as mature as entrenched vendors such as Microsoft, Open Source Software has made significant strides in terms of stability, maturity and interoperability. It is believed that the current Open Source options can meet most, if not all, of LDC’s information technology needs. For those areas where viable alternatives do not exist, several alternate means of providing Windows software to Open Source platforms exist. Lastly, the viability of Open Source Software on the desktop has improved drastically in the last year.
- **Independence.** Due to the market domination of Microsoft, LDC is at risk of suffering from vendor “lock-in”. This dependence on a single vendor can have a number of negative impacts, including unfavorable software licensing terms, reliance on predefined feature sets and increased information security risks, among others.
- **Cost.** The cost justification for migrating to Open Source Software is rapidly improving. Based on preliminary findings, both in this study and in others, it would appear that the cost of converting to Open Source Software has reached a “break even” point. In terms of software costs alone, Open Source Software is already much more cost-effective. Unfortunately, software costs alone do not represent the true costs of Open Source Software – a number of other factors such as conversion cost, training, and implementation must be considered. While a full conversion may not be justifiable simply on cost at this time, a measured and rational conversion over time could well lead to significant cost savings.

Based on these findings, Mark Lachniet has drafted the following high-level recommendations to establish a migration roadmap:

- **LDC should determine the criteria for making a migration decision.** There a number of reasons for migrating to Open Source Software, as detailed in this study. LDC should determine what the driving reasons for conversion truly are, for example - cost, security, avoiding vendor lock in, etc. - and make a business decision based on these criteria. While this study touches on a number of issues to consider, only the LDC staff can adequately rank and prioritize them.
- **LDC should establish Linux Project Management capabilities.** Before any progress can be made on analyzing and implementing Linux and Open Source Software, it is critical that one or more individuals be given formal project management duties and adequate time to perform these duties. Without strong advocacy and oversight from competent and

dedicated project management staff, it will be difficult (if not impossible) to proceed in an organized and efficient manner. LDC may wish to hire a Linux “expert” with project management skills, or even a dedicated project manager for this purpose.

- **LDC should initiate a desktop Linux pilot project.** Many of the issues associated with desktop Linux are difficult to quantify, particularly end-user acceptance, helpdesk support and training issues. As each organization is unique, it is difficult to base long-term strategy on the experiences of other organizations and research studies. Performing a pilot will give LDC the opportunity to take advantage of future movements in the maturing desktop Linux space, and allow the organization to “hit the ground running.”
- **LDC should use a phased approach to implementation.** Rather than a disruptive “lift and replace” approach, a carefully planned phased approach will allow LDC to realize potential cost savings and identify possible stumbling blocks while minimizing the potential of creating organizational problems. For example, LDC might consider identifying pilot user communities where replacing the Microsoft Office product with OpenOffice.org is appropriate, and use the cost savings from these pilot programs to fund future desktop Linux initiatives. Additionally, to avoid the costs associated with a complete replacement, LDC should consider migrating workstations to Open Source Software as part of the workstation replacement process, and not as a separate project. In this way, the impact on desktop support, helpdesk and other internal resources can be made more manageable. From an end user perspective, success stories and positive word of mouth from pilot users would help to calm anxiety about a future platform change. With a phased approach, LDC will be able to reflect upon the “lessons learned” after each project phase is completed, and make an informed decision on whether or not to proceed with further implementation.
- **LDC should formally develop internal Linux support capabilities.** Using available training materials and certification programs such as the Linux Professional Institute (LPI), LDC should identify and train a group of Linux advocates and support staff. This will not only allow LDC to pursue projects such as a desktop Linux pilot, but will also bolster support in other areas of the organization where Linux is already in use such as server and database platforms. In addition, it would be worthwhile to establish a Linux planning team to track internal successes, vendor support and industry changes over time in regards to the Linux platform. Establishing personnel standards that encourage the hiring of staff with experience in open source platforms and software would also increase the capability of LDC to pursue an Open Source agenda over time.
- **LDC should ensure that all future internally developed applications are cross-platform capable.** In order to minimize the future difficulty of migrating internally-developed applications to a Linux-capable platform, all future programming should be done in languages that can be easily ported and used on the Linux operating system. In addition, applications should be developed with the minimum possible reliance on Windows-specific Application Programming Interfaces (APIs).
- **LDC should encourage its software vendors to embrace Open Source technology.** Software vendors respond to financial pressures when deciding future direction. If LDC is able to wield its significant purchasing power to encourage its software vendors to move towards Open Source platforms, it may ultimately save cost and expense in its own conversion efforts.

2.1 Project Overview

The overall goal of this project is to analyze the current viability of Linux on the desktop, as of July, 2004, with an emphasis on the “big picture” factors that would accompany a migration to a Linux platform. This project will therefore attempt to survey the current state of the I.T. industry, as well as the past, current and future of Open Source Software and Linux within it. As such, a broad sampling of issues must be researched and analyzed in as objective a manner as possible. This sampling must include a number of issues, ranging from Total Cost of Ownership (TCO) to vendor support to end user resistance to change.

2.2 Project Scope

Mark Lachniet met multiple times with Gerald Garcia and a group of key subject matter experts at LDC to define the scope of this project. As a result of these meetings, the project objectives were defined as the following:

1. Interview key LDC stakeholders regarding goals, issues and concerns regarding a possible Linux migration
2. Perform user education, where appropriate, about the rationale behind analyzing Open Source Software in the enterprise
3. Identify success indicators of a Linux migration
 - Cost savings / reduced TCO (including maintenance and support)
 - Security
 - Compatibility with existing applications
 - System stability, availability and performance
 - Access to source code
 - User acceptance
 - Ability to support Linux
4. Identify market trends and key partners in a Linux migration
 - Identify historical trends regarding the use of Linux
 - Key vendors: Novell, HP, IBM, etc.
 - Identify vendor roadmaps for evidence of long-term Linux support
 - Identify pertinent initiatives in similar organizations
5. Perform extensive research on the viability of Linux on the desktop
 - Case studies
 - Research
 - White papers
 - Direct user and administrator testimony
6. Identify a suggested approach to implementing a Linux desktop migration
 - Identification of user communities
 - Identification of software compatibility
 - Identification of internal support training requirements and resources
 - Identification of external support options
 - Identification of dependencies, stumbling blocks, etc.
 - Identification of a theoretical implementation timeline
 - Identification of preliminary, estimated costs per implementation stage
7. Document all gathered information in a single deliverable document

8. Present findings to LDC staff

2.3 Explanation of Project Methodology

To attain the goals detailed above, a number of tasks were undertaken simultaneously, including

- Analyzing the history and current state of Open Source Software
- Attempting to perform a TCO cost analysis study
- Analyzing external research, white papers and relevant documents
- Identifying LDC user communities, the Windows software that is used and comparable Open Source replacements
- Analyzing vendor support of Open Source Software, with an emphasis on key LDC vendors such as IBM and Novell
- Proposing a suggested roadmap for further analysis and implementation steps, above and beyond the work performed in this feasibility study

2.4 Shortcomings of Project Methodology

While this feasibility study document is wide ranging in its scope of analysis and consideration, it is therefore necessarily lacking in depth, and should be considered only a preliminary step. Due to the limited time allocated to this preliminary study, it would be impossible to analyze all aspects of Linux on the desktop in this process. Ideally, LDC will identify and study in greater detail these issues as it moves forward in its assessment work, perhaps using this feasibility study as a starting point for this transition.

In addition, while the scope of this project originally included provisions for analyzing the actual costs and benefits of Linux (a TCO component) the Linux feasibility study team was unable to gather adequate information from the various LDC departments in the time allotted to do this work. Due to this fact, this study will not include specific cost analysis data, and will instead put forward the TCO templates as an example of the type of cost analysis work that will need to be undertaken over the coming years.

3.1 Definition of Linux / Open Source Software

Open Source Software (OSS) can be broadly defined as software released under a number of software licensing schemes where the source code is made available to the public. Linux is the most popular and well-recognized Open Source operating system, and has made significant inroads to enterprise I.T. departments due to its low cost, features and stability.

Most Open Source Software is released under a specific license scheme, such as the GNU Public License¹ (GPL) or the Modified BSD License². For a sampling of the various Open Source Software licensing scheme, refer to the GNU License Philosophy web page³. In general, Open Source Software not only requires the distribution of source code, but places specific restrictions on how the source code can be used and modified.

The primary elements addressed by Open Source license schemes, as found on the Open Source Initiative web page⁴ generally include:

1. Free Redistribution

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

***Rationale:** By constraining the license to require free redistribution, we eliminate the temptation to throw away many long-term gains in order to make a few short-term sales dollars. If we didn't do this, there would be lots of pressure for cooperators to defect.*

2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost—preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

***Rationale:** We require access to un-obfuscated source code because you can't evolve programs without modifying them. Since our purpose is to make evolution easy, we require that modification be made easy.*

3. Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

***Rationale:** The mere ability to read source isn't enough to support independent peer review and rapid evolutionary selection. For rapid evolution to happen, people need to be able to experiment with and redistribute modifications.*

¹ <http://www.gnu.org/licenses/licenses.html>

² <http://www.gnu.org/philosophy/license-list.html#ModifiedBSD>

³ <http://www.gnu.org/philosophy/license-list.html>

⁴ <http://www.opensource.org/docs/definition.php>

4. Integrity of The Author's Source Code

The license may restrict source-code from being distributed in modified form *only* if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

Rationale: *Encouraging lots of improvement is a good thing, but users have a right to know who is responsible for the software they are using. Authors and maintainers have reciprocal right to know what they're being asked to support and protect their reputations.*

*Accordingly, an open-source license **must** guarantee that source be readily available, but **may** require that it be distributed as pristine base sources plus patches. In this way, "unofficial" changes can be made available but readily distinguished from the base source.*

5. No Discrimination Against Persons or Groups

The license must not discriminate against any person or group of persons.

Rationale: *In order to get the maximum benefit from the process, the maximum diversity of persons and groups should be equally eligible to contribute to open sources. Therefore we forbid any open-source license from locking anybody out of the process.*

Some countries, including the United States, have export restrictions for certain types of software. An OSD-conformant license may warn licensees of applicable restrictions and remind them that they are obliged to obey the law; however, it may not incorporate such restrictions itself.

6. No Discrimination Against Fields of Endeavor

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

Rationale: *The major intention of this clause is to prohibit license traps that prevent open source from being used commercially. We want commercial users to join our community, not feel excluded from it.*

7. Distribution of License

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

Rationale: *This clause is intended to forbid closing up software by indirect means such as requiring a non-disclosure agreement.*

8. License Must Not Be Specific to a Product

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

Rationale: *This clause forecloses yet another class of license traps.*

9. License Must Not Restrict Other Software

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

Rationale: *Distributors of open-source software have the right to make their own choices about their own software.*

Yes, the GPL is conformant with this requirement. Software linked with GPLed libraries only inherits the GPL if it forms a single work, not any software with which they are merely distributed.

10. License Must Be Technology-Neutral

No provision of the license may be predicated on any individual technology or style of interface.

Rationale: *This provision is aimed specifically at licenses which require an explicit gesture of assent in order to establish a contract between licensor and licensee. Provisions mandating so-called "click-wrap" may conflict with important methods of software distribution such as FTP download, CD-ROM anthologies, and web mirroring; such provisions may also hinder code re-use. Conformant licenses must allow for the possibility that (a) redistribution of the software will take place over non-Web channels that do not support click-wrapping of the download, and that (b) the covered code (or re-used portions of covered code) may run in a non-GUI environment that cannot support popup dialogues.*

Much confusion has been generated by these types of licensing schemes. In particular, there tends to be confusion about the “free” portion of “free software”, and many people feel that open source software is by definition free of cost. This incorrect understanding has led to the clarification that open source software is “free in the sense of liberty, not beer.” Although licensing issues are unlikely to create barriers to an eventual Linux migration, it would be advisable to involve legal counsel, particularly if LDC intends to modify and redistribute open source software code.

3.2 Linux History and Trends

Linux has been in development more or less constantly since 1991, and was originally started as a “hobby” by a Finnish gentleman named Linus Torvalds. Since that time, thousands of individuals and organizations have contributed to the collective set of software packages known as Linux. While once the domain strictly of computer geeks and hobbyists, Linux has made significant progress in terms of software quality and usability. As there already exists a significant body of literature and documentation on this topic, it is not necessary to reiterate the extensive available information on this topic. However, while not indicative of Linux on the desktop specifically, a few key trends are worth mentioning in this context to demonstrate the phenomenal growth of Open Source.

3.2.1 Linux as a “Disruptive Technology”

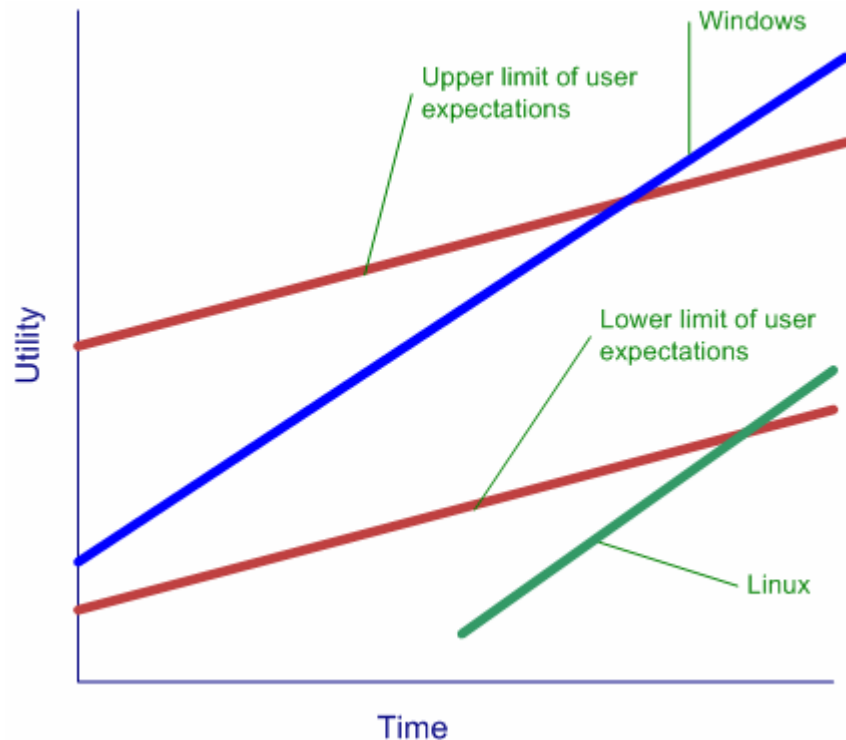
Linux has frequently been referred to as a “disruptive technology”⁵ in that it has radically changed the status quo of commercial computer software. It should be noted that disruptive, in this context, is to the status quo, and not to business operations. The most quoted theory of disruptive technology was in the book entitled “The Innovators Dilemma” by Clayton M. Christensen⁶. In this book, Christensen postulates that there are essentially two types of change, small and incremental, and large and profound. Linux obviously represents the latter, and takes its place with other radical innovations such as the printing press, electricity, and the production line. In many cases, these innovations were originally not seen as being necessary, and in some cases were actively resisted. It was not until later

⁵ <http://www.learnthat.com/define/view.asp?id=298>

⁶ <http://www.amazon.com/exec/obidos/tg/detail/-/0060521996/104-0675550-9376715>

that they were adapted for other purposes and became commoditized that their true potential was realized. This same model can be extrapolated to the use of Linux and Open Source Software.

In particular, it is important to analyze the way in which technology is invented, developed, and used. One theory holds that as products are developed, they go through a maturation process whereby they eventually meet, and exceed customer expectations. To better understand this concept, consider the following diagram from Jon Lam's web site⁷:



As is evident from this diagram, it is the opinion of Mr. Lan that while Open Source Software has finally surpassed the lower limit of user expectations, Microsoft products have far exceeded them. In essence, customers are now paying for features in Microsoft products that they no longer need or use. For example, many users do not use features in Microsoft Office that have been added since Office 97. Conversely, the capabilities of Open Source Software, for example OpenOffice.org have finally achieved enough features to perform the vast majority of required tasks at a much lower price. Indeed, one study⁸ has found that for users with no prior experience on either platform are able to perform tasks at nearly the same speed on a Linux desktop as on a Windows XP desktop. Perhaps more importantly, the *rate* (in this diagram the slope) of development of Open Source Software is proceeding at a much faster rate than that of user expectations, leading to the conclusion that it will rapidly eclipse the needs of even advanced users. If this model is correct, it bodes well for the long-term prospects of Linux adoption in enterprises such as LDC.

3.2.2 Success Story - Apache Web Server

Apache, the Open Source web server, has quickly become the most popular web server on the Internet, displacing even commercial products such as Microsoft's Internet Information Server. Netcraft, an organization that tracks and compiles data on web server statistics, has been monitoring the number and type of Internet web servers for several years. This monitoring gives a direct indication of web server usage on the Internet at large. While not representative of Linux on the

⁷ <http://www.iunknown.com/000226.html>

⁸ http://www.infoworld.com/article/03/08/04/HNusabilitystudy_1.html

desktop, it is indicative of the decisions that I.T. staffers have made in selecting their technology of choice over time.

As part of their monthly Web Server Survey⁹, the Open Source Apache web server is compared directly to its' Microsoft and Sun Microsystems competitors. As of April, 2004, given a sampling of 49,750,568 unique web sites, Apache comprised 67.20% web sites, while Microsoft and Sun came in at 21.02% and 3.44% respectively (see figure below). When viewed over time, it is clear that Open Source Software, at least in regards to web servers, has been rapidly adopted as a “best of breed” technology, despite the lack of a commercial vendor offering.

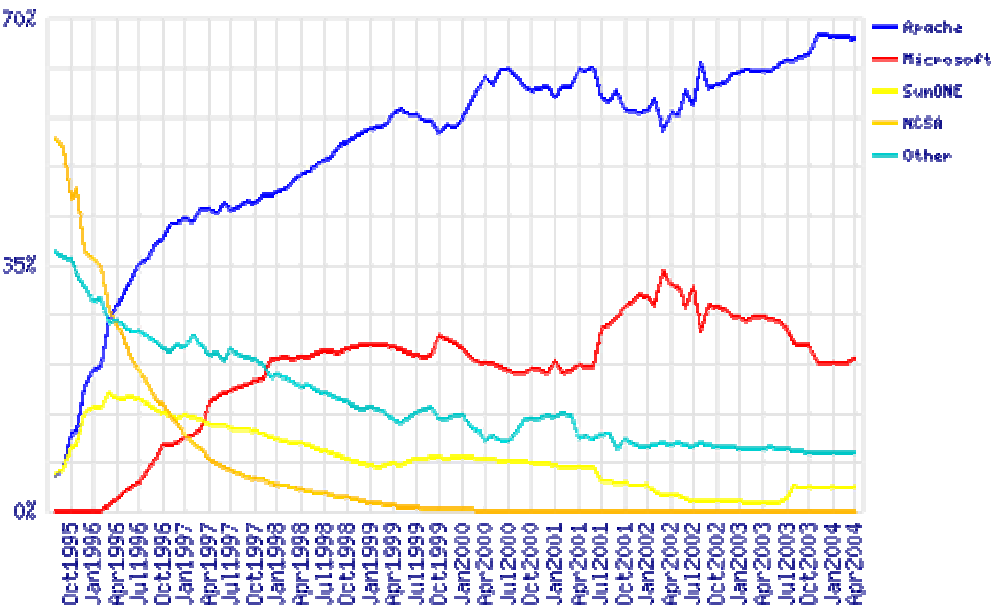


Figure 1 – Netcraft Web Server Survey

Developer	March 2004	Percent	April 2004	Percent	Change
Apache	32280582	67.20	33329879	66.99	-0.21
Microsoft	10099760	21.02	10691683	21.49	0.47
SunONE	1651575	3.44	1661229	3.34	-0.10
Zeus	762716	1.59	763302	1.53	-0.06

3.2.3 Linux Adoption Overall

A number of studies have been conducted regarding the overall adoption of Linux in the IT industry. However, as might be expected, these studies vary greatly in approach and focus, yielding vastly different (and sometimes confusing) results. In addition, many studies have been paid for by organizations with a specific motive, such as Microsoft or Open Source Advocates. For this reason, all such studies should be interpreted critically, and not used as a stand-alone justification for any course of action.

In general, Linux usage can be broken into three broad categories:

- **Edge / utility devices.** The niche in which Linux fits most easily, perhaps due to its enthusiastic adoption by I.T. staff, is in the areas of network utility servers such as firewalls,

⁹ http://news.netcraft.com/archives/web_server_survey.html

proxy servers, vulnerability assessment tools, and network monitoring tools. As many of these devices are configured and used by individual engineers, and not part of a concerted organizational effort, obtaining statistics for this category is difficult. However, informal and “word of mouth” shows significant deployment in these areas.

- **Linux servers.** The use of Linux as a server platform provides the most tangible evidence of industry adoption of Open Source Software. In particular, Linux has extensive penetration in functions such as web and mail servers, database servers and application servers. Although it is nearly impossible to track the usage of Linux servers, due in large part in the inability to track how no-cost software is downloaded and installed, some statistics on commercially purchased Linux software do exist. For example, according to a summary¹⁰ of the IDC study released in 2003:

“Microsoft's Windows accounted for 55.1 percent of new shipments of server operating systems in 2002, up from 50.5 percent in 2001, while paid versions of Linux accounted for 23.1 percent of new shipments in 2002, up from 22.4 percent in 2002”

With these statistics in mind, a few mitigating factors should be considered. First, one might reasonably assume that *purchased* copies of Linux make up only a fraction of actual real-world usage, especially in the consumer and Small-Medium Business (SMB) space. Second, these statistics (from the 2002 financial year) represent a snapshot in time where major vendor support (notably HP, IBM, Novell / SuSE, Red Hat) were not as mobilized as it currently is in terms of marketing and support. As a result of this, some analysts speculate that server sales from FY2003 and beyond will show a significant increase. Indeed, as noted in the referenced article:

“IDC projected that Linux platform revenue will increase at more than four times the overall industry average for all platforms through 2007. In the [Server Operating Environment] market, paid Linux OS software will comprise 32.3 percent of all server shipments by 2007, IDC forecasted, with 2.8 million new Linux paid [Server Operating Environments] being shipped each year.”

Such claims are highly speculative, and should be considered as such. Only time will tell if these predictions are accurate, both in terms of over-estimation and under-estimation of future adoption.

- **Desktop Linux.** The use of Linux as a desktop platform, the specific focus of this feasibility study, represents the least-embraced form of Linux adoption. In this area, usage statistics are particularly difficult to obtain, largely due to the unrestricted downloading and installation of the Linux OS. However, a couple of sources do exist to quantify this. First, according to statistics from the previously mentioned IDC study:

“According to the report, license shipments by Microsoft, on the client side, increased to 93.8 percent of the worldwide market in 2002, up from 93.2 percent in 2001. In 2002, paid Linux client license shipments accounted for 2.8 percent of the market total.”

And interestingly...

“The Mac OS, from Apple Corp., accounted for 2.2 percent of the COE market in 2002, to take the second place position after Microsoft. By 2004, the Mac OS will concede that place to Linux, IDC forecasted.”

An additional source of qualitative information on desktop Linux usage comes from the Google search engine's Zeitgeist web page¹¹. According to the April, 2004 statistics, Linux users account for only 1% of Google search requests. While this is not a particularly encouraging statistic, it also may not be indicative of real world usage. For example, many

¹⁰ http://www.infoworld.com/article/03/10/08/HNmsdominance_1.html

¹¹ <http://www.google.com/press/zeitgeist.html>

Linux users are likely to make their Google requests from a workstation at their place of employment, where Windows desktops are prevalent. Additionally, users of the Google service are could use any number of search engines besides Google, such as search engines that are focused towards specific regions or languages. Due to the disproportionate adoption of Linux in developing non-English speaking countries¹², the statistical validity of the Google information is dubious at best.

In summary, the history of Open Source Software has been one of rapid adoption, particularly in the data center and by technical staff, with a notable weakness in adoption by end users. However, based on the work of most objective analysts, it seems likely that desktop Linux will continue to grow and mature as a viable option, indicating that continued testing and analysis is a prudent use of organizational resources.

3.2.4 The City of Munich

On May 28th, 2004, the City of Munich in Germany officially announced that it will move 14,000 PC computers and 16,000 users from the Windows environment to Linux¹³. Although specific details of the change have not been announced, including the final decision on hardware and software vendors, IBM and Novell / SuSE Linux were heavily involved in the process. Although cost considerations were part of the decisions process, it is interesting to note that “the decision was a matter of principle: the municipality wanted to control its technological destiny. It did not wish to place the functioning of government in the hands of a commercial vendor with proprietary standards which is accountable to shareholders rather than to citizens”¹⁴.

3.3 Linux Capabilities and Issues

Linux, and Open Source Software in general, tends to have a polarizing effect in terms of personal opinion. Unfortunately, this polarization makes it difficult to make informed decisions about the *business value* of the technology. On the one hand, Microsoft has made a concerted effort to discredit and compete with Linux at every turn, even going so far as to make available a set of discretionary funds to be used in competitive bid situations against Linux alternatives¹⁵. On the other hand, Linux enthusiasts have been accused of vastly overstating the benefits of Open Source software, while over-emphasizing the shortcomings (particularly in cost and security) of Microsoft products. While each perspective has some merit, it makes an objective analysis difficult. While extensive discussion about the pro's and con's of Open Source Software already exists, it is relevant to this study to briefly touch on some of the key points. Interestingly, many of the points (such as technical support) are claimed to be both an advantage as a disadvantage, as noted below.

3.3.1 Perceived Advantages to Linux and Open Source

Open Source Software is touted as having several significant advantages over commercial software. However, these claims have varying levels of merit, and all software (Open Source or commercial) should be considered in light of its place in the overall I.T. environment. The following are generally believed to be advantages to Open Source Software:

- **Security.** Open Source Software is believed to have better security for two main reasons. First, it is possible to analyze the source code of the application to identify security risks, and fix them before a problem occurs. However, this “feature” can work both ways, allowing a dedicated hacker to find security holes as well. Also, as has been noted by commercial

¹² <http://zdnet.com.com/2100-1104-1000992.html>

¹³ <http://www.desktoplinux.com/news/NS3199247984.html>

¹⁴ http://www.economist.com/business/displayStory.cfm?story_id=2054746

¹⁵ http://www.infoworld.com/article/03/05/16/HNdiscount_1.html

software companies, just because someone *can* scan an application for holes, doesn't mean that they *will* do so, especially when there is no financial reward for doing so. Secondly, it is believed that Open Source Software developers respond more quickly to software issues, and tend to release fixes and patches much more rapidly. While some studies have questioned this claim¹⁶, solid evidence exists that some critical flaws have been reported to Microsoft for as long as 200 days before being resolved¹⁷. The primary issue with time-to-patch issues seems to be the need to retroactively test patches to system libraries and binaries, so as not to release patches that cause problems when applied. In this aspect, perhaps due to the modular nature of software on a Linux distribution, Linux does indeed have an advantage.

- **Performance.** Linux is frequently believed to be a higher-performance operating system, when used on the same hardware. A number of studies have been conducted on various common functions, including file serving, web serving and database performance. Without going into the details and criticisms of each study, it is reasonable to state that the performance of any application depends on the specifics of the software, how it is used, and how it is tested. For example, Microsoft IIS has frequently benchmarked higher than the Apache web server, while certain databases, namely Oracle, seem to run faster on the Linux platform. It is recommended that LDC analyze the performance of critical business applications on a case-by-case basis prior to migration.
- **Stability.** Linux is considered to be a more stable platform, and less prone to crashes or the "blue screen of death." In regards to stability, hard data is difficult to obtain, but anecdotal evidence (e.g. user testimony) does indicate that this opinion is well-founded. Again, LDC should analyze the stability of critical business applications on a case-by-case basis prior to migration. It may well be that existing applications are sufficiently stable, and that this is not an important criteria.
- **Cost.** The cost of some Open Source Software is indeed compelling. However, as noted in this feasibility study, the purchase price of software is not the only factor to be considered when analyzing organizational Return On Investment (ROI).
- **Access to source code.** One benefit to Open Source Software that cannot be disputed is that of source code access. Although restrictions do exist on the *redistribution* of modified source code, there are generally no restrictions on modifying source code to meet the needs of the user. For this reason, organizations with internal development capabilities can achieve a significant benefit from Open Source Software. Depending upon the development activities at LDC, this may be a significant advantage.
- **Reduced vendor lock-in.** The issue of "vendor lock-in", and the subsequent risk that a vendor will abuse this dominance is an actual risk. As discussed in a later portion of this study, many have questioned the wisdom of such widespread adoption of a single vendor's software. The recent anti-trust cases against Microsoft in the United States and the European Union give some credence to the idea that Microsoft has abused their dominance. However, it should be noted that Linux, in the form of a branded distribution such as SuSE or Red Hat also presents vendor lock-in issues. In particular, unless an organization is willing to deploy their own patches and feature upgrades, as opposed to waiting for updates from their Linux vendor, there will still be a reliance on a single vendor.
- **Technical support.** Many open source advocates believe that the technical support, although often informal, for Open Source Software is superior to commercial alternatives. The primary benefit cited by Linux enthusiasts is that they can simply "e-mail the person who wrote the software", often getting a response more quickly than would be possible through regular technical support channels. While this may currently be true, this is hardly a benefit that LDC would wish to count on, as there is no guarantee of support, free or otherwise for many Open Source Software packages.

¹⁶ <http://www.forrester.com/Research/Document/Excerpt/0,7211,33941,00.html>

¹⁷ <http://www.informationweek.com/story/showArticle.jhtml?articleID=17700169>

3.3.2 Perceived Advantages to Microsoft Windows

A number of organizations, most notably the Microsoft Corporation, have made a concerted effort to stress the value of their commercial software offerings versus Open Source solutions such as Linux. The Microsoft Corporation, with their “Get the Facts”¹⁸ web site has even begun a vigorous marketing campaign, stressing the advantages of Windows in the enterprise. In addition, a number of studies, both funded by Microsoft and independent, have observed several perceived advantages held by the Microsoft Windows operating system:

- **Availability of support engineers.** Simply put, there are a large number of I.T. staffers who understand the various versions of the Windows operating system. However, it should be noted that there is a vast difference between the number of MCSE engineers certified in Windows NT (394,807)¹⁹ and those certified in the Windows 2003 platform (5,604), although this will surely change as the more modern technology is embraced. In addition, it has been suggested that the cost to hire a Microsoft engineer, perhaps due to the large number of candidates to choose from, is significantly less than that for comparable Linux administrator. In a study conducted by Certification Magazine in the fall of 2003²⁰, it was found that the average Microsoft MCSE candidate earned \$64,000, while a comparable Red Hat Certified Engineer (RHCE) earned \$71,660. Interestingly, the two vendor-agnostic Linux certification salaries came in significantly lower, at \$68,170 for a Linux Professional Institute certified engineer, and \$52,380 for a CompTIA Linux+ certified engineer.

Certification Magazine Fall 2003 Salary Survey	
Certification	Average. Salary (USD)
Microsoft MCSE	\$64,000
Red hat RHCE	\$71,660
LPI LPIC	\$68,170
CompTIA Linux+	\$52,380

The claims regarding the lower cost of hiring Microsoft support engineers seems to be borne out by a number of salary studies. However, a few interesting criticisms of these studies have been made that are worth considering:

1. **Engineer quality.** The Microsoft certification program has received much criticism regarding the low quality of MCSE-certified engineers. In particular for early versions of the certification (prior to Windows 2000), it was widely regarded as being far too easy to pass, resulting in a large number of “paper MCSE’s” who had passed the test (often through boot camps or cram sessions) but did not have real-world experience. Recently, the quality and difficulty of the MCSE certification has been increased, and those engineers who have passed the Windows 2000 or 2003 MCSE certifications are regarded as being of a higher caliber than NT 4 MCSEs.
2. **Engineer skill set cross-over.** While quantitative data for this aspect is difficult to find, anecdotal evidence suggests that the cross-over skill set for Microsoft MCSEs is not as comprehensive as that of engineers skilled in UNIX and Linux operating systems. Simply put, Microsoft engineers rarely know anything about Linux, while Linux engineers frequently know a good deal about the Microsoft operating system. Thus, for an enterprise that must support both Microsoft and Linux operating systems, hiring Linux engineers

¹⁸ <http://www.microsoft.com/getthefacts>

¹⁹ <http://www.microsoft.com/learning/mcp/certified.asp>

²⁰ http://www.certmag.com/articles/templates/cmag_feature.asp?articleid=523

may allow for a broader base of operating system support ability and therefore be more cost effective.

- **Availability of technical support.** Aside from hiring in-house engineers, the availability of commercial technical support should be considered. In this arena, due in part to the large number of certified engineers, the established presence of the operating system, and the large number of support vendors, Microsoft has a clear advantage. However, in the past, many claims were made that Linux is not a commercially supported operating system. While Microsoft may be supported by a larger number of companies, it is possible to purchase quality commercial Linux support from a number of organizations including Linux vendors such as Red Hat, Novell, IBM, as well as commercial support companies such as Linuxcare²¹, The Linux Box²² in Ann Arbor, Michigan and others. In addition, many developers of Open Source software provide free or commercial support directly for their products. For example, it is possible to purchase technical support for the popular MySQL database platform directly²³.
- **More mature software.** As a commercial software vendor, Microsoft has put extensive effort into developing software that is user friendly and mature. While some may complain about the security risks of Microsoft software, the end user experience in terms of usability and user interface is generally better with Microsoft products than with comparable Open Source packages. Some exceptions, notably software such as the Open Source Mozilla browser, do exist. In addition to the user experience, there is a wider variety of applications that have been written for the Windows operating system. While some software, such as Microsoft Office, does have an analog in the Open Source world, the vast majority of business applications are Windows-based.
- **Enterprise management features.** Due to the integration between the Windows server and client platforms, Microsoft has an advantage when it comes to managing large numbers of machines. Using software such as the Novell ZEN Works package, or built-in features such as Active Directory's group policy, it is possible to easily configure and provide services to large numbers of machines. Although Linux does have extensive scripting capabilities, as well as the ability to automatically configure and update machines through products such as SuSE's Red Carpet²⁴, these products are not currently as robust as comparable Windows applications.
- **End user comfort level.** Many users, particularly non-technical users, use Microsoft software both at work and at home. For this reason, many end users feel that they are already well acquainted with the Windows operating system and Microsoft Office products, and not wish to change to a new platform. This comfort with Microsoft products can be a two-edged sword, however. On the one hand, a user who is comfortable with an application may need less support, and hence generate less technical support costs when using Microsoft applications. On the other hand, this familiarity may also lead a user to "tinker" and operate outside of their technical comfort zone, possibly leading to the installation of unsanctioned applications, dangerous changes to the operating system, etc. In an environment with a locked down desktop, this concern may be lessened significantly.

²¹ <http://linuxcare.com>

²² <http://www.linuxbox.com>

²³ <http://www.mysql.com/support/>

²⁴ <http://www.ximian.com/products/redcarpet/>

A number of white papers and studies have been released analyzing Linux in the enterprise. As part of this feasibility study, a large number of these documents were analyzed. The results of these studies and research projects vary drastically, perhaps due in part to their scope of focus, level of detail, and methodology. In addition, many of the studies were contracted by an interested party such as IBM or Microsoft, and as such were geared towards studying an area where they “thought they could win” the analysis. For this reason, while many such suspicious studies exist, they will not be summarized in this document. Rather, studies that were conducted without direct vendor financing will be discussed.

The following documents are provided as supplemental material, and were considered as part of this feasibility study. Where possible, a high-level overview of the studies will be presented, as well as a summary of the study’s findings. When specific criticisms about a study were identified, these are noted as well. Hopefully, this approach will allow for a more rounded interpretation of the documents and their shortcomings.

4.1 Gartner Study

In 2003, Gartner released a number of documents analyzing the viability of Linux on the desktop. These documents, available only by subscription or purchase have the following ID’s and titles:

115640 Linux Desktop TCO Labor Details

115725 Linux Desktop TCO An Overview.pdf

115729 Linux Desktop TCO Migration Cost Model.pdf

115733 Linux Desktop TCO Hardware and Software Details.pdf

In these documents, a detailed approach to Total Cost of Ownership (TCO) is used to quantify the cost of various desktop Linux options.

4.1.1 Findings of Gartner Study

The fact that Gartner performed an analysis of the TCO of Desktop Linux is evidence enough that the business world is taking note. The Gartner study attempts to take a “holistic view of costs across the enterprise over time” and includes a number of criteria including hardware, software and support costs based on an organization with 2,500 users. The study explicitly compared Microsoft Windows 95 and XP with Microsoft Office to two Open Source scenarios. In the first scenario, a Windows XP operating system was used with the Star Office software, and in the second, a Linux OS with Star Office was used.

Overall, the study found that Linux solutions were similar in TCO costs to comparable, modern Microsoft equivalents. Interestingly, Linux fared significantly better than Windows 95 solutions. The study makes certain interesting assumptions. First, the figures show a cost for Microsoft Office at \$101/year per user, or \$303 for the duration of the study. Comparatively, LDC is currently paying a one time cost of \$340 for Office 2003, for a difference of -\$37 from the study’s results. Additionally, Gartner assumes that Microsoft Office will be replaced with Star Office (as opposed to OpenOffice.org) for a cost of \$35, for a difference of \$35 from the study’s results. With these differences in mind, a cost difference of \$72 would result in findings of \$5,331 for a WinXP/OpenOffice.org combination and \$5,375 for Linux/OpenOffice.org.

The Gartner study also notes that there is a significant cost difference between “locked down” solution in a “well managed” environment, and environments where technology is not as well managed. In the specific instance of a Linux/StarOffice installation, it was actually the cheapest option, coming in at \$3,308, as opposed to a “well managed” WinXP/MS Office installation at \$3,317 (see below). Due to the significant impact of management systems on overall TCO, LDC should

evaluate its I.T. management practices to determine where it falls on the maturity continuum as part of its evaluation criteria.

In general, the Gartner study seems to indicate that the TCO costs for all of the assessed solutions (excluding Windows 95) were fairly similar. The study goes to great pains to note that different user communities have different needs, and that some user communities (specifically low-end workers who primarily use office applications) are a particularly good fit, while others (such as high-end “knowledge workers”) are not. A difference of an additional \$162 over three years for an all-Linux solution is not as wide a difference as might be expected, and could be easily justified based on the needs of the organization. Also, as noted below, the Gartner study did not address a few critical aspects of desktop TCO, including the cost of security.

4.1.2 Criticisms of Gartner Study

The criticisms of the Gartner study, from analysts such as Benjamin Robson and Con Zymaris²⁵ generally find fault on a number of minor points, but most specifically on the unmentioned costs of security. Specifically, the Gartner study explicitly did not analyze the cost of system security, namely patching and incident response, in their study. For many organizations, this is a significant source of costs, both in terms of labor and in terms of software to manage security. While a Linux installation also needs to be patched on a regular basis, these patches are typically more granular – pertaining to a specific package – and need not be applied to each and every workstation. Thus, while the average Windows workstation requires most patches to be applied, the average Linux desktop client (i.e., not running a web server or similar server daemon) would only require that a small number of patches be applied. Also, the vast majority of viruses and worms target the Windows platform, so major events requiring incident response are significantly less. This situation is likely to change as Linux sees increased adoption, and hackers create attacks against the platform.

4.2 Statskontoret Study

Statskontoret, the Swedish Agency for Public Management, released a study²⁶ in late 2003 regarding their findings and recommendations about implementing Open Source Software in government agencies. This study, like many other studies being conducted in European countries took a somewhat different viewpoint from those studies conducted in the United States. In particular, there was a much greater emphasis on goals such as promoting competition in the software market and minimizing vendor lock-in. Given the recent anti-trust lawsuits between Microsoft and the European Union²⁷, this is not surprising. The Statskontoret also touched on the logistics of embracing Linux in public government, particularly in establishing a procurement system that allows for the consideration of Open Source Software. For the full text of this study, please refer to *Appendix 9.3.1*.

4.2.1 Findings of the Statskontoret Study

The Statskontoret study closes by recommending that “a forum be created in order to facilitate and accelerate the use of free and open source software” with the goal that this be done “as soon as possible.” In addition, it made the following general recommendations that generally support the adoption of open standards and technology, including Linux, in the Swedish public arena:

- In order to avoid locking-in effects and to attain interoperability, an administration must define and place demands on open standards when procuring and developing systems. This pertains especially to file formats for office software and systems, which include communication with the general public.
- When communication with the general public via websites, an administration should work for website compliance with open standards (according to W3C) and avoid discriminating against individual products.

²⁵ <http://www.theage.com.au/articles/2003/09/16/1063625013703.html>

²⁶ www.statskontoret.se/pdf/200308eng.pdf

²⁷ <http://news.com.com/2100-1014-5178465.html>

- Work should be carried out for the inclusion of the most common open office applications in those covered by ECDL, the European Computer Driving License. For example, software such as Linux, OpenOffice and MySQL should be included in the ECDL.
- A web-based software catalogue with information on free and open source software should be available within Swedish public administration.
- A project should be started for defining standard solutions based on free and open source software, aimed at school and educational systems, preferably also in cooperation with corresponding projects in Denmark, Finland and Norway.

Clearly, the Statskontoret is taking the position that it has a responsibility to accept, if not promote, open standards and software in government.

4.2.2 Criticisms of the Statskontoret Study

Unfortunately, the Statskontoret study does not justify many of the arguments it makes. For example, the paper states that the advantages to Open Source Software include “increased security” and “increased quality and stability” without providing supporting documentation.

4.3 The South African Study

In January, 2003 the Government Information Officer’s Council (GITOC) of South Africa released a study entitled “Using Open Source Software in the South African Government.”²⁸ Once again, this study analyzed the benefits of using Open Source Software in a government. This document seemed to encompass a wider variety of issues, and included an analysis of where Open Source software is *not* a good fit for the South African government. For the complete text of this document, please see *Appendix 9.3.2*.

4.3.1 Findings of the South African Study

The South African study made many of the observations that have already been discussed in this document. However, the document also made a few additional observations that are worth noting.

- **Level Playing Field.** The study discussed not only adopting Open Source Software, but open software development practices in general. In addition, the study recommends implementing purchasing procedures that ensure that open source software is given equal, if not greater preference in the purchasing process. For example, in section 10.2, the study recommends that “where the direct advantages and disadvantages of OSS and PS are equally strong, and where circumstances in the specific situation do not render it inappropriate, opting for OSS will be preferable.” In addition, it made the following purchasing recommendations to promote a level playing field (original document citations are used):

117. Avoid any bias against OSS solutions in Government procurement procedures.

118. Wherever possible, avoid acquisition of hardware that does not support OSS.

119. ITAC will ensure that tenders and contracts are free of any specifications that unjustifiably discriminate against OSS.

120. The Procurement Standing Committee of GITOC will work with SITA to find the best practical ways of implementing this principle.

121. The Communication Strategy mentioned below will, among others things, aim to remove any biased mindset that may exist among relevant users and decision makers.

²⁸ http://www.oss.gov.za/docs/OSS_Strategy_v3.pdf

122. Tender evaluation teams will be equipped to deal with the relevant options fairly.

- **Implementation Strategy.** The study also details a specific implementation strategy, with assigned responsibilities and timelines. This strategy can be found on page 28 of the document.
- **Shortcomings of Open Source Implementation and Development.** The study explicitly notes that the Open Source implementation process has some shortcomings, and requires dedicated resources to ensure success. For example:

“OSS projects require a large user base to provide the necessary volunteers – projects must be useful and interesting to the OSS community that is hoped will contribute. This immediately rules out projects that are highly customised, such as website development and niche applications. Also, since it is more difficult to manage deadlines with an OSS project, it may be risky to consider OSS development for projects with critical short-term deadlines (unless specific countermeasures are in place).”

And...

“Managing an OSS project is an involved process, the details of which are often overlooked. An OSS project requires initial funding for development, as well as funding for a deployment system (concurrent version control, bug tracking, mailing lists, etc.). Most importantly, a successful OSS project needs a champion, or gatekeeper – a skilled individual who will take responsibility for the project, make strategic project decisions and prevent forking.”

While application development practices are not directly relevant to enterprise support of Desktop Linux, the point is well taken that success with Open Source Software requires “champions” who are well versed in the software, issues and communities surrounding Open Source Software.

4.3.2 Criticisms of the South African Study

Once again, certain statements regarding the positive benefits of Open Source Software are given without supporting documentation. In particular, the security benefits of Open Source Software are touted, without a qualitative analysis.

4.4 CyberInsecurity Report

In a report entitled “CyberInsecurity: The Cost of Monopoly – How the Dominance of Microsoft’s Products Poses a Risk to Security”²⁹ dated September 24th, 2003, a number of high-profile security Lachniet and academics raised grave concerns about the security implications of a Microsoft-dominated computer industry.

4.4.1 Findings of the CyberInsecurity Report

In this paper, it is postulated that there is a significant security risk, both to individual organizations specifically and to the technological world in general because a single platform (Microsoft Windows) is used by the vast majority of organizations. Due to the fact that computer software is now absolutely necessary to be financially viable, this insecurity translates directly to a financial risk due to

²⁹ www.ccianet.org/papers/cyberinsecurity.pdf

the use of a single platform. These sentiments, along with the corollary issue of “vendor lock-in” have been echoed in many other studies, and were the basis of decisions from organizations such as the German city of Munich. For the complete text of this paper, please refer to *Appendix 9.3.3*. Among the main points of the CyberInsecurity report are the following excerpts:

- Our society’s infrastructure can no longer function without computers and networks.
- A monoculture of networked computers is a convenient and susceptible reservoir of platforms from which to launch attacks; these attacks can and do cascade.
- Risk diversification is a primary defense against aggregated risk when that risk cannot otherwise be addressed; monocultures create aggregated risk like nothing else.
- Microsoft is a near-monopoly controlling the overwhelming majority of systems.
- Microsoft has a high level of user-level lock-in; there are strong disincentives to switching operating systems.
- Microsoft’s operating systems are notable for their incredible complexity and complexity is the first enemy of security.
- The near universal deployment of Microsoft operating systems is highly conducive to cascade failure; these cascades have already been shown to disable critical infrastructure.
- We must take conscious steps to counter the security threat of Microsoft’s monopoly dominance of computing.

4.4.2 Criticisms of the CyberInsecurity Report

Among the criticisms leveled in response to this report were accusations of financial motive and an incorrectly placed focus. In an article entitled “OS dominance paper branded ‘marketing by fear’”³⁰, it was noted that the organization releasing the report, the Computer and Communications Industry Association (CCIA) was comprised primarily of Microsoft competitors. To counter this accusation, the CCIA noted that the paper was written by the authors “on their own” and later given to the CCIA. In regards to the content, the primary criticism seems to be that the underlying problem is not a Microsoft monopoly, but the poor quality of programmed code. Indeed, there is much to this criticism, as poorly written software is prevalent regardless of the vendor. Unless programming standards are improved across the industry, the matter of which vendor is producing the software is less important.

³⁰ <http://www.computerweekly.com/articles/article.asp?liArticleID=125179&liFlavourID=1&sp=1>

5.1 Vendor Support is Critical

In order to successfully implement a Desktop Linux strategy, the support of key Linux-capable vendors is critical. For this reason, a brief analysis of the current support options and corporate direction of a few vendors that have a stated Open Source direction has been performed. This analysis is based on information obtained from publicly available sources.

5.2 Key LDC Linux Vendors

The following key vendors were identified as being potential sources of Linux support, and would most likely be critical to the success of a Linux implementation at LDC:

5.2.1 Vendor Overview: *Novell*

With the recent acquisition of the SuSE Linux distribution, Novell has quickly become a significant player in the Open Source and Linux arenas. With a stated goal to port core Novell software to the Linux platform, including the Netware operating system, it seems likely that Novell will continue to mature as a support resource for Open Source Software and Linux. Novell is currently offering the SuSE Linux server and desktop operating systems³¹, Nterprise services for Linux (including file and print, messaging, management, etc.)³² and several others. In addition, Novell staffs teams of engineers and support staff to assist with technical design and implementation issues and can sub-contract with many integrators and partners.

Novell offers a number of products that could be of value at LDC during a desktop Linux migration:

- **SuSE Linux Desktop.** Novell now support SuSE Linux for both desktop and server applications. With a variety of bundled applications including SUN Star office, x3270 and x5250 terminal emulation software, and the Codeweaver Crossover Office suit to provide integration with Microsoft Office applications as needed. With the Codeweaver suite, a number of difficult-to-migrate applications such as Microsoft Access 2000, Visio 2000, Lotus Notes 6.5 and Internet Explorer can be run directly from a Linux desktop.
- **Ximian Desktop.** Novell also offers Ximian Desktop, version 2.0. Geared towards desktop users, Ximian also includes a number of user applications such as OpenOffice.org, and the Evolution mail client, which provides support for Exchange servers.

5.2.2 Vendor Overview: *IBM*

In the last few years, IBM has made a concerted effort to support Open Source Software and the Linux operating system. This effort has occurred on numerous fronts, including marketing, retooling support systems, and new product offerings. By IBM's own reports, over one billion dollars was spent on its Linux initiatives in the first year of its concerted effort³³. IBM currently hosts a Linux portal describing its offerings, and has recently announced³⁴ that it will adopt Linux software internally, in an effort to "eat its own dog food." IBM offers a variety of support services that could be of value to LDC in its Linux initiatives, including:

- **Linux Application Porting.** "IBM Linux Application Porting services offer you an easy path to identifying and assessing applications for porting, testing, or redeveloping critical applications."³⁵

³¹ <http://www.suse.com/us/business/products/server/sles/index.html>

³² <http://www.novell.com/products/linuxservices/>

³³ <http://www.internetnews.com/dev-news/article.php/576091>

³⁴ <http://www.theinquirer.net/?article=13485>

³⁵ <http://www-1.ibm.com/linux/solutions/igsapplicationportinglinuxsolution.shtml>

- **Linux Implementation Services.** “IBM Linux Implementation Services solution provides a range of installation and implementation services to help customers get the Linux up and running as quickly – and painlessly – as possible.”³⁶
- **Support Line for Linux.** “IBM Support Line for Linux offers support options for customers needing round-the-clock remote technical expertise. This service provides a one-stop-shop for all IBM-supported Linux distributions.”³⁷

These support options, and many others, are detailed in the IBM Linux Portal - Solutions³⁸ web site.

IBM also offers a number of Linux-ready software packages³⁹, including Lotus, WebSphere, DB2, Tivoli and many others.

5.2.3 LDC Specific Vendors

In addition to the major hardware and software vendors that LDC currently partners with, there are a number of other companies that provide goods and services to the organization. While a complete survey of each of these vendors, and their stated intentions towards Open Source software, would be impossible to identify in this modest study, it is worthwhile to do so. In particular, it is recommended that LDC identify their key vendors and encourage them, through dialog and purchasing power, to embrace Open Source software.

³⁶ <http://www-1.ibm.com/linux/solutions/igslinuximplimentationlinuxsolution.shtml>

³⁷ <http://www-1.ibm.com/linux/solutions/igssupportlinelinuxsolution.shtml>

³⁸ http://www-1.ibm.com/linux/va_4049.shtml

³⁹ <http://www-306.ibm.com/software/os/linux/software/index.jsp>

6.1 Critical Software Overview

In order to identify key software applications and potential migration stumbling blocks, survey forms were distributed to subject matter experts and departmental contacts at LDC. As a result of this data collection, and through further review and discussions with the LDC feasibility study team, the following applications were deemed “mission critical” and are therefore worthy of analysis. In particular, for each of the following applications, an individual was tasked with determining what, if any, Open Source or Linux alternative might exist. The key applications are noted in the following table:

ID#	Software Application
1	Microsoft Office
2	Siebel
3	Great Plains
4	Microsoft SQL server
5	Billysoft payroll
6	RUMBA 3270 Access
7	RUMBA Database Connector
8	Cold Fusion
9	Internet Explorer

As LDC moves forward with its Open Source initiatives, the migration of these applications (as well as all other Windows-based software) must be addressed in some way.

6.2 Non-Windows Software Options

A number of options exist for migrating Windows-based software functions to the Linux operating system. Among these options are the following:

6.2.1 Linux Native Binaries

Some packages will have equivalent Linux native binary software. In these cases, the impact to the organization to adopt the new platform will be minimal. Issues to consider with native binaries include cost differences, differences in the user interface (especially those that might require additional training of users) and differences in functionality.

6.2.2 Web Access

In some cases, it may be possible to access mission-critical applications via a web interface. If possible, this may be the easiest way to provide applications to Linux desktops. Using web access has the added advantage that it would be possible to support a mixed environment (both Windows and Linux) during the implementation process. In that way, the transition could be less disruptive to employees, and provide continuity between the two platforms. Web access must not be dependant upon operating systems specific browsers such as Internet Explorer, or extensions such as ActiveX or COM.

6.2.3 Remote Desktop

If an application is not supported, and is unlikely to be ported in the future, it is possible to run the application on a native Windows platform, and access this software through remote desktop software.

Common examples of this are the Citrix Metaframe⁴⁰ line of products, and the native Windows RDP (Remote Desktop Protocol)⁴¹. Linux based client software exists for both Citrix remote access⁴² and the RDP client⁴³. Unfortunately, remote desktop options do have some limitations, including high licensing and hardware costs on the server side, inability to run all software, poor performance and issues with printing support.

6.2.4 Windows Emulation

For some applications, it may be possible to use Linux-based Windows emulators to run software that does not have a Web or Linux binary. There are a number of emulation options available, that range widely in terms of cost and application support. On the lower end is free software such as the Windows Emulator (WINE)⁴⁴ that supports a more limited subset of Windows functionality. In the middle range are more complex commercial products such as Win4Lin⁴⁵ and Code Weavers⁴⁶ that have been developed to explicitly support “troublesome” applications such as Microsoft Office. In high range are emulators such as VMware⁴⁷ that allow you to run an entire Windows operating system, and virtually all Windows software in a separate Windows desktop. Between these various options, it should be possible to run the vast majority of conventional productivity applications on the Linux platform. However, the costs and hardware requirements of the various options should be carefully analyzed, as some emulation options require fast computers, and some may require a full Windows client OS and associated license.

6.2.5 Redevelopment and Replacement

In those exceptional cases where it is not possible to obtain an alternative such as web access, and where it is difficult to provide remote desktop software to all users of a particular software package, it may make the most business sense to redevelop or replace the application entirely. Examples of software packages that might require this approach could include applications that rely on software packages that are not portable to the Linux platform, such as internally developed applications that rely on Win32 terminal emulator “screen scraper” software to function. In these cases, it may be necessary to enter into a new System Development Life Cycle (SDLC) to determine whether or not to internally develop or purchase a replacement package. In addition to the obvious cost considerations, this option also presents additional difficulties. One difficulty is the familiarity of software developers with Open Source Software languages and development tools. In short, the current software development staff might not be able to readily adapt to Open Source programming languages, interfaces and development environments. Fortunately, a variety of development environments such as Code Forge⁴⁸ exist, and are mature. Finally, there are likely to be problems in redeveloping applications written in Microsoft programming languages such as Visual Basic, ASP and .NET. Historically, support for VB, ASP and .NET, while possible, has not had the same level of support in Linux. However, this is likely to change, particularly with the introduction of software packages such as the Novell MONO⁴⁹ software are providing better vendor-supported options.

6.2.6 No Migration

In some cases, migration to Linux – for a number of technical reasons – might place such an unwieldy burden on the LDC staff as to be impractical. For example, some user communities might require far more effort to convert than the savings that would be justified. In these instances, some user communities might remain on the Windows platform, while others are migrated to the Linux

⁴⁰ <http://www.citrix.com/>

⁴¹ <http://www.microsoft.com/windowsserver2003/evaluation/overview/technologies/terminalserver.msp>

⁴² <http://www.citrix.com/site/SS/downloads/details.asp?dID=2755&downloadID=3323&pID=186>

⁴³ <http://www.rdesktop.org/>

⁴⁴ <http://www.winehq.com/>

⁴⁵ <http://www.netraverse.com/products/index.php>

⁴⁶ <http://www.codeweavers.com/site/products/>

⁴⁷ <http://www.vmware.com/>

⁴⁸ <http://www.codeforge.com>

⁴⁹ <http://www.go-mono.com/>

operating system. This solution is not ideal, as the I.T. overhead involved in simultaneously supporting multiple platforms could be significant, but might be justified in some cases. In particular, this option probably only makes sense for smaller user communities with very specific and difficult to solve software needs.

6.3 Critical Software Data Collection

In order to identify and analyze the software applications that are critical to the ongoing success of LDC, a Critical Software Data Collection Template was created. This template is intended to collect data on specific software applications, including the following criteria:

- Name and function of the application
- The current support status of the application
- The availability of non-windows alternatives (browser access, alternative product, etc.)
- Approximate cost to redevelop the application (if developed in-house)
- Approximate level of effort to migrate to an Open Source alternative

With help from the user communities identified by the team, a number of critical software packages were analyzed to determine what Linux-based alternatives exist. Due to the large effort analyzing the LDC environment in detail, this analysis is still underway.

While additional information must still be collected, the existing model may be of use for future efforts. For this reason, it is worthwhile to discuss the data collection document, and the criteria that it contained. For an example of a blank data collection document, please refer to *Appendix 9.1.2*.

The following criteria are used in this document:

- **Vendor.** The name of the vendor who provides the software application, and a URL reference for more information.
- **Function.** What service the software package provides (office productivity, host access, financial processing, etc.)
- **Available alternatives.** What alternatives are currently available that are similar in features.
- **Recommended action.** The “best guess” recommendation, based on an analysis of available options. These recommended actions must be more carefully analyzed prior to replacement, and each is worthy of an individual study. The recommended action will generally include one of the migration options mentioned in *6.2 Non-windows Software Options*.
- **Organizational Impact.** A qualitative judgment as to how difficult it will be to implement the recommended action, in terms of cost (labor, software purchase, etc.) and change (missing features, necessary re-training, etc.) Organizational impact is determined based on a general consensus from the LDC staff, and is rated on a scale of 1-10, where 1 is very simple and 10 is exceptionally difficult.
- **Outstanding Issues.** A listing of those features that are difficult to address, or functionality that does not exist in the recommended action.

The following summary of the Microsoft Office application is included as a demonstration:

6.3.1 Example Analysis: Office XP / Outlook / Microsoft Access

Vendor: Microsoft, <http://office.microsoft.com/home/default.aspx>

Function(s): Basic office productivity tools: word Processing, spreadsheets, presentation tools, Access database, Exchange e-mail client.

Available Alternative(s): There are a number of viable alternatives to Microsoft Office, and Outlook clients. A direct Linux binary equivalent to Microsoft Access is an outstanding problem, and

may require the use of an emulator or remote desktop program, or porting of data to a relational database such as MySQL. Among the alternatives are the following:

- **Star Office.** From Sun Microsystems, <http://www.sun.com/software/star/staroffice/>, prices range from \$25 to \$50 per user. Fully vendor supported. Supports printing to PDF format directly. Allows export of presentations to Flash format directly.
- **OpenOffice.org.** From OpenOffice.org, <http://www.openoffice.org>, software is free of charge. Supported through online user communities, as well as some vendors, including Sun Microsystems. Allows printing to PDF.
- **Word Perfect Office 12.** Although details are not currently available, it is stated⁵⁰ that Word Perfect Office 12 will support Linux. It is also unknown which subsets of the Microsoft Office functionality will be supported.

Since the original authoring of this document, LDC has stated its intention to replace Microsoft Outlook with Lotus Notes. Due to this fact, the following information is offered for demonstrative and reference purposes only. In addition to supporting Microsoft Office features, it is also necessary to support the Outlook exchange client. The following Linux alternatives are considered to be the most viable Outlook replacements:

- **Novell Evolution.** From Novell, <http://www.novell.com/products/evolution/>. Currently requires the free “Evolution Connector for Microsoft Exchange” and an additional Linux or Solaris box. Functions as a “Microsoft Exchange 2000 or 2003 client, with access to scheduling, mail, public folders, and global address book features.”
- **Outlook Web Access (OWA).** Microsoft Outlook already has a web-based client to access Exchange functionality. The OWA software for Exchange 2003 also has a number of features that were previously only available in the binary Outlook client. For a complete listing of the differences between Outlook and OWA, refer to the appropriate Microsoft comparison document⁵¹. It is important to note that there are two different versions of OWA described in this document – Premium and Basic. The features of OWA premium are only available when using Internet Explorer 5.01 for Microsoft Windows⁵².
- **Generic POP3/IMAP Client.** It is also possible to access BASIC e-mail functions through a standard POP3 e-mail client. This does not allow access to a variety of important features, including scheduling, and is unlikely to be a viable option.
- **Replace Exchange.** A drastic, but perhaps viable step, would be to remove Microsoft Exchange from the environment completely, and replace it with another groupware package such as Lotus Notes or GroupWise that has better Linux or Web Access support.

Recommended Action: Initially, it is recommended that LDC consider replacing Microsoft Office with OpenOffice.org, and use an alternate method for dealing with legacy Access databases. For Exchange access, it is recommended that LDC upgrade to Outlook 2003 and use the Outlook Web Access client to access the groupware system. In the long term, removing Exchange entirely may be a viable option.

Organizational Impact: [3] There will be cost savings of approximately \$340/workstation from reduced licensing costs, and these costs may be sufficient to offset the labor involved in supporting the new software.

Outstanding Issues: The Microsoft Access database software does not have an equivalent that can read native Access files in any of the listed options. Access databases will need to be redeveloped for a relational database, or accessed through another means such as an emulator.

⁵⁰ <http://news.zdnet.co.uk/software/linuxunix/0,39020390,39150682,00.htm>

⁵¹ http://www.microsoft.com/exchange/evaluation/OutlookVowa_1.asp

⁵² http://www.microsoft.com/exchange/techinfo/outlook/OWA2k3_55.asp

6.4 Non-Critical but Fiscally Relevant Software

A number of additional applications are not necessarily “mission critical” in terms of conducting business, but have a significant impact on the ability of the I.T. to support the organization. These applications are identified by their ability to leverage increased security and efficiency within the I.T. department. The most important of these is ZEN Works.

6.4.1 ZEN Works

LDC has invested a significant amount of effort in configuring Novell ZEN works to lock down desktops and provide access to important software. Currently, a stable Linux version of ZEN Works is not available, but is reportedly in development⁵³ for ZEN Works 6.5, and a BETA version is available for download. The Linux management features of ZEN Works 6.5 (formerly the Red Carpet Enterprise⁵⁴) do not currently have the same robust feature set as the ZEN client for Windows. In particular, it does not provide remote desktop trouble-shooting or desktop “lock down” features. Due to this fact, it should be assumed that comparable functionality will not be immediately available, and that alternatives should be identified. Without a comparable Linux alternative, the workload of the I.T. department could increase, and might necessitate hiring additional staff. Fortunately, a number of options exist to replace some (if not all) of the functions currently performed by ZEN Works

- **Shell Scripting.** Due to the open nature of the Linux operating system, it should be possible to perform a wide variety of tasks through shell scripting. For example, installing software, identifying software inventories, configuring packages, checking system security and locking down desktops should all be possible through shell scripts. It may be necessary to develop these scripts from scratch.
- **VNC Remote Control.** To provide for remote desktop access for trouble-shooting purposes, Linux desktops can be configured with the VNC Server. This VNC server could be started and stopped by the user on an as-needed basis for security, or could be in an “always on” state. The VNC server software is included by default in many Linux distributions.
- **‘dd’ Desktop Imaging.** To provide for imaging of the desktop, it is possible to use the Linux ‘dd’ command to create and restore hard drive images. There are a number of other software packages, such as Ghost, that can easily install a Linux OS image to a machine. It may also be possible to use a ZEN Works boot disk to install a Linux OS on a desktop, although this may not allow for some advanced features such as scheduling of image updates.

6.4.2 HP OpenView

While there are a number of ways to manage Linux workstations, the use of a network-based enterprise management platform such as HP OpenView could provide additional uptime and performance monitoring. In addition to supporting various Linux on its various hardware platforms, HP has begun porting its enterprise management software products to the Linux platform. In a press release⁵⁵ from 2003, several Linux options were announced, including:

- **HP OpenView Network Node Manager Starter Edition 7.0 for Linux:** An entry-level product designed for smaller networks needing basic network management from a single management station. A new graphical user interface provides an easy-to-understand summary of network status and quick access to detailed event data and targeted maps.
- **HP OpenView GlancePlus for Linux:** Software that maximizes uptime of applications through real-time performance monitoring and diagnostics on Linux; enables quick problem

⁵³ <http://www.novell.com/news/press/archive/2004/03/pr04027.html>

⁵⁴ <http://www.novell.com/documentation/lg/ximian/rcserver.pdf>

⁵⁵ http://www.openview.hp.com/news/press/pr/2003/pr_0052.html

resolution; and optimizes resource utilization with details on systems, applications, processes and end-to-end response times.

- **HP OpenCall Media Platform on Linux:** A carrier-grade software-based media platform for use in a wide range of applications, including voice portals, mass alarming and alerting, and voicemail replacement. The platform is open, highly scalable and easy to manage with the ability to support next-generation services and network architectures as well as VoiceXML.

In particular, LDC would benefit from the monitoring capabilities of the OpenView GlancePlus product⁵⁶. According to the HP documentation, GlancePlus is a “a powerful system performance monitoring and diagnostic tool.” that “lets you easily examine system activities, identify and resolve performance bottlenecks, and tune your system for more efficient operation.” Among the features of the GlancePlus product are the following:

- Hierarchy of system, application and process-level information from summary to diagnostic detail
- System table resource information providing status of key configurable system parameters, such as process tables, buffer cache, buffer headers, and shared memory
- Network performance information, such as Network File System (NFS) and local area network (LAN) metrics
- Customizable rules-based diagnostics to isolate performance problems and bottlenecks
- Customizable threshold-based alarming that generates visual notifications and can also be configured to execute commands or scripts for automated actions
- Both an intuitive customizable graphical user interface and a flexible character mode interface are included, allowing you to use GlancePlus on non-X-Window displays
- Context sensitive help and online user’s guide—complete with a guided product tour and flexible search capabilities
- Dynamic system performance graphs and alarms displayed from GlancePlus while it runs as an icon
- The ability to display and alarm on Application Response Measurement (ARM) data, such as transaction average response time, distribution of transaction response times, service level objective value, and number of times it has been exceeded
- The ability to display groups of transactions and drill down to a single transaction and also display resource consumption by the transactions

6.5 Software Replacement Matrix

Although the exact requirements for any given application will require extensive analysis, a number of widely used Windows applications have viable open source or Linux alternatives. Some resources have been developed to assist in this identification of alternate software packages. For example, the “The table of equivalents / replacements / analogs of Windows software in Linux.” Web site at linuxshop.ru⁵⁷ has a fairly extensive list, and is dated 16.07.2003. As neither Mark Lachniet nor the LDC staff created or modified this list, it is presented for informational purposes only. Refer to *Appendix 9.2.8* for the complete matrix.

⁵⁶ http://www.openview.hp.com/products/glanceplus/pb/gplus_pb_feb03.pdf

⁵⁷ <http://linuxshop.ru/linuxbegin/win-lin-soft-en/table.shtml>

7.1 User Community / Conversion Costs Summary

The most critical task of evaluating any software migration, whether it be to Open Source or otherwise, is to identify the organization's user communities and the software that they use. This allows the organization's decision makers to make informed cost-benefit decisions in regards to software platform and applications. With a possible migration to Open Source software, this evaluation is even more critical, as the small applications that users take for granted may not function on a non-Windows platform. In addition, different user communities have different requirements and levels of complexity. For example, while a call center group may only require access to mainframe data, word processing, and the Internet, the billing department may use a number of specific programs that do not have open source alternatives, or have built a number of complex Excel macros over the years that do not directly translate to alternative software packages.

In an attempt to identify the information needed to identify the feasibility and costs of a possible Microsoft to Linux migration, a user community spreadsheet was created for data collection and cost analysis. The blank version of this document is attached as *Appendix 9.1.1*

7.2 User Communities

The first tab (worksheet) of the spreadsheet is intended to identify all of the discrete user communities at LDC. At the time of this report, only eight departments were identified, and full data was not collected for those eight.

The User Communities worksheet collects the following information that is relevant to future Open Source migration efforts:

- **Basic community information.** A unique identifier (such as 2.1), the name of the community, a brief description, and a subject matter expert (SME) to assist in data collection
- **User information.** For each department, the type of "average" user. This is assessed in terms of training and skill level (low, medium, high) and in terms of user type. For the purposes of future cost estimation, per hour rates for the different categories of users can be used. The following user types were identified:
 - **Data entry.** Data entry users are considered the least sophisticated, and primarily require only a small number of applications. Tasks performed by data entry staff are primarily repetitive, and do not require extensive analytical skills. Data entry users are typically the easiest to migrate to new technologies and software.
 - **Knowledge worker.** Knowledge workers are sophisticated users who apply technology in unique and sophisticated ways. Tasks performed by these workers are typically more complex, and rely on multiple software packages and processes. For example, knowledge workers might be responsible for exporting data from a database, manipulating it in Excel, and importing it into another database. Knowledge workers are typically more difficult to migrate to new technologies and software because they use sophisticated features in multiple software packages, and typically do not have advanced technological skills.
 - **Technical worker.** Technical workers are sophisticated users who are trained in systems such as databases, operating systems, and specific software packages. Tasks performed by these workers vary widely, and tend to be focused on specific I.T. areas such as database administration. Technical workers are typically moderately easy to migrate to new technologies and software. While technical workers are usually skilled at learning new technologies, they may have a large educational and emotional investment in existing technologies and be unwilling to change. Technical workers who are willing and enthusiastic make ideal "guinea pigs" for pilot programs.

7.3 User Software by Community

The second through ninth tabs (worksheets) contain more detailed information about the software used by each user community. For each application, it is important to note the costs associated with both the existing application and any proposed replacement. In order to get accurate costs for each application, a detailed study (as per *Section 6* of this document) must be performed for each critical application.

The User Software by Community spreadsheet (*See Appendix 9.1.1*), worksheet 2.1 contains the following criteria:

- **Basic software information.** The name of the application, its criticality, and its software interface should be identified.
 - **Criticality.** Rated on a scale of 1 to 10, where 10 is absolutely critical for the organization. If an application is not critical, it may be easy to replace with an alternative software package that provides the same functionality.
 - **Software Interface.** Identify the way in which the application is accessed, such as via a binary, web access, terminal session, or other. If a web interface is available for the software package, and it is not dependant upon a specific browser (such as Internet Explorer), it may be possible to migrate the client desktop to Linux without adversely affecting the application. If a binary is the only available option, it may be possible to use terminal emulation (see Section 6.2.4) or remote desktop software (see Section 6.2.3) to access the application from a Linux desktop.
- **Existing software costs.** The costs associated with the existing application, including purchase prices, ongoing support and maintenance agreements, and training should be identified. The spreadsheet specifically collects costs on a one-time (SW Cost) and recurring (SW/Yr Cost) basis. Where applications have been developed internally, the total cost of development, divided by the number of users should be used.
- **Replacement software costs.** Once a suitable replacement has been identified, the costs of the replacement software should be quantified. Both one-time costs and recurring costs should be considered.
- **Conversion costs.** Very few software packages can be adopted without a significant conversion effort. The one-time cost associated with a conversion must be taken into consideration. Specific items to consider include:
 - **Conversion costs.** At the top of the spreadsheet, a small table is used to estimate the costs of document and macro conversion. A starting assumption of \$50/hr for an engineer's time was used. A time estimate of 30 minutes per document conversion, and 60 minutes per macro conversion was given. These costs should be adjusted based on real-world experience, as appropriate.
 - **Number of documents to convert.** If it is necessary to manually convert documents, this cost should be accounted for. In some cases, up-front data conversion may not be necessary. For example, Open Office can effectively open simple Microsoft Office documents on the fly.
 - **Number of macros to convert.** In some cases, advanced users have created extensive macros within their software packages. It is unlikely that these macros can be easily converted to a new program, and they will probably have to be re-created by hand.
 - **Lost productivity.** While difficult to estimate, it may be possible to quantify a cost associated with lost productivity. In many other feasibility studies, this cost was explicitly left out of scope, as it was presumed to be overly difficult to identify, and that employees would absorb the time difference in their day-to-day work. However, for some tasks (such as data entry) it may be possible to identify a productivity cost associated with a new software package. For example, if entering

a premium could previously be done on one data entry screen in 3 minutes, and now requires 5 minutes on 2 data entry screens, this difference in time could be identified and associated with a cost.

7.4 Training Options

In order to successfully migrate to an Open Source platform, it will be necessary to employ a workforce that is capable of both using open source software and supporting it. This will require training for at least three distinct groups within LDC:

- **End users.** Regardless of the platform in use, end users will need to be trained on new software packages as they are implemented. As most end users tend to use Windows at home they are likely to require training in the Linux operating system as well.
- **Application developers.** LDC currently employs an in-house application development staff. Many of these developers do not currently code for portable application development languages such as JAVA or C++. In order to be able to develop and support on platforms other than Windows, it will be necessary to provide training for application developers.
- **I.T. support staff.** LDC will also need to train its internal support staff on open source technologies. Fortunately, a variety of training options exist for I.T. staff. There are several well-recognized certification programs available, and training in these certification tracks can be obtained from multiple sources. Among the most popular training and certification programs are the following:
 - **CompTia Linux+.** CompTia offers the Linux+ certification⁵⁸. While generally regarded as a lower-end certification for engineers with six months of experience, it may be a good benchmark for a minimum standard skill level for engineers.
 - **The Linux Professional Institute.** The LPI⁵⁹ certification program is the premier vendor-agnostic Linux certification program available. This certification program has been officially adopted by IBM, and is the building block of the new Novell Certified Linux Engineer certification⁶⁰ program. For those interested in self-study, IBM has made its study materials available for free⁶¹. Training manuals can be purchased from a number of vendors, and are easily found in any bookstore. The cost of LPI tests are typically low, and are available at most prometric testing centers. The LPI track currently offers two levels, the LPIC-1 and LPIC-2, and is planning on offering more advance level 3 certifications in topic areas such as Linux security.
 - **Novell Corporation.** The Certified Linux Engineer (CLE) program builds on the LPI certification, and provides specific training on Novell products that run on top of the Linux platform, including eGuide, iFolder, NMAS, DirXML, ZENworks for Servers and eDirectory⁶². Self study kits and classroom training is available from Novell and its partners.
 - **Red Hat Corporation.** Red Hat offers their distribution-specific Red Hat Certified Engineer (RHCE)⁶³. While geared specifically towards the Red Hat Enterprise Server line of products, this certification is well regarded. In particular, the test includes a hands-on lab component as well as a written component. Due to the focus on a specific platform, the RHCE certification may also be a better option for engineers who need hands-on training on a specific operating system.

⁵⁸ <http://www.comptia.com/certification/linux/default.asp>

⁵⁹ <http://www.lpi.org>

⁶⁰ <http://www.novell.com/training/certinfo/cle/index.html>

⁶¹ <http://www-106.ibm.com/developerworks/edu/l-dw-linux-lpir21-i.html>

⁶² <http://www.novell.com/training/certinfo/clefaqfinal.pdf>

⁶³ <http://www.redhat.com/training/rhce/courses/>

7.5 Security Management Costs

Every software package, whether from Microsoft or not, has a security overhead associated with it. As discussed in *Section 3.3.1* there are potential security management costs to be realized by migrating away from the Microsoft platform. Among these savings are decreased labor spent on patching systems, and virus and incident management. Although it may be possible to quantify these costs, it is difficult to do so. Perhaps more importantly, if Linux and Open Source software becomes more popular in the next few years, it is reasonable to assume that more and more malicious users and software will begin to target the platform. Thus, while there may be a modest savings in security management costs in the short term, they were not deemed significant to the overall cost of migration at LDC.

8.1 Implementation Overview

Although a formal implementation plan should be developed and adopted internally, a number of implementation steps were discussed by Mark Lachniet and the LDC team members. As a result of this discussion, and analysis of similar initiatives, the following implementation steps are recommended. These steps can be broken into three functional areas – administrative, application development, and Information Technology.

8.1.1 Administrative Tasks

Without support “from the top”, a drastic action such as implementing Open Source Software is doomed to failure. In order to put into place the administrative and procedural structures necessary to support a move towards Open Source Software, the following tasks are recommended:

- **Develop a high level policy of support.** Unless upper management has gone “on the record” as supporting and mandating the effort to implement Open Source Software, there will be very little documented requirement for employees to do so. It should be taken for granted that many employees, including managers and key decision makers, will be vehemently opposed to the idea of Open Source Software. Thus, it is likely that there will be a natural resistance, either overt or covert, to thwart the process. By formally declaring the intention of the company to move towards Open Source Software, these individuals will have little justification for opposing the process. It is recommended that a policy of support be drafted immediately.
- **Implement purchasing requirements.** One of the most effective ways to achieve a technological agenda is to put strict controls on the purchasing process. Specifically, LDC should include purchasing provisions that require all hardware and software purchases to take into consider Open Source Software. In the case of hardware, it would be advisable to require hardware that is compatible with common Linux distributions, and that comes from a company with a stated goal of supporting Open Source Software such as IBM or HP. In the case of software, any Request For Proposals (RFPs) should be formally required to solicit and consider bids from companies that embrace Open Source Software. While not all purchases can (or should) favor Open Source Software, any decision that does not consider, or ultimately select such a product should be justified in writing. Lastly, an administrator with authority should review purchasing requests before they are approved, and verify that all requests follow these guidelines.
- **Implement hiring requirements.** Just as with the purchasing of software, the hiring of new employees should consider, if not favor, skills and experience with Open Source Software. This will not only help to establish organizational momentum, but will reduce the cost and burden of training and retraining employees. As with software purchasing, a written justification for the selections made should be documented, and a manager with authority should approve all employee hires.
- **Develop a Linux “tiger team”.** LDC has a wealth of technological and business experience in its existing staff. It is highly recommended that a formal task force or “tiger team” be developed to continue the analysis and implementation of Open Source Software. This task force should not only include advocates for Open Source Software, but also those who have the greatest objections to it. While individual employees should not have the power to derail the implementation and business goals established by management, their perspectives on the risks and problems of implementing Open Source Software will be essential in ensuring the success of the process.
- **Hire an Open Source expert / project manager.** During discussions with the LDC staff, it was identified that there is not currently an individual identified as a Linux technical expert. A person with a background in UNIX, Linux and Open Source software should

either be hired from outside, or promoted from within. The ideal candidate for this position will also have skills in project management and business processes, and should be able to lead the tiger team and manage the many tasks that will arise.

- **Establish a budget.** Finding resources for project managers, analysts, evaluation software, training and other costs will be necessary. Rather than place an additional burden on existing departmental budgets, it may be worthwhile to create a specific budget line item for this purpose. This will improve the ability to account for the real costs of the implementation effort and minimize political friction.

8.1.2 Application Development Tasks

LDC has a large investment in internally-developed applications. According to information gathered during this study, millions of dollars have been put into the development of key LDC applications. Indeed, it is the internally developed applications that represent the greatest difficulty for migrating to Open Source Software. In order to prepare the in-house development teams for this possibility, it is recommended that LDC begin taking steps to ensure the ability to migrate to Open Source Software in the future. Among these are the following recommendations:

- **Identify a portable development platform for future development.** In order to promote the ability to migrate to other platforms, Open Source or otherwise, it would be helpful to identify and standardize on a development platform and language that is portable to multiple architectures. The current reliance on Win32 application development is particularly dangerous, because it is difficult to port to Linux and other platforms. An alternative development platform such as JAVA would ease the transition between platforms, and ultimately promote the best interests of LDC. In addition, LDC should minimize the use of Windows-specific API calls, regardless of the development platform.
- **Cease and desist all non-portable development.** If LDC truly wishes to migrate to Open Source Software, every hour of code spent developing new features and applications on non-portable development platforms will equate to wasted money. While it is recognized that bugfixes, minor updates and other changes will be needed, it would be ideal if all future development was mandated to occur on a portable development platform. A mandate such as this can only come from upper management.
- **Obtain training for application developers.** The current LDC application development staff is primarily trained on Windows technology and development tools. In order to be successful, application developers will need to be retrained on whatever portable development platform is necessary.
- **Identify a migration strategy for internally developed applications.** As previously noted, LDC has a significant investment in internally developed applications. Each of these applications will need to be carefully analyzed, and have a migration strategy identified for it. Additional data collection work remains to be done in this area. In some cases, a complete re-write of applications, particularly those that rely on Win32 API calls, will be necessary.

8.1.3 Information Technology Tasks

The Information Technology staff at LDC will have a significant amount of work to do in order to migrate the organization to Open Source and Linux software. Just as with the application development staff, I.T. workers will require training, resources and support from upper management in order to be successful. Among the recommendations for aligning the I.T. staff with the implementation process are the following:

- **Identify and migrate back-end applications.** A number of back-end applications, such as e-mail, file and print, Internet servers and application servers will need to be migrated to an Open Source platform. A process to identify these and identify a migration strategy should be developed as soon as possible.

- **Obtain training for I.T. staff.** The current LDC I.T. staff is primarily trained on Windows technology such as Windows XP and Novell ZEN works. In order to be successful, the I.T. staff will need to be trained on the Linux operating system and associated tools.
- **Pilot Open Source Software with “willing victim” user communities.** One of the least intrusive ways to test the waters for future Open Source Software implementations is to perform small pilot tests with willing participants. If possible, LDC should identify one or more user communities that would not resist adopting different software. The lessons learned from this experience should be invaluable to the process, and minimize unforeseen problems in the future. Resources for these pilot tests, to purchase software, obtain training, and pay for head counts, should be established up front, if possible.

9.1 Data Collection Templates

9.1.1 User Communities Data Collection Worksheet

This document is intended to be used to collect data about the various user communities at LDC, including the most significant software packages.

9.1.2 Critical Applications Data Collection Form

This document is intended to collect data about specific applications in use at LDC, including their platform, estimated cost to replace, and other criteria.

9.2 Collected Data

9.2.8 Software Replacement Matrix

This spreadsheet, taken in whole from the linuxshop.ru web site and dated 7/16/2003 lists a number of commonly used Windows applications, and some possible replacements for them. This document was not modified by the LDC Linux feasibility team or Mark Lachniet.

9.3 Papers referenced

The following studies were referenced, and are included in printed format for your perusal

9.3.1 Dutch (Statskontoret) study

This is the study regarding the Danish government's analysis of Open Source Software, as discussed in *Section 4.2*.

9.3.2 South African OSS Strategy v3

This is the study regarding South Africa's analysis of Open Source Software, as discussed in *Section 4.3*.

9.3.3 CyberInsecurity Report

This is the study regarding the risks of a software monoculture, as discussed in *Section 4.5*.