

Dieser Artikel ist Teil des
Open Source Jahrbuchs 2006

Bernd Lutterbeck
Matthias Bärwolff
Robert A. Gehring (Hrsg.)

Open Source
Jahrbuch 2006

Zwischen Softwareentwicklung und Gesellschaftmodell

erhältlich unter www.opensourcejahrbuch.de.

Die komplette Ausgabe enthält viele weitere interessante Artikel. Lob und Kritik zu diesem Artikel sowie weitere Anregungen können Sie uns einfach und unkompliziert mitteilen per E-Mail oder auf www.opensourcejahrbuch.de/feedback/.

Einblicke in das Microsoft- Linux-/Open-Source-Software-Lab

BILL HILF*



(CC-Lizenz siehe Seite 499)

Inmitten einer der wohl größten Umgebungen von Microsoft-Produkten haben wir ein Open-Source-Software-Labor aufgebaut mit dem Ziel, Open Source Software besser zu verstehen und die Interoperabilität zwischen unserer Software und der der Open-Source-Community zu verbessern. Hierfür haben wir eine Reihe von Fachleuten und Entwicklern aus dem Open-Source-Bereich gewinnen können und eine umfangreiche Linux- und Unix-Umgebung geschaffen. Microsoft versucht, mit Hilfe dieses Labors das Wesen von Open Source sowie die Strukturen und Prozesse dahinter zu verstehen.

Schlüsselwörter: Microsoft · Wettbewerb · Kooperation · Interoperabilität
· Linux-/Open-Source-Software-Labor

1 Einleitung

Microsofts Linux-/Open-Source-Software-Labor als ambitionierte Forschungseinrichtung zu bezeichnen, ist schlicht eine Untertreibung. Denn in dem Labor befinden sich über 300 Server der unterschiedlichsten Art und Größe, auf denen insgesamt über 15 Unix-Versionen und mehr als 48 unterschiedliche Linux-Distributionen laufen. Das Forschungsteam besteht aus führenden Linux- und Unix-Programmierern sowie Systemadministratoren. Darunter sind auch einige Entwickler von populären Linux-Distributionen sowie Autoren renommierter Bücher über Unix. Insgesamt ist das Microsoft-Labor weltweit eines der wenigen Labors, die über diese umfassende Ausstattung, Komplexität und das hohe fachliche Niveau verfügen – alles im Namen der Open-Source-Forschung.

* Bill Hilf leitet als *Platform Technology Strategy Director* die Technologieabteilung *Linux and Open Source* bei Microsoft und ist dort seit Dez. 2005 auch für die Shared-Source-Initiative verantwortlich.

Immer, wenn das Thema auf das Linux-/Open-Source-Software-Labor und meine Tätigkeit als Direktor kommt, werde ich gefragt: „Bei Microsoft? Warum beschäftigen Sie sich in einer gemischten Umgebung bei Microsoft mit Linux?“ Theorien darüber gibt es im Überfluss – von „Microsoft arbeitet an einer eigenen Linux-Implementierung“ bis zu „Microsoft erwägt die Portierung auf Linux“. Die Wahrheit ist, dass das Linux-/Open-Source-Software-Labor dem Unternehmen einen tiefen Einblick in die Welt der Open-Source-Software (OSS) ermöglicht und darauf abzielt, dass Microsoft-Produkte besser mit Linux und anderen OSS-Anwendungen zusammenarbeiten.

Entgegen der gängigen Annahme ist Microsoft kein Gegner von OSS. Mit Sicherheit leben die meisten Kunden nicht in einer Entweder-oder-Welt. Vielmehr entscheiden sie sich für die Technologie – für ein Betriebssystem oder eine Anwendung –, die ein bestimmtes Problem löst oder eine spezielle Geschäftsanforderung erfüllt und entscheiden nicht auf Basis eines Entwicklungsmodells.

In unserem Labor laufen sowohl Linux als auch andere OSS in einer Microsoft-Umgebung. Dies ermöglicht uns, zu erforschen, wie diese Technologien besser mit Microsoft-Technologien zusammenarbeiten können, damit für Microsoft-Kunden eine größere Auswahl an interoperabler Software besteht.

Dabei untersuchen wir z. B. wie die Funktionalität der Management-Tools von Microsoft in heterogenen Umgebungen optimiert werden kann. Verwendet ein Kunde einen *Microsoft Systems Management Server (SMS)* oder einen *Microsoft Operations Manager (MOM)* und möchte einen Linux- oder Unix-Server einsetzen, können wir über Erkenntnisse aus den Tests informieren, die wir über verschiedene Technologien von Drittanbietern durchgeführt haben und die ein bestimmtes Szenario ermöglichen.

Ein weiteres Beispiel für unsere Forschungsaktivitäten besteht in der Arbeit an der Version „Release Candidate 2“ (R2) des Microsoft Windows Server 2003. Die neue Version enthält verschiedene Technologien, die als Subsystem für Unix-basierte Anwendungen bezeichnet werden können und Services für die Interoperabilität mit Unix- und Linux-Systemen zur Verfügung stellen. Zu diesen Technologien gehören Unix-Network-Services wie *Network File Sharing (NFS)* und *Network Information Service (NIS)*. Wir haben umfangreiche Tests durchgeführt, um festzustellen, wie gut Windows Server 2003 R2 tatsächlich mit anderen Unix- und Linux-Systemen interagiert. So haben wir beispielsweise unterschiedliche Open-Source-Anwendungen, *NFS* und *NIS* in diesem Subsystem eingesetzt, um die Interoperabilität der Anwendungen und Services mit anderen Elementen in der Data-Center-Umgebung zu testen.

2 Die Kunst der „Coopetition“¹

Das Testen der Interoperabilität von OSS mit Microsoft-Produkten ist eine der Hauptaktivitäten des Labors – aber nicht die einzige. Ein wichtiges Ziel ist unsere Wettbe-

1 „Coopetition“ setzt sich zusammen aus den Wörtern *cooperation* und *competition* und steht für „konkurrierende Zusammenarbeit“.

werbsfähigkeit – die Microsoft-Produkte sollen mit Hilfe tief greifender Erkenntnisse über Linux und Open Source verbessert werden. Wir analysieren, testen und prüfen die Tauglichkeit von OSS mit Microsoft-Produkten und vergleichen unterschiedliche Server-Workloads, Desktop-Szenarien, Virtualisierungs- und Sicherheitstechnologien, Management-Tools oder einfach spezifische Anwendungen, die in vertikalen Industrien eingesetzt werden. Die Daten aus diesen Analysen geben wir an die Produktteams weiter, die unsere Erkenntnisse bei der Planung und Entwicklung berücksichtigen.

Ein aktuelles Beispiel für unsere Forschungsaktivitäten ist das Testen einer „Beta 2“-Version des *Microsoft Windows Compute Cluster Server 2003*, dessen Verfügbarkeit Microsoft im vergangenen Monat ankündigte und mit dem Microsoft in den High-Performance-Computing-Markt (HPC-Markt) einsteigt – einem Marktsegment, das größtenteils von Linux bedient wird. Vor der Entwicklung des Produkts informierte sich das Produktteam zunächst beim Linux-/Open-Source-Software-Labor über die bestmögliche HPC-Lösung aus dem Blickwinkel von Open Source.

Da unsere Labormitarbeiter über Expertenkenntnisse im HPC-Bereich verfügen, konnten wir ein umfassendes, geclustertes System entwickeln und die Anwendungen intensiv testen. Anschließend wurden Vergleichstests mit Linux durchgeführt, die Installation abgebaut und dieselben Tests auf einem *Windows Compute Cluster Server (CCS)* unter Einsatz identischer Hardware und Netzwerkeinrichtungen erneut durchgeführt. Die Ergebnisse aus unseren Tests wurden dem Produktteam weitergeleitet. Damit stehen dem Team Erkenntnisse über die Stärken und Schwächen der verschiedenen Linux-HPC-Lösungen zur Verfügung, die dazu beitragen, dass *Windows CCS* bis zur Auslieferung im nächsten Jahr optimiert wird.

Microsoft beherrscht die Kunst der „Coopetition“, der „konkurrierenden Zusammenarbeit“: Wettbewerb mit Open Source auf der einen und Erhöhung der Interoperabilität auf der anderen Seite. Da in den nächsten Jahren sowohl Microsoft als auch OSS-Technologien existieren werden, ist es wichtig, dass Microsoft an der Umsetzung beider Ziele arbeitet. Entsprechend spiegeln sich die zentralen Ziele des Linux-/Open-Source-Software-Labors wider.

3 Ein Glasfaserkabel und ein Loch in der Wand

Natürlich gibt es unterschiedliche Wege, um Wissen über OSS zu sammeln. Beispielsweise wäre es einfacher, auf die Interpretation der Daten anderer Organisationen zuzugreifen. In vielerlei Hinsicht wäre dieser Ansatz aber mit dem Versuch gleichzusetzen, ein fremdes Land zu verstehen, ohne jemals dort gewesen zu sein. Der Erwerb einer Sprach-CD oder eines Reiseführers mag einem Touristen den Eindruck vermitteln, er sei Teil der Kultur des Landes, das er bereist. In Wahrheit bleibt man solange Tourist, bis man tatsächlich einige Zeit in dem Land gelebt hat.

Diese Philosophie verfolgen wir auch bei unserem Open-Source-Labor. Anstatt OSS „von außen“ zu bewerten, ist es unser Ziel, uns in diese Welt zu vertiefen und

Experten zu beschäftigen, die auf Fakten basierte, vorurteilslose und wissenschaftliche Informationen sammeln. Aufgabe des Labors ist, wissenschaftliche Erkenntnisse zusammenzutragen, welche die Aussagen belegen, die über Linux gemacht werden. Auf diese Weise muss Microsoft nicht über Linux philosophieren oder Mutmaßungen über OSS anstellen. Da das Labor Kompetenzzentrum für das Thema Open Source ist, werden den Produktteams aussagekräftige Daten und Schlussfolgerungen unserer Open-Source-Experten zur Verfügung gestellt, wenn diese z. B. nach dem Status bei den Management-Tools für Linux oder beim Linux-Desktop fragen.

Als ich 2003 zu Microsoft kam, um das Linux-/Open-Source-Software-Labor aufzubauen und ein Team aus Linux- und Unix-Forschern zu leiten, damit das Unternehmen Kenntnisse über OSS gewinnen kann, war mir nicht ganz klar, dass man mich buchstäblich mit dem „Aufbau“ des Labors beauftragt hatte.

Da die IT-Abteilung bei Microsoft praktisch nur Microsoft-Software nutzt, wurde mir die Aufgabe übertragen, ein Labor mit einer reinen Linux-/OSS-Umgebung aufzubauen. Das bedeutete: Mir stand ein Raum zur Verfügung. Die ersten Tage verbrachte ich in einem ziemlich leeren Raum und wartete, bis die IT-Leute über mir ein Netzkabel durch ein frisch in die Deckenwand gebohrtes Loch gezogen hatten. Da stand ich nun und starrte auf das Stück Glasfaserkabel in meiner Hand, als die IT-Leute herunterkamen und sagten: „Das ist alles, was wir für dich tun können. Ab hier stehst du auf eigenen Füßen.“ Abgesehen von den Wänden, der Decke und einem Stückchen Kabel mussten wir das Labor von Grund auf aufbauen.

Die erste Phase bestand natürlich darin, Mitarbeiter einzustellen – das wichtigste Kapital in unserem Labor. Heute beschäftigen wir Vollzeitmitarbeiter und Subunternehmer, die alle Entwicklungsexperten oder Systemadministratoren im Bereich Open Source sind. Einige Teammitglieder, wie z. B. Daniel Robbins, Gründer von *Gentoo Linux*, der Anfang 2005 zu Microsoft kam, sind führende Architekten oder Entwickler von Linux-Distributionen. Andere verfügen über weitreichende Unix-Kenntnisse und sind Autoren populärer Bücher über Unix oder Tools. Einige sind Sicherheitsexperten für Linux/OSS, *Embedded Developers*, Experten für Virtualisierung und Clustering oder Entwickler mit Erfahrung in den Bereichen *GTK+*, *GNOME/KDE* und Lokalisierung. Dazu kommen Mitarbeiter, die sich mit dem Einsatz von Microsoft-Produkten in großen Data-Center-Umgebungen wie MSN auskennen. Die Breite unserer Expertise im Team ist sehr eindrucksvoll; fast alle Teammitglieder haben Erfahrungen mit komplexen und stark gemischten IT-Umgebungen.

In der zweiten Phase musste das Labor eingerichtet werden. Wir schafften es, das Labor in knapp zwei Jahren mit einer enormen Anzahl unterschiedlicher Technologien und verschiedensten Hardware- und Software-Elementen sowie Anwendungen auszustatten. Mit über 300 Servern von Anbietern wie Dell, Hewlett-Packard, IBM, Microtel, Penquin, Pogo und Sun sowie über 20 Unix- und 48 Linux-Versionen, zu denen auch weniger bekannte Distributionen wie *Asianux*, *CentOS* und *NetBSD* gehören, ist das Linux-/Open-Source-Labor eines der am besten ausgestatteten For-

Betriebssystem	Version/Distribution
Windows	Windows 2000 Server, Windows Server 2003 Enterprise, Windows Vista, Windows XP
Linux	Arch Linux, Ark Linux, Asianux, Crux Linux, Debian, Fedora Core, Foresight Linux, Freedows, Linux From Scratch, Gentoo, Libranet, Mandrake Linux, Mandriva, MEPIS, Novell Open Enterprise Server, Red Hat Enterprise Linux, Red Hat Linux, Rocks, Slackware, SuSE Linux Enterprise Server, SuSE Linux Standard Server, SuSE Pro, Tinsyofa, TurboLinux, Vector Linux, Vida Linux, Ubuntu
Unix	AIX5L, FreeBSD, OpenBSD, NetBSD, Solaris, Java Desktop System
Andere	MacOS

Tabelle 1: Installierte Betriebssysteme im Open-Source-Software-Lab

schungslabors, die sich ganz auf Open Source konzentrieren.

Da Dutzende unterschiedliche Versionen verschiedenster Produkte permanent im Einsatz sind, können wir die Interoperabilität von Open Source in einer Vielzahl von Szenarien testen. Da wir unser eigenes Netzwerk und unsere eigenen Sicherheits-Services betreiben sowie unser *Patching* und *Updating* selbst durchführen, reproduziert unser Labor die tatsächlichen IT-Umgebungen bei unseren Kunden. Ich sage unseren Produktteams oft: „Wenn es euer Produkt durch dieses Labor schafft, dann wird es 90 Prozent der Linux-/Unix-/OSS-basierten Kundenumgebungen überleben.“

4 Eine Open-Source-Blase in einem Microsoft-Meer

Interessant und unerwartet ist die Dynamik, die sich daraus ergibt, dass das Labor mit seiner umfassenden Linux- und Unix-Umgebung eingebettet ist in die weltweit größte, komplett aus Microsoft-Produkten bestehende Umgebung, mit der alle Microsoft-Mitarbeiter unterstützt werden. Dazu gehören Windows-basierte Security-Services, Internet-Proxies, Mail-Services, Human-Resource-Services und alle anderen Systeme, die im Unternehmen verwendet werden.

Wir mussten also Wege finden, um die Interoperabilität dieser unheimlich komplexen Umgebung nicht nur im Labor zu sichern, und uns dabei nicht auf die Microsoft-Umgebung wie *Exchange* oder *Active Directory* beschränken, sondern auch die Interoperabilität zwischen dem Labor und Microsoft gewährleisten.

Viele Kunden, die ihre Systeme in gemischten Umgebungen einsetzen, haben uns gefragt, wie wir Linux, Unices und andere OSS in einer Microsoft-zentrischen IT-

Hersteller	Hardware
Compaq	Proliant DL580, Proliant BL10e, nx5000
Dell	PowerEdge 2450, PowerEdge 4350, PowerEdge 1855, PowerEdge 1500SC, Optiplex GX280, Optiplex GX270, PowerEdge 1855, PowerVault 745N
HP	Proliant DL380, Proliant DL585
HP Compaq	nx5000, D530
IBM	xSeries x342, xSeries x340, xSeries x330, xSeries x350, pSeries 630
Microtel	Computer System SYSMAR715
Neoware	CA5
Pogo	PW 1464, PW 1180, Vorticon64
Sun	SunFire V240, SunFire V20Z, SunFire 280R
Toshiba	Tecra M2, Protégé

Tabelle 2: Verwendete Hardware im Open-Source-Software-Lab (ohne Spezialanfertigungen)

Umgebung verwenden: „Wie schaffen wir es, dass diese Systeme zusammenarbeiten? Wie sollte der Einsatz der Software erfolgen? Welche Tools verwendet Microsoft?“

Die vielen unterschiedlichen Management-Tools, die wir verwenden, entsprechen der Anzahl an Servern, Betriebssystemen und anderen Anwendungen, die in unserem Labor eingesetzt werden. Für das Software-Management und die Verteilung verwenden wir Tools wie *Microsoft Systems Management Server* mit *Vintela Management Extensions (VMX)*, *Kickstart*, *Red Carpet*, *Portage* und *Red Hat Network*. Um die Infrastruktur des Labors fernzuverwalten, nutzen wir *SSH*, *VNC*, *X-Windows Tunneling* und *Windows Terminal Services*. Auch hier ist sehr unwahrscheinlich, dass ein Kunde alle Tools verwendet. Wir versuchen aber, so viele verschiedene Szenarien wie möglich zu reproduzieren, die die unterschiedlichsten Kunden verwenden könnten. Auf diese Weise erhalten wir tief greifende Erkenntnisse zur Lösung von Problemen.

Durch diese unterschiedlichen Linux-orientierten Workloads, Server, Desktops, Laptops, Software- und Gerätekonfigurationen innerhalb einer komplexen Microsoft-Umgebung können wir täglich mit der Interoperabilität experimentieren und sie testen. Im Verlauf dieses Prozesses ergeben sich interessante Erkenntnisse: simple Problemlösungen, zum Beispiel, wie man mit Linux in einer Windows-Umgebung ins Internet gelangt, aber auch komplexere Fragestellungen, beispielsweise, wie man die Authentifizierung gegen *Active Directory* von Linux-Clients durchführt oder wie man OSS-Mail-Clients mit Microsoft Exchange Server verwendet.

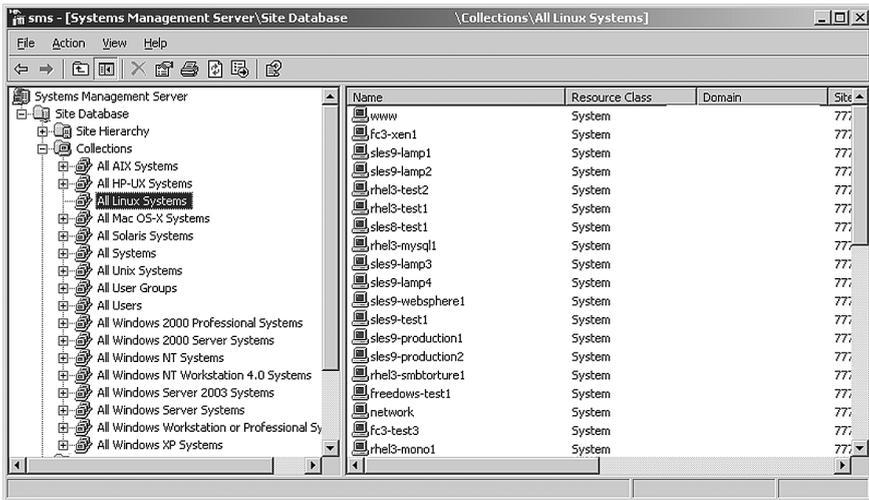


Abbildung 1: SMS-Interface zur Verwaltung der Linux-Server im Microsoft-Open-Source-Labor

5 Aufbau und Testen der Interoperabilität im Labor

Äußerst interessant beim Testen der Interoperabilität der von uns verwendeten Management-Tools war ein Szenario, in dem der *Microsoft Systems Management Server* erweitert wurde. Ziel war die Verwaltung von Unix-, Linux- und sogar Apple-Systemen durch den SMS, der so aufgebaut wurde, dass ein offenes Protokoll mit der Bezeichnung *OpenWBEM* genutzt werden konnte, um mit anderen Software-Elementen zu kommunizieren, zum Beispiel mit *VMX*, die auf Nicht-Microsoft-Systemen laufen. Durch die Erweiterung des SMS unter Verwendung von *VMX* waren wir im Labor in der Lage, das *SMS-Framework* für die Verwaltung sämtlicher Server und Clients einzusetzen. Dieses Setup stellt denjenigen im Labor, die daran gewöhnt sind mit Windows zu arbeiten, ein bekanntes Tool zur Verwaltung von Nicht-Windows-Systemen zur Verfügung. Gleichzeitig stehen unseren Linux- und Unix-Forschern die Management-Tools zur Verfügung, mit denen sie vertraut sind: *SSH-Clients*, *X-Windows* und *Red Hat Config Tools*. Abbildung 1 zeigt das *SMS-Interface* zur Verwaltung unserer Linux-Server im Labor.

Darüber hinaus haben wir wichtige Informationen über den Einsatz von kommerzieller Software von Drittanbietern zur Erweiterung von Microsoft-Produkten erhalten, vor allem bei der Verwendung einer *Centrify-DirectControl-Lösung* zur Integration unserer Unix- und Linux-Plattformen mit den Identitäts-, Zugangs- und Policy-Management-Services von *Microsoft Active Directory*.

Vor dem Einsatz von *DirectControl* musste sich jeder Mitarbeiter mittels Benutzer-

namen und Passwort auf jedem Server anmelden, um sich auf einem unserer Server einzuloggen – lästig, wenn mehr als 300 Server und viele unterschiedliche Betriebssysteme installiert sind. Durch das Einrichten einer *Active-Directory-Domäne* wurden für jeden Mitarbeiter ein Domänen-Benutzername und ein Passwort erstellt, mit denen wir auf alle Server zugreifen können, nachdem wir uns durch einmaliges Anmelden in die Domäne eingeloggt haben. Damit steht uns eine leistungsstarke Authentifizierungslösung zur Verfügung. Vielen ist nicht bewusst, dass ein *Active-Directory-Server* die Authentifizierung von Linux-Rechnern durchführen kann – ich habe vor CIOs, die diese Lösungsmöglichkeit nicht kannten, entsprechende Vorträge gehalten.

Das Linux-/Open-Source-Labor spielte auch eine Rolle bei der Unterstützung des Test-Supports von Microsoft für Linux in *Microsoft Virtual Server 2005 SP1*, der die Betriebssysteme Linux und Sun Solaris auf Windows-Servern virtualisieren kann. Wir haben diesen Support intensiv unter Einsatz von *Virtual Server 2005* auf einem Single-Machine-Modell getestet und alle 48 Linux-Distributionen als Gastbetriebssystem verwendet. Mit diesem Setup konnten wir unterschiedliche Linux-Distributionen und alle Server, die auf einem Microsoft-Betriebssystem laufen, testen – ohne separate Server für jede Linux-Maschine einzusetzen.

6 Unix- und Linux-Experten sprechen mit Windows eine bekannte Sprache

Die Interoperabilität von Produkten ist ein wichtiger Aspekt. Eine größere Problematik entsteht allerdings dadurch, dass viele Unix- und Linux-Experten mit Windows nicht vertraut sind.

Als langjähriger Unix-Experte besteht für mich eine der größten Schwierigkeiten in einem *Windows Data Center* darin, dass keine Verbindung zwischen meiner Sprache, also meinen Kenntnissen und meinem Wissen über Unix, und der Sprache der Windows-Plattform vorhanden ist. Diese Inkongruenz lässt sich auf unterschiedliche Weise beheben, zum Beispiel durch Verwendung des Subsystems für Unix-Applikationen in R2, durch Einsatz eines Fremdproduktes wie *CygWin* oder sogar mit einem Virtualisierungsprodukt. Aber viele dieser Migrations-Tools dienen lediglich dazu, eine Anwendung von einer Plattform auf eine andere zu übertragen. Was ich eigentlich benötigte, war eine integrierte Technologie, die Teil von Windows war und die mir, dem langjährigen Unix-Experten, dabei helfen würde, auf einem Windows-Server eine vertraute Sprache zu sprechen.

Mit der Entwicklung einer solchen Technologie beschäftigt sich derzeit unser Windows-Server-Management-Produktteam. Unter der Bezeichnung *Monad* wird eine Befehlszeilen- und Shell-Umgebung der nächsten Generation für Windows erarbeitet, die unter .NET läuft. Im Rahmen des Projektes besteht Microsofts Vision darin, die einfache Automatisierung für die lokale und ferne Administration zu ermöglichen, indem wir eine konsistente, schnelle, umfassende und zusammensetzbare Befehlszeilen-

und Skripting-Umgebung zur Verfügung stellen, die die gesamte Windows-Plattform umfasst. Für einen Unix-Experten ist die Möglichkeit wichtig, eigene Befehlszeilen und eine eigene Skripting-Umgebung zu erstellen, die denen aus der Unix- und Linux-Welt ähneln, um die Windows-Umgebung besser kontrollieren und die Barriere zwischen Unix und Windows deutlich reduzieren zu können.

7 Meinungen wissenschaftlich testen

Wir kennen viele unterschiedliche Meinungen unserer Kunden über Linux, Open Source und Windows. Wir analysieren diese Meinungen in unserem Labor und stützen uns dabei auf die Wissenschaft – nicht auf Ideologien oder Meinungen.

Gängig ist zum Beispiel die Meinung, dass Linux auf jeder beliebigen Hardware eingesetzt werden kann. So haben wir die Unterstützung von Legacy-Hardware für Linux untersucht, um herauszufinden, ob sich diese Ansicht bestätigen würde.

Wir untersuchten acht aktuelle Linux-Distributionen, darunter *SuSE Pro 9.2*, *Xandros* und *Fedora Core 3*, in Kombination mit Windows XP und Windows Server 2003 und versuchten, jede einzelne Komponente auf Computern zu installieren, die den durchschnittlichen Computern, die 1995, 1997, 1999 und 2001 verfügbar waren, entsprachen. Dieser Test stellte nicht die Frage, ob man das Betriebssystem verändern kann. Wir verwendeten eine Out-of-the-Box-Installation auf CD-ROM. Sollten wir in der Lage sein, die Betriebssysteme erfolgreich zu installieren, würden wir die Leistung der Systeme untersuchen können. Es stellte sich heraus, dass es nicht viele Fälle gab, in denen eine moderne Linux-Distribution auf einem PC einsetzbar war, der älter war als die Computer, auf denen auch Windows läuft. Mit diesem Fallbeispiel haben wir eine weit verbreitete Ansicht wissenschaftlich widerlegt.

Darüber hinaus untersuchen wir auch andere Dynamiken von OSS. Dazu gehört die Frage, inwieweit sich kommerzielle und nicht-kommerzielle Linux-Distributionen von ihren ursprünglichen Open-Source-Projekten unterscheiden. Abbildung 2 stellt einige Unterschiede dar – die Anzahl der Dateiänderungen zwischen dem ursprünglichen Linux-Kernel (von kernel.org) und derselben Version dieses Kernels, der dann als Linux-Distribution kommerziell ausgeliefert wurde. Auf der Basis dieser Betrachtungen können wir die Dynamik „Community versus Commercial“ besser verstehen.

Dies sind nur zwei Beispiele der vielfältigen Forschungsergebnisse des Labors, die uns dabei helfen, ein besseres Verständnis von Open Source zu entwickeln, aus dem wir wiederum für uns und unsere Kunden sinnvolle technische und strategische Zukunftsentscheidungen ableiten können.

8 Soziologische Aspekte von Open Source unter der Lupe

Zu den signifikantesten Aspekten von OSS, die wir im Labor ganz genau unter die Lupe nehmen, gehört das Phänomen des Community-Developments. In diesem

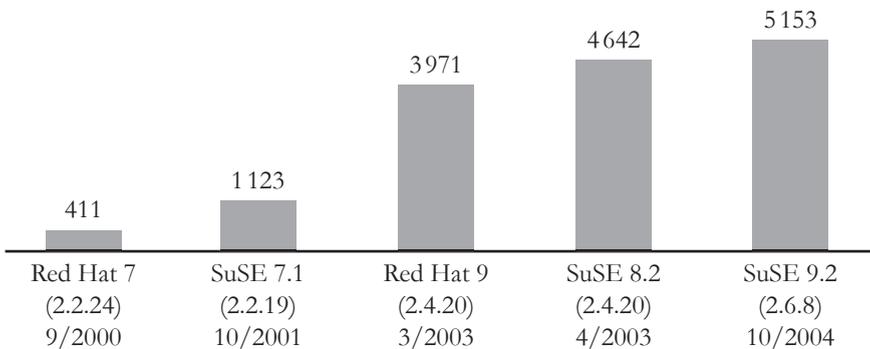


Abbildung 2: Anzahl der Dateiänderungen zwischen dem ursprünglichen Linux-Kernel (von *kernel.org*) und derselben Version dieses Kernels in der ausgelieferten Linux-Distribution

Bereich lernt Microsoft von der Open-Source-Community.

Rund 20 Prozent unserer Zeit verbringen wir damit, diesen Prozess zu verstehen. Wir unterstützen die Entwickler und Tester bei Microsoft dabei, mehr über die Open-Source-Entwicklung zu erfahren und unsere Microsoft-Produkte der Entwicklergemeinschaft zugänglicher zu machen. Als Ingenieure und Technologen sind sie von diesem Modell begeistert – sie möchten verstehen, wie das Testen in einer kollaborativen Gemeinschaft tatsächlich funktionieren kann, welche Tools dafür verwendet, wie Testfälle geschrieben, wie Programmfehler gemeldet, verfolgt und beseitigt und wie die Tester geschult werden. Wir versuchen, uns dem Community-Modell auf objektive Weise zu nähern, analysieren die positiven und negativen Aspekte und trennen Fakten von Fiktionen, so dass wir die technischen Vor- und Nachteile der Community-Entwicklung erfassen können, ohne uns dabei von philosophischen Gedanken oder künstlich erzeugtem Rummel ablenken zu lassen.

Das Community-Development-Modell hat Microsoft dabei geholfen, neue Denkansätze über eigene Entwicklungsprojekte wie *Microsoft Shared Source* zu verfolgen und darüber nachzudenken, wie der Prozess der gemeinsamen Entwicklung von Produkten aufgegriffen werden kann.² Ausbilder aus unserem Labor arbeiten mit den Produktteams zusammen, um sicherzustellen, dass sie alle Aspekte des Prozesses in ihre Überlegungen einbezogen und die wahrscheinlich auftretenden Problematiken erkannt haben. Die Tatsache, dass sich Microsoft mit der Open-Source-Community auseinandersetzt, um herauszufinden, wie man dort Software entwickelt, hilft Microsoft dabei, die Ansichten über die Zusammenarbeit mit Software-Entwicklern, die mit unterschiedlichen Entwicklungsmodellen arbeiten, zu erweitern.

Im Gegenzug ist unser Labor in den vergangenen zwei Jahren zu einer techni-

² Informationen zu *Microsoft Shared Source* gibt es unter <http://www.microsoft.com/sharedsource>.

schen Anlaufstelle für Entwickler von OSS geworden, die mit Microsoft kommunizieren möchten. Wenn Entwickler beispielsweise an der Interoperabilität mit einem Microsoft-Produkt arbeiten und Fragen auftreten, setzen sie sich verstärkt mit uns in Kontakt. In unserem Labor treffen sie auf andere Entwickler und Forscher, die die Open-Source-Community, ihre Sprache und ihre Probleme kennen. Dieser Trend ist positiv – denn zu lange wurden Microsoft und die Open-Source-Community als zwei verschiedene Welten angesehen. Die Vorstellung, dass auf dieser Ebene eine Brücke geschlagen werden kann, ist sehr ermutigend und deshalb machen wir in diesem Bereich mit echten Ergebnissen weitere Fortschritte.

9 Ausblick: Die Zukunft von Open Source

Ein weiterer Bereich, den wir im Linux-/Open-Source-Software-Labor untersuchen, ist das historische *Trending* von OSS. Wir haben die letzten bedeutenden Versionen des Linux-Kernel analysiert, um Fragen über die Evolution des Kernels beantworten zu können. Wird der Code einfacher oder komplexer? Nehmen die Fehler im Jahresvergleich zu oder ab? Wird der Code größer oder kleiner? Wird er effizienter oder verliert er an Effizienz? Auf Basis dieser Forschungsergebnisse haben wir drei Trends ausgemacht:

Wenn man nur die Software-Eigenschaften und den Code betrachtet, wächst OSS ziemlich linear. Jedes Jahr nimmt die Anzahl der Codezeilen zu, die Komplexität erhöht sich entsprechend und dadurch treten häufiger Fehler auf. Das überrascht nicht und versetzt OSS auch keinen Tiefschlag – genau auf dieselbe Weise wird und wurde kommerzielle und nicht-kommerzielle Software entwickelt. Dieses Wachstum sollte aber nicht mit der Modularität von OSS verwechselt werden. Zu Unrecht verstehen viele unter Modularität, dass sich die Systeme nicht weiterentwickeln und die Wartung einfach ist. Das ist aber nicht der Fall – auch modulare Software muss weiter wachsen.

Bei der Entwicklung von OSS bleibt abzuwarten, ob das Community-Development-Modell adäquat auf die steigende Komplexität reagieren kann. Kann dieses lose zusammenhängende Modell, in dem Entwickler mit Linux arbeiten und das Betriebssystem weltweit mit sehr lose definierten Befugnissen, Planungen, Testverfahren oder Strukturen verbreiten, dem Wachstum der Software Rechnung tragen? Oder ist es möglich, dass gerade aufgrund der Struktur, auf der das Community-Development-Modell basiert, die Software nicht weiter wächst und auch nicht komplexer wird?

Auch mit dieser Frage konzentrieren wir uns auf die Erforschung von Software-Techniken – wir suchen dabei nicht nach einer positiven oder negativen Antwort. Wir beobachten diese Entwicklung sehr genau und erforschen diese Thematik auch bei anderen Applikationen außerhalb des Kernels.

Der zweite Trend, den wir beobachten, ist das Wachstum der kommerziellen und professionellen Firmen im Bereich OSS. Im Laufe der Zeit haben wir die Arbeit von Entwicklern an verschiedenen OSS-Projekten nachverfolgt und festgestellt, wer

den Code bearbeitet. Durch die Analyse von Forschungsarbeiten in den Repositories, der wissenschaftlichen Forschung in diesem Bereich und unserer Kommunikation mit der Community wurde das Muster deutlich: In den vergangenen fünf Jahren stammten immer mehr Beiträge von Entwicklern, die in einer kommerziellen Organisation beschäftigt sind. Diese Organisationen schöpfen entweder direkt Geld aus dem OSS-Projekt (wie MySQL oder JBoss) oder indirekt durch Hardware, kommerzielle Software und Services (IBM, Novell, HP). Für alle in der OSS-Entwicklergemeinschaften ist dies nicht erstaunlich. Jeden außerhalb dieses Marktes, der glaubt, dass Entwickler immer noch „in ihrer Freizeit“ an Linux oder anderen OSS-Projekten arbeiten, mag es schon eher überraschen.

Der dritte Trend, den wir ausmachen können, besteht darin, dass man am Markt das OSS-Modell insgesamt deutlicher wahrnimmt – die eigentliche Quintessenz des Phänomens. Bei der Erhebung und Analyse der verfügbaren OSS stellt man fest, dass die meisten Projekte auf System-Software basieren, die für andere Entwickler oder Systemadministratoren erstellt wurde. Ich denke also, dass es angemessen ist, festzustellen, dass OSS in einem größeren, historischen Zusammenhang überwiegend ein Entwickler-Phänomen ist. Und hierin besteht der wesentliche Unterschied zwischen der Entwicklung von Software in kommerziellen Software-Unternehmen und der Entwicklung von OSS: Kommerzielle Software-Unternehmen bauen und konstruieren Software, um damit eine Kundenanforderung zu erfüllen, wohingegen OSS größtenteils von und für Entwickler und technische Benutzer entwickelt wird.

In einigen Bereichen hat dieses Phänomen kräftige Auswirkungen und führt zu starkem Wachstum. Natürlich gibt es einige Ausnahmen, aber dieses Unterscheidungsmerkmal schlägt sich in der modernen Denkweise über OSS nieder. Zudem verdeutlicht es, dass viele Entwicklungsmodelle im Software-Ökosystem nebeneinander existieren können und dies auch in Zukunft tun werden – tatsächlich haben wir festgestellt, dass viele populäre OSS-Server-Applikationen zu Wachstum bei Windows führen (wie JBoss und MySQL).

10 Mehrwert durch ausgewogene Betrachtung der OSS-Trends

Durch die objektive und unvoreingenommene Erforschung der Dynamik von OSS, die sich auf technische Daten stützt, ist das Linux-/Open-Source-Software-Labor in der Lage, Verbesserungen und Veränderungen bei Microsoft selbst und den Kunden zu realisieren, die uns gebeten haben, häufige Fragen und Probleme von Linux und OSS zu untersuchen. Wir sind sehr stolz auf die Ergebnisse aus unserer Forschungsarbeit und werden weiterhin das Gleichgewicht zwischen Kooperation und Wettbewerb mit OSS finden. Wir sind zuversichtlich, dass unsere zukünftigen Forschungsarbeiten für Microsoft, seine Kunden, Partner und die Open-Source-Community von großem Nutzen sein werden. Wir befinden uns in einer sehr spannenden Zeit!