Support für Linuxclients durch opsi

Inhaltsverzeichnis

1. Linux Installationen mit opsi

1.1. Ziel

Ziel des Konzeptes ist eine integrierte Verwaltung von heterogenen Rechnerparks mit Linux- und Windows-Maschinen unter Verwendung eines einheitlichen Managementtools.

1.2. Problemstellung

Um das oben genannte Ziel zu erreichen, müssen einige Probleme betrachtet und gelöst werden:

I. Konzeptionelle Unterschiede in der OS-Installation und Softwareverteilung zwischen Windows und Linux

II. Technische Detailunterschiede zwischen verschiedenen Linuxdistributionen

III. Multibootkonzepte
(mehrere unterschiedliche Betriebssysteme auf einer Hardware)

1.2.1. Konzeptionelle Unterschiede in der OS-Installation und Softwareverteilung zwischen Windows und Linux

Eine Windows-Installation liefert ein vom Umfang her definiertes, aber recht nacktes Betriebssystem. Softwareinstallationen für Windows müssen daher alle Bibliotheken und sonstige Komponenten, die über diesen Grundbestand hinausgehen, selbst mitbringen.

Linuxdistributionen enthalten neben den Grundfunktionalitäten eine Fülle von weiteren Programmen. Dabei werden Komponenten und Bibliotheken, die von verschiedenen Programmen benötigt werden, als gesonderte Pakete angeboten und über Abhängigkeiten in die Installation eingebunden. Dies bedeutet, dass eine Linuxinstallation sich eines Paketmanagementsystems bedienen muss, welche diese Abhängigkeiten auflöst. Gleichzeitig bedeutet dies auch, dass ein Großteil der Software aus den Standardrepositories installiert werden kann. Dies führt dazu, dass zur Installation einer Software wie z.B. Firefox nicht das Installationsprogramm für eine mitgebrachte spezielle Firefoxversion aufgerufen wird, sondern vielmehr per Scriptbefehl das Linuxsystem aufgefordert wird den Firefox zu aktualisieren und dabei auf das Standardrepository zurück zugreifen.

Beide Betriebssysteme verfügen über typische Paketformate zur Softwareinstallation. In der Praxis muss aber davon ausgegangen werden, dass auch Software eingesetzt werden soll, welche zwar nicht im passenden Paketformat vorliegt, aber trotzdem einfach in die Softwareverteilung aufgenommen werden muss. Für eine solche Software die distributionstypischen Pakete zu erstellen, ist mit einem nicht unerheblichen Aufwand verbunden, der nach unserer Auffassung einem normalen Systemadministrator nicht zuzumuten ist. Daher muss es hierfür die Möglichkeit einer einfacheren Einbindung in das Zielsystem geben.

Beide Betriebssysteme benötigen über die Softwareinstallation hinaus die Möglichkeit, einfache Konfigurationsaufgaben in einfach zu erstellenden Scripten als Paket bzw. als Bestandteil eines Paketes (Customizing) zur Verfügung zu stellen.

1.2.2. Technische Detailunterschiede zwischen verschiedenen Linuxdistributionen

Die unterschiedlichen Linuxdistributionen unterscheiden sich bei den verwendeten Paketnamen, Paketformaten, Paketmanagementsystemen und den Automatismen zur OS-Installation.

1.2.3. Multibootkonzepte

Spätestens mit der Verfügbarkeit von unterschiedlichen Betriebsystemen (Windows/Linux) entsteht der Wunsch auch multiboot Lösungen zu unterstützen. Die bedeutet zwischen der Hardware und unterschiedlichen auf dieser Hardware installierten Konfigurationen zu differenzieren. Dies erfordert einen erheblichen Umbau der jetzigen Datenstruktur von opsi.

1.3. Automatisierte OS-Installation

1.3.1. Diskutierte Optionen

- Verwendung von distributionseigenen Methoden wie autoyast (Suse), Kickstart (RedHat), Netinstaller (Debian)
- Verwendung von FAI (http://fai-project.org/)
- Installation vom opsi-bootimage aus

Die Verwendung der distributionseigenen Methoden hat den Nachteil, dass sie sich aufgrund Ihrer starken Unterschiede und teilweise schnellen Wechsel nur schwer in opsi integrieren lassen und eine Quelle eines hohen Pflegeaufwands sein können.

Die für die Betriebssysteminstallation verwendete Mimik über das opsi-linux-bootimage lässt sich auch für eine Installation von Linuxsystemen verwenden. Gleichzeitig bietet diese Option die beste Integration in das opsi-Management. Wir haben uns daher für diese im Folgenden näher beschriebene Methode der Installation entschieden, welche sich am besten in die opsi-Administration eingliedert und distributionsübergreifend möglich ist.

1.3.2. opsi-Konzept

Zur Installation wird über die in opsi vorhandenen Methoden das opsi-linux-bootimage gebootet. Dieses führt dann ein auf dem Server hinterlegtes Script aus. In diesem Script werden folgende Arbeiten ausgeführt:

- Partionierung der Festplatte und Erstellung von Filesystemen
- Installation des Kernels der gewünschten Linuxdistribution
- Installation eines Basis-Sets an Paketen
- Installation eines opsi-Werkzeugs zur weiteren Softwareverteilung (siehe unten)

Das Konzept zur Installation von Linuxsystemen aus dem opsi-bootimage heraus wurde bereits für die Zielsysteme Suse und Debian erfolgreich getestet.

1.4. Softwareverteilung

1.4.1. Diskutierte Optionen

- Verwendung von distributionseigenen Methoden wie zypper (Suse), yum (RedHat), apt-get (Debian)
- Verwendung des distributionsübergreifenden Paketmanagement-werkzeugs 'smart' (http://labix.org/smart)
- Bereitstellung eine opsi eigene Installationsshell (opsiclientd/opsi-winst)

Als Beispiel für die Problematik sei hier die Aufgabe aufgeführt das Programm '7-zip' zu installieren. Hier stellen sich einige Fragen:

- gibt es hierfür ein distributionstypisches Paket ?
- in welchem Repository ?
- wie heißt das Paket
- mit welchem Befehl installiere ich es

Ein distributionsübergreifender Installer wie smart kann von der letzten Frage abstrahieren (mit welchem Befehl installiere ich es). Die Beantwortung der restlichen Fragen lassen sich aber schlecht automatisieren.

Geplant ist daher eine Portierung des unter Windows eingesetzten opsi-client-agenten mit opsiclientd und opsi-winst. Die vom opsi-winst bereitgestellte Scriptsprache ermöglicht es:

- über einfache Funktionsaufrufe die laufende Distribution zu ermitteln um dann distributionstypische Installationsbefehle (yum,zypper,apt-get) abzusetzen
- Nach der Installation kann mit den opsi-winst Befehlen das Customizing des frisch installierten Produktes durchgeführt werden.

1.4.2. opsi-Konzept

Unter Windows verwendet opsi den opsi-client-agenten um Installationen und Konfigurationsarbeiten auszuführen. Dieser besteht im wesentlichen aus zwei Komponenten:

- opsiclientd (Kommunikation mit dem opsi-server, eventgesteuerter Start der Installationen)
- opsi-winst (die eigentliche Installationsshell)

Der opsiclientd wurde Plattform übergreifend konzipiert und in der Sprache Python implementiert. Der Aufwand den opsiclientd nach Linux zu portieren ist daher überschaubar.

Der opsi-winst ist ursprünglich ein reines Windowsprogramm. Durch eine inzwischen abgeschlossene Portierung auf die Sprache Freepascal/Lazarus welche auch unter Linux zur Verfügung steht, ist die Bereitstellung eines opsi-winst unter Linux nun in absehbarer Zeit möglich. Natürlich steckt in der Implementierung einiger Funktionen wie z.B. der Manipulation von Startmenüeinträgen und DeskopIcons noch erhebliche Anpassungsarbeit.

Somit lassen sich für Linux analog wie für Windows opsi-Pakete bauen, welche Software installieren und Konfigurationsarbeiten durchführen können.

Diese opsi-Pakete lassen sich in opsi analog zu den opsi-Paketen für Windows verwalten.

1.5. Konfigurationsmanagement

Durch die schon erwähnte Portierung des opsi-winst und dessen Fähigkeiten Konfigurationsdateien zu patchen, können über opsi Konfigurationsarbeiten am Linux-System vorgenommen werden.

1.6. Hardware-Inventarisierung

Die Hardware-Inventarisierung unter Linux ist bereits programmiert und liegt produktiv in opsi vor.

Spätestens im Rahmen eines Multibootkonzeptes müssen diese Ergebnisse allerdings der Hardware und nicht den unterschiedlichen Betriebssystemen auf dieser Hardware zu geordnet werden. Gleichzeitig sollte aber trotzdem nach Quellen der Hardwareinventarisierung differenziert werden, da sich die unter unterschiedlichen Betriebssystemen erhobenen Daten nicht vollständig aufeinander abbilden lassen (auch wenn es die exakt selbe Hardware ist).

1.7. Software-Inventarisierung

Eine Abfrage der installierten Software, welche im distributionseigenen Paketformat installiert wurde, ist mit den distributionstypischen Methoden problemlos möglich. Anderweitig installierte Software wird dabei nicht erfasst. Zu erstellen sind die Methoden, um die so erhobenen Daten dem vorhandenen Webservice zur Software-Inventarisierung zu übergeben. Die Datenstruktur zur Ablage muss eventuell geringfügig angepasst werden.

1.8. Integration in das opsi-Management Interface

Soweit wie bisher vorgestellt nur opsi-Pakete und nicht direkt die distributionseigenen Pakete gemanagt werden müssen, stellt sich die Integration vergleichsweise einfach dar:

- Für alle Clients muss das installierte Betriebssystem mit Version erfasst und im Management Interface bei der Clientauswahl dargestellt werden. Nach dem Merkmal Betriebssystem kann sortiert werden
- Je nach dem welches Betriebssystem ein ausgewählter Client (oder Clientgruppe) hat, werden in der 'Produktkonfiguration' die für dieses Betriebssystem bereitgestellten Pakete angezeigt.
- Die 'netboot-Produkte' sind Betriebssystem übergreifend und enthalten neben den opsi-Produkten zur Windowsinstallation auch die zur Linuxinstallation.
- Die Hardware-Inventarisierung kann unverändert bleiben
- Die Software-Inventarisierung benötigt eventuell geringfügige Änderungen.

1.9. Zeitlicher Ablauf (Stufenplan)

Das Ziel ein Plattformübergreifendes Clientmanagement System zu schaffen, ist eine große Aufgabe bei deren Bewältigung mit Problemen zu rechnen ist, die bei Beginn noch nicht (oder nicht vollständig) gesehen werden können. Um nicht eine große aber erst in unabsehbarer Zeit fertig werdende Baustelle zu eröffnen, sollte das Projekt in kleine, klar überschaubare und einzeln finanzierbare Abschnitte zerlegt werden.

Die im folgenden vorgestellten Schritte sind Vorschläge welche nach den konkreten Bedürfnissen der Kunden angepasst und modifiziert werden können. Weiterhin dienen Sie als Vorlage zu einer Diskussion mit der opsi Community.

1. Basis-Installation 32Bit
   Basierend auf dem existierenden opsi-linux-bootimage, werden Netbootprodukte bereitgestellt, welche einen Linuxclient mit einem festgelegten Softwareinventar bereitstellen. Die Installation erfolgt aus Standard Repositories.
   - Festlegung des Softwareinventars über Properties
   - Automatische Installation eines opsi-servers
2. ssh Zugriff auf Linuxsysteme vom opsi Management Interface aus
   - Start über das Kontextmenü des Clienttabs
     - In Kombination mit der Möglichkeit auch andere Remote Control systeme (rdp, VNC, teamviewer, dame, ...) entsprechend zu starten.
3. opsi-client-agent für Linux
   - opsi-winst:
     - Aufruf von Programmen und scripten
     - Manipulation von Konfigurationsdateien (ausser XML)
     - Kommunikation mit dem opsi-server
     - noch keine Manipulation von Desktop und Startmenü Einträgen
   - opsiclientd
     - Aktivierung beim Start, zeitgesteuert und von aussen
     - Kommunikation mit dem opsi-server
     - Notifier zur Kommunikation mit dem user
     - keine Aktivierung beim aktiv werden von Netzwerkinterfaces

- opsi-produkte mit obligatorischen Property use_in_os
  Damit wird gekennzeichnet, für welche OS ein Produkt gebaut ist. Existiert das Property nicht, so wird aus Gründen der Rückwärtskompatibilität 'Windows' angenommen' Um Produkte als nur für Linux zu deklarieren bekommt das Property den Wert 'Linux'.

4. Inventarisierung
   Bereitstellung von Scripten zur Hard- und Softwareinventarisierung auf Basis der bisherigen Datenstrukturen.
5. Erweiterung opsi-winst: Manipulation von XML-Dateien sowie Erweiterte Funktionen zur Manipulation Linuxtypischer Konfigurationsdateien. Manipulation von Desktop und Startmenü Einträgen.
6. Basis-Installation 64Bit/LVM
   Bereitstellung eines 64 Bit Linuxbootimage als Basis für Installation von 64 Bit Linuxsystemen
   Erweiterung der Linux-Netbootprodukte um die Möglichkeit des Einsatzes eines Logical Volume Managers
7. Erweiterung der Datenstrukturen im Webservice sowie im Backend.

   Die konkreten Datenstrukturen welche hier implementiert werden, richten sich nach den Erfahrungen, welche in den vorherigen Schritten gemacht wurden. Hier daher nur einige Ideen aus jetziger Sicht.

   - Einführung einer kurzen ergänzenden Information zum Status 'failed' aus dem die Ursache erkennbar ist.
   - neue Inventarisierungs Datenstrukturen
     - zur Berücksichtigung von Linuxspezifika
     - zur Inventarisierung von unmanagend objects (zur Berücksichtigung von Komponenten welche von der Inventarisierung zwar gesehen werden aber nicht durch opsi gemanaged werden wie z.B. Netzwerkprinter, Switches, Unix-Server)
   - Trennung Hardware object und unterschiedliche OS auf dieser Hardware

   Diese Arbeiten werden voraussichtlich nur für das MySQL-Backend durchgeführt. Daher ist die freie Verfügbarkeit des MySQL-Backend hierfür eine Voraussetzung.

8. Anpassung des opsi Management Interface an die neuen

Datenstrukturen

9. Multiboot
   Sowohl Windows- wie Linux-Netbootprodukte werden so angepasst, das eine Koexistenz von zwei (oder mehr) opsi Installationen (Windows, Linux, gemischt) auf einem System möglich ist.
   Dazu werden die Netbootprodukte so erweitert, dass clientspezifisch festgelegt werden kann:
   - wie viele Systempartitionen in welcher Größe angelegt werden sollen, falls diese noch nicht vorhanden sind.
   - auf welche Partition das Produkt installiert werden soll
   Weiterhin wird ein Bootloader installiert, welcher die Möglichkeit einer Auswahl zwischen den Systemen ermöglicht.
   Ergänzt wird dies durch ein Netbootprodukt in welchem der Bootdefault umstellt werden kann und eines welches die angegebene Partition direkt bootet

1.10. Aufwand

Schätzungen, konkrete Summen im Rahmen einer Angebotserstellung.

- Basis-Installation 32Bit (10.000 €) (Interessent)
- ssh Zugriff auf Linuxsysteme vom
- opsi Management Interface aus (5.000 €) (verkauft)
- opsi-client-agent für Linux (15.000 €)
- Inventarisierung (5.000 €)
- Erweiterung opsi-winst (5.000 €)
- Basis-Installation 64Bit/LVM (10.000 €)