Automatisierte Konfiguration und Überwachung einer IT-Laborumgebung mit Open-Source-Tools

Ulrich Greveler
Fachhochschule Münster
greveler@fh-muenster.de

Abstract: Wir beschreiben die Architektur einer Laborinfrastruktur, die allein mithilfe von Open-Source-Tools eine automatisierte (Re-)Konfiguration und Sicherung des Gesamtzustandes erlaubt. Darüberhinaus erfolgt die Überwachung der einzelnen Komponenten und die Visualisierung der damit gewonnenen Daten unter Nutzung freier Software. Die dargestellte Architektur wird produktiv an der FH Münster eingesetzt.

1 Hintergrund

Die im Rahmen dieses Beitrags beschriebene mit Open-Source-Tools (OS-Tools) zu bewältigende Herausforderung besteht im Aufbau einer IT-Laborinfrastruktur, die für Forschung und Lehre genutzt werden soll und besonderen Anforderungen genügen muss.

Unvermeidbare Randbedingung ist die in der Hochschullandschaft weit verbreitete geringe Ausstattung mit finanziellen Mitteln. Im Rahmen eines Antragsverfahrens konnten Mittel zur Anschaffung der notwendigen Hardwareausstattung erzielt werden; eine weitere Finanzierung von kommerzieller Software bzw. von Werkverträgen zur Entwicklung einer maßgeschneiderten Softwaresteuerung ist mit öffentlichen Geldern kaum realisierbar – tatsächlich ist sie im Sinne des Gebots sparsamer Verwendung von Haushaltsmitteln aber auch nicht sinnvoll, da Open-Source-Lösungen verfügbar sind, die eine umfassende, wartbare und damit zukunftsfähige Ausstattung ermöglichen.

1.1 Anforderungen

Die zu nutzende Gesamtinfrastruktur (kurz *Labor* genannt) besteht aus derzeit zwölf PC-Arbeitsplätzen und mehreren Server- bzw. Netzwerkkomponenten. In der Lehre besteht ein Hauptnutzungszweck des Labors in der Realisierung von Praktika zum Thema *IT-Sicherheit*, wodurch sich besondere Umstände ergeben, die über den üblichen Betrieb eines PC-Pools hinausgehen:

- Nutzer erhalten Administrationsrechte und können das Betriebssystem und die Softwareinstallation ändern bzw. (unabsichtlich) unbrauchbar machen.
- Nutzer haben ggf. Zugriff auf den VLAN-Switch bzw. Router und richten Subnetze,

DMZs, Routen etc. ein.

- die Verbindung zum Hochschulnetz bzw. Internet muss für die Nutzungsart eingeschränkt bzw. unterbrochen werden können, ein autarker Betrieb muss daher möglich sein.

Aufgrund weiterer Nutzungsszenarien (z. B. Simulationen auf verteilten Rechnern für Forschungszwecke, Nutzungs als PC-Pool für Programmierübungen, sonstige Nutzung) und unter Berücksichtigung begrenzter Personalkapazitäten ergeben sich folgende Anforderungen

- Das Labor (d. h. alle veränderlichen Komponenten) muss automatisiert in einen von mehreren definierten Grundzuständen zurückkehren können (unabhängig vom aktuellen Zustand).
- Ein konfigurierter Zustand (alle PC-Platteninhalte, Router- und Switch-Konfiguration) muss abgespeichert und wiederhergestellt werden können, ohne dass jeder PC einzeln durch personellen Eingriff berücksichtigt werden muss.
- Der Gesamtzustand soll kontinuierlich visualisiert werden.
- Einzelne Arbeitsplätze sollen auf eine zentrale Ausgabe (Beamer, Leinwand) geschaltet werden können.

Die genannte Einteilung in Muss- bzw. Soll-Anforderungen stellt eine mögliche Priorisierung innerhalb der Umsetzungsphase dar.

2 Realisierung

Die formulierten Anforderungen können durch Nutzung von Open-Source-Produkten in Verbindung mit selbsterstellten Shell- bzw. *Perl*-Skripten umgesetzt werden. Perl [CT98] ist eine mächtige, plattformunabhängige Interpretersprache, die unter freier Lizenz genutzt werden kann; die verfügbaren Interpreter sind zudem OS-Produkte.

2.1 Infrastruktur

In Abb. 1 wird die Infrastrukur dargestellt. Das Labor umfasst 12 PC-Arbeitsplätze, die über Powerswitches (Stromschaltung via TCP/IP) mit dem Stromnetz verbunden sind. Eine zentrale Steuerung erfolgt über einen Linux-Server, der die angeschlossenen PCs an- und abschalten kann. Auf diese Weise können einzelne Rechner eingeschaltet und (mittels *Wake-Up-On-LAN*-Mechanismus) gebootet werden. Das BIOS ist so konfiguriert, dass (via DHCP-BOOTP) ein Booten über das Netz ermöglicht wird; jeder Rechner wird von der Steuerung mit einem Mini-Image versorgt, das ein kompaktes Ramdisk-basiertes Linux enthält und Rechner-individuell Skripte ausführt (Identifikation über MAC-Adresse).

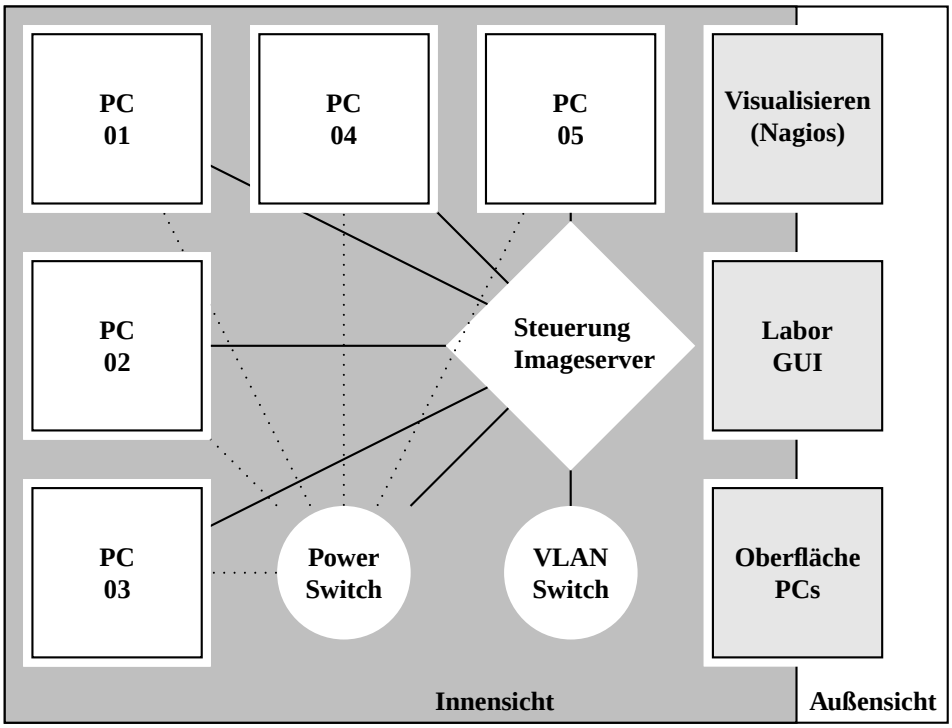

Abbildung 1: Komponenten des Labors

Die PCs sind strikt hardwareidentisch aufgebaut, so dass konfigurierte Betriebssysteme bzw. Software (fast) ohne Anpassungen mittels Plattenimage-Transfer dupliziert werden können[1].

2.2 Wiederherstellen eines Grundzustandes

Ist das Labor in einem undefinierten Zustand (z. B. nach Beenden einer Lehrveranstaltung), kann über eine Labor-GUI die Steuerung mit dem Wiederherstellen eines Grundzustandes (von mehreren) beauftragt werden. Die dabei automatisiert ablaufenden Einzelschritte seien hier exemplarisch genannt.

- Switch und Router werden skriptgesteuert in einen Basiszustand zurückgesetzt.
- Die PCs werden skriptgesteuert mittels Fernadministration heruntergefahren (soweit möglich) und schließlich durch Netztrennung abgeschaltet und (nacheinander) ein-

[1] Beschränkungen existieren hier trotz Hardwaregleichheit, da sich die PCs noch immer unterscheiden (z. B. MAC-Adresse, Festplatten-ID, Prozessor-ID) und daher ggf. Anpassungen beim Systemstart vorzunehmen sind, damit das Betriebssystem die vorgefundene Hardware akzeptiert.

geschaltet. Zur skriptgesteuerten Bedienung der Powerswitches wird das OS-Tool *w3m* eingesetzt, das hier das maschinelle Schalten eines Hardwarewebinterfaces umsetzt.

- Jeder PC bootet nach Wake-up-on-LAN via Netzwerk vom Image-Server (Komponente der Laborsteuerung) ein Ramdisk-basiertes Linux. Dieses startet ein Skript, das unter Nutzung des OS-Tools *udpcast* [Kna07] eine Übertragung des neuen Festplatteninhaltes bzw. einzelner Partitionen vornimmt. Es werden regelmäßig identische Images an alle PCs gleichzeitig übertragen, was durch Broadcast-Technologie seitens des eingesetzten *udpcast* ermöglicht wird.
- Switch und Router werden skriptgesteuert mit der zum Grundzustand gehörigen Netztopologie konfiguriert.
- Die PCs booten nun von ihren Festplatten. Alternativ können alle PCs stromlos geschaltet werden, wenn die Nutzung erst zu einem späteren Zeitpunkt erfolgt. (Hierdurch wird außerhalb der tatsächlichen Nutzungszeiten ein gewisses Energiesparpotential realisiert.)

Die PC-Betriebssysteme (es werden je nach Anforderung verschiedene Linux- bzw. FreeBSD-Distributionen aber auch Microsoft-Betriebssysteme genutzt) sind so konfiguriert, dass eine Remote-Desktopverbindung möglich ist. Genutzt werden zu diesem Zweck OS-Tools, die *Virtual Network Computing* (VNC) implementieren [Kap06]. So kann zu jedem Zeitpunkt ein PC auf den Beamer-PC geschaltet werden, um Vorgänge im Rahmen einer Lehrveranstaltung für alle Teilnehmer zu präsentieren. Darüberhinaus enthält jedes eingesetzte Betriebssystem einen Nagios-Client, der sich mit einem Visualisierungsrechner verbindet, um die Netzwerkstruktur und den Zustand der einzelnen Komponenten sichtbar für alle grafisch abzubilden. Nagios [Bar06] ist eine freie Software, die es ermöglicht, komplexe IT-Strukturen betriebssystemübergreifend abzubilden und zu überwachen.

2.3 Sichern eines Gesamtzustandes

Analog zur im letzten Abschnitt beschriebenen Vorgehensweise kann ein vorhandener Zustand gesichert werden: jeder Festplatteninhalt wird nach Herunterfahren, Power-Off/On nach einem Netzwerkbootvorgang einzeln zum Imageserver übertragen und dort zusammen mit der Netzwerkkonfiguration gespeichert[2]. Auf diese Weise kann eine unterbrochene Labor-Nutzung gesichert und nach einer anderen Nutzung vollständig wiederhergestellt werden, so dass die Nutzer ihre Arbeit fortsetzen können.

Die Möglichkeit, den Gesamtzustand zu sichern, ist neben anderen Bedingungen wichtige Voraussetzung für die effiziente Nutzung der Hardware zu Forschungszwecken. Nach Beenden einer Lehrveranstaltung kann beispielsweise ein Image gebootet werden, das verteiltes Rechnen ermöglicht (Anwendungen: Durchsuchen eines Schlüsselraumes, rechenintensive Simulationen). Das Umschalten auf eine andere Nutzung erfolgt dabei automati-

[2] Eine differentielle Speicherung ermöglicht den Umgang mit großen Datenmengen

siert; es besteht insbesondere keine Notwendigkeit, dass eine Person einen zeitraubenden manuellen Eingriff an jedem PC vornehmen muss.

3 Zwischenfazit und Ausblick

Die beschriebene Infrastruktur wird in mehreren Etappen seit April 2007 aufgebaut und sie wird bereits in der Lehre eingesetzt. Die Versorgung mit Images über einen Netzwerk-Bootvorgang, die automatisierte Stromschaltung und die konfigurierten Remote-Desktop-verbindungen sind bereits produktiv.

Das Kostenersparnis-Potential freier Software konnte mit der vorgestellten Lösung in vollem Umfang genutzt werden. Vergleichbare kommerzielle Produkte, die für einzelne Teilaufgaben herangezogen werden könnten, würden die Investitionskosten des Labors nach grober Schätzung mehr als verdoppeln, wobei Hardware-Gesamtkosten im unteren fünfstelligen Bereich angesetzt wurden. Das für die öffentliche Verwaltung vorherrschende Gebot sparsamer Haushaltsführung legt die Nutzung von Open-Source-Produkten für die dargestellte Architektur nahe.

Ein begünstigender Umstand zur Nutzung freier Software im Umfeld von Forschung und Lehre, der sich nicht auf Unternehmensinfrastrukturen übertragen lässt, besteht in der kostenneutralen Verfügbarkeit studentischer Ressourcen. Kleinere maßgeschneiderte Tools, die für den laufenden Betrieb benötigt werden (z. B. Skripte), können innerhalb von Programmierübungen entwickelt und getestet werden. Adminstrationsaufgaben können von Hilfskräften übernommen, größere Ausbauabschnitte in den Rahmen praxisnaher und projektorientierter Veranstaltungen gefügt werden. Inwieweit das hier skizzierte Kostenersparnispotential außerhalb der Hochschullandschaft umgesetzt werden kann, muss daher im Einzelfall geprüft werden.

Für die Sicherung einer nachhaltigen Labor-Ausstattung stellen OS-Produkte einen weiteren Vorteil dar, der sich bereits nach kurzer Produktivnutzung bewährt hat: die eingesetzten Tools können auf Sourcecode-Ebene modifiziert werden, um eine nahtlose Einpassung in Gesamtvorgänge zu bewirken. Diese Eigenschaft bietet *Closed-Source* grundsätzlich nicht bzw. benötigte Modifikationen können beim Hersteller nur mit erheblichen Folgekosten beauftragt werden. Nach bisherigem Erfahrungsstand wird die nicht zuletzt durch äußere Bedingungen motivierte Entscheidung, auf OS-Produkte auszuweichen, daher als richtig bewertet.

Literatur

[Bar06] Wolfgang Barth. *Nagios: System and Network Monitoring*. No Starch Press, 2006.

[CT98] Tom Christiansen und Nathan Torkington. *Perl Cookbook*. O'Reilly, 1998.

[Kap06] Constantin Kaplinsky. Virtual Network Computing: TightVNC, URL: http://www.tightvnc.com, 2006.

[Kna07] Alain Knaff. UDPcast, URL: http://udpcast.linux.lu, 2007.