

Automatische Installation und Überwachung von Xen Instanzen

Henning Sprang

Silpion IT Solutions GmbH/B1 Systems GmbH

8. Dezember 2006

Über den Redner - Henning Sprang

- ▶ Linux seit 1997 - Free Software Enthusiast und aktiv in diversen Projekten (Debian, FAI, xen-tools, Xen, früher PHP)
- ▶ Systemadministration:
 - ▶ Geofox - Fahrplanauskunft Hamburg
 - ▶ Infrastruktur-Konsolidierung und Entwicklungssysteme
- ▶ Virtualisierung:
 - ▶ Nutzung für Test- und Entwicklungssysteme
 - ▶ VMWare, Qemu seit ca. 2002
 - ▶ Xen seit Ende 2005
- ▶ Monitoring:
 - ▶ 2 Jahre Geonetwatch Testing als interner Kunde
- ▶ Software-Entwicklung:
 - ▶ FAI mit Shell und Perl seit ca. 3 Jahren
 - ▶ Client/Server, Web, XML mit Python, PHP, Ruby(Rails)

Über Xen

Xen ist eine Software mit einigen interessanten Eigenschaften:

- ▶ Open Source Virtualisierung für den Produktivbetrieb
- ▶ Hohe Performance - geringer Virtualisierungsverlust
- ▶ Bis zu 64 physikalische CPU's auf 64Bit
- ▶ Unterstützung von HW-Virtualisierung in neuen Prozessoren
- ▶ Support wichtiger Firmen: HP, IBM, RedHat, Novell, Microsoft, Intel, AMD
- ▶ Rasante Entwicklung
- ▶ Grosse Benutzer-Community
- ▶ Gerüchte sagen: Xen wird de fakto Virtualisierungsstandart

Inhalt - Automatische Installation

Überlegungen zur Systeminstallation

xen-tools

Installation mit FAI

Inhalt - Automatische Installation

Überlegungen zur Systeminstallation

xen-tools

Installation mit FAI

Inhalt - Automatische Installation

Überlegungen zur Systeminstallation

xen-tools

Installation mit FAI

Überlegungen zur Systeminstallation - Ausgangslage

- ▶ Manuelle Installation hat einige Nachteile:
 - ▶ Menschen machen Fehler bei mehreren Wiederholungen
 - ▶ Uneffizient: mehrere VMs installieren dauert lange
- ▶ Lösung: Automatisierung

Festplatten-Images

- ▶ Vorlagen erstellen von sauberen Installationen
- ▶ Diese werden bei Bedarf kopiert und angepasst
- ▶ Vorteile:
 - ▶ Geringer Lernaufwand
 - ▶ Schnell realisiert
- ▶ Nachteile:
 - ▶ Unflexibel - kleinste Änderung bedeutet das Image neu zu erstellen
 - ▶ Immer noch manueller Aufwand
 - ▶ Hohes Datenvolumen bei verschiedenen Konfigurationen

Installation mit Shellsripten

- ▶ Handarbeit wird durch Shellscrippte ersetzt
- ▶ Vorteile:
 - ▶ Deutlich geringerer Speicherbedarf als Disk-Images
 - ▶ Flexibilität
 - ▶ Massgeschneidert
- ▶ Nachteile:
 - ▶ Eigenes Software-Entwicklungsprojekt - Zeit für eigentliche Aufgaben geht verloren - obwohl Entwicklung Spass macht
 - ▶ Man löst alle Probleme selbst - auch die, die andere schon gelöst haben

Nutzung vorhandener Auto-Installer Software

Es gibt fertige Lösungen, die uns diese Aufgabe abnehmen. . .

Die wichtigsten Projekte und Ihre Ziel-Distributionen:

- ▶ Anaconda und Kickstart: Fedora-basierte
- ▶ Autoyast: SuSE-basierte
- ▶ Nlite/Unattend: Windows 2000, XP x86/x64 and 2003 x86/x64
- ▶ Xen-tools: Debian-basierte, verschiedene Fedora/Redhat-basierte, SuSE
- ▶ FAI: Debian-basierte, Fedora-basierte, Mandriva, Windows, Solaris
- ▶ andere: Solaris Jumpstart, RedHat Cobbler/Koan

xen-tools

Xen-tools - eine fortgeschrittene Variante der Scripting-Lösung

- ▶ Von Steve Kemp in Perl geschrieben
- ▶ Legt auch Xen VM-Konfigurations-Dateien an
- ▶ Debootstrap, rpmstrap, tar-archive
- ▶ Hohe Anpassungs- und Erweiterbarkeit durch hook- und role-Scripte
- ▶ Umfassend konfigurierbar, alle Optionen beim Aufruf anpassbar - *Presets* für einzelne Hosts
- ▶ Templates für VM-Konfigurationen über Perl Text::*Template*

Installation und Konfiguration

- ▶ Debian-Pakete
- ▶ Von Quellen (nett: mit `uninstall make target`)
- ▶ Konfigurationsdatei: `/etc/xen-tools/xen-tools.conf`
- ▶ Wichtige Konfigurationsoptionen: `dir`, `kernel`, `initrd`, `debootstrap`, `dhcp`

xen-tools.conf

```
dir=/var/xen/      # storage dir for vm files

debootstrap = 1    # type of installation
dist      = etch   # Default distribution to install
mirror    = http://ftp.de.debian.org/debian/

size      = 2Gb    # Disk image size.
memory    = 128Mb  # Memory size
swap      = 128Mb  # Swap size
fs        = ext3   # use the EXT3 filesystem for the disk image
image     = sparse # Specify sparse vs. full disk images.
dhcp      = 1

kernel    = /boot/vmlinuz-2.6.17-2-xen-686
initrd    = /boot/initrd.img-2.6.17-2-xen-686
```

Benutzung

Anlegen und Installieren:

```
xen-create-image --hostname xen-tools-test
```

Löschen:

```
xen-delete-image xen-tools-test
```

Ausblick

- ▶ Steve Kemp setzt gute Anregungen sehr schnell um
- ▶ Xen-tools funktioniert in ersten Tests auch unter Fedora, Voraussetzungen:
 - ▶ Debootstrap aus Quellen oder mit Alien konvertieren
 - ▶ Paket perl-Text-Template erforderlich
- ▶ SuSE und andere als Basis-Systeme leicht realisierbar
- ▶ Einsatz von `yum`, `mach` und anderen Werkzeugen für weitere Gast-Distributionen

Fragen zu xen-tools?

Fragen zu xen-tools?

Hintergrund - warum FAI

- ▶ Flexibel und leicht erweiterbar
- ▶ Einfacher, aber mächtiger Aufbau
- ▶ Offenes Entwicklungsmodell
- ▶ Hervorragender Community-Support von erfahrenen „Installateuren“
- ▶ Diverse Client- und Server- Distributionen

Hintergrund - Geschichte

- ▶ Entstanden 1999 an der Universität zu Köln
- ▶ Grundidee der strukturierten, geplanten Infrastruktur
- ▶ „Plane Deine Installation und FAI installiert Deinen Plan“
- ▶ Hauptentwickler Thomas Lange
- ▶ Heute etwa 10 aktive Entwickler, aktive Community mit kurzer Reaktionszeit
- ▶ Bestandteil der Debian Distribution

Hintergrund - Wer nutzt FAI wofür

- ▶ LiMux Projekt in München installiert hunderte, final 14000 Clients und Server
- ▶ Thomas Krenn installiert Kundenserver zur Auslieferung
- ▶ Einige top 500 High Performance Cluster
- ▶ Kleine Heimnetzwerke mit 3 Rechnern
- ▶ Lycos-Europe mit 850 Systemen

Funktionen

- ▶ Einige einfache Beispiele, leicht erweiterbar mit eigenen Konfigurationen und Scripten
- ▶ FAI-Klassen: Modularität, „Baukastenprinzip“
- ▶ Server-Distribution: Debian-basierte (Abhängigkeiten aber nur NFS, TFTP, debootstrap)
- ▶ Ziel-Distributionen: Debian, Ubuntu, mit Erweiterungen: Mandriva, Fedora, Suse, Windows
- ▶ Verschiedene Installationsarten
- ▶ Integrierte Versionierung mit Subversion und CVS
- ▶ Extra: `softupdate` zur Wartung und Aktualisierung

Funktionen - Installationsarten

- ▶ Manuelle Nutzung von `dirinstall`
- ▶ `dirinstall` mit `xen-tools` nutzen
- ▶ Netzwerkinstallation mit zentralem Installationsserver
 - ▶ Client/Server Architektur
 - ▶ Paravirtualisiert: durch Eintrag in VM Konfiguration `config nfsroot` mounten
 - ▶ HVM: Fai-Boot-CD iso booten (Xen-VMs noch nicht PXE-fähig)
- ▶ `fai-cd` (hauptsächlich zu Testzwecken/FAI Entwicklung)

Funktionen - Ablauf der Installation

Leicht vereinfacht:

- ▶ Vorbereitung: Mounten des Zielverzeichnis, Booten über PXE, Boot CD oder Install (je nach Installationsart)
- ▶ Ab hier beginnen die eigentlichen FAI tasks:
- ▶ `defclass`: Klassendefinition für den Zielrechner
- ▶ `partition`: Partitionierung
- ▶ `debconf`: Debconf-Voreinstellungen
- ▶ `instsoft`: Software Installation
- ▶ `configure`: Scripts ausführen zur Konfiguration
- ▶ `savelog`: Logdateien auf Installserver speichern

Vorüberlegungen

- ▶ Entscheidung für die passende Installationsart (net/cd/dirinstall)
- ▶ Planung der Installation
 - ▶ Anwendungsfälle
 - ▶ Netzwerk
 - ▶ Software-Pakete
 - ▶ Konfigurationsdateien und Anpassungen

Xen Kernel vorbereiten

- ▶ Für Netzwerkinstallation muss der Kernel bestimmte Funktionen unterstützen:
 - ▶ DHCP auf Kernel-Ebene: `IP_PNP_DHCP`
 - ▶ Root Filesystem via NFS: `ROOT_NFS`
- ▶ Oder: Initial Ramdisk anlegen, die IP und Root location mit `udhcpc` holt, und NFS-Root mounted

FAI einrichten und konfigurieren

1. Installation unter Debian etch:
 - ▶ `apt-get install fai-quickstart`
2. Server-Konfiguration anpassen in `/etc/fai`:
 - ▶ `fai.conf`: `LOGUSER=fai`, `LOGPROTO=ssh` für Logging via ssh statt rsh
 - ▶ `apt/sources.list`: lokalen Mirror eintragen wenn vorhanden
3. FAI Server ist konfiguriert - NFS-Root erzeugen:
 - ▶ `fai-setup`
4. Für HVM (PXE-Boot noch nicht möglich mit Xen):
 - ▶ Fai-Boot-CD erstellen: `make-fai-bootfloppy`

VM Konfiguration und Block device anlegen

- ▶ Möglichkeiten:
 - ▶ Manuell, wie bei jeder VM-Installation
 - ▶ Mit xen-tools - viel bequemer und schneller
- ▶ Besondere Optionen für Netzwerkinstallation:

```
dhcp = 'dhcp'  
extra = "FAI_ACTION=install"  
root = "/dev/nfs"
```

- ▶ Durch Scripting in VM config nur eine Konfigurationsdatei

Infrastrukturdienste einrichten

► Für Netzwerk-Installation:

- DNS Eintrag für FAI Server
- DHCP Server konfigurieren - Host/IP/MAC wie üblich
- FAI-spezifisch:

```
authoritative; # wegen IP_PNP_DHCP
option root-path "/usr/lib/fai/nfsroot ..."
server-name "faiserver";
next-server 172.20.2.64;
```

Configspace prüfen und anpassen - Übersicht

- ▶ Hier sind die Konfigurationen der Install-Clients abgelegt
- ▶ Die ermittelten Anforderungen und der Plan der Installation sind hier abgebildet
- ▶ Struktur von `/srv/fai/config`:

```
class
debconf
disk_config
files
hooks
package_config
scripts
```

Configspace anpassen - class

- ▶ `class` enthält Klassen- und Variablen-Zuordnungen.
- ▶ Am einfachsten erfolgt dies auf Basis von Hostnamen
- ▶ Mitgelieferte Beispielklassen: FAISERVER, GNOME, DEMO, XORG
- ▶ Alles, was anhand eines Scriptes einen Klassennamen als String ausgeben kann, ist geeignet, Klassen zuzuweisen.
- ▶ Also: Abfrage eingebauter Hardware, Festplatten-Grösse, Mac oder IP-Adresse - alles ist möglich

Configspace anpassen - disk_config

- ▶ Konfiguration der Festplatten-Partitionierung und Mount-Points
- ▶ Bei paravirtualisierten Domains nicht nötig - Partitionierung ist gegeben

```
#<type> <mountpoint> <size mb> [mount options][;extra options]
```

```
disk_config disk1
```

```
primary / 150-300 rw,errors=remount-ro ; -c -j ext3
logical swap 40-500 rw
logical /var 90-1000 rw ; -m 5 -j ext3
logical /tmp 50-1000 rw ; -m 0 -j ext3
logical /usr 200-4000 rw ; -j ext3
logical /home 50- rw,nosuid ; -m 1 -j ext3
# logical /home preserve9 rw,nosuid ; -m 1 -j ext3
```

Configspace anpassen - debconf

- ▶ Voreinstellungen für Paket-Installations-Skripten
- ▶ Nur für dpkg-basierte Distributionen
- ▶ Funktionieren bei FAI analog zum Debian-Installer

Configspace anpassen - package_config

- ▶ Inhalte: Dateien, nach Klassennamen benannt
- ▶ Zweck: Definition der zu installierenden Pakete
- ▶ Unterstützt eine Vielzahl an Installationsmethoden:
 - ▶ install (apt-get)
 - ▶ aptitude
 - ▶ taskinst (Debian tasks=Paketsammlungen)
 - ▶ urpmi (mandriva)
 - ▶ yumi (Fedora)
 - ▶ y2i (SuSE y2pms)
 - ▶ yast (SuSE yast -i)
 - ▶ Und noch einige spezielle. . .

Configspace anpassen - package_config II

Beispiel package_config/DEMO aus simple examples:

```
PACKAGES aptitude  
fortune-mod fortunes  
rstat-client #rstatd  
rusers rusersd
```

```
# only when also class XORG is defined  
PACKAGES aptitude XORG  
bb frozen-bubble xpenguins
```

Configspace anpassen - scripts

- ▶ scripts enthält Skripte, die nach der Paketinstallation ausgeführt werden
- ▶ Üblicherweise Shell-, Perl- und cfengine-Scripte
- ▶ Andere einfach: Nur Interpreter in NFS-Root installieren
- ▶ Namensschema:
 <KLASSENNAME>/<NUMMER>-<SCRIPTNAME>
- ▶ Nummer gibt Reihenfolge der Ausführung an
- ▶ Scriptname beliebig, nur zur Lesbarkeit

scripts Beispiel:

```
.
|-- AMD64
|   '-- 99-discover-bug
|-- DEMO
|   |-- 10-misc
|   '-- 30-demo
|-- FAIBASE
|   |-- 10-misc
|   |-- 20-removable_media
|   |-- 30-interface
|   '-- 40-misc
|-- FAISERVER
|   |-- 10-conffiles
|   '-- 20-copy-mirror
|-- GRUB
|   '-- 10-setup
'-- LAST
    '-- 50-misc
```

Configspace anpassen - files

- ▶ Abbild des Filesystems ab /
- ▶ Zur Nutzung mit fcopy/ftar = klassenbasiertes Kopieren und Entpacken
- ▶ Für zusätzliche Distributionen auch Base-Images:
base-*KLASSENNAME*.tgz
- ▶ Kopieren ausdrücklich in Scripten oder rekursiv von /

Configspace anpassen - files II

- ▶ Zieldatei=Verzeichnis
- ▶ Quelldatei=Klassenname

```
.
'-- etc
  |-- X11
  |   '-- xorg.conf
  |       |-- ATI
  |       '-- NVIDIA
  |-- apache2
  |   '-- conf.d
  |       '-- FAISERVER
  '-- fai
      '-- fai.conf
          '-- FAISERVER
```

...

Configspace anpassen - hooks

- ▶ Hook Namensschema:
`<TASKNAME>.<KLASSENNAME> [.source]`
- ▶ Ausführung vor dem entsprechenden Task
- ▶ Optionales `skip_task`: der eigentliche Task wird übersprungen, zum Beispiel die Partitionierung für paravirtualisierte Xen Domains
- ▶ Beispiel:
`partition.XENU`
`instsoft.FAIBASE`
`savelog.LAST.source`

Installation durchführen

Je nach gewählter Installationsart:

- ▶ VM starten mit `xm create fai-test-vm install=1`
- ▶ `fai dirinstall <ZIELVERZEICHNIS>` in das gemountete Blockdevice
- ▶ `dirinstall` in `xen-tools` integriert aus `hooks` ausführen
- ▶ HVM-VM mit `fai-cd` als Boot-Device starten

Inbetriebnahme

- ▶ System ist je nach Anzahl der Pakete in 5- Minuten installiert
- ▶ Starten mit der Produktions-Konfiguration
- ▶ Funktionstests

Ausblick/Weitere Möglichkeiten

- ▶ Hilfs-Skripte und Konfigurationen für weitere Distributionen mit Zusatz-Paket *fai-multi-distribution*
- ▶ GOSA als LDAP und FAI GUI
- ▶ Automatische Tests der installierten Systeme:
 - ▶ hooks/scripts prüfen Dateien/Konfigurationen
 - ▶ Crucible Test Framework
- ▶ In Arbeit:
 - ▶ FAI Server als Xen VM herunterladen
 - ▶ Xen-FAI-Live-CD

Weitere Informationen

- ▶ Im WWW:
 - ▶ <http://www.informatik.uni-koeln.de/fai/>
 - ▶ <http://faiwiki.informatik.uni-koeln.de/>
 - ▶ <http://www.infrastructures.org/>
- ▶ Email und Chat:
 - ▶ IRC-Channel #fai im OFTC-Network
 - ▶ linux-fai-users und linux-fai-devel Mailinglisten

Ende des ersten Teils

Fragen zu FAI/automatischer
Installation?

Inhalt - Monitoring von Xen Instanzen

Überblick

Wichtige Daten auf Xen-Systemen

Durchzuführende Aktionen

Einrichten der Überwachung am Beispiel von Nagios

DTC-Xen

Inhalt - Monitoring von Xen Instanzen

Überblick

Wichtige Daten auf Xen-Systemen

Durchzuführende Aktionen

Einrichten der Überwachung am Beispiel von Nagios

DTC-Xen

Inhalt - Monitoring von Xen Instanzen

Überblick

Wichtige Daten auf Xen-Systemen

Durchzuführende Aktionen

Einrichten der Überwachung am Beispiel von Nagios

DTC-Xen

Inhalt - Monitoring von Xen Instanzen

Überblick

Wichtige Daten auf Xen-Systemen

Durchzuführende Aktionen

Einrichten der Überwachung am Beispiel von Nagios

DTC-Xen

Inhalt - Monitoring von Xen Instanzen

Überblick

Wichtige Daten auf Xen-Systemen

Durchzuführende Aktionen

Einrichten der Überwachung am Beispiel von Nagios

DTC-Xen

Überlick

- ▶ Auch für Xen gilt: Kontinuierliche Überwachung ist Trumpf!
- ▶ Dienst-Qualität und -Verfügbarkeit von aussen wie gehabt
- ▶ Einige Werte sind anders zu überwachen (z.B. CPU)
- ▶ Zusätzlich einige Xen-spezifische Werte
- ▶ Voraussetzung: Kenntnis des physikalischen und virtuellen Netzwerks, der Dienste und Anforderungen, SLA

Fragen, die das Monitoring beantworten soll

- ▶ Laufen alle Dienste mit der dem Kunden zugesagten Leistung?
- ▶ Steht ein Desaster kurz bevor (Z.B. Festplatte läuft voll)?
- ▶ Hat ein Xen Host Kapazitäten für weitere VMs?

- └ Wichtige Daten auf Xen-Systemen
- └ Daten, die diese Fragen beantworten

Daten, die diese Fragen beantworten I

Grundlegende Daten über einzelne DomU Systemen

- ▶ Anwendungen und Dienste - korrekte Antworten im definierten Zeitraum wie im SLA geplant
- ▶ Daten der virtuellen Hardware:
 - ▶ RAM- Auslastung
 - ▶ Swap-Nutzung
 - ▶ Festplatten-Auslastung
 - ▶ *nicht* CPU - siehe Domain-0

Daten, die diese Fragen beantworten II

Allgemein wichtige Information über den Xen Host Server:

- ▶ Ebenso grundlegende Daten wie bei anderen Systemen
- ▶ S.M.A.R.T Daten
- ▶ SNMP Daten, insbesondere bei Routern und Netzwerkdiensten
- ▶ *Füllstand* von LVM VGs = freier Platz für neue VMs

Daten, die diese Fragen beantworten III

Xen-spezifische Werte auf dem Xen Host Server:

- ▶ Wichtige, für Xen relevante Prozesse:
 - ▶ xend, xenwatch, xenbus, xenstored, xenconsole
 - ▶ Fehlermeldungen in xend Logfiles
- ▶ Prozessorverbrauch pro Domain und gesamt
- ▶ Prozessorauslastung Gesamtsystem (= Summe aller Domains)
- ▶ Netzwerk und Disk-I/O-Auslastung

Daten, die diese Fragen beantworten IV

Ja nach konkreter Fragestellung und Umgebung zu beachten:

- ▶ Aktuelle Echtzeit-Daten
- ▶ Überschreiten von Schwellwerten
- ▶ Durchschnittswerte/Zeitraum
- ▶ Spitzenwerte/Zeitraum
- ▶ Trends/Zeitraum

Datenquellen I

Erprobte, funktionierende Wege:

- ▶ xentop (-b -i 2 für maschinelle Verarbeitung)
- ▶ xm list (-long für maschinelle Verarbeitung und Details)
- ▶ xenmon.py (allerdings etwas kryptische Ausgaben)
- ▶ dtc-xen Statistiken (wie XenStats, zusätzlich Netz I/O)
- ▶ XenStats (Grafiken für CPU-Last aus xm list/rrdtool)

Datenquellen II

Weitere naheliegende - noch nicht weiter untersucht:

- ▶ libvirt remote/lokal
- ▶ xend XML-RPC API Unix Socket oder HTTP
- ▶ xenbaked
- ▶ xenoprof
- ▶ dtc-xen soap interface

Datenquellen III

In Planung/Entwicklung:

- ▶ XenMon XML-RPC API
- ▶ libxenstats (xentop-C-API, generierte Python Bindings)

Durchzuführende Aktionen

Verschiedene Aktionen möglich anhand der Monitoring-Daten

- ▶ Problembekämpfung bei Fehlfunktionen in:
 - ▶ Xen/Domain-0
 - ▶ Linux-System
 - ▶ Anwendungs- und Dienst-Software
- ▶ Manuelle oder automatische Migration
- ▶ Veränderungen von RAM- und CPU-Scheduling-Einstellungen
- ▶ LVM-VGs und VM-Dateisysteme vergrößern wenn möglich
- ▶ Hardware verstärken in vorhandenen Systemen oder durch zusätzliche Systeme

Überblick

- ▶ Überwachung der zuvor genannten Werte mit Nagios
- ▶ Nagios Grundlagen werden nicht vertieft
- ▶ Soweit möglich, werden Standard-Plugins verwendet
- ▶ Einige spezielle Plugins sind zu entwickeln
- ▶ Daten werden per NRPE geholt
- ▶ Nagios-Installation sollte nicht auf Domain-0 liegen
- ▶ Bestenfalls externes System: Aussagekräftig für Dienst-Erreichbarkeit

- └ Einrichten der Überwachung am Beispiel von Nagios
- └ Nagios installieren und grundlegend konfigurieren

Nagios Grund-Konfiguration

- ▶ Nagios Basisinstallation
- ▶ NRPE-Server und Plugins installieren auf Domain-0 System
- ▶ Hostgroup xen-servers einrichten
- ▶ Basis-Services und Checks einrichten wie bei realer Hardware
- ▶ Gute Anleitung: „Nagios“ von Wolfgang Barth - erschienen bei Opensourcepress

- └ Einrichten der Überwachung am Beispiel von Nagios
- └ Xen-spezifische Werte überwachen

Xen-spezifische Werte überwachen

- ▶ Prozesse: zuvor genannte Prozesse mit `check_procs` prüfen
- ▶ `xend.log` und `xend-debug.log` auf Auftreten von „ERROR“ und „WARNING“ mit `check_log`
- ▶ Füllstand in LVM VGs: `check_lvm` entsprechend erweitern

Ein Nagios Xen Plugin

- ▶ Geschrieben in Python - leichte Anbindung an Xen-API's
- ▶ Prüft momentan CPU-Gesamtlast aus `xentop -b -i 2`
- ▶ Noch im Anfangsstadium - Alpha-Version im internen Test
- ▶ Weiterentwicklung geplant
- ▶ Veröffentlichung Anfang 2007 auf Nagios-Exchange

Nagios Xen Plugin - Offene Punkte

- ▶ Weitere Datenquellen: XenMon, libxenstat, XenoPerf
- ▶ Anpassen an Systeme mit mehreren CPU's oder Kernen
- ▶ Längere Messintervalle für bessere Durchschnittswerte
- ▶ Trends mittels perf2rrd und Nagiosgraph
- ▶ Nach *Redaktionsschluss* entdeckt: check_xen von Sebastian Clanzett - gegenseitige Ergänzungsmöglichkeiten prüfen

Xen-Daten visualisieren mit DTC-Xen

- ▶ DTC-xen ist eine Erweiterung von DTC
- ▶ Domain Technologie Control: Hosting-Management
- ▶ Schnell und einfach Netz- und CPU-Last-Diagramme erzeugen

DTC-xen installieren und nutzen I

▶ **sources.list:**

```
ftp://ftp.gplhost.com/debian/ stable xen
```

▶ **Installieren:**

```
apt-get install dtc-xen
```

▶ **Graphen sind sofort auf <http://hostname/dtc-xen/> verfügbar**

DTC-xen installieren und nutzen II

- ▶ Nachteil: dtc-xen enthält noch viele andere Dinge, für den Betrieb mit DTC Hosting Panel
- ▶ Lösung: Code für Statistik-Graphen extrahieren, unabhängig installieren, paketieren, weiterentwickeln

Das *Xen* Buch

Dieser Vortrag ist eine Vorschau auf das Buch *Xen - Virtualisierung unter Linux*

- ▶ Erscheinungsdatum: ca. 3/2007
- ▶ Verlag: Opendevelopmentpress - <http://www.opendevpress.de>
- ▶ Autoren: Timo Benk, Jarek Zdrzalek, Ralph Dehner und Henning Sprang
- ▶ Sponsoren:
 - ▶ B1-Systems GmbH - <http://www.b1-systems.de>
 - ▶ Silpion IT Solutions GmbH - <http://www.silpion.de>
- ▶ ISBN 3-937514-29-5
- ▶ WWW (ab Anfang 2007): <http://www.xen-info.de>

Ende des zweiten Teils

Fragen?