

Linux in einer mittelgroßen Institution

... mit der Aussicht zu wachsen

A. Hirczy C. Pfaffel-Janser

Technische Universität Graz
Institut für Theoretische Physik – Computational Physics

Grazer Linxutage 2005
14. Mai 2005



Wir stellen unsere Methoden vor, einen Cluster von Linux-Computern mit minimalem Aufwand in Betrieb zu halten.

Unsere Ziele:

- ▶ Benutzer sind mit dem angebotenen Service zufrieden :)
- ▶ Alle PCs verhalten sich für alle Benutzer **identisch!**
- ▶ Jede administrative Tätigkeit muss vernünftig **skalieren!**
- ▶ **Sicherheit ist uns wichtig** – aus purem Egoismus!

Worüber wir heute sprechen:

- ▶ Vorstellen von Design-Entscheidungen
- ▶ Vor- und Nachteile unserer Lösungen
- ▶ Probleme bei der Implementierung

Vorstellung

Die Installation umfasst das **Institut für Theoretische Physik – Computational Physics** an der Technischen Universität Graz und einen **Computerlehrsaal** für Mathematik- und Physik-Studenten.

Der Arbeitsschwerpunkt liegt bei der mathematischen Behandlung von Fragestellungen vor allem im Bereich der **Vielteilchenphysik** und der **Plasmaphysik** – dabei kommen sowohl analytische als auch numerische Verfahren zum Einsatz.

Wir betreiben etwa 80 PCs (130.000 MIPS und 45.000 MFLOPS) und 10 Notebooks unter Linux und haben zur Zeit 640 Useraccounts, von denen im Regelfall ungefähr 50 gleichzeitig aktiv sind.

Welche Betriebssysteme werden noch eingesetzt?

- ▶ *MS Windows 2000/XP* — Sekretariat und 2 Laptops
- ▶ *Apple MacOS 9*
- ▶ *Apple MacOS X*
- ▶ *Tru64 UNIX V5.x* — 2 × DEC AlphaStation — sind bis jetzt wegen des besseren Fortran Compilers erhalten geblieben, können aber mit GCC 4.0 möglicherweise auf Linux portiert werden
- ▶ *OpenBSD* — für besonders sicherheitskritische Anwendungen (Authentizierung)

Aufgaben

- ▶ Was macht die Arbeit an einem TU-Institut einfacher?
 - ▶ keine enormen Kostenprobleme bei kurzzeitigen Ausfällen
- ▶ Was macht die Arbeit an einem TU-Institut komplizierter?
 - ▶ Einschränkungen sind kaum möglich – jeder wissenschaftliche Mitarbeiter muss imstande sein, eigene Software auszuführen und mitgebrachte Datenträger aller Art zu lesen.
 - ▶ Gelegentlich sind Hilfeleistungen beim Programmieren notwendig.
 - ▶ Hardware bleibt aus finanziellen Gründen teilweise weit über eine sinnvolle Lebensdauer hinaus in Verwendung.
 - ▶ Bei Fehlern stehen 10 Personen hinter mir, die sich für die Lösung des Problems interessieren.
 - ▶ Spieltrieb – alle vorhandenen Möglichkeiten werden genutzt, manches neue Gadget vehement gefordert

Bürobetrieb

- ▶ Grafische Benutzeroberflächen: GNOME, KDE, fvwm, fluxbox, ...
- ▶ vi, emacs, kile/kate, ...
- ▶ \LaTeX mit diversen Erweiterungen als primäre Umgebung für wissenschaftlichen Textsatz
- ▶ OpenOffice, AbiWord, Gnumeric, Scribus, xfig, InkScape, GIMP, ...
- ▶ Mozilla, Firefox, Thunderbird, Evolution, pine, mutt, ...
- ▶ wine: MS Office, Adobe Acrobat, Microcal Origin
- ▶ VMware: als letzte Ausweichmöglichkeit

Serverdienste

- ▶ Fileservice – durch AFS und Kerberos weltweit, sicher und ausreichend schnell möglich
- ▶ Electronic Mail
- ▶ Drucken im Netzwerk – aus Kostengründen haben wir kaum Drucker an den Arbeitsplätzen, sondern je einen hochqualitativen SW-Laser und Farbdrucker je Stockwerk
- ▶ WWW-Server mit Wiki und Forum
- ▶ externer Zugang mit SSH

Forschung & Lehre – Programmierung und Applikationen

- ▶ Fortran IV, 77, 95
- ▶ C und C++
- ▶ Perl, Python, Shell, Java
- ▶ Matlab mit SimuLink und FEMLab
- ▶ Maple
- ▶ Mathematica
- ▶ WWW-Server mit Wiki und Forum

High Performance Computing (HPC)

- ▶ MPI – Message Passing Interface
- ▶ PVM – Parallel Virtual Machine
- ▶ werden vorwiegend zum Testen von selbstentwickelten Programmen verwendet:
 - ▶ Unsere Rechner sind über 6 Jahre hinweg angeschafft worden und in ihrer Leistung sehr heterogen.
 - ▶ Die Leistungsfähigkeit der einzelnen Knoten ist durch die Belastungen des interaktiven Betriebes schwer vorherzusehen.

Die Programme laufen später vor allem auf dedizierten Clustern (ZID).

High Throughput Computing (HTC)

Der Schwerpunkt numerischer Simulationen am Institut liegt im Bereich des High Throughput Computing: Alle PCs mit nennenswerter Rechenleistung sind Teil eines Linux-Clusters, ungenutzte Ressourcen werden vom Batchsystem **Condor** (<http://www.cs.wisc.edu/condor/>) an Prozesse in einer Warteschlange vergeben.

- ▶ bei Interaktion durch den Benutzer der Konsole werden die Batch-Prozesse gestoppt
- ▶ Auslastung der PCs wird durch Hintergrundprozesse laufend bestimmt
- ▶ wenn Vordergrundprozesse Rechenleistung benötigen, werden Batch-Prozesse gestoppt
- ▶ gestoppte Prozesse werden nach 15 Minuten auf eine andere, freie Maschine migriert

Existierende Hardware

Die angespannte Finanzsituation bedingt durch die lange Einsatzdauer von Hardware ein sehr heterogenes Bild an eingesetzter Hardware: Von Pentium II bis AMD Opteron ist alles im Einsatz. Eingesetzt werden:

- ▶ Hauptsächlich Dual-Prozessor x86 Maschinen an den Arbeitsplätzen und in einem Regal
- ▶ Single-Prozessor x86 Maschinen wenn es die Lärmentwicklung oder die Finanzen nicht anders zulassen
- ▶ Klimatisierter Rechnerraum
- ▶ Dedizierte Server mit SCSI-Lauwerken, bzw. SATA-RAIDs
- ▶ verschiedene Alphas, verschiedene Macs
- ▶ zentrale Drucker, Scanner und Brenner
- ▶ Nespresso-Maschine

Neubeschaffung von Hardware

Punkte zur Berücksichtigung bei der Anschaffung von Hardware:

- ▶ ohne Betriebssystem
- ▶ Geräte soweit möglich in Ausführung bulk ohne Treiber und Software
- ▶ Manuals für Mainboard, Netzteil und Gehäuse
- ▶ Der Luftstrom durch das Gehäuse darf nicht eingeschränkt sein; besonders beim vorderen Lüfter muss das Gehäuse ausreichende Öffnungen aufweisen
- ▶ alle Komponenten müssen für durchlaufenden Betrieb geeignet sein
- ▶ Garantie für Kompletgerät: 3 Jahre vor Ort
- ▶ Möglichst nur gleiche Geräte

Aktuelle Ausstattung Workstation

Wie schaut die derzeit gewünschte Ausstattung einer Workstation aus:

- ▶ Dual Opteron
- ▶ RAM \geq 1 GB
- ▶ Festplatte, SATA, \geq 35 GB
- ▶ Sorgenkind Graphikkarte: NVidia

Ausstattung Server

Bei der Neuanschaffung von Servergeräten wird Wert gelegt auf

- ▶ 19" Rackmount Gehäuse mit ausreichender Kühlung
- ▶ RAID 1/5
- ▶ Festplatten in Wechselrahmen
- ▶ Redundante Stromversorgung
- ▶ USV

Warum Linux?

Traditionell wurden in der Physik und Mathematik für anspruchsvolle Aufgaben UNIX-Workstations verwendet – das Erscheinen von Linux erlaubte es plötzlich, die (bereits bekannten) Vorteile der UNIX-Umgebung zu wesentlich geringeren Preisen deutlich mehr Personen zugänglich zu machen.

Die Stabilität von Linux trägt angesichts der Programmlaufzeiten von oft mehreren Tagen wesentlich zur Akzeptanz bei.

Ein Umstieg auf MS Windows hätte darüberhinaus bedeutet, wesentliche Teile von bereits existierenden Programmen neu zu schreiben und dabei Zeit in wissenschaftlich uninteressante Bereiche (GUIs) zu investieren oder mit dem inferioren *command line interface* von MS Windows zu arbeiten.

Warum Debian GNU/Linux?

- ▶ automatische Installation
 - ▶ Replicator
 - ▶ FAI – Fully Automatic Installation
- ▶ automatische Security-Updates
- ▶ automatisches Upgrade auf neue Releases (fast)
- ▶ automatisches, nachträgliches Installieren von Software

Wir haben den Cluster von 1998 bis 2004 mit *Red Hat Linux* betrieben; alle Punkte waren (vor allem durch apt) auch unter Redhat möglich. Den Wechsel von Redhat auf Fedora wollten wir wegen der zu kurzen Releasewechsel nicht mehr nachvollziehen.

Updates und Nachinstallation

Nach der Installation mit **replicator** ist ein laufendes Update notwendig, insbesondere, da wir uns für Debian Sarge/testing entschieden haben. Um uns ausreichenden Spielraum zu schaffen, haben wir ein eigenes Repository von Debian Paketen am Instituts-Webserver geschaffen.

Gesteuert wird die Nachinstallation durch das shell script **pkgtool**, dem als Parameter jeweils ein Directory übergeben wird.

Dateien `do_upgrade` und `do_dist-upgrade` triggern nach einem Vergleich der Zeitstempel mit Log-Files die entsprechenden Aktionen von `apt-get`, Directories mit Paketnamen in den Subdirectories `add` und `delete` veranlassen die Installation / Deinstallation von Paketen. Innerhalb der Paketdirectories können in Dateien `debconf` Daten für die Konfiguration untergebracht werden; zusätzlich können für besonders schwer zu installierende Pakete `expect`-Skripte verwendet werden.

Konfigurationsdirectory von pkgtool

```
/afs/itp.tugraz.at/common/local/etc/pkgtool/workstation/  
|-- add/  
|   |-- auctex/  
|   |   '-- debconf  
|   ....  
|   |-- emacs-wiki/  
|   ....  
|   |-- postfix/  
|   |   '-- debconf  
|   ....  
|   |-- unzip/  
|   '-- zip/  
|-- do_upgrade  
'-- remove/  
    '-- chrony/
```

Cfengine

Cfengine – die Configuration Engine – kann automatische Änderungen der Rechnerkonfiguration auf allen unseren Systemen durchführen. Sie nimmt eine Beschreibung des gewünschten Systemzustandes in einer sehr abstrahierten Sprache entgegen und versucht auf portable und möglichst unauffällige Weise diesen Zustand zu erreichen.

Die notwendigen Aktionen werden durch Klassen gesteuert – eine Reihe von Klassen (Architektur, Datum und Uhrzeit, Hostname, ...) erkennt Cfengine selbst; eigene Klassen können einfach definiert werden.

<http://www.cfengine.org/>

Mounten, links und permissions setzen, aufräumen

control:

```
actionsequence = ( mountinfo addmounts mountall
                  links files tidy )
```

miscmounts:

```
faeppc22:/itp/faeppc22/temp /itp/faeppc22/temp mode=intr
```

links:

```
/temp ->! /itp/faeppc22/temp
```

files:

```
faeppc22::
```

```
/itp/faeppc22/temp recurse=0 act=fixall
mode=1777 o=root g=root
```

tidy:

```
faeppc22.Saturday.Hr21::
```

```
/itp/faeppc22/temp/ pattern=* recurse=inf age=365
```

Klassen definieren, shell commands zur RAID-Überwachung

control:

```
actionsequence = ( shellcommands )
```

groups:

```
HAS_RAID_3WARE = ( FileExists(/usr/local/sbin/tw_cli) )  
HAS_RAID_SW     = ( FileExists(/sbin/mdadm) )
```

shellcommands:

```
HAS_RAID_3WARE.Monday.Hr07::  
  "/usr/local/sbin/tw_cli info"  
  "/usr/local/sbin/tw_cli info c1"
```

```
HAS_RAID_SW.Monday.Hr07::  
  "/bin/cat /proc/mdstat"  
  "/sbin/mdadm --detail /dev/md0"
```

Konfiguration von SSH

editfiles:

linux::

```
{ /etc/ssh/sshd_config
  DefineClasses "CONF_CHANGE_SSHD"
  DeleteLinesMatching "#.*"
  DeleteLinesMatching "^$"
  DeleteLinesMatching "PrintMotd .*"
  DeleteLinesMatching "X11Forwarding no"
  AppendIfNoSuchLine "X11Forwarding yes"
}
```

shellcommands:

CONF_CHANGE_SSHD.debian::

```
"/etc/init.d/ssh-krb5 restart"
```

CONF_CHANGE_SSHD.redhat::

```
"/etc/init.d/sshd restart"
```

Eingriffe in Homedirectories

Wir überarbeiten manchmal alle Homedirectories um alte *core dumps* zu löschen, den Browser-Cache nach `/tmp` zu legen, Permissions und Access Control Listen auf wichtige Dateien und Directories korrekt zu setzen. Um diese lästige Arbeit zu vermeiden und Eingriffe in die Privatsphäre der Benutzer gering zu halten, haben wir dazu ein Perl-Skript **fritz, den Dateibaumspion** geschrieben.

- ▶ Permissions und ACLs korrigieren (`public_html`, Mail)
- ▶ konvertiert Mail-Forwards von `.qmail` nach `.forward` (und `.qmail-*` nach `.forward-*`)
- ▶ identifiziert User mit übergroßer Mailbox oder langer Inaktivität ($> 1,5$ Jahre)
- ▶ warnt bei Annäherung an Quotas
- ▶ ...

Manuelles Testen

Testen von Funktionalität

- ▶ Reboot nach gravierenden Änderungen – startet der Computer noch?
- ▶ Erfüllt der Computer seine Dienste für einen Testbenutzer? Für mich? Für einen Freiwilligen?
- ▶ Arbeitet jemand in *regelmäßigen Abständen* eine Checkliste mit GUI-Aspekten ab?

Monitoring um Probleme frühzeitig zu erkennen

- ▶ Clustermonitoring mit Ganglia
- ▶ Überwachung des Systems: mbmon, sensors
- ▶ Überwachung von Festplatten: hddtemp, smartmontools

Automatisches Testen

- ▶ *cfengine* sieht stündlich nach, ob notwendige Services laufen – und startet Services bei Bedarf neu
- ▶ *monit* überwacht Services von innen – einfacher, aber nicht so flexibel wie *cfengine*
- ▶ Überwachung der Services von außen: (bei uns noch nicht implementiert)
 - ▶ *PICA*, *PIKT*, *mon*, ...
 - ▶ *Big Brother*, *Big Sister* und Abkömmlinge
 - ▶ *Nagios*
- ▶ Shellskripte testen Benutzeraspekte: kann man mit einem bestimmten Fortran-Compiler eine einfaches Programm compilieren und bringt das erzeugte Executable das erwartete Ergebnis?
- ▶ eigene Shell-Skripte zum Testen von IMAP und POP3 Service - viele mögliche Fallen

Dokumentation / Notizen

- ▶ *script* zum **Protokollieren von Shell-Kommandos**
- ▶ *emacs record-mode* als **Tagebuch** mit Inhaltsverzeichnis
- ▶ *emacs wiki-mode* und *freemind* zum **Sammeln von Ideen**
- ▶ *MediaWiki* für die **Benutzerdokumentation**
- ▶ *phpBB* und Email-Listen für **Ankündigungen**
- ▶ *Shellscripts* und Konfiguration für *Cfengine* sind auch Dokumentation!

In Zukunft wollen wir auch automatisches **bug tracking** anbieten: *RT* oder *roundup*.

Aufbau des Netzwerks

- ▶ 2 IP-Subnetze
- ▶ 3 Physikalische Standorte;
 - ▶ Instituträumlichkeiten verbunden mit 1 GBit Glasfaser Verbindung zum
 - ▶ Serverraum
 - ▶ Computerlehrsaal mit 100 MBit GF über einen Router des ZID angebunden

Basisdienste

- ▶ Adresszuweisung erfolgt über DHCP; jeweils zwei Server pro IP Subnetz arbeiten als failover peers
- ▶ Nameservice
 - ▶ extern: DNS über den ZID der TU
 - ▶ intern: aus DNS generierte /etc/hosts, verteilt von cfengine
- ▶ Authentifizierung:
 - ▶ MIT Kerberos auf OpenBSD auf Mini-ITX Rechnern mit CF anstelle von HDD
 - ▶ NIS
 - ▶ Kein LDAP
- ▶ Benutzerverwaltung mit selbstgeschriebenen Programmen & Skripten, z.B. createuser: Kümmert sich um Kerberos, AFS, NIS/LDAP
- ▶ Security:
 - ▶ nessus
 - ▶ iptables

Fileservices - AFS

Seit ungefähr 2001 wird primär AFS zur Speicherung von Benutzerdaten verwendet:

- ▶ Vorteile:
 - ▶ globaler Namespace
 - ▶ Client für fast alle gängigen Plattformen erhältlich
 - ▶ Daten in sogenannten *Volumes* gespeichert; leicht managebar
 - ▶ verschlüsselter Dateitransfer
 - ▶ ACL
 - ▶ sehr gute Skalierbarkeit
- ▶ Nachteile
 - ▶ steile Lernkurve für die Administrierung
 - ▶ ACL nur auf ein Verzeichnis setzbar
 - ▶ Lebensdauerbeschränkung von Kerberos Tickets bzw. AFS Tokens

AFS Server

AFS-Server Prozesse:

- ▶ bos
- ▶ Fileserver fs
 - ▶ fileserver
 - ▶ volserver
 - ▶ salvager
- ▶ Datenbankserver
 - ▶ buserver
 - ▶ ptserver
 - ▶ vlserver

Bei uns eingesetzte Server

- ▶ 3 Fileserver die auch als Datenbankserver fungieren
- ▶ 2 Fileserver, einer davon temporär

AFS Token - der Passierschein

Für den Zugriff auf AFS Volumes wird (noch) ein *AFS Token* benötigt; es wird unter Verwendung eines gültigen *Kerberos V Tickets* vom Kerberos Server unter Verwendung von *aklog* generiert.

Probleme:

- ▶ begrenzte Laufzeit des Kerberos Tickets und, damit verbunden, des AFS Tokens
- ▶ passwortfreies remote login via OpenSSH 3.8x
- ▶ kscreensaver

Lösung:

- ▶ gepatchtes libpam-krb5
- ▶ eigenes libpam-aklog
- ▶ OpenSSH 4.x, gepatchtes OpenSSH 3.8x oder Verzicht auf *privilege separation*
- ▶ kein kscreensaver → xlock, xscreensaver

AFS ACL - Fluch oder Segen?

AFS verwendet *ACL* (Access Control Listen)

- ▶ r: read
- ▶ l: lookup
- ▶ w: write
- ▶ i: insert
- ▶ d: delete
- ▶ k: lock
- ▶ a: administer

Mögliche Mitglieder einer ACL sind Benutzer, Gruppen (System- und Benutzerdefinierte), IPs (bedingt)

Einschränkungen:

- ▶ nur auf Verzeichnisniveau
- ▶ Unix Gruppenrechte werden ignoriert
- ▶ Unix Benutzerrechte wirken nur zusätzlich einschränkend

AFS Backup

AFS Backups werden durch AFS bordeigene Mittel bewerkstelligt;
Möglichkeiten des AFS Backupsystems:

- ▶ per Backup Volume direkter Zugriff auf den letzten Snapshot eines Volumes
- ▶ Wiederherstellung einzelner Volumes, einzelner Partitionen und aller Partitionen eines Servers mit einem Kommando möglich

Wir steuern das Backup über ein selbstgeschriebenes Skript *backup2disk*, das auch Backups von Unix-Dateibäumen und Windows-Shares durchführt.

Tägliches Backup erfolgt auf ein SATA RAID-5; im Moment ist ein Zugriff auf die letzten 3 Monate vorgesehen. DLT-IV Tapes werden zu Archivzwecken unter Verwendung eines Bandroboters geschrieben.

Fileservices - NFS

NFS wird nur für Berechnungen und für Installationen als NFS root eingesetzt.

Wieso noch NFS?

- ▶ Sticky bit auf Verzeichnisse
- ▶ ACLs
- ▶ AFS root nur schwer machbar

Fileservices - diverse

Um die wenigen PCs mit Windows und MacOS 9 mit Datei (und Druckdiensten) zu versorgen, setzen wir **Samba** und **Netatalk** ein.

Drucken mit LPRng

Als Drucksystem setzen wir **LPRng** in Verbindung mit dem Druckerfilter **ifhp** ein. Wenn ich nicht gerade die Konfiguration zerstört habe, funktioniert diese Lösung ausgesprochen zuverlässig – und ist mit etwas Übung auch sehr einfach zu betreuen. Praktisch ist, daß für Printserver und Clients dieselbe Konfigurationsdatei `/etc/printcap` verwendet werden kann.

Versuche mit **CUPS** sind bisher nicht besonders zufriedenstellend verlaufen. CUPS druckt zwar und ist in Konfiguration und Benutzung sehr einfach, wir konnten damit aber die Zuverlässigkeit von LPRng und ifhp nicht erreichen. Um die Anbindung der Rechner mit MacOS X zu vereinfachen, haben wir CUPS installiert, leiten aber alle Druckaufträge über LPRng an die Drucker weiter.

Electronic Mail

Ein Teil der Arbeit mit Electronic Mail (Viren- und SPAM-Filter) wird uns vom ZID abgenommen, ein zuverlässiges Mailsystem ist am Institut trotzdem wichtig. Wir haben in den vergangenen Jahren **Qmail** als MTA verwendet, seit dem Umstieg auf Debian nutzen wir **Postfix**. Beide Mailer sind einfach zu konfigurieren, zuverlässig und einfach zu überwachen, die Lizenz von Qmail verhindert aber eine komfortable Weiterentwicklung.

Die Mail wird in eine Datei `Mailbox` im Homedirectory zugestellt. Zur Auslieferung verwenden wir einen leicht modifizierten **uw-imapd** – den IMAP und POP-Daemon der University of Washington.

Zugriffsrechte erhalten IMAP und POP3 über die PAM-Konfiguration, der MTA über eine Kerberos *keytab* Datei, die beim täglichen Start gelesen wird.

Die verwendeten Mail-Clients stellen wir frei, bieten aber nur Support für **Pine**, **Thunderbird** und **Outlook**; verwenden selbst natürlich **Gnus**.

WWW mit Wiki und Forum

Als Webserver dient beinahe klassisch ein **Apache 2** – die statischen Seiten werden mit der **Website Meta Language (WML)** generiert.

Wir verwenden ein **Media Wiki** als Diskussionsplattform vor allem für Studenten und für die Benutzerdokumentation. Entscheidungsgrund war vor allem die Unterstützung von \LaTeX bzw. \TeX -Syntax für mathematische Ausdrücke. 2 Studienassistenten pflegen neben ihren anderen Tätigkeiten Texte im Wiki und üben eine gewisse redaktionelle Aufsicht aus.

phpBB wird vor allem von einer Arbeitsgruppe als geschlossenes Forum zur Kollaboration mit auswärtigen Kollegen genutzt – nebstbei auch als institutsinterne Ankündigungsplattform.

Unsere größten Probleme!

- ▶ Testen!
- ▶ defekte Fesplatten
- ▶ Tests sind unvollständig
- ▶ ausfallende CPU- und Netzteil Lüfter
- ▶ defekte Netzeile
- ▶ Tests sind nie vollständig genug!
- ▶ Wir brauchen mehr automatische Tests!
- ▶ Hardware!

Pläne für die Zukunft

- ▶ Serverkonsolidierung mit besser geeigneter Hardware – verstärkter Einsatz redundanter Netzteile, RAID, größere Racks, mehr USV, ...
- ▶ weiterer Ausbau der Rechenkapazität
- ▶ Umstellen der Installation von *Replicator* auf *FAI*
- ▶ Alternativen zu *Condor* suchen
 - ▶ Sun ONE Grid Engine
 - ▶ Globus Toolkit
 - ▶ Mosix + OpenPBS

Literatur

- ▶ Thomas A. Limoncelli, Christine Hogan . *The Practice of System and Network Administration* . Addison-Wesley . 2002 . ISBN 0201702711
- ▶ Mark Burgess . *Analytical Network and System Administration: Managing Human-Computer Systems* . John Wiley & Sons . 2004 . ISBN 0470861002
- ▶ Jason Garman . *Kerberos: The Definitive Guide* . O'Reilly . 2003 . ISBN 0596004036

Danke für Ihre Aufmerksamkeit!

Andreas Hirczy

<http://itp.tugraz.at/~ahi/>
<mailto:ahi@itp.tugraz.at>

Christian Pfaffel-Janser

<mailto:flash@itp.tugraz.at>

Einige der angesprochenen Programme findet man auf der Webseite <http://itp.tugraz.at/~ahi/computer.html>; an unsere Umgebung angepasste Debian Pakete gibt es unter <http://itp.tugraz.at/Comp/debian/>.