Configuration Engine - effiziente Administration eines Linux/UNIX-Clusters

Andreas Hirczy

TU Graz
Institut für Theoretische Physik – Computational Physics

Grazer Linuxtage 2006
20. Mai. 2006

Vorstellung

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

Vorstellung

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

Der Arbeitschwerpunkt 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.

Vorstellung

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

Der Arbeitschwerpunkt 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 800 Useraccounts, von denen im Regelfall ungefähr 50 gleichzeitig aktiv sind.

Einen Überblick über Anforderungen, installierte Software und Probleme gab es im Vortrag der Grazer Linuxtage 2005 ([PJH05]).

Ziele:

- Benutzer sind mit dem angebotenen Service zufrieden :)
- Alle PCs verhalten sich für alle Benutzer **identisch!**

Ziele:

- Benutzer sind mit dem angebotenen Service zufrieden :)
- Alle PCs verhalten sich für alle Benutzer **identisch!**
- Jede administrative Tätigkeit muss vernünftig **skalieren!**

Ziele:

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

Ziele:

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

Worüber ich heute spreche:

- eine Methode, einen Cluster von Linux und UNIX-Computern mit minimalem Aufwand in Betrieb zu halten
- Nachteile

Philosophie

Divergenz Im Lauf der Zeit entfernen sich die einzelnen Maschinen durch *ad hoc* Änderungen weiter:

- auf jeder Maschine andere Software (Installation nur nach Bedarf)
- Software ist auf unterschiedlichen Release-Ständen

Philosophie

Divergenz Im Lauf der Zeit entfernen sich die einzelnen Maschinen durch *ad hoc* Änderungen weiter:

- auf jeder Maschine andere Software (Installation nur nach Bedarf)
- Software ist auf unterschiedlichen Release-Ständen

Kongruenz Alle Geräte einer **Klasse** werden gleichartig installiert und ständig identisch gehalten.

Philosophie

Divergenz Im Lauf der Zeit entfernen sich die einzelnen Maschinen durch *ad hoc* Änderungen weiter:

- auf jeder Maschine andere Software (Installation nur nach Bedarf)
- Software ist auf unterschiedlichen Release-Ständen

Kongruenz Alle Geräte einer **Klasse** werden gleichartig installiert und ständig identisch gehalten.

Konvergenz Die Konfiguration konvergiert auf eine gemeinsamen *Idealzustand* hin.

manuelle Administration

webmin

yast

linuxconf sind gut für die Arbeit auf einzelnen Geräten geeignet – für größere Installationen viel zu aufwendig

SAM gibt auf Wunsch auch aus, was im Hintergrund passiert – man kann also wenigstens davon lernen und ein Shellskript zusammenstellen

vi, emacs

Alle Installationen mit mehr als 2 Maschinen, die manuell konfiguriert werden, divergieren!

Abhilfe suchen die meisten erfahreneren Admins in **Shell**- oder **Perl-Skripten** – diese sind aber oft schwer portabel zu schreiben und unflexibel.

Cfengine – die Configuration Engine

Cfengine kann automatische Änderungen der Rechnerkonfiguration auf allen Linux- und UNIX-Systemen durchführen. Sie nimmt eine Beschreibung (**policy**) des gewünschten Systemzustandes in einer sehr abstrahierten Sprache entgegen und versucht auf portable und möglichst unauffällige Weise diesen Zustand zu erreichen.

Cfengine – die Configuration Engine

Cfengine kann automatische Änderungen der Rechnerkonfiguration auf allen Linux- und UNIX-Systemen durchführen. Sie nimmt eine Beschreibung (**policy**) 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, Subnet, ...) erkennt *cfengine* selbst; eigene Klassen können einfach definiert werden.

`http://www.cfengine.org/` — [Bur05]

Aufbau

Im folgenden werden nur Eigenschaften von Cfengine in Version 2.x betrachtet.

cfagent interpretiert die Policy, verändert dabei Einträge im Dateisystem, startet und stoppt Prozesse, mounted und unmounted, ...

cfservd stellt die Policy und andere Konfigurationsdateien im Netzwerk bereit.

Aufbau

Im folgenden werden nur Eigenschaften von Cfengine in Version 2.x betrachtet.

cfagent interpretiert die Policy, verändert dabei Einträge im Dateisystem, startet und stoppt Prozesse, mounted und unmounted, ...

cfservd stellt die Policy und andere Konfigurationsdateien im Netzwerk bereit.

cfexecd ermöglicht starten von *cfagent* aus der Ferne.

cfrun triggert *cfexecd*.

cfkey erzeugt Host-Keys für eine sichere Datenübertragung.

Aufbau

Im folgenden werden nur Eigenschaften von Cfengine in Version 2.x betrachtet.

cfagent	interpretiert die Policy, verändert dabei Einträge im Dateisystem, startet und stoppt Prozesse, mounted und unmounted, ...
cfservd	stellt die Policy und andere Konfigurationsdateien im Netzwerk bereit.
cfexecd	ermöglicht starten von *cfagent* aus der Ferne.
cfrun	triggert *cfexecd*.
cfkey	erzeugt Host-Keys für eine sichere Datenübertragung.
cfenvd	sammelt Informationen über den zeitlichen Verlauf des Betriebszustandes.
cfenvgraph	bereitet die Daten des *cfenvd* für eine graphische Ausgabe auf.

Policy

- das Ziel wird beschrieben
- nicht unbedingt eine detaillierte Liste mit Anweisungen

Policy

- das Ziel wird beschrieben
- nicht unbedingt eine detaillierte Liste mit Anweisungen

Gruppen

- vordefinierte Gruppen: cfengine_2_1_14, linux, compiled_on_linux_gnu
- beim Start festgestellte Gruppen: 129_27_161_68, linux_2_6_16_16, debian_3_1
- dynamische Gruppen: HDFULL_ROOT, CONF_CHANGE_SSHD, CONF_CHANGE_NTPD

Ausführen: cfagent -qvK

```
GNU Configuration Engine - 2.1.14
[....]
Defined Classes = ( 129_27_161 129_27_161_68 32_bit CPU_ATHLON
 CPU_MODEL_6 Day17 HAS_AFS HAS_CURRENT_KERNEL HAS_GMOND
 HAS_HW_USB HAS_POSTFIX [...] HAS_SW_MATHEMATICA HAS_SW_NAG
 HAS_SW_VTEX Hr19 Hr19_Q1 LOC_ITP LoadAvg_high_dev1 May Min10
 Min10_15 Q1 Wednesday Yr2006 any cfengine_2 cfengine_2_1
 cfengine_2_1_14 compiled_on_linux_gnu debian debian_3
 debian_3_1 faeppc38 faeppc38_tu_graz_ac_at i686 ipv4_129
 ipv4_129_27 ipv4_129_27_161 ipv4_129_27_161_68 linux
 linux_2_6_16_16 linux_i686 linux_i686_2_6_16_16
 net_iface_eth0 net_iface_lo )

Installable classes = ( no_default_route HDFULL_ROOT
 CONF_CHANGE_NIS MAIL_POSTFIX_CONF CONF_CHANGE_NTPD
 CONF_CHANGE_SSHD str_once_a_day NETWORK_INETD_CONF )
```

Funktionen

- **Steuerung den Ablaufs:** classes/groups, control, import, strategies
- **Manipulationen des Dateisystems:** copy, disable/rename, links, files, tidy, required/disks
- **Netzwerk-Konfiguration:** defaultroute, resolve
- **Ändern von Dateien:** editfiles
- **Mounten externer Resourcen (NFS):** miscmounts, mountables, unmount
- **Prozessmanagement:** processes, shellcommands

Die Online-Hilfe findet man mit „info“:

- cfengine-Reference
- cfengine-Tutorial

Steuerung den Ablaufs

- **control:** ermöglichst das Setzen von Variablen – natürlich kann man dabei schon das Klassenkonzept ausnützen. Unbedingt notwendig ist das Setzen der Bearbeitungsreihenfolge **actionsequence**
- **classes/groups:** erlaubt die Definition eigener Klassen
- **import:** bindet andere Konfigurationsdateien ein
- **strategies:** erzeugt Klassen auf Basis eines zufälligen Ereignisses

Manipulationen des Dateisystems

- **copy:** kopiert Dateien, durch den *cfservd* auch von anderen Maschinen
- **disable/rename:** benennt eine Datei oder ein Directory um, indem „.cfdisabled“ an den Namen gehängt wird – verwendbar um gefährliche Dateien zu entsorgen.
- **links:** erzeugt Links – einzelstehende, auf Wunsch auch ganze Wälder
- **files:** erzwingt die Existenz von Files und setzt bei Bedarf Permissions und Besitzer/Gruppe
- **tidy:** entsorgt Dateien
- **required/disks:** prüft, ob Dateien bzw. Dateisysteme verfügbar sind

Netzwerk-Konfiguration

- **defaultroute:** setzt die defaultroute :)
- **resolve:** erleichtert die Konfiguration der Datei „/etc/resolv.conf“

Ändern von Dateien

Die Anweisung **editfiles** ermöglicht vielfältige Manipulationen von (Text-)Dateien – dabei können auch reguläre Ausdrücke genutzt werden.

Ich werde später einige einfache Beispiele bringen, aber der vollständige Funktionsumfang kann hier leider unmöglich präsentiert werden.

Mounten externer Resourcen (NFS)

- **miscmounts:** definiert Mountpoints (ein Beispiel folgt in Kürze)
- **mountables:** definiert spezielle Mountpoints für Homedirectories oder Server von Programmen, die je nach Clientarchitektur verwendet oder ignoriert werden.
- **unmount:** unmount :)

Prozessmanagement

- **processes:** testet, ob Prozesse aktiv sind, die einem bestimmten Muster entsprechen – schickt an dies Prozesse entweder beliebige Signale oder aktiviert gegebenenfalls einen Restart.
- **shellcommands:** werden ausgeführt – das ist im Wesentlichen ein zentralisiertes und einheitliches Interface zu „cron“.

meine Organisation

- Policy und Konfigurationsdateien werden in einem Subversion-Archiv verwaltet und in meinem Homedirectory bearbeitet - damit ist die gleichzeitige Arbeit mehrerer Admins einigermaßen problemlos möglich.
- Für jedes Fachgebiet benutze ich eine eigene Datei, die von „cfagent.conf“ importiert wird: cf.accounts.conf, cf.network.conf, cf.software.linux.conf, cf.software.hp-ux.conf, ...
- Dateien werden von *rsync* in ein Repository kopiert und von dort stündliche über *cfagent* und *cfservd* auf die Maschinen im Netz verteilt.

Bereitstellen: cfservd.conf

```
groups:
   ConfServer = ( faeppc14 )

control:
   domain                = ( tu-graz.ac.at )
   AllowConnectionsFrom = ( 129.27.161  129.27.157 )
   TrustKeysFrom         = ( 129.27.161  129.27.157 )
   MaxConnections        = ( 50 )
   MultipleConnections   = ( true )
   redhat::
      cfrunCommand       = ( "/var/cfengine/bin/cfagent" )
   debian::
      cfrunCommand       = ( "/usr/sbin/cfagent" )

grant:
   ConfServer::
      /root/cfengine2          129.27.161.*  129.27.157.*
      /root/config             129.27.161.*  129.27.157.*
```

Abholen: update.conf

```
control:
   actionsequence  = ( copy tidy )
   domain          = ( tu-graz.ac.at )
   policyhost      = ( faeppc14.tu-graz.ac.at )
   master_cfinput  = ( /root/cfengine2 )
   cf_install_dir  = ( /usr/sbin )
   SplayTime       = ( 10 )
   redhat::
      workdir         = ( /var/cfengine )
   debian::
      workdir         = ( /var/lib/cfengine2 )

copy:
   debian_3_1::
     $(master_cfinput)           dest=/etc/cfengine
                                 r=inf
                                 mode=700
                                 type=binary
                                 exclude=*.lst
                                 exclude=*~
                                 exclude=#*
                                 server=$(policyhost)
                                 trustkey=true
```

Einnisten: Auschnitte aus cfagent.conf

```
editfiles:
   redhat.!NOTEBOOK::
      { /etc/crontab
         AppendIfNoSuchLine "00 * * * * root /var/cfengine/bin/cfag
      }
   debian.!NOTEBOOK::
      { /etc/crontab
         DeleteLinesContaining "/var/lib/cfengine/bin/cfagent"
         AppendIfNoSuchLine "00 * * * * root /var/lib/cfengine2/bin
      }

processes:
   redhat::
      " cfservd " restart "/var/cfengine/bin/cfservd"
   debian::
      " cfservd " restart "/var/lib/cfengine2/bin/cfservd"
```

Mounten, links und permissions setzen, aufräumen

```
miscmounts:
  !NOTEBOOK.!SERVER::
    faepsv03:/itp/faepsv03/temp /itp/faepsv03/temp mode=intr

links:
  !NOTEBOOK.!SERVER::
    /temp       ->! /itp/faepsv03/temp

files:
  faepsv03::
    /itp/faepsv03/temp          r=0     mode=1777 o=root g=root  act=fixall

tidy:
  faepsv03.Saturday.Hr21::
    /itp/faepsv03/temp/    pattern=*  recurse=inf  age=365


unmount:
    faeppc03:/itp/faeppc03/temp deletedir=true deletefstab=true force=true
  SERVER::
    faepsv03:/itp/faepsv03/temp deletedir=true deletefstab=true force=true
```

Klassen definieren, shell commands zur RAID-Überwachung

```
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 c0"
    "/usr/local/sbin/tw_cli info c1"

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

editfiles:
  HAS_RAID_SW::
    { /etc/sysctl.conf
      AppendIfNoSuchLine  "dev.raid.speed_limit_min = 1000"
      AppendIfNoSuchLine  "dev.raid.speed_limit_max = 50000"
      DeleteLinesMatching "dev.raid.speed_limit_max = 20000"
      DeleteLinesMatching "dev.raid.speed_limit_max = 100000"
    }
```

Konfiguration von SSH und SSHD

```
editfiles:
  linux::
    { /etc/ssh/sshd_config
      DefineClasses "CONF_CHANGE_SSHD"
      DeleteLinesMatching "#.*"
      DeleteLinesMatching "^$"

      DeleteLinesMatching "UsePAM no"
      AppendIfNoSuchLine  "UsePAM yes"
      DeleteLinesMatching "Protocol .,."
      DeleteLinesMatching "Protocol 1"
      AppendIfNoSuchLine  "Protocol 2"
      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"

processes:
  redhat::
    "sshd"  restart "/etc/init.d/sshd restart && /sbin/chkconfig --add sshd"
  debian::
    "sshd"  restart "/etc/init.d/ssh-krb5 restart"
```

Probleme und Nachteile

- Aufwand für Einarbeitung
- Aufwand zum Aufbau der Infrastruktur (5 Minuten beim zweiten Mal)

Probleme und Nachteile

- Aufwand für Einarbeitung
- Aufwand zum Aufbau der Infrastruktur (5 Minuten beim zweiten Mal)
- Wann sollen Laptops im Betrieb automatisch konfiguriert werden?
  - Beim System-Start ist meist kein Netz verfügbar.
  - Bei nächtlichen Cron-Jobs ist das Gerät nicht aktiv.
  - Seltene, dafür aufwändigere Läufe während der Arbeit stören den Benutzer.
  - Benutzergesteuert – aber mit Hintertüren!

Probleme und Nachteile

- Aufwand für Einarbeitung
- Aufwand zum Aufbau der Infrastruktur (5 Minuten beim zweiten Mal)
- Wann sollen Laptops im Betrieb automatisch konfiguriert werden?
  - Beim System-Start ist meist kein Netz verfügbar.
  - Bei nächtlichen Cron-Jobs ist das Gerät nicht aktiv.
  - Seltene, dafür aufwändigere Läufe während der Arbeit stören den Benutzer.
  - Benutzergesteuert – aber mit Hintertüren!
- Man kann sich in kürzester Zeit alles selbst zerstören!
  Self destruct in 5 seconds — have a nice day!

Template Tree II

Template Tree II (`http://isg.ee.ethz.ch/tools/tetre2/`) arbeitet als Präprozesor, der Konfigurationseinträge aus einer großen Zahl kleiner, modularer Dateien in Anweisungen für die cfengine transformiert und diese im Format POD gleichzeitig dokumentiert.

Nach meinem Gefühl lohnt sich diese Lösung erst, wenn man eine größere Anzahl an Admins (> 2) gleichzeitg an der Konfiguration arbeiten läßt.

isconf – Infrastructure Architecture

Den Anspruch vollständiger Kongruenz verfolgt *isconf* – man geht davon aus, daß alle Computer innerhalb einer Domäne stets gleich installiert werden und alle Änderungen auf allen Computern in exakt derselben Reihenfolge ([TH98] [TB02]) durchgeführt werden.

isconf funktioniert in der aktuellen Version 4 ähnlich einem verteiltem Versionskontrollsystem, in das man manuelle Konfigurationsänderungen eincheckt, die später von den anderen Maschinen zeitgesteuert nachvollzogen werden können.

`http://www.infrastructures.org/` `http://www.isconf.org/`

- Ist die Hardware wirklich ähnlich genug?
- Testaufwand!
- Wesentlich unflexibler als Cfengine.
- Leicht defekte Computer werden von Grund auf neu installiert.

lcfg – Local ConFiGuration system

lcfg scheint von der Philosophie der *cfengine* sehr ähnlich zu sein, hat wegen eines Präcompilers das Potential zu besserer Laufzeiteffizienz – ist aber nicht so flexibel und umfangreich.

lcfg wird im **European Data Grid** verwendet, sollte also durchaus eine weite Verbreitung finden.

`http://www.lcfg.org/`

Danke für Ihre Aufmerksamkeit!

Andreas Hirczy

http://itp.tugraz.at/~ahi/

mailto:ahi@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/.

Fragen?

Literatur

Burgess, Mark:
A Tiny Overview of Cfengine: Convergent Maintenance Agent.
In: *Proceedings of the 1st International Workshop on Multi-Agent and Robotic Systems*
MARS/ICINCO, 2005. –
http://www.iu.hio.no/~mark/papers/tiny_intro.pdf

Pfaffel-Janser, Christian ; Hirczy, Andreas.
Linux in einer mittelgroßen Institution.
Vortrag Grazer Linuxtage.
2005

Traugott, Steve ; Brown, Lance:
Why Order Matters: Turing Equivalence in Automated Systems Administration.
In: *Proceedings of LISA 2002: 16th Systems Administrators Conference*.
Berkeley, CA, 2002. –
http://www.usenix.org/publications/library/proceedings/lisa98/traugott.html,
S. 99

Traugott, Steve ; Huddleston, Joel:
Bootstrapping an Infrastructure.
In: *Proceedings of the Twelfth Systems Administrators Conference*.
Berkeley, CA, 1998. –
http://www.infrastructures.org/papers/bootstrap/bootstrap.html, S. 181