

Linux Cluster in Theorie und Praxis

Environment Modules, Compiler, Backup und Klonen

09. November 2009

INF 1046
Nöthnitzer Straße 46
01187 Dresden
0351 - 463 38783

Oliver Knodel & Stefan Höhlig

Verfügbarkeit der Folien

Vorlesungswebseite:

http://tu-dresden.de/die_tu_dresden/zentrale_einrichtungen/zih/lehre/ws0910/lctp

Inhalt

- 1 Environment Modules
 - Motivation
 - Das module Kommando
 - Beispiel modulefile
- 2 Compiler
 - Aufbau
 - Arten von Compilern
 - Beispiele aktueller Compiler
- 3 Backup
 - Sicherungsarten
 - Überblick Tools und Methoden
- 4 Klonen
 - Vorbereitung
 - Methoden
- 5 Quellen

Teil 1 - Modules und Compiler

Environment Modules

Probleme

Auf einem Cluster arbeiten meist viele Nutzer an unterschiedlichen Projekten und verwenden dabei Software und Bibliotheken die sich mitunter nur in der Version unterscheiden.

Wie vermeidet man dadurch entstehende Konflikte?

Wie erleichtert man den Nutzern den Zugriff auf für sie nötige Software?

Lösung

Eine saubere Installation, lokale Trennung und durchdachte Nomenklatur helfen hierbei nur bedingt. Einen mächtigen und flexiblen Ansatz bietet hierfür die Software **Environment Modules**.

- Saubere und einfache Lösung zur Änderung der Umgebungsvariablen zur Laufzeit
- Unterstützt alle verbreiteten Shells (bash, ksh, zsh, sh, csh, tcsh...)
- Unterstützung für Script-Sprachen wie z.B. Perl
- Basis bilden sog. `modulefiles` zur Shell-Konfiguration
- Benutzt `eval` zum Ausführen von Kommandos

- **Aktuelle Version:** 3.2.7 (Juli 2009)
- **URL:** <http://modules.sourceforge.net/>
- **Sourcen:** TCL (experimental), C
- TCL/TCL-dev benötigt
- Fester Bestandteil der *Open Source Cluster Application Resource (OSCAR)*

- Änderung von nutzerspezifischen Umgebungsvariablen
- Konfliktvermeidung beim Verwenden mehrerer Versionen der gleichen Applikation
- Aufbau einheitlicher Umgebungen auf unterschiedlichen Systemen
- Suite-Bildung mit Hilfe von Meta-Modulen

Ausgewählte Parameter für module I

- **add** | **load** MODFILE [MODFILE...]
 - Fügt der Umgebung ein oder mehrere modulefiles hinzu
 - Textausgabe nur bei Fehlern (muss manuell eintragen)
 - Bei mehreren verfügbaren Versionen kann generisch ein "default" geladen werden
- **rm** | **unload** MODFILE [MODFILE...]
 - Entfernt ein oder mehrere modulefiles aus der Umgebung
 - Die Reihenfolge spielt dabei keine Rolle
- **switch** OLD NEW
 - Austausch zweier modulefiles (z.B. bei Versionswechsel)
 - Wird nur ein modulefile angegeben, so wird dessen Basisname zur Identifizierung für das aktuell geladene modulfile verwendet
- **display** | **show** MODFILE [MODFILE...]
 - Zeigt Änderungen an der Umgebung durch das modulefile

Ausgewählte Parameter für module II

- **av** | **avail**
 - Auflistung aller verfügbaren modulefiles
- **list**
 - Zeigt alle aktuell geladenen modulefiles
- **initadd** MODFILE [MODFILE...]
 - Hängt ein oder mehrere modulefiles an das Shell-Startup-Script
- **initprepend** MODFILE [MODFILE...]
 - Fügt ein oder mehrere modulefiles zu Beginn des Shell-Startup-Scripts ein
- **help** [MODFILE]
 - Zeigt Hilfe zu entsprechendem MODFILE
 - Auflistung verfügbarer Kommandos wenn kein modulfile übergeben wurde

Beispiel modulefile

- In TCL geschriebene Skripte
- <http://en.wikibooks.org/wiki/Programming:Tcl>
- https://www.bg.bib.de/portale/bes/pdf/Einfuehrung_Tcl.pdf
- Dokumentation in 'man 4 modulefile'

```
1 #%Module1.0#
2 ##
3 ## intel-11.0.069 modulefile
4 ##
5 proc ModulesHelp { } {
6     puts stderr "\tIntel C/C++/Fortran Compiler 11.0.069"
7 }
8
9 module-whatIs    "Intel C/C++/Fortran Compiler 11.0.069"
10
11 set pkgroot_c /home/ich/compiler/intel
12
13 # C/C++ & Fortran
14 prepend-path PATH $pkgroot_c/Compiler/11.0/069/bin/intel64
15 prepend-path LD_LIBRARY_PATH $pkgroot_c/Compiler/11.0/069/lib/intel64
16 prepend-path MANPATH $pkgroot_c/Compiler/11.0/069/man
17 prepend-path INTEL_LICENSE_FILE $pkgroot_c/licenses
18
19 if [ module-info mode load ] {
20     puts stderr "Intel C/C++/Fortran Compiler 11.0.069 was loaded"
21 }
```

Teil 1 - Modules und Compiler

Compiler

Definition: Compiler

Ein Compiler überführt ein Programm in einer höhere Programmiersprache (Quellsprache) in ein äquivalentes Programm einer anderen Sprache (Zielsprache), z.B. in ein Maschinenprogramm. Eine wichtige Teilaufgabe eines Compilers besteht darin, dem Benutzer Fehler, die im Quellprogramm enthalten sind, zu melden. - Taschenbuch der Informatik S.271

- Heute eine Sammlung von mehreren Programmen, da mehrere Phasen nötig
 - Preprozessor, (mehrere) Compiler, Assembler, Linker

Compiler Frontend:

- Aufbau eines Syntaxbaumes aus dem Quelltext
- Phasen:
 - **Lexikalische Analyse:** Erkennung der Symbole des Quelltextes (Schlüsselwörter, Ziffern, Buchstaben...) durch den **Lexer**
 - **Syntaktische Analyse:** Bildung von Sätzen der Quellgrammatik aus den Symbolen durch den **Parser**
 - **Semantische Analyse:** Analyse der Bedeutung der gebildeten Sätze (Korrektheit)

Compiler Backend:

- Baut aus dem Syntaxbaum den Code der Zielsprache
- Phasen:
 - **Zwischencodeerzeugung:** relativadressierte maschinennahe Sprache wird erzeugt
 - **Programmoptimierung** (optional)
 - **Codeerzeugung:**
 - Entweder direkt aus Syntaxbaum oder aus Zwischensprache
 - Kann entweder ausführbaren oder Objekt-Code erzeugen

- **Native Compiler:** Erzeugt Code für eigene Plattform
- **Cross Compiler:** Erzeugt Code für andere Plattform
- **Single-Pass Compiler:** Nur ein Durchlauf zur Programmerzeugung
- **Multi-Pass Compiler:** Mehrere Durchläufe zur Programmerzeugung

Lader und Binder (linker)

Der Lader wandelt die verschiebbaren Adressen des Maschinencodes in Absolutadressen um und lädt das Programm an eine geeignete Stelle im Speicher. Der Binder ermöglicht es, verschiedene Dateien, die jeweils verschiebbaren Maschinencode enthalten, zu einem Programm zusammenzufassen. Diese Dateien können aus mehreren getrennten Übersetzungen stammen oder aber vom System bereitgestellte Bibliotheken sein. - Taschenbuch der Informatik S.272

Statisches Linken:

- Vollständiges Auflösen aller Symbole zur Linkzeit
- Einbinden aller verwendeten Bibliotheksfunktionen in das Programm
- Pro:
 - Keine zusätzlichen Bibliotheken beim Anwender nötig
 - Versionskonflikte zwischen Anwendung und Bibliothek unmöglich
- Kontra:
 - Redundanzbildung verwendeter Funktion in unterschiedlichen Programmen
 - Erhöhter Speicherbedarf

Dynamisches Linken:

- Nur teilweises Auflösen der Symbole zur Linkzeit
- Laden und Binden der benötigten Bibliotheksfunktionen zum Programmstart bzw. zur Laufzeit
- Pro:
 - Keine Redundanzbildung
 - Verringerter Speicherbedarf
 - Leichtere Aktualisierung gemeinsam genutzter Funktionen
- Kontra:
 - Verlängerter Programmstart
 - Evtl. verlangsamte Ausführung
 - Anwender benötigt evtl. zusätzliche Bibliotheken

- Am 22 März 1987 erschien erste Version 0.9 Beta
- **Aktuelle Version:** 4.4.2
- **Sprachen:** C/C++, Object-C, Fortran(77/90/95/2003), Java, Ada
- **Plattformen:** Nahezu alle
- **Features:** SIMD, OpenMP, -march=atom Flag ab Version 4.4
- **URL:** <http://gcc.gnu.org/>
- **Kosten:** Open Source

Open64 Compiler

- Ursprünglich von SGI entworfen
- Von Intel, Pathscale und diversen Universitäten weiterentwickelt
- Basis vieler aktueller Compiler
- **Aktuelle Version:** 4.2.1
- **Sprachen:** C/C++, Fortran(77/9x)
- **Plattformen:** x86, x86_64, IA64, MIPS, ARM
- **Betriebssysteme:** Linux (Sourcen verfügbar)
- **Features:** OpenMP, von NVIDIA für CUDA benutzt
- **URL:** <http://www.open64.net/>
- **Kosten:** Open Source

Intel Compiler - ICC/IFORT

- **Aktuelle Version:** 11.1.059
- **Sprachen:** C/C++, Fortran (77/90/95/2003)
- **Plattformen:** x86, x86_64, früher auch XScale
- **Betriebssysteme:** Linux, Windows, Mac OS X
- **Features:** SIMD, OpenMP, spezielle Atom-Version verfügbar
- **URL:** <http://software.intel.com/en-us/intel-compilers/>
- **Kosten:** Kostenfrei für nichtkommerzielle Projekte

Pathscale Compiler Suite

- **Aktuelle Version:** 2.3.99
- **Sprachen:** C/C++, Fortran9x
- **Plattformen:** x86, x86_64 (optimiert)
- **Betriebssysteme:** Linux
- **Features:** SIMD, OpenMP 2.5, optimierte ACML
- **URL:** <http://www.pathscale.com/>
- **Kosten:** 1.995 \$ (Trial Version verfügbar)

The Portland Group (PGI)

- **Aktuelle Version:** 9.0
- **Sprachen:** C/C++, Fortran
- **Plattformen:** x86, x86_64
- **Betriebssysteme:** Linux, MacOS, Windows
- **Features:** SIMD, OpenMP, CUDA toolchain enthalten
- **URL:** <http://www.pgroup.com/>
- **Kosten:** Akademische Lizenz für 699 \$

- 1 Environment Modules
 - Motivation
 - Das module Kommando
 - Beispiel modulefile
- 2 Compiler
 - Aufbau
 - Arten von Compilern
 - Beispiele aktueller Compiler
- 3 Backup
 - Sicherungsarten
 - Überblick Tools und Methoden
- 4 Klonen
 - Vorbereitung
 - Methoden
- 5 Quellen

Teil 2 - Backup und Klonen

Backup

Einführung

Definition: Backup

Datenverlust vorzubeugen erfordert vom Benutzer u. a. inkrementelle oder vollständige Sicherung von Dateien, Verzeichnissen oder ganzen Datenträgern auf ein alternatives (häufig transportables) Speichermedium.

Backups sind notwendig um ein System komplett wieder her zu stellen oder um letzte Änderungen an einzelnen Dateien rückgängig zu machen. -

Taschenbuch der Informatik S.200

Sicherungsarten I

- Vollständiges Backup
 - Bei einem vollständigen Backup werden die zu sichernden Daten stets komplett neu auf das Sicherungsmedium geschrieben
- Inkrementelles Backup
 - Bei einem inkrementellen Backup werden alle Veränderungen seit dem letzten vollständigen oder inkrementellen Backup auf das Sicherungsmedium geschrieben
 - Zu Beginn jedes inkrementellen Backups muss ein vollständiges Backup durchgeführt werden
 - Vorteil: Geringerer Speicherbedarf
 - Nachteil: Zum Wiederherstellen des letzten Systemzustands sind alle inkrementellen und das vollständige Backup notwendig
- Differenzielles Backup
 - Alle Veränderungen seit dem letzten vollständigen Backup werden gesichert

Sicherungsarten II

- Vorteil: Zum Wiederherstellen des Systems nur vollständiges und letztes differentielles Backup notwendig
- Nachteil: Höherer Speicherbedarf
- Netzwerk Backup
 - Bei einem Netzwerk Backup werden die Daten von einem Client auf einem Backup-Server gesichert, wobei die gesamte Kommunikation über das Netzwerk erfolgt
- Dump Backup
 - Bei Dump Backups werden vollständige Festplattenpartitionen oder das Dateisystem auf das Sicherungsmedium übertragen
 - Demzufolge ist es notwendig diese auch wieder im Ganzen herzustellen
 - Das Dump Backup ist ein komplettes Dateisystem, was bedeutet das es auf einer gleich großen Partition wieder hergestellt werden muss

Zu sichernde Daten

- Nur bestimmte Dateien
- Nur bestimmte Verzeichnisse
- Vollständige Partition
- Komplette Festplatte
- Exclude/Include bestimmter Dateien

- Im Praktikum: Komplettes Laufwerk und anschließend lediglich die Änderungen

Zeitpunkt

- **Häufigkeit** und **Zeitpunkt** sind festzulegen
 - Nach Änderungen
 - Zu festen Terminen (cron)
 - Bei geringer Systemlast
- Im Praktikum:
 - **Head-Node:** Täglich
 - **Compute-Node:** Vollständiges Backup vor dem Klonen ausreichend
- **Wiederherstellung**
 - Einzelne Dateien und Ordner
 - Das gesamte Dateisystem

Tools – dd

dd

Data Dump dient zum Umleiten von Datenströmen und kann so beliebiges Dateisystem einfach kopieren. *dd* erstellt ein bitweise vollständiges Backup des Eingangsdevices.

- Bei der Wiederherstellung eines dd-Images muss das Ziel die exakt gleiche Geometrie besitzen
- **Funktionsweise:** `dd if=<input file/device> of=<outputfile/device> bs=<blocksize>`
- Auch mit Kompression möglich: `dd if=<input file/device> | gzip > image.dd`
- Für eine Wiederherstellung sind Inputfile und Outputfile umzukehren (gegebenenfalls muss mit gzip wieder entpackt werden)

Tools – Rsync

- Dient zur unidirektionalen Synchronisation von Daten
- Netzwerk Backup möglich
- Standardmäßig wird der TCP-Port 873 zur Kommunikation verwendet
- Nur Änderungen werden übertragen also u.U. effizienter als dd bei regelmäßigen Backups
- Inkrementelle Backups mit Wiederherstellung einzelner Dateien möglich
- Kopiert keine Partitionstabelle oder den Bootloader

Tools – dump/restore

- Gebräuchlichste und älteste Möglichkeit unter Linux Backups zu erstellen
- Inkrementelles Verfahren
- dump erstellt eine Liste der Daten die seit dem letzten Dump geändert wurden
- Partitionen müssen einzeln gedumpt werden
- Mit restore können einzelne Dateien und auch das gesamte Dateisystem wiederhergestellt werden

Tools – rsnapshot

- <http://www.rsnapshot.org/>
- Platzsparende inkrementelle Backups unter Einsatz von Hardlinks
- Basiert auf rsync

Tools – duplicity

- <http://www.nongnu.org/duplicity/>
- Backup mit verschlüsselten tar-Dateien (mittels GnuPG)
- Sicher bei einem Backup, welches auf einem Fileserver gelagert wird
- Inkrementelles Backup

Teil 2 - Backup und Klonen

Klonen

Methoden

- dd
 - Punkt-zu-Punkt Verbindung über externe Festplatte
 - Vollständige Platte samt Bootloader und swap-Partition wird übertragen
- rsync
 - Partitionen, Dateisystem und Bootloader müssen separat erstellt werden

Vorbereitung

- Wechsel zwischen den Nodes ohne Nutzerinteraktion
- StrictHostKeyChecking muss deaktiviert werden
- Update aller installierten Pakete
- In *udev* alle MAC-Adressen entfernen
- Hostname über DHCP beziehen




Images über das Netzwerk verteilen

- udpcast
 - Multicast des Images mit beliebig vielen Empfängern
 - Dauer des Klonvorgangs ist unabhängig von der Anzahl der Nodes
 - Head-Node sendet PXE-Image (Preboot eXecution) von dem die Work-Nodes booten
 - Notwendig ist ein TFTP-Server und zusätzliche Einträge beim DHCP-Server auf dem Head-Node
 - <http://udpcast.linux.lu/source.html>

Quellen I

-  [EM07] Environment Modules
Environment Modules auf Sourceforge, 2007
<http://modules.sourceforge.net/>
-  [TB04] Uwe Schneider, Dieter Werner
Taschenbuch der Informatik, 2004
<http://www.debianhelp.co.uk/backup.htm>
-  [DL05] Wladislaw Eckhardt
Konzepte von Betriebssystem-Komponenten, Programmstart &
dynamische Bibliotheken , 2005
http://www4.informatik.uni-erlangen.de/Lehre/SS05/PS_KVBK/talks/Eckhardt_Handout.pdf
-  [DH09] Debian Help
Debian Linux Backup, 2009
<http://www.debianhelp.co.uk/backup.htm>

Quellen II

-  [LUGL09] Linux User Group Luxembourg
UDPCast, 2009 <http://udpcast.linux.lu/source.html>
-  [DT09] duplicity-team
duplicity, 2009 <http://www.nongnu.org/duplicity/>
-  [NRDC09] Nathan Rosenquist, David Cantrell
rsnapshot, 2009 <http://www.rsnapshot.org/>