TECHNISCHE UNIVERSITÄT DRESDEN

Center for Information Services and High Performance Computing (ZIH)

Linux Cluster in Theorie und Praxis

Betriebssysteme

19.10.2009

Nöthnitzer Straße 46
01187 Dresden
INF 1038
+49 351 - 463 38781

Robin Geyer

ZIH
Center for Information Services & High Performance Computing

Foliensatz

Verfügbarkeit der Folien

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

Inhalt

1. Einführung
   - Literatur
   - Motivation
   - Top 500

Literatur

- Linux - Wegweiser zur Installation Konfiguration
  - http://www.oreilly.de/german/freebooks/rlinux3ger/linux_wegIVZ.html
- Linux Praxishandbuch
  - http://www.oreilly.de/german/freebooks/runux5ger/
- Linux - Wegweiser für Netzwerker
  - http://www.oreilly.de/german/freebooks/linag2/inhalt.htm
- SuSE 9.3 Administrations Handbuch
  - http://www.novell.com/de-de/documentation/suse93/pdfdoc/admin93-screen/admin93-screen.pdf
- Galileo Computing - Linux
  - http://download.galileo-press.de/openbook/linux/galileocomputing_linux.zip
- Galileo Computing - Ubuntu GNU/Linux
  - http://download.galileo-press.de/openbook/ubuntu/galileocomputing_ubuntu.zip
- Galileo Computing - Wie werde ich UNIX-Guru
  - http://download.galileo-press.de/openbook/unix_guru/galileocomputing_unix_guru.zip

Wozu ein Betriebsystem?

- Jeder Cluster Node ist ein eigenständiger Rechner welcher, in irgend einer Weise, ein Betriebsystem braucht.
  - Eigenständige Installation
  - Netboot
- Prinzipiell aber auch ohne realisierbar
  - Cray T3D/E mit Front-End unter UNICOS und Binarys mit UNICOS/MAX direkt auf Processing Elements ausgeführt

Ein Blick in die Top 500

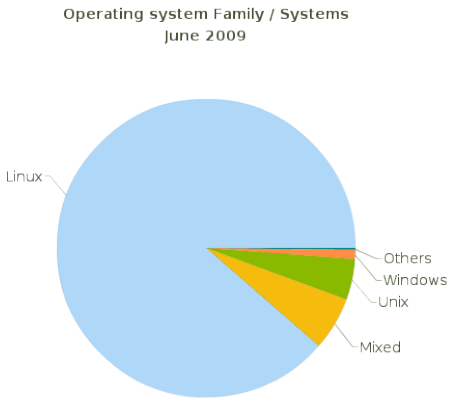


Abbildung: 88,6% Linux, 1% Windows

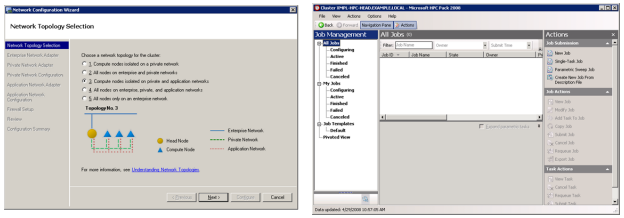
2. Verschiedene Systeme
   - Windows HPC
   - Linux

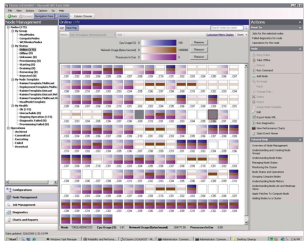
Windows HPC I

- 2006 mit Windows Compute Cluster Server 2003 (CCS) eingeführt
- Mindestens ein Headnode und mehrere sog. “Processing Nodes” die rechnen
- Eigenes MPI v2 (Microsoft Messaging Passing Interface)
- “Job Launcher” als Batchsystem
- Großer Memory Footprint (in jedem Fall größer 1GB)
- Auf jedem Compute Node läuft Windows Desktop
- Microsoft Windows Server Core als Alternative für Nodes (ohne Desktop Umgebung), aber dann nur WinAPI nutzbar
- Gravierende Schwierigkeiten beim kompilieren wissenschaftlicher Anwendungen

Windows HPC II

Screenshots: o.l.: Interconnect Wizzard, o.r.: Job Scheduler, u.: Monitoring





Linux I

- Erfüllt alle Anforderungen die an Cluster Betriebsysteme gestellt werden:
  - Netzwerkfähig
  - Sicherheitskonzept
  - Niedrige Kosten pro Node
  - Skalierbar bis weit über 1M CPUs
  - Wenig Overhead (insbesondere für Nodes)
  - Sehr flexibel
  - Wartung tausender Nodes möglich
  - Großteil der wissenschaftlichen Anwendugen ist für GNU/Linux und GNU/Utils geschrieben
- Große Erfahrung in der HPC Community bzgl. UNIX/Linux
- 88% Usage in Top500

Linux II

Distributionen / andere Unixe

Auswahl des Betriebsystems richtet sich nach zu verwendender Hardware/Software, nicht umgekehrt!

- Auf Treibersupport durch den Hersteller achten
- Bei Ethernet Technologien als Verbindungsnetz kann beliebige Distribution verwendet werden[1]
- Vorsicht bei Quadrics, Myrinet und speziellen DMA basierten Kernelmodulen/Karten
  - Meist nur für SLES und RHES zertifiziert
  - Unmöglich bis sehr schwierig auf anderen Systemen/Distributionen zu benutzen

[1] Falls andere Rahmenbedingungen die Distribution nicht ausschließen

Distributionen I

Grobe Unterscheidung

Unterscheidung 1: Zwischen Enterprise Distributionen (SuSE Enterprise, Red Hat Enterprise) und Freien
Unterscheidung 2: Geschichtliche Herkunft (Debian vs. Slackware linux (SuSE) vs. Red Hat vs. Sonstige) und damit meistens zwischen RPM, Source und deb basierten Paketmanager
Weitere Unterteilungen sind möglich.

Einige für Cluster häufig eingesetzten Distributionen sind (Reihenfolge ohne Wertung):

- **SuSE Linux Enterprise Server (SLES)** und **Red Hat Enterprise Linux (RHEL)**
  - Klarer Vorteil für große Produktivsysteme (Haftungsfrage)
  - Hersteller bieten nur Support auf zertifiziertes Betriebsystem

Distributionen II

- **Debian**
  - Urdistribution, große Anzahl fertiger Pakete
  - Sehr flexibel
  - deb basiert
- **OpenSuSE**
  - Offene Variante von SLES
  - Weniger flexibel, aber einfacher zu bedienen
  - RPM basiert
- **Ubuntu/Kubuntu/Xubuntu (Server)**
  - Freie aber kommerziell betreute Debian Variante
  - Im Prinzip wie Debian, ein wenig unflexibler
- **Gentoo**
  - Sourcebasiert
  - Hoch flexibel, schwerer zu beherschen

Distributionen III

- **Scientific Linux**
  - Gern von Physikern verwendet, am CERN und Fermilab entwickelt
  - Kompatibel zu RHEL (rebuild)
- **Fedora**
  - Freie Red Hat Community Variante
  - Schnelle Release Zyklen, teilweise etwas instabil

Entscheidung für das Praktikum

Debian (lenny)

Gründe:

- Universell auf PPC (Playstation) und Atom einsetzbar
- Verbreitet für kleinere Cluster
- Sehr flexibel für Lehrbetrieb
- Kostenlos
- Keine Anforderungen von Seiten des Verbindungsnetzes (Ethernet)
- Stellt (unserer Meinung nach) einen guten Querschnitt durch alle Distributionen dar

Andere Distributionen verwenden

Wenn eine Gruppe glaubhaft versichern kann das sie eine andere Distribution so gut beherrscht das es **keine Probleme** gibt, kann sie **auf eigene Gefahr**[a] auch diese benutzen.

[a] bei Problemen ohne Hilfe von Seiten der Betreuer und damit evt. Zensurverschlechterung

Genau überlegen!

Besonderheiten bei Debian I

- Deb basiert
  Grundfunktionalität:
  - `apt-get install/remove <package>` für Installieren/Deinstallieren von Software (wahlweise auch `aptitude`)
  - `apt-cache search <stichwort>` um nach Paketen zu suchen
  - `dpkg -l` um installierte Pakete anzeigen zu lassen

  Zum lesen:
  - http://www.debian.org/doc/FAQ/ch-pkgtools.en.html
  - http://www.debian.org/doc/manuals/apt-howto/
  - http://www.debian.org/doc/debian-policy/
- UID, GID
  - `0 - 99`: gleich für alle Debian Systeme, reserviert
  - `100 - 999`: für Systemuser
  - `1000 - 29999`: für normale User
  - `30000 - 59999`: reserviert
- `update-rc.d` um runlevel Konfiguration zu ändern
- Kernel Kompileren "the debian way"

NetInstall über USB Stick I

Eine Möglichkeit Debian zu installieren ist einen bootbaren USB Stick mit einem minimalen Installer zu benutzen. Dieser holt dann automatisch die benötigten Pakete aus dem Internet und installiert diese.

http://www.debian.org/releases/stable/amd64/ch04s03.html.en

Ablauf:

1. Bootimage besorgen:
   http://ftp.nl.debian.org/debian/dists/lenny/main/installer-amd64/current/images/hd-media/boot.img.gz
2. Mit `zcat boot.img.gz > /dev/sdx` das image auf den USB Stick schreiben
3. ISOs für Head und Compute Nodes besorgen:
   - http://ftp.acc.umu.se/debian-cd/current/i386/iso-cd/debian-503-i386-netinst.iso
   - http://ftp.acc.umu.se/debian-cd/current/amd64/iso-cd/debian-503-amd64-netinst.iso
4. USB Stick mounten und die jeweilige .iso Datei darauf kopieren
5. Im BIOS des zu installierenden Rechners USB Boot einschalten

NetInstall über USB Stick II

6. Stick anstecken, Rechner starten, Anweisungen folgen (englischsprachige Installation)
   - http://www.debian.org/releases/stable/amd64/ch05.html.en

Kernel kompilieren „The Debian Way" I

Vorgehensweise:

1. Grundlegende Tools installieren: `apt-get install kernel-package libncurses5-dev gcc`
2. Kernel Sourcen herunterladen (vorher `cd /usr/src/`)
   - Bei Verwendung von vanilla Sourcen: `wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.18.tar.gz`
   - Bei Verwendung von Sourcen mit Debian Patches: `apt-get install linux-source-2.6.18`
3. `tar xjf /usr/src/linux(-source-)2.6.18.tar.(bz2|gz)`
4. `cd linux-source-2.6.18`
5. `make menuconfig`
6. `make-kpkg clean`
7. `fakeroot make-kpkg --initrd --revision=meinkernel.1.0 kernel_image`
   - mit initrd: http://kernel-handbook.alioth.debian.org/ch-initramfs.html

Kernel kompilieren „The Debian Way" II

8. `dpkg -i ../linux-image-2.6.18_meinkernel.1.0_i386.deb`
9. `shutdown -r now`

4 Performance Counters & PAPI

Performance Counters & PAPI

Performance Counter

Ein Performance Counter ist ein Teil eines Mikroprozessors (meist in Form eines speziellen Registers) welches Informationen zu performancerelevanten Ereignissen sammelt. Dies können zum Beispiel Level 1 Data Cache Misses oder Floating Point Operations sein. Meist stehen mehrere Dutzend solcher Counter für einen Mikroprozessor zur Verfügung.

perfctr: the Linux performance monitoring counters kernel extension
PAPI: Performance Application Programming Interface

Unter Linux auf x86, amd64 und Itanium greift PAPI auf das perfctr Kernelmodul zu.

Performance Counters & PAPI installieren

Grundlegend:

- Kernel config sichern
- perfctr Patch in Kernel Sourcen einspielen
- make oldconfig durchführen und dabei perfctr aktivieren
- Kernel kompilieren
- perfctr Testen
- PAPI Bibliothek bauen

Sonstiges

Hausaufgaben

Bitte bis zum nächsten Praktikum unbedingt durchlesen/lernen/probieren!

Was man noch beherrschen sollte:

- Shell Scripts
  - http://tldp.org/LDP/abs/html/
  - http://freeos.com/guides/lsst/
- pdsh einrichten
  - https://computing.llnl.gov/linux/pdsh.html
- iptables und DenyHosts
  - http://iptables-tutorial.frozentux.net/iptables-tutorial.html
  - http://www.netfilter.org/documentation/HOWTO/de/packet-filtering-HOWTO.html
  - http://denyhosts.sourceforge.net/