



## **Implementierung eines skalierbaren Terminal-Servers für plattenlose Clients unter Linux**

## > Übersicht

### ▶ Grundlagen

- ▶ Server Based Computing
- ▶ Diskless-/NetBoot-Clients
- ▶ Cluster/Lastverteilung

### ▶ Umsetzung

- ▶ Terminal-Server
- ▶ Boot-Server
- ▶ openMosix-Kernel

### ▶ Probleme / Lösungen

- ▶ lokale Peripherie
- ▶ Traffic/Bandbreite
- ▶ openMosix mit KDE

### ▶ Fazit



## > Grundlagen > **Server Based Computing**

### Beim **Server Based Computing** ...

- ▶ ...arbeiten viele Benutzer über dumme/einfache Terminals (z.B. ThinClients) auf einem grossen Terminal-Server
- ▶ ...arbeiten die Benutzer genauso, wie wenn sie direkt am Terminal-Server sitzen würden
- ▶ ...laufen alle Prozesse/Programme auf dem Terminal-Server
- ▶ ...sind alle Anwendungen und der Zugang zu den Daten zentral administrierbar

#### ▶ **Grundlagen**

- ▶ **Server Based Computing**
- ▶ Diskless-/NetBoot-Clients
- ▶ Cluster/Lastverteilung

#### ▶ **Umsetzung**

- ▶ Terminal-Server
- ▶ Boot-Server
- ▶ openMosix-Kernel

#### ▶ **Probleme / Lösungen**

- ▶ lokale Peripherie
- ▶ Traffic/Bandbreite
- ▶ openMosix mit KDE

#### ▶ **Fazit**



## > Grundlagen > **warum Diskless-/NetBoot-Clients**

Somit verbleibt aber noch folgendes dezentral an jedem Client:

- ▶ Client-Dateisysteme
- ▶ Betriebssystem-Kernel
- ▶ Betriebssystem-Komponenten und Anwendungen

Lösung:

- ▶ festplattenlose Clients: alles kommt von einem Boot-Server (Diskless-/NetBoot-Clients)

### ▶ **Grundlagen**

- ▶ Server Based Computing
- ▶ **Diskless-/NetBoot-Clients**
- ▶ Cluster/Lastverteilung

### ▶ **Umsetzung**

- ▶ Terminal-Server
- ▶ Boot-Server
- ▶ openMosix-Kernel

### ▶ **Probleme / Lösungen**

- ▶ lokale Peripherie
- ▶ Traffic/Bandbreite
- ▶ openMosix mit KDE

### ▶ **Fazit**



## > Grundlagen > **Diskless-/NetBoot-Clients**

### **Bootvorgang eines Diskless-/NetBoot-Computers:**

- ▶ Computer startet, BIOS leitet Boot-Phase ein
- ▶ **Netzwerk-Bootloader** wird gestartet (PXE oder Etherboot, ca. 16KB) dieser enthält:
  - ▶ Treiber für die jeweilige Netzwerkkarte.
  - ▶ DHCP-Client (IP-Konfiguration und weitere Parameter)
  - ▶ TFTP-Client (Kernel abholen)
- ▶ Starten des Linux-Kernels
- ▶ Mounten des root-Dateisystemes per NFS (readonly: ein Image für alle)
  - ▶ RAMdisk für Schreibzugriffe wird erstellt (/tmp)
  - ▶ Automatische Hardware-Erkennung: NIC, VGA
- ▶ Starten von init -> weitere Prozesse, zentral konfigurierbar
- ▶ z.B. Start einer Terminal-Sitzung oder lokaler Anwendungen

#### ▶ **Grundlagen**

- ▶ Server Based Computing
- ▶ **Diskless-/NetBoot-Clients**
- ▶ Cluster/Lastverteilung

#### ▶ **Umsetzung**

- ▶ Terminal-Server
- ▶ Boot-Server
- ▶ openMosix-Kernel

#### ▶ **Probleme / Lösungen**

- ▶ lokale Peripherie
- ▶ Traffic/Bandbreite
- ▶ openMosix mit KDE

#### ▶ **Fazit**



## > Grundlagen > Cluster

- ▶ Computer im Cluster-Verbund teilen sich die Rechenlast
- ▶ mögliche Ziele:
  - ▶ Skalierbarkeit
  - ▶ Ausfallsicherheit
  - ▶ Performance
- ▶ schnelle (Netzwerk)-Verbindungen nötig
- ▶ i.d.R. spezialisierte Anwendungen nötig
- ▶ nur effektiv bei CPU-intensiven Anwendungen (wenig IO)

### ▶ **Grundlagen**

- ▶ Server Based Computing
- ▶ Diskless-/NetBoot-Clients
- ▶ **Cluster/Lastverteilung**

### ▶ **Umsetzung**

- ▶ Terminal-Server
- ▶ Boot-Server
- ▶ openMosix-Kernel

### ▶ **Probleme / Lösungen**

- ▶ lokale Peripherie
- ▶ Traffic/Bandbreite
- ▶ openMosix mit KDE

### ▶ **Fazit**



## > Umsetzung

- ▶ alle Dienste auf einem Computer / Server (1GB RAM)
- ▶ für optimale CPU-Nutzung: Gentoo Linux  
(zuvor mit SuSE 9.1 & 9.2 getestet)
- ▶ root-Dateisystem für DisklessClients: LTSP -  
(zuvor SuSE 9.1-Minimalinstalltion – ein Image pro Client)

### Terminal-Server:

- ▶ X11 / XDMCP *oder wahlweise*
- ▶ nomachine NX

### Boot-Server:

- ▶ DHCP
- ▶ TFTP
- ▶ NFS

### openMosix-Cluster:

- ▶ Kernel patchen / kompilieren & Tools installieren

### ▶ Grundlagen

- ▶ Server Based Computing
- ▶ Diskless-/NetBoot-Clients
- ▶ Cluster/Lastverteilung

### ▶ Umsetzung

- ▶ Terminal-Server
- ▶ Boot-Server
- ▶ openMosix-Kernel

### ▶ Probleme / Lösungen

- ▶ lokale Peripherie
- ▶ Traffic/Bandbreite
- ▶ openMosix mit KDE

### ▶ Fazit



## > Umsetzung > Terminal-Server

- ▶ **X11 / X-Window**-System schon seit ca. 20 Jahren netzwerktransparent

Client-Server Prinzip:

- ▶ **X-Server** = Ansteuerung von Monitor, Tastatur & Maus (nur grundlegende Grafikfunktionen)
- ▶ **X-Clients** = Alle grafischen Anwendungen – auch der Windowsmanager

Remote X-Login:     X -query <ip.ad.re.sse>

### ▶ Grundlagen

- ▶ Server Based Computing
- ▶ Diskless-/NetBoot-Clients
- ▶ Cluster/Lastverteilung

### ▶ Umsetzung

- ▶ **Terminal-Server**
- ▶ Boot-Server
- ▶ openMosix-Kernel

### ▶ Probleme / Lösungen

- ▶ lokale Peripherie
- ▶ Traffic/Bandbreite
- ▶ openMosix mit KDE

### ▶ Fazit



## > Umsetzung > **Boot-Server**

### ▶ **DHCP**

Vergabe einer IP-Konfiguration und weiterer Parameter wie:

- Adresse des TFTP-Servers und Kernel-Dateiname
- Adresse und root-Pfad des NFS-Servers

### ▶ **TFTP**

- **Triviales FTP** – sehr kleines/einfaches Client-Programm
- Bereitstellung eines Kernels für die Clients

### ▶ **NFS**

- Export eines(!) root-Dateisystems für alle Clients (ro)

### ▶ **Grundlagen**

- ▶ Server Based Computing
- ▶ Diskless-/NetBoot-Clients
- ▶ Cluster/Lastverteilung

### ▶ **Umsetzung**

- ▶ Terminal-Server
- ▶ **Boot-Server**
- ▶ openMosix-Kernel

### ▶ **Probleme / Lösungen**

- ▶ lokale Peripherie
- ▶ Traffic/Bandbreite
- ▶ openMosix mit KDE

### ▶ **Fazit**



## > Umsetzung > openMosix-Kernel

### ▶ openMosix

- ▶ Kernel-Patch für Linux



*"This kernel extension turns a network of ordinary computers into a supercomputer for Linux applications"*

- ▶ transparente Prozessverteilung auf openMosix-Nodes: keine speziellen Anwendungen nötig!
- ▶ dynamische Node-Allokation per multicast
- ▶ nahezu linear skalierbar (im Idealfall)

### ▶ Grundlagen

- ▶ Server Based Computing
- ▶ Diskless-/NetBoot-Clients
- ▶ Cluster/Lastverteilung

### ▶ Umsetzung

- ▶ Terminal-Server
- ▶ Boot-Server
- ▶ **openMosix-Kernel**

### ▶ Probleme / Lösungen

- ▶ lokale Peripherie
- ▶ Traffic/Bandbreite
- ▶ openMosix mit KDE

### ▶ Fazit



## > Probleme > lokale Peripherie

### ▶ **Problem:**

Drucker, direkt an den Clients angeschlossen.  
Alle Druck-Prozesse laufen aber auf dem Server.

### ▶ **Lösung:**

lokale Spooler an den Clients, Druckserver am  
Terminal-Server leitet Aufträge an die Clients  
weiter.

### ▶ **Problem:**

Soundausgabe an den Clients

### ▶ **Lösung:**

Spezielle Sound-Server leiten Ausgabe automatisch  
an den verursachenden Client weiter.

### ▶ **Grundlagen**

- ▶ Server Based Computing
- ▶ Diskless-/NetBoot-Clients
- ▶ Cluster/Lastverteilung

### ▶ **Umsetzung**

- ▶ Terminal-Server
- ▶ Boot-Server
- ▶ openMosix-Kernel

### ▶ **Probleme / Lösungen**

- ▶ **lokale Peripherie**
- ▶ Traffic/Bandbreite
- ▶ openMosix mit KDE

### ▶ **Fazit**



## > Probleme > Traffic / Bandbreite

### ▶ Problem:

Hoher Netzwerktraffic, v.a. bei Grafik-intensiven Anwendungen (Bildschirmschoner, Spiele, Videos, ...)

### ▶ Lösung:

Transparentes komprimieren der Bildinformationen.

-> Verwendung von nomachine's **NX** = hochoptimiertes Protokoll zur bandbreitenschonenden Remote-Sitzung (grafische Sitzungen bei nur 9600 Bit/s möglich!)

### ▶ Grundlagen

- ▶ Server Based Computing
- ▶ Diskless-/NetBoot-Clients
- ▶ Cluster/Lastverteilung

### ▶ Umsetzung

- ▶ Terminal-Server
- ▶ Boot-Server
- ▶ openMosix-Kernel

### ▶ Probleme / Lösungen

- ▶ lokale Peripherie
- ▶ **Traffic/Bandbreite**
- ▶ openMosix mit KDE

### ▶ Fazit



## > Probleme > **openMosix & KDE**

### ▶ **Problem:** bei Verwendung von KDE mit openMosix:

- ▶ vereinzelte (KDE)-Prozesse stürzen ab
- ▶ manche Prozesse verbrauchen zuviel Arbeitsspeicher
- ▶ Ursache noch nicht genau lokalisiert

### ▶ **Lösung:**

- ▶ openMosix vorübergehend am Server deaktiviert bis problematische KDE-Prozesse aufgespürt sind.  
-> Bei 1GB RAM sind 22 "Surf"-Clients kein Problem.
- ▶ openMosix kann einzelne Prozesse von der Migration ausnehmen.

### ▶ **Grundlagen**

- ▶ Server Based Computing
- ▶ Diskless-/NetBoot-Clients
- ▶ Cluster/Lastverteilung

### ▶ **Umsetzung**

- ▶ Terminal-Server
- ▶ Boot-Server
- ▶ openMosix-Kernel

### ▶ **Probleme / Lösungen**

- ▶ lokale Peripherie
- ▶ Traffic/Bandbreite
- ▶ **openMosix mit KDE**

### ▶ **Fazit**



## > Fazit

### ▶ **besonders geeignet für:**

- ▶ Büroarbeitsplätze
- ▶ Klassenzimmer
- ▶ Surfstationen

### ▶ **nicht / nur eingeschränkt geeignet für:**

- ▶ Videobearbeitung / Multimedia
- ▶ 3D-Spiele

### ▶ **Grundlagen**

- ▶ Server Based Computing
- ▶ Diskless-/NetBoot-Clients
- ▶ Cluster/Lastverteilung

### ▶ **Umsetzung**

- ▶ Terminal-Server
- ▶ Boot-Server
- ▶ openMosix-Kernel

### ▶ **Probleme / Lösungen**

- ▶ lokale Peripherie
- ▶ Traffic/Bandbreite
- ▶ openMosix mit KDE

### ▶ **Fazit**





**Vielen Dank für Ihre  
Aufmerksamkeit !**

**... noch Fragen ?**