

# Virtual Elastic Services

Seminar zum Schwerpunkt Systemtechnik  
von  
Kevin Wennemuth

10. März 2007

## .: Überblick .:

- Ausblick auf Virtualisierungskonzepte
- Interessante Themen aus Betriebssystem- und Netzwerktechniken
- 60 Minuten Vortrag
- 15 Minuten Diskussion

# .: Inhalt dieses Vortrages :.

- ① Einleitung
- ② Begriffe und Projekte
- ③ Anwendungsgebiete
- ④ Zusammenfassung

# ∴ Virtuell != Virtuell ∴

- **Native Virtualisierung**

- Volle Emulation aller Hardwarekomponenten
- Schlechte Performance durch Overhead
- Vertreter: *VMWare*, *VirtualBox*

- **Paravirtualisierung**

- Nur Verwaltung von Ressourcen
- Kaum Overhead (3-5%), aber inflexibler
- Vertreter: *Xen<sup>TM</sup>* und *KVM*

# ∴ **Virtuell** != **Virtuell** ∴

- **Native Virtualisierung**

- Volle Emulation aller Hardwarekomponenten
- Schlechte Performance durch Overhead
- Vertreter: *VMWare*, *VirtualBox*

- **Paravirtualisierung**

- Nur Verwaltung von Ressourcen
- Kaum Overhead (3-5%), aber inflexibler
- Vertreter: *Xen<sup>TM</sup>* und *KVM*

## .: Xen - virtual machine monitor :.

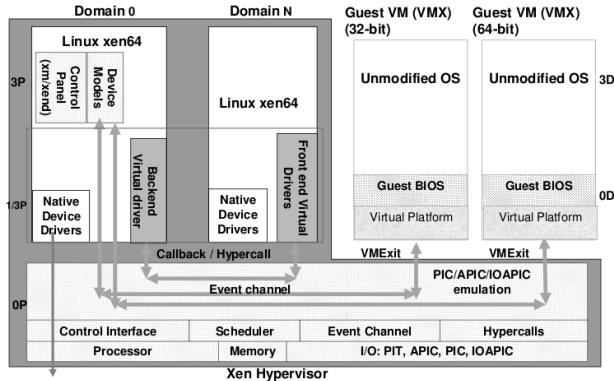
- Entwickelt an der Universität Cambridge
- Teil des XenoServers Frameworks
- Aktuelle Version 3.0.4
- Aufteilung in Anwendungsdomänen (dom0 und domU)

## .: Xen - virtual machine monitor :.

- Entwickelt an der Universität Cambridge
- Teil des XenoServers Frameworks
- Aktuelle Version 3.0.4
- Aufteilung in Anwendungsdomänen (dom0 und domU)

# .: Xen - Aufbau .:

- Hypervisor und Hypercalls





## .: Xen - Features :.

- Unterstützung für Intels *Vanderpool* und AMDs *Pacifica*
- Hardware-Hotplugging
- *shared-storage* zwischen domUs

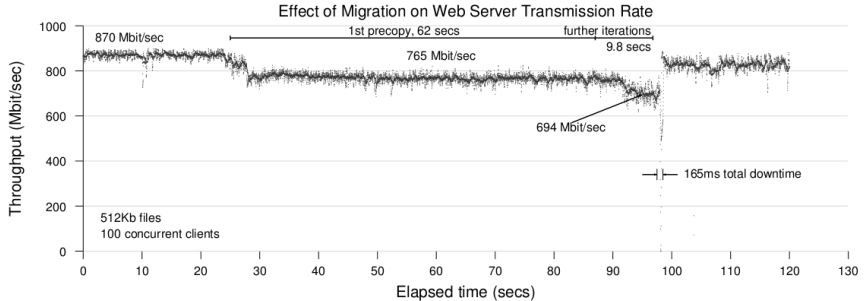
## .: Xen - Features :.

- Unterstützung für Intels *Vanderpool* und AMDs *Pacifica*
- Hardware-Hotplugging
- *shared-storage* zwischen domUs
- Virtuelle Netzwerkverwaltung
- Live-Migration

## .: Xen - Features :.

- Unterstützung für Intels *Vanderpool* und AMDs *Pacifica*
- Hardware-Hotplugging
- *shared-storage* zwischen domUs
- Virtuelle Netzwerkverwaltung
- Live-Migration

## .: Xen - Live Migration :.



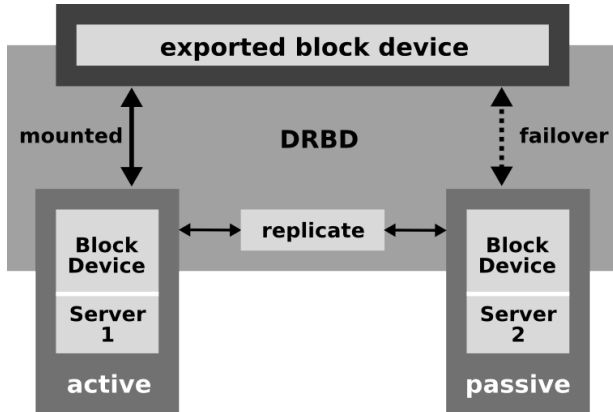
# .: Distributed Redundant Block Device :.

- Spiegelt Block-Geräte
- RAID-1 über Netzwerk
- Unabhängig vom Dateisystem
- *Aktiv-Passiv* und *Aktiv-Aktiv* Betrieb möglich

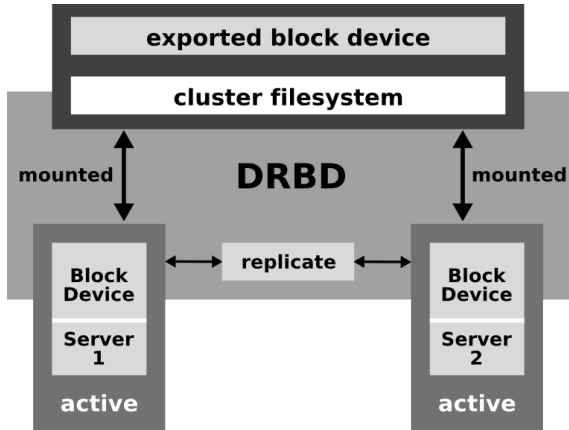
# .: Distributed Redundant Block Device :.

- Spiegelt Block-Geräte
- RAID-1 über Netzwerk
- Unabhängig vom Dateisystem
- *Aktiv-Passiv* und *Aktiv-Aktiv* Betrieb möglich

## .: Aktiv-Passiv :.



## .: Aktiv-Aktiv :.





# .: OCFS2 :.

- Entwickelt für Datenbankcluster
- *POSIX* Kompatibel
- Konkurrierende Zugriffe auf Ressourcen
- *Quorum* und *Fencing*

## .: OCFS2 :.

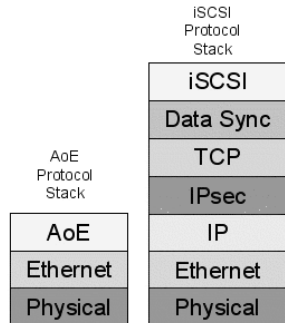
- Entwickelt für Datenbankcluster
- *POSIX* Kompatibel
- Konkurrierende Zugriffe auf Ressourcen
- *Quorum* und *Fencing*

## .: Festplatte am Netz :.

- Client/Server Architektur
- ATA Remote Procedure Calls
- Verbindungsloses Protokoll
- Nicht routingfähig

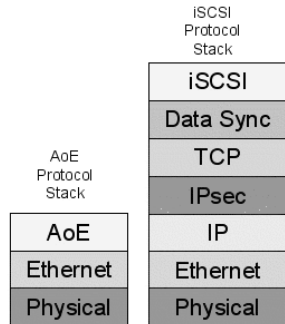
## .: Festplatte am Netz .:

- Client/Server Architektur
- ATA Remote Procedure Calls
- Verbindungsloses Protokoll
- Nicht routingfähig
- Wenig Overhead, schnell



## .: Festplatte am Netz .:

- Client/Server Architektur
- ATA Remote Procedure Calls
- Verbindungsloses Protokoll
- Nicht routingfähig
- Wenig Overhead, schnell



## .: SSI Hosting :.

- Hohes Isolationslevel
- Server Konsolidierung

## .: SSI Hosting :.

- Hohes Isolationslevel
- Server Konsolidierung
- Wenig manuelle Administration

## .: SSI Hosting .:

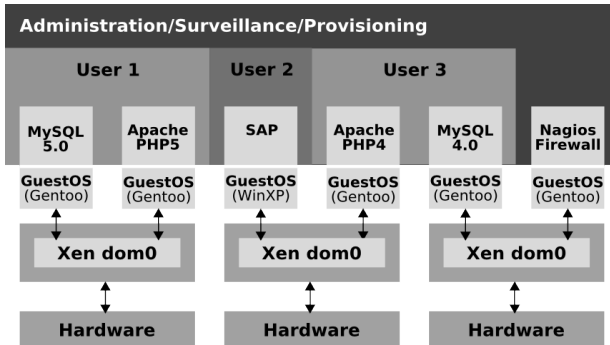
- Hohes Isolationslevel
- Server Konsolidierung
- Wenig manuelle Administration
- Dedizierte Umgebungen - *Out-of-the-box*



## .: SSI Hosting .:

- Hohes Isolationslevel
- Server Konsolidierung
- Wenig manuelle Administration
- Dedizierte Umgebungen - *Out-of-the-box*

## .: SSI Hosting .:



## .: Vorteile für Produktionssysteme :.

- Failovermechanismen durch Paravirtualisierung
- Isolierte Fehleranalyse
- Prototypische Systeme
- Snapshotting durch SSIs
- Bessere Releasezyklen

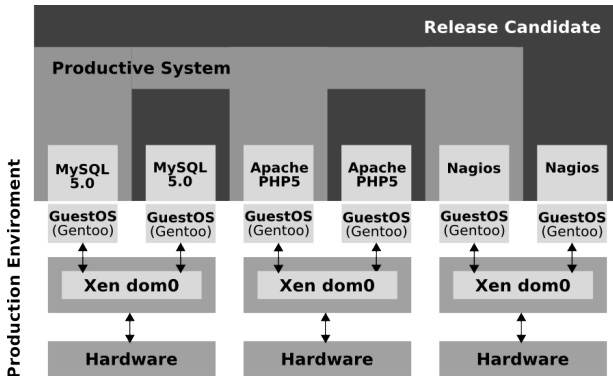
## .: Vorteile für Produktionssysteme :.

- Failovermechanismen durch Paravirtualisierung
- Isolierte Fehleranalyse
- Prototypische Systeme
- Snapshotting durch SSIs
- Bessere Releasezyklen
- Höhere Auslastung der IT Infrastruktur

## .: Vorteile für Produktionssysteme :.

- Failovermechanismen durch Paravirtualisierung
- Isolierte Fehleranalyse
- Prototypische Systeme
- Snapshotting durch SSIs
- Bessere Releasezyklen
- Höhere Auslastung der IT Infrastruktur

## .: Vorteile für Produktionssysteme .:



## .: Virtual Clustering .:

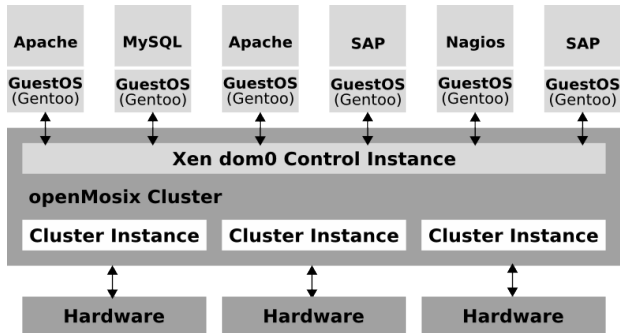
- openMosix oder BeoWulf als Container
- Xen™ *dom0* läuft in Container
- *Dom0* als Prozess durch Cluster verteilt
- *DomUs* mit SSIs

## .: Virtual Clustering .:

- openMosix oder BeoWulf als Container
- Xen™ *dom0* läuft in Container
- *Dom0* als Prozess durch Cluster verteilt
- *DomUs* mit SSIs



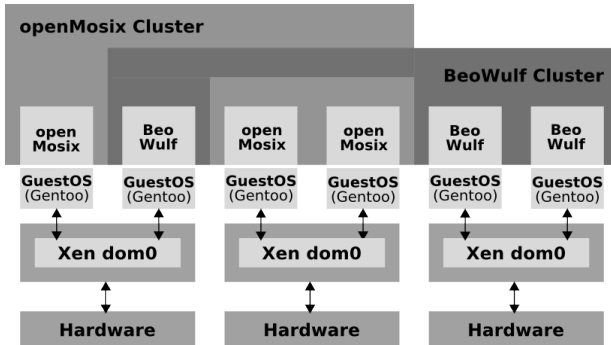
# .: Virtual Clustering .:



## .: Virtual Elastic Clustering :.

- Cluster Instanzen SSIs
- Isolation einzelner Cluster
- Konkurrentes Betreiben mehrerer Cluster
- Clusteranalyse

# .: Virtual Elastic Clustering .:



# .: Projekt ATLAS :.

- Hohe Verfügbarkeit
- Ausfallsicherheit

# .: Projekt ATLAS :.

- Hohe Verfügbarkeit
- Ausfallsicherheit
- Kaum manuelle Administration

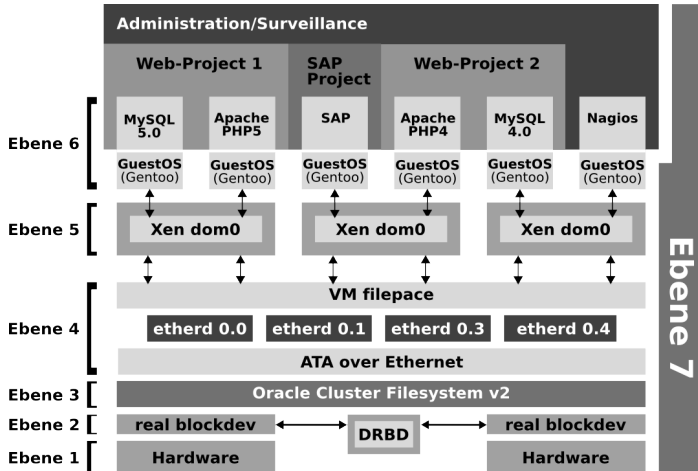
# .: Projekt ATLAS :.

- Hohe Verfügbarkeit
- Ausfallsicherheit
- Kaum manuelle Administration
- Dedizierte Entwicklungs- und Produktumgebungen

## .: Projekt ATLAS :.

- Hohe Verfügbarkeit
- Ausfallsicherheit
- Kaum manuelle Administration
- Dedizierte Entwicklungs- und Produktivumgebungen

# .: Projekt ATLAS :.





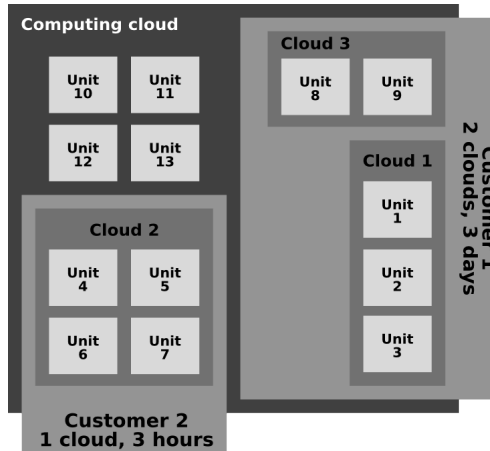
# .: Amazon - Elastic Compute Cloud (EC2) :.

- Amazon Machine Images (*AMI*)
- On-Demand Cluster
- Zeitliche Provisionierung
- Xen™ basierend

# .: Amazon - Elastic Compute Cloud (EC2) :.

- Amazon Machine Images (*AMI*)
- On-Demand Cluster
- Zeitliche Provisionierung
- Xen™ basierend

# .: Amazon - Elastic Compute Cloud (EC2) .:



## .: Fazit :.

- *try-and-Error* Building
- Verwaltungsoberflächen noch nicht ausgereift (*openQRM*)

## .: Fazit :.

- *try-and-Error* Building
- Verwaltungsoberflächen noch nicht ausgereift (*openQRM*)
- Schnelle Marktbewegung
- Innovative Technologien

## .: Fazit :.

- *try-and-Error* Building
- Verwaltungsoberflächen noch nicht ausgereift (*openQRM*)
- Schnelle Marktbewegung
- Innovative Technologien
- Zwei Paradigmenwechseln (*OS, Rechenleistung*)

## .: Fazit :.

- *try-and-Error* Building
- Verwaltungsoberflächen noch nicht ausgereift (*openQRM*)
- Schnelle Marktbewegung
- Innovative Technologien
- Zwei Paradigmenwechseln (*OS, Rechenleistung*)

## .: Diskussion :.

- Vielen Dank für Ihre Teilname
- Fragen und Diskussionen sind herzlich willkommen