

RSIO Konfiguration und Betrieb



Aufsetzen einer RSIO Infrastruktur

Christian Schmidt
Systemingenieur

Warum Storage über Ethernet?

Anforderungen und Möglichkeiten

- **Anforderungen und Erwartungen**

- *Erfordernisse der Anwendungen und Protokolle (Kommunikation, Filesharing etc.)*
- *Preisliche Motivationen*
- *Einheitliche Infrastruktur, weniger Ports ?*
- *Einfachheit, Flexibilität ?*
- *Virtualisierungstechnologien, Verfügbarkeit von Treibern*
- *Zusatzfunktionen (Konvertierungen, Filesystemsnapshots ...)*

- **Möglichkeiten**

- *Gigabit-LAN heute vergleichsweise preiswert*
- *Gigabit-LAN heute in vernünftiger Relation zur Geschwindigkeit einer Festplatte bzw. eines RAID-Systems*
- *Gigabit-LAN heute in günstiger Relation zu Durchsatz-Anforderungen der Applikationen*
- *Mehrere Gigabit-Ports je Server*
- *Ethernet ist eigentlich (fast) kein Ethernet mehr -> Switching-Technologie*
- *RAID-Systeme / Filer sprechen direkt die erforderlichen Protokolle*
- *Neue Performance-Erwartungen an 10Gbit-Ethernet*

Warum Storage über Ethernet?

Anforderungen und Möglichkeiten

- **Anforderungen und Erwartungen**

- Erfordernisse der Anwendungen und Protokolle (Kommunikation, Filesharing etc.)
- Preisliche Motivationen
- Einheitliche Infrastruktur, weniger Ports ?
- Einfachheit, Flexibilität ?

**Vielfalt an Virtualisierungstechnologien, Plattformen ...
erhöht Uniformierungsdruck bei Connectivity**

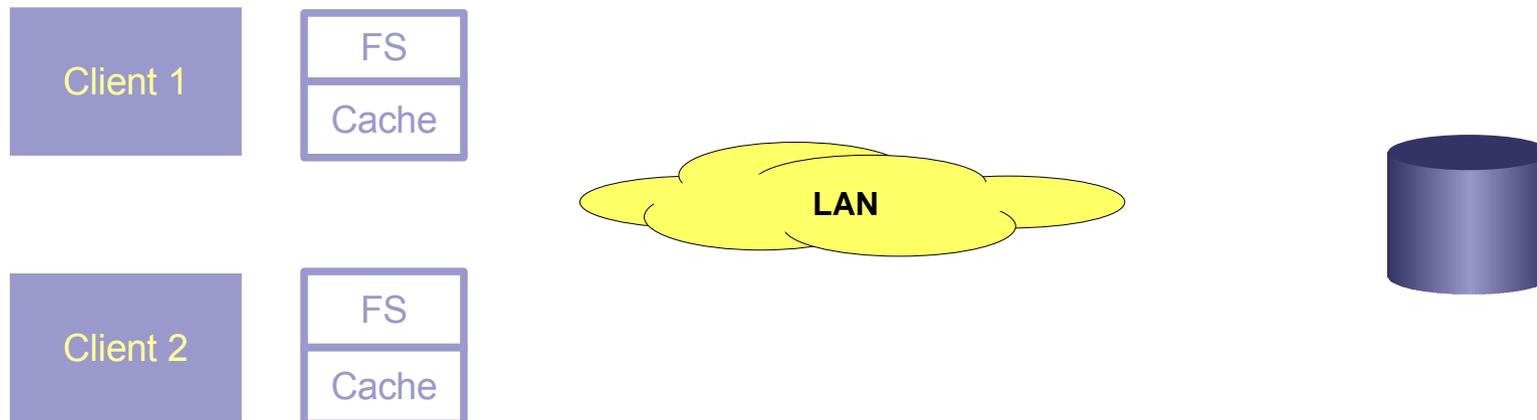
- **Möglichkeiten**

- Gigabit-LAN heute vergleichsweise preiswert
- Gigabit-LAN heute in vernünftiger Relation zur Geschwindigkeit einer Festplatte bzw. eines RAID-Systems
- Gigabit-LAN heute in günstiger Relation zu Durchsatz-Anforderungen der Applikationen
- Mehrere Gigabit-Ports je Server
- Ethernet ist eigentlich (fast) kein Ethernet mehr -> Switching-Technologie
- RAID-Systeme / Filer sprechen direkt die erforderlichen Protokolle
- Neue Performance-Erwartungen an 10Gbit-Ethernet

Starke Argumente für Block-I/O im RZ

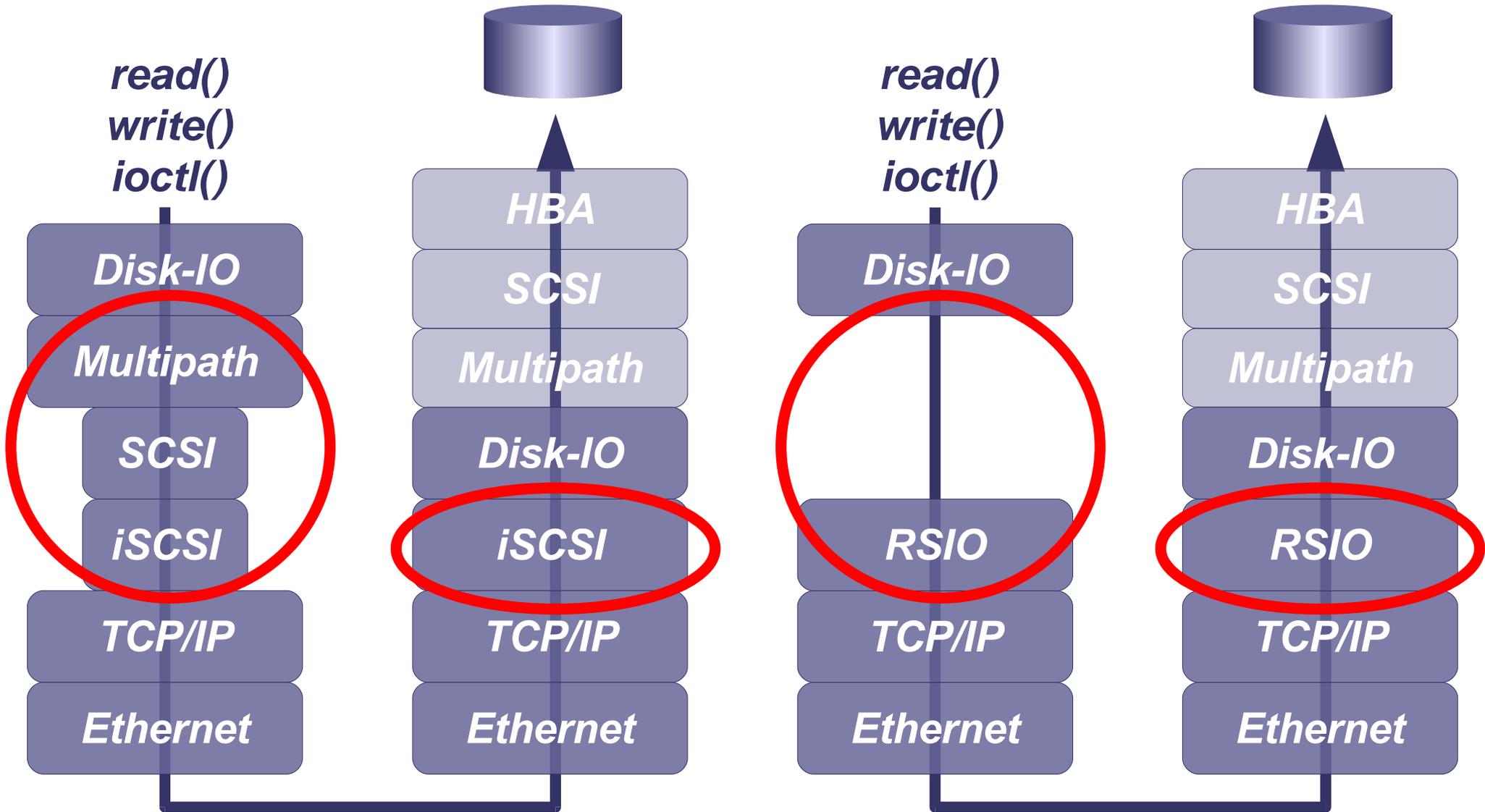
Jenseits von Filesharing (NFS, SMB) überwiegen die Vorteile

- Volle Kontrolle des Client-OS über das Storage-Device
- Nutzbar für beliebige Filesysteme
- Keine Kopplung an Server-OS (Isolation, privates Identity Management)
- Nur Übertragung von I/O, nicht von Cache-Inhalten
- Cache liegt beim Client -> schnellster Zugriff, Client-Caches summieren sich auf
- Einfache Administration, schlankes Protokoll, hohe Geschwindigkeit



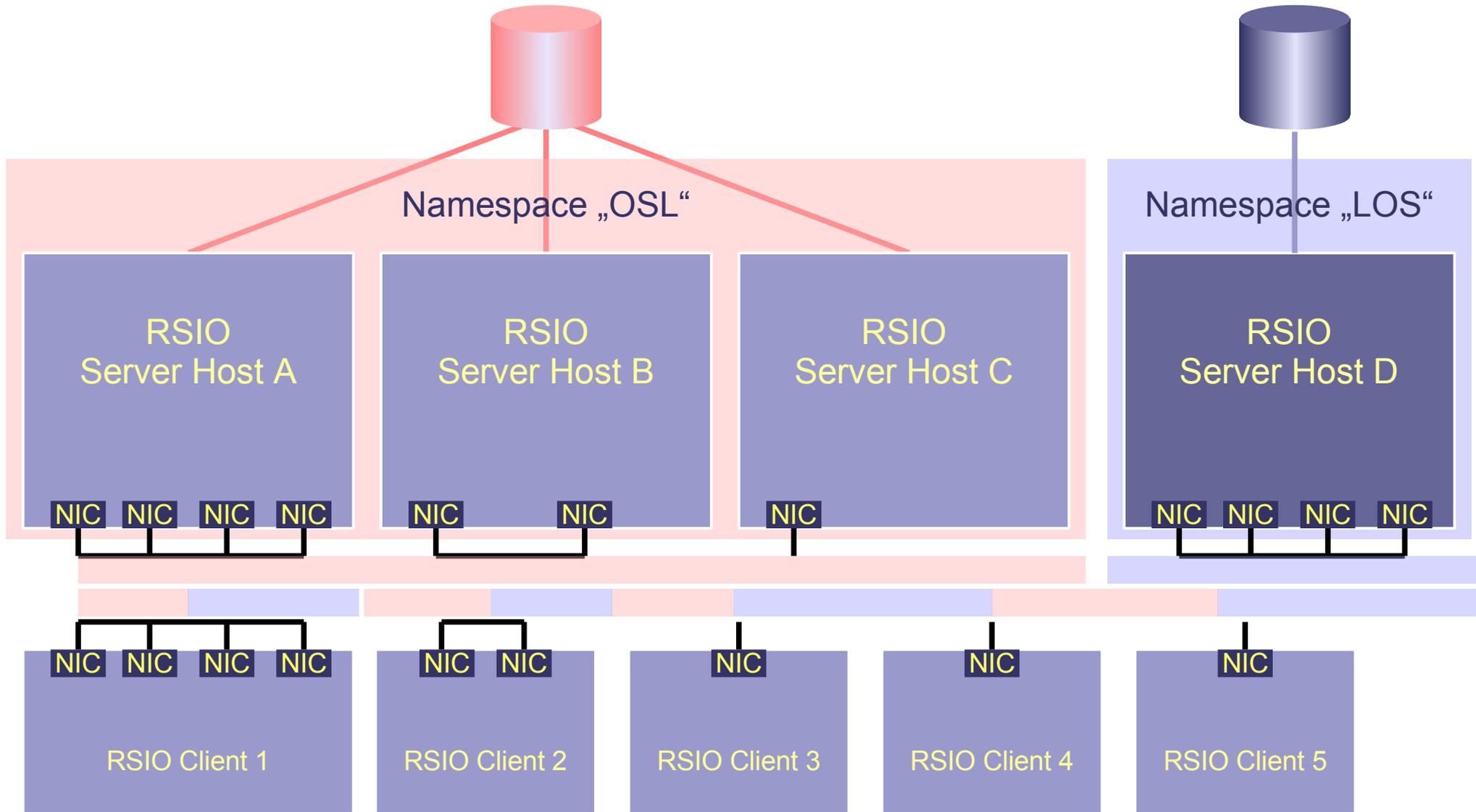
Block-I/O über Ethernet/IP – ohne SCSI

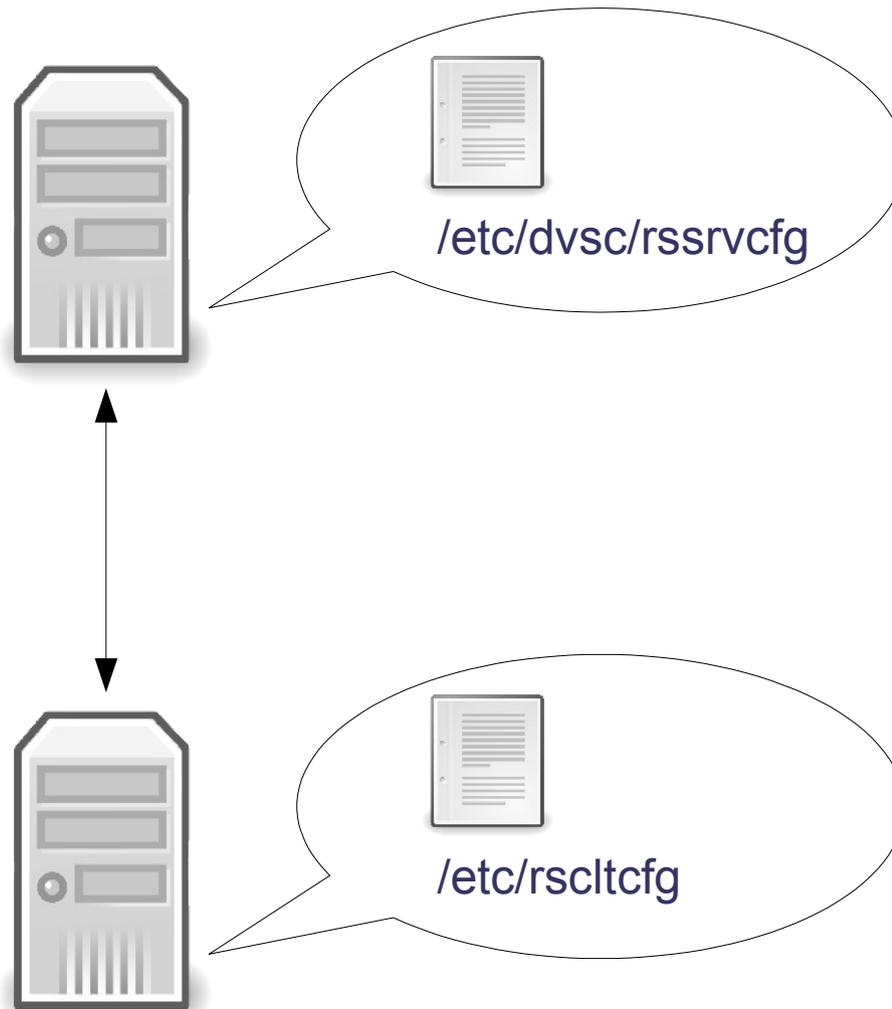
Ein anderer Ansatz mit RSIO



RSIO – Die Architektur im Überblick

Klar gegliedertes und flexibles administratives Konzept





- **RSIO Server Konfiguration**

- *Namespace*
- *Server ID*
- *Interfaces*
- *Clients + Clientkeys*

- **RSIO Client Konfiguration**

- *Namespace + Clientidentifikation*
- *Interfaces*
- *Server*

- **RSIO Server: *rssrv***
 - *exportiert derzeit per Default Virtual Volumes vom Typ Application (OSL SC Base)*
 - *im Moment nur unter Solaris verfügbar*
 - *Stellt alle Application Volumes des RSIO-Servers den Clients zur Verfügung*
- **Ausgabe der Server Logfiles: *rsslogcat [-f]***
 - *Loggt alle RSIO Ereignisse des Servers: Start & Stop, Aushandlung der Clientverbindung, Pfadfehler, Beendigung von Clientverbindungen*
 - *Selbes Verhalten wie bei der OSL-SC Logfacility*

```
big-9@rsio 2010_09_14-06:27:16 (GMT) INFO (osl:sles1/1-ctl): new child server started
big-9@rsio 2010_09_14-06:27:16 (GMT) INFO (osl:sles1/1-ctl): client is: Linux
```

```
# cat /etc/dvsc/rssrvcfg
```

```
[namespace os1]
namespaceid = 200
protocol    = tcp
port       = 5000
```

Namespace Konfiguration

Name, ID und Standardeinstellungen

```
[server big-9]
serverid    = 1
```

Server Name und ID

```
[interface if1]
address     = 192.168.45.10
```

Interface Sektionen

Ein Eintrag für jedes zu nutzende Interface. Port und Protokoll kann auf für jedes Interface individuell gesetzt werden.

```
[interface if2]
address     = 192.168.46.10
```

```
[clients]
big-2      0xbig2
venus      0xvenus
big-6      0xbig6
```

Client Sektion

Ein Eintrag für jeden Client (Name und Schlüssel)

- **Client Dämon: *rsiod***
 - *Läuft im Userspace*
 - *Kommuniziert mit allen konfigurierten und erreichbaren Servern*
 - *Schnittstelle zwischen dem rs-Gerätetreiber und dem Netzwerk*
- **Konfigurationsprogramm: *rsconfig***
 - *Attach und Detach von rs-Devices*
 - *Anzeigen der Geräte (lokale Sicht, Namespace-Sicht)*
 - *Anzeigen von Multipfadinformationen*

```
sles1:~ # rsconfig -q
000 osl
    clt: sles1
    srv: 000 big-9
          0  RSIO_oracle      disk      20971520 blocks of 512 bytes
          0  S10_oracle       disk      20971520 blocks of 512 bytes

sles1:~ # rsconfig -lv
osl:RSIO_oracle@0      20971520 blocks,      1 server(s)
osl:S10_oracle@0      20971520 blocks,      1 server(s)

sles1:~ # rsconfig -m status
0 if0 IP(TCP) 192.168.45.20/5000
    rep: 000 IP(TCP) 192.168.45.10/5000 connected tx: ok rx: ok
1 if1 IP(TCP) 192.168.46.20/5000
    rep: 001 IP(TCP) 192.168.46.10/5000 connected tx: ok rx: ok
```

● *Attach der Volumes*

- *Alle Application Volumes werden per Default ganz normal unter /dev/av[0-3] angelegt*
 - *Selber Pfad wie auf dem RSIO Server*
 - *Applikationen können sowohl auf dem Server, als auch auf dem Client laufen -> keine Anpassungen der Applikation (im Cluster: der ARD) notwendig*
- *RSIO Clients können in mehreren Namespaces registriert sein*
 - *Volumes können nicht mehr unter dem selben Pfad eingehangen werden wie auf dem Server, da Namen doppelt vorkommen können*
 - *RSIO Volumes können beim Attach relociert werden (rsconfig -ar) und erhalten dann wiederum einen durch Einbeziehung des Namespace eindeutigen Namen*

```
root@big-2# rsconfig -ar
root@big-2# rsconfig -lvv
osl:oracle@0                               31457280 blocks,    1 server(s)
  c: /dev/rs/rdisk/osl:oracle@0
  b: /dev/rs/dsk/osl:oracle@0

osl:RSIO_oracle@0                           20971520 blocks,    1 server(s)
  c: /dev/rs/rdisk/osl:RSIO_oracle@0
  b: /dev/rs/dsk/osl:RSIO_oracle@0
```

Der RS-Treiber stellt die Geräte bereit und ist die Schnittstelle zu den Userspace Anwendungen

- ***Userspace-Programme von Solaris und Linux sind fast identisch – die Unterschiede werden beim Treiber sichtbar***
 - *i. d. R. keine Characterdevices unter Linux (für Sonderfälle implementierbar)*
 - *bei Linux klassisch begrenzter Vorrat an Minor-Nummern für die Geräte*
 - *aggressives Caching für Block-Devices durch den Linux-Kern*
 - *Modulhandling*
 - *Treiberkonfiguration*

- *keine Characterdevices unter Linux*

Linux:

```
sles1:~ # rsconfig -lvv
osl:SLES11_oracle@0          20971520 blocks,   1 server(s)
  c:
  b: /dev/av0/SLES11_oracle
```

Solaris:

```
root@big-2# rsconfig -lvv
osl:SLES11_oracle@0          20971520 blocks,   1 server(s)
  c: /dev/av0/rSLES11_oracle
  b: /dev/av0/SLES11_oracle
```

- **Modulhandling und Treiber Konfiguration**

Linux:

- *Laden und Entladen: insmod, rmmod*
- *Module anzeigen: lsmod*
- *Konfigurieren: modprobe.conf*
- *Automatisches Laden: modprobe, /etc/sysconfig/kernel*

Solaris:

- *Laden und Entladen: modload, modunload*
- *Module anzeigen: modinfo*
- *Konfigurieren: /kernel/drv/rs.conf*
- *Automatisches Laden: beim Zugriff auf das Device (s. a. /etc/name_to_major)*

- **Der Client pflegt ein eigenes Logfile**

- Ausgabe mit `rsclogcat [-f]`
- Selbes Verhalten wie die Storage Cluster Logfacility
- Logging von Verbindungsauf- und abbau und Netzwerkereignisse

```
big-2@rsio 2010_09_14-15:02:32 (GMT) INFO (6-if0-tx0): started transmit  
for rep 192.168.45.10/5000
```

- Das Kernelmodul loggt in `/var/adm/messages` (Solaris) bzw. `/var/log/messages` (Linux)

```
Sep 14 16:50:25 big-2 rs: [ID 567783 kern.notice] NOTICE: generic -  
rs_txwait: interrupted
```

RSIO Client Konfiguration

Client Konfiguration

```
root@big-2# cat /etc/rscltcfg
```

```
[defaults]
protocol   = tcp
port       = 5000
```

Default-Werte für Protokoll und Port.
Abweichende Werte können in der Namespace
und Interface Sektion angegeben werden

```
[interface if0]
address    = 192.168.45.20

[interface if1]
address    = 192.168.46.20
```

Interface Sektion
Alle Clientinterfaces die von rsiod genutzt werden
sollen.

```
[namespace os1]
namespaceid = 200
nodename    = big-2
nodekey     = 0xbig-2
```

Namespace Sektion
Für jeden Namespace einen eigenen Abschnitt
mit der Clientidentifikation und den Namespace
Angaben

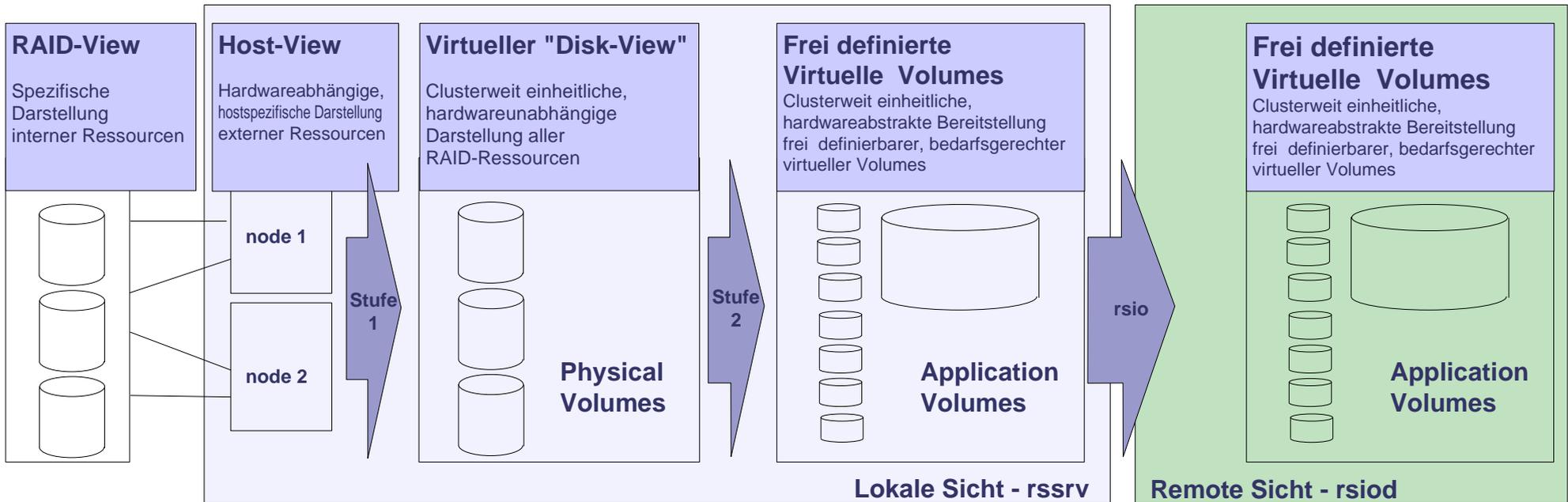
```
[server big-9-1]
address     = 192.168.45.10

[server big-9-2]
address     = 192.168.46.10
```

Server Interfaces
Pro Serverinterface ein Abschnitt. Server gehören
zum vorhergehenden Namespace.

Integriert: RSIO Server + OSL Storage Cluster

Wie Virtual Storage zum Client kommt



RAID

Clusternode mit SAN-Zugriff

Clusternode mit LAN/RSIO – Zugriff

Virtual Storage, d.h.:

- einfachste Handhabung
- kein „format“ etc. notwendig
- viele intelligente Zusatzfunktionen
- online (re)konfigurierbar
- online verschiebbar ...

OSL Gesellschaft für offene Systemlösungen mbH

www.osl.eu

- *noch nie war es so einfach, Standard-Server in leistungsfähige Storage-Server zu verwandeln*
- *noch nie war es so einfach und so preiswert, Server performant an Storage anzubinden*
- *noch nie konnten Storage-Server und Storage-Clients einen so hochintegrierten, leistungsfähigen Cluster bilden*
- *noch nie gab es alles über ein Kabel
(Virtual Storage bis hin zu HA-Cluster und Administration)*
- *noch nie war so eine enge Verflechtung von Solaris und Linux möglich*
- *noch nie gab es so vielfältige Kombinationsmöglichkeiten*