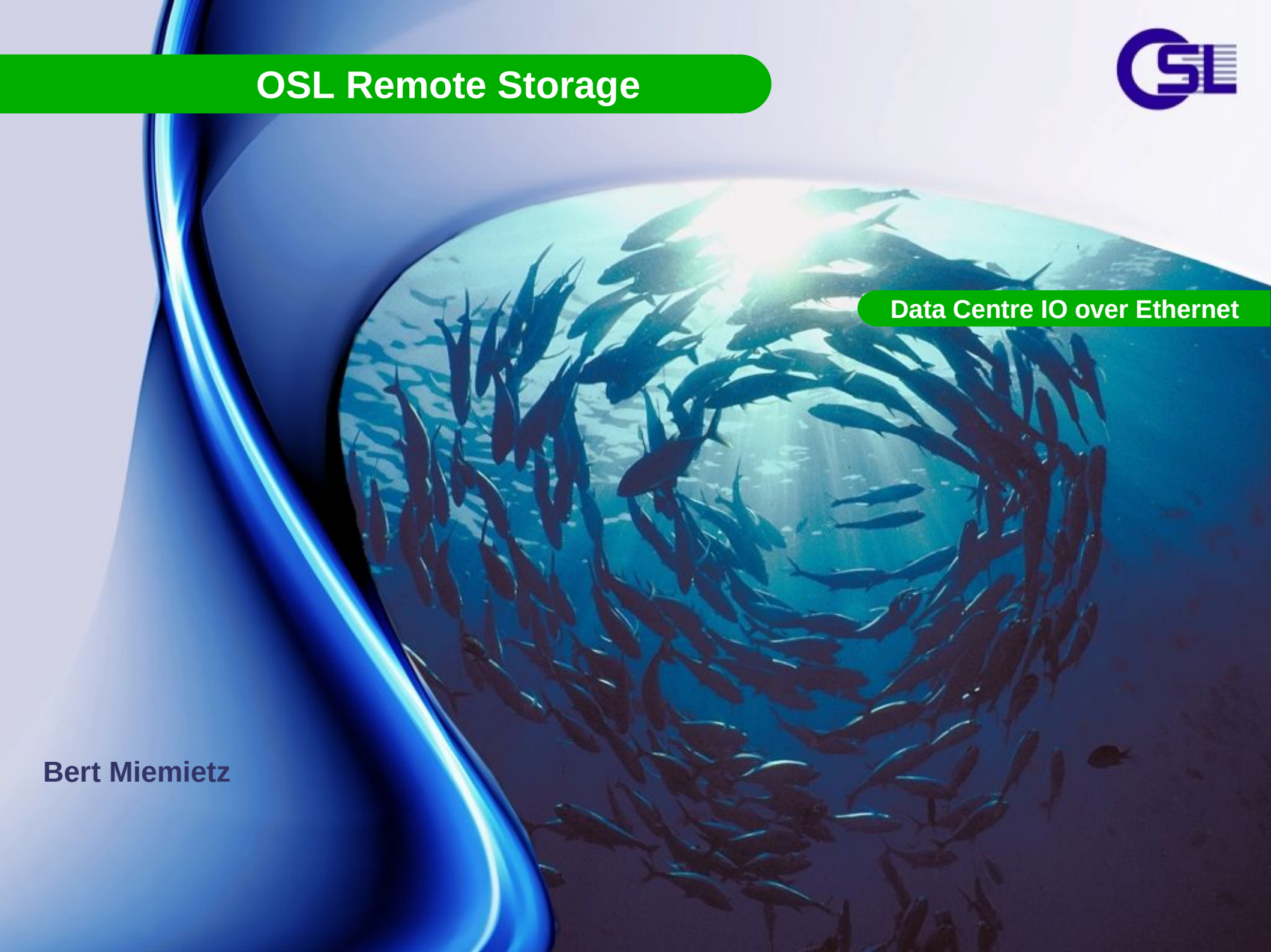
OSL Remote Storage



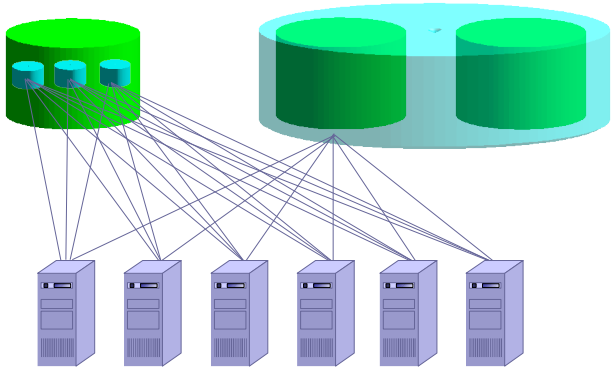
Data Centre IO over Ethernet

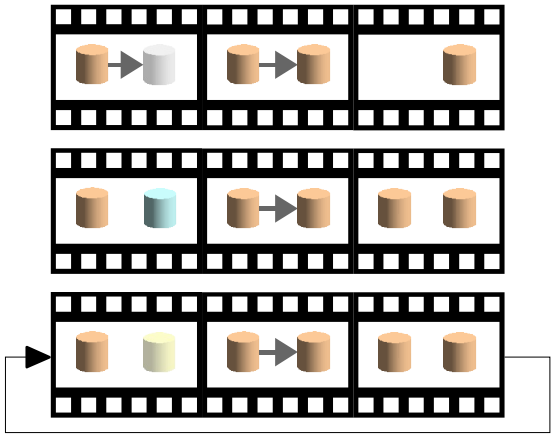
Bert Miemietz

OSL Storage Cluster als Speichervirtualisierung

Blockbasierte Virtualisierung – bedarfsgerecht und zuverlässig

- **Basis-Virtualisierung**
- **clusterweit**
- **globale Pools**
- **Daten verschieben**
- **Daten clonen**
- **Daten spiegeln**
- **Sonderfunktionen**





Physical Volumes + Application Volumes
linear oder integriert (simple, concat, stripe)
HW-Abstraktion und IO-Multipathing
systemgestützte Allokation
Online-Konfig./-Dekonfig./-Vergrößerung

global devices / global namespace
integrated access management

rechnerübergreifend
global inventory
verschnittstellenbasiert
automatische Priorisierung Anwendungs-IO

hostbasiert
anwendungsorientiert
hochverfügbar
skalierbar

einmalig online aufhebbare Ziele kopieren
atomare Operationen für mehrere Volumes

Dauerhafte Beziehung Master -> Image
bis zu 3 Images, inkrementeller Resync
atomare Operationen für mehrere Volumes
Überbrückung Fehler auf Master

XVC (Extended Volume Controls)
z. B. Pause, Stop, Trigger, Aktionen
Bandbreitensteuerung
detaillierte Statistik

Druck in Richtung NAS / Storage over IP

Vielfältige Angebote

- *SNIA*
  - *IP Storage Forum*
  - *Ethernet Storage Forum (SIG iSCSI, SIG NFS, SIG CIFS?)*
  - *(Compellent, Dell, EMC, HP, Intel, Microsoft, NEC, NetApp, Sun. Panasas)*
- *lt. IDC 2009 ca. 30% des Marktes für Speichernetzwerke bei Ethernet*
- *es werden vielfältige Vorteile reklamiert*
  - *einheitliche Infrastruktur*
  - *Kostensenkung*
  - *Flexibilität*
  - *einfachere Handhabung, speziell auch mit virtuellen Maschinen*
  - *Data Sharing und weitere Zusatzfunktionen bei Filern*

klassische Ausprägung als Fileserver

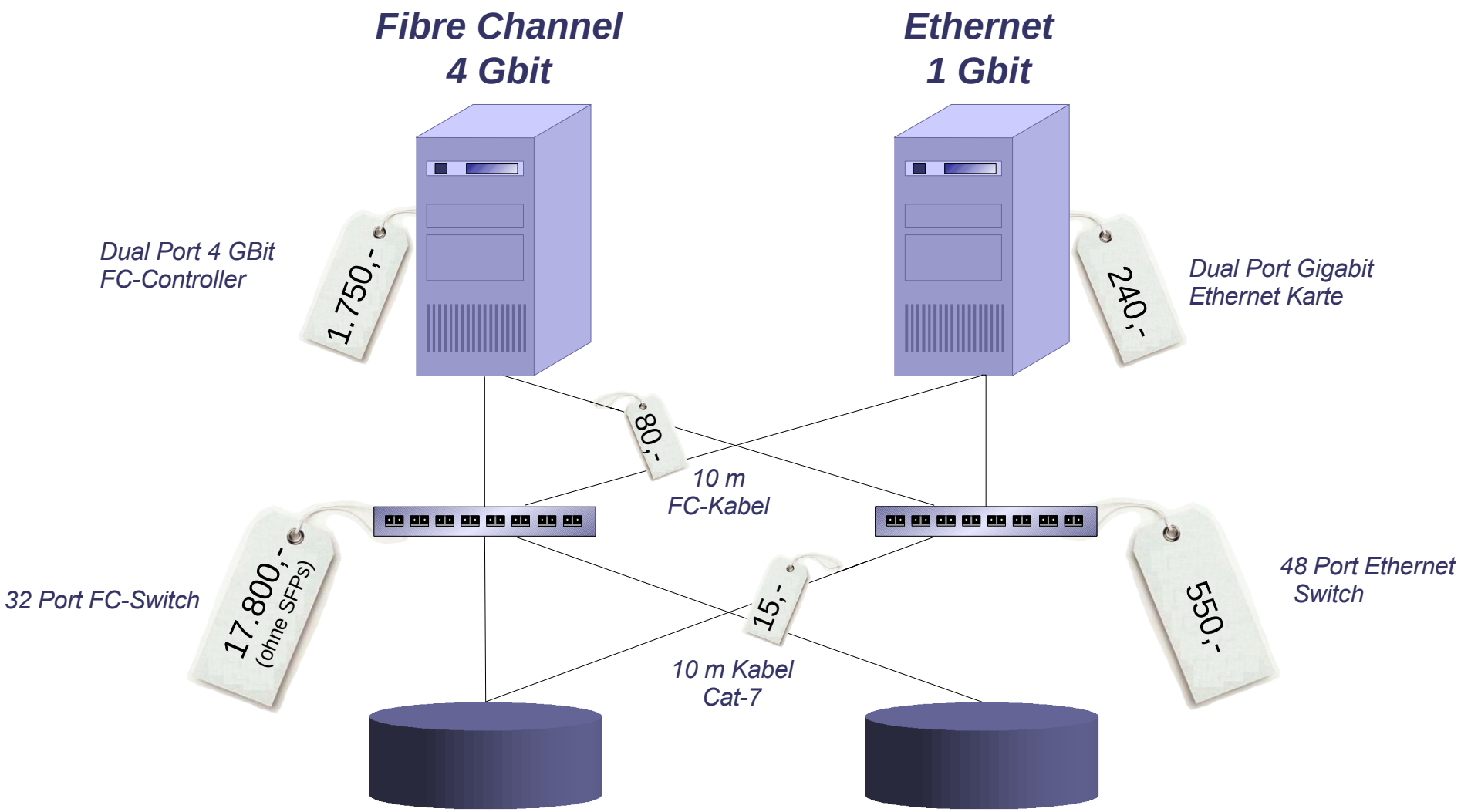
- *NFS*
- *pNFS*
- *CIFS*

aber auch blockorientiert

- *iSCSI*
- *FCoE (?)*
- *iFCP*
- *FCIP*
- *(e)NBD*

Was ist mit den Kostenvorteilen?

Ein Vergleich mit Fibre Channel (Stand Ende 2008)



Gibt es auch technische Argumente?

Was möchte man erwarten?

- *sich anbahnendes 10GBase Ethernet ermöglicht Durchsätze in neuen Dimensionen*
- *Server bringen ab Werk bereits mehrere (Gigabit)-Ethernet Ports mit, weitere lassen sich preiswert nachrüsten*
- *Hardware für effektives Multipathing vorhanden*
- *Gigabit Ethernet liefert theoretisch Durchsätze bis 117 MB/s und wird in praxi heute kaum mehr als Shared Medium betrieben*
- *Durchsatz für viele Anwendungen also ausreichend*
- *SAN-Administration nicht immer einfach – vielleicht geht es hier besser ?*
- *ggf. Verbesserungen in der Bediensicherheit (Verfügbarkeit) möglich?*

Also alles gut?
Mögliche Probleme

- *10GBit Ethernet mit klassischen Programmiermodellen und heutigen CPUs bei weitem nicht ausnutzbar*
- *In der Praxis haben sich nur TCP/IP-basierte Lösungen behauptet*
- *Offload-Engines haben einen schweren Stand -*
  noch immer trauen die Anwender primär dem OS-Stack
- *Blockorientierte Lösungen fristen noch immer ein Schattendasein*
- *Kosten können de facto auch hoch sein (FCoE)*
- *iSCSI ist für den Administrator nicht trivial:*
  - *komplexe Nomenklatur*
  - *bisweilen undurchsichtige Handhabung in anspruchsvollen Umgebungen*
  - *kaum Erleichterungen für Cluster und IO-Multipathing*
  - *Abweichungen von Voreinstellungen (z.B. Cache) können komplex sein*

Block-IO: iSCSI (Solaris) auf der Kommandozeile

Die Target-Seite (Server)

```
[root@big-5] iscsitadm list target
Target: target-0
    iSCSI Name: iqn.1986-03.com.sun:02:06df3360-bb85-ee33-bf59-f2d03474f708.target-0
    Connections: 2
[root@big-5] iscsitadm list target -v
Target: target-0
    iSCSI Name: iqn.1986-03.com.sun:02:06df3360-bb85-ee33-bf59-f2d03474f708.target-0
    Connections: 2
        Initiator:
            iSCSI Name: iqn.1986-03.com.sun:01:ba1adef3ffff.4a431482
            Alias: unknown
        Initiator:
            iSCSI Name: iqn.1986-03.com.sun:01:ba1adef3ffff.4a431482
            Alias: unknown
    ACL list:
        Initiator: big-6
    TPGT list:
        TPGT: 1
    LUN information:
        LUN: 0
            GUID: 600144f04aae26ed00000ae488148e00
            VID: SUN
            PID: SOLARIS
            Type: disk
            Size: 1.0G
            Backing store: /dev/av0/target
            Status: online
```

Block-IO: iSCSI (Solaris) auf der Kommandozeile

Die Initiator-Seite (Client)

So meldet sich die Platte ohne MPXIO in “format”

```
29. c3t227d0 <DEFAULT cyl 1021 alt 2 hd 64 sec 32>
          /iscsi/disk@0000iqn.1986-03.com.sun%3A02%3A06df3360-bb85-ee33-bf59-
f2d03474f708.target-00001,0
31. c3t229d0 <DEFAULT cyl 1021 alt 2 hd 64 sec 32>
          /iscsi/disk@0001iqn.1986-03.com.sun%3A02%3A06df3360-bb85-ee33-bf59-
f2d03474f708.target-00001,0
```

und so mit MPXIO

```
28. c4t600144F04AAE26ED00000AE488148E00d0 <DEFAULT cyl 1021 alt 2 hd 64 sec 32>
          /scsi_vhci/disk@g600144f04aae26ed00000ae488148e00
29. c4t600144F04AAE26EE00000AE488148E00d0 <DEFAULT cyl 1021 alt 2 hd 64 sec 32>
          /scsi_vhci/disk@g600144f04aae26ee00000ae488148e00
```

weiterführende komplexere Themen:

- *MPXIO, Routing, Load-Balancing*
- *Target Portal Groups*
- *Raw-Device Access und IO-Caching*
- *Failover von Targets zwischen verschiedenen Servern*

NAS-Standardprotokolle im OSL Storage Cluster

Integration in die Speichervirtualisierung und die Anwendungssteuerung

- *NFS, SMB, iSCSI bereits heute mit OSL Storage Cluster auf Solaris-basierten Servern darstellbar*
- *in die ACO-Engine integriert bzw. über Templates realisiert*
- *auch in größeren Failover-Clustern einsetzbar*
- *kann nicht mehr / weniger als der jeweilige Solaris-Stand bieten*
- *„Huckepack“ auf vorhandenen Systemen mit installierbarer Software*
- *iSCSI durch Nutzung des Solaris-Targets für viele Plattformen nutzbar*
- *gute Erfahrungen bis NFS-V3*
- *durchaus gute Erfahrungen mit SMB-Stack (Blastwave + Sun)*

NFS, SMB, CIFS

Die Klassiker im Thema NAS (NAFS)

- *handfeste Vorteile*
  - *Spezialisierung auf Fileservices, dafür vergleichsweise einfache Handhabung*
  - *kann komplexe RAID-Funktionen verbergen*
  - *dateisystemtypische Funktionalitäten wie Snapshots*
  - *ermöglicht Filesharing*
  - *Sonderfunktionen*
  - *weite Verbreitung und Unterstützung der wichtigsten Protokolle*
  - *im Rahmen des heute Vorstellbaren Möglichkeiten fast vollständig ausgereizt*
- ***aber auch prinzipbedingte Nachteile***
  - *aufwendige Integration mit Server-OS (Zugriffskontrolle, User-Management)*
  - *Cache- und Cohärenzproblematik, schwierige Nutzung der Client-Ressourcen*
  - *nicht trivial: Multipathing, Skalierbarkeit und Hochverfügbarkeit*
  - *feste Bindung an File-Access-Semantik*
  - *mit zunehmender Funktionalität auch Zunahme von Komplexität und ggf. Inkompatibilitäten*
- ***damit starke Argumente für Block-IO im Rechenzentrum!***

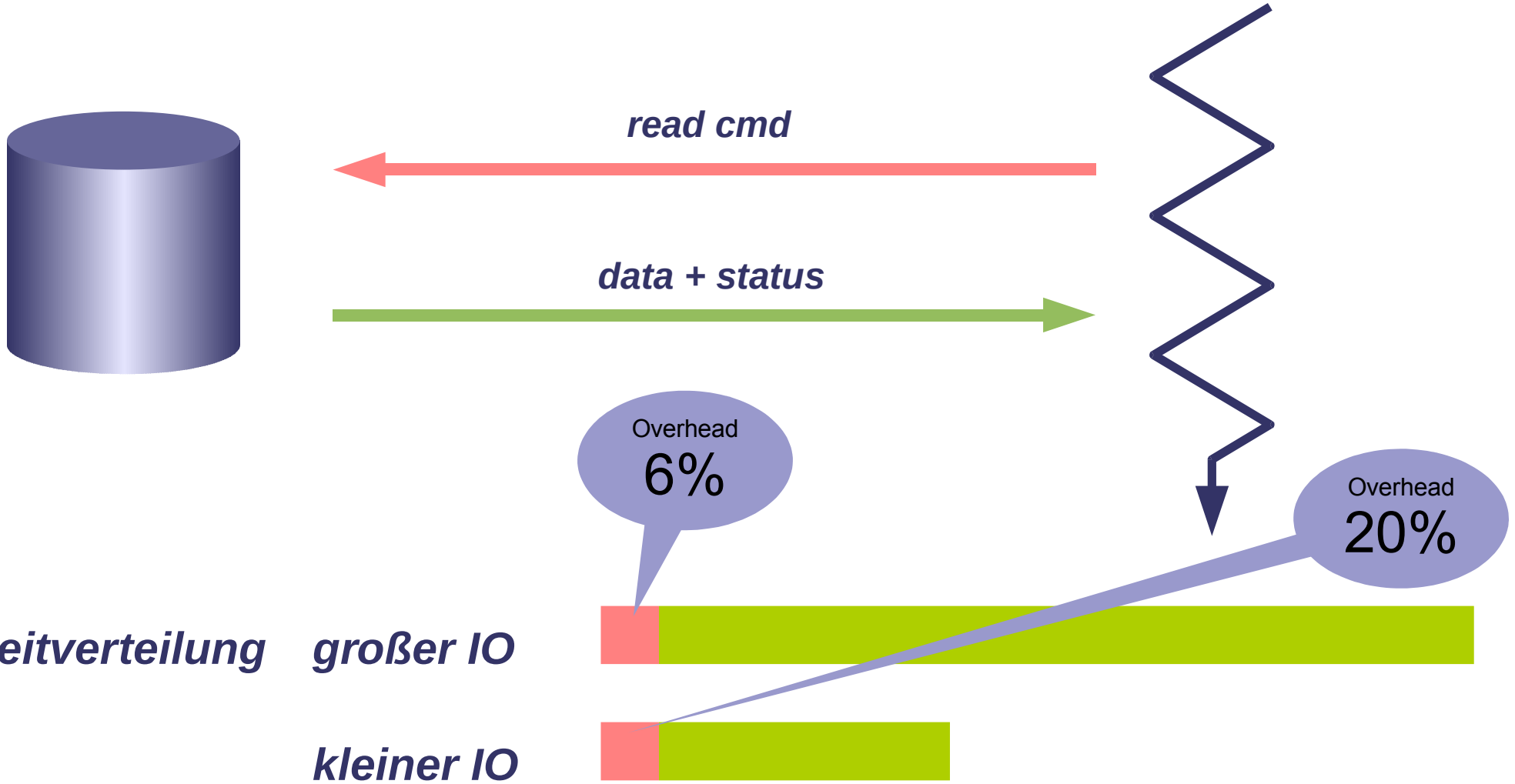
Storage over Ethernet: Taugt das für's RZ?

Fakten und Vorurteile

- *4GBit FC erlaubt den meisten Applikationen mehr IO-Durchsatz als 10GBit Ethernet -> klarer Fall bei extremen IO-Anforderungen.*
- *Brauche ich in jedem Fall > 300MB/s pro Port?*
- *Über 2xGigabit sind nominell 234 MB/s möglich.*
- *IP bedeutet hohe CPU-Belastung, aber:*
- *heutige Multicore-Systeme bieten i.d.R. reichlich Rechenleistung*
- *auch größere SAP-Systeme fordern oft nur 40-60 MByte/s ab*
- *Via Dual-Port Gigabit kann ich meine Systeme unschlagbar preiswert anbinden*
- *Was ist mit Latenzen?*

Latenzen: Das Problem beim Block-IO

Details zum in heutigen Betriebssystemen Unvermeidbaren



Storage over Ethernet: Taugt das für's RZ?

Welche Rolle die Latenzen spielen

Zeit zur Übertragung von 32 Byte an 1 GBit-Lanboard

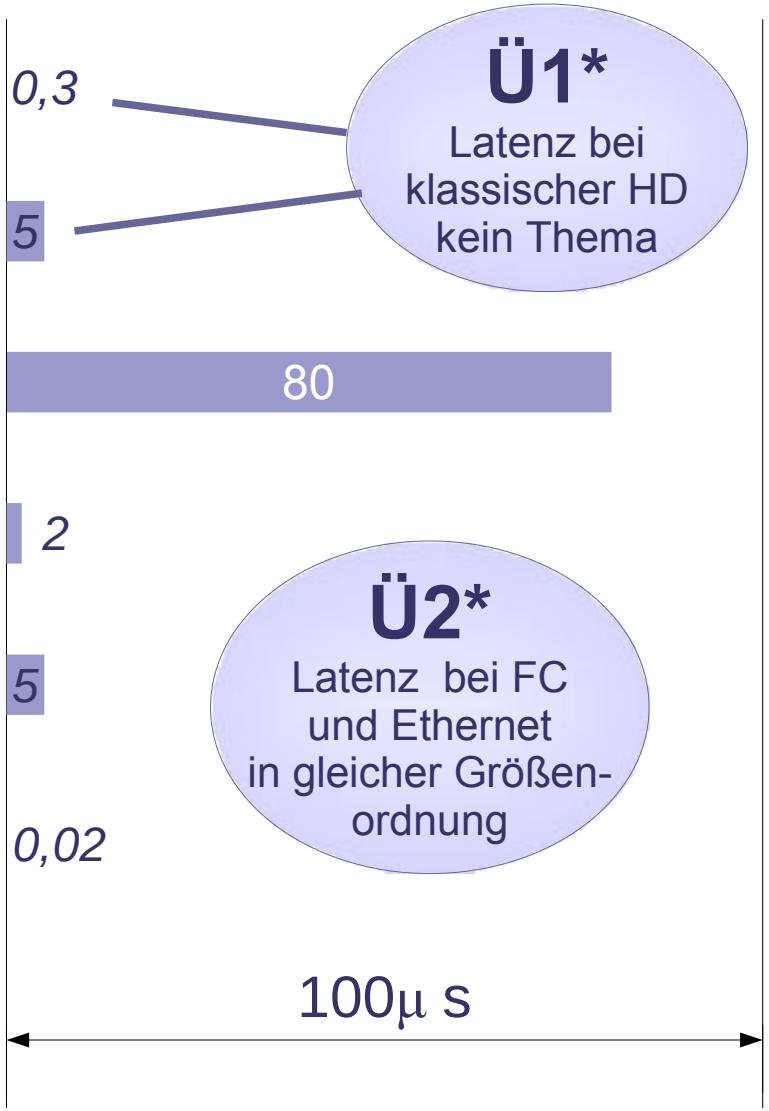
Zeit zur Übertragung von 512 Byte an 1GBit-Lanboard

Zeit zur Übertragung von 8kByte an 1GBit-Lanboard

memcpy 8k auf 600MHz-System

Threadwechsel per CV auf 600MHz-System

32Bit-Typkonvertierung auf 600MHz-System



Storage over Ethernet: Taugt das für's RZ?

Welche Rolle die Latenzen spielen

Zeit zur Übertragung von 32 Byte an 1 GBit-Lanboard

Zeit zur Übertragung von 512 Byte an 1GBit-Lanboard

Es geht also:

Zeit zur Übertragung von 8kByte an 1GBit-Lanboard

1. Weniger um die Hardware.
2. Entscheidend um das Protokoll.

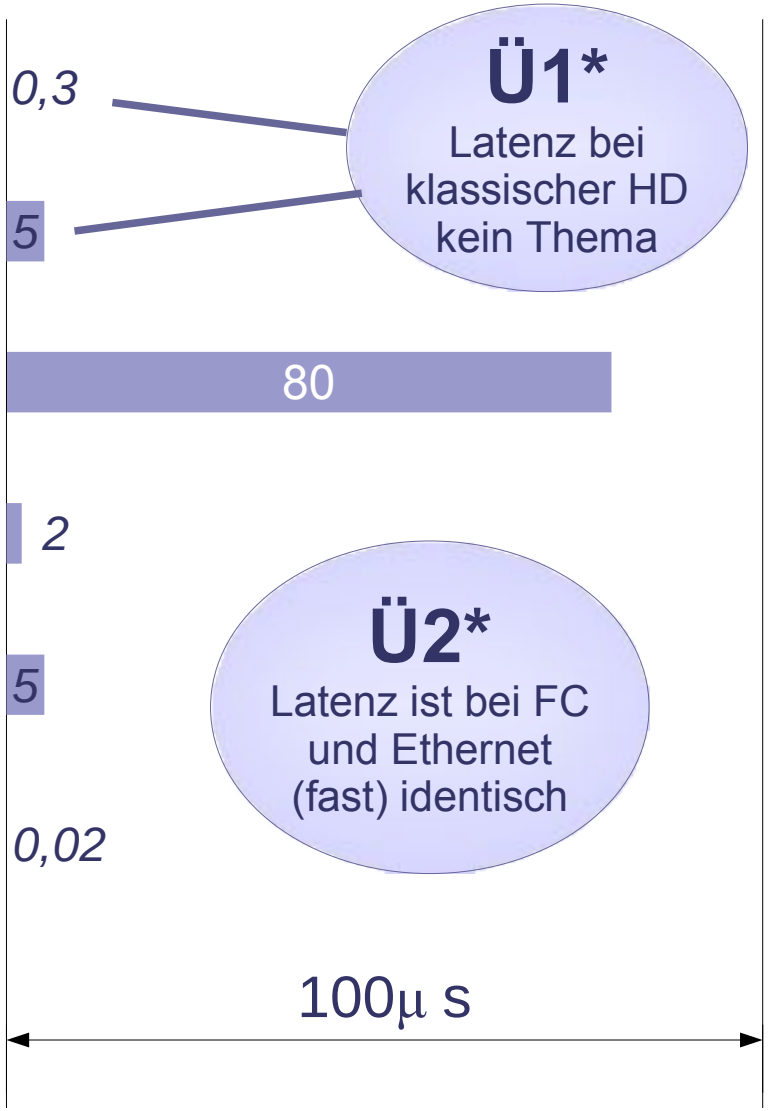
memcpy 8k auf 600MHz-System

3. Entscheidend um die Systemsoftware.

Threadwechsel per CV auf 600MHz-System

4. Nicht unerheblich um die Applikation.

32Bit-Typkonvertierung auf 600MHz-System



Storage over Ethernet? Ja, aber ...

Was wir uns wünschen

- ***Neben NAFS braucht man eine überzeugende Lösung für Block-IO !***
- *Zuverlässigkeit und Skalierbarkeit*
- *einfache Handhabung auch in komplexeren Topologien (kein Zoning)*
- *Unterstützung heutiger wie zukünftiger Transport-Technologien*
- *Nutzbarkeit preiswerter Komponenten*
- *vollständige Abbildung aller relevanten IO-Aufrufe (read, write, ioctl ...)*
- *Multithreading-Support*
- *mit IP: Routingfähigkeit*
- *Erweiterbarkeit, Raum für intelligente IO-Lösungen*
- *Einbindung in OSL-Clustertechnologie*
- *integrierte und bequeme Administration vom Host aus*

Unsere Antwort: Remote Storage IO (RSIO)

Die Eckdaten der neuen Technologie

- *eigenes, von OSL entwickeltes Protokoll*
- *natürliche Erweiterung des OSL Storage Clusters auf Ethernet*
- *voller Clustersupport möglich (Client und Server)*
- *löst die Protokollprobleme im Umfeld shared (LAN-attached) Block Device*
- *zielt auf vollständige Umsetzung der vorgenannten Aufgabenstellungen*
- *geplant zunächst für Solaris und Linux*
- *hochportable Implementierung*
- *internes Layout berücksichtigt moderne CPU- und Serverkonzeptionen*
- *guter Durchsatz und gute Verfügbarkeit mit heutiger Technik*
- *designed to KISS*

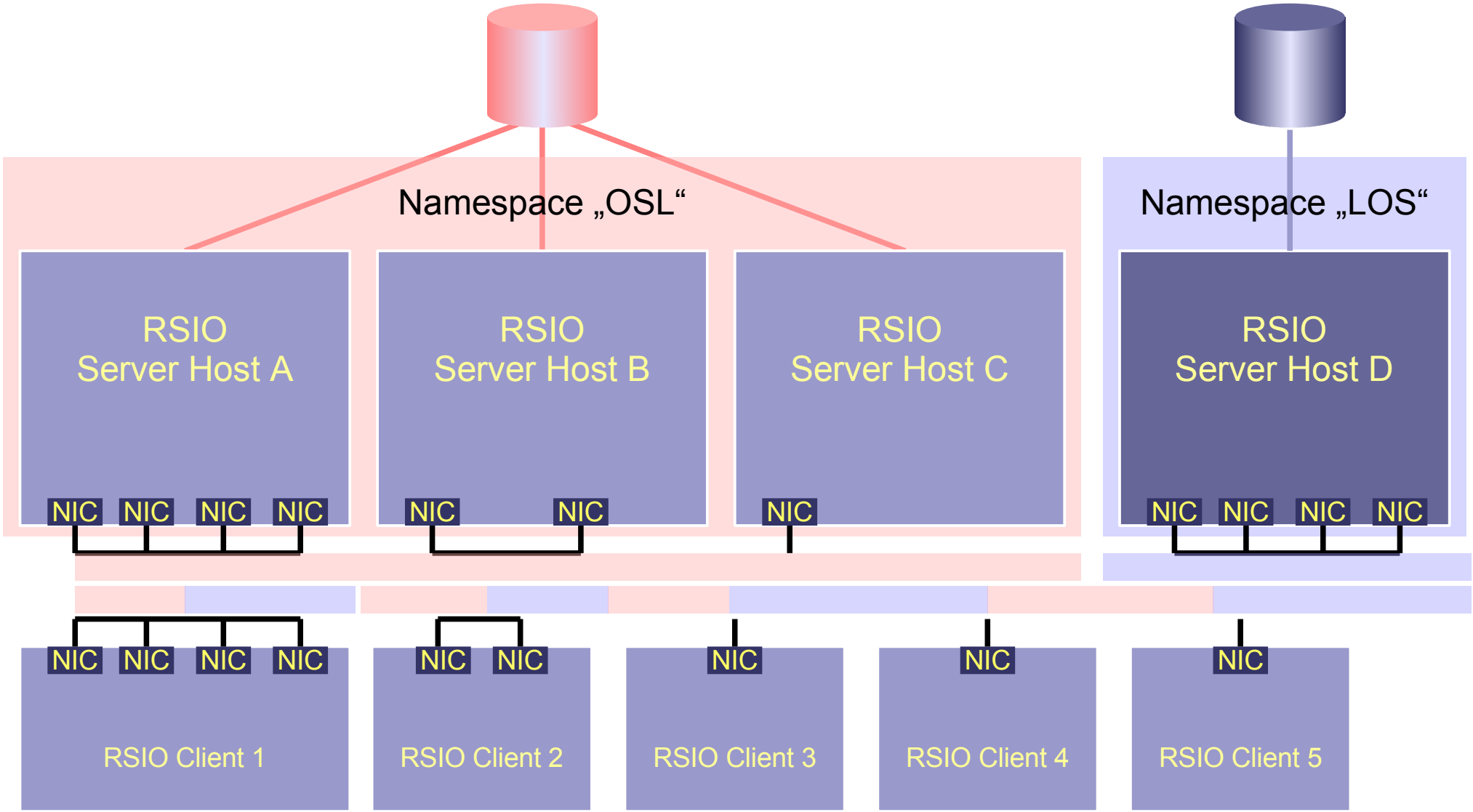
OSL Remote Storage IO (RSIO)

Etwas zu den Details der Implementierung

- *Definition eigener Frames*
  - *Unabhängigkeit von TCP*
  - *ermöglicht Zusatzfunktionen wie Checksum / Encryption*
  - *Frames mit variabler Größe*
  - *Overhead per Frame nur 16 Byte*
- *Modularisierung des Treiber- und Transportsystems*
  - *größerer Funktionsumfang bei hoher Portabilität*
  - *besseres Error-Handling*
  - *hochflexibler Multithreading-Support*
  - *gute Abschirmung des Kerns*
- *Integriertes Multipathing und Trunking*
- *Selbstkonfiguration und Error Recovery*
- *Unterstützung geclusterter Server*
- *vollständige Abbildung relevanter Schnittstellen geplant (z. B. dkio, mtio)*

RSIO - Architektur im RZ

Klar gegliedertes und flexibles administratives Konzept



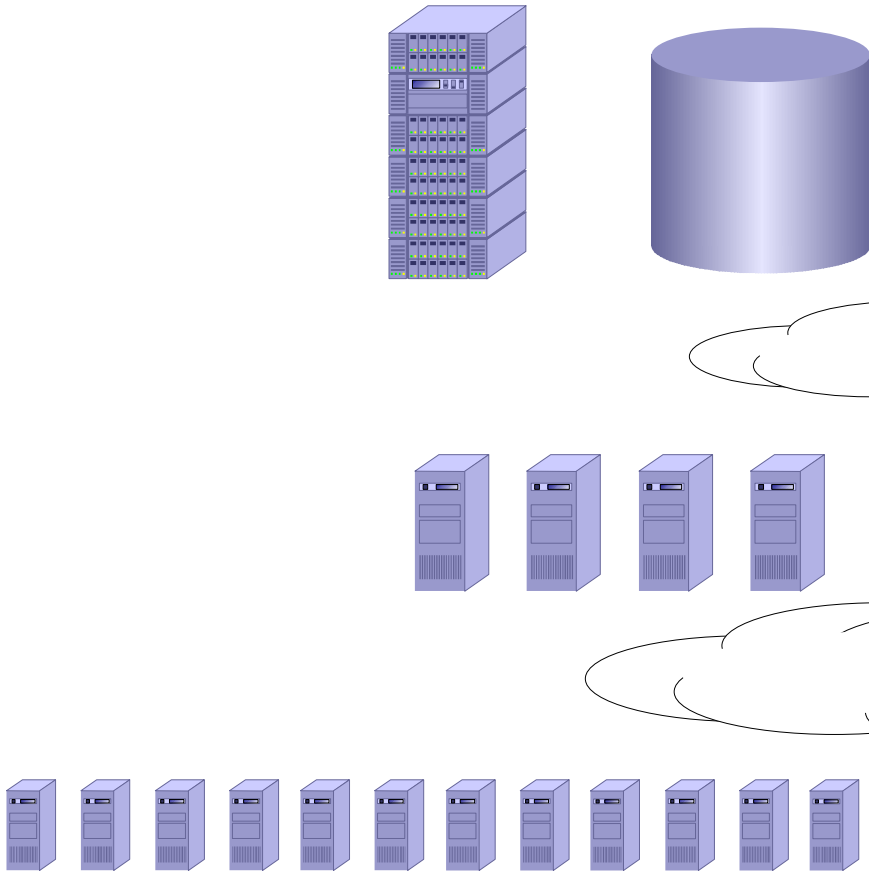
Parameter der RSIO-Architektur

Flexible Client-Server-Implementierung

- *Ein Namespace definiert Server (und Clients) mit Zugriff auf dieselben Storage-Ressourcen*
- *Auf einem Serverhost können (nahezu) beliebig viele Server(prozesse) laufen*
- *Jeder Serverhost kann (nahezu) beliebig viele Clients bedienen*
- *jeder Client unterstützt den Zugriff auf bis zu 256 Server*
- *jede Maschine (Client und Server) unterstützt bis zu 8 Interfaces*
- *Das Protokoll erlaubt Clients simultan Zugriff auf mehrere Namespaces*
- *Auto-Explorer*
  - *Ermitteln verfügbarer Verbindungen*
  - *Ermitteln der Schnittstelleneigenschaften*
  - *Test der Parameter auf der Übertragungsstrecke*

Was kann ich prinzipiell damit tun?

SAN-LAN-Konvergenz

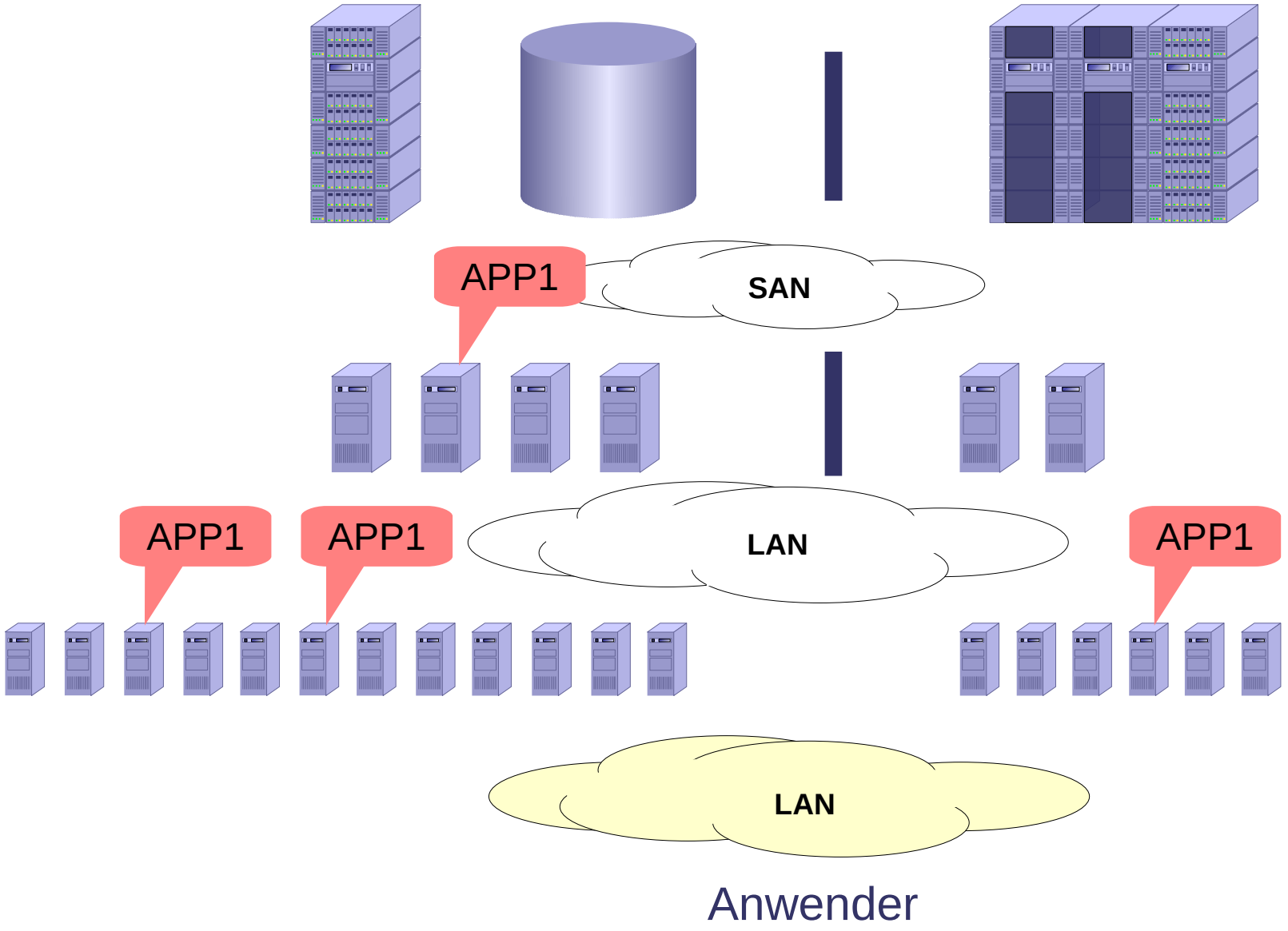


- *Komplexität reduzieren*
- *SAN-Ports einsparen*
- *Portauslastung verbessern*
- *Hosts preiswert und einfach anbinden*
- *Performance-Rightsizing*
- *erleichterte OS-Virtualisierung*
- *Standardkomponenten*

Anwender

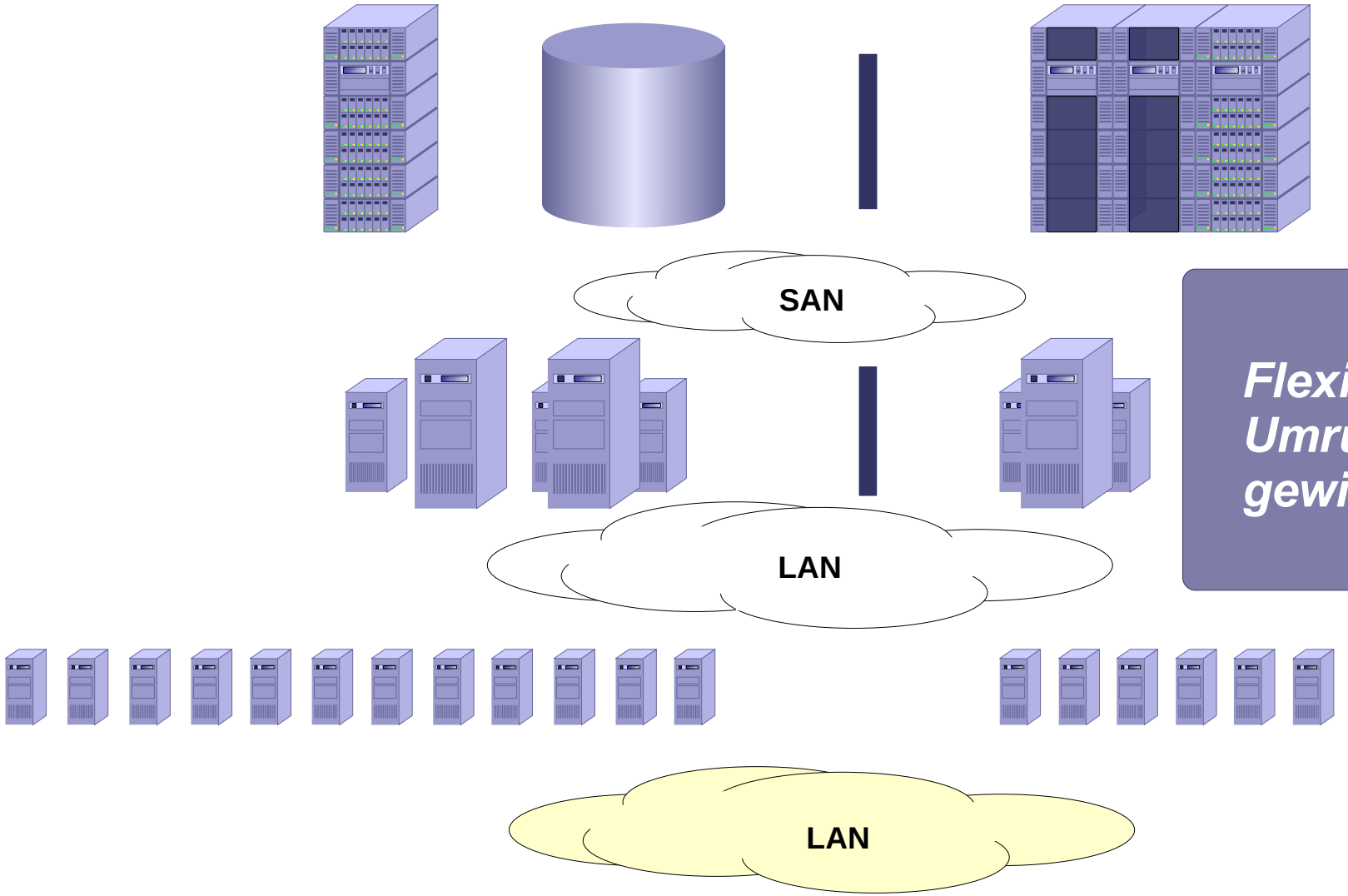
Was wird mit der Hochverfügbarkeit ?

Auch hier: SAN-LAN-Konvergenz



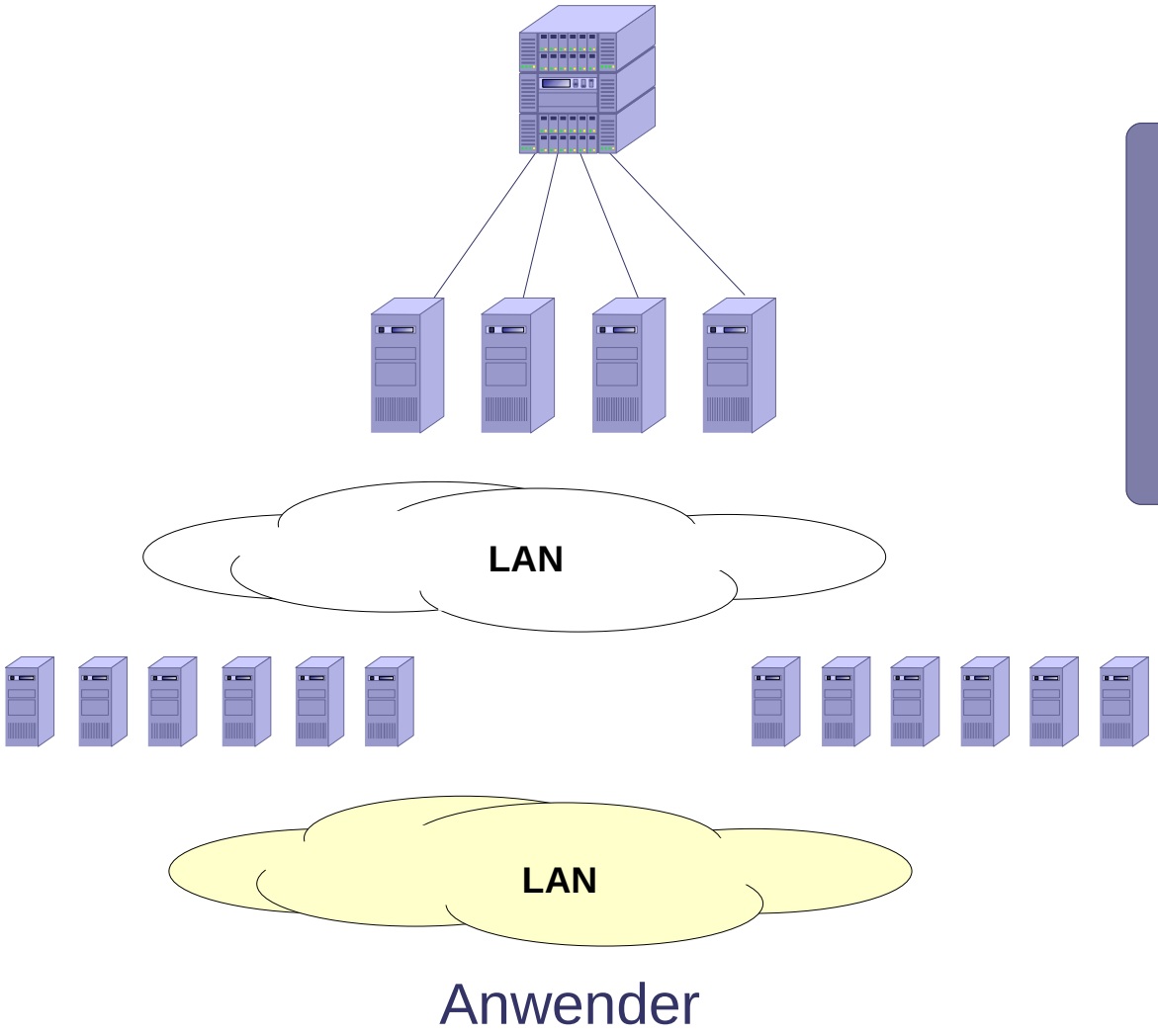
Was kann ich noch anstellen?

SAN-LAN-Konvergenz



Flexibilität bei Umrüstungen gewinnen

Und auch das geht:
Collapsed SAN = gar kein FC-SAN

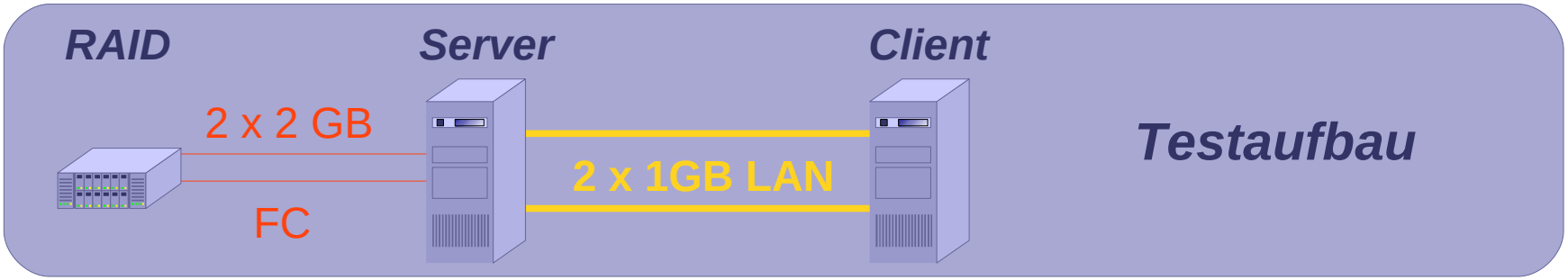


ganz auf das SAN verzichten, nicht aber auf Funktionalität

Und die Performance ?

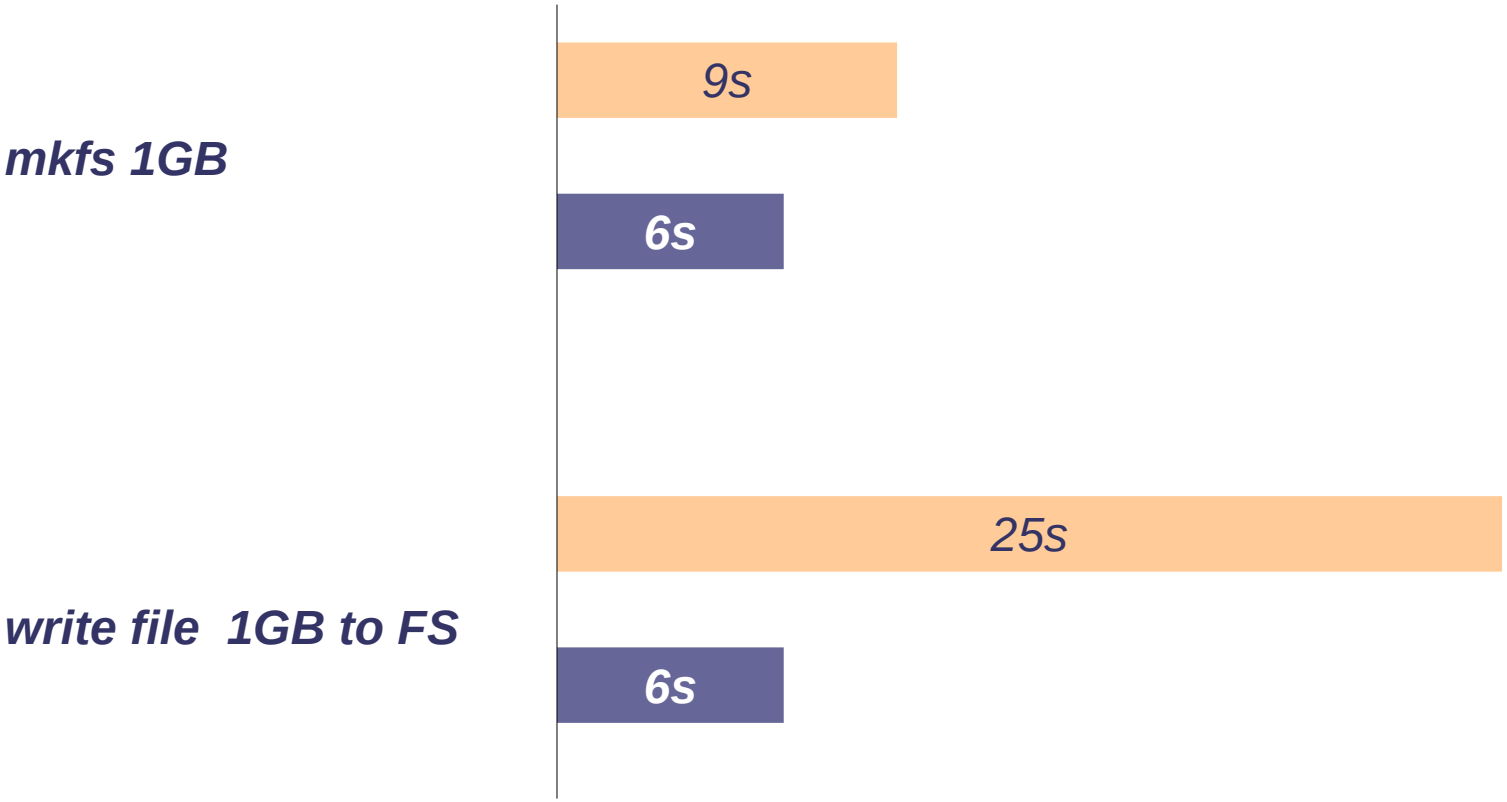
Theoretische Vorbetrachtungen

- *RSIO wird zumeist schwächer sein als FC*
  - *nutzt schwächeres Universalprotokoll TCP/IP*
  - *bei IP werden erhebliche Kommunikationsteile im OS gerechnet*
  - *ausnahmsweise stärker dort, wo der Server nicht über SAN/SCSI auf Storage zugreift*
- *Performance-Boost gegenüber NAFS durch IO-Vermeidung*
  - *virtueller IO-Cache bei exklusivem Zugriff*
  - *hier Vorteil gegenüber NFS/SMB*
  - *in manchen Situationen aber auch leichte Nachteile denkbar*
- *sollte bei vergleichbarer Last stärker sein als iSCSI*
  - *erheblich schlankeres Protokoll*
  - *moderneres Design*
- *Performance-Vorteile durch Multithreading*



Performance in der Praxis

Einige Vergleiche zu iSCSI - Filesysteme



iSCSI

RSIO

Performance ist nicht alles

Die Sicht des Client-Administrators auf den Server

```
[root@big-6] rsconfig -q
000 osl
    clt: big-6
    srv: 000 big-5
         0  tvol1           disk         2097152 blocks of  512 bytes
         0  shadow          disk         2097152 blocks of  512 bytes
         0  z1_root         disk        10485760 blocks of  512 bytes
         0  sparse          disk        10485760 blocks of  512 bytes
         0  whole           disk        41943040 blocks of  512 bytes
         0  iscsit_cfg      disk           20480 blocks of  512 bytes
         0  target          disk         2097152 blocks of  512 bytes
         0  tconf           disk           20480 blocks of  512 bytes
         0  p07             disk       585920023 blocks of  512 bytes
         0  p08             disk       585920023 blocks of  512 bytes
         1  b07             disk       976545023 blocks of  512 bytes
         1  b08             disk       976545023 blocks of  512 bytes
         2  b09             disk       976545023 blocks of  512 bytes
         2  b10             disk       976545023 blocks of  512 bytes
         3  b11             disk       976545023 blocks of  512 bytes
         3  b12             disk       976545023 blocks of  512 bytes
[root@big-6] rsconfig -qv
000 osl (12345)
    clt: big-6 (0139dfX982)
    srv: 000 big-5 (id 1)
         0  tvol1           disk         2097152 blocks of  512 bytes
            c: /dev/av0/rtvol1
            b: /dev/av0/tvol1
```

Performance ist nicht alles

Wie sich die Volumes auf dem Client darstellen

```
[root@big-6] rsconfig -lvv
osl:tvol1@0                                    2097152 blocks,   1 server(s)
    c: /dev/av0/rtvol1
    b: /dev/av0/tvol1

osl:shadow@0                                   2097152 blocks,   1 server(s)
    c: /dev/av0/rshadow
    b: /dev/av0/shadow

osl:z1_root@0                                 10485760 blocks,   1 server(s)
    c: /dev/av0/rz1_root
    b: /dev/av0/z1_root

osl:sparse@0                                  10485760 blocks,   1 server(s)
    c: /dev/av0/rsparse
    b: /dev/av0/sparse

osl:whole@0                                   41943040 blocks,   1 server(s)
    c: /dev/av0/rwhole
    b: /dev/av0/whole
```

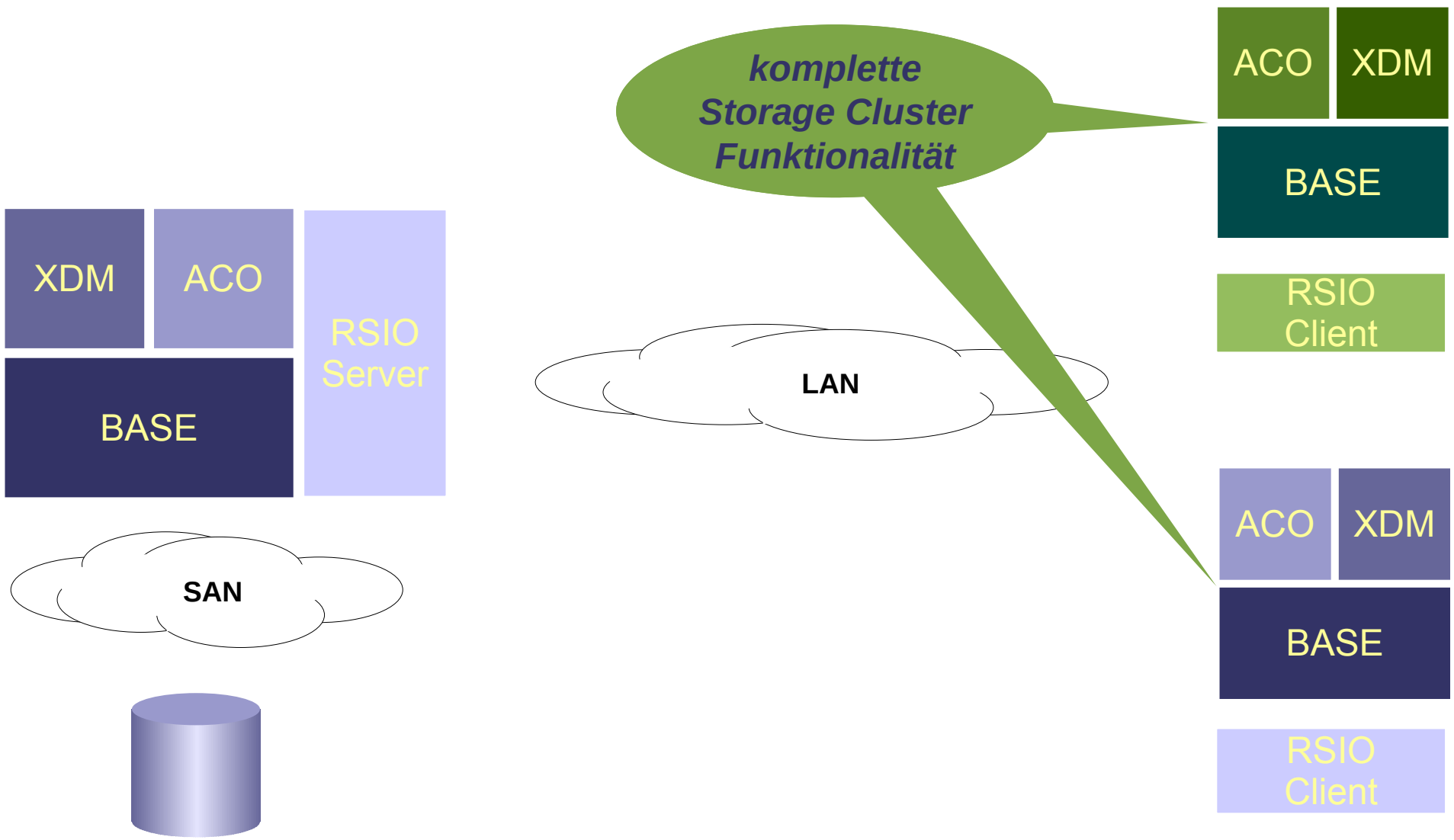
Worüber wir uns noch freuen
Details für die Praxis

- *überlegene Performance*
  - *keine Spezialsettings für TCP/IP -> Performance “out of the Box”*
  - *bis jetzt TCP, UDP wird folgen*
  - *prinzipielle Eignung für beliebige Medien*
  - *noch diverse Verbesserungsmöglichkeiten*
- *gewaltige Gestaltungs-, Entwicklungs- und Tuningmöglichkeiten*
- *extrem schlankes Kernelmodul*
- *Isolation der Serverprozesse*
- *Server paßt sich automatisch an die anstehende Last an*
- *komplettes Handling der Speichervirtualisierung vom Client aus möglich*
- *eingebaute Einfachheit (TCP/IP-Handling, Trunking ...)*
- *vielfältige Nutzungsmöglichkeiten / unglaublich viele Szenarien darstellbar*

Wir hoffen auf Ihre Vorschläge und Ideen!

Wie wird das Produkt aussehen?

Das Design aus heutiger Sicht (Planung)



Die nächsten Schritte

RSIO ist noch immer in einer sehr frühen Phase
Wenn wir uns etwas wünschen dürfen:

- *Testanwender*
- *Feedback und Anregungen*
- *die Möglichkeit, noch vor der Pilotierung Praxiserfahrungen im Design zu berücksichtigen*
- *Vorlauf in der Planung größerer Projekte*

Bitte unterstützen Sie uns –
wir suchen die für Sie geeignetste Lösung!

OSL Remote Storage

Data Centre IO over Ethernet

Besuchen Sie uns auf der SNW !

Frankfurt/M. 26./27. Oktober 2009

Bert Miemietz