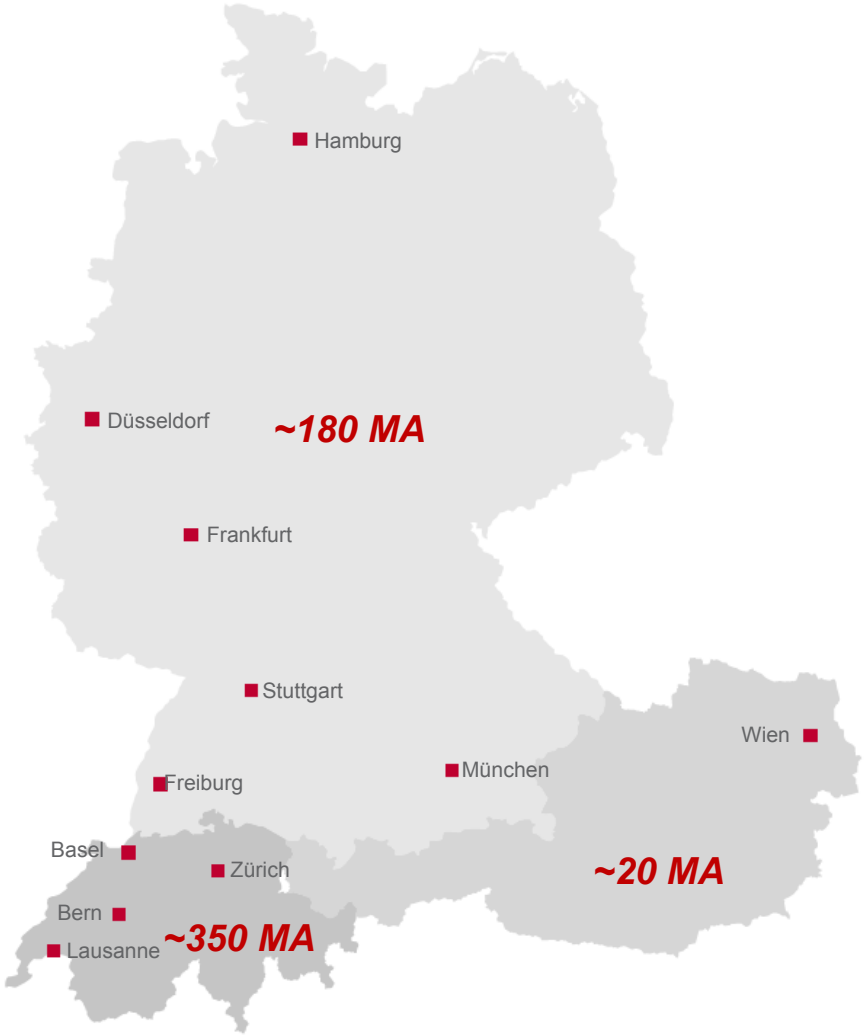
Kopfschmerzen mit ZFS

DOAG Konferenz 2012

Roman Gächter
Principal Consultant
Trivadis AG

21. November 2012
Nürnberg

Trivadis Facts & Figures



11 Trivadis Niederlassungen mit über 600 Mitarbeitern

Finanziell unabhängig und nachhaltig profitabel

Kennzahlen 2011

- Umsatz CHF 104 / EUR 84 Mio.
- Dienstleistungen für über 700 Kunden in mehr als 1‘800 Projekten
- Über 170 Service Level Agreements
- Mehr als 5'000 Trainingsteilnehmer
- Forschungs- und Entwicklungs-budget: CHF 5.0 / EUR 3.6 Mio.

Das Besondere

Kundenindividuelle Lösungskompetenz und Herstellerunabhängigkeit	▪ bietet fundierte Methodenkenntnisse und eigenentwickelte Vorgehensweisen ▪ garantiert wiederholbare Qualität und Realisierungssicherheit
Technologiekompetenz	▪ hat über 17 Jahre Expertise in Oracle und Microsoft ▪ verfügt über ein eigenes Technology Center und setzt auf technologische Exzellenz
Lösungs- und Integrations-Know-how	▪ hat eine breite, branchenübergreifende Kundenbasis und jährlich über 1800 Projekte ▪ verbindet technologisches Spezialistenwissen mit dem Verständnis für die Business-Spezifika des Kunden
Begleitung über den gesamten IT-Projekt-Lifecycle	▪ begleitet den gesamten IT-Projekt-Lifecycle mit einem modularen Dienstleistungsportfolio ▪ bietet für jeden „Reifegrad“ die passende Dienstleistungs- und Lösungskombination

AGENDA Kopfschmerzen mit ZFS

1. **Ausgangssituation**
2. System Architektur
3. Eingrenzung Problem
4. Übersicht ZFS
5. Analyse von ZFS Performance Problemen
6. ZFS und Oracle RDBMS
7. Problemlösung
8. Fazit

Ausgangssituation

„don’t touch a running system“

- Routine Installation von Solaris „recommended Patches“
  - „CPU OS Patchset“ Januar 2012
- Zuerst auf Test Systemen
  - Keine Probleme erkannt
  - OK vom Business
- Installation auf Produktions-Systemen
  - Nach 3 Wochen massive Performance Probleme
  - SWIFT Verbindungsabbrüche
- Einzige Änderung Solaris Patch Installation

AGENDA Kopfschmerzen mit ZFS

1. Ausgangssituation
2. **System Architektur**
3. Eingrenzung Problem
4. Übersicht ZFS
5. Analyse von ZFS Performance Problemen
6. ZFS und Oracle RDBMS
7. Problemlösung
8. Fazit

System Architektur

Redundant aufgebaute Bankenplattform

- Verteilt auf zwei Rechenzentren
- Mehrere Sparc Enterprise M4000 Systeme
  - Solaris 10
- Solaris Virtualisierung mit Zonen
- Alle Applikationen laufen in „Solaris Containern“
- UFS Filesystem für OS, ZFS für Solaris Container, „delegated zpool‘s“
- Daten in Oracle 11G RDBMS
  - Daten auf SAN LUN‘s
  - Ganze Oracle Installation in einem Zpool
    - Synchrone Replikation des Zpool‘s mit SNDR in das andere RZ
    - Datenmenge gering, 50 GB
    - Full Replication darf nicht zu lange dauern
- CPU Last der Applikation ist gering

AGENDA Kopfschmerzen mit ZFS

1. Ausgangssituation
2. System Architektur
3. **Eingrenzung Problem**
4. Übersicht ZFS
5. Analyse von ZFS Performance Problemen
6. ZFS und Oracle RDBMS
7. Problemlösung
8. Fazit

Eingrenzung Problem

Engpass I/O

- Sar zeigt 100% busy
  - Zpool Device Oracle
  - SNDR Replikations-Device
- Storage Durchsatz bescheiden
- AWR Reports
  - Antwortzeiten zum Teil über 50ms (Average Wait)
- Iostat (Intervall 1 Sekunde)
  - asvc_t (average service time of active transactions in milliseconds) oft über 50 ms
- Dedizierter Link zwischen RZ
  - Kapazität nicht ausgeschöpft
    - Flaschenhals nicht bei Replikation
- Erste Analyse zeigt
  - Oracle RDBMS und ZFS muss genauer untersucht werden

AGENDA Kopfschmerzen mit ZFS

1. Ausgangssituation
2. System Architektur
3. Eingrenzung Problem
4. **Übersicht ZFS**
5. Analyse von ZFS Performance Problemen
6. ZFS und Oracle RDBMS
7. Problemlösung
8. Fazit

Übersicht ZFS (Zettabyte File System)

Zitat Wikipedia

ZFS ist ein von Sun Microsystems entwickeltes transaktionales Dateisystem, welches zahlreiche Erweiterungen für die Verwendung im Server- und Rechenzentrumsbereich enthält. Hierzu zählen die enorme maximale Dateisystemgrösse, eine einfache Verwaltung selbst komplexer Konfigurationen, die integrierte RAID-Funktionalität, das Volume-Management sowie der prüfsummenbasierte Schutz vor Datenübertragungsfehlern.

Übersicht ZFS

Wichtigste Features

- Volume Manager und Filesystem kombiniert
- Adressierbare Daten Kapazität ist enorm
  - Speicherkapazität bis 256 Quadrillionen Zettabytes (ZB),
  - 1 ZB = 1,000,000,000,000,000,000,000 bytes
  - 256‘000‘000‘000‘000‘000‘000‘000‘000‘000‘000‘000‘000‘000‘000‘000
- Daten Integrität
  - Prüfsummen gegen Daten Korruption
  - „copy-on-write“ Methode
- Snapshots
- RAID-Z
- native NFSv4 ACLs
- OpenSource Software
  - Lizensiert unter der „Common Development and Distribution License (CDDL)“

Übersicht ZFS

ZFS Caches

- “first level cache” im RAM
  - Variante des ARC Algorithmus (Adaptive Replacement Cache)
- Optional “second level” Disk Cash z. Bsp. SSD Disks
  - „read cache“, L2ARC
    - zpool property cachefile
    - zfs property scondarycache (all | none | metadata)
  - „write cache“, ZIL (ZFS Intent Log)
    - erfüllt POSIX Vorgaben für synchrone Transaktionen
    - ohne ZIL Device, Teil des Zpool‘s
    - „zpool status“ zeigt „log devices“
  - Einfach während dem Betrieb anzufügen und zu entfernen

Übersicht ZFS

Beispiele “command line”

```
# zpool help command
# zpool list
# zpool status -T d 5 3
# zpool create -m /ftpdata ftppool mirror c1t4d0 c1t10d0
# zpool iostat -v ftppool 5 9999
# zpool scrub ftppool
# zpool history
# zfs help command
# zfs list
# zfs get all
# zfs create rpool/data01
# zfs diff u00pool/linda@0913 u00pool/cindy@0914
```

Übersicht ZFS

Betrieb

- status
  - zpool status –x
  - zpool list
- scrubing
  - zpool scrub mypool
  - zpool scrub –s mypool
- upgrade
  - zfs upgrade [-a | myzfs]

Übersicht ZFS

Betrieb

- resilvering

```
# fmadm faulty
# zpool status -x
# zpool replace mypool c1t3d0 c4t3d0
# zpool clear mypool
# zpool online mypool
# zpool status mypool
# fmadm repair cb38fdeb-f64d-6986-a496-fa761e4b898b
(EVENT-ID verwenden)
# fmadm faulty
```

AGENDA Kopfschmerzen mit ZFS

1. Ausgangssituation
2. System Architektur
3. Eingrenzung Problem
4. Übersicht ZFS
5. **Analyse von ZFS Performance Problemen**
6. ZFS und Oracle RDBMS
7. Problemlösung
8. Fazit

Analyse von ZFS Performance Problemen

Tools und Informationen

- Diverse Oracle Whitepapers
- zpool iostat
- iostat

```
$ iostat -xn 1 86399
```

- ARC cache usage

```
root# echo "::memstat" | mdb -k
```

Analyse von ZFS Performance Problemen

Fragmentierung

- Freespace beobachten
  - 80% “usage” nicht überschreiten!!!!!!!!!!!!!!!
- Fragmentation
  - Sind meine Zpool’s fragmentiert?

```
#  dtrace -qn 'fbt::zio_gang_tree_issue:entry { @[pid]=count();  }' -c "sleep 300"
     26574             7141
     26575            18949
     26570           416399
 # ps –eaf | egrep "26574|26575|26570"
     root 26574    0  0  May 26 ?       2778:02 zpool-i-ora-pro06-arc1-pl
     root 26570    0  0  May 26 ?       9155:49 zpool-i-ora-pro06-dat1-pl
     root 26575    0  0  May 26 ?       574:51 zpool-i-ora-pro06-rdo1-pl
 #
```

Analyse von ZFS Performance Problemen

Fragmentierung

- Fragmentation
  - Bitesize.d aus Dtrace Toolkit

Solaris10:
DTraceToolkit-0.99/Disk/bitesize.d
Solaris11:
/usr/dtrace/DTT/Disk/bitesite.d

```
371  zpool-Oracle_R\0

       value  ------------- Distribution ------------- count
         256 |                                         0
         512 |@@@                                      4498
        1024 |@@@@@                                    7236
        2048 |@@@@@@@@@@                               14340
        4096 |@@@@@@                                   8388
        8192 |@@@@@                                    7461
       16384 |@@@@@                                    7710
       32768 |@@@@@                                    7068
       65536 |                                         425
      131072 |                                         40
      262144 |                                         0
```

Analyse von ZFS Performance Problemen

- Nützliche Dtrace “one-liners”
  - Read bytes by process name

```
# dtrace -n 'sysinfo:::readch { @bytes[execname] = sum(arg0); }'
```

  - Write bytes by process name

```
# dtrace -n 'sysinfo:::writech { @bytes[execname] = sum(arg0); }’

  DBBL                                3252103
  ora_cli_hos.exe                     3560996
  ora_cli_snd.exe                     3562250
  ora_cli_get.exe                     3571166
  BBL                                 3725414
  kcfd                                4753085
  srvsql.ORA.exe                      4910908
  ora_recon.x                         9266482
  ora_cli_process.                   21361900
  oracle                             22219303
```

Analyse von ZFS Performance Problemen

- DTrace Toolkit
  - DTraceToolkit-0.99/Docs (Solaris 10)
  - /usr/dtrace/DTT/Docs (default installed Solaris11)

dexplorer hotuser iosnoop procsystime statsnoop
dtruss execsnoop install **iotop** **rwsnoop**
dvmstat hotkernel iopattern opensnoop **rwtop**

- Änderungen im Zpool

zpool history

Analyse von ZFS Performance Problemen

- Mdb
  - zeigt Kernel Parameter
  - Parameter dynamisch ändern
  - Zeigt Memory Usage
- Tunable ZFS parameters, in /etc/system setzen

```
# echo "::zfs_params" | mdb –k
```

- Zeigt Statistiken und Settings der ARC Usage

```
# echo "::arc" | mdb -k
```

- Solaris memory allocation

```
# echo "::memstat" | mdb -k
```

AGENDA Kopfschmerzen mit ZFS

1. Ausgangssituation
2. System Architektur
3. Eingrenzung Problem
4. Übersicht ZFS
5. Analyse von ZFS Performance Problemen
6. **ZFS und Oracle RDBMS**
7. Problemlösung
8. Fazit

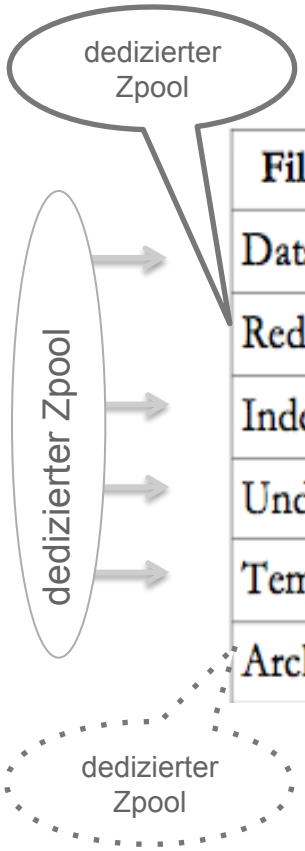
ZFS und Oracle DB

Information

- Configuring Oracle® Solaris ZFS for an Oracle Database
  - Oracle Whitepaper
- ZFS Best Practise Guide
  - http://www.solarisinternals.com/wikiindex.php/ZFS_Best_Practices_Guide
- ZFS Evil Tuning Guide
  - http://www.solarisinternals.com/wiki/index.php/ZFS_Evil_Tuning_Guide
- ZFS for Databases
  - http://www.solarisinternals.com/wiki/index.php/ZFS_for_Databases#ZFS_for_Databases

ZFS und Oracle DB

Daten auf unterschiedliche Zpool‘ s / ZFS Filesysteme verteilen



File System	Record Size	`logbias` Value	`Primarycache` Value
Data files	8 KB	`throughput`	`all` (data and metadata)
Redo logs	128 KB (default)	`latency` (default)	`all` (data and metadata)
Index files	8 KB	`throughput`	`all` (data and metadata)
Undo data	128 KB or 8 KB	`throughput`	`metadata`
Temp data	128 KB (default)	`throughput`	`all` (data and metadata)
Archive data	128 KB, compression on	`throughput`	`metadata`

ZFS und Oracle DB

- “zfs record size” analog zu “Oracle db_block_size”
  - Für “data files”
  - Für “index files”

```
root# zfs create -o recordsize=8k -o \
      mountpoint=/my_db_path/data Oracle/datafiles

root# zfs create -o recordsize=8k -o \
      mountpoint=/my_db_path/index Oracle/index
```

ZFS und Oracle DB

Zfs Properties richtig setzen

- Data files, index files, undo data, temp data, archive data
  - logbias=throughput
    - Only Solaris 10 10/08 to 10/09, parameter in /etc/system

```
set zfs:zfs_immediate_write_sz=8000
```

```
root# zfs set logbias=throughput Oracle/datafiles
root# zfs set logbias=throughput Oracle/temp
root# zfs set logbias=throughput Oracle/undo
root# zfs set logbias=throughput Oracle/archive
```

ZFS und Oracle DB

Zfs Properties richtig setzen

- Undo data and archiv data
  - primarycache=metadata

```
root# zfs set primarycache=metadata Oracle/temp
root# zfs set primarycache=metadata Oracle/archive
```

- Monitor ARC cache

```
root# echo "::memstat" | mdb -k
```

  - Limitieren wenn notwendig

ZFS und Oracle DB

Storage

- Dedizierte Zpool's
  - "data files"
  - "redo logfiles"
  - "archivelog files"

- Zpool's
  - Ganze LUN's verwenden
  - Keine Partitionen, Slices verwenden

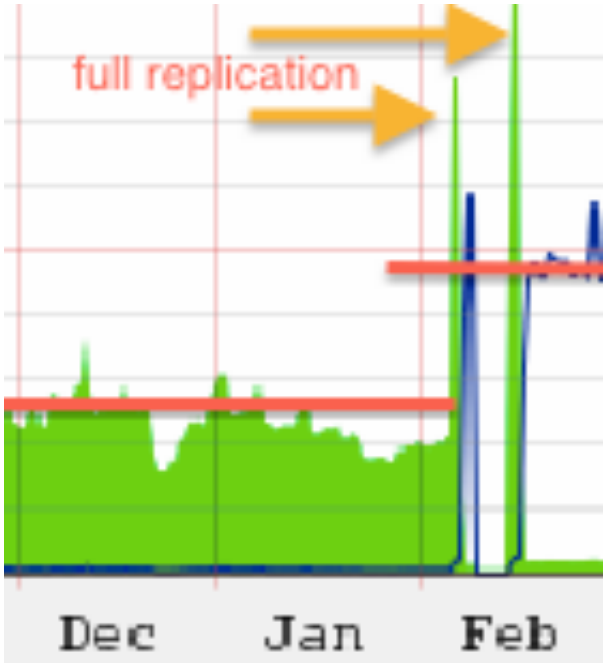
AGENDA Kopfschmerzen mit ZFS

1. Ausgangssituation
2. System Architektur
3. Eingrenzung Problem
4. Übersicht ZFS
5. Analyse von ZFS Performance Problemen
6. ZFS und Oracle RDBMS
7. **Problemlösung**
8. Fazit

Problemlösung

Schritt 1

- Sprunghafter Anstieg der replizierten Datenmenge nach Update



Problemlösung

Schritt 1

- ZFS neue Properties durch Kernel Patches
  - 147440-10 und 144500-19
- Logbias Property
  - Default ist latency
  - Mit „latency“ werden „synchronous writes“ zuerst in ZIL geschrieben
    - Für Oracle DB Writes muss alles zweimal geschrieben werden
    - In Oracle Solaris 10 10/08 bis 10/09
      - Kernel Parameter „set zfs:zfs_immediate_write_sz=8000“
  - Oracle empfiehlt für Database Files

```
logbias = throughput
```

  - Dadurch wird der ZIL umgangen
- Resultat der Änderung
  - Durchsatz auf der Replikationsleitung wieder wie vor dem Update

Problemlösung

Schritt 2

- Im ZFS für die „redolog files“ war „recordsize“ auf 8k eingestellt
  - Sollte 128k sein
- „bitesize“ Verteilung war anders als erwartet

```
6. May
    371  zpool-Oracle_R\0

          value  ------------- Distribution ------------- count
            256 |                                         0
            512 |@@@                                      4498
           1024 |@@@@@                                    7236
           2048 |@@@@@@@@@@                               14340
           4096 |@@@@@                                    8388
           8192 |@@@@@                                    7461
          16384 |@@@@@                                    7710
          32768 |@@@@@                                    7068
          65536 |                                         425
         131072 |                                         40
         262144 |                                         0
```

Problemlösung

Schritt 2

- Anpassen der „recordsize“ für „redolog files“ auf 128k
- Save und Restore aller Daten im Oracle Zpool
- „bitesize“ Verteilung sieht besser aus

```
6. May
   371  zpool-Oracle_R\0

         value  ------------- Distribution ------------- count
           256 |                                         0
           512 |@@@                                      4498
          1024 |@@@@@                                    7236
          2048 |@@@@@@@@@@                               14340
          4096 |@@@@@                                    8388
          8192 |@@@@@                                    7461
         16384 |@@@@@                                    7710
         32768 |@@@@@                                    7068
         65536 |                                         425
        131072 |                                         40
        262144 |                                         0
```

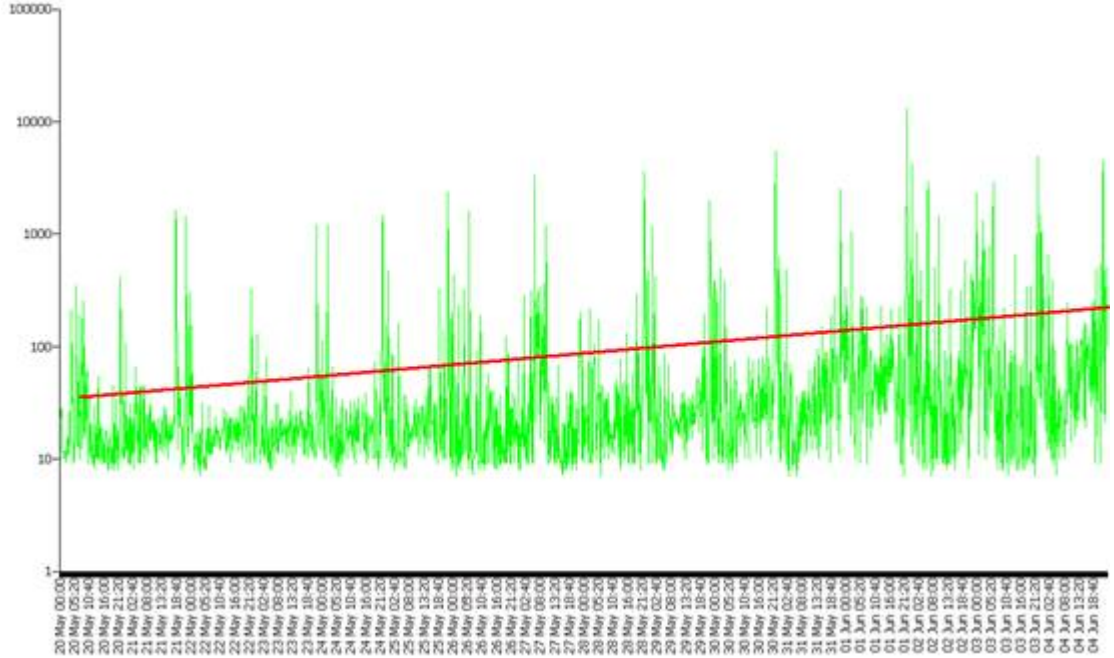
```
11. May
   371  zpool-Oracle_R\0

         value  ------------- Distribution ------------- count
           256 |                                         0
           512 |@                                        119079
          1024 |@                                        146371
          2048 |@@                                       311663
          4096 |@@@@@                                    886102
          8192 |@@@@@@@@@@@@@@@@                         2623578
         16384 |@@@@@@                                   1076413
         32768 |@@@@@@                                   1042969
         65536 |@@                                       319642
        131072 |@@                                       380511
        262144 |                                         0
```

Problemlösung

Schritt 2

- Nach „relocation“
  - AWR Werte für „AVERAGE WAIT„ von 14 ms
- Erneut langsame Verschlechterung der Performance



Problemlösung

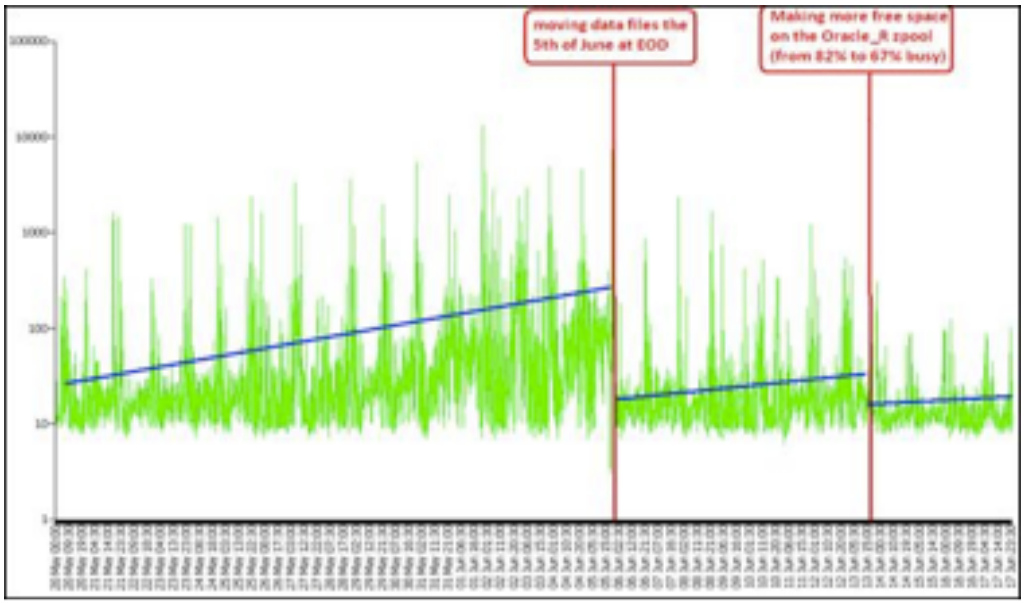
Schritt 3

- Oracle Consultant on site
  - Storage Layout nicht optimal
  - Grundproblem Fragmentierung ZFS
  - ZFS Patches sind nicht Ursache des Problems
- Empfehlungen
  - Weitere Performance Messungen
  - Überprüfen der „I/O sizes and disk subsystems“
    - „match“ Block Grössen ZFS, Datenbank, Disk-Subsysteme
  - Trennen von I/O Pfaden
    - So viele LUN‘s pro Zpool wie unabhängig drehende Disks
    - Unabhängige Zpool‘s für „database files“ und „redo logfiles“

Problemlösung

Schritt 4

- Benchmarks auf Test Systemen
  - „ganging“ ist abhängig von Füllgrad des ZFS Pools
- „freespace“ erhöht
  - „cleanup“ im Oracle Zpool



Problemlösung

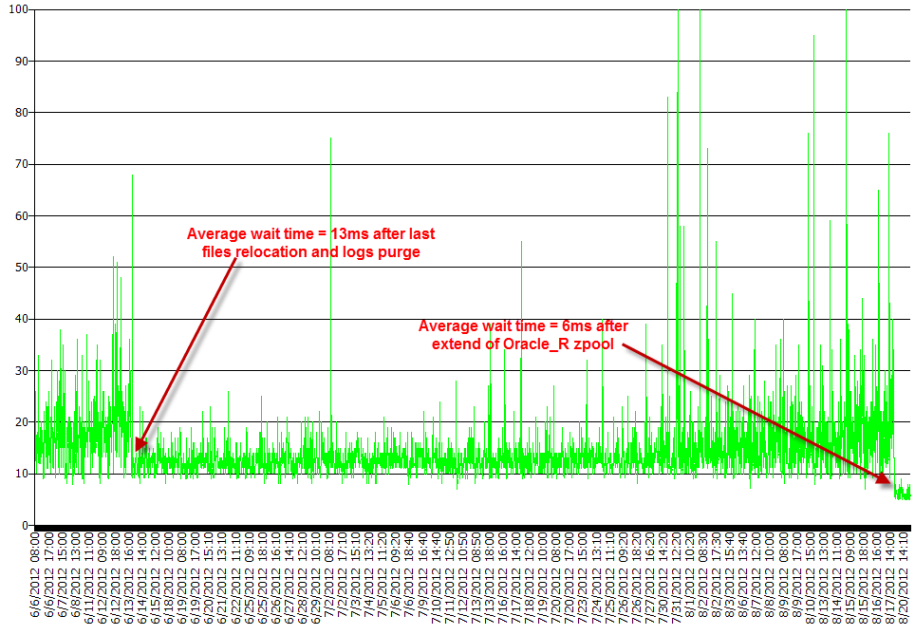
Schritt 4

- Messungen „ganging“

Problemlösung

Schritt 5

- „freespace“ so gross wie machbar
  - Auswirkung auf Dauer der „full replication“
- Fragmentierungsproblem eliminiert
  - bei „freespace“ > 50%



Fazit

Lösung sehr einfach

- Erhöhung des freien Platzes im Zpool auf > 50%
- Waren die Patches die Ursache des Problems?
  - Lief ca. ein Jahr ohne Probleme
  - „freespace“ war vorher bei ca. 80% über längeren Zeitraum stabil
- Verschweigt Oracle das ZFS Fragmentierungsproblem?
  - Kein Wort in Dokumentation
- Separate Zpool‘s für „data files“ und „redo logfiles“ verwenden
  - Storage Reorganistaion ist geplant
- Tool für Defragmentierung fehlt

VIELEN DANK!

DEN TRIVADIS STAND
FINDEN SIE AUF

EBENE 3,
STAND Nr. 304

Trivadis AG

Roman Gächter

Europa-Strasse 5

CH-8152 Glattbrugg

Tel. +41-44-808 70 20
Fax +41-44 808 7021

info@trivadis.com
www.trivadis.com

BASEL BERN LAUSANNE ZÜRICH DÜSSELDORF FRANKFURT A.M. FREIBURG I.BR. HAMBURG MÜNCHEN STUTTGART WIEN

trivadis
makes IT easier.