RAID-Optimierungen

...oder auch nicht...

Dirk Geschke

Linux User Group Erding

28. November 2012

Gliederung

Hardware

- Mainboard: Supermicro X9DRi-LN4+/X9DR3-LN4+/X9DRi-LN4+/X9DR3-LN4+, BIOS 1.0c 05/01/2012
- 2 CPUs mit jeweils 8 Cores plus Hyperthreading: Intel(R) Xeon(R) CPU E5-2690 0 @ 2.90GHz
- Hauptspeicher: 256 GB RAM
- RAID bus controller: LSI Logic / Symbios Logic MegaRAID SAS TB (rev 03) SAS 9266-8i
- 2 Backplanes
- 8 SSDs OCZ-VERTEX4, 256 GB (238 GB)

RAID-Controller LSI SAS 9s66-8i

- Eight-port 6Gb/s PCI Express 2.0 SATA+SAS RAID controller
- Eight internal SATA+SAS ports
- Two mini-SAS SFF-8087 x4 connectors - side mount
- x8 PCI Express 2.0 host interface
- 1GB DDRIII cache (1333MHz)

Theorie

- Controller 6Gb/s per lane
- 8 lanes $\longrightarrow$ 6 GB/s
- Jede Backplane ist via SFF-8087 mit 4 lanes verbunden

$$\Longrightarrow \text{3 GB/s per Backplane!}$$

- Benchmarks im Internet liegen in diesem Bereich!

RAID

- Redundant Array of *Indispensible***Independent** *Disks*
- *Verschiedene Level:*
  - RAID-0 *verteiltes Schreiben auf mehrere Platten, sehr schnell, unsicher*
  - RAID-1 *paralleles Schreiben auf mehrere Platten, langsam aber sicher*
  - RAID-5 *verteiltes Schreiben auf mehrere Platten mit Parität, schnell und sicher, eine Festplatte kann ausfallen, sehr langsam bei Ausfall.*
    Kombinationen möglich: RAID-10, RAID-50, . . .

Testüberlegungen

- 8 Consumer SSDs OCZ Vertex4 256 GB
- eine SSD schafft rund 400 MB/s schreiben/lesen
- RAID-0: gleichzeitiges Schreiben auf alle Disks
- Maximale Performance: rund 3.2 GB/s

Tools

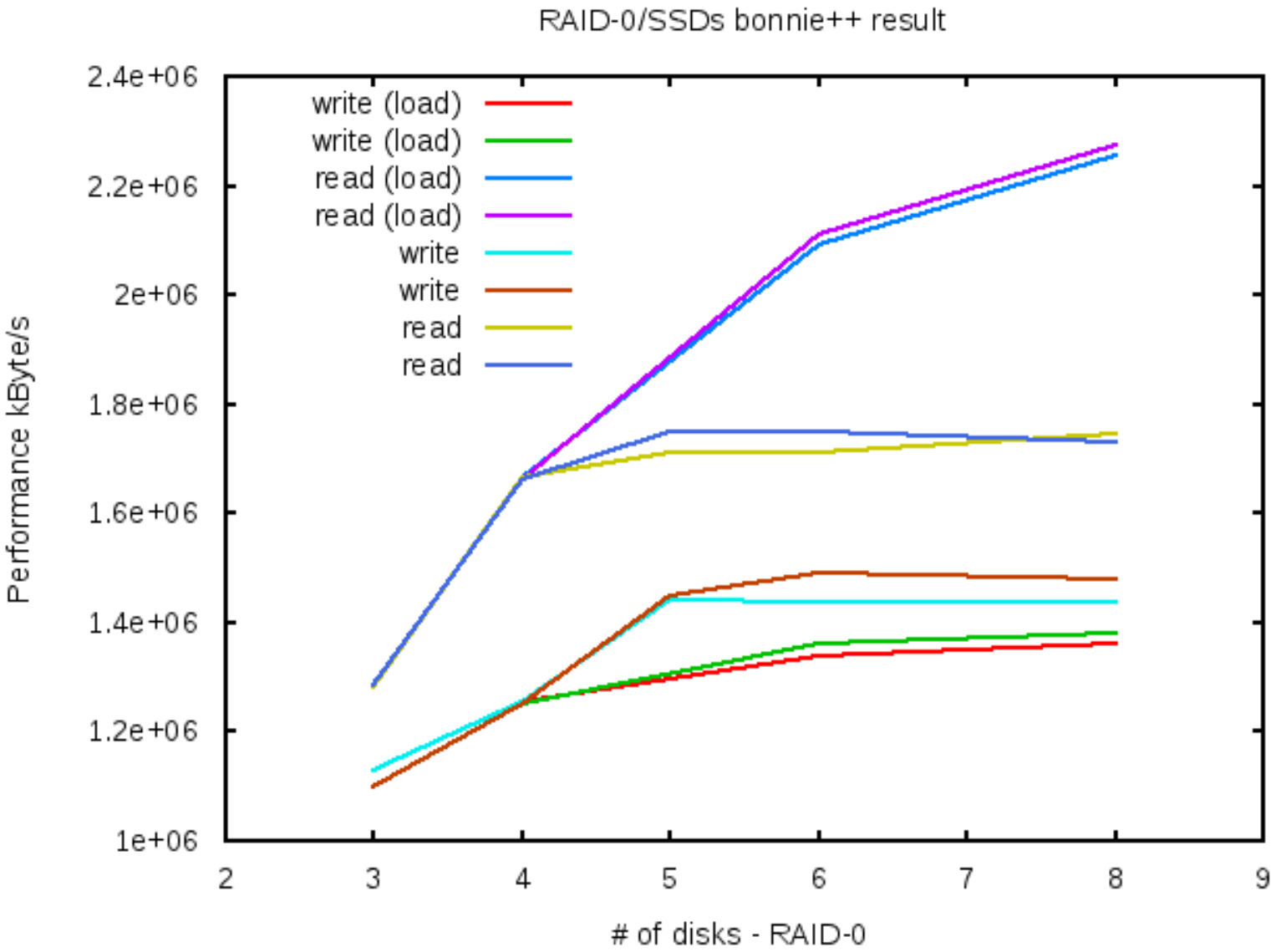
hdparm einfaches Tool zum direkten Lesen von Daten, eigentlich SATA/IDE

dd direktes Lesen und Schreiben möglich, mit `conv=fdatasync` oder `oflags=sync` ohne Buffereffekte des OS

IOzone sehr umfangreich und aufwändig, ein Testdurchlauf kann mehrere Tage dauern

bonnie++ einfacheres Tool als IOzone, durchaus ausreichend und schnell

RAID-0 mit unterschiedlicher Zahl an SSDs



Beobachtungen und Überlegungen

- niedrige Performance liegt nicht an den SSDs, da müsste mehr möglich sein
- hohe Leserate unter Last ist seltsam, Schreibrate am niedrigsten
- besser auf Schreibraten konzentrieren
- Vielleicht falsches Tool?
- Oder liegt es am Dateisystem?
- Kann man noch etwas tunen?

Erste Vergleiche

- `dd` liefert ungefähr gleiche Werte
- `hdparm` ist schlechter, mit O_DIRECT (`-direct`) besser: 2 GB/s lesen
- `ext4` statt `xfs` verbessert nichts:

Tabelle: RAID mit 6 SSDs

Filesystem	write	read
ext4	1186652	1713403
ext4	1151185	1739247
xfs	1519895	1742583
xfs	1447835	1740164

- Tests mit `dd` auf dem `raw`–Device zeigt auch keine Änderungen
- Anpassungen beim `xfs` der *stripe size* mit `-d su=256k -d sw={Zahl der Disks}` ändert nichts

2 Backplanes nutzen?

- verteilen von jeweils 4 SSD auf eine Backplane
- parallele Nutzung beider Backplanes:

Tabelle: RAID-0 mit 2x4 SSDs

Messung	write	read
1.	1419002	1818083
2.	1441311	1801801

- keine wirkliche Verbesserung!

Idee: Testen der einzelnen Backplanes mit 4 SSDs

- erst 4 SSDs als RAID-0 auf der ersten Backplane
- anschließend 4 SSDs als RAID-0 auf der zweiten Backplane
- Ergebnis:

Tabelle: RAID-0 4 SSDs auf einer Backplane

Backplane	write	read
1st	1407529	1746556
2nd	1405938	1726442

- 4 SSDs: Es ergibt keinen Unterschied!

Idee: 2xRAID-0 mit 4 SSDs als RAID-00

- RAID-00 ist ein RAID-0 mit zwei RAID-0 als Basis (statt 2 HDs)
- Realisierung per LVM mit 2 stripes (2 PVs in einer VG)
- Ergebnis:

Tabelle: (Software-) RAID-00

write	read	Kommentar
1428675	3012491	
1447650	2992583	cachedBBU
1527926	2999098	Read Ahead statt Adaptive RA
1579117	3039371	bonnie -s512g:256
1592776	3011215	bonnie -s512g:256
1733286	2179877	bonnie -s512g:256 (Last?)

- Leseperformance durch LVM-Read-Cache?

Idee: zeitgleiche Benchmarks auf 2 x 4 SSDs

- Wenn die hohen Leseraten von der Hardware stammen, dann müsste diese auch bei parallelen Benchmarks zu sehen sein.
- 2 RAID-0 a 4 SSDs und zeitgleiche Benchmarks auf jedem RAID-0
- Ergebnis:

Tabelle: zeitgleicher Benchmark, 2 RAID-0, kB/s

Backplane	write	read
1st	1387159	1273411
2nd	1412843	1337237
Summe	2800002	2610648

- sieht eigentlich gut aus?!?

Idee: zeitgleiche Benchmarks auf 2 x 4 SSDs, gleiche Backplane

- 2 RAID-0 zu je 4 SSDs auf der gleichen Backplane
- Ergebnis:

Tabelle: zeitgleicher Benchmark, 2 RAID-0, gleiche Backplane

write1	read1	write 2	read 2
918887	1041548	909917	1063715
917272	1085540	908869	1077413
1836159	2127088	1818786	2141128

- sieht interessant aus. . .
- Aber wir haben ja noch ein paar Ideen...

Elevatoren

- können via `/sys/block/{Device}/scheduler` eingestellt werden.
- je nach Scheduler können weitere Optimierungen möglich sein
- default: `cfq`
- Ergebnis, wieder 8 SSDs:

Tabelle: Elevator: `noop`

write	read
1319920	1868901
1490747	2015505
1437696	2263168
1404564	1907931

Kernel 3.6 mit minimalen Treibern

- Deaktivierung der automatischen *cgroups*
- Ergebnis:

Tabelle: Kernel 3.6

Elevator	write	read
cfq	1618012	1644082
noop	1551156	1783303
deadline	1370977	1591597

- nicht wirklich besser. . .

Analyse Mainboard Manual - Slots

- Es gibt einen `CPU1 Slot2 PCIe 3.0x4 (in x8 Slot)`
- PCIe 2.0 unterstützt der Controller $\longrightarrow$ 5 GT/s bei 2.5 GHz
- **x4** liefert bei **8b10b**-Kodierung 2 GB/s
- $\longrightarrow$ könnte eine Erklärung sein!
- `lspci` liefert aber:

  `LnkSta: Speed 5GT/s, Width x8...`
- damit sind 4 GB/s via PCIe-Bus möglich!

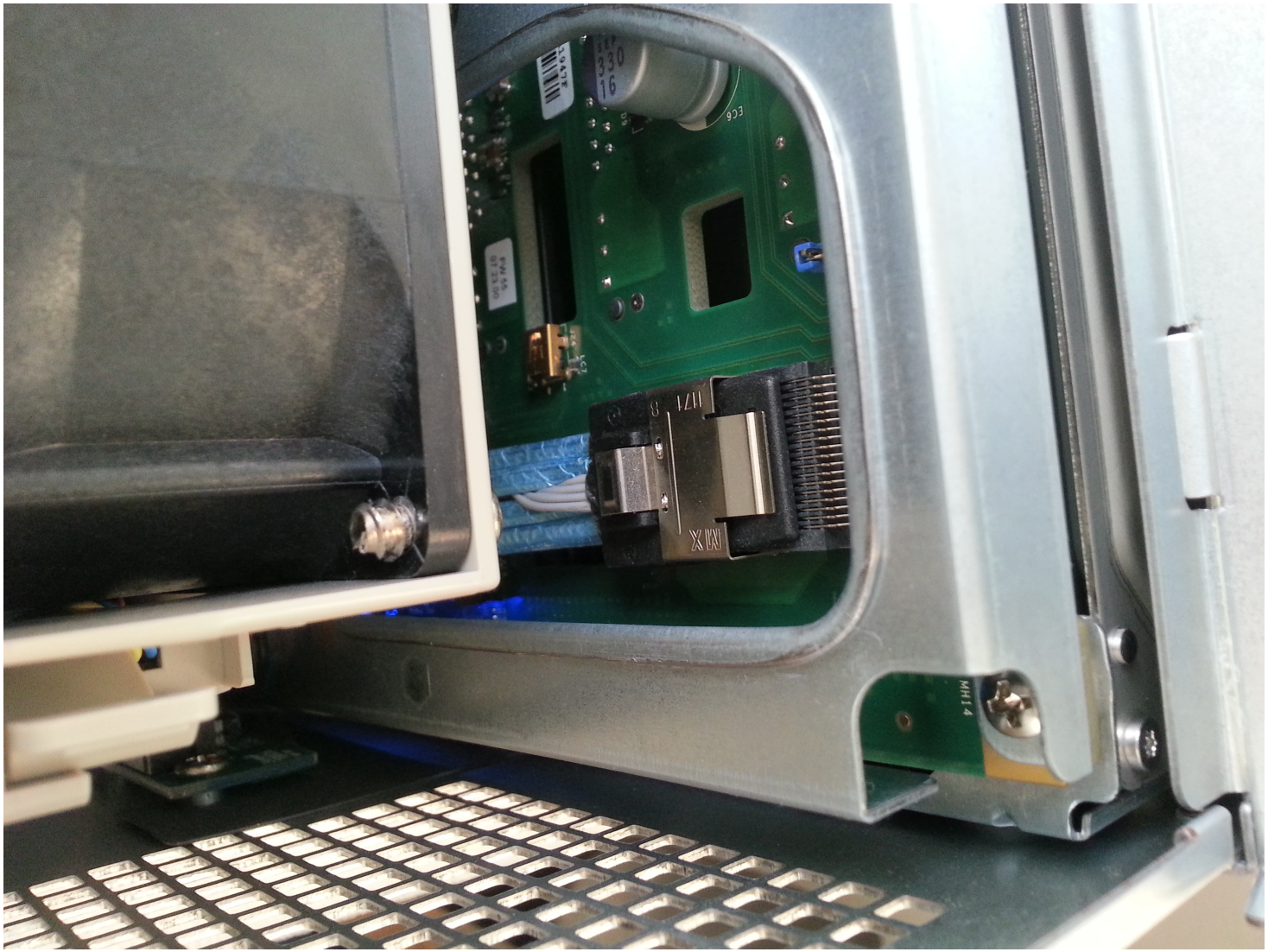
Analyse des LSI-RAID-Controllers

- Ausgabe von `MegaCli -AdpAllInfo -a0` liefert:

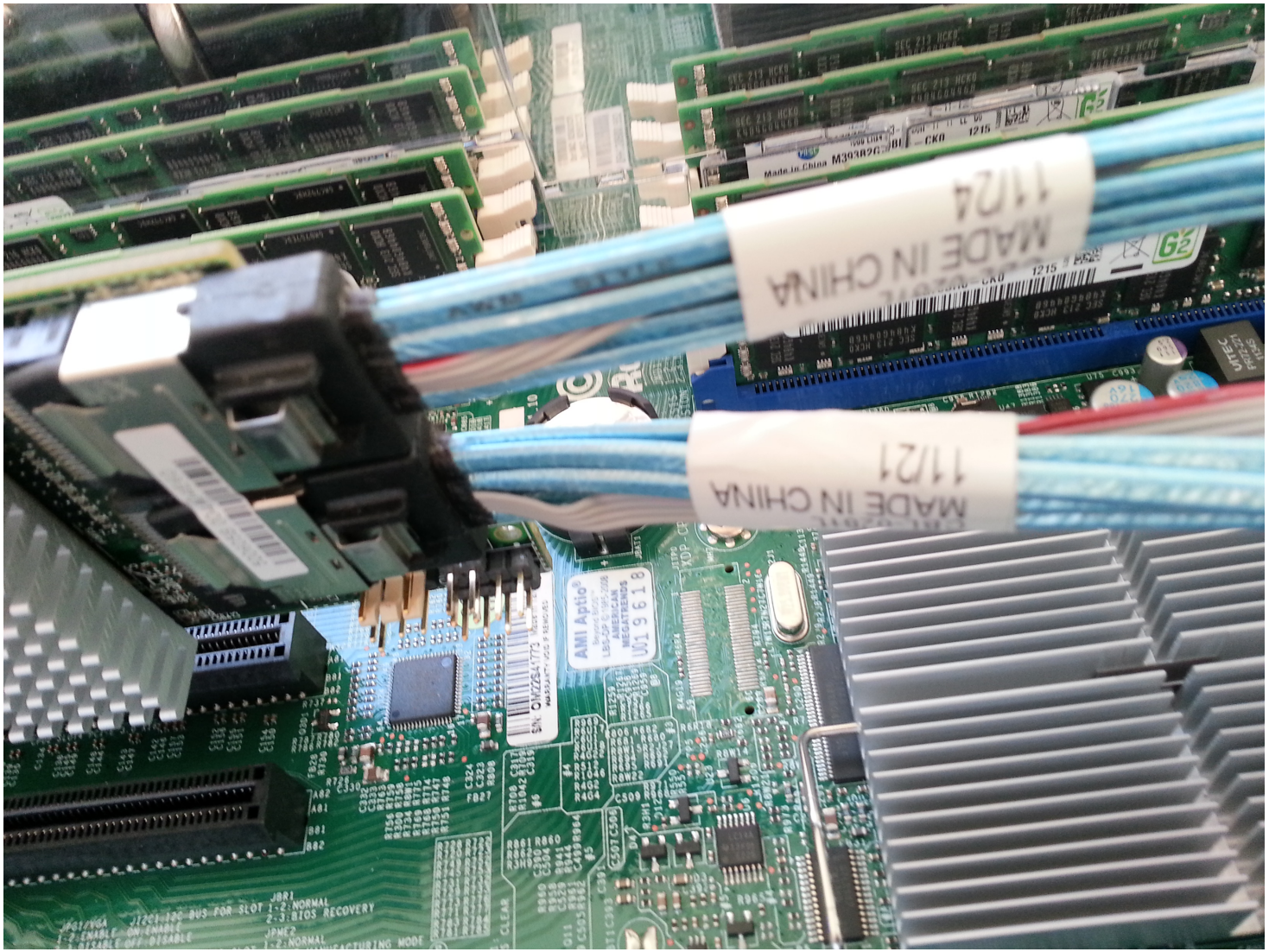
```
Number of Backend Port: 8
Port : Address
0 5003048001a1947f
1 5003048001b511ff
2 0000000000000000
3 0000000000000000
4 0000000000000000
5 0000000000000000
6 0000000000000000
7 0000000000000000
```

- 2 Ports? 2 $\times$ 6 GBit/s = 1.5 GB/s, alle Kabel an eine Backplane und zweite via Bridging an die erste?

LSI-Controller im Gehäuse



LSI-Controller im Gehäuse



LSI-Controller im Gehäuse

- Kein Bridging, jede Backplane ist mit 4 *lanes* angeschlossen
- Aussage von LSI: Ausgabe von *Ports* betrifft nur Anschlüsse auf dem Controller, nicht identisch zu *lanes*.
- CPU Slot ist nun auch klar:
  `CPU2 Slot4 PCIe 3.0 x16`
- Idee: Slot hängt an CPU2, IRQs (Modul `megasas` in `/proc/interrupts`) werden aber von CPU1 verarbeitet!

Idee: IRQs an zweite CPU verschieben

- verschieben kann über `/proc/irq/{irq}/smp_affinity` oder `/proc/irq/{irq}/smp_affinity_list` erfolgen.
- Ergebnis:

Tabelle: IRQ via zweiter CPU

write	read
1531611	1764494

- effektiv kein Unterschied!

Idee: IRQs über alle Cores

- es sind 16 IRQs, es gibt 16 physikalische Cores
- Ergebnis:

Tabelle: IRQ verteilt auf 16 Cores, `noop`

write	read
1611591	1762749
1591267	1724631
1558705	1809124

- ein bisschen hat es geholfen, aber nicht wirklich!

Idee: IRQ-Verteilung bei parallelem Schreiben

- Vielleicht bringen die IRQs etwas, wenn man parallel schreibt?
- Ergebnis:

Tabelle: IRQ verteilt auf 16 Cores, 4/8 disks, `noop`

write 4	read 4	write 8	read 8
926801	1097489	870560	1126598
924409	1115369	869335	1123164
1851210	2212858	1739895	2249762

- 8 Disks sind langsamer beim Schreiben? Interessant...

Priorisierung mit ionice

- `cfq` bietet die Möglichkeit mit `ionice` den Durchsatz zu steuern, schellste Version mit `ionice -c 1 -n 0 ...`
- Zum Vergleich Messungen mit `cfq`, `noop` und `deadline`
- Ergebnis:

Tabelle: `cfq` mit `ionice` versus `deadline` und `noop`

write	read	
1615423	1731223	cfq mit ionice
1034862	1666484	deadline
1601699	2094683	noop

- `noop` scheint am besten zu sein, sofern das System *idled*

WriteThrough statt WriteBack

- LSI hat WT statt WB vorgeschlagen
- WriteBack schreibt in den Cache, der Cache auf die Platte
- WriteThrough schreibt direkt auf die Platte, speichert die Daten aber im Cache.

Tabelle: WriteBack vs. WriteThrough mit `noop`

write	read	
1615423	1731223	WriteBack (s.o.)
1497058	2314050	WriteThrough
1448679	2234858	WriteThrough

- Es verbessert das Lesen, verschlechtert das Schreiben. . .

No Read Ahead

- weiterer Vorschlag von LSI: `ReadAhead` deaktivieren
- endlich ein deutlicher Effekt:

Tabelle: Kein Read Ahead mit `noop`

write	read	
1604035	496854	
1467815	483319	
1508270	501029	IRQs verteilt

- Leserat bricht dramatisch ein!

Neue Firmware für LSI 9266-8i

- Neuere Firmware des RAID-Controllers war erschienen

Tabelle: Read Ahead mit neuer Firmware und `noop`

write	read	Cache
1504866	1993970	?
1479499	1755349	?
1565605	1648603	?
1546453	1654260	WriteBack
1590832	1679014	WriteBack
1335200	1775061	WriteThrough

Es wird exotisch

- `cfq` bietet Optionen für *slice_idle* und *group_idle*: längere Verweildauer von Prozessen in der I/O-Queue für weitere Aufgaben.
- Idee dahinter: Der Schreibkopf steht schon ideal

Tabelle: `cfq`–Optionen

write	read	
1588987	1682927	slice_idle=0
1559710	1684906	slice_idle=0, group_idle=0
1534921	1685089	mit ionice priorisiert

- $\longrightarrow$ kein spürbarer Effekt!

Spielen mit der Strip–Size

- per Voreinstellung beträgt die Strip-Size 256kB (Controller)
- Idee: Variation der Größe

Tabelle: Effekt der Strip–Size

Strip–Size	write	read
8kB	473883	990133
64kB	1225679	1973300
64kB	1370498	1686235
256kB	1559710	1684906
1MB	1164984	2177400
2MB/256kB	1551860	1804623

- $\Longrightarrow$ der *default* ist in Ordnung.

Noch einmal `dd` zum Vergleich

- verschiedene Tests mit allen SSDs auf gleicher Backplane
- Testen mit `dd`, 2×Raid-0 mit 4 SSDs
- Lesen:

Tabelle: paralleles Lesen von 2 RAID-0 in MB/s

1st	2nd	
876	888	bs=1M, 1GB
943	950	bs=1G, 512 GB
1212	—	nur ein RAID

- 1.2 GB/s sind für 4 SSDs ok.
- maximale Leserate rund 1.8 GB/s

Noch einmal `dd` zum Vergleich

- Schreiben liefert:

Tabelle: paralleles Schreiben von 2 RAID-0 in MB/s

1st	2nd	
880	878	bs=1G, 512 GB
1314	—	nur ein RAID

- 1.3 GB/s sind für 4 SSDs ok. Die
- maximale Schreibrate rund 1.8 GB/s

Noch einmal `dd` zum Vergleich

- Schreiben von 512 GB liefert:

Tabelle: 2× paralleles Schreiben von 4 RAID-0 in MB/s

1st	2nd	3rd	4th	
888	849	893	845	raw-device (OS kombiniert schon?)
451	451	454	452	in separate Dateien

- maximale Schreibrate rund 1.8 GB/s

Noch einmal `dd` zum Vergleich

- Schreiben von 256 GB auf RAID-0-Systeme aus je 2 SSDs liefert:

Tabelle: paralleles Schreiben auf RAID-0 aus 2 SSDs in MB/s

1st	2nd	3rd	4th	Summe	
737				737	Limit der SSDs
768	700			1468	Limit der SSDs
590	590	590		1770	
448	447	447	447	1789	

- maximale Schreibrate bleibt bei 1.8 GB/s

Zwischenstand

- Die Performance ist weit hinter den theoretischen Werten.
- Fast alle Optimierungen auf OS- und Controller-Seite zeigen keine deutliche Verbesserung.
- Interessant sind die Ergebnisse beim parallelen Schreiben auf beide Backplanes.
- **Vermutung**: Irgendwo ist ein Engpass bei der Hardware. Daher zeigen auch die vielen Optimierungen keinen spürbaren Effekt.
- Suche nach Benchmarks im Internet zeigt: 1.8 GB/s Schreibperformance.
- Allerdings war das auch die Grenze der verwendeten SSDs.

Test mit `/dev/null` und `/dev/zero`

- Vielleicht liegt die Grenze auch beim RAM?
- Test mit dd:

Tabelle: Schreiben von `/dev/zero` nach `/dev/null`

Blocksize	Count	Performace
1 GB	1	1.4 GB/s
1 GB	2	1.6 GB/s
1 GB	3	2.0 GB/s

- Das klingt interessant, es ist nicht so weit vom RAID entfernt!

Simple Disk Benchmark

- Fokus auf sequentiellem Lesen und Schreiben $\longrightarrow$ Simple
- vor der Messung wird der Speicher allokiert und initialisiert
- Buffergröße einstellbar
- `fsync(), O_DIRECT, O_SYNC` nutzbar
- funktioniert auch mit Dateien $<<$ RAM
- Threads: paralleles Lesen und Schreiben möglich
- Buffer kann randomisiert werden.
- Timing-Werte für `memset(), memcpy(), bcopy(), memmove()` möglich
- Verbose-Modus: zahlreiche relevane Systemparameter werden gelistet

Ergebnisse mit sdb - /dev/null und /dev/zero

- Test mit /dev/null und /dev/zero:

Tabelle: Ergebnisse für 4 GB Datenmenge

Zugriffsart	Ziel/Quelle	Durchsatz
write	/dev/null	148.30 GB/s
read	/dev/zero	10.08 GB/s
write	ramfs	3.81 GB/s
read	ramfs	5.00 GB/s

- Unterschied zu dd ist offensichtlich.
- RAM-Disk ist irritierend, liegt im Bereich des RAID-Controllers

Ergebniss mit sdb

- Test einer RAM-Disk mit 4 Threads:

Tabelle: Ergebnisse - write - RAM-Disk und Cores

Durchsatz	mittel	Cores pro CPU	Größe
10.56 GB/s	2.56 GB/s	3 + 1	256m
8.82 GB/s	2.20 GB/s	4 + 0	256m
11.89 GB/s	2.97 GB/s	3 + 1	256m
13.85 GB/s	3.46 GB/s	2 + 2	256m
15.53 GB/s	3.88 GB/s	2 + 2	256m/2g
9.75 GB/s	2.44 GB/s	4 + 0	256m/2g

- sieht interessant aus: RAM einmal prüfen!

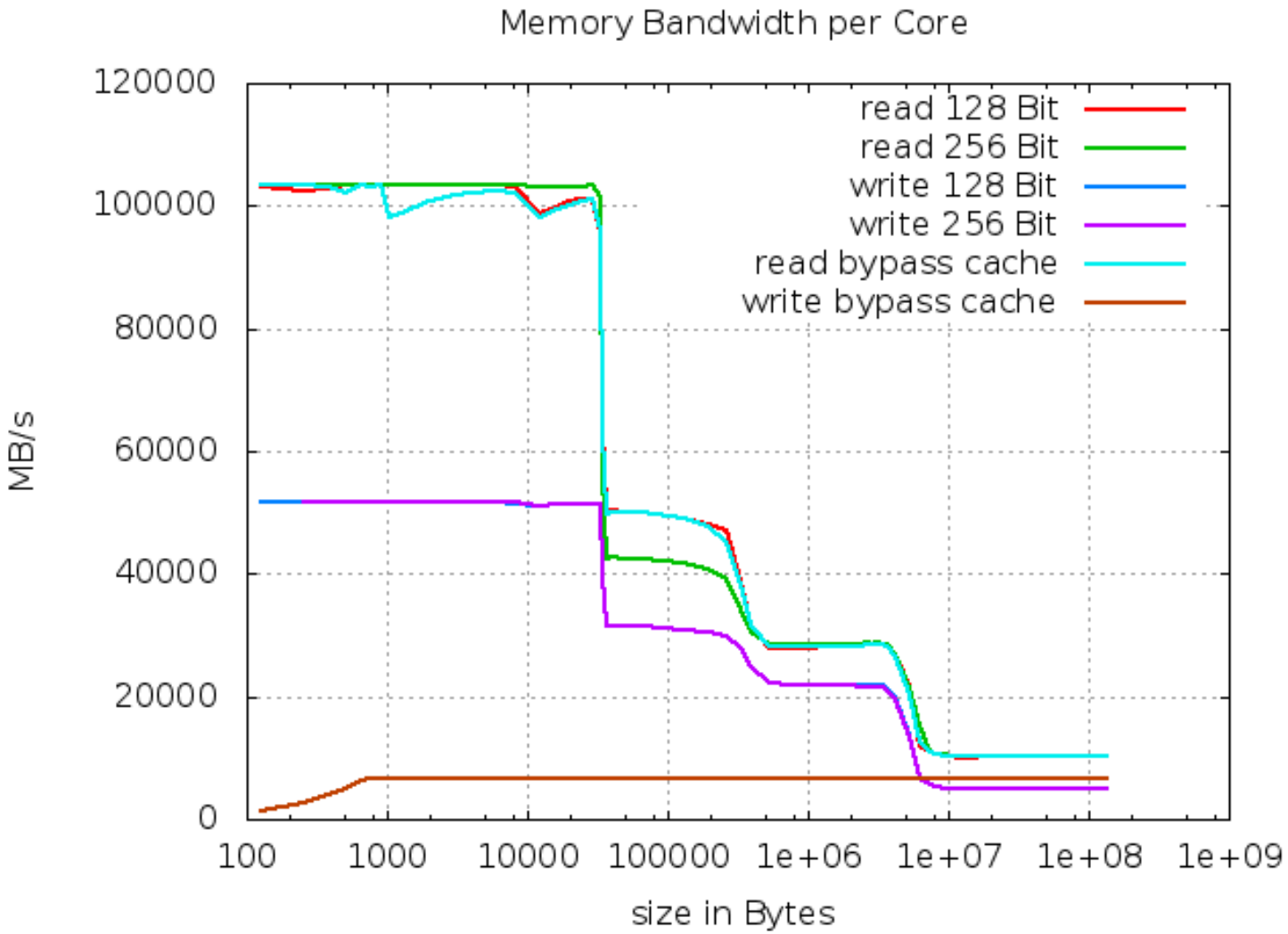
Exkursion RAM – Speicherbenchmarks

- Theoretische Grenze von DDR3-1600: 12.8 GB/s
- 4-channel sollte sogar vervierfachen: 51.2 GB/s
- 2 CPUs, also noch einmal verdoppeln? $\longrightarrow$ 102.4 GB/s
- Intel sagt zur CPU:
  - L1-Cache: 32 kB
  - L2-Cache: 256 kB
  - L3-Cache: 20 MB (shared)
  - Max. Speicherbandbreite: 51.2 GB/s, 4 Speicherkanäle

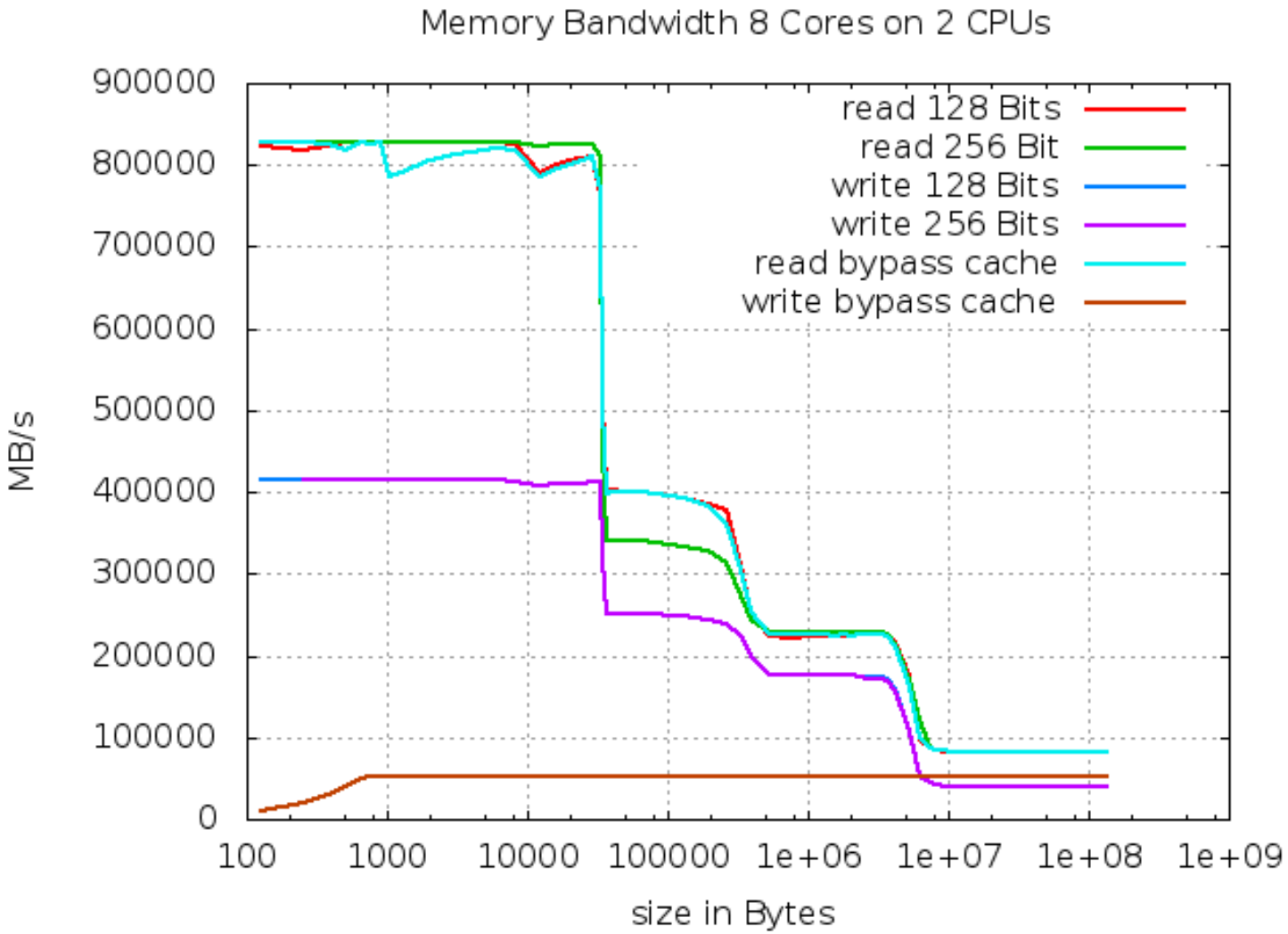
Speicherbenchmarks

- Messung mit Boardmitteln schwierig.
- 2 Tools, assemblerbasiert:
  - **bandwidth**: `http://zsmith.co/bandwidth.html`
  - **ramsmp**: `http://www.alasir.com/software/ramspeed`
- $\Longrightarrow$ beide liefern nahezu identische Zahlen.

Benchmark mit einem Core



Benchmark mit 8 Cores auf 2 CPUs



Ergebnis Memory-Test

- Performance ist per Core limitiert
- Schreibperformance liegt im Bereich von rund 5 GB/s je Core
- sollte zum Testen des RAIDs ausreichen
- Problem mit `bonnie++`: benötigt relativ lang.
- vorerst interessiert nur sequentielles Schreiben und Lesen
- Problem mit `dd`:
  - misst Zeit für Lesen und Schreiben
  - rechnet mit Vielfachen von 1000 statt 1024
- daher neuer Ansatz mit: `sdb`

Simple **D**isk **B**enchmark

Ergebnisse mit sdb - RAW-Disk

- Test mit RAW-Disk und großen Dateien:

Tabelle: Ergebnisse ohne Dateisystem, 10 und 100 GB

Zugriffsart	Durchsatz	
read	1.98 GB/s	ReadAhead
write	1.85 GB/s	WriteBack
write	1.71 GB/s	WriteThrough

- Ergo: Durchsatz im Bereich von 1.8-2.0 GB/s pro Backplane

Ergebniss mit sdb - Dateigröße

- interner Cache des Controllers: 799 MB

Tabelle: Schreibergebnisse abhänging von der Größe

Größe	Durchsatz
64 MB	2.83 GB/s
128 MB	2.90 GB/s
256 MB	3.17 GB/s
512 MB	3.17 GB/s
1024 MB	2.73 GB/s

- Bestwert bislang mit 2 Threads: 3.36 GB/s

Ergebnisse mit sdb - XFS

- Test mit xfs:

Tabelle: Ergebnisse mit Dateisystem

Zugriffsart	Größe	Durchsatz
write	256 MB	3.16 GB/s
write	10 GB	1.85 GB/s
write	100 GB	1.76 GB/s
read	10 GB	1.94 GB/s
read	100 GB	1.94 GB/s

- Dateisystem stellt kein Problem dar.

Zwischenstand

- Durchsatzgrenze pro Backplane: 1.8-2.0 GB/s
- LSI spricht von 6 GBit/s per lane, 4 *lanes* pro Backplane: 3 GB/s
- Durchsatz zum Controller-Cache: 3.2-3.4 GB/s
- theoretische Grenze von PCIe 2 x8: 4 GB/s

$\Longrightarrow$ **Da passt etwas nicht!**

PCIe 2

- Durchsatz 5 GT/s pro lane
- **x8** und **8b10b**-Kodierung: 4 GB/s
- Protokolloverhead: 26 Bytes (Header, Squenznummer, CRCs)
- Payload aus `lspci`:

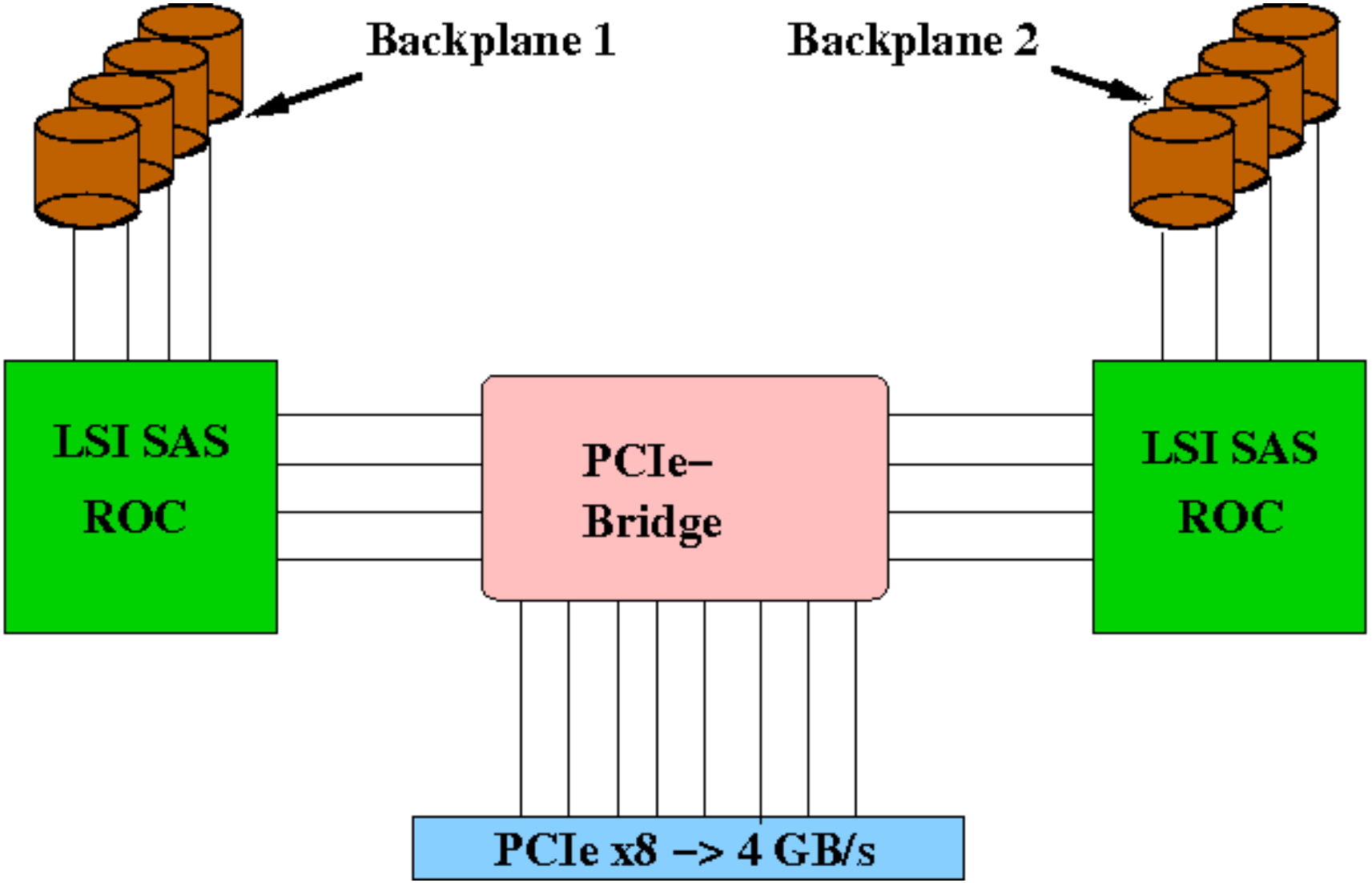
```
DevCap: MaxPayload 4096 bytes, ...
DevCtl: ...MaxPayload 256 bytes, ...
```

- Das sind rund 10% Protokolloverhead!
- Protokollstatus- und Steuerpakete spielen auch eine Rolle
- Overhead vermutlich bei 15-20%

$\longrightarrow$ 3.2-3.4 GB/s realistisch!

$\Longrightarrow$ **Controller-Cache-Performance**!

Theorie zum LSI-Controller



Theorie zum LSI-Controller

- durchreichen des Taktes $\longrightarrow$ 2 GB/s und Backplane
- passt zu beobachteten Ergebnissen!
- aber: 3.2-3.4 GB/s per PCIe 2 x8 $\longrightarrow$ 1.6-1.7 GB/s
- gemessen haben wir bessere Werte!
- dennoch: klingt plausibel!

Auskunft von LSI

- realistische Werte bei PCIe 2 x8: 3.2-3.4 GB/s
- Controller arbeitet intern nicht mit PCIe
- es findet dennoch eine **8b10b**-Kodierung statt
- $\longrightarrow$ 6 GBit/s sind 4.8 GBit/s per *lane*.
- Protokolloverhead auch hier vorhanden
- realistisch laut LSI: 2.0-2.2 GB/s
- mit 1.8-2.0 GB/s sind wir gar nicht so schlecht!
- sollte man einen PCIe-3-Controller verwenden?

Tests über beide Backplanes

- schwierig, da auf einer Backplane SSDs auf der anderen HDDs
- zweites HDD-RAID-0 mit gleichem Durchsatz erstellt: 14 Disks
- Durchsatz unterschiedlich!

Tabelle: Ergebnisse paralleles schreiben, 2 Backplanes

SSD	HDD	Durchsatz	
1.28 GB/s	1.57 GB/s	2.85 GB/s	WriteThrough, 10 GB
1.41 GB/s	1.40 GB/s	2.81 GB/s	SSD-WriteBack, 10 GB
1.37 GB/s	1.37 GB/s	2.74 GB/s	SSD-WriteBack, 100 GB
1.57 GB/s	1.59 GB/s	3.16 GB/s	WriteBack, 256 MB

Tests über beide Backplanes - Upgrade Firmware

- neue LSI-Firmware, neuer Kernel 3.7.0-rc6

Tabelle: Ergebnisse 2 Backplanes

SSD	HDD	Durchsatz	
1.39 GB/s	1.76 GB/s	3.15 GB/s	write, WriteThrough, 10 GB
1.34 GB/s	1.48 GB/s	2.83 GB/s	read, 10 GB
2.39 GB/s	2.39 GB/s	4.77 GB/s	write, 10 GB, OS-Cache

- Durchsatz noch immer unterschiedlich!
- Schreibrate passt nun zu PCIe 2
- Leserate kleiner, laut LSI ist das normal.

Fazit RAID–Controller

- Katz', alles für die?
- bis jetzt nur sequentielle Zugriffe getestet
- **FastPath** und **CacheCade** existieren auch noch...
- **CacheCade** derzeit auf 512 GB beschränkt
- Marketing ist alles?
- immerhin: viel über Hardware und Linux gelernt!
- out-of-the-box ist Linux schon sehr gut!

Studie: PC-Kenntnisse gehen in Deutschland zurück

. . . Im Interview mussten sie ihre Fähigkeiten selbst einschätzen und Angaben zu verschiedenen Kriterien machen . . .

Quelle: `http://heise.de/-1644936`

Ich habe auch **keine** PC-Kenntnisse!

Sch. . . Marketing!