

Monday, February 27, 2012

Being relaxed in regard of retention

Robin Harris of StorageMojo pointed to an interesting research paper: "Optimizing NAND Flash-Based SSDs via Retention Relaxation". The idea of the method described in that paper is basically: The physics behind the process of writing to NAND-flash allows you to write you faster, when you shorten the retention time. The shorter the retention time, the faster the write. Retention time? That the span of time you have to be capable to read the data from a device.

Obviously I mapped that automatically into the ZFS world: For an separated ZIL device write latency is a really, really important metric. However: The normal live time of the data written to the sZIL? Just until the transaction group has been committed to the pool. A few seconds. However the need retention time is not that short, because you have address failures. You write something like a ZIL for a reason. In the case of a system failure the data has to survive at least as long until the same or a different system is able to commit the data - that wasn't committed before the failure - that is still on the sZIL to the pool devices. However how long is this? A day ... perhaps a week. But surely not the standard of 1 year or 10 years. However the standard mechanism writes so it could be read for 1 or 10 years as mandated by some standard bodies. And this is exactly the margin the idea in the paper uses. The model in the paper suggest a write speedup of 2.5 in response time for a TPC-C-similar load (figure 13) when you target for a retention time of two weeks instead of a year.

When you combine sZIL and flash storage capable of such a relaxed retention you could basically use such an optimization for pretty much all data storage needs, as the data is still stored in the pool. Further more you can forget about the refreshment of the data as suggested by the paper for data with longer retention time needs, because by it's nature all data is really short-lived and other processes take care of making the data persistent for the next 10 years. And as a weird idea at the end: You could even use a retention-relaxed SSD as an sZIL device in front of a pool of SSD without relaxed retention.

Posted by Joerg Moellenkamp in English, Solaris at 21:16

Using encrypted lofi devices for backup

My colleague Constantin wrote a nice article about using encrypted loopback file based pools for backup purposes for example. And don't forget you can do deduplication or snapshots as well for your backedup data. And in case you don't know how migrate your data into this files ... use shadow migration

Posted by Joerg Moellenkamp in English, Solaris at 16:26

Sunday, February 26. 2012

More than this

I had many interesting discussions in the last few days. However some of them gave me the impression that i should explain one thing. I want to use one example for it: A customer asked me while having some coffee at the Tech Days: "Zones? Isn't that just jails like in FreeBSD?". It's that old question, i get since the introduction of Zones. Just to make it clear: It's not a text in regard of FreeBSD. It's a text about the tendency of people just to pick a single feature and to say "Isn't that feature like ...". But it isn't that easy.

I don't want to discuss about that question, if a feature is really like another. However i want to introduce a different thought into this: "A feature is not only a feature when you have an overarching architecture and an overarching idea where the architecture is heading to. It's an enabler. And a feature is often just the next question".

A feature is just the next question? Yes, that right. Because every feature you introduce is just the starting point for the next feature. When we introduced Zones years ago, we had a lot open questions not much later: How do you patch those zones? Especially when you have dozens of them? How to delegate administration? How do you install zones in a fast manner? How to implement bootenvironments, for the OS as well as the Zones? How do you reduce hard-to-find problems of an architecture that shares a kernel, but has several copied userlands? Questions, that have perhaps no technical background, but resulting in technical changes because of operational requirements. Perhaps, at the beginning Jails and Zones may have been similar concepts. But when you look today into the construct, Zones is a lot more. Zones is a large interdependent web of features inside of zones and outside of zones to enable customers to work with them as efficient as possible.

However: Some of the challenges are just solvable when you have an overarching architecture and the power to decide on the architecture. And this is what i want to say with "a feature is not only a feature". Sometimes it's an enabler for a different feature. For example you have to be capable to say "ZFS is the only filesystem for booting and keeping zone roots" then you have a foundation you can use to implement other features. You can take all the mechanisms of ZFS for example for granted to base other features on it.

An example from the automobile sector? Ever asked why the automatic parking for a VW Golf is that cheap? Well, it just reuses an electro motor that it's there for power steering. You just need some software and a computer to give directly orders to this already existing power steering motor? When you allow all engineering teams to use it's own power steering implementing automatic parking in all vehicles is much more difficult.

Or in Solaris: Fast zone cloning? You need snapshots for it. So go to the ZFS people. Exclusive IP-Stack for a zone? You need a revamped IP stack for it. Ask the Crossbow people. Many things are severely depending on each other? Boot environments like in Solaris 11? Just feasible with a filesystem capable to do snapshots. IPS was invented to a part to have a packaging format that is much more aware of a concept like zones than just post/pre install scripts where zones was just an afterthought. Bandwidth Management? The Crossbow people again. Resource Management for your zones? You could use the foundation already laid out by the people with the Solaris Resource Manager years ago.

And there i'm at the point where i'm saying it doesn't suffice to have a feature, for example like Jails. At the introduction of a feature, a new journey just begins to solve all implication of a new feature. And you have to go all the way to make it really good.

Posted by Joerg Moellenkamp in English, Solaris at 08:36

Tuesday, February 21. 2012

Solaris 11 in Common Criteria evaluation

As i got the question on the Solaris 11 Techdays: As reported by the Oracle Tech Network Solaris 11 is now in evaluation under the Operating System Protection Profile using the extended packages Advanced Management, Extended Identification and Authentication, Label Security, and Virtualization at level EAL4+.

Posted by Joerg Moellenkamp in English, Solaris at 08:12

Saturday, February 18. 2012

Veranstaltungshinweis für Düsseldorf

Ich möchte noch mal einen Veranstaltungshinweise loswerden. Mein Kollege Michael Faerber organisiert für den 8.3 ein Oracle Breakfast in Düsseldorf. Das ist quasi "Vorträge mit Mampf". Zwei Vorträge sind für dieses Event geplant. Von 09:15 bis 10:30 "Einsatz und Administration von LDOMs" und von 10:45 "Datenmanagement mit ZFS unter Solaris 11". Bei beiden Vorträgen wird es nur wenig Folien, aber dafür sehr viele Livevorführungen am Objekt geben. Beide Vorträge werde ich halten. Wenn Ihr kommen möchtet, schickt bitte eine Mail an oraclebreakfast_dus@c0t0d0s0.org. Da ist ein Forwarder an den Kollegen hinter. Anmeldeschluss ist der 6. März. Weiss nicht ob es der Kollege so toll findet, wenn ich seine Mailadresse hier poste

Für die Solaris 11 Tech Days am 28. Februar in Zürich kann man sich übrigens immer noch anmelden: Solaris 11 Tech Days 2012

Posted by Joerg Moellenkamp in General at 14:50

So far ...

It's a few days since the event in Munich, the last one in the series in Germany. The last two weeks were really cool. There were many many people at the Solaris 11 Techdays 2012. That are the moments where i really love my job. Speaking in front of an interested audience about technology. I hope that Zurich will be as great as the events in Germany, but after the last two weeks i have no doubts about it.

That said we had really luck with the dates: A day after the event in Munich snow got a major problem there and as you may know flying to or from FRA is a major pain in the asymmetric photons at the moment due to the apron.

Posted by Joerg Moellenkamp in General at 11:05

Sunday, February 12, 2012

Less known Solaris 11 features: Shadow Migration

In the ZFS Storage Appliance we have little nice feature enabling you to do migrations of data in the background. It's called Shadow Migration. It's a really useful feature. Imagine you have a RAIDZ. After a time you recognize that RAIDZ wasn't a good decision for your workload and RAID10 would be much better choice. But how to transform it into a RAID10 and how to do it with minimal interruption? You can do this with the Shadow Migration feature. With the Shadow Migration feature, you can migrate the data from one local or remote filesystem to another, while you are already accessing the new one to get the data on the old ZFS filesystem. This feature is available in Solaris 11 as well.

```
For this demonstration we will use two zfs pools consisting out of files. So we have to create the files
first:root@test:/test/brainslug# mkfile 128m source1
root@test:/test/brainslug# mkfile 128m source2
root@test:/test/brainslug# mkfile 128m source3
root@test:/test/brainslug# mkfile 128m source4
root@test:/test/brainslug# mkfile 128m target1
root@test:/test/brainslug# mkfile 128m target2
root@test:/test/brainslug# mkfile 128m target3
root@test:/test/brainslug# mkfile 128m target4
root@test:/test/brainslug# mkfile 128m target5
root@test:/test/brainslug# mkfile 128m target6Now the pools are created. At first our RAIDZ pool consisting out of 4
files. It's named sourceroot@test:/# zpool create source raidz \
/test/brainslug/source1 \
/test/brainslug/source2 \
/test/brainslug/source3 \
/test/brainslug/source4The second one is the future target of the shadow migration. It consists out of six
"disks"root@test:/# zpool create target
mirror /test/brainslug/target1 /test/brainslug/target2 \
mirror /test/brainslug/target3 /test/brainslug/target4 \
mirror /test/brainslug/target5 /test/brainslug/target6When you did a basic install, the tools and daemons needed for
shadow-migration are not included. You have to install them and enable the shadowd
afterwards:root@test:/test/brainslug# pkg install shadow-migration
root@test:/test/brainslug# svcadm enable shadowdNow you should see the shadowd daemon
running.root@test:/test/brainslug# ps -ef | grep "shadow"
root 3292 1 0 14:32:33 ? 0:03 /usr/lib/fs/shadowdOkay ... to test the shadow migration we create a
filesystem in the source pool:root@test:/test/brainslug# zfs create source/somestuffNow we have to fill this file with a
some data. Let's create some play data.root@test:/test/brainslug# dd if=/dev/urandom of=myfile bs=1024 count=300000
300000+0 records in
300000+0 records out
root@test:/test/brainslug# mkdir demodata
root@test:/test/brainslug# cd demodata
root@test:/test/brainslug/demodata# split -b 128k -a 5 ../myfileThis should yield a significant number of 128k files. Now
we copy them to the newly created filesystem source/somestuffWe will copy the files into the zfs filesystem posing as
our old filesystem:root@test:/test/brainslug/demodata# cp * /source/somestuff/
root@test:/test/brainslug/demodata# cd /
root@test:/# zfs list source
NAME USED AVAIL REFER MOUNTPOINT
source 294M 42,1M 46,4K /sourceJust to have something to compare, you could simply count the files and calculate
the md5 checksum of a file.root@test:/# ls -l /source/somestuff | wc -l
2345
root@test:/# md5sum /source/somestuff/xaadmd
3fb4a6be2f93c3d93998db52061244aa /source/somestuff/xaadmdShadow migration will only works, when the source
filesystem read-only. So we have to put the source filesystem into such a state:root@test:/# zfs set readonly=on
source/somestuffOkay, now let's configure the shadow migration:root@test:/# zfs create -o
shadow=file:///source/somestuff \
target/newlocationforsomestuffThat's all. The command may take some moments to get back. The migration of data
```

starts right in the moment you create the new filesystem. It runs in the background and starts to copy all data to the new filesystem. Important to know: You can do shadow migration via NFS as well and it can be an UFS filesystem as well. you just have to declare the source of the shadow migration like `nfs://fileserver/directory`

Okay. With shadowstat we can check the process of migration.`root@test:/# shadowstat`

```
EST
  BYTES  BYTES      ELAPSED
DATASET   XFRD  LEFT  ERRORS  TIME
target/newlocationforsomestuff 25,5M - - 00:01:10
The cool think about shadow migration is: You can already use the new filesystem. Despite the fact that the migration is still running, you will already see all files and when you access one file it will be migrated in the moment you access the file on the new filesystem. You don't have to wait with the access, until the block would be migrated by the normal background migration. When you try to access data, that isn't already migrated, it's migrated in the moment you access it in the new filesystem.root@test:/# md5sum /target/newlocationforsomestuff/xaadmd
3fb4a6be2f93c3d93998db52061244aa /target/newlocationforsomestuff/xaadmd
root@test:/# ls -l /target/newlocationforsomestuff | wc -l
```

2345Afterwards it proceeds with the further migration of all data in the pool. You can observe that with the shadowstat command.`root@test:/# shadowstat`

```
EST
  BYTES  BYTES      ELAPSED
DATASET   XFRD  LEFT  ERRORS  TIME
target/newlocationforsomestuff 97,8M - - 00:01:50
target/newlocationforsomestuff 128M - - 00:02:00
target/newlocationforsomestuff 147M - - 00:02:10
target/newlocationforsomestuff 165M - - 00:02:20
target/newlocationforsomestuff 186M - - 00:02:30
target/newlocationforsomestuff 202M - - 00:02:40
target/newlocationforsomestuff 211M - - 00:02:50
target/newlocationforsomestuff 224M - - 00:03:00
target/newlocationforsomestuff 236M - - 00:03:10
target/newlocationforsomestuff 243M - - 00:03:20
target/newlocationforsomestuff 249M - - 00:03:30
target/newlocationforsomestuff 256M - - 00:03:40
target/newlocationforsomestuff 260M - - 00:03:50
target/newlocationforsomestuff 266M - - 00:04:00
target/newlocationforsomestuff 272M - - 00:04:10
target/newlocationforsomestuff 278M - - 00:04:20
target/newlocationforsomestuff 286M - - 00:04:30
No migrations in progress
```

`root@test:/#Successfully migrated.`

Do you want to learn more?

Docs

[docs.oracle.com: Migrating ZFS File Systems"](https://docs.oracle.com/...)

[docs.oracle.com: Migrating File System Data to ZFS File Systems](https://docs.oracle.com/...)

Blogs

[blogs.oracle.com: What is Shadow Migration](https://blogs.oracle.com/...)

[blogs.oracle.com: Shadow Migration Internals](https://blogs.oracle.com/...)

Posted by Joerg Moellenkamp in English, Solaris at 08:08

Tuesday, February 7. 2012

Oracle Solaris 11 Techday Düsseldorf geschafft ...

... bleiben noch sechs Veranstaltungen (man kann sich noch anmelden). Der Event in Düsseldorf war schon mal ein guter Auftakt. Viele Leute da, viele Fragen gehabt in den Kaffeepausen. Morgen nun Stuttgart ...

Fand meinen Vortrag ein wenig holprig, zuviel Reuse, bin irgendwie nicht in den Flow gekommen. Habe meinen Vortrag aber für die nächsten Veranstaltungen noch ein wenig umgebaut. Und bevor jemand fragt: Ja ... die Slideanzahl ist noch zweistellig Hab doch nur 30 Minuten

PS: Mag mal jemand den Flughafenbetreibern in DUS sagen, das ein Hinweisschild ganz nett wäre, das es da einen Starbucks oben bei der Besucherterasse gibt? Ich bin stumpf dran vorbeigelaufen.

Posted by Joerg Moellenkamp in German, Solaris at 21:52

Thursday, February 2. 2012

Reminder: Oracle Solaris 11 Techdays 2012

Ich möchte nochmal auf die nächste Woche startende Veranstaltungsreihe zum Thema Solaris 11 hinweisen. Es gibt zwar schon viele Anmeldungen aber ich will "die Hütte voll sehen" . Mehr Informationen sowie eine Agenda dazu findet ihr hier.

Posted by Joerg Moellenkamp in German at 21:55