Удалённая миграция с UFS на ZFS

Автор: lissyara.
Оригинал:
http://www.lissyara.su/articles/freebsd/file_system/remote_migration_from_ufs_to_zfs/

Итак, решил что пора мигрировать на ZFS - много всего хорошего рассказывают, да и в рассылке народ активировался - уже с raidZ можно грузиться, плюс в ожидаемом релизе 8-ки оно будет "реди фор продакшен". Значит - пора ставить на свой любимый тестовый тазик =)
Ожидаемые проблемы - тазик капризный до памяти - да и мало на нём её - всего 128Mb. Для ZFS рекомендовано минимум 512. Поэтому, проблемы обязательно будут =). До кучи, хочеться всё сделать через ssh - т.е. без однопользовательского режима, всяких IP-KBM и т.п.
Итак, обновляемся до 8-ки (у меня там 7.2 стояла пустая), получаем следующее:2xPIII-500MHz$ uname -a
FreeBSD 2xPIII-500MHz 8.0-PRERELEASE FreeBSD 8.0-PRERELEASE #0:
Thu Nov 12 18:25:58 UTC 2009
root@2xPIII-500MHz:/usr/obj/usr/src/sys/GENERIC i386
2xPIII-500MHz$

Система стоит на небольшом сказёвом диске, на отдельном контроллере. Переносить буду с использованием дополнительного диска:2xPIII-500MHz$ dmesg | grep -E "amrd[0-9]|ad[0-9]"
ad0: 176700MB <IC35L180AVV207 1 V26OA63A> at ata0-master UDMA33
amrd0: <LSILogic MegaRAID logical drive> on amr0
amrd0: 8700MB (17817600 sectors) RAID 0 (optimal)
Trying to mount root from ufs:/dev/amrd0s1a
2xPIII-500MHz$

Разбит одним шматком, смонтирован асинхронно (да, я извращенец =) Но - машинка тестовая - чё хочу то и делаю):2xPIII-500MHz$ df -h
Filesystem Size Used Avail Capacity Mounted on
/dev/amrd0s1a 7.7G 3.5G 3.7G 48% /
devfs 1.0K 1.0K 0B 100% /dev
2xPIII-500MHz$
2xPIII-500MHz$ mount
/dev/amrd0s1a on / (ufs, asynchronous, local)
devfs on /dev (devfs, local, multilabel)
2xPIII-500MHz$

Загрузчик (loader), в 8-ке, по-умолчанию собран без поддержки загрузки с ZFS. Вернее, даже не так. Для ZFS используется отдельный загрузчик, и изначально его нет. Поэтому, вносим такую строку в make.conf:2xPIII-500MHz$ grep -i zfs /etc/make.conf
# for zfs boot
LOADER_ZFS_SUPPORT=yes
2xPIII-500MHz$

и пересобираем всё что касается загрузки системы:2xPIII-500MHz$ cd /usr/src/sys/boot && make obj depend all install

Прописываем загрузку модуля ZFS:2xPIII-500MHz$ grep zfs /boot/loader.conf
zfs_load="YES"
2xPIII-500MHz$

и монтирование файловых систем при загрузке:2xPIII-500MHz$ grep zfs /etc/rc.conf
zfs_enable="YES"
2xPIII-500MHz$

Создаём пул (о том что такое пул, и с чем его едят можно почитать в доке по утилите zpool, ну а вкратце - это набор девайсов, предоставляющих физическое хранилищще для ZFS):2xPIII-500MHz$ zpool create rootFS /dev/ad0

Смотрим, чё получилось:2xPIII-500MHz$ zpool list
NAME SIZE USED AVAIL CAP HEALTH ALTROOT
rootFS 172G 73,5K 172G 0% ONLINE -
2xPIII-500MHz$
2xPIII-500MHz$ zpool status
pool: rootFS
state: ONLINE
scrub: none requested
config:

NAME STATE READ WRITE CKSUM
rootFS ONLINE 0 0 0
ad0 ONLINE 0 0 0

errors: No known data errors
2xPIII-500MHz$

Экспортируем пул - чтобы ZFS, при дальнейших наших действиях, точно не трогала диск

на котором он живёт:2xPIII-500MHz$ zpool export rootFS
2xPIII-500MHz$ zpool list
no pools available

Записываем загрузчики - для первой и второй стадии загрузки:2xPIII-500MHz$ dd if=/boot/zfsboot of=/dev/ad0 bs=512 count=1
1+0 records in
1+0 records out
512 bytes transferred in 0.000513 secs (997901 bytes/sec)
2xPIII-500MHz$
2xPIII-500MHz$
2xPIII-500MHz$ dd if=/boot/zfsboot of=/dev/ad0 bs=512 skip=1 seek=1024
64+0 records in
64+0 records out
32768 bytes transferred in 0.020961 secs (1563299 bytes/sec)
2xPIII-500MHz$

Цепляем пул обратно:2xPIII-500MHz$ zpool import rootFS
2xPIII-500MHz$ zpool list
NAME SIZE USED AVAIL CAP HEALTH ALTROOT
rootFS 172G 73,5K 172G 0% ONLINE -
2xPIII-500MHz$

В установках пула, выставляем отсутствие точки монтирования (корень ("/"), выставить сразу не можем - потому как пустой пул тут же будет примонтирован в качестве корневой системы, и повествование свернёт в другую сторону - на рассказ по теме "чё же делать если всё пошло не так")):2xPIII-500MHz$ zfs set mountpoint=none rootFS
2xPIII-500MHz$ zfs get mountpoint rootFS
NAME PROPERTY VALUE SOURCE
rootFS mountpoint none local
2xPIII-500MHz$

Монтируем файловую систему куда вам удобно - мне - в /mnt:2xPIII-500MHz$ mount rootFS /mnt/
mount: rootFS : No such file or directory
2xPIII-500MHz$ mount -t zfs rootFS /mnt/

Смотрим:2xPIII-500MHz$ df -h
Filesystem Size Used Avail Capacity Mounted on
/dev/amrd0s1a 7.7G 3.5G 3.7G 48% /
devfs 1.0K 1.0K 0B 100% /dev

```
rootFS          169G      0B    169G     0%    /mnt
2xPIII-500MHz$ mount
/dev/amrd0s1a on / (ufs, asynchronous, local)
devfs on /dev (devfs, local, multilabel)
rootFS on /mnt (zfs, local)
2xPIII-500MHz$
```

Переносим систему:2xPIII-500MHz$ dump -0Lf - / | (cd /mnt/; restore -rf -)

Вот тут начались грабли. Через несколько минут словил панику - ругалось что мало рамы ядру. Добавил рамы до 320Mb. После ребута, и монтирования раздела с ZFS сделал:2xPIII-500MHz$ rm -rf /mnt/*

Снова паника с теми же симптомами. Грустно... Обычной SDRAM у меня больше не было. Пришлось дёрнуть с ближайшего старенького сервера 2 планки SDRAM ECC по 2Gb - на этой машине они увиделись как 2x256. Снова запустил dump/restore - снова паника. Нашёл в заначке ещё одну на гиг - тоже увиделась как 256 - всего получилось 700 с чем-то мегов - процесс прошёл нормально.
Прописываем в loader.conf откуда монтировать корневой раздел:2xPIII-500MHz$ grep zfs /boot/loader.conf

```
zfs_load="YES"
vfs.root.mountfrom="zfs:rootFS"
2xPIII-500MHz$
```

Убираем из fstab, что на разделе с ZFS все записи:2xPIII-500MHz$ more /mnt/etc/fstab

```
# Device                Mountpoint      FStype  Options         Dump    Pass#
#/dev/amrd0s1b          none            swap    sw              0       0
#/dev/amrd0s1a          /               ufs     rw,async        1       1
#/dev/acd0              /cdrom          cd9660  ro,noauto       0       0
2xPIII-500MHz$
```

Перезагружаемся, видим такую картинку:2xPIII-500MHz$ df -h

```
Filesystem    Size    Used   Avail Capacity  Mounted on
rootFS        169G    3.5G    166G     2%    /
devfs         1.0K    1.0K      0B   100%    /dev
2xPIII-500MHz$ mount
rootFS on / (zfs, local)
devfs on /dev (devfs, local, multilabel)
2xPIII-500MHz$
```

Итак, что имеем - загрузка ядра, всё ещё, произошла по старому - с первого SCSI диска, с UFS. А вот файловая система уже монтируется с другого - на котором ZFS. Дальше, стрёмный момент - убиваем всё содержимое первого диска - именно ради него, чуть раньше, я прописывал загрузчики на второй диск - если после убиения, но до конца переноса машина будет перезагружена - можно будет загрузиться со второго диска. Если же на него не прописать загрузчик - грузиться будет не с чего. Итак, убиваем всё на загрузочном диске:2xPIII-500MHz$ ll /dev/amrd0*
crw-r----- 1 root operator   0, 83 13 ноя 14:15 /dev/amrd0
crw-r----- 1 root operator   0, 83 13 ноя 14:15 amrd0s1
crw-r----- 1 root operator   0, 83 13 ноя 14:15 amrd0s1a
crw-r----- 1 root operator   0, 83 13 ноя 14:15 amrd0s1b
2xPIII-500MHz$
2xPIII-500MHz$ dd if=/dev/zero of=/dev/amrd0 bs=1m count=1
1+0 records in
1+0 records out
1048576 bytes transferred in 0.079149 secs (13248126 bytes/sec)
2xPIII-500MHz$

Проверяем, что все разделы пропали:2xPIII-500MHz$ ll /dev/amrd0*
crw-r----- 1 root operator   0, 83 13 ноя 14:15 /dev/amrd0
2xPIII-500MHz$

Прописываем загрузчики:2xPIII-500MHz$ dd if=/boot/zfsboot of=/dev/amrd0 bs=512 count=1
1+0 records in
1+0 records out
512 bytes transferred in 0.017034 secs (30058 bytes/sec)
2xPIII-500MHz$ dd if=/boot/zfsboot of=/dev/amrd0 bs=512 skip=1 seek=1024
64+0 records in
64+0 records out
32768 bytes transferred in 0.030083 secs (1089247 bytes/sec)
2xPIII-500MHz$

А теперь, финт ушами - говорим zpool, что надо поменять один диск на другой:2xPIII-500MHz$ zpool replace rootFS /dev/ad0 /dev/amrd0
cannot replace /dev/ad0 with /dev/amrd0: device is too small

Обломалося земноводное зелёного цвета... (© "Красная Плесень", какой-то из рассказов про Гену и Чебурашку). Вначале надо до конца читать ман, а потом делать. Девайс нельзя заменить девайсом меньшего размера (непонятно лишь почему - данных там меньше чем размер самого маленького диска. Видимо, для замены используется зеркалирование, и отключение второго диска от зеркала), тока такого же или большего размера. Тут пришлось начать думать и плотно раскуривать доку (а с первого диска я

уже всё снёс =))). Ключ "force" не помог:2xPIII-500MHz$ zpool replace -f rootFS /dev/ad0 /dev/amrd0
cannot replace /dev/ad0 with /dev/amrd0: device is too small
2xPIII-500MHz$

Ладно. Тогда попробуем реплицировать - в мане есть пример удалённого, должно и локально прокатить. Создаём новый пул, на SCSI диске:2xPIII-500MHz$ zpool create rootVG /dev/amrd0

(VG - виртуальная группа, такое именование в AIX принято. Куда удобней чем tank'и из доки по ZFS) Посмотрим, срослось ли:2xPIII-500MHz$ zpool list
NAME SIZE USED AVAIL CAP HEALTH ALTROOT
rootFS 172G 3,50G 169G 2% ONLINE -
rootVG 8,44G 72K 8,44G 0% ONLINE -
2xPIII-500MHz$

Делаем снимок файловой системы:2xPIII-500MHz$ zfs snapshot rootFS@now
2xPIII-500MHz$ zfs get all | grep @now
rootFS@now type snapshot -
rootFS@now creation пт ноя 13 14:46 2009 -
rootFS@now used 538K -
rootFS@now referenced 3,50G -
rootFS@now compressratio 1.00x -
rootFS@now devices on default
rootFS@now exec on default
rootFS@now setuid on default
rootFS@now shareiscsi off default
rootFS@now xattr on default
rootFS@now version 3 -
rootFS@now utf8only off -
rootFS@now normalization none -
rootFS@now casesensitivity sensitive -
rootFS@now nbmand off default
rootFS@now primarycache all default
rootFS@now secondarycache all default
2xPIII-500MHz$

Передаём снимок с одной файловой системы на другую:2xPIII-500MHz$ zfs send rootFS@now | zfs receive rootVG
cannot receive new filesystem stream: destination 'rootVG' exists
must specify -F to overwrite it
warning: cannot send 'rootFS@now': Канал разрушен

2xPIII-500MHz$ zfs send rootFS@now | zfs receive -F rootVG
2xPIII-500MHz$

Процесс, заметим, весьма ресурсоёмкий. Но, происходит быстрей чем dump/restore - в разы (но сильно медленней чем зеркалирование через zfs - там вообще всё очень шустро). Посмотрим, что получилось:2xPIII-500MHz$ zfs list
NAME USED AVAIL REFER MOUNTPOINT
rootFS 3,50G 166G 3,50G none
rootVG 3,50G 4,81G 3,49G /rootVG
2xPIII-500MHz$

Цепляю в другую точку монтирования - для своего удобства, вначале выставляю отстуствие её для этого пула (заметим, в /rootVG оно автоматом примонтировалось, видимо, во время предыдущей операции, также, замечу, что убрать эту точку монтирования надо обязательно - иначе на загрузке вместо "/" пул смонтируется в "/rootVG" - это не совсем то, что нам надо =)):2xPIII-500MHz$ zfs set mountpoint=none rootVG
2xPIII-500MHz$

Раздел при этом, автоматически отмонтируется - если не было открытых файлов:2xPIII-500MHz$ zfs list
NAME USED AVAIL REFER MOUNTPOINT
rootFS 3,50G 166G 3,50G none
rootVG 3,50G 4,81G 3,49G none
2xPIII-500MHz$

Монтирую:2xPIII-500MHz$ mount -t zfs rootVG /mnt/
2xPIII-500MHz$ df -h
Filesystem Size Used Avail Capacity Mounted on
rootFS 169G 3.5G 166G 2% /
devfs 1.0K 1.0K 0B 100% /dev
rootVG 8.3G 3.5G 4.8G 42% /mnt
2xPIII-500MHz$

Подправляем loader.conf - это надо сделать в любом случае, неважно, были и проблемы с меньшим диском, как у меня, или у вас первый диск был больше/равен второму - ибо предыдущий раз этот файл трогали после зеркалирования, на диске который уже убили:2xPIII-500MHz$ grep zfs /mnt/boot/loader.conf
zfs_load="YES"
vfs.root.mountfrom="zfs:rootVG"
2xPIII-500MHz$

Товарисчи с большими/равными дисками должны прописать "rootFS" а не "rootVG". Теперь можно перезагрузиться, и посмотреть чё вышло:2xPIII-500MHz$ df -h
Filesystem Size Used Avail Capacity Mounted on
rootVG 8.3G 3.5G 4.8G 42% /
devfs 1.0K 1.0K 0B 100% /dev
2xPIII-500MHz$ mount
rootVG on / (zfs, local)
devfs on /dev (devfs, local, multilabel)
2xPIII-500MHz$ zpool list
NAME SIZE USED AVAIL CAP HEALTH ALTROOT
rootFS 172G 3,50G 169G 2% ONLINE -
rootVG 8,44G 3,50G 4,94G 41% ONLINE -
2xPIII-500MHz$

Экспотрируем "rootFS" - чтоб не мешалось:2xPIII-500MHz$ zpool export rootFS
2xPIII-500MHz$ zpool list
NAME SIZE USED AVAIL CAP HEALTH ALTROOT
rootVG 8,44G 3,50G 4,94G 41% ONLINE -

Создаём раздел в полгига размером - под свап. Тут тоже моё упущение - свап рекомендуют размещщать в начале диска, у меня он получился в середине. Надо было эти действия сделать сразу после создания пула, до переноса данных:2xPIII-500MHz$ zfs create -V 512Mb rootVG/swap
2xPIII-500MHz$ zfs list
NAME USED AVAIL REFER MOUNTPOINT
rootVG 4,00G 4,31G 3,49G none
rootVG/swap 512M 4,81G 16K -

Выставляем переменные для раздела - тип файловой системы, и отключаем подсчёт контрольных сумм:2xPIII-500MHz$ zfs set org.freebsd:swap=on rootVG/swap
2xPIII-500MHz$ zfs set checksum=off rootVG/swap
2xPIII-500MHz$

Дальше, я попытался этот свап подцепить:2xPIII-500MHz$ zfs mount -a
2xPIII-500MHz$ swapinfo
Device 1K-blocks Used Avail Capacity
2xPIII-500MHz$

Неподцепился. Логично - это не файловая система же. Тогда, пойдём обычным путём:2xPIII-500MHz$ swapon /dev/zvol/rootVG/swap
2xPIII-500MHz$ swapinfo

Device 1K-blocks Used Avail Capacity
/dev/zvol/rootVG/swap 524288 0 524288 0%
2xPIII-500MHz$

Дальше по желанию. Диск я буду отцеплять, поэтому на втором убиваю файловую систему:2xPIII-500MHz$ dd if=/dev/zero of=/dev/ad0 bs=1m count=1
1+0 records in
1+0 records out
1048576 bytes transferred in 0.091236 secs (11493023 bytes/sec)
2xPIII-500MHz$ zpool import rootFS
cannot import 'rootFS': no such pool available
2xPIII-500MHz$

Теперь потестим то, что получилось. Накатил MySQL, apache2, php5, раскатал архив своего форума. Словил жёсткий зависон на попытке получить страницу через fetch. Мда. Ладно. За прошедшие выходные нашёл ещё рамы - 512Mb ECC, которая увиделась как 128. До гига так и не дотянул - получилось 917Mb. Краш-тест, в виде параллельного скачивания форума в 200 потоков система пережила. LA был 160, свап заюзан на 10%.
Нарисовал в loader.conf такие переменные (получено приблизительной экстраполяцией, и дальнейшей подгонкой значений из стабильно работающей системы с 768Mb рамы), и уменьшил количество ОЗУ до 256Mb:vm.kmem_size="200M"
vm.kmem_size_max="180M"
vfs.zfs.arc_max="16M"
vfs.zfs.vdev.cache.size="5M"

не упало - тестил переносом всего с однойго пула на другой, через:cd / && pax -p eme -X -rw . /mnt

Правда, перед тем как вышел на эти значения оно у меня три раза падало. Значит направление мысли было верное. Оттестил ещё раз, запустив в параллель к pax скачиваться форум в 200 потоков, и индексировать базу самого форума. Полёт нормальный.

Выводы. Работоспособно. Даже на небольшом количестве рамы, и архитектуре x32 можно допилить чтоб работало стабильно. Под amd64 всё и само будет стабильно работать.
Тем не менее - рамы рекомендуется 1Gb и более и архитектуру amd64 - там работа с памятью грамотней организована.