

FreeBSD 7.0: быстрее, современней, надёжней

*FreeBSD и раньше не была плохой.
Просто она стала ещё лучше, чем была.*

*Андре Опперман,
один из разработчиков FreeBSD*



FreeBSD

The power to serve.

Сергей Супрунов

27 февраля состоялся долгожданный анонс новой версии одной из популярнейших открытых операционных систем – FreeBSD 7.0-RELEASE; на ftp-зеркалах образы системы были выложены чуть раньше. Посмотрим, что нового в ней появилось.

Ожидание первого релиза в новой ветке любого программного продукта, а операционной системы в особенности, всегда порождает двойственные чувства: с одной стороны, от него ждёшь кардиналь-

ных улучшений и решения всех проблем и неудобств, присущих нынешним версиям; с другой – присутствуют определённые опасения по поводу стабильности, производительности, безопасности...

Итак, спустя два с половиной года, потраченных на разработку, в свет вышел FreeBSD 7.0-RELEASE, который уже сейчас считается достаточно стабильным, несмотря на довольно большое количество нововведений.

Традиции инсталляции

Итак, скачиваем iso-образ, записываем на диск, загружаемся. Нас встречает до боли знакомый sysinstall. То есть слухи о том, что основным инсталлятором системы начиная с 7.0-RELEASE станет BSDInstaller, не подтвердились. Так что если вы уже ставили FreeBSD, то изучать инсталляцию заново не придётся – пробегаем все шаги с закрытыми глазами, и можно приступать к знакомству с системой.

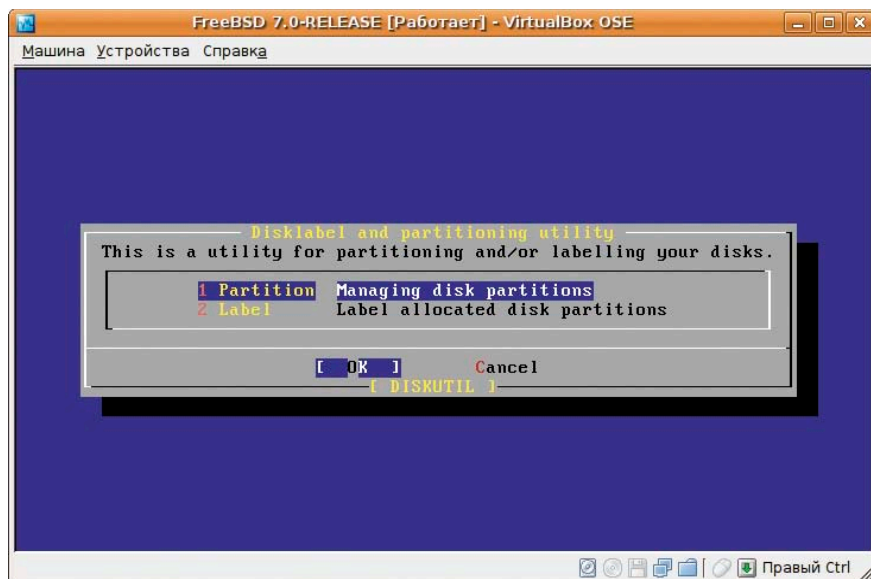
Перестройка и ускорение

Существенные изменения, нацеленные на повышение производительности многопроцессорных систем, коснулись кода ядра и многих драйверов.

Пожалуй, самая яркая и активно обсуждаемая новость – планировщик ULE 3.0 (или SHED_ULE, ранее носивший имя SHED_SMP). Как экспериментальный код его прежняя версия появилась ещё в 5-й ветви системы и широко использовалась на многопроцессорных машинах.

Сейчас разработчики обещают высокую стабильность кода и заметное повышение производительности даже на однопроцессорных системах (особенно хорошо чувствуется более высокая «отзывчивость» сильно загруженной системы при работе в консоли). Хотя в качестве планировщика по умолчанию в 7.0 пока ещё сохранился проверенный временем SHED_4BSD, но SHED_ULE уже в следующем релизе – 7.1 – должен занять его место. Так что желающим попробовать его прямо сейчас всё же придётся пересобирать ядро, заменив «options SHED_4BSD» на «options SHED_ULE».

Ещё один существенный шаг к повышению эффективности системы – отказ от глобальных блокировок (Giant locks) в коде стека TCP/IP и некоторых других подсистемах. В итоге значительно повышается производительность сетевых операций в многопоточной среде. Вместе с другими улучшениями (динамически изменяемый размер буфера TCP-сокетов, дополнительный указатель в структуре mbuf, общая оптимизация кода) это, по словам разработчиков, позволило повысить производительность стека в 3-5 раз, а на некоторых операциях и выше [2].



Утилита sade – урезанный sysinstall для работы с дисками

Существенной оптимизации подверглась и библиотека libthr, отвечающая за работу с потоками. Теперь она используется по умолчанию. Одно из следствий этого – код объектов планирования ядра (Kernel Scheduled Entities, KSE) теперь не является необходимым и потому исключён из ядра по умолчанию. При необходимости он может быть добавлен пересборкой ядра с опцией «options KSE».

Сеть нового поколения

Помимо улучшения производительности сетевой подсистемы, можно отметить и ряд шагов, направленных на более качественную поддержку современных протоколов. Поддержка IPv6, и в прежних версиях осуществляемая на должном уровне, теперь стала ещё шире: этот протокол поддерживается новой реализацией IPSec, появилась возможность пропускать IPv6-трафик через GRE-туннели, v6 теперь поддерживается драйвером ядра rrr, и т. д. Наконец-то разработчики окончательно избавились от кода ip6fw – теперь ipfw полностью поддерживает и шестую версию протокола IP.

Раз уж зашла речь об ipfw, то нужно отметить одно важное нововведение – теперь этот пакетный фильтр, как и его собратья PF и IPFilter, умеет самостоятельно выполнять NAT-трансляцию, не требуя запуска демона natd:

```
# ipfw add 3000 nat 5 ip from 1
any to any
# ipfw nat 5 config if nfe0 1
same_ports
```

Чтобы это работало, нужно добавить в /etc/make.conf строку:

```
CFLAGS+= -DIPFWALL_NAT
```

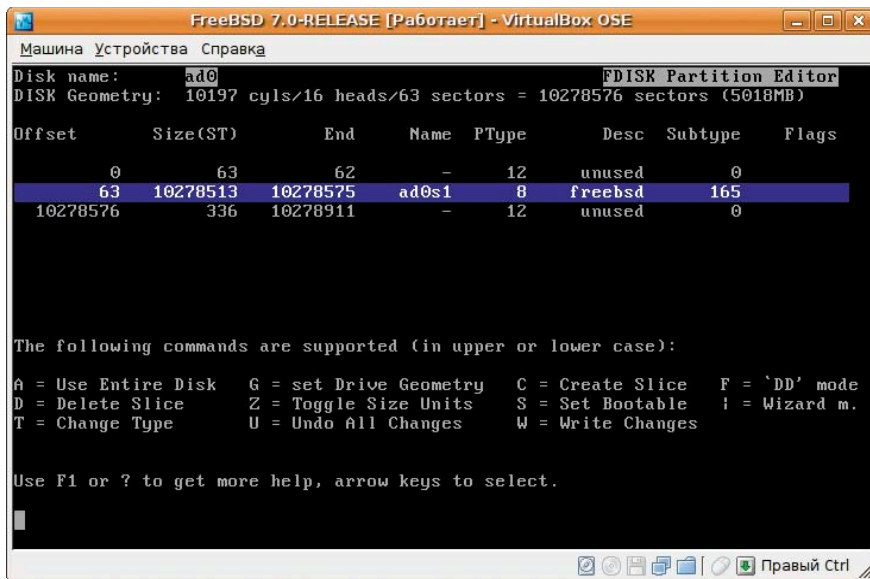
и пересобирать ipfw.ko, чтобы модуль ipfw собрался с привязками к libalias:

```
# cd /usr/src/sys/modules/ipfw
# make && make install
```

Приведёнными выше правилами ipfw мы добавляем в цепочку правило трансляции за номером 3000, сама трансляция выполняется согласно заданной в следующей команде конфигурации – интерфейсом определяется pfe0 (изменения его IP-адреса будут отслеживаться динамически) и даётся указание пытаться сохранять исходный номер порта. Помимо собственно трансляции адресов, можно выполнять и редирект (синтаксис опций близок к конфигурации natd). Подробности ищите на странице справки man ipfw(8).

Появилась поддержка протокола SCTP (транспортный протокол общего назначения, который рассматривается как будущая замена TCP, более соответствующая потребностям современных сетей), включённая по умолчанию в ядро GENERIC.

IPSec теперь разработана на базе кода FAST_IPSEC вместо реализации KAME. В результате опция ядра FAST_IPSEC утратила свою актуальность и была исключена. Добавлена поддержка шифра Camellia (см. RFC4132). Строка «device crypto» по-прежнему обязательно должна при-



Работа с разделами в sade – всё привычно и знакомо

существовать в конфигурации, чтобы сборка ядра с поддержкой IPSec прошла успешно.

Из систем NetBSD и OpenBSD портирован драйвер `lagg`, позволяющий объединять несколько сетевых интерфейсов в один, обеспечивающий повышение пропускной способности, надёжность и балансировку нагрузки. Во «FreeBSD Handbook» этому вопросу посвящён отдельный параграф [3]. По сравнению с `ng_one2many`, `lagg` предоставляет более широкие возможности.

Железные аргументы

Как и подобает любой новой системе, в 7.0-RELEASE улучшена поддержка оборудования. В частности, ряд изменений коснулся реализации ACPI. Система теперь будет поддерживать динамическое изменение рабочей частоты процессора, что называется, «из коробки», без дополнительной перекомпиляции ядра.

Обновлён драйвер `coretemp`, отвечающий за получение информации о температуре ядра процессора – теперь он поддерживает (по крайней мере, должен поддерживать) процессоры Intel Core. Но будьте с ним осторожны – на двух моих «древних» машинах (процессоры Intel Celeron 900 МГц и 1000 МГц) попытка загрузить модуль `coretemp` приводила к панике ядра «general protection fault». Сборка ядра с параметром «device `coretemp`» на таких процессорах приведёт соответственно к невозможности загрузки

системы. Поэтому сначала обязательно проверьте работу драйвера в режиме модуля.

Традиционно расширена поддержка и прочего оборудования – сетевых карт, USB-адаптеров, улучшена работа двухъядерных процессоров. Любителей качественного звука новый релиз тоже порадует – расширена поддержка различных аудиочипов, многие из них переработаны для лучшей работы в потоках. Некоторые устаревшие драйверы исключены из системы, либо поддержка обслуживаемых ими устройств передана в ведение других драйверов. Подробности см. в RELEASE NOTES [4].

Файлы – наше всё

Работе с файлами нельзя не уделять повышенного внимания – в современных системах именно дисковая подсистема зачастую становится «узким местом», существенно влияющим на производительность. К тому же, в отличие от оперативной памяти, нарастить которую – вопрос нескольких минут, истощение ёмкости жёсткого диска может вылиться в проблему (пусть и не слишком серьёзную, но всё же неприятную). Поэтому качество работы файловых систем всегда находилось под пристальным вниманием разработчиков.

ZFS

Самое заметное изменение в области хранения данных – поддержка ZFS, портированной из Solaris. Преимущес-

тва этой файловой системы уже описывались на страницах журнала [5]. Во FreeBSD почти все они будут вам доступны (хотя некоторые опции, похоже, ещё реализованы не до конца – в частности, мне так и не удалось создать ФС с произвольным размером блока, используя ключ `-b`).

Однако сразу нужно отметить, что ZFS весьма требовательна к ресурсам – разработчики рекомендуют использовать её только в том случае, если объём оперативной памяти вашей машины не менее 512 Мб (о чём вас предупредят). Последствия игнорирования этого предупреждения я ощутил на одной из тестовых машин (с объёмом ОЗУ 128 Мб) – попытка скопировать на ZFS-раздел даже не слишком большой объём данных (каталог `/usr`, созданный во время минимальной инсталляции, общим размером 126 Мб) привела к панике ядра с сообщением «`kmem_map too small`», даже несмотря на предусмотрительно заготовленный `swap`-раздел в 1,5 Гб.

Unionfs

Помимо появления долгожданной ZFS, есть и другие, не столь яркие, но весьма полезные новшества. Так, код `unionfs` был существенно переработан, в результате чего в нём устранены многие проблемы функциональности и стабильности (хотя на странице предупреждение о нестабильности и экспериментальности пока сохранилось). Кроме старого режима перекрытия каталогов (который теперь называется `traditional`), вводятся ещё два – `transparent` (когда файл или каталог, создаваемый на верхнем уровне, наследует права файла или каталога таким же именем нижнего уровня) и `masquerade` (когда права вновь создаваемых объектов определяются правами точки монтирования либо явно задаются в опциях команды `mount`).

Небольшое тестирование подтвердило, что теперь `unionfs` готова к «промышленному» использованию, что может быть полезно для работы с `read-only`-носителями в «режиме чтения-записи», для различных экспериментов с исходным кодом (первоначальный каталог, «прикрытый» с помощью `unionfs` рабочим каталогом верхнего уровня, будет надёжно защищён

от любых изменений), для построения jail-сред, и т. д.

Другие

В дополнение к mdmfs, позволяющей создавать файловые системы в оперативной памяти, в 7.0 появилась реализация tmpfs, портированная из NetBSD. В отличие от mdmfs, которая резервирует определённое количество памяти, указанное в параметрах команды mount, tmpfs динамически использует свободную память. Создать и смонтировать tmpfs-раздел можно простой командой:

```
# mount -t tmpfs tmpfs /tmp
```

Добавьте соответствующую строку в /etc/fstab, чтобы раздел монтировался автоматически. Только имейте в виду, что код пока является экспериментальным.

Кстати, обратите внимание на то, что утилиты mount_<тип файловой системы>, например, mount_msdosfs, теперь объявлены устаревшими (хотя некоторые из них в /sbin и /usr/sbin пока ещё сохранились) и для монтирования «чужих» файловых систем следует явно использовать опцию -t утилиты mount:

```
# mount -t msdosfs /dev/ad0s2 /mnt
```

Помимо старой, но не слишком доброй FAT, FreeBSD поддерживает, правда, с ограниченной возможностью записи или вообще только для чтения, файловые системы NTFS, EXT2, REISERFS и ряд других. В новой версии этот список пополнила XFS. Документация, похоже, несколько отстаёт от действительности, поскольку упоминания XFS на страницах man mount(8) пока ещё нет.

Кроме того, ряд существующих (псевдо)файловых систем (в частности, NFS, procfs и др.) переработаны для эффективной и безопасной работы в многопоточной системе.

Здесь же пару слов нужно сказать и о появлении нового класса GEOM — GEOM_JOURNAL. Он добавляет полноценное журналирование в различные файловые системы (пока поддерживается только UFS), что, в свою очередь, повышает надёжность и сокращает время восстановления при сбоях.

Обновление до 7.0

Процедура обновления с предыдущих версий системы до 7.0-RELEASE с использованием CVSup, в принципе, стандартна и подробно расписана в документации (прежде всего следует руководствоваться файлом /usr/src/UPDATING, содержащим наиболее актуальные инструкции).

Но если у вас нет ни желания, ни необходимости возиться с исходными кодами, то начиная с FreeBSD 6.3-RELEASE вы можете воспользоваться системой двоичных обновлений:

```
# freebsd-update -r 7.0-RELEASE .
# freebsd-update upgrade
# freebsd-update install
# ... перезагрузка ...
# freebsd-update install
# ... перекompляция стороннего ПО...
# freebsd-update install
```

Всего несколько команд, и ваша система будет обновлена до указанной версии. Правда, не стоит слишком полагаться на интеллектуальность этой утилиты — она обновляет только бинарные системные файлы, кое-что из /etc да ядро (и то лишь в том случае, когда используется одно из стандартных — GENERIC или SMP; в остальных случаях вам будет предложено пересобрать ядро вручную). Первая из указанных команд выполнит основную часть

По умолчанию опция UFS_GJOURNAL включена в ядро GENERIC, так что при желании вы сразу можете воспользоваться новыми возможностями:

```
# gjournal label /dev/ad1
# gjournal load
# ls /dev/ad1*
/dev/ad1 /dev/ad1.journal
# newfs -O 2 -J /dev/ad1.journal
# mount /dev/ad1.journal /mnt
```

Этими командами мы создали журналируемую файловую систему на всём диске ad1 (без разбивки на слайсы). Обратите внимание, что после выполнения команды «gjournal load» в каталоге /dev должен появиться файл соответствующего псевдоустройства с расширением journal, и в дальнейших командах следует использовать именно его.

Правда, модуль ядра geom_journal.ko, необходимый для работы журнальных ФС, автоматически не подгружается, поэтому, чтобы обеспечить монтирование ФС из /etc/fstab при загрузке системы, либо пересоберите ядро с опцией «options GEOM_JOURNAL»,

работы — скачает всё необходимое, предложит в режиме mergemaster сопоставить два основных файла — /etc/master.passwd и /etc/group, выведет списки файлов, которые будут удалены, добавлены или изменены. С последними настоятельно рекомендуется ознакомиться очень внимательно (особенно это касается конфигурационных файлов), чтобы потом не было сюрпризов. Ну и учтите, что на это удовольствие вам понадобится порядка 100 Мб трафика.

Инсталляцию обновлений придётся выполнять трижды — при первом подходе выполнится некоторая подготовительная работа и установится новое ядро. Если вы используете нестандартное ядро, вам придётся пересобрать его самостоятельно. Беря за основу свой старый конфигурационный файл, не забудьте добавить строку «options COMPAT_FREEBSD6», изменить «options GEOM_GPT» на «GEOM_PART_GPT» и удалить «device lnc» и другие исключённые из ядра драйверы. Хотя лучше всё-таки построить новое ядро на основе GENERIC. После перезагрузки повторный запуск «freebsd-update install» установит основные компоненты системы. Далее потребуется пересобрать стороннее ПО, установленное из Портов, после чего завершить инсталляцию.

либо добавьте строку «geom_journal_load=YES» в /boot/loader.conf.

Курс на POSIX

Также следует отметить некоторые улучшения поддержки стандартов POSIX. В частности, появилась экспериментальная поддержка MQUEUE (очередь сообщений, message queue), приведено в соответствие с POSIX поведение системных вызовов библиотеки setenv и некоторых других.

Искусство дружить

Эмулятор Linux продвинулся до версии ядра 2.6.16. Правда, данный код всё ещё имеет статус экспериментального, поэтому по умолчанию используется эмуляция ядра 2.4.2. Если всё же хочется воспользоваться преимуществами ядра 2.6, дайте команду:

```
# sysctl compat.linux.osrelease=2.6.16
```

```
compat.linux.osrelease: 2.4.2 -> 2.6.16
```

Очевидно, что данная sysctl-переменная станет доступной только пос-

ле загрузки соответствующего модуля (вручную это делается командой `kldload linux`), либо если ядро собиралось с опцией `COMPAT_LINUX`.

Здесь же упомяну, что в 7-й ветви системы прекращена поддержка платформы Alpha, зато теперь будет поддерживаться UltraSPARC-T1 от Sun Microsystems.

Безопасность превыше всего

Значительное улучшение подсистемы аудита, впервые появившейся в версии 6.2, позволило поднять её статус с «экспериментальной» до «используемой по умолчанию». Добавив в `/etc/rc.conf` строку `«auditd_enable=YES»`, вы получите возможность отслеживать большое число событий, так или иначе связанных с безопасностью системы: подключения к системе с помощью `ssh`, использования утилиты `su`, включение-выключение системы, и т. д. Логи аудита сохраняются по умолчанию в каталог `/var/audit`, просмотреть их содержимое в текстовом виде можно с помощью утилиты `praudit`.

Ядерный паритет

Все эти изменения не могли не затронуть и конфигурацию ядра, хотя нужно заметить, что по традиции они не столь значительны. Так, исчезла из `GENERIC` опция `COMPAT_43` (хотя она всё ещё поддерживается, если в ней есть необходимость), зато появились опции `SCTP`, `UFS_GJOURNAL`, `AUDIT`, устройство `crufreq`. Также обратите внимание, что опция `SMP` теперь включена в ядре `GENERIC` по умолчанию.

Среди устройств тоже есть некоторые изменения – одни добавлены, другие удалены, третьи переименованы. Например, вместо устройства «bridge» теперь следует использовать «if_bridge». Впрочем, таких изменений не слишком много.

Помимо изменений в конфигурации, добавился ряд новых `sysctl`-переменных. Некоторые были упомянуты выше. Ещё одна – `kern.confctx`, которая позволяет просмотреть, с какой конфигурацией собрано текущее ядро (следует использовать команду «`sysctl -b kern.confctx`»). Правда, доступной она будет только в случае, если ядро собрано с опцией `INCLUDE_CONFIG_FILE` (в файле `GENERIC` она отсутству-

ет). Также можно воспользоваться командой «`config -x /boot/kernel/kernel`», извлекающей конфигурацию из указанного двоичного файла ядра. Раньше опция `INCLUDE_CONFIG_FILE` тоже поддерживалась, но конфигурацию приходилось «выдирать» из бинарного файла с помощью утилиты `strings`.

Прочие (не)удобства

Одна из радостных новостей – в базовую систему теперь входит утилита `sade` (см. **рисунки**), позволяющая работать с дисковыми разделами «в стиле `sysinstall`» – теперь не обязательно дробить разделы с помощью `fdisk` с калькулятором в руках или запускать `sysinstall` только лишь для того, чтобы подготовить к работе новый жёсткий диск. Судя по всему, `sade` – это просто «вырезка» из `sysinstall`, предоставляющая функции управления разделами (`partitions`) и метками (`labels`), так что работать с ней достаточно привычно.

РАМ-модуль `ram_nologin` из категории `auth` переведён в `account`, поэтому если будете переносить в новую систему свои старые конфигурационные файлы, обратите на это внимание и внесите в них необходимые изменения.

Обновилась версия `Sendmail` до 8.14.2. Тем, кто пользуется этим MTA, нужно учесть небольшое изменение поведения – раньше при запуске системы автоматически проверялась дата обновления файла `/etc/mail/aliases` и при необходимости запускалась утилита `newaliases`, чтобы пересоздать базу. Теперь этого не будет, так что администратор получает больший контроль над системой (а заодно и большую ответственность). Чтобы вернуться к прежнему поведению, добавьте в `/etc/rc.conf` опцию «`sendmail_rebuild_aliases=YES`».

Также «посвежили» `PF`, `IPFilter`, `BIND`, `awk`, `ncurses` и ряд других библиотек и утилит. Компилятор `GCC` теперь используется версии 4.2.1 (вместо 3.4.6 в 6-й ветви), так что определённый запас на будущее обеспечен.

В `/etc/rc.d` появился сценарий `ftpd` для запуска стандартного `ftpd` в автономном режиме (раньше конфигурационными файлами предусматривалась работа только через `inetd`; запуск демона приходилось обеспечивать самостоятельно).

Добавлен конфигурационный файл `/etc/src.conf` (по умолчанию отсутствует), который отвечает за опции сборки дерева системы из исходных кодов. В связи с этим из `make.conf` была удалена часть опций, дублирующих функции `src.conf`. Например, вместо опции «`NO_GAMES`» в `make.conf`, теперь следует использовать «`WITHOUT_GAMES`» в `src.conf`. Подробности см. в документации, текущие опции сборки можно получить с помощью команды «`make showconfig`» в каталоге `/usr/src`.

Это, естественно, не все изменения, об остальных читайте в `RELEASE NOTES`.

Ну и огорчу любителей виртуализации – поддержка `XEN` пока не появилась.

Итоги

Большинство надежд, связанных с выходом версии `FreeBSD 7.0`, оправдалось. В целом система сохранила преемственность, так что многое из того, что работало раньше, должно сохранить работоспособность и теперь. Отказ от глобальных блокировок и общая оптимизация кода делают новую версию системы ещё более привлекательной для работы на сильно нагруженных серверах, а новый планировщик и улучшенная поддержка оборудования должны порадовать пользователей настольных систем. Ну а `ZFS` вообще выше всяких похвал – осталось только найти 512 Мб ОЗУ для «боевого» сервера.

В общем, `FreeBSD` «стала лучше во всех отношениях», за что всей команде разработчиков хочется сказать огромное спасибо! 🍀

1. `FreeBSD 7.0-RELEASE` Announcement – <http://www.freebsd.org/releases/7.0/announce.html>.
2. F. Biancuzzi. What's New in `FreeBSD 7.0` – <http://www.onlamp.com/lpt/a/7230>.
3. A. Thompson. 29.6 Link Aggregation and Failover. `FreeBSD Handbook` – http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/network-aggregation.html.
4. `FreeBSD 7.0-RELEASE` Release Notes – <http://www.freebsd.org/releases/7.0/relnotes.html>.
5. Коробкин А. Используйте преимущества файловой системы `ZFS` в `Solaris`. // Системный администратор, № 3, 2007 г. – С. 28-34.