Solaris Volume Manager. Конфигурирование RAID устройств

Юрий Кажаров

В статье описываются функциональные возможности Solaris Volume Manager (SVM), и рассматриваются механизмы конфигурирования RAID устройств.

Введение

Основная нагрузка в любой серверной системе ложиться на ее дисковую подсистему, что приводит к быстрому износу носителей информации и, как следствие, потере данных и вынужденному простою сервисов. Построение современных высокодоступных информационных систем накладывает жесткое требование по минимизации времени простоя сервисов и предоставление их в режиме 24x7x365.

Для обеспечения надежности функционирования дисковых систем и сохранности данных, используется технология RAID (Redundant Array of Inexpensive(Independent) Disks). RAID устройства представляют собой набор дисков, объединенных по особым правилам в один виртуальный дисковый носитель. Правила объединения дисков определяют тип создаваемого RAID устройства, например: 0, 1, 0+1, 1+0, 5 и т.д.

RAID устройства бывают:

- Аппаратные – когда для их организации используются специальные RAID-контроллеры
- Программные – когда RAID устройства создаются с помощью программных продуктов.

В операционной системе Sun Solaris используется довольно мощный программный продукт для организации RAID устройств - Solaris Volume Manager (SVM). Начиная с Solaris 9 (в Solaris 8 это был отдельный продукт – Solstice Disk Suite), он интегрирован в систему и предоставляет механизмы по созданию не только RAID устройств, но и расширяет функциональные возможности дисковой системы в Sun Solaris.

SVM предоставляет следующие функциональные возможности:

Возможность	Описание	Применение
- организация RAID-0	Совокупность физических частей (slices), которые объединяются в единое логическое устройство	Кода необходимо повысить производительность дисковой системы, либо увеличивать ее объем
- организация RAID-1	Зеркало диска либо его части	Повышение надежности дисковой подсистемы за счет параллельного копирования данных на зеркальный носитель
- организация RAID-5	Распределение данных по нескольким дискам и использование механизма контроля четности	Повышение надежности дисковой подсистемы за счет распределения данных на нескольких носителях с использованием механизма контроля четности

- создание soft-partitions	Разбиение физической части диска на несколько логических частей	Расширяет предел разбиения диска с 7 слайсов до 8192 так называемых soft-partitions и позволяет динамически перераспределять soft-partitions между разделами, увеличивая, таким образом, размер файловой системы
- организация пулов горячей замены (Hot spare pools)	Набор резервных дисков либо их частей, которые автоматически используются при нарушениях работы основных дисков RAID системы	Повышает надежность RAID-1 и RAID-5 систем
- организация распределенных дисковых систем (Disk sets)	Совокупность сетевых дисков, выставленных на разных системах и объединенных в логический диск, которые могут использоваться другими системами	Организация распределенных дисковых систем.

SVM в своей работе оперирует виртуальными логическими дисками, которые называются томами (volume) либо, исторически сохранившимся названием, - метаустройствами (metadevice). Эти логические устройства прописаны в системе в */dev/md/dsk* («блочный доступ») и */dev/md/rdsk* («прямой доступ») под названием – d<*номер_устройства*>, и могут создаваться как из физических дисков или их частей, так и из самих логических дисков, созданных SVM.

Создавать, конфигурировать и управлять виртуальными логическими дисками можно с помощью ряда команд либо с помощью графической консоли управления - Solaris Management Console (*/usr/sbin/smc*).

RAID-устройства. Теория

RAID-0. Этот вид устройств предназначен для расширения объемов файловой системы за счет «склеивания» различных частей дисковых носителей и не обеспечивает избыточность. Они могут использоваться как самостоятельные устройства, так и в организации более «сложных» – RAID-1 (зеркало).

RAID-0 устройства можно создавать на всю партицию диска и на отдельные части файловой системы: /(root), /usr, /var, /opt, swap и т.д.

Существует три функциональных механизма работы RAID-0:

- Распределение (stripe)
- Склеивание (concatenation)
- Склеивание «распределенных» томов (concatenated stripe)

При механизме распределения данные рассеиваются по всем частям тома одновременно (параллельно), что позволяет получить повышение производительности дисковой системы.

При создании распределенных систем необходимо помнить, что их невозможно создавать на уже существующей файловой системе. Нужно вначале сделать backup файловой системы, затем сконструировать логическое устройство, и восстановить сохраненные данные.

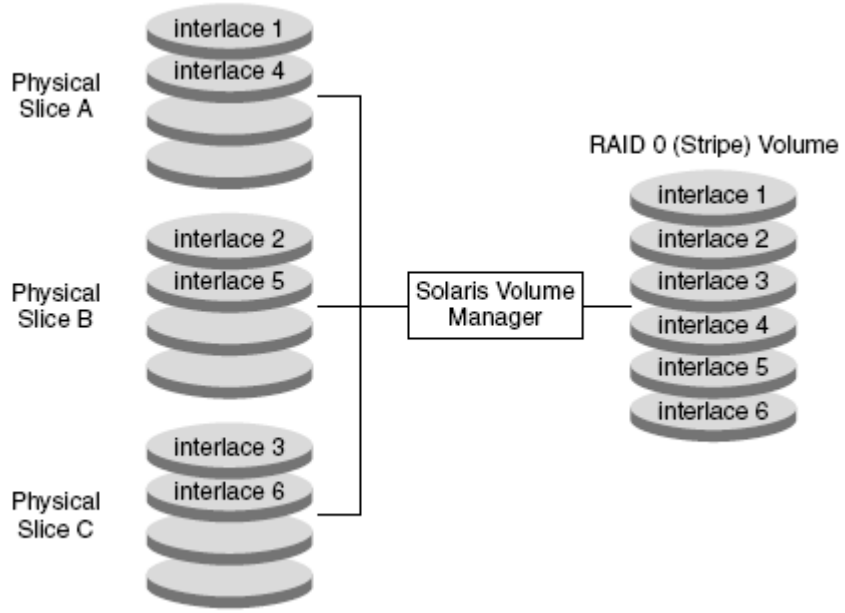

Рис.1 RAID-0. Пример распределенного тома (stripe)

При механизме «склеивания», данные вначале записываются в первую часть логического тома, по его заполнении – во вторую, и т.д. Очевидно, что в данном случае никакого повышения производительности операций ввода/вывода не происходит, так как данные записываются последовательно.

RAID-0 в режиме «склеивания» оптимально подходит для ситуаций, когда необходимо организовать дисковый раздел определенного размера, собрав его из нескольких частей.

Важной особенностью данного механизма, является возможность динамически расширять размер логического тома, добавляя новые части дисков. Кроме того, эти действия можно проводить на «активной» файловой системе без нарушения ее работоспособности.

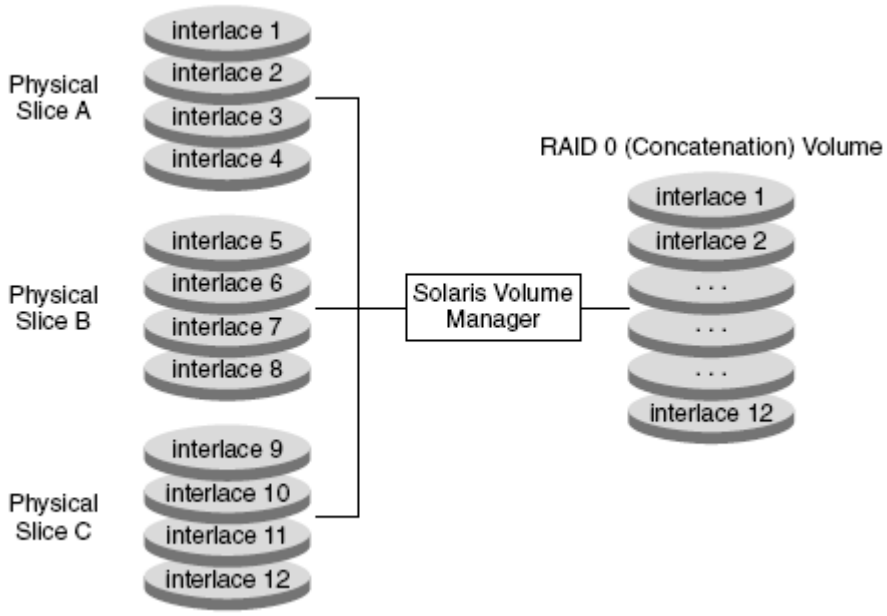

Рис.2 RAID-0. Пример «склеенного» тома (concatenation)

Режим склеивания «распределенных» томов (concatenated stripe) представляет собой совокупность описанных выше механизмов: сначала создаются «распределенные» тома, а потом они склеиваются в единое целое.

RAID-1. Его еще часто называют – зеркало. Зеркало представляет собой устройство, состоящее из RAID-0 томов, в котором данные дублируются параллельно на нескольких томах. Тома подразделяются на «главное зеркало» и «вторичные зеркала». Первые представляет собой основной дисковый раздел, который мы хотим продублировать, а вторые – копии. Всего может быть три таких копии.

Стандартное «зеркало» состоит из двух частей: «главного зеркала» и «вторичного зеркала». Вы можете добавить еще одно либо два дополнительных «вторичных зеркала» и использовать их для backup-а системы. Таким образом, можно «на горячую» производить архивирование файловой системы, организованной механизмом RAID-1.

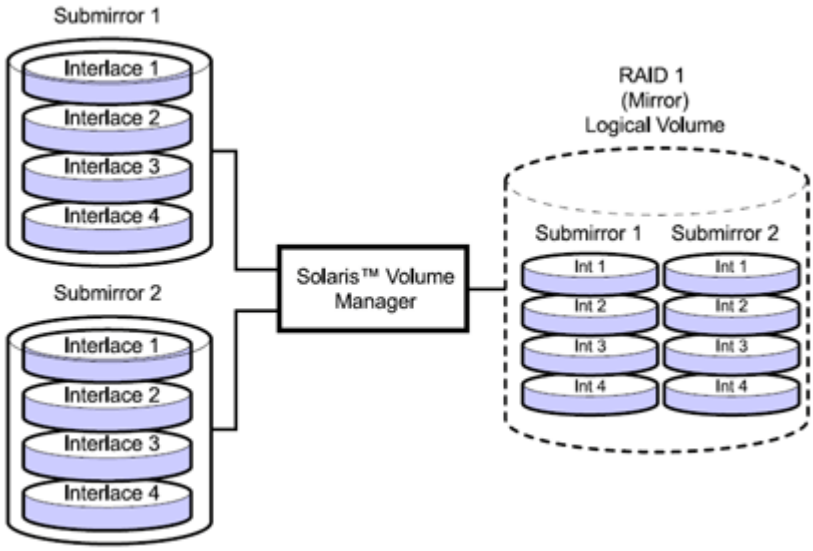

Рис.3 RAID-1. Пример зеркала (mirror)

Добавлять и отключать «вторичные зеркала» можно на работающей дисковой системе. Единственное требование заключается в том, чтобы хоть одно из этих зеркал было «активным». Вторичные зеркала доступны для монтирования в режиме «только чтения».

Как и в случае с RAID-0, вы можете создавать зеркала на всю партицию диска, либо только на выборочную его часть. Фактически, это определяется теми логическими томами, из которых данное зеркало собирается.

SVM поддерживает два вида RAID-1 устройств:

- RAID-0+1 – технология зеркалирования «распределенных» томов
- RAID-1+0 – технология распределения «зеркальных» томов

В случае использования технологии RAID-0+1, выход из «строя» одной из частей «распределенного» тома повлечет за собой нарушение работы всего «вторичного зеркала».

Пример организации технологии RAID-0+1 представлен на следующем рисунке:

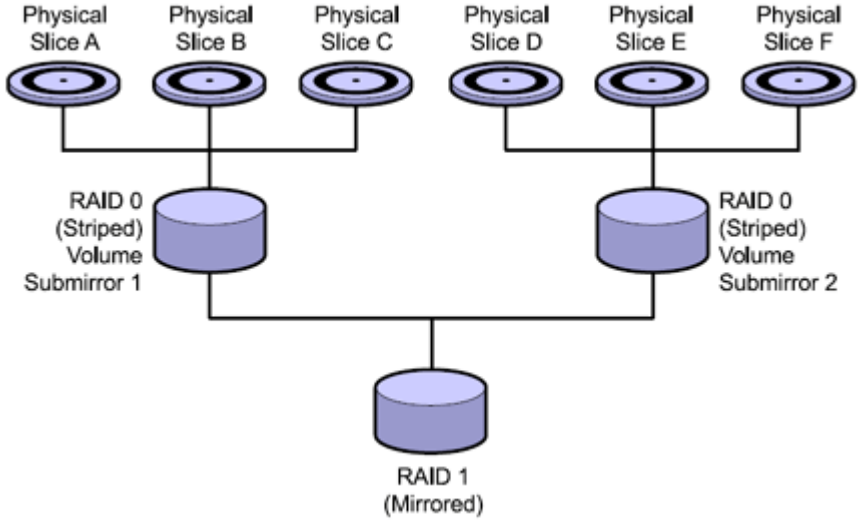

Рис.4 RAID-0+1

В отличие от RAID-0+1, технология 1+0 более защищена от возможных сбоев – при выходе из «строя» одного из компонентов в любом зеркале, нарушения работы не произойдет.

Как видно на следующем рисунке, RAID-1+0 представляет собой совокупность зеркальных томов, объединенных в «распределенное» логическое устройство:

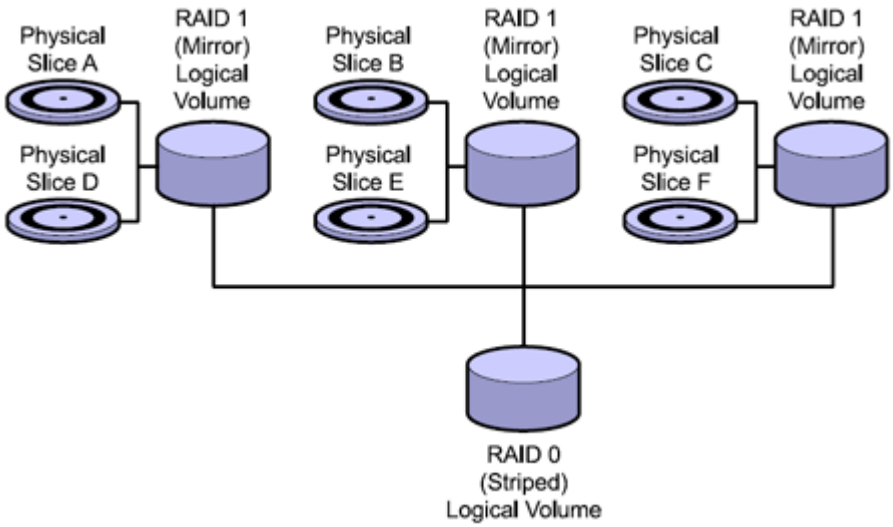

Рис.5 RAID-1+0

При конфигурировании RAID-1 устройств необходимо учитывать ряд определяющих параметров, которые влияют на производительность созданного зеркала. Эти параметры устанавливаются как для операций чтения, так и для операций записи, и могут быть изменены по необходимости в любое время.

Для операций чтения задаются следующие параметры:

Правила	**Описание**
Round Robin (default)	Сбалансированная загрузка между зеркалами
Geometric	Позволяет разделять операции чтения между зеркалами на основе адресных блоков логических дисков
First	Направлять все операции чтения в «главное» зеркало

Для операций записи задаются следующие параметры:

Правила	**Описание**
Parallel (Default)	Осуществляет запись данных одновременно (параллельно) на все зеркала в логическом томе
Serial	Указывает на то, что запись на следующее зеркало будет начата только по завершении операции записи на предыдущее

Для того чтобы технология RAID-1 обеспечила максимальную надежность и производительность, рекомендуется соблюдать несколько правил:

- Размер «вторичных зеркал» должен быть не меньше размера «главного зеркала». Больше – можно, но экономически не целесообразно, поскольку избыточная область просто «повиснет в воздухе»
- Размещайте части зеркала на разных дисках, а диски на разных контроллерах. Это обеспечит высокую надежность и производительность
- Старайтесь использовать диски одной модели и не размещайте их на контроллерах разного типа (SCSI, IDE и т.д.)
- Не рекомендуется монтировать «вторичные зеркала» до тех пор, пока они не переведены в состояние offline
- Экспериментируйте с параметрами чтения/записи для получения максимальной производительности зеркального тома

RAID-5. Данный тип устройств очень похож на технологию RAID-0, но, в отличие от «распределенных» томов, содержит специальную область на каждом диске с данными контроля четности одного из компонентов тома.

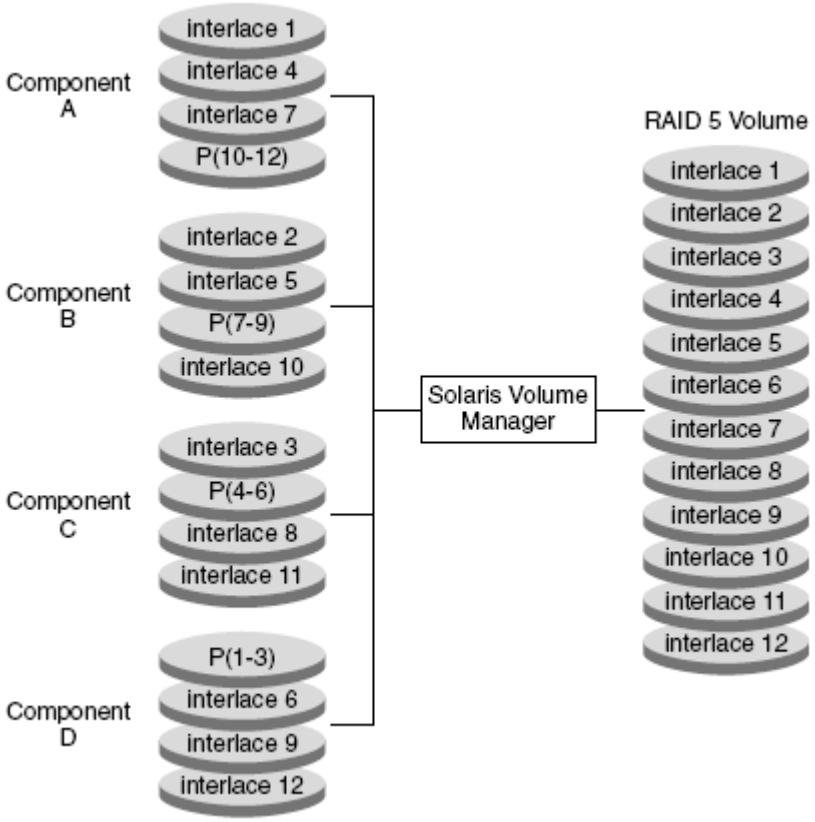

Рис.5 RAID-5

Если по какой-либо причине компонент RAID-5 системы выходит из «строя», то данные, которые были на нем, автоматически восстановятся на основании информации с других компонентов тома. Для того чтобы данный механизм успешно функционировал, количество дисков, объединяемых в RAID-5, должно быть больше трех. Если вы не используете Hot Spare Pools (пул дисков «горячей» замены), то система может восстановить только один поврежденный компонент тома.

Суммарный размер RAID-5 тома равен количеству дисков, объединенных в том, уменьшенному на размер одного дискового носителя.

RAID-5 нельзя использовать для разделов: /(root), /usr и swap, а также для уже существующих файловых систем.

В заключение рассмотрения теории RAID-устройств, хочу привести таблицу сравнительных характеристик для описанных выше технологий:

	RAID-0 (concatenation)	**RAID-0 (stripe)**	**RAID-1**	**RAID-5**
Избыточность данных	Нет	Нет	Да	Да
Повышение производительности (чтение)	Нет	Да	По разному	Да
Повышение производительности (запись)	Нет	Да	Нет	Нет
Увеличение размера дисковой системы	Да	Да	Нет	Да

Hot Spare Pools (пулы «горячей» замены). Пулы «горячей» замены предоставляют возможность повысить надежность функционирования RAID-1 и RAID-5 устройств.

Они представляют собой набор дисковых частей, которые по мере необходимости используются вместо поврежденных частей RAID-тома. Когда происходит ошибка в работе RAID, система просматривает пул и выбирает доступные и подходящие по размеру предварительно заготовленные «разделы замены». Найдя подходящий раздел, она помечает его как “In-Use”, включает в RAID-том и проводит синхронизацию данных.

Вы можете создавать и изменять пулы, назначать их разным RAID устройствам, но сами эти устройства могут ассоциироваться только с одним объявленным пулом.

RAID устройства используют пулы до тех пор, пока поврежденные части не будут заменены. После этого ресурс пула освобождается, устанавливается в состояние “Available” и может использоваться снова.

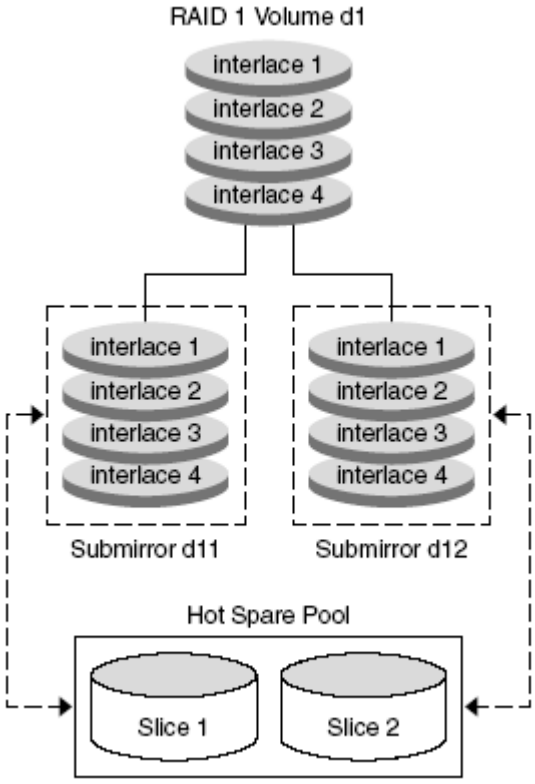

Рис.6 Hot Spare Pools (пулы «горячей» замены)

SVM. Конфигурирование

Перед тем как приступать к созданию логических устройств, необходимо выполнить ряд подготовительных действий:

1. составить структуру будущего логического тома и определить необходимые для этого компоненты. На этом этапе вы определяете: какие разделы будут организованы в логические устройства, на каких дисковых носителях они будут расположены и как необходимо форматировать эти диски.
2. подготовить дисковые разделы, согласно составленной структуре, для включения их в логические тома. Используя команду format либо Solaris Management Console, вы «разбиваете» диски на необходимые части.
3. создать реплики базы структуры SVM (State Database replica)

Для работы SVM обязательным требованием является наличие одной копии базы структуры SVM. В этой базе хранится информация обо всех логических устройствах и их конфигурациях, созданных SVM. Пока эта база не будет сформирована, данный продукт функционировать не будет.

Для повышения надежности работы данной технологии, рекомендуется создавать несколько копий данной базы – реплик. Если SVM не сможет прочитать информацию из базы, то вся дисковая структура будет безвозвратно потеряна. Оптимальным решением для данного вопроса является создание 3-5 реплик на каждом дисковом носителе, которые

используются в SVM. В случае если в логические тома объединяются пять и выше дисковых носителей, то количество реплик можно свести по одной на диск. Таким образом, вы создаете необходимую избыточность данных, и в случае потери информации на одном из носителей необходимая конфигурация будет прочитана с «рабочих» копий.

В целях повышения надежности работы, SVM не будет стартовать до тех пор, пока количество «рабочих» реплик не превысит количество поврежденных. Если количество поврежденных реплик достигнет половины в момент работы системы, то SVM будет продолжать свое функционирование, а система будет постоянно сообщать о критической ситуации (system panic).

Реплики обязательно создаются на выделенных частях (slices) диска и по объему занимают приблизительно 4-5Мб на каждую реплику. Эти дисковые части не должны использоваться другими задачами.

Рассмотрим теперь практическую реализацию данных механизмов. В качестве примера, рассмотрим технологию создания зеркала корневого (/) файлового раздела.

Для начала, используя команду *format*, «разобьем» имеющиеся в системе диски и создадим два раздела – один для реплик, другой – для зеркала.

1. Используем команду *format* и создаем на будущем «зеркальном» диске раздел для реплик. Я планирую создать по 3 реплики на диск и с учетом этого, раздел размечаю объемом 15Мб (5Мбх3=15Мб):

```
# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
       0. c1t0d0 <HITACHI-DK32EJ36NSUN36G-PQ0B cyl 24620 alt 2 hd 27 sec 107>
          /pci@1c,600000/scsi@2/sd@0,0
       1. c1t1d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /pci@1c,600000/scsi@2/sd@1,0
Specify disk (enter its number): 1      - выбираем второй диск
selecting c1t1d0
[disk formatted]
Warning: Current Disk has mounted partitions.

FORMAT MENU:
        disk       - select a disk
        type       - select (define) a disk type
        partition  - select (define) a partition table
        current    - describe the current disk
        format     - format and analyze the disk
        repair     - repair a defective sector
        label      - write label to the disk
        analyze    - surface analysis
        defect     - defect list management
        backup     - search for backup labels
        verify     - read and display labels
        save       - save new disk/partition definitions
        inquiry    - show vendor, product and revision
        volname    - set 8-character volume name
        !<cmd>     - execute <cmd>, then return
        quit
format> partition          - переходим в раздел «разбиения» диска
partition> print           - смотрим на существующую структуру

Current partition table (original):
Total disk cylinders available: 24620 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm     512 -  1237        1.00GB    (726/0/0)    2097414
  1 unassigned    wu       0                0         (0/0/0)            0
  2     backup    wm       0 - 24619       33.92GB    (24620/0/0) 71127180
  3 unassigned    wu       0                0         (0/0/0)            0
  4 unassigned    wu       0                0         (0/0/0)            0
  5 unassigned    wm       0                0         (0/0/0)            0
  6 unassigned    wm       0                0         (0/0/0)            0
  7       home    wm    1238 - 15756       20.00GB    (14519/0/0) 41945391

partition> 3  - выбираем 3 раздел, который будет использоваться для реплик

Part      Tag    Flag     Cylinders         Size            Blocks
  3 unassigned    wu       0                0         (0/0/0)            0

Enter partition id tag[unassigned]:  - оставляем не «используемым»
Enter partition permission flags[wu]:  - читать/записывать и не монтировать
Enter new starting cyl[0]: 15757 – стартовый цилиндр
Enter partition size[0b, 0c, 15757e, 0.00mb, 0.00gb]: 15m  - размер 15Мб
partition> print
Current partition table (unnamed):
Total disk cylinders available: 24620 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm     512 -  1237        1.00GB    (726/0/0)    2097414
  1 unassigned    wu       0                0         (0/0/0)            0
  2     backup    wm       0 - 24619       33.92GB    (24620/0/0) 71127180
  3 unassigned    wu   15757 - 15767       15.52MB    (11/0/0)       31779
  4 unassigned    wu       0                0         (0/0/0)            0
  5 unassigned    wm       0                0         (0/0/0)            0
```

2. Создадим раздел для зеркала. Поскольку на моем системном диске корневой (/) раздел имеет размер 512Мб, то и на «зеркальном» диске я создаю аналогичный. Заметьте, что номер создаваемого «зеркального» раздела может отличаться от номера зеркалируемого (на системном диске номер партиции=0):

```
partition> 4
Part      Tag    Flag     Cylinders         Size            Blocks
  4 unassigned    wu       0               0         (0/0/0)           0

Enter partition id tag[unassigned]:
Enter partition permission flags[wu]: wm
Enter new starting cyl[0]: 15768
Enter partition size[0b, 0c, 15768e, 0.00mb, 0.00gb]: 512m
partition> print
Current partition table (unnamed):
Total disk cylinders available: 24620 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm     512 -  1237        1.00GB    (726/0/0)    2097414
  1 unassigned    wu       0               0         (0/0/0)            0
  2     backup    wm       0 - 24619       33.92GB    (24620/0/0) 71127180
  3 unassigned    wu   15757 - 15767       15.52MB    (11/0/0)       31779
  4 unassigned    wm   15768 - 16130      512.06MB    (363/0/0)    1048707
  5 unassigned    wm       0               0         (0/0/0)            0
  6 unassigned    wm       0               0         (0/0/0)            0
  7       home    wm    1238 - 15756       20.00GB    (14519/0/0) 41945391

partition> label    - сохраним все изменения на диск
Ready to label disk, continue? y

partition> q


FORMAT MENU:
        disk       - select a disk
        type       - select (define) a disk type
        partition  - select (define) a partition table
        current    - describe the current disk
        format     - format and analyze the disk
        repair     - repair a defective sector
        label      - write label to the disk
        analyze    - surface analysis
        defect     - defect list management
        backup     - search for backup labels
        verify     - read and display labels
        save       - save new disk/partition definitions
        inquiry    - show vendor, product and revision
        volname    - set 8-character volume name
        !<cmd>     - execute <cmd>, then return
        quit
format>q
#
```

Таким образом, создаете на всех дисках разделы для реплик, и где необходимо разделы для будущих логических томов.

3. Создадим три реплики базы SVM в только что созданном разделе */dev/dsk/c1t1d0s3* и проверим результат:

```
# metadb -a -f -c 3 /dev/dsk/c1t1d0s3


# metadb -i
      flags       first blk   block count
     a        u         16          8192        /dev/dsk/c1t1d0s3
     a        u         8208        8192        /dev/dsk/c1t1d0s3
     a        u         16400       8192        /dev/dsk/c1t1d0s3
 r - replica не содержит информации о структуре устройства
 o - replica установлена согласно последней модификации mddb
 u - replica обновлена
 l - местоположение реплики прочитано успешно
 c - replica расположена в /etc/lvm/mddb.cf
 p - расположение реплики изменено в ядре
 m - главная replica, используется в режиме чтения
 W - проблемы с операциями записи на устройстве
 a - replica в «активном» состоянии
 M - проблемы с master blocks
 D - проблемы с data blocks
 F - проблемы с форматированием
 S - replica слишком мала для размещения информации
 R - проблемы с операциями чтения с устройства
#
```

При создании реплик можно одновременно указывать все диски, на которых они будут размещены (через пробел).

После создания реплик и разделов-заготовок для логических томов, можно непосредственно приступать к конфигурированию RAID устройств.

Для начала, необходимо создать RAID-0 массивы, которые, впоследствии, будут объединены в зеркало RAID-1.

1. Создаем RAID-0 (concatenation) устройство d12 для нашего будущего зеркального раздела из *c1t1d0s4*.

Для создания логических устройств используется команда *metainit*. Опции для этой команды определяют тип и структуру создаваемого устройства.

```
# /usr/sbin/metainit -f d12 1 1 c1t1d0s4
d12: Concat/Stripe is setup
```

Первая «1» определяет общее количество разделов в данном устройстве, а вторая «1» - количество разделов, которые будут «подключены» в данное устройство. В нашем случае – это обязательно должны быть «1».

Для примера, приведу команду создания распределенного устройства, состоящего из трех разделов:

```
# metainit d20 1 3 c0t1d0s2 c0t2d0s2 c0t3d0s2
d20: Concat/Stripe is setup
```

и пример создания «склеенного» устройства, состоящего из четырех разделов:

```
# metainit d40 4 1 c0t1d0s2 1 c0t2d0s2 1 c0t2d0s3 1 c0t2d1s3
d40: Concat/Stripe is setup
```

2. Создадим, используя в данном случае Solaris Management Console, RAID-0 устройство d11 из нашего функционирующего корневого раздела:

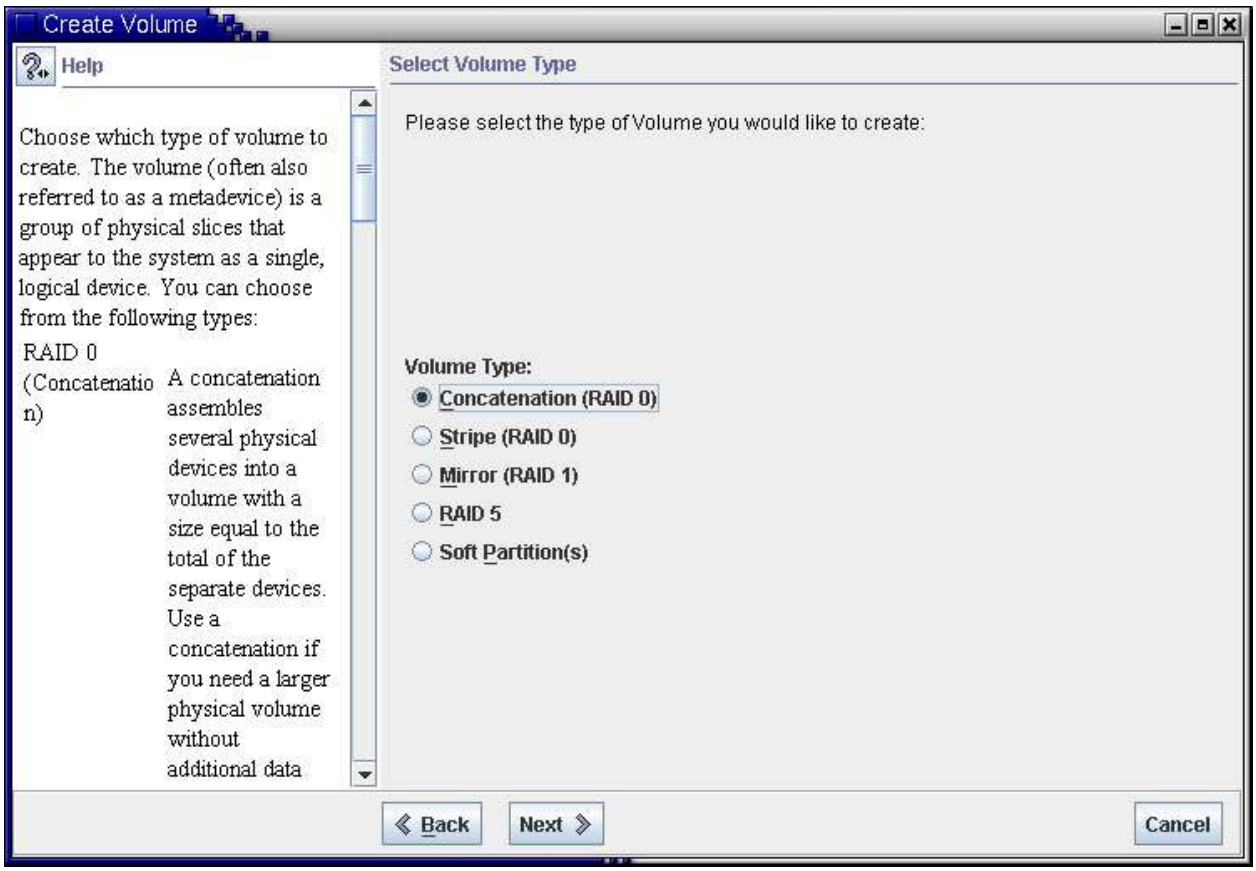

Рис.7 Выбор типа создаваемого логического устройства

Нам необходимо выбрать «склеенное» устройство. Во всех остальных случаях мы уничтожим наш «рабочий» корневой раздел!

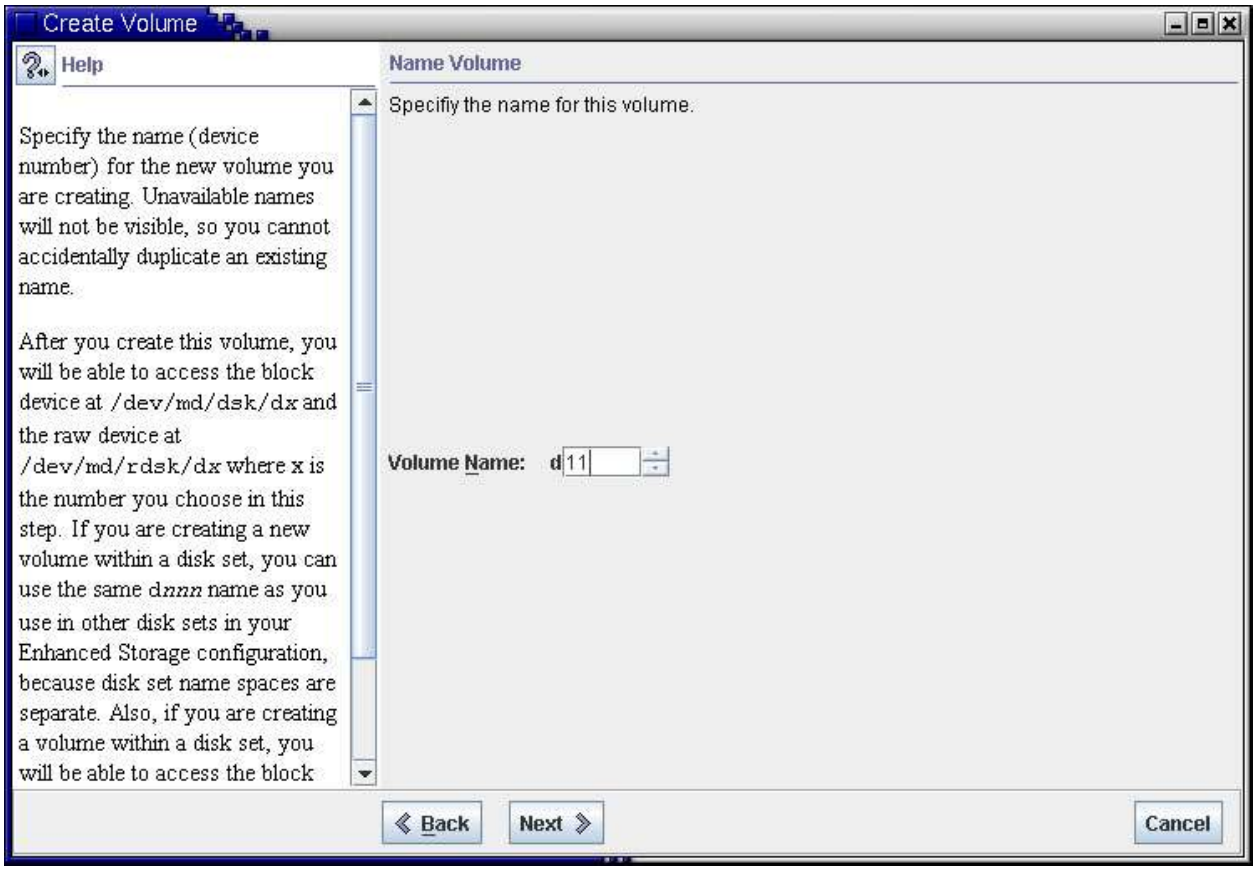

Рис.8 Выбор номера создаваемого логического устройства

После определения номера устройства выбираем раздел, который хотим подключить в RAID-0. В нашем случае – это корневой раздел *c1t0d0s0*:

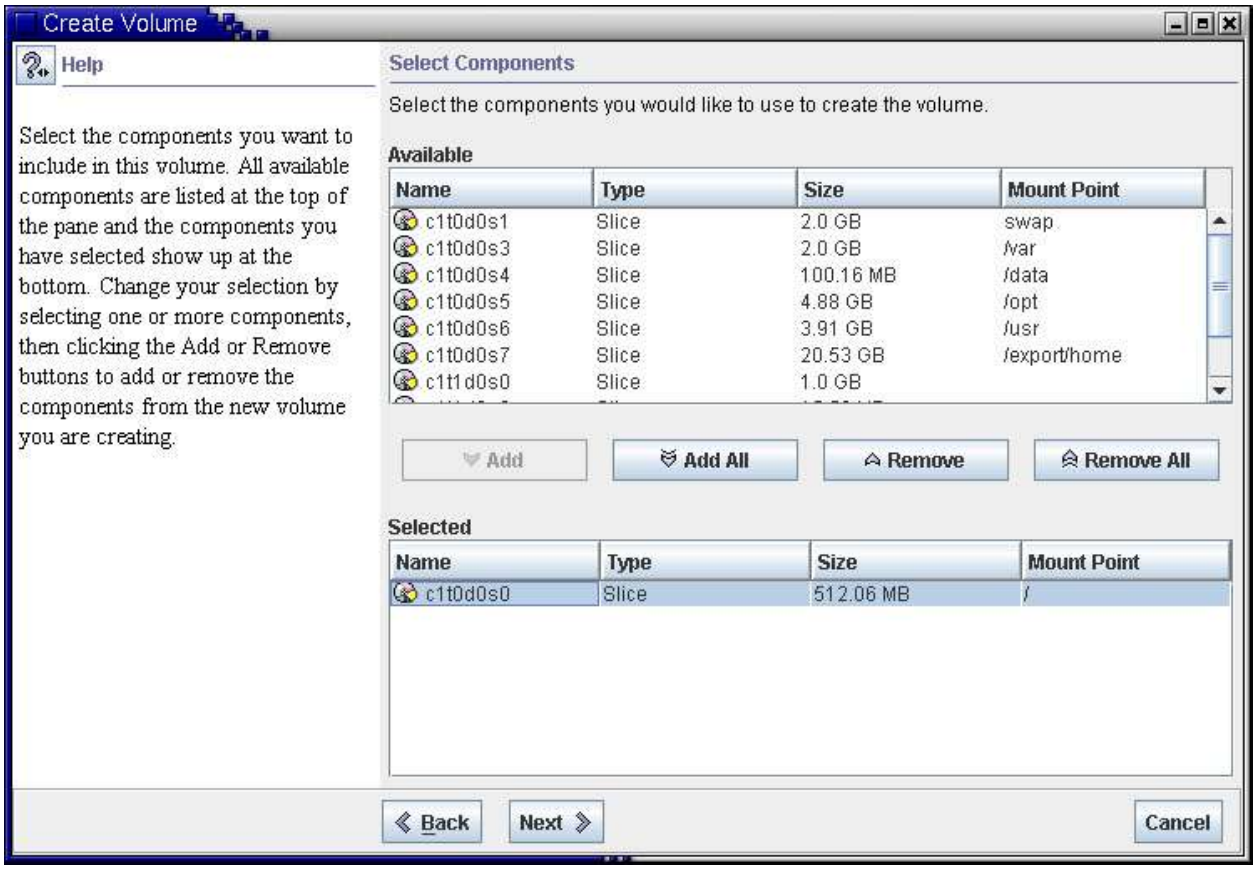

Рис.9 Выбор компонентов создаваемого логического устройства

Следующим этапом определяем пулы «горячей» замены для данного RAID устройства. Поскольку мы пулов еще не создавали, просто пропускаем этот ход:

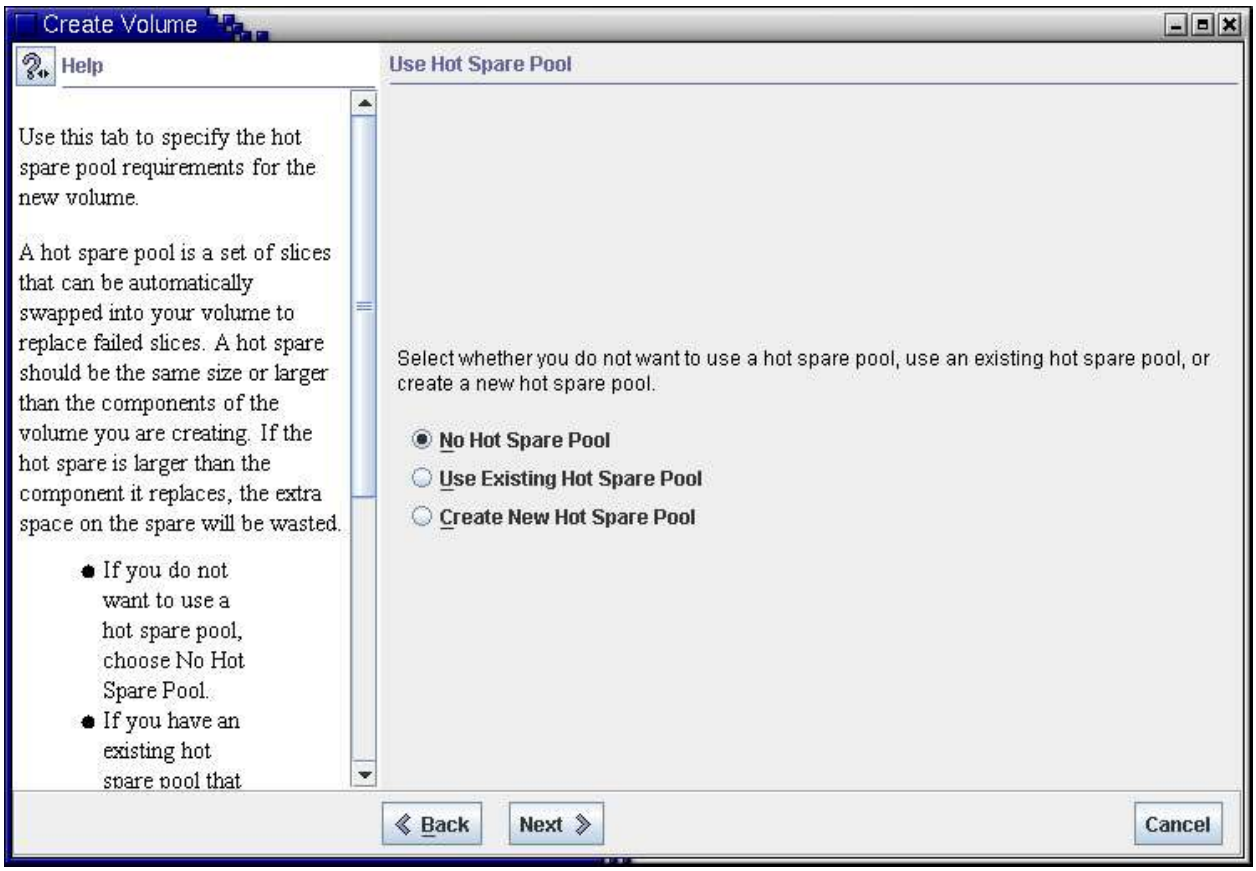

Рис.10 Выбор пулов «горячей» замены

И в заключении, подтверждаем введенные выше значения конфигурации.

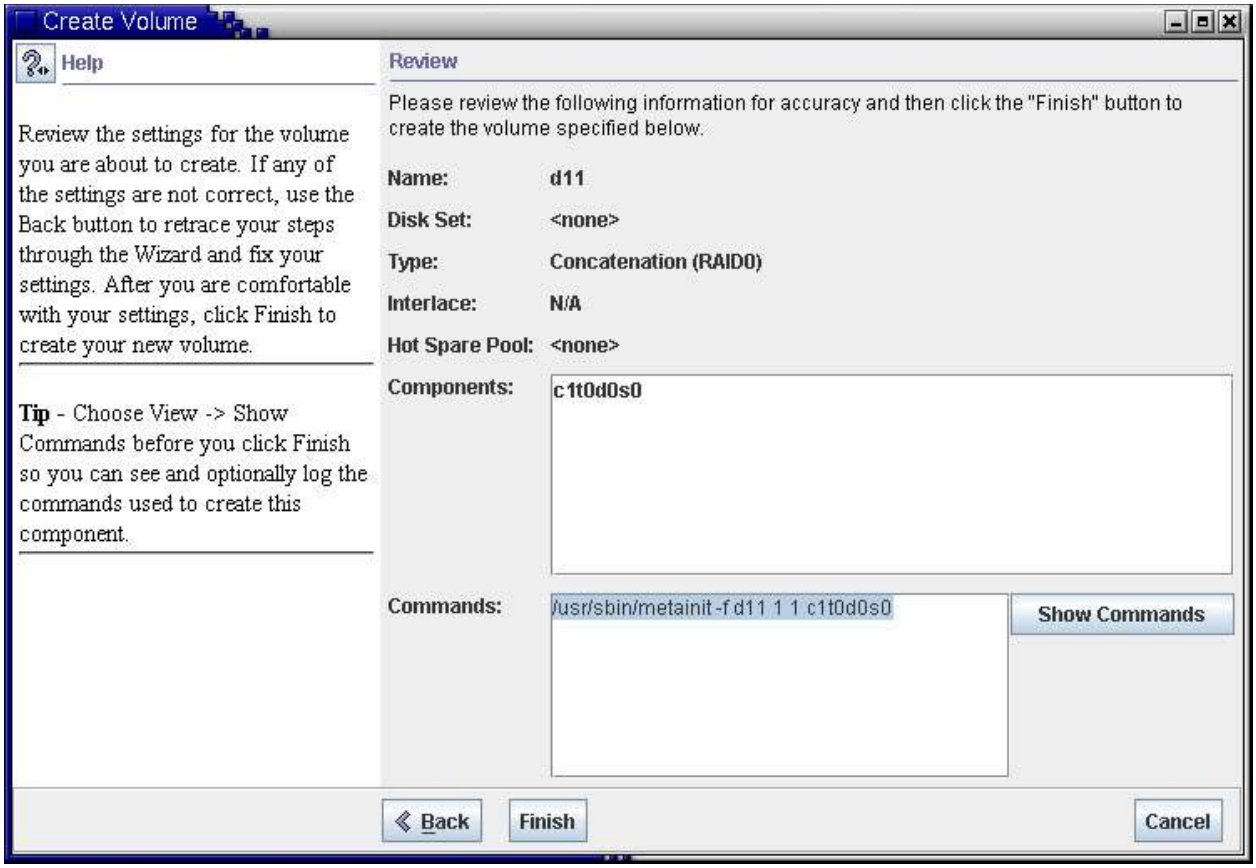

Рис.11 Создание устройства

Если создавать устройство d11 с командной строки, то ее вид будет таким, как он представлен на Рис.11

3. Следующим шагом мы определяем «главное зеркало». Для этого необходимо, используя команду *metainit*, подключить наше устройство d11 в будущее зеркало – d10. Номера устройств, кстати, вы можете задавать любые доступные.

```
# /usr/sbin/metainit d10 -m d11
d10: Mirror is setup
```

4. Указываем системе, что «boot»-устройством у нас является созданное нами зеркало – d10:

```
# metaroot d10
```

И смотрим на те изменения, которые произошли в системе при выполнении данной команды:

```
# cat /etc/system
...
...
* Begin MDD root info (do not edit)
rootdev:/pseudo/md@0:0,10,blk
* End MDD root info (do not edit)
#
```

```
# cat /etc/vfstab
#device           device            mount       FS    fsck  mount mount
#to mount   to fsck           point       type  pass  at boot     options
#
fd    -     /dev/fd     fd    -     no    -
/proc -     /proc proc  -     no    -
/dev/dsk/c1t0d0s1 -     -     swap  -     no    -
/dev/md/dsk/d10   /dev/md/rdsk/d10  /     ufs   1     no    -
/dev/dsk/c1t0d0s6 /dev/rdsk/c1t0d0s6      /usr  ufs   1     no    -
/dev/dsk/c1t0d0s3 /dev/rdsk/c1t0d0s3      /var  ufs   1     no    -
/dev/dsk/c1t0d0s4 /dev/rdsk/c1t0d0s4      /data ufs   2     yes   -
/dev/dsk/c1t0d0s7 /dev/rdsk/c1t0d0s7      /export/home      ufs   2     yes   -
/dev/dsk/c1t0d0s5 /dev/rdsk/c1t0d0s5      /opt  ufs   2     yes   -
/dev/dsk/c1t1d0s7 /dev/rdsk/c1t1d0s7      /virtual    ufs   2     yes   -
/devices    -     /devices    devfs -     no    -
swap  -     /tmp  tmpfs -     yes   -
```

5. После этого необходимо перезагрузить систему:

```
# init 6
```

6. После того, как система загрузиться, подключаем к нашему зеркалу d10 «вторичное зеркало» - d12:

```
# metattach d10 d12
d10: submirror d12 is attached
```

7. Теперь можно посмотреть на состояние логического устройства:

```
# metastat d10
d10: Mirror
    Submirror 0: d11
      State: Okay
    Submirror 1: d12
      State: Resyncing
    Resync in progress: 6 % done
    Pass: 1
    Read option: roundrobin (default)
    Write option: parallel (default)
    Size: 1048707 blocks (512 MB)

d11: Submirror of d10
    State: Okay
    Size: 1048707 blocks (512 MB)
    Stripe 0:
      Device     Start Block  Dbase        State Reloc Hot Spare
      c1t0d0s0          0     No            Okay   Yes


d12: Submirror of d10
    State: Resyncing
    Size: 1048707 blocks (512 MB)
    Stripe 0:
      Device     Start Block  Dbase        State Reloc Hot Spare
      c1t1d0s4          0     No            Okay   Yes


Device Relocation Information:
Device   Reloc    Device ID
c1t0d0   Yes
id1,sd@w484954414348492044 4b3332454a2d33364e432020202020343335 4e39373135
c1t1d0   Yes      id1,sd@SSEAGATE_ST336607LSUN36G_3JA0FAX0000073248QWL
#
```

Как видно на примере, все устройства имеют статус «Okay» и на момент выполнения команды *metastat* выполнено 6% синхронизации. Для полного соответствия содержимого зеркал необходимо 100% выполнение синхронизации.

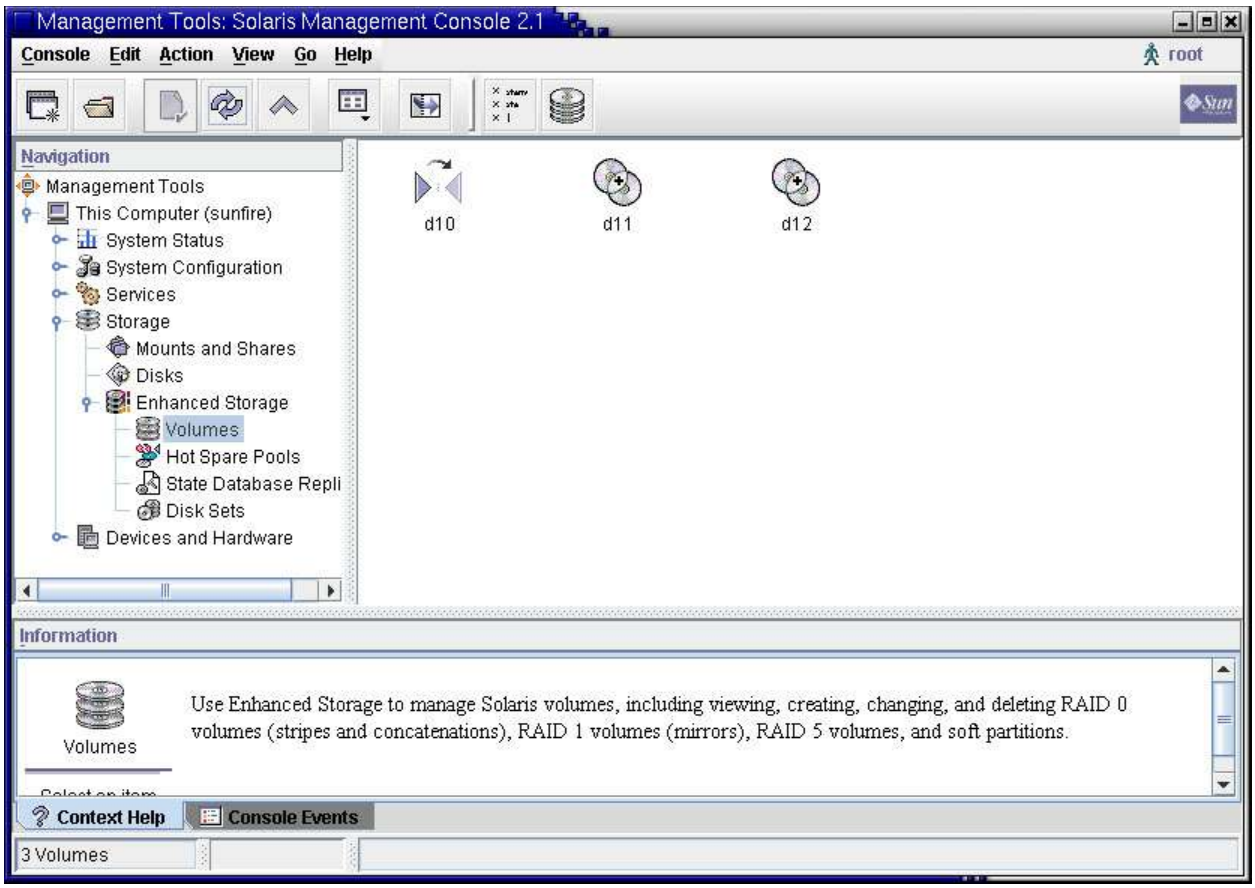

Рис.12 Сконфигурированные логические устройства

8. Посмотрим на файловые системы и попробуем смонтировать логические тома d10, d11 и d12 в каталог */mnt*:

```
# df
/                  (/dev/md/dsk/d10   ):  827150 blocks   242380 files
/devices           (/devices          ):       0 blocks        0 files
/usr               (/dev/dsk/c1t0d0s6 ): 4443604 blocks   432616 files
/proc              (proc              ):       0 blocks    29880 files
/etc/mnttab        (mnttab            ):       0 blocks        0 files
/dev/fd            (fd                ):       0 blocks        0 files
/var               (/dev/dsk/c1t0d0s3 ): 3824184 blocks   329094 files
/var/run           (swap              ): 7118416 blocks   292875 files
/tmp               (swap              ): 7118416 blocks   292875 files
/virtual           (/dev/dsk/c1t1d0s7 ):40920852 blocks  2546500 files
/data              (/dev/dsk/c1t0d0s4 ):  189318 blocks    54072 files
/opt               (/dev/dsk/c1t0d0s5 ): 9922772 blocks   623902 files
/export/home       (/dev/dsk/c1t0d0s7 ):42260308 blocks  2624203 files
#


# mount -F ufs -o ro /dev/md/dsk/d11 /mnt
# ls /mnt
bin         dev         home        mnt         proc        usr
cdrom       devices     kernel      net         sbin        var
data        etc         lib         opt         tftpboot    virtual
detach.txt  export      lost+found  platform    tmp         vol

# umount /mnt

# mount -F ufs /dev/md/dsk/d10 /mnt
mount: /dev/md/dsk/d10 is already mounted or /mnt is busy

# mount -F ufs -o ro /dev/md/dsk/d12 /mnt
# ls /mnt
bin          dev          kernel       opt          tmp
cdrom        devices      lib          platform     usr
data         etc          lost+found   proc         var
detach.txt   export       mnt          sbin         virtual
detach.txt~  home         net          tftpboot     vol

# umount /mnt
```

Для того чтобы можно было загружаться с «вторичного» зеркала, необходимо установить в него загрузочную область (bootblk) и в OpenBoot прописать альтернативное загрузочное устройство.

1. Установить bootblk можно с помощью команды *installboot*:
- для sparc-платформы команда будет выглядеть так:

```
# installboot /usr/platform/`uname -i`/lib/fs/ufs/bootblk  /dev/rdsk/c1t1d0s0
```

- для x86-платформы немного по другому:

```
# installboot /usr/platform/`uname -i`/lib/fs/ufs/pboot  \
  /usr/platform/`uname -i`/lib/fs/ufs/bootblk  /dev/rdsk/c1t1d0s2
```

2. Определяем физическое имя для нашего раздела *c1t1d0s4*:

```
# ls -l /dev/rdsk/c1t1d0s4
lrwxrwxrwx    1 root   root   47 Dec 28 09:05 /dev/rdsk/c1t1d0s4 -> ../../
devices/pci@1c,600000/scsi@2/sd@1,0:e,raw
```

При использовании полученного физического имени в OpenBoot, необходимо заменять «sd» на «disk»!

3. В «командном» режиме OpenBoot (можно использовать команду *eeprom* из консоли) задаем:

```
ok nvalias backup /pci@1c,600000/scsi@2/disk@1,0:e

ok printenv boot-device
boot-device= disk net

ok setenv boot-device disk backup net
boot-device= disk backup net
```

ok boot backup **- загрузимся с зеркала для проверки**

В случае нарушения работы основного диска, система будет автоматически загружаться с резервного зеркального раздела.

Если по каким-либо причинам вам необходимо удалить существующее RAID-1 устройство, то процедура будет выглядеть следующим образом:

1. Отключаем «вторичное зеркало»:

```
# metadetach d10 d12
d10: submirror d12 is detached
```

2. Указываем системе «boot»-устройство и перезагрузим ее:

```
# metaroot /dev/dsk/c1t0d0s0

# init 6
```

3. Удаляем зеркало d10 и ассоциированное с ним RAID-0 устройство d11:

```
# metaclear -r d10
d10: Mirror is cleared
d11: Concat/Stripe is cleared
```

4. Удаляем RAID-0 устройство d12:

```
# metaclear d12
d12: Concat/Stripe is cleared
```

RAID-5. Создание RAID-5 устройств намного проще, описанных выше RAID-0 и RAID-1. Для их установки достаточно выполнить всего одну команду:

```
# metainit d20 -r c1t0d0s2 c2t0d0s2 c3t0d0s2
d45: RAID is setup

# metastat d20
d20: RAID
State: Okay
Interlace: 32 blocks
Size: 10080 blocks
Original device:
Size: 8087040 blocks
Device Start Block Dbase State Hot Spare
c1t0d0s2 330 No Okay
c2t0d0s2 330 No Okay
c3t0d0s2 330 No Okay
```

Но необходимо помнить, что все данные на включаемых дисках будут уничтожены, и вам нужно конструировать на созданном логическом устройстве новую файловую систему.

Добавляются новые дисковые устройства в «рабочий» RAID-5 аналогично тому, как это делалось с зеркальными устройствами с помощью команды *metattach:*

```
# metattach d20 c4t0d0s2
d20: column is attached
```

Замена поврежденных томов в RAID-5 устройстве осуществляется с помощью механизма пулов «горячей» замены либо в «ручную», используя команду *metareplace*:

```
# metareplace d20 c3t0d0s2 c3t1d0s2
d20: device c3t0d0s2 is replaced with c3t1d0s2
```

Hot Spare Pools. Пулы «горячей» замены, как мы помним из теоретической части данной статьи, представляют собой набор резервных дисков для замены поврежденных компонентов RAID-устройств.

Создаются пулы с помощью команды *metainit*:

```
# metainit hsp001 c2t2d0s2 c3t2d0s2
hsp001: Hotspare pool is setup

# metastat hsp001
hsp001: 1 hot spare
c2t2d0s2 Available 16800 blocks
c3t2d0s2 Available 16800 blocks
```

Добавить диски в существующий пул можно так:

```
# metahs -a hsp001 /dev/dsk/c3t0d0s2
hsp001: Hotspare is added

# metastat hsp001
hsp001: 1 hot spare
c2t2d0s2 Available 16800 blocks
c3t2d0s2 Available 16800 blocks
c3t0d0s2 Available 16800 blocks
```

Если у нас создано несколько пулов, и мы хотим добавить данный диск во все имеющиеся у нас пулы «горячей» замены, то необходимо добавить опцию –all:

```
# metahs -a -all /dev/dsk/c3t0d0s2
hsp001: Hotspare is added
hsp002: Hotspare is added
hsp003: Hotspare is added
```

После того как пулы созданы, их необходимо назначить RAID-устройствам, которые будут ими пользоваться в случае необходимости.

Добавим к созданному нами зеркалу d10, точнее к его компонентам d11 и d12, пул «горячей» замены - hsp001:

```
# metaparam -h hsp001 d11
# metaparam -h hsp001 d12

# metastat d10
d10: Mirror
Submirror 0: d11
State: Okay
Submirror 1: d12
State: Okay

d11: Submirror of d10
State: Okay
Hot spare pool: hsp001

d12: Submirror of d10
State: Okay
Hot spare pool: hsp001
```

Для RAID-5 процедура ассоциации с пулом «горячей» замены будет выглядеть следующим образом:

```
# metaparam -h hsp001 d20
# metastat d20
d20: RAID
State: Okay
Hot spare pool: hsp001
...
```

Если необходимо изменить пул, который используется для RAID-устройства, то выполнить это можно с помощью команды *metaparam* и опции –h:

```
# metastat d20
d20: RAID
State: Okay
Hot spare pool: hsp001
...
# metaparam -h hsp002 d20

# metastat d20
d20: RAID
State: Okay
Hot spare pool: hsp002
...
```

А открепить пул можно так:

```
# metastat d20
d20: RAID
State: Okay
Hot spare pool: hsp002
...
# metaparam -h none d20
# metastat d20
d20: RAID
State: Okay
Hot spare pool:
...
```

Заключение

С полной информацией о технологии Solaris Volume Manager можно ознакомиться на сайте http://docs.sun.com, в материалах «Solaris 10. “Solaris Volume Manager administration Guide”.