

ViPNet Coordinator Linux 4. Система защиты от сбоев

Руководство администратора



1991–2015 ОАО «ИнфоТеКС», Москва, Россия

ФРКЕ.00132-02 32 02

Этот документ входит в комплект поставки программного обеспечения, и на него распространяются все условия лицензионного соглашения.

Ни одна из частей этого документа не может быть воспроизведена, опубликована, сохранена в электронной базе данных или передана в любой форме или любыми средствами, такими как электронные, механические, записывающие или иначе, для любой цели без предварительного письменного разрешения ОАО «ИнфоТеКС».

ViPNet[®] является зарегистрированным товарным знаком ОАО «ИнфоТеКС».

В продукте использованы изобретения, защищенные патентами РФ №№ 2517411, 2526282, 2507569.

Все названия компаний и продуктов, которые являются товарными знаками или зарегистрированными товарными знаками, принадлежат соответствующим владельцам.

ОАО «ИнфоТеКС»

127287, г. Москва, Старый Петровско-Разумовский пр., дом 1/23, строение 1

Тел: (495) 737-61-96 (hotline), 737-61-92, факс 737-72-78

Сайт компании «ИнфоТеКС»: <http://www.infotecs.ru>

Электронный адрес службы поддержки: hotline@infotecs.ru

Содержание

Введение.....	5
О данном документе	6
Соглашения документа	7
Обратная связь.....	8
Глава 1. Общие сведения	9
Назначение системы защиты от сбоев.....	10
Принципы работы системы защиты от сбоев	11
Одиночный режим	11
Режим кластера горячего резервирования	12
Требования к системе.....	15
Глава 2. Установка системы защиты от сбоев	17
Глава 3. Обновление версии ПО ViPNet Coordinator Linux.....	19
Глава 4. Настройка системы защиты от сбоев	21
Файл конфигурации системы защиты от сбоев.....	22
Секция [channel].....	23
Секция [network].....	25
Секция [sendconfig]	27
Возможные конфликты при резервировании файлов.....	29
Секция [misc]	30
Секция [debug]	31
Типовая схема организации кластера горячего резервирования	32
Схема организации кластера в условиях ограничений по выделению IP-адресов	35
Ограничения при использовании дополнительных адресов на интерфейсах кластера	39
Глава 5. Запуск системы защиты от сбоев и работа с ней	42
Глава 6. Просмотр информации о работе системы защиты от сбоев	44
Утилита для просмотра информации о работе системы защиты от сбоев	45
Текущее состояние.....	46
Журнал переключений.....	48
Глава 7. Тонкая настройка системы защиты от сбоев.....	49

Нестандартная конфигурация сетевых настроек.....	50
Использование кластера горячего резервирования серверов совместно с ОС Solaris	51
Работа кластера горячего резервирования серверов совместно с коммутационным оборудованием Cisco.....	52
Приложение А. Задание дополнительных IP-адресов на кластере.....	53



Введение

О данном документе	6
Соглашения документа	7
Обратная связь	8

О данном документе

Данный документ предназначен для администраторов, отвечающих за настройку и поддержку кластера горячего резервирования, организованного на базе ПО ViPNet Coordinator Linux. В нем описаны назначение и принципы работы системы защиты от сбоев, обеспечивающей устойчивость к сбоям как одиночного сервера, так и кластера горячего резервирования серверов.

Описание системы защиты от сбоев, функционирующей на одиночном сервере, приведено в документе «ViPNet Coordinator Linux. Руководство администратора». Данный документ содержит подробное описание системы защиты от сбоев при функционировании в режиме кластера горячего резервирования. В документе приведены схемы организации кластера и примеры настройки параметров для этих схем, команды для управления кластером, а также описаны особенности обновления ПО на кластере.

Соглашения документа

Ниже перечислены соглашения, принятые в этом документе для выделения информации.

Таблица 1. Обозначения, используемые в примечаниях




Обозначение	Описание
	Внимание! Указывает на обязательное для исполнения или следования действие или информацию.
	Примечание. Указывает на необязательное, но желательное для исполнения или следования действие или информацию.
	Совет. Содержит дополнительную информацию общего характера.

Таблица 2. Обозначения, используемые для выделения информации в тексте

Обозначение	Описание
Название	Название элемента интерфейса. Например, заголовок окна, название поля, кнопки или клавиши.
Клавиша+Клавиша	Сочетание клавиш. Чтобы использовать сочетание клавиш, следует нажать первую клавишу и, не отпуская ее, нажать вторую клавишу.
Меню > Подменю > Команда	Иерархическая последовательность элементов. Например, пункты меню или разделы на панели навигации.
Код	Имя файла, путь, фрагмент текстового файла (кода) или команда, выполняемая из командной строки.

При описании команд в данном документе используются следующие условные обозначения:

- Команды, которые могут быть выполнены только в режиме администратора, выделены красным цветом. Например:
`команда`
- Параметры, которые должны быть заданы пользователем, заключены в угловые скобки. Например:
`команда <параметр>`
- Необязательные параметры или ключевые слова заключены в квадратные скобки. Например:
`команда <обязательный параметр> [необязательный параметр]`
- Если при вводе команды можно указать один из нескольких параметров, допустимые варианты заключены в фигурные скобки и разделены вертикальной чертой. Например:
`команда {вариант-1 | вариант-2}`

Обратная связь

Дополнительная информация

Сведения о продуктах и решениях ViPNet, распространенные вопросы и другая полезная информация собраны на сайте ОАО «ИнфоТекС»:

- Веб-портал документации ViPNet <http://docs.infotecs.ru>.
- Описание продуктов ViPNet <http://www.infotecs.ru/products/line/>.
- Информация о решениях ViPNet <http://www.infotecs.ru/solutions/>.
- Сборник часто задаваемых вопросов (FAQ) <http://www.infotecs.ru/support/faq/>.
- Форум пользователей продуктов ViPNet <http://www.infotecs.ru/forum>.
- Законодательная база в сфере защиты информации <http://www.infotecs.ru/laws/>.

Контактная информация

С вопросами по использованию продуктов ViPNet, пожеланиями или предложениями свяжитесь со специалистами ОАО «ИнфоТекС». Для решения возникающих проблем обратитесь в службу технической поддержки.

- Техническая поддержка для пользователей продуктов ViPNet: hotline@infotecs.ru.
- Форма запроса в службу технической поддержки <http://www.infotecs.ru/support/request/>.
- Регистрация продуктов и консультации по телефону для клиентов, имеющих расширенный уровень технического сопровождения:

8 (495) 737-6196,

8 (800) 250-0260 — бесплатный звонок из любого региона России (кроме Москвы).

Распространение информации об уязвимостях продуктов ОАО «ИнфоТекС» регулируется политикой ответственного разглашения <http://infotecs.ru/products/disclosure.php>. Если вы обнаружили уязвимости в продуктах компании, сообщите о них по адресу security-notifications@infotecs.ru.

1

Общие сведения

Назначение системы защиты от сбоев	10
Принципы работы системы защиты от сбоев	11
Требования к системе	15

Назначение системы защиты от сбоев

Система защиты от сбоев предназначена для создания отказоустойчивого решения на базе ПО ViPNet Coordinator Linux. Данная система может функционировать в двух режимах:

- 1 Одиночный режим (режим одиночного сервера).
- 2 Режим кластера (режим кластера горячего резервирования серверов).

При работе в одиночном режиме, который устанавливается автоматически при установке ПО ViPNet Coordinator Linux, система защиты от сбоев выполняет функции, обеспечивающие постоянную работоспособность основных служб в составе ПО:

- постоянный контроль состояния служб и вывод статистики использования системных ресурсов;
- обнаружение факта сбоя службы и осуществление последующих попыток восстановления работоспособности сбойного приложения;
- предотвращение внутренних сбоев в работе самой системы защиты от сбоев;
- предотвращение сбоев при обработке пакетов драйвером сетевой защиты iplir.

Режим кластера горячего резервирования предназначен для горячей замены функций одного из серверов с ПО ViPNet другим сервером в случае сбоя первого. Кластер горячего резервирования серверов состоит из двух взаимосвязанных компьютеров, один из которых (активный) выполняет функции сервера (координатора) ViPNet, а другой компьютер (пассивный) находится в режиме ожидания. В случае сбоев, критичных для работоспособности ПО ViPNet на активном сервере (в первую очередь в случае сбоев в работе сети или сетевого оборудования), пассивный сервер переключается в активный режим, принимая на себя нагрузку и выполняя функции координатора вместо сервера, который зафиксировал сбой. При работе в режиме кластера горячего резервирования система защиты от сбоев также выполняет функции одиночного режима, то есть обеспечивает постоянную работоспособность основных служб, входящих в состав ПО ViPNet Coordinator Linux.

Далее описываются принципы работы и приводятся сведения об установке и настройке системы защиты от сбоев. При этом предполагается, что читатель знаком с процессом установки и настройки ПО ViPNet Coordinator Linux (см. документ «ViPNet Coordinator Linux. Руководство администратора»).

Принципы работы системы защиты от сбоев

Одиночный режим

В одиночном режиме работы система защиты от сбоев выполняет функции для обеспечения работоспособного состояния основных служб ПО ViPNet Coordinator Linux. Данные функции реализуются совместной работой драйвера watchdog и программы-демона failoverd, работающей в фоновом режиме. Драйвер watchdog работает на низком уровне и в большинстве случаев сохраняет работоспособность даже в случаях, когда система уже не реагирует на внешние события. При соответствующей настройке (см. параметр `reboot` в секции `[misc]` (см. «Секция `[misc]`» на стр. 30)) демон failoverd при запуске регистрируется в драйвере и периодически опрашивает его, подтверждая работоспособность системы. Если по истечении заданного промежутка времени драйвер обнаруживает, что опроса не было, то он перезагружает систему. Перед этим он делает попытку записать на диск кэш-буферы системы, чтобы не возникло ошибок в файловой системе, однако это не всегда возможно. При корректной остановке демона failoverd (например, для изменения настроек системы защиты от сбоев) он сообщает об этом драйверу, и драйвер перестает следить за временем опроса, так что система не будет перезагружена. Такой механизм обеспечивает предотвращение внутренних сбоев в демоне failoverd.

Демон failoverd осуществляет постоянный контроль работоспособности следующих служб ПО ViPNet:

- управляющий демон ViPNet (`iplircfg`);
- транспортный модуль MFTP (`mftpd`);
- демон обработки прикладных протоколов (`algd`);
- сервер веб-интерфейса (`axis2.cgi`).

Демон системы защиты от сбоев failoverd осуществляет старт подконтрольных служб при старте ОС, а также дальнейшее слежение за ними. Контроль работы этих служб осуществляется путем их регистрации в системе защиты от сбоев в момент старта с установкой периода оповещения. В процессе работы контролируемая служба периодически определяет свое состояние и оповещает о нем систему защиты от сбоев. Если контролируемая служба в течение периода оповещения не сообщила о своем состоянии или сообщила о внутреннем сбое, то система защиты от сбоев идентифицирует сбой службы и инициирует процедуру восстановления ее работоспособности. Для этого сначала делается попытка корректной остановки службы. Если эта попытка оказывается неудачной, то осуществляется принудительная «некорректная» остановка службы. После этого система защита от сбоев перезапускает остановленную службу.

В процессе работы демон failoverd ведет статистику сбоев для каждой контролируемой службы, в том числе для самого себя. Если обнаруживается, что для какой-либо из служб произошло 5 сбоев подряд, то есть в течение 5-и попыток восстановления работоспособности служба не смогла

корректно стартовать, то делается вывод о ее полной неработоспособности. В этом случае, в зависимости от настроек системы защиты от сбоев (см. «Секция [misc]» на стр. 30), производится либо перезагрузка ОС, либо остановка сбойной службы и прекращение слежения за ней.

Система защиты от сбоев в ПО ViPNet Coordinator Linux отслеживает также сбои, которые могут произойти в потоках обработки пакетов драйвера сетевой защиты `drviplir` (см. документ «ViPNet Coordinator Linux. Руководство администратора»). Для этого как следящее приложение, так и демон `iplircfg` при старте осуществляют специальный запрос к драйверу `drviplir`. Если в ответ на этот запрос был получен код ошибки, соответствующий сбою одного из потоков обработки пакетов в драйвере, то контролируемое приложение сообщает факт внутреннего сбоя следящему приложению, которое в свою очередь отрабатывает стандартную логику, описанную выше. Помимо старта, управляющий демон `iplircfg` осуществляет периодические запросы к драйверу `drviplir` и в процессе своей работы (запрос информации о журнале пакетов). При этом логика обнаружения сбоев в потоках обработки пакетов такая же, как и на старте контролируемого приложения. Описанный механизм позволяет оперативно (от нескольких десятков секунд до нескольких минут – в зависимости от производительности компьютера и ряда других внешних факторов) отследить факт сбоя в работе драйвера и осуществить корректирующие действия (перезагрузка компьютера в случае включения соответствующей настройки). Однако этот механизм не сможет отследить все возможные сбои в драйвере `drviplir`, например, зависание одного из потоков обработки и так далее. Поэтому его нельзя расценивать как универсальное средство от любого типа сбоев уровня драйверов.

Если контролируемое приложение было корректно остановлено администратором системы с помощью соответствующей команды (`iplir stop`, `mftip stop`, `vipnet-web-gui stop` или `alg stop`), то оно производит deregистрацию в системе защиты от сбоев, слежение за ним отключается. В этом случае для дальнейшей работы администратор должен вручную запустить приложение (соответственно командой `iplir start`, `mftip start`, `vipnet-web-gui start` или `alg start`).

Если при запуске демона `failoverd` выясняется, что какие-либо из подконтрольных демонов были остановлены вручную, то об этом выдается предупреждение в `syslog`. Предупреждение в `syslog` выдается также в случае, если в течение 10-ти проверок подряд одного демона он находится в режиме ручной остановки.

Режим кластера горячего резервирования

Весь кластер, с точки зрения других компьютеров сети, имеет один IP-адрес на каждом из своих сетевых интерфейсов. Этим адресом обладает сервер, находящийся в данный момент в активном режиме. Сервер, находящийся в пассивном режиме, имеет другой IP-адрес, который не используется другими компьютерами для связи с кластером. В отличие от адресов активного режима, в пассивном режиме каждый из серверов имеет свой собственный адрес на каждом из интерфейсов, эти адреса для двух серверов не совпадают.

В случае использования типовой схемы организации кластера горячего резервирования серверов (см. «Типовая схема организации кластера горячего резервирования» на стр. 32) все используемые IP-адреса (один общий адрес для активного режима, два адреса для пассивного режима) должны находиться в одном адресном пространстве (сети). Если возможности по

выделению адресов ограничены, может применяться схема, рассчитанная на выделение только одного реального IP-адреса на интерфейс для активного режима (см. «[Схема организации кластера в условиях ограничений по выделению IP-адресов](#)» на стр. 35).

Стек IP на каждом из серверов настраивается администратором таким образом, чтобы после перезагрузки сервер получал свои адреса пассивного режима. При загрузке запускается демон системы защиты от сбоев failoverd, который стартует в пассивном режиме. В этом режиме пассивный сервер периодически посылает в сеть запросы на поиск IP-адресов активного сервера. Если все адреса активного сервера недоступны в течение заданного времени (следовательно, активного сервера нет в сети), то пассивный сервер переходит в активный режим. При этом он устанавливает на своих интерфейсах соответствующие адреса активного сервера (адреса, под которыми кластер известен другим компьютерам сети) и начинает цикл проверки своих сетевых интерфейсов.

Активный сервер периодически проверяет работоспособность сети на каждом заданном в настройках интерфейсе следующим образом. Периодически, по истечении заданного в настройках временного интервала, анализируется входящий и исходящий сетевой трафик, прошедший через интерфейс. Если разница в количестве пакетов между началом и концом интервала положительна, то считается, что интерфейс функционирует нормально, и счетчик отказов для этого интерфейса сбрасывается. Если в течение данного интервала не было отправлено и принято ни одного пакета, то включается дополнительный механизм проверки, заключающийся в посылке эхо-запросов до стабильных объектов сети. Данный механизм можно использовать не только как дополнительный, но и вместо основного, это настраивается конфигурацией системы защиты от сбоев (см. «[Секция \[channel\]](#)» на стр. 23). Если на какой-либо из интерфейсов в заданное время не приходят ответы на эхо-запросы, счетчик отказов для этого интерфейса увеличивается на единицу. При достижении счетчиком определенного значения фиксируется полный отказ интерфейса. При возникновении полного отказа одного из интерфейсов активный сервер перезагружается. В момент перезагрузки (она занимает, как правило, около 30 секунд) все адреса активного сервера становятся недоступны, что служит сигналом для пассивного сервера на переход в активный режим. После перезагрузки сервер, как описано выше, переходит в пассивный режим, и при работоспособном втором сервере из пары, который теперь работает как активный, остается в этом режиме. Если для проверки работоспособности интерфейсов в настройках заданы открытые объекты сети (см. параметр `testip` в секции `[channel]`), то для пропуска эхо-запросов и ответов на них автоматически создаются соответствующие разрешающие фильтры открытой сети для протокола ICMP.

События, связанные с работой режима кластера горячего резервирования, записываются демоном failoverd в базу данных, которая называется журналом переключений. В журнал переключений попадают следующие типы событий:

- **Загрузка системы** – это событие записывается при старте демона failoverd при загрузке ОС.
- **Старт в пассивном режиме** – это событие записывается при старте демона failoverd вручную в пассивном режиме (при выполнении команд `failover start` или `failover start passive`).
- **Старт в активном режиме** – это событие записывается при старте демона failoverd вручную в активном режиме (при выполнении команд `failover start` или `failover start active`).
- **Переключение серверов** – это событие записывается при переключении сервера из пассивного режима в активный.

Журнал переключений ведется на каждом из серверов кластера. С активного сервера журнал периодически передается на пассивный сервер, заменяя его журнал. При нормальной работе кластера журнал будет содержать только события переключения режима, так как события старта сервера в пассивном режиме будут теряться при получении журнала с активного сервера.

Если сервер стартует в пассивном режиме, а затем, не получив журнал переключений от активного сервера, сам становится активным, то журнал будет содержать события старта в пассивном режиме либо загрузки системы, из чего можно заключить, что второй сервер кластера неработоспособен (завис или отключен).

Журнал переключений можно посмотреть с помощью команды `failover view` (см. «Журнал переключений» на стр. 48).

Чтобы поддерживать конфигурационные файлы и журналы ViPNet на обоих серверах в актуальном состоянии, между серверами создается резервный канал, по которому с активного сервера на пассивный периодически передаются необходимые файлы. Этот канал используется только для передачи файлов с целью резервирования, и его проверка по общей схеме не выполняется. Резервный канал представляет собой соединенные кросс-кабелем адаптеры Ethernet.



Внимание! Адреса резервного канала на серверах кластера не должны совпадать с адресами защищенных узлов, иначе трафик на резервном канале будет блокироваться. Если все же есть необходимость использовать совпадающие адреса, для разрешения конфликта рекомендуется настроить видимость соответствующих защищенных узлов по виртуальным адресам.

Система защиты от сбоев также выполняет резервирование MFTP-конвертов, чтобы обеспечить хранение копий принятых и готовых к отправке конвертов на пассивном сервере. Передача конвертов осуществляется активным сервером по резервному каналу. При переключении пассивного сервера в активный режим сохраненные копии обрабатываются, что практически исключает потерю данных.

Если используется поддержка SNMP, то при запуске на активном сервере управляющего демона на компьютер, который указан в файле конфигурации SNMP как Trap sink, посылается соответствующее событие (Trap). Эта возможность может быть использована администратором для оповещения о сбое одного из серверов.

Оба сервера в используемой схеме абсолютно равноправны. При начальном запуске кластера активным станет тот сервер, который будет запущен раньше. Однако, поскольку переключение сервера из одного режима в другой занимает некоторое время, то при практически одновременном старте серверов может случиться, что они оба перейдут сначала в пассивный режим, а затем в активный. Для предотвращения такой ситуации серверы постоянно обмениваются по резервному каналу пакетами синхронизации, содержащими информацию о режиме работы сервера. Если обнаруживается, что оба сервера находятся в активном режиме, то запускается специальная схема выборов, которая однозначно определяет один из серверов, который должен перезагрузиться и перейти в пассивный режим.

Требования к системе

Система защиты от сбоев предназначена для работы в ОС Linux следующих дистрибутивов и версий:

- Дистрибутивы, поддерживающие архитектуру процессора x86:
 - ALT Linux 6.0 Server;
 - ALT Linux 6.0 Desktop;
 - ALT Linux СПТ 7.0;
 - Astra Linux 1.3 (32/64-разрядная);
 - Astra Linux 1.4;
 - CentOS 5.7 (32/64-разрядная);
 - CentOS 6.4 (32/64-разрядная);
 - Mandriva Linux 2010 Powerpack;
 - RedHat Enterprise Linux 5.7 (32/64-разрядная);
 - RedHat Enterprise Linux 6.4 AS (32/64-разрядная);
 - Slackware Linux 12.0 (только ядро 2.6.16.52 с FTP-сервера [ftp://kernel.org](http://kernel.org));
 - Slackware Linux 12.2;
 - SUSE Linux Enterprise Server 10 SP4 (32/64-разрядная);
 - SUSE Linux Enterprise Server 11 SP3 (32/64-разрядная);
 - Ubuntu 12.04 (32/64-разрядная).
- Дистрибутивы, поддерживающие архитектуру процессора ARM:
 - Debian 7 wheezy;
 - Ubuntu 12.04;
 - Picuntu 12.04.

На других дистрибутивах Linux работа системы защиты от сбоев возможна, но не гарантируется. Также не гарантируется нормальная работа ПО ViPNet Coordinator Linux на одном компьютере совместно со сторонними средствами преобразования трафика, со сторонними средствами защиты, работающими на канальном, сетевом или транспортном уровне (в частности, ipchains и iptables), и со сторонними средствами шифрования трафика (IPSec и так далее).

В системе должно использоваться ядро Linux из подмножества версий 2.6.x (от 2.6.18 до 2.6.39 включительно) и 3.x (до 3.6 включительно). Подробную информацию см. в документе «ViPNet Coordinator Linux. Руководство администратора».

При работе в режиме кластера горячего резервирования компьютеры кластера должны быть настроены так, чтобы корректно перезагружаться без участия человека.

Сетевые интерфейсы, посредством которых кластер общается с внешним миром, должны представлять собой сетевые платы или коммуникационные порты, работающие по протоколу Ethernet, PPP или SLIP и соединенные с локальной сетью или между собой посредством кабелей. Работа с платами, использующими нестандартные средства связи (Radio Ethernet и тому подобные), возможна, но не гарантируется. В качестве резервного канала могут быть использованы платы Ethernet, соединенные между собой кросс-кабелем, или нуль-модемное соединение, работающее по протоколу PPP или SLIP.

Компьютеры кластера должны иметь одинаковое число сетевых интерфейсов, которые должны быть подключены в одни и те же внешние сети. Совпадения имен интерфейсов, подключенных к соответствующим сетям, и аппаратной идентичности сетевых плат не требуется.

Для проверки работоспособности сетевых интерфейсов необходимо наличие отдельного стабильного объекта сети для каждого проверяемого интерфейса. В противном случае проверка будет выполняться не для всех интерфейсов, и кластер окажется неработоспособным.

Сеть должна быть полностью работоспособна до установки ПО ViPNet и системы защиты от сбоев. Если в качестве резервного канала используется нуль-модем, необходимо настроить демон rrr или slip таким образом, чтобы при перезапуске компьютеров связь по нуль-модему устанавливалась автоматически. Как это сделать, можно узнать из документации по ОС Linux.

2

Установка системы защиты от сбоев

Установка системы защиты от сбоев выполняется автоматически в процессе установки ПО ViPNet Coordinator Linux (см. документ «ViPNet Coordinator Linux. Руководство администратора»). По умолчанию система защиты от сбоев устанавливается в одиночном режиме работы.

Для установки системы защиты от сбоев в режиме кластера горячего резервирования необходимо выполнить следующие действия:

- 1 Установите ПО ViPNet Coordinator Linux на обоих серверах, входящих в состав кластера.
- 2 После успешной установки запустите управляющий демон командой `iplir start` на одном из серверов кластера.
- 3 С помощью команды `ps ax |grep iplir` убедитесь, что демон запускается. Если демон не запускается, просмотрите сообщения `syslog` и устраните возникшие неполадки.
- 4 Остановите управляющий демон и другие демоны командой `vipnet stop`.
- 5 Просмотрите файлы `*.conf`, которые будут созданы в подкаталоге `user` каталога `so` справочно-ключевой информацией, и убедитесь, что данные обо всех интерфейсах появились в файлах конфигурации. Особенно это касается случая, когда в качестве резервного канала используется нуль-модем (иногда управляющий демон не может определить наличие интерфейса `ppp` или `sl`). В этом случае вручную укажите данные об интерфейсе в основном файле конфигурации. После старта управляющего демона автоматически будет создан соответствующий конфигурационный файл для этого интерфейса (`iplir.conf-ppp0`, `iplir.conf-sl0` и так далее).
- 6 Настройте сетевые фильтры и туннелирование компьютеров сети и выполните прочие настройки, которые будут использоваться активным сервером кластера.

На этом этапе необходимо выбрать интерфейс, который будет использоваться в качестве резервного канала. Обмен данными по резервному каналу идет открытым трафиком. Именно поэтому при использовании в качестве резервного канала плат Ethernet не следует подключать их в общую сеть, а следует соединить кросс-кабелем.

7 Указанные процедуры проверки и настройки файлов конфигурации необходимо выполнить только на одном сервере кластера, затем скопируйте все файлы конфигурации на другой сервер.

8 После копирования проверьте работоспособность ViPNet Coordinator Linux с файлами конфигурации. При этом проверяйте серверы по одному и не допускайте ситуации, когда управляющий демон работает на обоих серверах кластера одновременно — это может привести к нежелательным последствиям.

9 Настройте работу демона транспортного модуля mftpd. Для этого:

9.1 Остановите демон командой `mftp stop`.

9.2 Выполните команду `mftp check`, которая создаст файл конфигурации `mftp.conf` в подкаталоге `user` каталога со справочно-ключевой информацией, если он еще не был создан.

10 Настройте файл конфигурации системы защиты от сбоев на обоих серверах кластера. Этот файл различен для двух серверов кластера, и его нужно настраивать на каждом сервере отдельно (см. «[Файл конфигурации системы защиты от сбоев](#)» на стр. 22).

11 Выполните команду `failover install`. Эта команда включает режим кластера горячего резервирования серверов. Если впоследствии нужно будет отключить режим кластера горячего резервирования и вернуться к работе в одиночном режиме, выполните команду `failover uninstall`.

Обе команды (включения и выключения режима кластера) не обеспечивают перезапуск всех служб ПО ViPNet Coordinator Linux в соответствии с заданным режимом. Поэтому:

- После выполнения команды `failover install` перезапустите демон `failoverd` командами `failover stop` и `failover start`. В этом случае система защиты от сбоев запустится в пассивном режиме горячего резервирования серверов. Для запуска в активном режиме выполните команду `failover start active`.
- Если была выполнена команда `failover uninstall`, то система выдаст предупреждение о том, что все службы ViPNet будут остановлены, и запросит подтверждение у администратора. При положительном решении система защиты от сбоев переключается в одиночный режим работы, предварительно остановив все службы ViPNet. Для дальнейшей работы в этом режиме выполните команду `failover start`, в результате выполнения которой будет запущен демон `failoverd` и все остальные службы ViPNet.

3

Обновление версии ПО ViPNet Coordinator Linux

В процессе эксплуатации ПО ViPNet периодически возникает необходимость обновления версии ПО на более новую. Процесс обновления ПО ViPNet Coordinator Linux описан в документе «ViPNet Coordinator Linux. Руководство администратора». Однако при работе в режиме кластера горячего резервирования серверов процесс обновления ПО имеет ряд особенностей, связанных с обеспечением бесперебойной работы кластера в процессе обновления. Ниже описан алгоритм обновления ПО ViPNet Coordinator Linux на кластере, обеспечивающий бесперебойную работу системы.

- Перед началом обновления необходимо временно отключить коммутацию серверов по резервному каналу.
- Обновление необходимо начинать с сервера, находящегося в пассивном режиме резервирования. Информацию о текущем режиме можно получить с помощью команды `failover info` (см. «[Текущее состояние](#)» на стр. 46). Обновление необходимо провести по алгоритму, описанному в соответствующем разделе документа «ViPNet Coordinator Linux. Руководство администратора».
- После обновления ПО на пассивном сервере кластера необходимо убедиться в стабильной работе данной версии ПО на данном сервере в течение выбранного интервала времени. В случае каких-либо проблем в работе новой версии ПО следует произвести возврат к версии, установленной ранее. Процедура возврата к предыдущей версии описана в соответствующем разделе документа «ViPNet Coordinator Linux. Руководство администратора».

Работа сервера считается стабильной, если:

- Все службы ViPNet запущены и работают корректно, система защиты от сбоев работает в режиме кластера. Информацию о текущем состоянии служб и режиме работы системы защиты от сбоев можно получить с помощью команды `failover info` (см. «[Текущее состояние](#)» на стр. 46).

- Все процессы-демоны работают в пассивном режиме (passive). Для проверки текущего режима работы процессов используется команда операционной системы `ps aux` (подробнее см. документ «ViPNet Coordinator Linux. Руководство администратора»).
- В течение выбранного интервала времени (около 15 минут) сервер не перезагружается.
- В случае стабильной работы новой версии ПО в пассивном режиме необходимо перевести данный сервер в активный режим. Для этого следует выключить сервер, находящийся в данный момент в активном режиме, командой `halt` или `shutdown`. Через тайм-аут, заданный в настройках системы защиты от сбоев (см. «Секция [misc]» на стр. 30), пассивный сервер с новой версией ПО должен перейти в активный режим. Если в результате каких-либо сбоев в работе новой версии ПО этого не произошло или после перехода в активный режим новая версия ПО функционирует некорректно, следует включить сервер со старой версией ПО, а на сервере с новой версией произвести процедуру возврата к предыдущей версии.
- В случае стабильной работы новой версии ПО в активном режиме в течение выбранного периода времени необходимо включить сервер, на котором установлена старая версия ПО, а затем произвести на нем процедуру обновления ПО по алгоритму, описанному в соответствующем разделе документа «ViPNet Coordinator Linux. Руководство администратора».
- После обновления ПО на обоих серверах кластера необходимо включить коммутацию серверов по резервному каналу и убедиться, что резервирование работает. Для этого нужно изменить какую-либо настройку в одном из файлов конфигурации на активном сервере и через некоторое время проверить, что эти же изменения появились на пассивном сервере. Чтобы изменения попали на пассивный сервер, необходимо включить резервирование группы файлов конфигурации (см. «Секция [sendconfig]» на стр. 27).



Примечание. При обновлении ПО ViPNet Coordinator Linux с версии 3.7 до версии 4.1 и выше журнал переключений кластера горячего резервирования будет перемещен в архив. После обновления в новом файле журнала переключений кластера содержится только информация с момента обновления. Если вы хотите просмотреть прежний журнал переключений, обратитесь в службу поддержки ОАО «ИнфоТеКС».

4

Настройка системы защиты от сбоев

Файл конфигурации системы защиты от сбоев	22
Секция [channel]	23
Секция [network]	25
Секция [sendconfig]	27
Секция [misc]	30
Секция [debug]	31
Типовая схема организации кластера горячего резервирования	32
Схема организации кластера в условиях ограничений по выделению IP-адресов	35
Ограничения при использовании дополнительных адресов на интерфейсах кластера	39

Файл конфигурации системы защиты от сбоев

Файл конфигурации системы защиты от сбоев находится в каталоге `/etc` и называется `failover.ini`. Он состоит из нескольких секций. Каждая секция начинается со строки, содержащей имя секции в квадратных скобках. Каждая секция содержит несколько параметров. Строка с параметром начинается с имени параметра, затем идет знак «=» и пробел, затем значение этого параметра. Имена секций и параметров могут повторяться.

Параметры настройки системы защиты от сбоев можно разделить на две группы:

- параметры, отвечающие за настройку системы в режиме кластера горячего резервирования серверов;
- общие параметры, характерные для обоих режимов работы (как для одиночного режима, так и для режима кластера).

Секция [channel]

Каждый сетевой интерфейс, работоспособность которого должна проверять система защиты от сбоев при работе в активном режиме кластера горячего резервирования, описывается секцией [channel].

Секция [channel] содержит следующие параметры:

- `device` — имя сетевого интерфейса (eth0, eth1 и так далее), который описывает эта секция.
- `activeip` — IP-адрес, который данный интерфейс будет иметь в активном режиме.

В качестве необязательного значения в параметре может указываться маска подсети в нотации CIDR (число значащих бит) или в обычной прямой нотации. Значение IP-адреса должно отделяться от значения маски символом «/» (косая черта). Например:

`activeip= 192.168.201.1/24` — маска задана в CIDR-нотации.

`activeip= 68.21.12.34/255.255.252.0` — маска задана в прямой нотации.

Если маска не указана, то будет использовано значение маски, установленное в системе. Независимо от того, в какой нотации была задана маска сети, после перезаписи файла конфигурации `failover.ini` в процессе старта демона `failoverd` маска будет переведена в CIDR-нотацию и сохранена в файле в таком виде.



Примечание. Явное указание значения маски сети используется при организации схемы кластера горячего резервирования в условиях ограничений по выделению IP-адресов (см. «[Схема организации кластера в условиях ограничений по выделению IP-адресов](#)» на стр. 35).

- `passiveip` — IP-адрес, который данный интерфейс будет иметь в пассивном режиме.

В качестве необязательного значения в параметре может указываться маска сети. Требования к формату задания маски аналогичны параметру `activeip`.

- `testip` — IP-адрес маршрутизатора или другого стабильного объекта сети, которому будут посылаться эхо-запросы для проверки работоспособности этого интерфейса. Можно указывать несколько параметров `testip`, в этом случае эхо-запросы будут посылаться на все указанные адреса, и сбоем интерфейса будет считаться ситуация, когда ни от одного из адресов не получен ответ.



Внимание! Для каждого интерфейса, описанного секцией [channel], в параметре `testip` должен быть задан свой адрес, принадлежащий подсети данного интерфейса. Если указать один и тот же адрес для нескольких интерфейсов, система защиты от сбоев будет проверять работоспособность только одного из этих интерфейсов.

- `ident` — текстовая строка, идентифицирующая данный интерфейс.

- `checkonlyidle` — указывает, нужно ли проверять только неактивные интерфейсы. Может принимать значение `yes` или `no`. Если параметр установлен в `yes`, то активный сервер посылает эхо-запросы до адресов, указанных в параметрах `testip`, только в том случае, если за период опроса IP-адресов (параметр `checktime` в секции `[network]`) на данном интерфейсе не было входящих или исходящих пакетов. Если параметр установлен в `no`, то эхо-запросы посылаются всегда. По умолчанию значение параметра — `yes`.



Примечание. Рекомендуется использовать режим `checkonlyidle=yes`, так как в режиме `checkonlyidle=no` кластер может ошибочно идентифицировать сбой интерфейса при отсутствии связи с тестовым узлом (по внешним причинам, не связанным с работой кластера).

- `afterifconf` — параметр, содержащий команды, выполняемые непосредственно после конфигурирования данного интерфейса при смене режима. Является необязательным параметром и по умолчанию отсутствует в файле конфигурации. Данный параметр может использоваться в специфических ситуациях, например, если необходимо выполнить какие-либо нестандартные команды, связанные с настройкой сетевого интерфейса.

Таким образом, секции `[channel]` необходимо создать для каждого интерфейса, работа которого будет проверяться.



Примечание. Все параметры секций `[channel]` интерпретируются только при работе в режиме кластера горячего резервирования серверов.

Чтобы отключить слежение за работоспособностью какого-либо интерфейса, необходимо удалить из файла конфигурации секцию `[channel]`, описывающую этот интерфейс.

Секция [network]

Секция [network] описывает различные параметры работы системы защиты от сбоев, относящиеся к отправке пакетов в сеть в режиме кластера горячего резервирования.

Секция [network] содержит следующие параметры:

- `checktime` — период опроса IP-адресов (в секундах). На активном сервере проверка работоспособности интерфейса будет проводиться с интервалом `checktime`. На пассивном сервере с интервалом `checktime` будут отправляться запросы на поиск IP-адресов активного сервера. Значение по умолчанию — 10 (секунд).
- `timeout` — время ожидания (в секундах) ответа на запрос (эхо-запрос или запрос IP-адресов), по истечении которого делается вывод о том, что результат запроса отрицательный. Значение по умолчанию — 2 (секунды).
- `channelretries` — число полученных подряд отрицательных результатов, на основании которых делается вывод о неработоспособности интерфейса на активном сервере. Значение по умолчанию — 3 (секунды).
- `activeretries` — число полученных подряд отрицательных результатов, на основании которых на пассивном сервере делается вывод об отсутствии в сети данного IP-адреса активного сервера. Значение по умолчанию — 3 (секунды).
- `synctime` — период времени (в секундах) между отсылками пакетов синхронизации по резервному каналу. Значение по умолчанию — 5 (секунд).
- `fastdown` — указывает, нужно ли принудительно останавливать сетевые интерфейсы перед перезагрузкой сервера. Может принимать значение `yes` или `no`. Установка этого параметра в `yes` позволяет быстрее устранить присутствие сервера в сети и дать возможность второму серверу переключиться в активный режим, однако при этом завершение работы работающих сетевых сервисов происходит уже при отключенных интерфейсах и может быть некорректным. Значение по умолчанию — `yes`. Необходимо выбирать значение этого параметра, исходя из того, какие сервисы работают на компьютерах кластера.
- `afterifconf` — параметр, содержащий команды, выполняемые непосредственно после конфигурирования всех интерфейсов при смене режима. Является необязательным параметром и по умолчанию отсутствует в файле конфигурации.
- `beforeifconf` — параметр, содержащий команды, выполняемые перед конфигурированием всех интерфейсов при смене режима. Является необязательным параметром и по умолчанию отсутствует в файле конфигурации.

Последние два параметра могут использоваться в специфических ситуациях, например, если необходимо выполнить какие-либо нестандартные команды, связанные с настройкой сетевых интерфейсов. Данные параметры используются для организации схемы кластера горячего резервирования в условиях ограничений по выделению IP-адресов (см. «[Типовая схема организации кластера горячего резервирования](#)» на стр. 32).



Примечание. Все параметры секции `[network]` интерпретируются только при работе в режиме кластера горячего резервирования серверов. Их рекомендуется делать одинаковыми на обоих серверах кластера.

Секция [sendconfig]

В секции [sendconfig] задаются параметры, которые контролируют пересылку файлов с активного сервера на пассивный с целью резервирования.

Секция [sendconfig] содержит следующие параметры:

- `activeip` — адрес, который имеет на резервном канале второй сервер кластера, находящийся в противоположном режиме. Для каждого сервера в этом параметре должен быть задан адрес резервного канала другого сервера.
- `sendtime` — период резервирования (в секундах), то есть период между попытками переслать файлы. Значение по умолчанию — 60 (секунд).
- `config` — включение или отключение резервирования группы файлов конфигурации.
- `keys` — включение или отключение резервирования группы файлов справочников и ключей.
- `journals` — включение или отключение резервирования группы файлов журналов ПО ViPNet.
- `file` — произвольный файл для резервирования.
- `device` — системное имя интерфейса, который используется для организации резервного канала.
- `port` — номер порта, на котором данный сервер в активном режиме ожидает соединения на резервном канале от пассивного сервера кластера для передачи ему заданных файлов. Значение по умолчанию 10090.
- `connectport` — номер порта, который данный сервер в пассивном режиме выбирает для соединения на резервном канале с активным сервером кластера для приема запрошенных файлов. Данный параметр может отсутствовать в конфигурационном файле. В этом случае его значение по умолчанию равно значению параметра `port`. Если параметр `port` также не указан, то значение параметра `connectport` равно 10090.



Примечание. Все параметры секции [sendconfig] интерпретируются только при работе в режиме кластера горячего резервирования серверов.

Параметры `config`, `keys` и `journals` могут принимать значение `yes` или `no`. Значение `no` означает отключение резервирования соответствующей группы. По умолчанию эти параметры установлены в значение `yes`.



Внимание! Не рекомендуется отключать резервирование групп `config` и `keys`, так как это может привести к некорректной работе ПО ViPNet.

В группу `config` входят следующие файлы конфигурации:

- `iplir.conf` — основной файл конфигурации управляющего демона `iplircfg`;

- `iplir.conf`-<имя интерфейса> – файлы конфигурации сетевых интерфейсов (кроме интерфейса резервного канала);
- `mftp.conf` – файл конфигурации транспортного модуля MFTP;
- ряд других служебных файлов конфигурации (список этих файлов не приводится, так как они являются вспомогательными и могут отсутствовать в ряде конфигураций ПО ViPNet), а также сетевые фильтры и политики безопасности.

В группу `keys` входят служебные файлы, относящиеся к справочникам и ключам. Список файлов может динамически меняться в процессе работы, что отслеживается демоном `failoverd` автоматически.

В группу `journals` входят следующие файлы:

- журналы пакетов сетевых интерфейсов (кроме интерфейса резервного канала);
- журнал конвертов транспортного модуля MFTP;
- ряд других служебных файлов журналов (список этих файлов не приводится, так как они являются вспомогательными и могут отсутствовать в ряде конфигураций ПО ViPNet).

Перечень файлов, входящих в группы `config`, `keys` и `journals`, определяется демоном `failoverd` автоматически на активном сервере. Пассивный сервер на каждом цикле резервирования запрашивает состав файлов, входящих в каждую из групп, для которых включено резервирование, и после этого запрашивает передачу файлов.



Внимание! Резервирование файлов, входящих в группы `config` и `keys`, производится только при запущенном на активном сервере управляющем демоне `iplircfg`. Резервирование файла `mftp.conf` производится только при запущенном на активном сервере демоне `mftpd`. Указанные ограничения позволяют предотвратить передачу на пассивный сервер неправильно отредактированных файлов конфигурации.

Параметры `file` описывают резервирование файлов, не являющихся файлами конфигурации или другими служебными файлами ПО ViPNet, то есть параметры `file` могут содержать только сторонние файлы, не входящие в вышеперечисленные группы, если резервирование этих групп включено.



Примечание. Если какой-либо из файлов, заданных параметром `file`, входит в одну из перечисленных выше групп, для которой включено резервирование, демон `failoverd` удаляет его из файла конфигурации `failover.ini` с выводом соответствующего сообщения в `syslog`.

Можно задавать любое количество параметров `file`, однако следует иметь в виду, что используемый протокол передачи файлов оптимизирован для передачи коротких файлов — как правило, файлов конфигурации, и передача через систему резервирования больших файлов не рекомендуется (максимальный рекомендуемый размер составляет примерно 1 Мбайт). Кроме того, размеры файлов должны быть согласованы с параметром `sendtime` таким образом, чтобы указанного в этом параметре времени хватило на пересылку файлов.

Если имя файла начинается с символа «/», то оно трактуется как абсолютное, если с другого символа, то оно воспринимается как имя относительно каталога, содержащего справочники и ключи.

Как и в случае групп, пересылка файлов, заданных параметрами `file`, производится по запросу пассивной стороны. Однако, в отличие от передачи групп, активный сервер не формирует никаких списков, передача происходит по запросу на каждый файл, то есть в этом случае список файлов в виде параметров `file` определяется настройками пассивного сервера.

Возможные конфликты при резервировании файлов

При передаче файлов `iplir.conf` (основного файла конфигурации) и `iplir.conf-<интерфейс>` (файлов конфигурации для интерфейсов) может возникнуть конфликтная ситуация, вызванная использованием на серверах кластера интерфейсов с разными именами для подключения к одним и тем же сетям. Это связано с заменой имен интерфейсов на идентификаторы и обратно. При передаче файлов с именами `iplir.conf-<интерфейс>` на активном сервере происходит замена имен интерфейсов на идентификаторы, указанные в секциях `[channel]` (см. «Секция [\[channel\]](#)» на стр. 23). При приеме этих файлов на пассивном сервере идентификаторы заменяются именами интерфейсов, для которых те же идентификаторы указаны в секциях `[channel]` на пассивном сервере (см. пример ниже). Файлы конфигурации для интерфейсов, которые не описаны секциями `[channel]`, передаются без изменения имен. При этом может получиться так, что после формирования на пассивном сервере списка файлов, которые следует принять, файлы для разных интерфейсов имеют одинаковые имена.

Например, пусть на активном сервере интерфейс `eth0` описан секцией `[channel]` и ему присвоен идентификатор «А» (`ident= A`), а также имеется интерфейс `eth2`, который не описан секцией `[channel]`. При этом на пассивном сервере с тем же идентификатором «А» описан интерфейс `eth2` (то есть предполагается, что он используется для подключения к той же сети, что и интерфейс `eth0` на активном сервере). В сформированный на активном сервере список файлов входят файлы `iplir.conf-eth0` и `iplir.conf-eth2`, которые после замены имен будут называться соответственно `iplir.conf-A` и `iplir.conf-eth2`. При формировании списка файлов на пассивном сервере файл `iplir.conf-eth2` останется с тем же именем, а файл `iplir.conf-A` будет переименован в `iplir.conf-eth2`, т.е. в списке принимаемых файлов окажутся два файла с одним и тем же именем. В этом случае система резервирования запишет в `syslog` сообщение об ошибке в конфигурации интерфейсов с указанием имени и идентификатора конфликтующего интерфейса. При обнаружении подобного конфликта резервирование файлов группы `config` не производится.

Секция [misc]

Секция [misc] содержит вспомогательные параметры как для режима кластера горячего резервирования серверов, так и для одиночного режима работы системы защиты от сбоев:

- `activeconfig` — файл конфигурации управляющего демона, который будет использоваться в активном режиме горячего резервирования.
- `maxjournal` — максимальное количество дней, за которое необходимо хранить записи в журнале переключений кластера горячего резервирования (см. «[Режим кластера горячего резервирования](#)» на стр. 12). Является необязательным параметром. По умолчанию значение параметра — 30.
- `passiveconfig` — файл конфигурации управляющего демона, который будет использоваться в пассивном режиме горячего резервирования.
- `reboot` — указывает, должен ли демон `failoverd` включать механизм регистрации в драйвере `watchdog` и должна ли производиться перезагрузка ОС в случае, если какая-либо из контролируемых служб не может восстановить свою работоспособность (см. «[Одиночный режим](#)» на стр. 11). Может принимать значение `yes` или `no`. Значение `yes` включает механизм регистрации демона `failoverd` и перезагрузки системы, значение `no` выключает его. Параметр является обязательным и интерпретируется независимо от режима работы системы защиты от сбоев. По умолчанию значение параметра — `yes`.

Примечание. Параметры `activeconfig`, `passiveconfig` и `maxjournal` интерпретируются только при работе в режиме кластера горячего резервирования.



Архитектура системы защиты от сбоев подразумевает использование одной и той же конфигурации ViPNet в активном и пассивном режимах, другие возможности не поддерживаются. Поэтому в параметрах `activeconfig` и `passiveconfig` нужно указывать один и тот же файл — `/etc/iplirpsw`.

Секция [debug]

Секция [debug] определяет параметры ведения журнала устранения неполадок демона failoverd (см. документ «ViPNet Coordinator Linux. Руководство администратора»). Она содержит следующие параметры:

- `debuglevel` – уровень отладки, число от -1 до 5, по умолчанию — 3. Значение параметра -1 отключает ведение журнала.
- `debuglogfile` – идентификатор, определяющий место хранения журнала. Формат данного идентификатора следующий: <спецификатор протокола>:<спецификатор URL для данного протокола>. Подробное описание возможных значений данного параметра приведено в документе «ViPNet Coordinator Linux. Руководство администратора». Значение параметра по умолчанию: `file:/var/log/failover.debug.log`, что соответствует записи журнала в указанный файл.

Типовая схема организации кластера горячего резервирования

Пример типовой схемы организации кластера горячего резервирования приведен ниже.

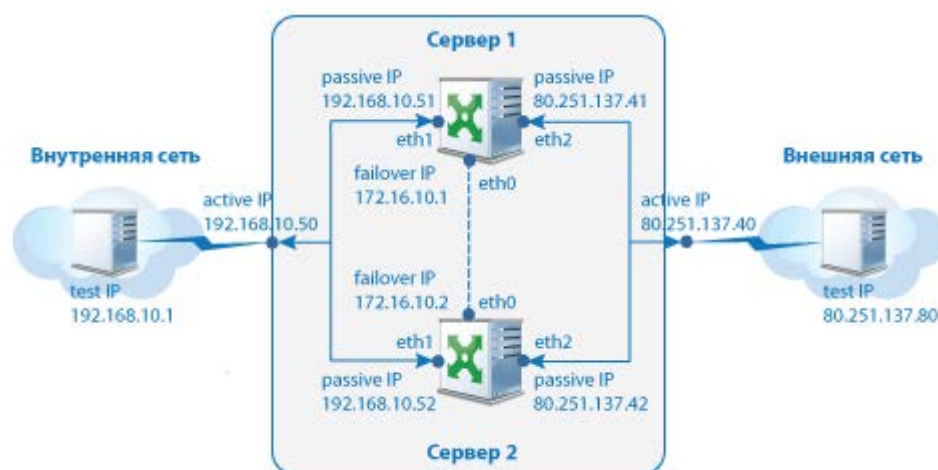


Рисунок 1. Типовая схема организации кластера горячего резервирования

На схеме интерфейсы eth0, соединенные кросс-кабелем, используются для организации резервного канала.

В случае использования типовой схемы необходимо соблюдать следующие требования:

- При настройке секций `[channel]` IP-адреса, указанные в параметрах `activeip` и `passiveip` для внутреннего (на схеме eth1) и внешнего (на схеме eth2) интерфейсов, должны обязательно находиться в одной подсети. При этом маску подсети указывать необязательно.
- Значения параметров `device`, `activeip` и `testip` должны быть одинаковыми на обоих серверах кластера, а параметры `passiveip` должны быть разными. Таким образом, в типовой схеме в каждой из сетей, к которым подключены контролируемые интерфейсы кластера, должны быть выделены три IP-адреса: один для `activeip` и два для `passiveip`. IP-адрес, указанный в параметре `passiveip`, должен совпадать с адресом, который установлен для данного интерфейса по умолчанию (настраивается в Linux).
- Для интерфейсов, подключенных к одинаковым сетям, параметры `ident` должны совпадать на обоих серверах кластера — именно по этим параметрам система защиты от сбоев определяет интерфейсы, которые выполняют одинаковые функции на серверах кластера.

Таблица ниже содержит настройки параметров системы защиты от сбоев, соответствующие приведенной выше типовой схеме.

Таблица 3. Настройки параметров системы защиты от сбоев для типовой схемы кластера

Настройки на первом сервере	Настройки на втором сервере
[channel] device= eth1 activeip= 192.168.10.50 passiveip= 192.168.10.51 testip= 192.168.10.1 ident= if-1 checkonlyidle= yes	[channel] device= eth1 activeip= 192.168.10.50 passiveip= 192.168.10.52 testip= 192.168.10.1 ident= if-1 checkonlyidle= yes
[channel] device= eth2 activeip= 80.251.137.40 passiveip= 80.251.137.41 testip= 80.251.137.80 ident= if-2 checkonlyidle= yes	[channel] device= eth2 activeip= 80.251.137.40 passiveip= 80.251.137.42 testip= 80.251.137.80 ident= if-2 checkonlyidle= yes
[network] checktime= 10 timeout= 2 activeretries= 3 channelretries= 3 synctime= 5 fastdown= yes	[network] checktime= 10 timeout= 2 activeretries= 3 channelretries= 3 synctime= 5 fastdown= yes
[sendconfig] device= eth0 activeip= 172.16.10.2 (соответствует failover IP второго сервера)	[sendconfig] device= eth0 activeip= 172.16.10.1 (соответствует failover IP первого сервера)

Алгоритм работы на активном сервере следующий. Через каждые `checktime` секунд проводится проверка работоспособности каждого из приведенных в конфигурации интерфейсов. Если параметр `checkonlyidle` установлен в `yes`, то анализируется входящий и исходящий сетевой трафик, прошедший через интерфейс. Если разница в количестве пакетов между началом и концом интервала положительна, то считается, что интерфейс функционирует нормально, и счетчик отказов для этого интерфейса обнуляется. Если в течение данного интервала не было

послано и принято ни одного пакета, то включается дополнительный механизм проверки, заключающийся в посылке эхо-запросов до ближайших маршрутизаторов. Если параметр `checkonlyidle` установлен в `no`, то механизм дополнительной проверки используется вместо основного, то есть каждые `checktime` секунд посылаются пакеты до адресов `testip`. Затем в течение времени `timeout` ожидаются ответы. Если на каком-либо интерфейсе ответа нет ни от одного адреса `testip`, то его счетчик сбоев увеличивается на единицу. Если хотя бы на одном интерфейсе счетчик сбоев не равен нулю, то немедленно посылаются новые пакеты до всех `testip` и ожидается ответ в течение `timeout`. Если в процессе новых посылок на интерфейс, счетчик сбоев которого не равен нулю, приходит ответ, его счетчик сбоев обнуляется. Если после какой-либо посылки счетчики сбоев на всех интерфейсах становятся равны нулю, то происходит возврат в основной цикл, новое ожидание в течение `checktime` и так далее. Если же после какого-то числа новых посылок счетчик сбоев хотя бы одного интерфейса достигнет значения `channelretries`, то фиксируется полный отказ интерфейса и начинается перезагрузка системы.

Таким образом, максимальное время неработоспособности интерфейса до того, как система защиты от сбоев сделает вывод об этом, равно `checktime + (timeout * channelretries)`.

На пассивном сервере алгоритм немного отличается. Раз в `checktime` секунд производится удаление записей в системной ARP-таблице для всех `activeip`. Затем посылаются UDP-запросы со всех интерфейсов на адреса `activeip`, в результате чего система сначала посылает ARP-запрос и только в случае получения ответа посылает UDP-запрос. После окончания интервала ожидания ответа `timeout` проверяется наличие ARP-записи для каждого `activeip` в системной ARP-таблице, по наличию которой делается вывод о работоспособности соответствующего интерфейса на активном сервере. Если ни от одного интерфейса не был получен ответ, счетчик сбоев (он один на все интерфейсы) увеличивается. Если хотя бы от одного интерфейса ответ был получен, счетчик сбоев обнуляется. Если счетчик сбоев достигает значения `activeretries`, то производится переключение в активный режим. Максимальное время, проходящее от перезагрузки активного сервера до обнаружения пассивным сервером этого факта, равно `checktime + (timeout * activeretries)`.

Общее время неработоспособности системы при сбое может быть немного больше, чем `checktime * 2 + timeout * (channelretries + activeretries)`. Это связано с тем, что после начала перезагрузки сбойного сервера система переводит его интерфейсы в нерабочее состояние не сразу, а через некоторое время, после остановки других подсистем. Поэтому, например, если проверяются два интерфейса и только на одном произошел сбой, то адрес второго интерфейса будет доступен еще некоторое время, в течение которого пассивный сервер будет получать от него ответы. Обычно время от начала перезагрузки до выключения интерфейсов не превышает 30 секунд, однако оно может сильно зависеть от быстродействия компьютера и количества работающих на нем сервисов.

Схема организации кластера в условиях ограничений по выделению IP-адресов

В некоторых случаях, когда выделение трех IP-адресов в одной сети для организации типовой схемы не представляется возможным (использование публичных IP-адресов, ограничения адресного пространства сети и так далее), можно использовать схему организации кластера, в которой требуется выделить лишь один IP-адрес для активного режима работы кластера. Пример такой схемы приведен ниже.

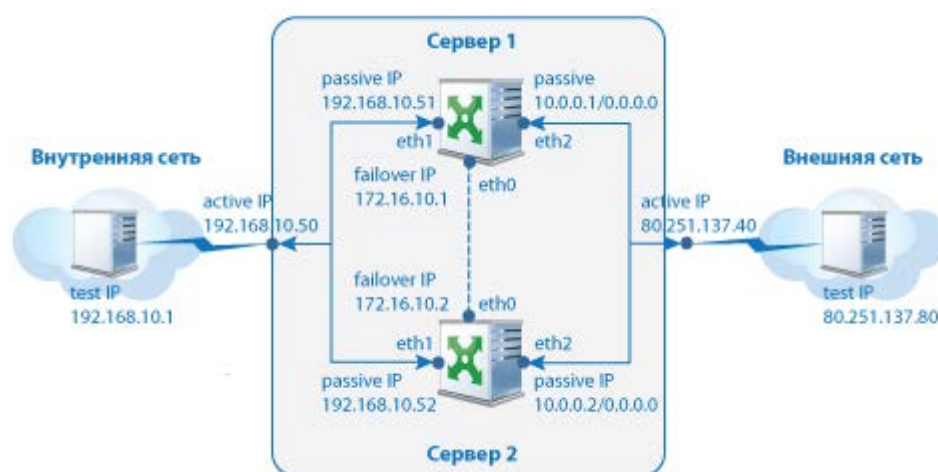


Рисунок 2. Схема организации кластера горячего резервирования в условиях ограничений по выделению IP-адресов

Из схемы видно, что, в отличие от типовой схемы, для внешних интерфейсов (на схеме eth2) выделен только один публичный IP-адрес (для активного режима) вместо трех публичных адресов в случае использования типовой схемы. Пассивные адреса интерфейсов выбраны из диапазона частной сети. Подключение к внутренней сети выполнено по типовой схеме.

Общий принцип работы состоит в том, чтобы использовать на пассивном сервере адреса из другой подсети — например, из диапазона частных адресов. Чтобы пассивный сервер мог проверить наличие в сети адресов активного, на пассивном сервере устанавливается на соответствующих интерфейсах маска подсети 0.0.0.0 и широковещательный адрес 255.255.255.255.



Примечание. Если на интерфейсе, для которого вы хотите установить маску 0.0.0.0, уже был задан IP-адрес, нужно предварительно очистить настройки адреса интерфейса с помощью команды:

```
hostname# ifconfig <имя интерфейса> reset,
```

Затем вы можете задать нужные IP-адрес и маску с помощью команды:

```
hostname# ifconfig <имя интерфейса> address <IP-адрес> netmask 0.0.0.0
```

При такой конфигурации интерфейса пассивный сервер пытается отправить пакет, проходящий через этот интерфейс, напрямую, запросив в сети MAC-адрес получателя. Таким образом, если пассивный сервер попытается отправить пакет активному серверу и маршрутизация на адрес активного будет настроена через интерфейс с маской 0.0.0.0, то пассивный сервер всегда запросит MAC-адрес активного путем посылки ARP-запроса — независимо от того, к каким подсетям принадлежат адреса активного и пассивного серверов, и затем отправит пакет на этот MAC-адрес. Такой механизм позволяет пассивному серверу контролировать работоспособность интерфейсов активного сервера по алгоритму, описанному выше, то есть путем проверки наличия ответов на ARP-запросы. При этом важно, чтобы маршрутизация через интерфейсы с маской подсети 0.0.0.0 была настроена только на адреса активного сервера, принадлежащие соответствующим интерфейсам, чтобы такая конфигурация не нарушала работу других интерфейсов.

Примечание. Важным условием при настройке схемы, использующей только один публичный IP-адрес, является явное задание масок подсети для параметров `activeip` и `passiveip` в файле конфигурации `failover.ini`. Причем для `activeip` необходимо использовать реальную маску подсети, а для `passiveip` — нулевую маску.



При задании статических маршрутов и шлюза по умолчанию для пассивного сервера в условиях ограничений по IP-адресам эти настройки будут сохранены в системе, но не будут применены немедленно, о чем выдается соответствующее предупреждение. Данные настройки будут применены только при переходе сервера в активный режим.

Для настройки правильной маршрутизации при использовании описанной схемы нужно использовать специальный сценарий (shell-скрипт), который будет задавать правильную маршрутизацию в системе как в активном, так и в пассивном режиме. Такой сценарий входит в комплект поставки ViPNet Coordinator Linux и называется `change_route.sh`. При установке ПО этот скрипт устанавливается в каталог `/sbin`. Для настройки описываемой схемы, помимо задания IP-адресов для пассивного режима, необходимо в файле конфигурации `failover.ini` прописать вызов данного скрипта для следующих параметров:

- В секции `[network]` в качестве значения параметра `afterifconf`:

```
afterifconf= /sbin/change_route.sh after
```

В этом случае скрипт задания маршрутов будет вызван демоном `failoverd` после конфигурирования сетевых интерфейсов. Для пассивного режима данный скрипт будет устанавливать в системе маршруты на адреса активного сервера.

- В секции `[network]` в качестве значения параметра `beforeifconf`:

```
beforeifconf= /sbin/change_route.sh before
```

В этом случае скрипт задания маршрутов будет вызван демоном `failoverd` до конфигурирования сетевых интерфейсов. Для активного режима данный скрипт будет удалять в системе маршруты, которые были установлены в пассивном режиме при вызове `/sbin/change_route.sh after`.

Передача скрипту необходимой служебной информации производится через набор переменных окружения.

Таблица 4. астройки параметров системы защиты от сбоев в условиях ограничений по выделению IP-адресов

Настройки на первом сервере	Настройки на втором сервере
[channel]	[channel]
device= eth1	device= eth1
activeip= 192.168.10.50	activeip= 192.168.10.50
passiveip= 192.168.10.51	passiveip= 192.168.10.52
testip= 192.168.10.1	testip= 192.168.10.1
ident= if-1	ident= if-1
checkonlyidle= yes	checkonlyidle= yes
 [channel]	 [channel]
device= eth2	device= eth2
activeip= 80.251.137.40/24	activeip= 80.251.137.40/24
passiveip= 10.0.0.1/0.0.0.0	passiveip= 10.0.0.2/0.0.0.0
testip= 80.251.137.80	testip= 80.251.137.80
ident= if-2	ident= if-2
checkonlyidle= yes	checkonlyidle= yes
 [network]	 [network]
checktime= 10	checktime= 10
timeout= 2	timeout= 2
activeretries= 3	activeretries= 3
channelretries= 3	channelretries= 3
synctime= 5	synctime= 5
fastdown= yes	fastdown= yes
afterifconf= /sbin/change_route.sh after	afterifconf= /sbin/change_route.sh after
beforeifconf= /sbin/change_route.sh before	beforeifconf= /sbin/change_route.sh before
 [sendconfig]	 [sendconfig]

Настройки на первом сервере	Настройки на втором сервере
device= eth0	device= eth0
activeip= 172.16.10.2 (соответствует failover IP второго сервера)	activeip= 172.16.10.1 (соответствует failover IP первого сервера)

Ограничения при использовании дополнительных адресов на интерфейсах кластера

При использовании дополнительных IP-адресов на интерфейсах кластера существует ряд ограничений, при несоблюдении которых корректная работа кластера не гарантируется. Это связано с тем, что для управляющего демона и драйвера все псевдоустройства, соответствующие дополнительным адресам, представляют одно физическое устройство. Как следствие, файлы конфигурации псевдоустройств не создаются, и с активного сервера на пассивный передаются файлы конфигурации только физических интерфейсов.

При использовании дополнительных адресов на интерфейсах кластера следует соблюдать следующие ограничения:

- Имена псевдоустройств должны быть сформированы по шаблону `<имя сетевого адаптера>:<номер>`, например, `eth0:1`.
- В файлах `failover.ini` на обоих серверах кластера обязательно должна присутствовать секция `[channel]` с описанием физического интерфейса, описание только псевдоустройств недопустимо.
- На обоих серверах кластера должно быть одинаковое число дополнительных IP-адресов для соответствующих интерфейсов.
- Для псевдоустройств, подключенных к одной сети, должны совпадать параметры `ident` и номера в именах псевдоустройств. При этом имена сетевых адаптеров могут меняться одновременно с именами соответствующих физических устройств. В случае изменения имени адаптера необходимо внести соответствующие изменения в файлы `failover.ini` на обоих серверах кластера, заменив имя как самого физического интерфейса, так и всех ссылающихся на него псевдоустройств.
- Параметр `checkonlyidle` должен быть установлен в значение `no` для всех псевдоустройств и для самого физического интерфейса.

На схеме ниже представлен пример организации кластера при использовании дополнительных IP-адресов на одном из интерфейсов. Эта схема основана на типовой схеме (см. «[Типовая схема организации кластера горячего резервирования](#)» на стр. 32), модифицированной следующим образом:

- сетевые адаптеры серверов кластера с одинаковым именем (кроме `eth0`) выполняют разные функции (адаптеры «перекрещены»);
- в активном режиме один из интерфейсов каждого сервера имеет дополнительный адрес.

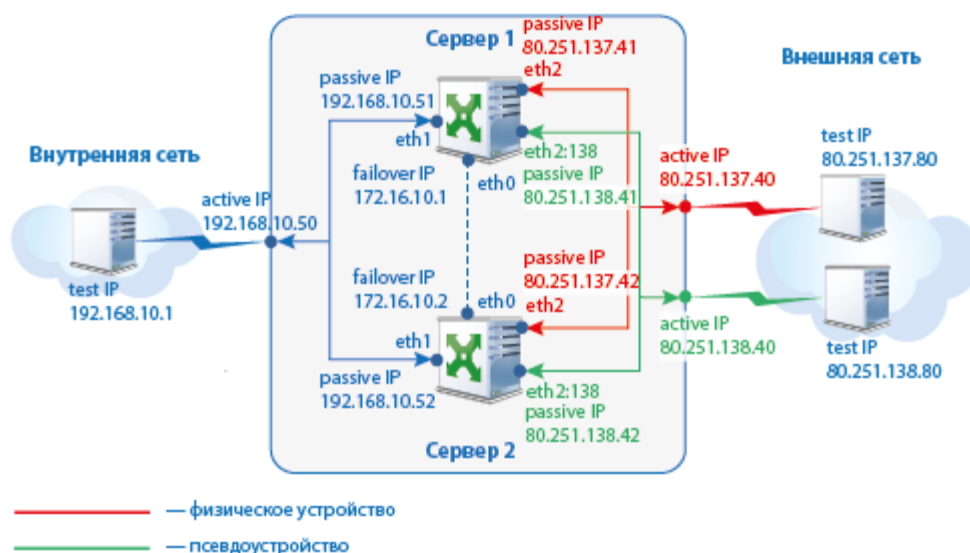


Рисунок 3. Схема организации кластера при использовании дополнительных адресов на интерфейсах

В таблице ниже приведены корректные настройки параметров системы защиты от сбоев для такой схемы (с учетом указанных ограничений).

Таблица 5. Пример настройки параметров системы защиты от сбоев при использовании дополнительных адресов на интерфейсах кластера

Настройки на первом сервере	Настройки на втором сервере
[channel]	[channel]
device= eth1	device= eth2
activeip= 192.168.10.50	activeip= 192.168.10.50
passiveip= 192.168.10.51	passiveip= 192.168.10.52
testip= 192.168.10.1	testip= 192.168.10.1
ident= if-internal	ident= if-internal
checkonlyidle= yes	checkonlyidle= yes
[channel]	[channel]
device= eth2	device= eth1
activeip= 80.251.137.40	activeip= 80.251.137.40
passiveip= 80.251.137.41	passiveip= 80.251.137.42
testip= 80.251.137.80	testip= 80.251.137.80
ident= if-external	ident= if-external
checkonlyidle= no	checkonlyidle= no

Настройки на первом сервере	Настройки на втором сервере
[channel]	[channel]
device= eth2:138	device= eth1:138
activeip= 80.251.138.40	activeip= 80.251.138.40
passiveip= 80.251.138.41	passiveip= 80.251.138.42
testip= 80.251.138.80	testip= 80.251.138.80
ident= if-external:138	ident= if-external:138
checkonlyidle= no	checkonlyidle= no
 [network]	 [network]
checktime= 10	checktime= 10
timeout= 2	timeout= 2
activeretries= 3	activeretries= 3
channelretries= 3	channelretries= 3
synctime= 5	synctime= 5
fastdown= yes	fastdown= yes
 [sendconfig]	 [sendconfig]
device= eth0	device= eth0
activeip= 172.16.10.2 (соответствует failover IP второго сервера)	activeip= 172.16.10.1 (соответствует failover IP первого сервера)

5

Запуск системы защиты от сбоев и работа с ней

Набор действий администратора по управлению системой защиты от сбоев в одиночном режиме работы сведен к минимуму: администратор может запустить или остановить демон failoverd. Управление производится с помощью управляющего скрипта failover, который помещается при установке в каталог /sbin.

Запуск системы производится командой `failover start`. При этом запускается демон failoverd. Останов системы производится командой `failover stop`. При этом демон failoverd завершает работу, сообщая об этом драйверу.

При загрузке системы автоматически загружается драйвер watchdog, а также выполняется команда `failover start`. В результате загружается демон failoverd, который в свою очередь производит старт остальных контролируемых служб ПО ViPNet Coordinator Linux (iplircfg, mftpd, algd, axis2.cgi). При выполнении команды `failover start` вручную в одиночном режиме перезапуска контролируемых приложений не производится.

Запрос информации о текущем состоянии системы защиты от сбоев производится командой `failover info` (см. «[Текущее состояние](#)» на стр. 46).

При работе в режиме кластера горячего резервирования серверов набор команд управления системой несколько отличается. Как уже было описано в разделе [Установка системы защиты от сбоев](#) (на стр. 17), включение данного режима осуществляется командой `failover install`, а отключение – `failover uninstall`. Чтобы запустить систему, используется команда `failover start`. При этом в случае первого старта после перехода из одиночного режима в режим кластера система стартует в пассивном режиме, а затем, если нет активной системы, по общему алгоритму переходит в активный режим.

Переход из одиночного режима работы в режим кластера необходимо осуществлять последовательно: сначала нужно дождаться перехода в активный режим одного из серверов, и

только после этого можно запустить систему на втором компьютере. Если все сделано правильно, второй компьютер останется в пассивном режиме. Посмотреть текущий режим можно командой `failover info`.

Для принудительного перевода сервера в активный или пассивный режим необходимо использовать соответствующие дополнительные аргументы команды: `failover start active` – для принудительного старта в активном режиме, `failover start passive` – для принудительного старта в пассивном режиме.



Внимание! Перед использованием команд принудительного перевода обязательно нужно убедиться, что второй сервер находится в режиме, противоположном режиму данного сервера. Помните, что запуск обоих компьютеров в активном режиме породит конфликт IP-адресов и другие неприятные последствия.

Если нужно остановить систему защиты от сбоев, например, для внесения изменений в файл конфигурации `failover.ini`, используйте команду `failover stop`. Последующая команда `failover start` запустит ее в том же режиме, в котором она была до остановки: если сервер работал в активном режиме – то произойдет старт в активном режиме, если в пассивном – то в пассивном. Таким образом, в штатном случае работы не нужно запоминать, в каком режиме работает данный сервер – система сделает это сама.

Если необходимо произвести изменения в файле конфигурации `iplir.conf`, то это нужно делать на активном сервере кластера, для чего нужно сначала остановить управляющий демон командой `iplir stop`. После редактирования файла нужно запустить управляющий демон командой `iplir start`. Для редактирования файла конфигурации транспортного модуля необходимо остановить демон `mftpd` командой `mftp stop`, отредактировать файл конфигурации, а затем запустить демон командой `mftp start`. Описанные процедуры рекомендуется производить только на активном сервере кластера. Все изменения конфигурации попадут на пассивный сервер автоматически.



Примечание. На пассивном сервере кластера управляющий демон и демон транспортного модуля MFTP функционируют в служебном режиме, поэтому запрос информации (`iplir info`, `iplir view`, `mftp info`) в этом случае работать не будет.

6

Просмотр информации о работе системы защиты от сбоев

Утилита для просмотра информации о работе системы защиты от сбоев	45
Текущее состояние	46
Журнал переключений	48

Утилита для просмотра информации о работе системы защиты от сбоев

Для просмотра информации о работе системы защиты от сбоев используется утилита `failover_info`, которая помещается после установки в каталог `/sbin`. Обычно она запускается из управляющего скрипта `failover` указанием дополнительных аргументов командой строки. Для работы утилиты `failover_info` необходимо корректно настроить конфигурационный файл `/etc/iplirnetpsw`.

Информацию, которую можно получить с помощью описываемой утилиты, можно разделить на две категории: информация о текущем состоянии (см. [«Текущее состояние»](#) на стр. 46) и информация из журнала переключений кластера горячего резервирования (см. [«Журнал переключений»](#) на стр. 48).

Текущее состояние

Информация о текущем состоянии системы защиты от сбоев включает в себя:

- версию ПО ViPNet Coordinator Linux и демона failoverd, установленных на сервере;
- идентификатор и имя сервера в сети ViPNet;
- режим работы системы защиты от сбоев (одиночный или режим кластера горячего резервирования серверов);
- локальное время на сервере;
- текущую информацию о состоянии управляющего демона, демонов mftpd, algd, failoverd и axis2.cgi.

Для запроса используется команда `failover info`, в результате выполнения которой необходимая информация выводится на текущую консоль.

Формат вывода информации при работе в одиночном режиме следующий:

```
Versions: ViPNet 4.0.0 (475), daemon 1.3 (14)
Workstation configured for ID 10E1079E (Vipnux-_1_3)
The workstation works in a single mode of protection against failures
Workstation time (utc: 1174558030) Thu Mar 25 13:07:10 2010
```

failover mode	* single — режим работы системы защиты от сбоев;
failover uptime	* 6d 0:23 — время работы демона failoverd;
total cpu	* 0% — общая загрузка CPU в системе;
failover state	* works — состояние демона failoverd;
failover cpu	* 0% — загрузка CPU демоном failoverd;
iplir state	* works — состояние управляющего демона iplircfg;
iplir cpu	* 0% — загрузка CPU демоном iplircfg;
mftp state	* works — состояние демона mftpd;
mftp cpu	* 0% — загрузка CPU демоном mftpd;
alg state	* works — состояние демона обработки прикладных протоколов algd;
alg cpu	* 0% — загрузка CPU демоном algd;
webgui state	* works — состояние демона axis2.cgi;
webgui cpu	* 0% — загрузка CPU демоном axis2.cgi.

Формат вывода информации при работе в режиме кластера горячего резервирования серверов следующий:

```
Versions: ViPNet 4.0.0 (475), daemon 1.3 (14)
Workstation configured for ID 1031F (Cluster for SGA2)
Workstation works in a mode of hot reservation
Workstation time (utc: 1174916130) Mon Mar 29 17:35:30 2010
```

	* local	* remote
failover mode	* active	* passive
failover uptime	* 3d 5:26	* 3d 4:11
total cpu	* 80%	* 0%
failover state	* works	* works
failover cpu	* 7%	* 0%
iplir state	* works	* works
iplir cpu	* 0%	* 0%
mftp state	* works	* works
mftp cpu	* 66%	* 0%
alg state	* works	* works
alg cpu	* 0%	* 0%
webgui state	* works	* works
webgui cpu	* 0%	* 0%

Значения столбцов таблицы в этом случае аналогичны значениям, выводимым для одиночного режима. Отличие состоит в том, что данные выводятся для обоих серверов кластера горячего резервирования. Обозначения `local` и `remote` соответствуют локальному серверу (с которого был произведен запрос информации) и второму компоненту кластера. При работе в режиме кластера команду `failover info` можно выполнять как на активном, так и на пассивном серверах.

При выводе информации используются следующие обозначения режимов:

- `single` – одиночный режим работы;
- `active` – активный режим кластера горячего резервирования серверов;
- `passive` – пассивный режим кластера горячего резервирования серверов.

Используемые обозначения состояний:

- `works` – приложение работает корректно (с точки зрения системы защиты от сбоев);
- `stopped` – приложение остановлено пользователем;
- `unknown` – состояние приложения неизвестно. Данное состояние может быть установлено в случае, если был идентифицирован сбой контролируемого приложения, попытка его перезапуска, но данных о его корректном старте пока нет.

Журнал переключений

В журнале переключений фиксируются события, происходящие в системе защиты от сбоев при ее работе в режиме кластера горячего резервирования серверов (см. «[Режим кластера горячего резервирования](#)» на стр. 12).

Для просмотра записей из журнала переключений выполните команду `failover view`, задав в параметрах начало и конец временного интервала:

```
failover view -b <DD.MM.YYYY[.hh.mm.ss]> -e <DD.MM.YYYY[.hh.mm.ss]>
```

В результате выполнения данной команды выводится следующая информация:

- версия ПО ViPNet в составе ViPNet Coordinator Linux и версия демона `failoverd`;
- идентификатор и имя сервера (как узла сети ViPNet);
- режим работы системы защиты от сбоев (всегда режим кластера горячего резервирования);
- локальное время на сервере;
- список записей из журнала переключений, попадающих в заданный интервал времени.

Информация выводится в следующем формате:

```
View journal of failover switching
Versions: ViPNet 4.0.0 (475), daemon 1.3 (14)
Workstation configured for ID 1031F (Cluster for SGA2)
Workstation works in a mode of hot reservation
Workstation time (utc: 1174916969) Mon Mar 29 17:49:29 2010

09 Mar 2014 12:51:42    <P_START> Start failover daemon in passive mode
22 Mar 2014 12:27:27    <A_START> Start failover daemon in active mode
22 Mar 2014 14:10:35    <A_START> Start failover daemon in active mode
22 Mar 2014 15:30:46    <BOOT> Boot the system
23 Mar 2014 11:09:07    <SWITCH> Switch server from passive mode to active mode
```

Первый столбец содержит дату и время события, второй столбец — идентификатор и полное наименование события.

Если в заданном интервале времени не было ни одного события, то выводится сообщение:

```
There are no records in journal of switchings
```

При выводе информации используются следующие обозначения событий:

```
<BOOT> Boot the system — загрузка ОС;
<P_START> Start failover daemon in passive mode — старт в пассивном режиме;
<A_START> Start failover daemon in active mode — старт в активном режиме;
<SWITCH> Switch server from passive mode to active mode — переключение серверов.
```

Команда `failover view` доступна только при работе системы защиты от сбоев в режиме кластера горячего резервирования серверов. Команда доступна на обоих серверах кластера.

7

Тонкая настройка системы защиты от сбоев

Нестандартная конфигурация сетевых настроек	50
Использование кластера горячего резервирования серверов совместно с ОС Solaris	51
Работа кластера горячего резервирования серверов совместно с коммутационным оборудованием Cisco	52

Нестандартная конфигурация сетевых настроек

Если в системе используется какая-либо нестандартная маршрутизация или при конфигурировании сетевых интерфейсов нужно производить какие-то дополнительные действия, то необходимо использовать специальные параметры `beforeifconf` и `afterifconf` в секциях `[channel]` и `[network]` (см. «[Файл конфигурации системы защиты от сбоев](#)» на стр. 22). В качестве значений этих параметров можно задать непосредственно команды, которые будут переданы на выполнение системной оболочке, или сценарий, включающий необходимый набор команд. Способ задания статической маршрутизации в случае использования схемы организации кластера горячего резервирования в условиях ограничений по выделению IP-адресов описан в разделе [Схема организации кластера в условиях ограничений по выделению IP-адресов](#) (на стр. 35).

Использование кластера горячего резервирования серверов совместно с ОС Solaris

Если кластер горячего резервирования должен взаимодействовать с узлами, на которых установлена ОС Solaris, то эти узлы необходимо дополнительно настроить для корректной работы с кластером. Особенность ОС Solaris заключается в ее работе с ARP. В отличие от многих других систем, ОС Solaris не обновляет информацию о MAC-адресе удаленного узла при приходе пакета от него, если в этом пакете указан не тот MAC-адрес, который система имеет в своей ARP-таблице для этого удаленного узла. Вместо этого при несовпадении MAC-адресов система блокирует все пакеты от удаленного узла, пока не истечет время хранения записи в таблице ARP (оно составляет по умолчанию 20 минут). Поэтому при переключении активного сервера в кластере на другой компьютер (при этом IP-адрес кластера остается прежним, но меняется MAC-адрес, поскольку большинство сетевых карт не позволяют его переключать) узлы с ОС Solaris блокируют все пакеты от кластера в течение 20 минут.

Исправить ситуацию можно путем установки меньшего времени жизни записи в таблице ARP на узлах с ОС Solaris. Для этого используются следующие команды:

```
ndd -set /dev/arp arp_cleanup_interval 30000
ndd -set /dev/ip ip_ire_arp_interval 60000
```

Эти команды используются в ОС Solaris версий 8 и 9. Для более ранних версий параметр `ip_ire_arp_interval` необходимо заменить на `ip_ire_flush_interval`.

При этом время жизни ARP-записи устанавливается равным минуте (меньшие значения системой не поддерживаются). Если узлы с ОС Solaris используются как тестовые для кластера горячего резервирования, то необходимо установить параметры проверки сети так, чтобы тайм-аут был не меньше минуты.

Работа кластера горячего резервирования серверов совместно с коммутационным оборудованием Cisco

На практике часто встречаются схемы подключения серверов кластера горячего резервирования к различному коммутационному оборудованию производства Cisco, так как данное оборудование имеет широкое распространение по всему миру. Это могут быть коммутаторы (switch), маршрутизаторы (router) и другое оборудование. Конфигурация данного оборудования может напрямую влиять на корректную работу механизмов кластера горячего резервирования. В частности, на указанном коммутационном оборудовании администратором могут быть заданы настройки, запрещающие прохождение тех или иных сетевых пакетов, среди которых могут оказаться служебные пакеты, необходимые для правильного функционирования ПО ViPNet Coordinator Linux в режиме кластера горячего резервирования. В связи с этим при организации схем включения кластера горячего резервирования необходимо проверить соответствующие настройки сетевого коммутационного оборудования:

- Должны пропускаться ICMP echo-запросы с IP-адресов активного сервера до всех заданных IP-адресов `testip` (см. «Секция [\[channel\]](#)» на стр. 23) и ответы на них.
- Должны пропускаться ARP-запросы с IP-адресов пассивного сервера для IP-адресов активного сервера и ответы на них.

Указанные правила касаются всех контролируемых сетевых интерфейсов (для которых существует секция `[channel]` в файле `failover.ini`) в схеме кластера горячего резервирования.



Задание дополнительных IP-адресов на кластере

ViPNet Coordinator Linux позволяет настроить дополнительные IP-адреса на интерфейсах кластера горячего резервирования. Такая необходимость возникает, когда на одном сетевом адаптере требуется поддерживать несколько IP-адресов. Например, с помощью дополнительных адресов можно предоставить внешний доступ к различным сервисам, размещенным на серверах локальной сети, подключив серверы к координатору. Пример подобного использования дополнительных адресов на одиночном координаторе приведен в документе «ViPNet Coordinator Linux. Руководство администратора». В этом примере одиночный координатор можно заменить кластером с той разницей, что дополнительные IP-адреса надо задать описанным ниже способом.

Для задания дополнительных IP-адресов на кластере необходимо использовать параметры `afterifconf` и `beforeifconf` в секции `[network]` (см. «Секция `[network]`» на стр. 25). В этих параметрах можно задать набор команд, которые будут переданы на выполнение системной оболочке, или сценарий, содержащий эти команды. Непосредственное задание дополнительных адресов осуществляется с помощью стандартных команд конфигурирования сети (например, `ifconfig` и `route`).

Например, пусть имеется файл `add_alias.sh`, содержащий следующий сценарий:

```
#!/bin/sh
if [ "$FAILOVER_MODE" = "ACTIVE" ] ; then
    ifconfig eth2:0 80.251.137.10
    ifconfig eth2:1 80.251.137.100
else
    ifconfig eth2:0 80.251.137.10 down
    ifconfig eth2:1 80.251.137.100 down
fi
exit 0
```

и этот сценарий задан в параметре `afterifconf` в секции `[network]`:

```
[network]
afterifconf= /sbin/add_alias.sh
```

Согласно этому сценарию, при переходе сервера кластера в активный режим на интерфейсе `eth2` будут установлены дополнительные адреса `80.251.137.10` и `80.251.137.100`. При переключении сервера в пассивный режим оба дополнительных адреса будут выключены.