Применение Pacemaker для повышения надежности доступа к критическим данным

Г.М. Михайлов[1], М.А. Жижченко[1], А.М. Чернецов[1,2]

[1] Вычислительный центр им. А.А. Дородницына ФИЦ ИУ РАН
[2] Национальный исследовательский университет «МЭИ»

Аннотация. В докладе рассматривается задача повышения надежности функционирования ресурсов организации, доступ к которым является критичным. Рассматривается случай двухузлового кластера высокой доступности (High Availability) на базе Pacemaker и распределенной файловой системы DRBD, обеспечивающий работу сайта организации и сервера баз данных MySQL в режиме active-passive. Приведена конфигурация для большинства используемых ресурсов HA-кластера.

Ключевые слова: PaceMaker, DRBD, MySQL, отказоустойчивые системы

Using Pacemaker to Improve Reliability of Access to Critical Data

G.M. Mikhaylov[1], M.A. Zhyzchenko[1], A.M. Chernetsov[1,2]

[1] Dorodnicyn Computing Centre FRC CSC RAS
[2] National Research University "MPEI"

Abstract. The problem of improving the reliability of the organization's critical access resources is discussed in this article. We consider the case of a two-node high availability cluster based on Pacemaker and a DRBD distributed file system, which ensures the operation of an organization's website and a MySQL database server in active-passive mode. The configuration is shown for most of the used HA-cluster resources.

Keywords: PaceMaker, DRBD, MySQL, High-Availability Clusters

1. Введение

В докладе рассматривается задача повышения надежности функционирования ресурсов организации, доступ к которым является критичным. В качестве примеров таких систем можно привести информационные системы, обеспечивающие выполнение государственных

услуг, услуг по обработке обращений граждан, банковских услуг. Для всех приведенных примеров важным является обеспечение работоспособности системы при различных сбоях, как программных, так и аппаратных.

Задача повышения надежности далеко не нова. Актуальность решения этой проблемы остается неизменной в течение всего исторического развития компьютерных технологий и информатики. В части построения информационных систем и баз данных единственным реально работающим способом является дублирование физического (виртуального) оборудования и настройка передачи данных и ресурсов между узлами. Даже при использовании систем виртуализации [1] при увеличении нагрузки на узлы решение проблемы приводит к появлению нового физического оборудования и обеспечению взаимодействия между узлами. Таким образом, мы говорим об обеспечении аппаратной избыточности.

2. Отказоустойчивые кластеры

Допустим, что перед организацией стоит задача обеспечения непрерывного доступа к некоторым приложениям (сайт, база данных и др.). Если доступ к каким-то ресурсам слишком важен, то ресурсы можно дублировать. Это известное понятие отказоустойчивого кластера (High Avalaibility) [2]. HA-кластеры проектируются в соответствии с методиками обеспечения высокой доступности и гарантируют минимальное время простоя за счёт аппаратной избыточности. Без применения кластеризации сбои сервера приводят к тому, что поддерживаемые им приложения или сетевые сервисы становятся недоступными до тех пор, пока работоспособность сервера не будет восстановлена. Отказоустойчивая кластеризация исправляет эту проблему, обеспечивая перезапуск приложения на других узлах кластера без вмешательства администратора в случае обнаружения аппаратных или программных сбоев.

Отказоустойчивые кластеры широко используются для поддержки важных баз данных, хранения файлов в сети, бизнес-приложений.

Коммерческие решения для HA-кластеров, среди которых можно упомянуть IBM PowerHA SystemMirror[3], HP Serviceguard [4] и Oracle Solaris Cluster [5], а также Microsoft Windows Failover Cluster [6], неоправданно дорогие. Кроме того, вероятность появления различных новых санкций и «правило» использования отечественного программного обеспечения (ПО) позволяет сделать заключение, что нужно использовать open-source решения. К таковым относятся в частности OpenStack[7], LinuxHA [8] и его развитие RedHat Cluster (Pacemaker) [9].

Рассмотрим задачу обеспечения отказоустойчивой связки ПО Apache+Mysql. Можно использовать разные общедоступные решения, но нами была выбрана типовая схема pcs/corosync (Pacemaker), а в качестве распределенной системы хранения (SAN) – DRBD [10](Distributed

Replicated Block Device). Для развертывания была использована стандартная инструкция по установке [11].

Одна из возникающих задач – обеспечение непрерывной репликации данных между узлами. Для этой цели была выбрана файловая система DRBD v.9.

Фактически DRBD обеспечивает межузловую синхронизацию между выделенными узлами. Конфигурация DRBD находится в каталоге */etc/drbd.d/*. Выделяется глобальная конфигурация и конфигурация устройств DRBD (адреса каждого из узлов, порт работы DRBD). На каждом узле создаются устройства вида /dev/drbdX. Для управления распределенной файловой системой используется команда **drbdadm**. После инициализации необходимо провести форматирование файловой системы (ФС). Авторы обращают внимание, что хотя формально ФС может быть любой (ext4, btrfs, xfs, …), но в реальности необходимо учитывать особенности ФС. Так, ФС ext4 при всех её преимуществах категорически использовать нельзя. Причина связана с тем, что экстенты ФС записываются на диск со значительной задержкой по времени, что недопустимо при синхронизации данных в DRBD. Использование ext4 приводит к сбоям репликации и потере данных.

Для корректной работы системы необходимо иметь минимум 3 узла (для корректного голосования). Увы, авторам в рамках доступных ресурсов было доступно только два. Pacemaker имеет специальный режим работы, учитывающий эту ситуацию, и в результате система будет функционировать. Существенным недостатком в этом случае является фактическая неработоспособность распределенной ФС DRBD. Хотя формально она работает и обеспечивает репликацию данных между узлами системы, в реальности при сбоях восстановление превращается в очень долгий и мучительный процесс. Если есть средства, лучше потратить их на третий узел хранилища данных и исключить эти проблемы.

В целом HA- кластер может функционировать в различных режимах работы [2], среди которых укажем следующие:

- Active-passive. Имеет полное резервирование (работоспособную копию) каждого узла. Резерв включается в работу только тогда, когда отказывает соответствующий основной узел. Эта конфигурация требует значительных избыточных аппаратных средств.
- Active-active. Часть трафика, обрабатывавшаяся отказавшим узлом, перенаправляется какому-либо работающему узлу или распределяется между несколькими работающими узлами. Схема используется в случае, когда узлы имеют однородную конфигурацию программного обеспечения и выполняют одинаковую задачу.
- N+1. Имеется один полноценный резервный узел, к которому в момент отказа переходит роль отказавшего узла. В случае гетерогенной программной конфигурации основных узлов дополнительный узел

должен быть способен взять на себя роль любого из основных, за резервирование которых он отвечает. Такая схема применяется в кластерах, обслуживающих несколько разнородных сервисов, работающих одновременно; в случае единственного сервиса такая конфигурация вырождается в Active / passive.

Для контроля работоспособности узлов в кластере обычно используется передача непрерывного периодического сигнала («пульса», heartbeat) во внутренней сети кластера от каждого из узлов. По наличию «пульса» управляющее ПО судит о нормальной работе соседних узлов. С этим связана серьёзная проблема split-brain — в случае одновременного разрыва множества соединений во внутренней сети кластера по какой-либо причине (сбоя питания, неисправности сети и т.п.), узел, не способный корректно обработать данную ситуацию, начинает вести себя так, как будто все остальные узлы кластера вышли из строя. Соответственно, он запускает дубликаты уже работающих в кластере ресурсов, что может привести к повреждению данных в SAN.

3. Система Pacemaker

Настройка и управление в Pacemaker осуществляется с использованием команд **pcs cluster**. Настройка и управление ресурсами (создание/удаление, просмотр состояния, блокировка/запуск) ресурсов осуществляется с использованием команд **pcs resource**.

В любой системе управления кластером, в том числе и Pacemaker, выделяют следующие ресурсы:

-Ip адреса. На каждом узле обязательно наличие двух интерфейсов: внутреннего и внешнего. Внутренний используется для взаимодействия процессов DRBD и иных нужд, внешний – это адрес, по которому идет доступ снаружи. При смене активного узла производится требуемая перенастройка (перевыбор адресов).

-Файловые системы (например, отдельный каталог – корень распределенной ФС).

-Ресурсы (например, apache, mysqld, vsftpd)

HA-кластер для внешних пользователей (запросов) выглядит логически единым сервером с постоянным IP-адресом. В реальности на запросы отвечает либо какой-то выделенный узел кластера (в случае режима работы active-passive), либо любой узел (active-active).

Узлы HA-кластера могут различаться по аппаратуре, но по программному обеспечению никогда, – везде должно быть установлено одинаковое ПО.

В рамках настройки работы есть взаимосвязи между службами, определяющие зависимости запуска служб друг от друга.

Например, СУБД можно запускать только после инициализации файловой системы. Веб-сервер следует запускать только после того, как

запустится СУБД. И в каждом из этих случаев до запуска DRBD «в работу» потребуется обеспечить функционирование связи между узлами кластера (внутренний трафик).

Важной особенностью работы с ресурсами является то, что нельзя вручную запускать/останавливать ресурсы. Все эти действия можно выполнять только через интерфейс Corosync. Это утверждение относится и к службе DRBD.

Было развернуто 2 гомогенных узла **web3** и **web4** в конфигурации на базе серверной платформы Intel R1304WT2GSR (Процессор Intel Xeon E5-2620v3 2.4GHz, оперативная память 64 GB DDR4-2400 Single Rank x4 CL17 1.2v ECC Registered DIMM, жёсткий диск SAS HDD SAS 3,5" 1TB, 12Gb/s 7.2K - 2 шт.). Жёсткие диски были установлены в режим RAID 1. В качестве операционной системы была выбрана CentOS 7.9. На узлах были проведены обновления всех пакетов через **yum update**.

Все узлы синхронизируют точное время по протоколу ntp. На каждом узле был создан пользователь hacluster с одинаковыми настройками, включая пароль. Доступ по протоколу *ssh* обеспечивался с использованием закрытых ключей узлов, что позволило избавиться от ввода паролей при выполнении команды **ssh**. Был настроен внутренний сетевой интерфейс для передачи между узлами служебного трафика. Предлагаемый по умолчанию фильтр *firewalld* был отключен. Вместо этого использовались классические *iptables*. Также полностью отключена SELinux. Обращаем внимание, даже перевод SELinux в режим *permissive* не решал возникающие проблемы.

Далее была настроена распределенная ФС DRBD. Приведем конфигурационный файлы: общие настройки и ресурс /site:

```
global {
        usage-count no;
udev-always-use-vnr; # treat implicit the same as explicit volumes
}
common {
handlers {
}
startup {
wfc-timeout 120;
degr-wfc-timeout 60;
}
options {
}
disk {
          on-io-error detach;
}
net {
        protocol C;
        verify-alg sha1;
        after-sb-0pri discard-least-changes;
```

```
        after-sb-1pri discard-secondary;
        after-sb-2pri call-pri-lost-after-sb;
    }
    syncer {
        rate 5M;
    }
    }
    resource site {
    net {
    allow-two-primaries;
    }
    on web3 {
    device     /dev/drbd0;
    disk       /dev/sda5;
    address    192.168.1.1:7788;
    meta-disk  internal;
    }
    on web4 {
    device     /dev/drbd0;
    disk       /dev/sda5;
    address    192.168.1.2:7788;
    meta-disk  internal;
    }
    }
```

Обращаем внимание, что настройки проведены с учётом работы СУБД MySQL и предусматривают возможность автоматического разрешения ситуации split-brain.

После настройки ФС были настроены ресурсы веб-сервера и СУБД. Для проверки работоспособности apache используется стандартный status-page вида

```
<Location /server-status>
SetHandler server-status
Require local
</Location>
```

Этот status-page подключен как параметр statusurl

```
pcs resource create WebSite ocf:heartbeat:apache \
configfile=/etc/httpd/conf/httpd.conf \
statusurl="http://localhost/server-status" \
op monitor interval=1min
```

Аналогично был создан ресурс для СУБД MySQL:

```
pcs -f clust_cfg resource create mysql-server ocf:heartbeat:mysql binary="/usr/sbin/mysqld" config="/etc/my.cnf" datadir="/site/mysql" pid="/var/lib/mysql/run/mysqld.pid" socket="/var/lib/mysql/mysql.sock" additional_parameters="--bind-address=0.0.0.0" op start timeout=60s op stop timeout=60s op monitor interval=20s timeout=30s
```

Обращаем внимание, что по умолчанию сервер СУБД разворачивается в /var/lib/mysql. В нашем же случае все файлы СУБД должны находиться на SAN. При обновлении версий ПО будет

происходить установка в /var/lib/mysql и администратору будет необходимо переносить эти обновления в соответствующее место на SAN.

После создания необходимых ресурсов с использованием **pcs constraint** были установлены зависимости между ресурсами. Были использованы зависимости двух видов: запуск одного ресурса после второго (Ordering) и запуск ресурсов одновременно друг с другом (Collocation):

```
# pcs constraint
Ordering Constraints:
  promote WebDataClone then start WebFS (kind:Mandatory)
  start ClusterIP then start mysql-server (kind:Mandatory)
  start mysql-server then start WebSite (kind:Mandatory)
Colocation Constraints:
  WebFS with WebDataClone (score:INFINITY) (with-rsc-role:Master)
  mysql-server with WebDataClone (score:INFINITY) (with-rsc-role:Master)
```

4. Заключение

В рамках проведенного эксперимента был полностью развернут и введен в эксплуатацию тестовый стенд. За время эксплуатации выявлены недостатки при работе с DRBD, связанные с малым числом узлов. Они описаны ранее в тексте. Разработанное решение было запущено в режим производственной эксплуатации и успешно функционирует в Национальном исследовательском университете «МЭИ».

Также обращаем внимание на следующее обязательное правило. При выполнении каких-либо настроек ОС (изменение конфигурационных файлов) или необходимости установки/обновления ПО необходимо все эти действия повторять на всех узлах НА-кластера.

Работа выполнена в рамках темы государственного задания «МАТЕМАТИЧЕСКИЕ МЕТОДЫ АНАЛИЗА ДАННЫХ И ПРОГНОЗИРОВАНИЯ» (ФИЦ ИУ РАН).

Литература

1. Михайлов Г.М., Рогов Ю.П., Чернецов А.М. Опыт использования виртуальных ресурсов в ИВС ВЦ РАН. Труды Международной суперкомпьютерной конференции "Научный сервис в сети Интернет: все грани параллелизма" (г. Новороссийск, 23-28 сентября 2013 г.). - М,: МГУ, 2013 – 589 С. С. 522-523.
2. van Vugt, Sander (2014), Pro Linux High Availability Clustering, Apress, ISBN 978-1484200803
3. IBM PowerHA SystemMirror URL: https://www.ibm.com/products/powerha
4. HP Serviceguard URL https://support.hpe.com/hpesc/public/docDisplay?docId=emr_na-c02960047-2

5. Oracle Solaris Cluster URL https://www.oracle.com/solaris/technologies/cluster-overview.html
6. Microsoft Windows Failover Cluster URL: https://docs.microsoft.com/en-us/windows-server/failover-clustering/failover-clustering-overview
7. OpenStack URL https://docs.openstack.org/ha-guide/
8. LinuxHA URL http://www.linux-ha.org/wiki/Main_Page
9. RedHat Cluster (Pacemaker). URL https://access.redhat.com/documentation/ru-ru/red_hat_enterprise_linux/5/html/cluster_suite_overview/ch.gfscs.cluster-overview-cso
10. https://linbit.com/drbd-user-guide/drbd-guide-9_0-en/
11. Документация по установке и развертыванию Pacemaker URL: https://clusterlabs.org/pacemaker/doc/en-US/Pacemaker/2.0/pdf/Clusters_from_Scratch/Pacemaker-2.0-Clusters_from_Scratch-en-US.pdf

References

1. Mikhaylov G.M., Rogov Yu.P., Chernetsov A.M. Opyt ispol'zovaniya virtual'nykh resursov v IVS VTs RAN. Trudy Mezhdunarodnoy superkomp'yuternoy konferentsii "Nauchnyy servis v seti Internet: vse grani parallelizma" (g. Novorossiysk, 23-28 sentyabrya 2013 g.). -M,: MGU, 2013 – 589 P. P. 522-523.
2. van Vugt, Sander (2014), Pro Linux High Availability Clustering, Apress, ISBN 978-1484200803
3. IBM PowerHA SystemMirror URL: https://www.ibm.com/products/powerha
4. HP Serviceguard URL https://support.hpe.com/hpesc/public/docDisplay?docId=emr_na-c02960047-2
5. Oracle Solaris Cluster URL https://www.oracle.com/solaris/technologies/cluster-overview.html
6. Microsoft Windows Failover Cluster URL: https://docs.microsoft.com/en-us/windows-server/failover-clustering/failover-clustering-overview
7. OpenStack URL https://docs.openstack.org/ha-guide/
8. LinuxHA URL http://www.linux-ha.org/wiki/Main_Page
9. RedHat Cluster (Pacemaker). URL https://access.redhat.com/documentation/ru-ru/red_hat_enterprise_linux/5/html/cluster_suite_overview/ch.gfscs.cluster-overview-cso
10. https://linbit.com/drbd-user-guide/drbd-guide-9_0-en/
11. Документация по установке и развертыванию Pacemaker URL: https://clusterlabs.org/pacemaker/doc/en-US/Pacemaker/2.0/pdf/Clusters_from_Scratch/Pacemaker-2.0-Clusters_from_Scratch-en-US.pdf