УДК 681.3
DOI 10.26732/2618-7957-2018-2-89-94

ОРГАНИЗАЦИЯ КЛАСТЕРА ВЫСОКОЙ ГОТОВНОСТИ

Н. Д. Иванов
Сибирский федеральный университет,
г. Красноярск, Российская Федерация

Одним из способов обеспечения надежности системы обработки и хранения данных, ежесекундного обеспечения пользователей оперативной и достоверной информацией является кластеризация серверных систем, за счет которой поддерживается ее высокий уровень.
В статье рассмотрено построение кластера высокой готовности. Для повышения отказоустойчивости используются несколько серверов, подключенных через несколько коммутаторов. В серверах предусматривается несколько сетевых интерфейсов с использованием технологии bonding и RAID-массив. Серверы и коммутаторы подключаются к источникам бесперебойного питания, которые тоже резервируются.
Приведены программные пакеты и их конфигурационные файлы, необходимые для функционирования кластера: bond, drbd, pacemaker, corosync. Дано описание конфигурации pacemaker и ресурс агентов: виртуальный IP-адрес – для того, чтобы у кластера был единый адрес, drbd – для зеркалирования RAID-массивов, programmControl – для управления приложениями, wakeUp – для управления включением и выключением серверов.
В результате был спроектирован кластер высокой готовности, у которого, помимо серверной части, резервируется сеть и память. При переходе от одного сервера к другому обслуживание клиентов сервером прерывается на очень короткое время.

Ключевые слова: кластер высокой готовности, ресурс агент, pacemaker, corosync, drbd, bound, WOL.

Введение

Надежность систем обработки данных, их способность ежесекундно обеспечивать пользователей оперативной и достоверной информацией – одно из важнейших условий эффективной работы различных компьютерных систем. Существует множество технических решений, обеспечивающих необходимый уровень надежности и отказоустойчивости информационных систем. Одно из таких решений – кластеризация серверных систем, за счет которой поддерживается высокий уровень готовности [1–4].

Система может использоваться для бесперебойного принятия и обработки данных, поступающих со спутников. Это система высокой готовности, которая в случае отказа одного из серверов или его останова для проведения профилактических работ, продолжит работу на другом сервере кластера, имеющем те же данные и сервисы.

Конфигурация серверов

В разрабатываемой системе будут использоваться четыре сервера. Три сервера объединены в кластере, а один сервер, предварительно настроенный и подготовленный для работы в составе кластера, находится в ЗИП (запасные части, инструменты, принадлежности). Один сервер в кластере активный, и на нем запущены нужные программы и сервисы. Второй сервер находится в горячем резерве – включен, но не запущены программы и сервисы. Третий сервер находится в холодном резерве (отключен).

При полном отказе активного сервера или сервера в горячем резерве сервер, находящийся в холодном резерве, должен перейти в горячий резерв и включиться. Для этого по сети на него посылается так называемый «магический пакет». Структура данного пакета следующая: в начале располагаются 6 байт, равных 0xFF, затем MAC-адрес включаемого сервера, повторенный 16 раз. Материнские платы, сетевые карты и BIOS всех серверов должны поддерживать функцию Wake-On-LAN (пробуждение по сети – WOL), а, кроме того, материнские платы должны поддерживать подачу питания на сетевую карту при холодном подключении.

Для обеспечения безотказной работы в части сетевого взаимодействия на каждом сервере будет использоваться резервирование сетевых интерфейсов. Необходимо использовать не

менее двух сетевых интерфейсов. Если главный сетевой интерфейс выйдет из строя, работу продолжит следующий сетевой интерфейс. Для связи серверов между собой понадобятся коммутаторы. Коммутаторов нужно столько же, сколько сетевых интерфейсов на одном сервере, среди которых также один – основной, другие – резервные. Если один коммутатор выйдет из строя, то связь с кластером не прервется. Соответственно будут работать другие сетевые интерфейсы, подключенные к резервному коммутатору.

Для избегания потери данных на каждом сервере понадобится не менее двух HDD (SSD) дисков, которые объединяются в RAID 1. В случае отказа одного диска работа сервера не прекратится.

Чтобы обеспечить работу серверов при отключении электроэнергии, каждый сервер подключается к своему источнику бесперебойного питания (ИБП). Коммутаторы также подключаются к ИБП: первый коммутатор – к первому и второму ИБП, второй – ко второму и третьему ИБП. Структурная схема кластера приведена на рис. 1.

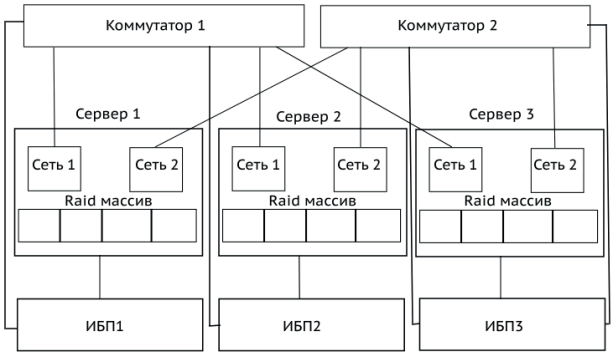

Рис. 1. Схема кластера

Подготовка серверов

В серверах будет использоваться операционная система семейства Linux. Программные средства, используемые для построения кластера, предназначены для ОС семейства Linux. В других операционных системах используются иные средства.

Для работы WOL необходимо в BIOS в разделе управления питанием выставить соответствующие опции в положение включено.

Для удобного копирования настроек с одного сервера на другие, а также для автоматического перевода серверов в холодный резерв, понадобиться установить программный пакет ssh.

Для работы сетевых интерфейсов будет использоваться технология bonding [5]. Bond будет работать в режиме *mode=1* (active-backup), при котором один из интерфейсов активен. Если активный интерфейс выходит из строя, другой интерфейс заменяет его. При переходе на другой интерфейс активные соединения не прерываются и данные не теряются. Для работы bond потребуется установить пакет ifenslave и настроить конфигурационный файл */etc/network/interfaces*:

```
auto bond0 eth0 eth1
iface bond0 inet static
address 10.2.110.1
netmask 255.255.255.0
network 10.2.110.0
gateway 10.2.110.255
bond_mode balance-tlb
bond_miimon 100
bond_downdelay 200
bond_updelay 200
slaves eth0 eth1
```

Данная конфигурация приведена для двух сетевых интерфейсов eth0 и eth1. Строчка address 10.2.110.1 означает, что теперь сервер имеет IP-адрес 10.2.110.1.

Для зеркалирования RAID-массивов на все узлы (серверы кластера) будет использоваться пакет drbd [6]. С его помощью данные с активного сервера будут записываться на сервер в горячем резерве в режиме реального времени. При включении сервера холодного резерва в течение некоторого времени будет происходить запись отсутствующих данных с активного сервера на вновь подключенный сервер. Для работы потребуется установить пакет drbd и настроить его конфигурационный файл */etc/drbd.conf*:

```
global { usage-count no; }
common { syncer { rate 100M; } }
resource r0 {
    protocol C;
    startup {
        wfc-timeout  15;
        degr-wfc-timeout 60;
    }
    net {
        cram-hmac-alg sha1;
        shared-secret "secret";
    }
    on node1 {
        device /dev/drbd0;
        disk /dev/sdb1;
        address 10.2.110.1:7788;
        meta-disk internal;
    }
    on node2 {
        device /dev/drbd0;
        disk /dev/sdb1;
        address 10.2.110.2:7788;
        meta-disk internal;
    }
}
```

Кроме того нужно изменить файл */etc/hosts* и присвоить соответствующим IP-адресам их названия. В данном случае это адреса node1 и node2. После этого необходимо инициализировать хранилище метаданных и запустить на всех серверах сервис drbd.

Ресурс агенты

Для управления кластером понадобится установить пакеты pacemaker и corosync[7]. После их установки нужно настроить конфигурационные файлы. Содержимое файла */etc/corosync/corosync.conf*:

```
totem {
     version: 2
    token: 3000
    token_retransmits_before_loss_const: 10
    join: 60
    consensus: 3600
    vsftype: none
    max_messages: 20
    clear_node_high_bit: yes
    secauth: off
   threads: 0
    rrp_mode: active
    transport: udpu
    interface {
                    ringnumber: 0
                    bindnetaddr: 10.2.110.0
                    mcastport: 5405
                    ttl: 1
                    member {
                            memberaddr: 10.2.110.1
                    }
                    member {
                            memberaddr: 10.2.110.2
                    }
                    member {
                            memberaddr: 10.2.110.3
                    }
                    member {
                            memberaddr: 10.2.110.4
                    }
     }
}
amf {
    mode: disabled
}
service {
    ver:      0
    name:     pacemaker
}
aisexec {
    user:   root
    group:  root
}
logging {
    syslog_priority: warning
    fileline: off
    to_stderr: no
    to_logfile: yes
    logfile: /var/log/corosync/corosync.log
    logfile_priority: notice
    to_syslog: no
    syslog_facility: daemon
    debug: off
    timestamp: on
    logger_subsys {
        subsys: AMF
        debug: off
        tags: enter|leave|trace1|trace2|trace3|trace4|trace6
    }
}
```

В конфигурационном файле прописаны IP-адреса для всех четырех серверов (member).

Так как в кластере несколько серверов со своими IP-адресами (bond), понадобится ресурс агент [8–10], отвечающий за (общий) виртуальный IP-адрес, который будет активен только на сервере, находящемся в активном режиме.

Сервис drbd работает по принципу master/slave. На активном сервере сервис drbd должен находиться в режиме master, а на сервере горячего резерва – в режиме slave. Для этого необходимо создать ресурс агент, чтобы серверы при переключении с разных режимов активности находились в соответствующих режимах.

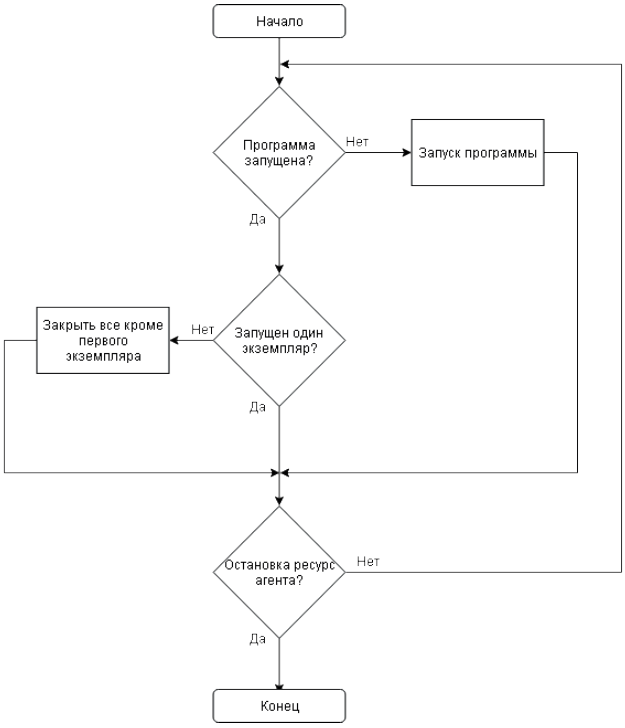

Рис. 2. Блок-схема работы ресурс агента programmControl

Для контроля программ нужно написать ресурс агент (programmControl). Агент должен отслеживать наличие программ в списке процессов ОС. Если программа не запущена, он должен ее запустить. Если запущено две и более копии программы, то он должен завершить при необходимости лишние копии. Алгоритм работы агента представлен на рис. 2.

Для отслеживания работоспособности кластера следует создать ресурс агент, который будет отслеживать количество серверов в кластере (wakeUp). В нормальном состоянии должно работать два сервера – активный и горячий резерв. В случае отключения одного из работающих серверов ресурс агент должен послать магический пакет на сервер в холодном резерве. При случайном включении третьего сервера агент должен его выключить, послав по ssh команду на выключение. Алгоритм работы агента представлен на рис. 3.

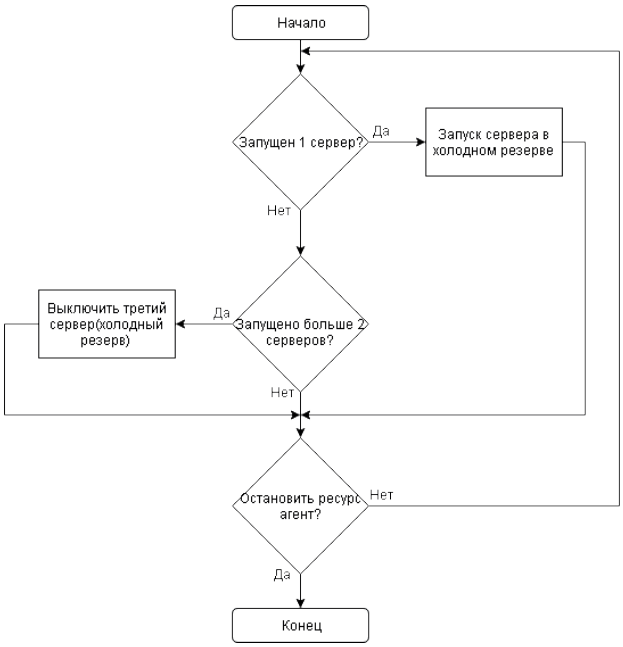

Рис. 3. Блок-схема работы ресурс агента wakeUp

Настройка конфигурации кластера

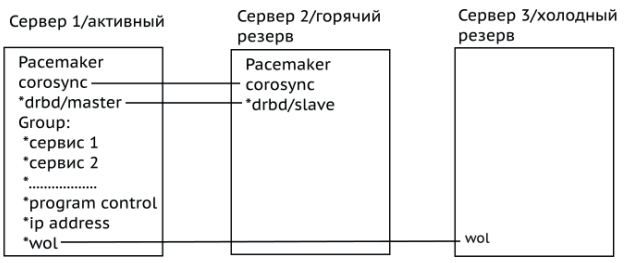

Рис. 4. Начальное состояние кластера

Все ресурс агенты будут находиться в одной группе для избегания ситуации, когда один из ресурс агентов случайно мигрировал на другой сервер, а остальные так и остались на прежнем сервере. На рис. 4 показано состояние серверов в начале работы. После отключения первого сервера состояние кластера изменится согласно рис. 5. На рис. 6 приведена схема работы кластера.

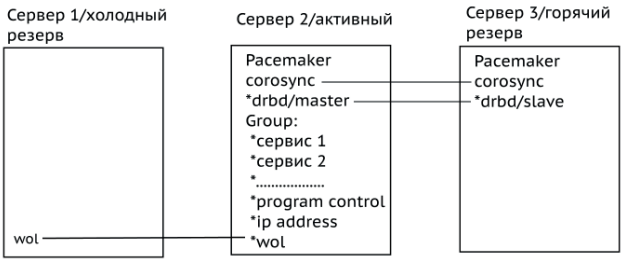

Рис. 5. Состояние кластера после отключения первого сервера

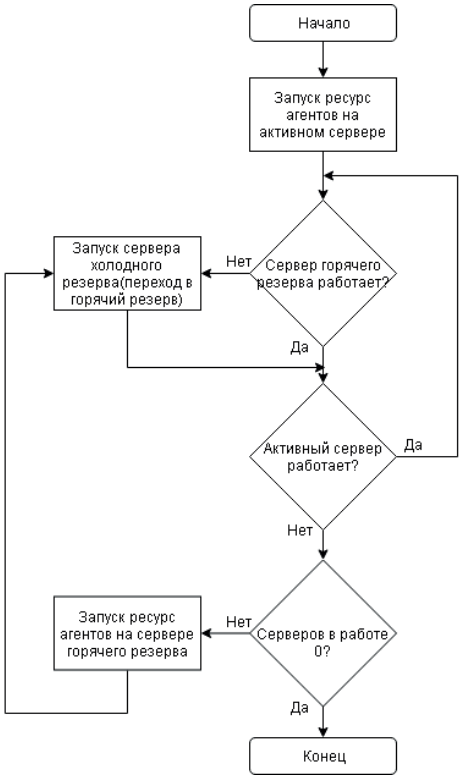

Рис. 6. Схема работы кластера

Заключение

Высокая доступность информационных сервисов, предоставляемых узлами кластера, обеспечивается кластерным ПО и с помощью специальных сервисов или скриптов, отслеживающих работоспособность информационных сервисов и выполняемых узлами кластера. В случае сбоя, вызванного отказом диска, сетевого интерфейса или самого приложения, кластерное ПО переносит соответствующий сервис на другой узел. Под переносом здесь понимается следующее – остановка приложения (если оно еще работало) на активном сервере, размонтирование общих дисковых томов, монтирование их на втором узле, перенос

IP-адреса с активного на резервный сервер, запуск приложения. Если в кластере больше двух серверов, то информационные сервисы вышедшего из строя сервера переносятся на другой сервер по правилам, заданным кластерным ПО на основе данных о работоспособных серверов.

Список литературы

1. Кластерные системы для приложений высокой готовности [Электронный ресурс]. URL: https://www.bytemag.ru/articles/detail.php?ID=6326 (дата обращения: 01.03.2018).
2. Подробное описание функционирования кластера высокой надежности (доступности) High-availability cluster (ha cluster) [Электронный ресурс]. URL: http://www.xnets.ru/plugins/content/content.php?content.69 (дата обращения: 01.03.2018).
3. Кластеры на ОС Linux как системы высокой доступности [Электронный ресурс]. URL: http://old.ci.ru/inform10_99/p_08_9.htm (дата обращения: 01.03.2018).
4. Кластеры высокой доступности [Электронный ресурс]. URL: https://studfiles.net/preview/1511462/page:2/ (дата обращения: 03.03.2018).
5. Linux bonding – объединение сетевых интерфейсов в Linux [Электронный ресурс]. URL: http://www.adminia.ru/linux-bonding-obiedinenie-setevyih-interfeysov-v-linux/ (дата обращения: 03.03.2018).
6. Строим кластер на связке DRBD+Pacemaker+OpenVZ+NFS+Zabbix [Электронный ресурс]. URL: http://borodatych.blogspot.ru/2011/02/drbdpacemakeropenvznfszabbix.html (дата обращения: 10.03.2018).
7. ClusterLabs [Электронный ресурс]. URL: http://clusterlabs.org (дата обращения: 05.03.2018).
8. OCF Resource Agents [Электронный ресурс]. URL: http://www.linux-ha.org/wiki/OCF_Resource_Agent (дата обращения: 05.03.2018).
9. OCF Resource Agent Developer's Guide [Электронный ресурс]. URL: https://www.linbit.com/en/resources/documentation/526-ocf-resource-agent-developers-guide/ (дата обращения: 05.03.2018).
10. Heartbeat Resource Agents [Электронный ресурс]. URL: http://www.linux-ha.org/wiki/Heartbeat_Resource_Agents (дата обращения: 05.03.2018).

История статьи
Поступила в редакцию 17 апреля 2018 г.
Принята к публикации 21 мая 2018 г.

ORGANIZATION OF A HIGH-AVAILABILITY CLUSTER

N. D. Ivanov
Siberian Federal University, Krasnoyarsk, Russian Federation

Clustering of server systems is one of the ways to ensure the reliability of data processing and storage systems, as well as constant provision of users with prompt and reliable information. With the clustering of server systems, a high level of availability is maintained.
The article presents the construction of a high availability cluster. Several servers are used to increase the fault tolerance, which are connected through several switches. Several network interfaces are provided in servers using the technology of bonding. Also, servers provide a RAID array. Servers and switches are connected to uninterruptible power supplies. Uninterruptible power supplies are backed up.
The software packages are presented with their configuration files. Software packages are necessary for the functioning of the cluster: bond, drbd, pacemaker, corosync. The pacemaker configuration and agent resource are described. Those agents was described: virtual IP address - for a cluster to have a single address, drbd for mirroring RAID arrays, programmControl for managing applications, wakeUp for controlling servers turning on and off. As a result, a high availability cluster was designed. It is reserved by the server part, network part and memory. Customer service is interrupted for a very short time when moving from one server to another.

Keywords: high availability cluster, resource agent, pacemaker, corosync, drbd, bound, WOL.

References

1. Cluster systems for high availability applications. Available at: https://www.bytemag.ru/articles/detail.php?ID=6326 (accessed 01.03.2018).

2. Detailed description of the functioning of the high-availability cluster High-availability cluster (ha cluster). Available at: http://www.xnets.ru/plugins/content/content.php?content.69 (accessed 01.03.2018).
3. Clusters on Linux OS as high-availability systems. Available at: http://old.ci.ru/inform10_99/p_08_9.htm(accessed 01.03.2018).
4. Clusters of high availability. Available at: https://studfiles.net/preview/1511462/page:2/ (accessed 03.03.2018).
5. Linux bonding - integration of network interfaces in Linux. Available at: http://www.adminia.ru/linux-bonding-obiedinenie-setevyih-interfeysov-v-linux/ (accessed 03.03.2018).
6. We build a cluster on a bunch DRBD + Pacemaker + OpenVZ + NFS + Zabbix. Available at: http://borodatych.blogspot.ru/2011/02/drbdpacemakeropenvznfszabbix.html (accessed 10.03.2018).
7. ClusterLabs. Available at: http://clusterlabs.org/ (accessed 05.03.2018).
8. OCF Resource Agents. Available at: http://www.linux-ha.org/wiki/OCF_Resource_Agent (accessed 05.03.2018).
9. OCF Resource Agent Developer's Guide. Available at: https://www.linbit.com/en/resources/documentation/526-ocf-resource-agent-developers-guide/ (accessed 05.03.2018).
10. Heartbeat Resource Agents. Available at: http://www.linux-ha.org/wiki/Heartbeat_Resource_Agents (accessed 05.03.2018).

Article history
Received 17 April 2018
Accepted 21 May 2018