

КОНФИГУРИРОВАНИЕ МАСШТАБИРУЕМОГО СЕРВИСА В КЛАСТЕРНОМ ОКРУЖЕНИИ

Берилген макалада Apache веб-серверинин мисалында жогору жеткиликтүү маштабдаштырылган scalable кластерди түзүү процедурасы жазылган.

В данной статье описана процедура создания и конфигурирования высокодоступного масштабируемого(scalable) кластера с балансировкой нагрузки на примере веб-сервера Apache.

This article describes how to create and configure a highly available, scalable (scalable) cluster with load balancing on the example web server Apache.

Масштабируемый (Scalable) кластер, это высокодоступный кластер, размещающий в себе сервисы которые работают одновременно на всех нодах кластера и по мере возрастания нагрузки на какой либо сервис, возможно наращивание аппаратной части без остановки сервиса. Т.е. если у нас уже есть 2-нодный кластер и нам существенно не хватает аппаратных ресурсов, мы можем добавить в кластерное окружение еще один сервер и сбалансировать нагрузку на все ноды без остановки сервиса.

Помимо высокой доступности, в Sun Cluster имеется и функция балансировки нагрузки. Эта кластерная функция позволяет настраивать одновременную работу приложений или служб данных для выполнения на нескольких системах одновременно для существующего оборудования кластера. Для обеспечения балансировки нагрузки на основе IP не требуется установки дополнительного оборудования.

Большинство приложений, которые могут быть высокодоступными, также могут быть настроены как масштабируемые(scalable) с учетом нескольких дополнительных требований, описанных в данном документе. Если ваше приложение соответствует указанным выше требованиям, с помощью нескольких простых этапов можно настроить масштабируемое приложение.

Краткое описание задачи

Архитектура масштабируемых сервисов позволяет некоторым сервисам, таким как веб-сервер Apache работать одновременно на нескольких нодах кластера при этом для клиентской стороны все будет представляться так, как будто все работает на одном сервере.

Клиенты подключаются к таким сервисам через один IP-адрес, используя механизм Shared address, который является глобальным для всех нод кластера. На рисунке 1 приведена схема масштабируемого кластера.

Так же как и отказоустойчивые сервисы, все данные для масштабируемых сервисов должны храниться на общем хранилище, подключенном ко всем нодам кластера. При этом файловые системы должны быть сконфигурированы как глобальные(Global), эта технология позволяет монтировать файловые системы одновременно на все ноды кластера.

Одним из главных барьеров внедрения любого приложения в кластер как масштабируемый сервис является отсутствие механизма блокировки файлов. В Sun Cluster приложение, которое выполняет изменение данных без всяких механизмов блокировки и синхронизации файлов не может быть настроено как масштабируемый сервис.

Веб-сервера разработаны так, что позволяют делать записи в файлы по отдельным сессиям, и для каждой сессии используется блокировка файлов, которая не допускает

одновременного изменения файлов несколькими клиентами. Соответственно такое приложение может быть сконфигурировано как масштабируемый сервис.

Конфигурирование

У нас уже есть настроенный классический двухнодный кластер состоящий из двух машин(Sun Sparc Enterprize M5000) и двух дисковых массивов HP P9500. Два разноудаленных офиса в черте одного города (протяженность около 10км) и оптический канал между офисами . Смотрите схему кластера на Рис. 2.

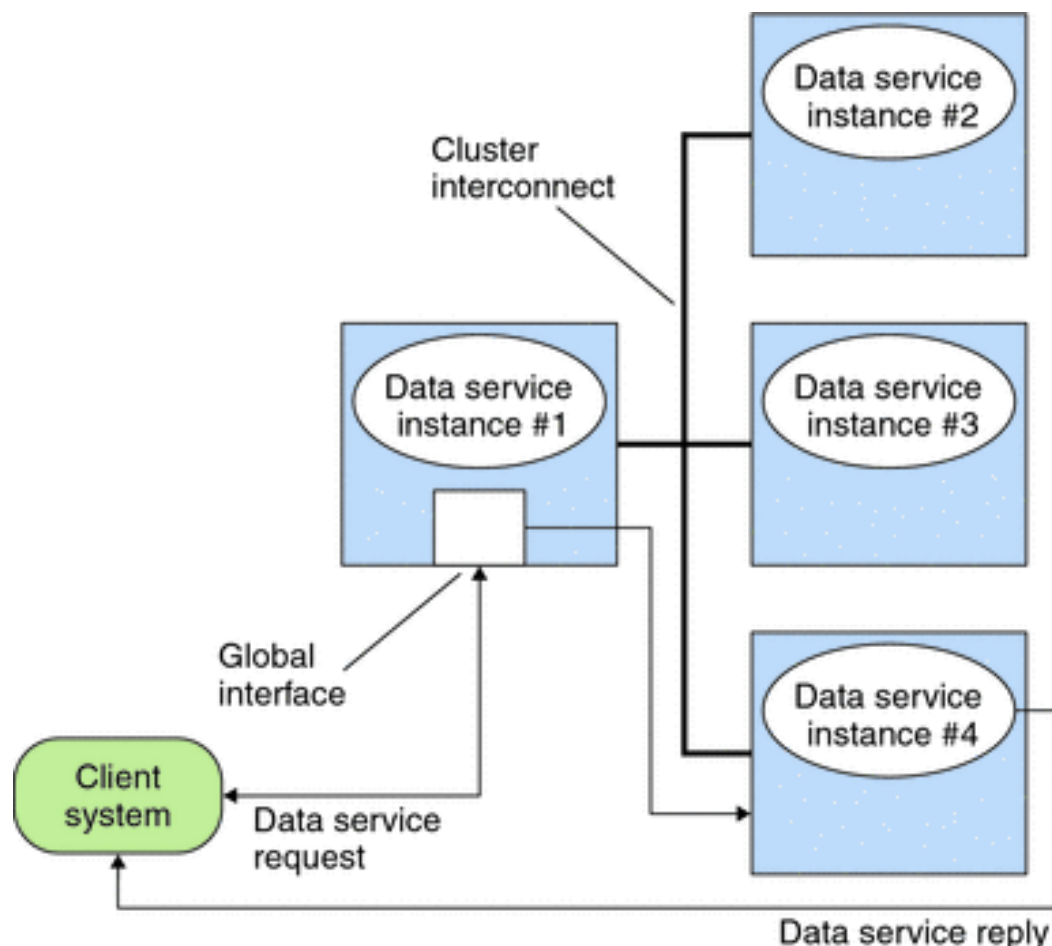


Рис. 1.Схема масштабируемого кластера

На нодах кластера установлена операционная система Solaris 10 Sparc. Так же установлен и настроен Veritas Volume Manager, используя который мы уже сконфигурировали дисковые группы, создали файловые системы, которые в дальнейшем будут монтироваться как глобальные, одновременно на все ноды.

В дальнейшем мы будем пользоваться веб-сервером Apache предустановленным в дистрибутив операционной системы Solaris.

У нас будет одна глобальная файловая система, которую мы будем монтировать одновременно на обе ноды средствами кластера как /global/web:

Файловая система VxVM	Точка монтирования
-----	-----
/dev/vx/dsk/web-dg/apachevol	/global/web

1. Регистрация типов ресурсов Apache и HAStoragePlus
clrt register SUNW.apache

```
# clrt register SUNW.HAStoragePlus
2. Создаем ресурсную группу для механизма SharedAddress:
# clrg create -n node1,node2 sa-rg
3. Создаем ресурс SharedAddress:
# clrssa create -g sa-rg apache-lh
4. Создаем масштабируемую ресурсную группу:
# clrg create -n node1,node2 \
-p Desired primaries=2 -p Maximum primaries=2 \
```

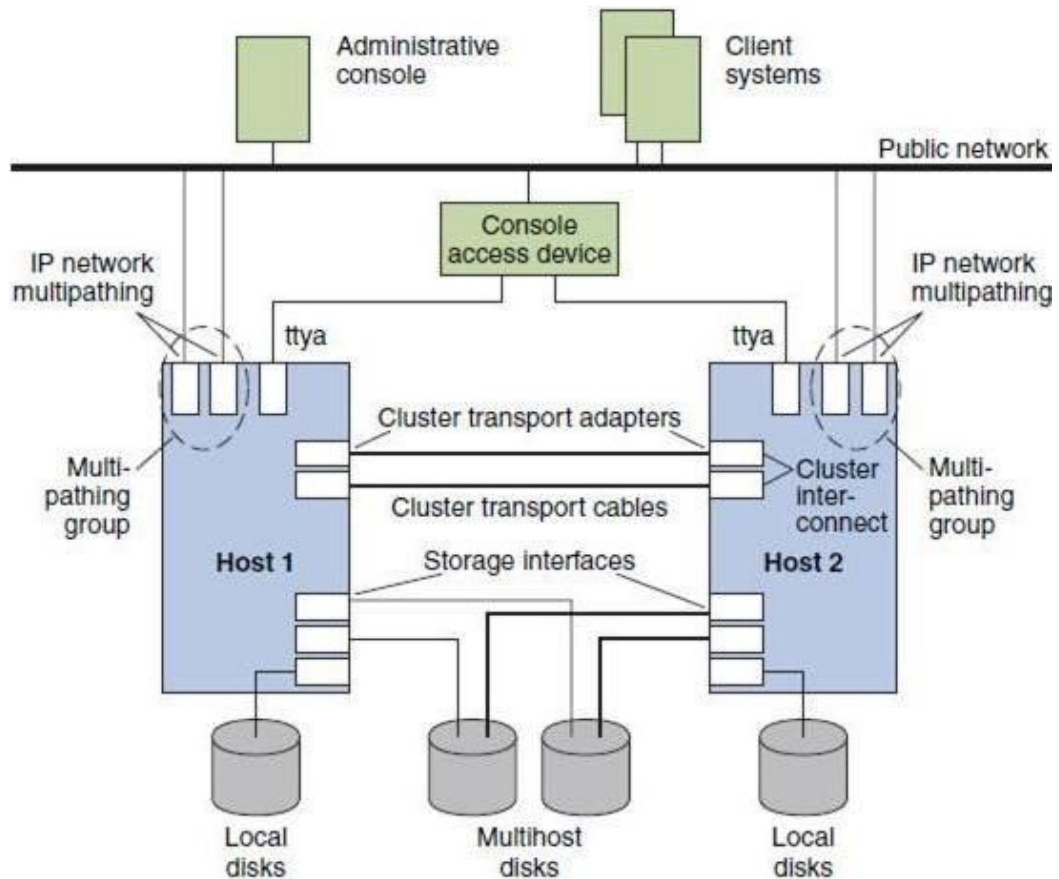


Рис. 2. Схема кластера

```
web-rg
5. Создаем ресурс HAStoragePlus:
# clrs create -t SUNW.HAStoragePlus -g web-rg \
-p FilesystemMountPoints=/global/web \
-p AffinityOn=false web-stor
6. Создаем ресурс для веб-сервера Apache:
# clrs create -t SUNW.apache -g web-rg \
-p Resource_dependencies=web-stor,apache-lh \
-p Bin_dir=/global/web/bin \
-p Port_list=80/tcp \
-p Load_balancing_policy=LB_STICKY
-p Scalable=TRUE apache-res
7. Переводим ресурсные группы в оперативный режим
# clrg online sa-rg
# clrg online apache-rg
```

Ниже приведены описания нескольких важных свойств.

- Scalable: TRUE указывает на масштабируемость ресурса, позволяя сетевой платформе выполнять балансировку нагрузки и запускать сервер Apache
- Port_list: список прослушиваемых сервером номеров портов.
- Load_balancing_policy: это значение используется для определения способа распределения нагрузки между узлами. Доступные варианты значения:
 - *LB_WEIGHTED*: нагрузка распределяется на основе весовых коэффициентов, используя свойство Load_balancing_weights.
 - *LB_STICKY*: нагрузка распределяется на основе IP-адреса клиента. В этом случае во время настройки известен набор портов, а все запросы от одного IP-адреса клиента распределяются на один и тот же узел, прослушивающий данный предварительно определенный порт.
 - *LB_STICKY_WILD*: нагрузка распределяется на основе IP-адреса клиента, и в этом случае порты заранее не известны и назначаются динамически. Все запросы от одного клиента будут переданы на один и тот же узел независимо от номера порта, для которого предназначается данный IP-адрес.

Подробное описание указанных выше свойств и иных свойств масштабирования приведено в соответствующей map странице

Проверка статуса созданных ресурсов

```
# clrg status
```

```
Cluster Resource Groups
```

```
=====
```

Group Name	Node Name	Suspended	Status
-----	-----	-----	-----
sa-rg	node1	No	Online
	node2	No	Offline
web-rg	node1	No	Online
	node2	No	Online

```
# clrs status -g sa-rg,web-rg
```

```
Cluster Resources
```

```
=====
```

Resource Name	Node Name	State	Status Message
-----	-----	-----	-----
orangecat-web	node1	Online	Online - SharedAddress online.
	node2	Offline	Offline
web-stor	node1	Online	Online
	node2	Online	Online
apache-res	node1	Online	Online - Service is online
	node2	Online	Online - Service is online.

Orangecat-web это ресурс типа SharedAddress на который назначен IP-адрес и все клиентские запросы попадают на него. Далее этот ресурс распределяет эти запросы по разным нодам кластера.

Web-stor ресурс типа SUNW.HAStoragePlus который отвечает за монтирование файловой системы, которая монтируется как /global/web

Apache-res это ресурс типа SUNW.apache который запускает сервер Apache на обоих нодах кластера.

Наш кластер готов к обслуживанию клиентских запросов. Схему масштабируемого кластера смотрите на Рис. 3.

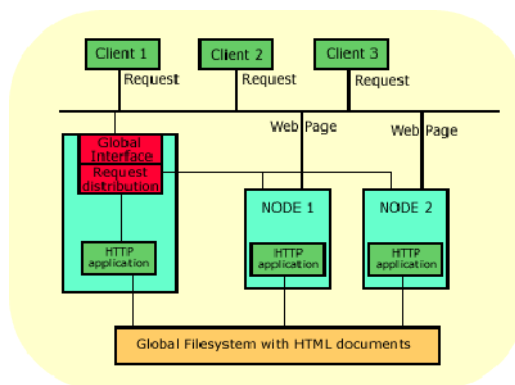


Рис. 3. Схема масштабируемого кластера

Из показанных выводов команд можно увидеть, что отказоустойчивые(Failover) ресурсные группы запущены только на одной ноде, тогда как масштабируемые(Scalable) запущены на всех нодах кластера.

Список литературы

- 1.Oracle Solaris Cluster Software Installation Guide. September 2013, E37727–04
- 2.Oracle Solaris Cluster Data Services Planning and Administration Guide. March 2013, E37725–01
3. Oracle Solaris Cluster System Administration Guide. March 2013, E37724–01
4. Veritas™ Volume Manager Administrator's Guide 5.1 – Solaris. August 2010
5. <http://docs.oracle.com>