

Оглавление

1	Кластерная обработка	1
1.1	<i>Сервисы и функции программного обеспечения кластеров:</i>	1
1.2	<i>Свойства кластеров</i>	2
1.3	<i>Масштабирование кластеров</i>	3
1.4	<i>Метода кластеризации</i>	4
1.5	<i>Обработка отказов в кластерах</i>	5
1.6	<i>Балансировка загрузки</i>	6
1.7	<i>Типа кластеров</i>	6
1.7.1	Кластеры серверов.....	7
1.7.2	NLB-кластеры.....	7
1.8	<i>Сравнение кластеров и SMP</i>	8

1 Кластерная обработка

Кластеризация представляет собой альтернативу симметричной многопроцессорной обработке в качестве подхода с высокой производительностью и доступностью

Кластер - группа взаимосвязанных совместно работающих компьютеров, представляющих собой **единый ресурс** и способных создавать иллюзию работы единой вычислительной системы.

Компьютеры, составляющие кластер, обычно называют **узлами (node)**.

Все компьютеры кластера могут работать самостоятельно, отдельно от кластера.

1.1 Сервисы и функции программного обеспечения кластеров:

1. **Единая точка входа.** Пользователь входит в кластер в целом, а не в один из его компьютеров.
2. **Единая иерархия файлов.** Пользователь видит единую иерархию каталогов файлов в одном корневом каталоге.
3. **Единая точка управления.** Имеется рабочая станция, по умолчанию используемая для слежения за кластером и управления им.
4. **Единая виртуальная сеть.** Узел кластера может обратиться к любому другому узлу, даже если реально кластер состоит из ряда взаимосвязанных сетей.
5. **Единое пространство памяти.** Распределенная совместно используемая память позволяет программам разделять переменные.

6. **Единая система управления заданиями.** Пользователь может передать задание планировщику кластера, не определяя узел, на котором это задание должно выполняться.
7. **Единый пользовательский интерфейс.** Все пользователи независимо от используемых рабочих станций получают доступ к кластеру посредством одинакового графического интерфейса пользователя.
8. **Единое пространство ввода-вывода.** Любой узел может получить удаленный доступ к любому периферийному или дисковому устройству, даже не зная его физического расположения.
9. **Единое пространство процессов.** Используется единая схема идентификации процессов. Процесс на одном узле может создавать процесс на удаленном узле или взаимодействовать с любым другим процессом на нем.
10. **Контрольные точки.** Эта функция периодически сохраняет состояние процессов и промежуточные результаты вычислений, с тем чтобы обеспечить восстановление системы при сбое.
11. **Миграция процессов.** Эта функция обеспечивает балансировку загрузки.

Сервисы и функции, указанные в последних четырех пунктах данного списка, увеличивают доступность кластера; указанное в остальных пунктах связано с предоставлением пользователю единого образа системы.

Кластер включает в себя также программное обеспечение, отвечающее за эффективную работу программ, способных к параллельному выполнению.

1.2 Свойства кластеров.

- **Абсолютная масштабируемость.** Можно создавать кластеры любых размеров, которые превысят по суммарной производительности любой самый мощный компьютер. Кластер может состоять из десятков или сотен машин, каждая из которых может быть многопроцессорной системой.

- **Инкрементальная масштабируемость.** Кластер конфигурируется таким образом, чтобы можно было добавлять новые системы малыми

порциями. Поэтому пользователь может начать с небольшого кластера, постепенно наращивая его мощность без кардинальной замены компьютеров.

- **Высокая доступность.** Поскольку каждая машина кластера представляет собой отдельный компьютер, сбой в работе или выход из строя одного узла не приводят к снижению уровня обслуживания. Во многих системах такие отказы автоматически обрабатываются аппаратным и программным обеспечением.

- **Отношение цена/производительность.** Использование недорогих компьютеров в качестве узлов кластера позволяет создать высокопроизводительную систему, стоимость которой меньше, чем стоимость компьютера с такой же мощностью.

1.3 Масштабирование кластеров

Это наращивание производительности кластера по мере роста потребностей.

Существуют два основных метода масштабирования кластеров:

- **вертикальное масштабирование (scaling up)** - это повышение производительности отдельных серверов путем изменения аппаратной конфигурации. Примеры вертикального масштабирования - наращивание оперативной памяти или кэш-памяти второго уровня, установка более производительного процессора и дополнительных процессоров. Так можно повышать производительность серверов в кластерах любой конфигурации, но это нужно делать с учетом производительности остальных узлов кластера. Например, наращивание производительности только активных узлов кластера серверов приведет к тому, что пассивные узлы не достигнут такого же уровня производительности, случись им заменить активные узлы. Чтобы обеспечить оптимальный уровень производительности в любых условиях, необходимо сбалансированное наращивание производительности отдельных серверов кластера;

горизонтальное масштабирование (scaling out) — это добавление серверов к существующему кластеру. Если нагрузка по исполнению

приложения распределяется между несколькими серверами, добавление к кластеру дополнительных серверов означает снижение нагрузки на отдельный сервер. **Определение необходимого числа кластеров**

Если требуется обеспечить высокую доступность для нескольких приложений, встает вопрос о необходимом числе кластеров. Кластер позволяет выполнять несколько приложений, поэтому можно разместить все приложения на одном кластере, создать для каждого приложения отдельный кластер, либо комбинировать эти подходы.

1.4 Метода кластеризации

- Кластеры с высокой производительностью
- Кластеры с высоким коэффициентом готовности
- Кластеры со сбалансированной нагрузкой

Использование дискового пространства кластеров:

- **отдельные серверы,**
- **серверы без совместного использования дисков,**
- **серверы с совместным использованием дисков.**

Первый подход характеризуется тем, что каждый компьютер представляет собой отдельный сервер со своими собственными дисками; при этом совместно используемых дисков в системе нет. Такой подход обеспечивает как высокую производительность, так и высокую доступность системы.

В данном случае требуется наличие дополнительного планирующего программного обеспечения для назначения серверов обработки запросов клиентов, которое обеспечивает сбалансированность загрузки и высокую степень использования серверов.

Требуется также обеспечить обработку ситуации выхода из строя одного из серверов, с тем, чтобы при сбое одного из компьютеров работающее с ним приложение продолжало корректно выполняться. Для того чтобы это было возможно, требуется постоянное копирование данных между серверами с

целью обеспечения каждой системе доступа к текущим данным других систем.

Высокая доступность достигается за счет увеличения накладных расходов, вызванных таким обменом данными, и некоторого снижения производительности.

Уменьшить эти накладные расходы позволяет **подключение к общим дискам** серверов, составляющих кластеры.

- При выборе одного из вариантов такого подхода — **без совместного использования (shared nothing)** — общие диски разбиты на **тома**, и каждый том принадлежит одному из узлов. Если этот узел выходит из строя, кластер должен быть переконфигурирован так, чтобы какой-то из оставшихся компьютеров стал владельцем диска вышедшего из строя компьютера.

- Возможно и решение, когда несколько компьютеров совместно используют одни и те же диски (**подход с совместным использованием дисков**), так что каждый из узлов имеет доступ ко всем томам всех дисков. Этот подход требует наличия средств блокировки для обеспечения взаимного исключения доступа компьютеров к данным.

1.5 Обработка отказов в кластерах

Способ обработки кластером отказов зависит от используемого метода кластеризации.

Обычно используется один из двух подходов:

- **высокодоступные кластеры**
- **отказоустойчивые кластеры.**

Высокодоступный кластер предусматривает высокую вероятность того, что все ресурсы будут доступны. При отказе, таком, как сбой системы или потеря тома диска, текущий запрос оказывается потерянным. При повторе запроса он будет отправлен другой машине кластера. Однако операционная система кластера не дает никаких гарантий относительно состояния частично выполненных транзакций, которые должны обрабатываться на уровне приложений.

Отказоустойчивые кластеры гарантируют постоянную доступность всех ресурсов. Это достигается за счет использования избыточных совместно используемых дисков и механизмов для восстановления как невыполненных, так и завершенных транзакций.

Функция переключения приложений и передачи ресурсов данных от сбойной системы другой системе кластера известна как преодоление сбоя (**failover**).

С ней связана функция восстановления приложений и ресурсов данных возвращенной в строй после отказа вычислительной системы, известная как восстановление после сбоя (**failback**).

Эта функция может быть автоматизирована, но желательно, чтобы она выполнялась только после полного устранения причин сбоя. Когда повторный сбой маловероятен. Иначе кластер может попасть в состояние, когда после восстановления вновь происходит сбой и все-ресурсы вновь перемещаются на другой компьютер, чтобы после возврата в строй только что отказавшей системы все повторилось вновь.

1.6 Балансировка загрузки

Кластер нуждается в эффективных средствах для балансировки загрузки работающих компьютеров (сюда же входит и требование инкрементального масштабирования).

Когда к кластеру добавляется новый компьютер, средства балансировки загрузки должны подключить эту машину к **программе-планировщику**.

Для распознавания появления доступных сервисов у разных членов кластера и возможности их перемещения с одной системы на другую требуется использование механизма **промежуточного программного обеспечения**.

1.7 Типа кластеров

- **кластеры серверов (server clusters)**
- **кластеры сбалансированной нагрузки сети (Network Load**

Balancing clusters),**1.7.1 Кластеры серверов**

Кластеры серверов предназначены для приложений, которые в течение долгого времени хранят в памяти сервера **большие объемы часто изменяющихся данных**, отражающих состояние приложения.

Такие приложения называются **приложениями с состоянием (stateful applications)**, их примером может служить

SQL Server,

почтовые серверы,

серверы сообщений типа Microsoft Exchange,

также серверы файлов и печати.

Все узлы кластера серверов подключены к **общему хранилищу данных через шину SCSI или сеть областей хранения (Storage Area Network, SAN)**.

Всем узлам доступны одни и те же прикладные данные, поэтому любой узел в любое время может обработать клиентский запрос.

Каждый узел кластера должен быть настроен как **активный** либо **пассивный**.

Активный узел принимает и обрабатывает запросы клиентов, а **пассивный** играет роль резерва на случай отказа активного узла.

1.7.2 NLB-кластеры

Другой тип кластеров использует механизм балансировки сетевой нагрузки (Network Load Balancing, NLB) и обеспечивает в дополнение к высокой доступности и надежности **высокую масштабируемость**.

Кластеры сбалансированной нагрузки сети или NLB-кластеры предназначены для приложений, работающих со **сравнительно небольшими наборами редко меняющихся или неизменяемых данных**, а также для приложений, которые не хранят свое состояние в памяти в течение длительного времени.

Такие приложения называются **приложениями без состояния (stateless applications)**, типичными примерами являются

Web-, FTP- и VPN-серверы.

Каждый клиентский запрос приложение без состояния выполняет как отдельную транзакцию, поэтому для балансировки сетевой нагрузки запросы распределяются между серверами.

Серверы - узлы NLB-кластера не подключены к общему хранилищу данных, как узлы обычного кластера, а имеют идентичные копии данных.

Все узлы NLB-кластера являются активными.

1.8 Сравнение кластеров и SMP

- Как кластеры, так и симметричная многопроцессорная обработка обеспечивают конфигурацию с несколькими процессорами для поддержки приложений с высокими требованиями. Оба решения коммерчески доступны, хотя SMP применяется несколько дольше.

- Основное достоинство SMP состоит в том, что симметричной многопроцессорностью проще управлять и что ее легче настроить, чем кластер. SMP ближе, чем кластер, к обычной однопроцессорной модели, для которой и разработаны практически все приложения.

При переходе от однопроцессорной системы к SMP требуется единственное принципиальное изменение — изменение **функции планирования**. Другим достоинством SMP является то, что такая система занимает меньше места и потребляет меньше электроэнергии, чем сравнимый по мощности кластер. И к тому же этот подход хорошо зарекомендовал себя, тщательно изучен и стабилен в работе.

О достоинствах кластеров свидетельствует то, что именно эта технология уже длительное время доминирует на рынке высокопроизводительных серверов. Кластеры существенно превосходят SMP с точки зрения инкрементальной и абсолютной масштабируемости, а также доступности, поскольку легко обеспечивается избыточность всех элементов системы.