нетология

Отказоустойчивость: **Кластеризация**

Александр
Зубарев

Александр Зубарев

Председатель цикловой комиссии “Информационной безопасности инфокоммуникационных систем”

АКТ (ф) СПбГУТ

Александр Зубарев

Предисловие

Сегодня мы:

- поговорим о технологиях и методах построения кластерных систем;
- рассмотрим различные типы кластерных систем;
- узнаем, где применяются кластерные системы и как используются в реальной жизни.

План занятия

1. Введение
2. Кластеры
3. Отказоустойчивые кластеры
4. Вычислительные кластеры
5. Системы распределенных вычислений
6. Stonith/Fencing
7. Итоги
8. Домашнее задание

Введение

Современный мир и кластеры

В современном мире практически любое производство использует кластерные технологии, так как современные информационные потоки требуют:

- большой вычислительной мощности,
- большого хранилища данных,
- отказоустойчивости,
- надежности и безопасности.

Кластера помогают выполнять выдвигаемые производством задачи.

Кластеры

Кластер

Кластер — группа компьютеров, объединённых высокоскоростными каналами связи, представляющая с точки зрения пользователя единый аппаратный ресурс.

Группы компьютеров можно рассматривать с точки зрения вычислительных систем:

- SISD (Single Instruction, Single Data),
- SIMD (Single instruction, Multiple data),
- MISD (Multiple Instruction stream, Single Data stream),
- MIMD (Multiple Instruction stream, Multiple Data stream).

Кластер

Другими словами, это разновидность параллельной или распределенной системы, которая:

- состоит из нескольких связанных между собой компьютеров;
- используется как единый, унифицированный компьютерный ресурс.

Кластерные системы делятся на следующие архитектуры вычислительных систем:

- SMP (Symmetric Multiprocessing) – сильно связанные,
- MPP (Massive Parallel Processing)– слабо связанные,
- NUMA (Non-Uniform Memory Access) – сильно связанные.

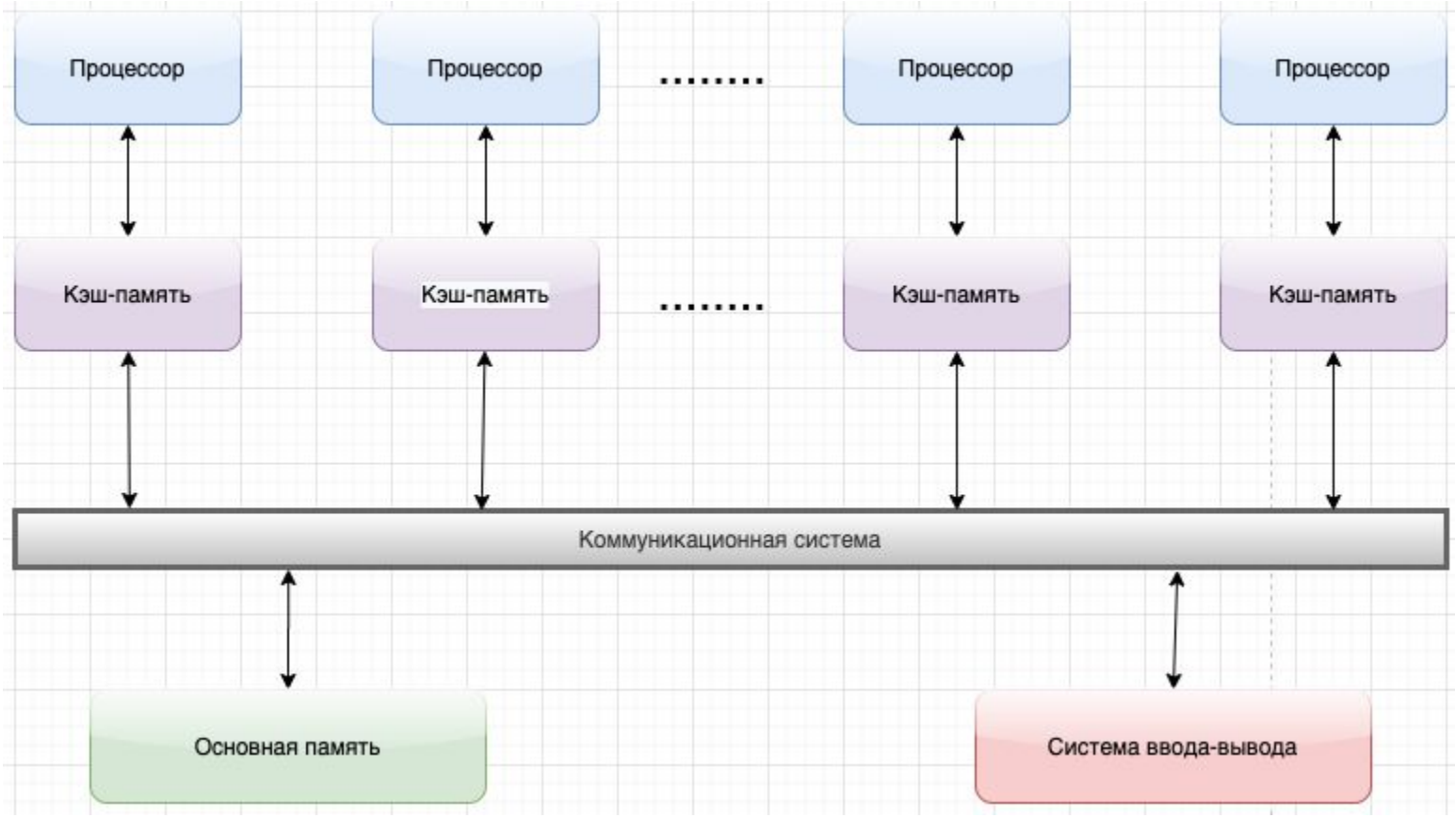
Сильно связанные системы

Сильно связанная система состоит из нескольких однородных процессоров и массива общей памяти (обычно из нескольких независимых блоков).

Представители таких систем:

- SMP (симметричные мультипроцессоры),
- NUMA (системы с неоднородным доступом к памяти).

Архитектура SMP-системы



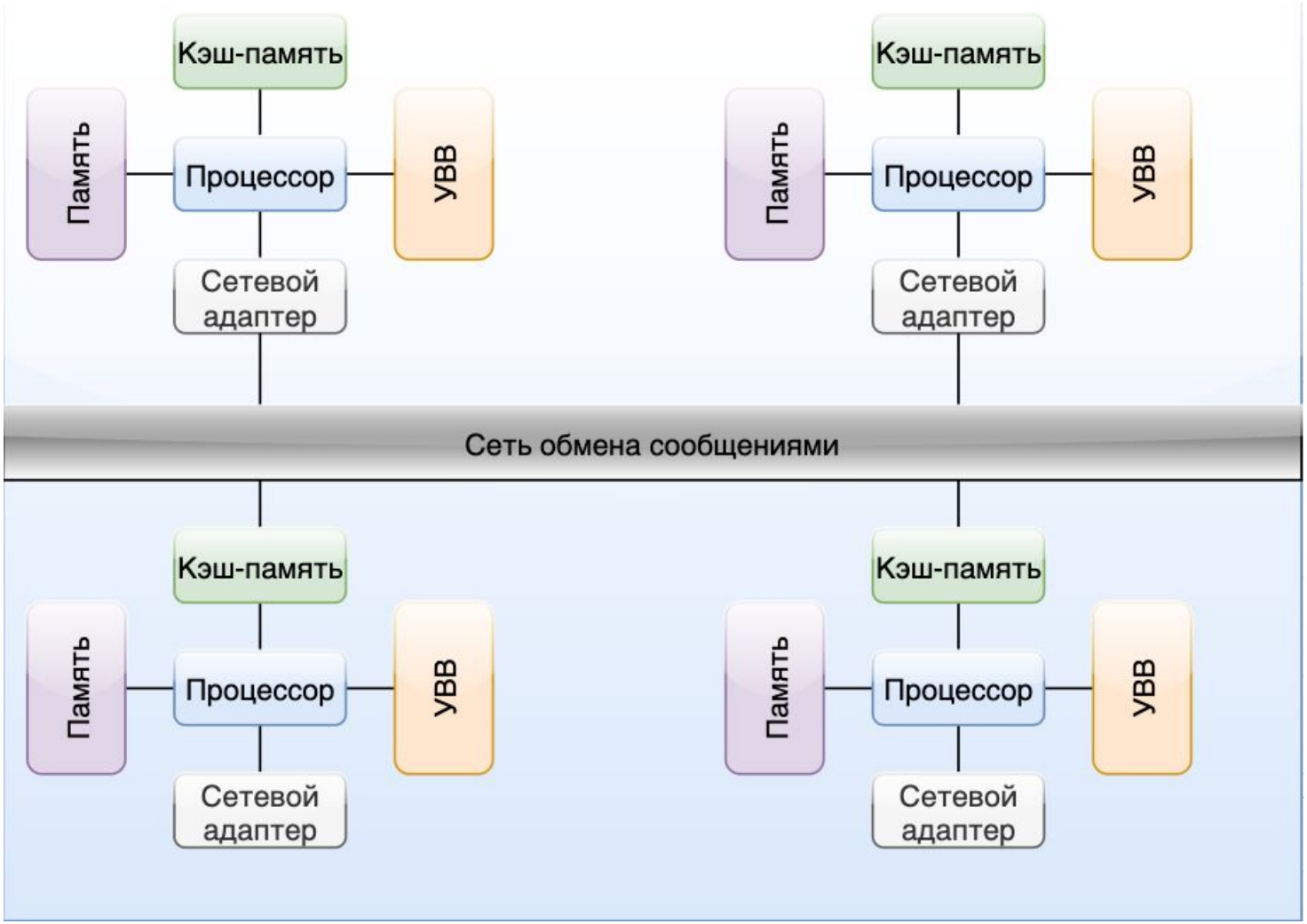
Слабо связанные системы

В слабо связанных системах вся память распределена между процессорными элементами.

Пример таких систем:

- системы с массовым параллелизмом (MPP),
- кластерные системы.

Архитектура MPP-системы



Кластерные вычислительные системы (ВС)

Кластерные вычислительные системы:

- формируют единый вычислительный ресурс,
- создают иллюзию наличия единственной вычислительной машины.

Каждый узел способен работать автономно. Узел может быть SMP, MPP или однопроцессорной ВМ.

Перед кластерами ставятся две задачи:

- достичь большой вычислительной мощности,
- обеспечить повышенную надежность ВС.

Работа кластерной системы

На уровне аппаратуры кластер — это сеть независимых ВС.

При соединении машин в кластер почти всегда поддерживаются прямые межмашинные связи.

Для контроля работоспособности узлов существует специальный сигнал — **heartbeat**.

Кластерная система обеспечивает бесперебойную работу при отказе одного или нескольких узлов.

Преимущества кластерных систем

- абсолютная масштабируемость,
- нет ограничений на размер узлов и кластеров,
- наращиваемая масштабируемость,
- можно расширять узлы по необходимости,
- высокий коэффициент готовности,
- отказоустойчивость, благодаря автономности узлов,
- соотношение цена/производительность,
- можно строить кластер из любых строительных блоков: чем проще и стандартнее блоки, тем дешевле обходится вычислительная мощность.

Классификация кластерных архитектур

Архитектура строится на основе представления, являются ли диски в кластере разделяемыми всеми узлами или нет.

Их можно разделить на:

- пассивное резервирование;
- резервирование с активным вторичным сервером;
- самостоятельные серверы;
- серверы с подключением ко всем дискам;
- серверы с совместно используемыми дисками.

Коммутация в кластерах

Протоколы:

- TCP (Transmission Control Protocol);
- UDP (User Datagram Protocol);
- VIA (Virtual Interface Architecture).

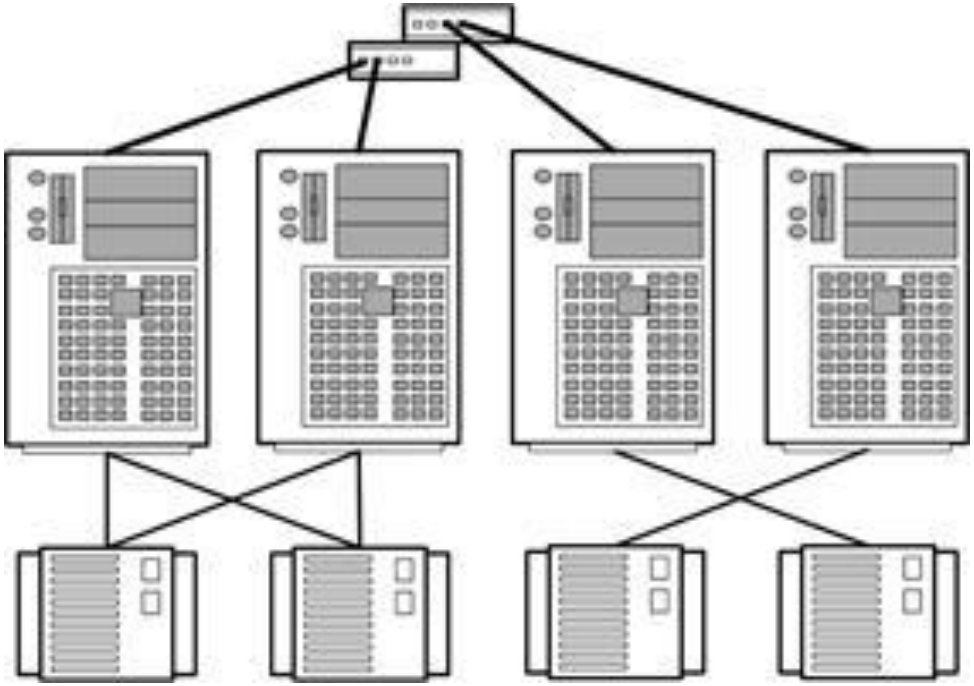
Методы:

- передача сообщений,
- разделяемая совместно используемая память.

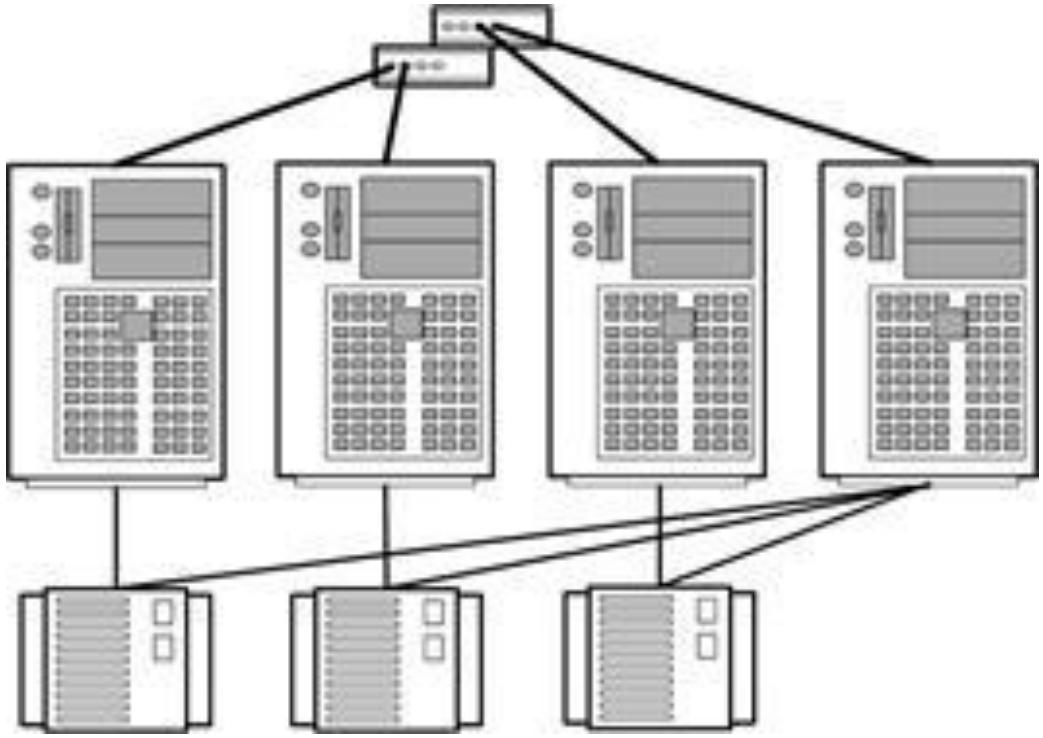
Топология кластеров

- Топология кластерных пар;
- Топология N + 1;
- Топология N x N;
- Топология с полностью раздельным доступом.

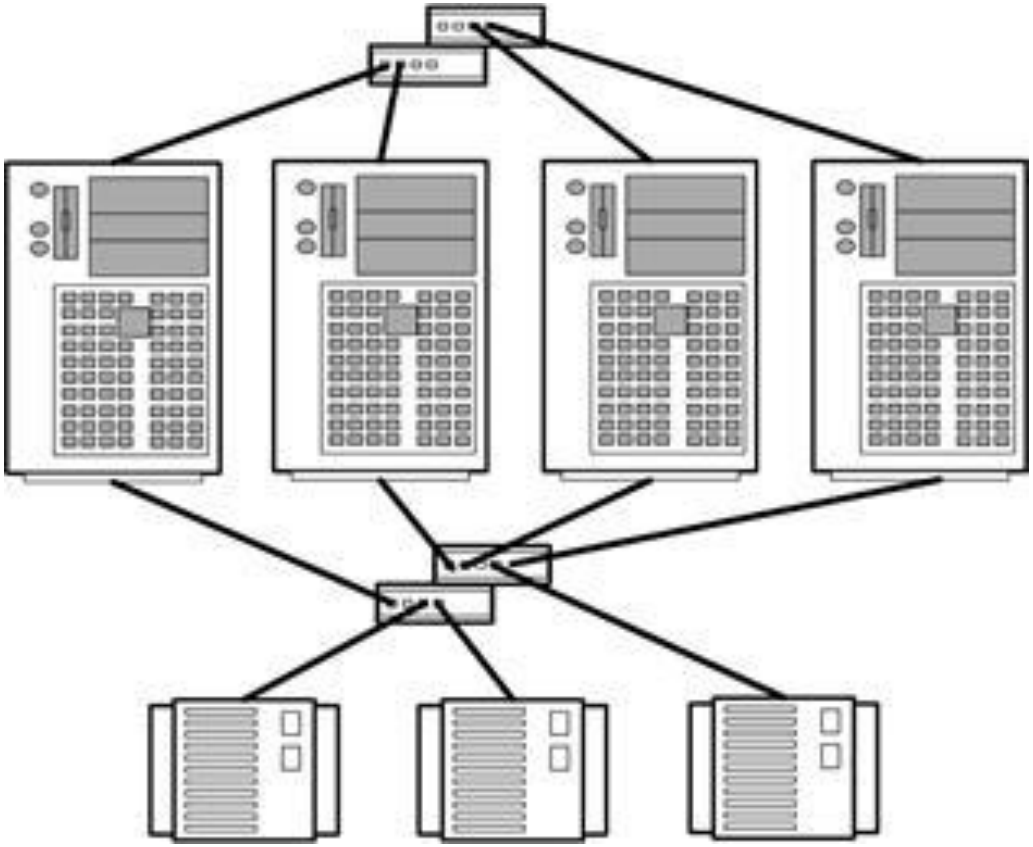
Топология кластерных пар



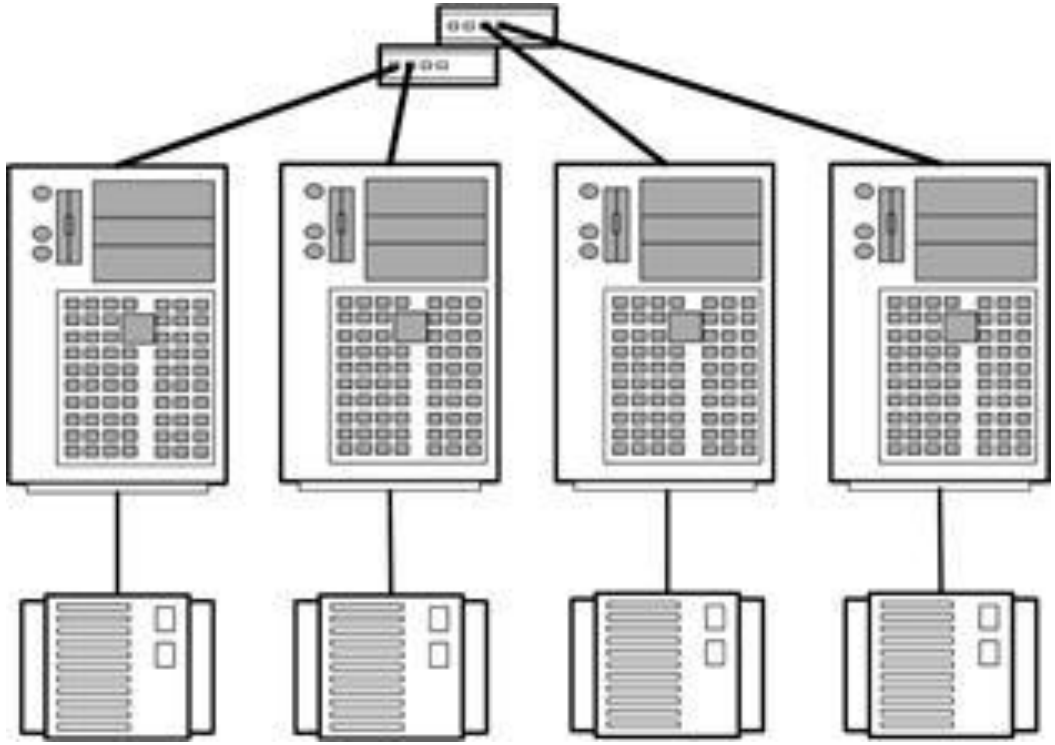
Топология N + 1



Топология N x N



Топология с полностью раздельным доступом



Типы современных кластерных систем

- отказоустойчивые кластеры (High-availability clusters, HA, кластеры высокой доступности);
- кластеры с балансировкой нагрузки (Load balancing clusters);
- вычислительные кластеры (High performance computing clusters, HPC);
- системы распределенных вычислений.

Отказоустойчивые кластеры

Отказоустойчивые кластеры

Основные игроки среди HA (High-availability):

- Corosync
- Red Hat Cluster Suite
- Microsoft Windows Server
- Linux — HA
- Pacemaker

Кластеры с балансировкой нагрузки

- сетевая балансировка;
- транспортная балансировка;
- прикладная балансировка.

Сетевая балансировка

Балансировка на сетевом уровне предполагает решение следующей задачи: нужно сделать так, чтобы за один конкретный IP-адрес сервера отвечали разные физические машины.

Такая балансировка может осуществляться с помощью множества разнообразных способов:

- DNS-балансировка;
- построение NLB-кластера;
- балансировка по IP с использованием дополнительного маршрутизатора;
- балансировка по территориальному признаку.

Сетевая балансировка. Способы

- **DNS-балансировка.**

  Основывается на алгоритмах, как правило Round Robin. Принцип работы основывается на том, что на одно DNS имя выделяется несколько IP адресов. Алгоритм позволяет в зависимости от нагрузки серверов перенаправлять трафик.

- **Построение NLB-кластера.**

  Network Load Balancing — это возможность балансировать трафик через два или более каналов глобальной сети, без использования сложных протоколов маршрутизации, таких как BGP. Используется на серверах Microsoft.

Сетевая балансировка. Способы

- **Балансировка по IP с использованием дополнительного маршрутизатора**

  Осуществляется путем размещения в сети нескольких точек входа к сервису, тем самым распараллеливания трафик между маршрутами.

- **Балансировка по территориальному признаку**

  Осуществляется путем создания в различных регионах или в странах однотипных сервисов с одинаковыми dns или ip адресами для более быстрого доступа к ним.

Транспортная балансировка

Этот вид балансировки является самым простым: клиент обращается к балансировщику, тот перенаправляет запрос одному из серверов, который и будет его обрабатывать.

Выбор сервера, на котором будет обрабатываться запрос, может осуществляться в соответствии с самыми разными алгоритмами (об этом ещё пойдёт речь ниже):

- путем простого кругового перебора,
- путем выбора наименее загруженного сервера из пула и т.п.

Транспортная балансировка

На транспортном уровне общение с клиентом замыкается на балансировщике, который работает как прокси. Он взаимодействует с серверами от своего имени, передавая информацию о клиенте в дополнительных данных и заголовках. Таким образом работает, например, популярный программный балансировщик **HAProxy**.

Прикладная балансировка

При балансировке на прикладном уровне балансировщик работает в режиме «умного прокси». Он анализирует клиентские запросы и перенаправляет их на разные серверы в зависимости от характера запрашиваемого контента.

Так работает, например, веб-сервер **Nginx**, распределяя запросы между фронтендом и бэкендом.

За балансировку в Nginx отвечает модуль Upstream.

Прикладная балансировка

В качестве еще одного примера инструмента балансировки на прикладном уровне можно привести **pgpool** — промежуточный слой между клиентом и сервером СУБД PostgreSQL.

С его помощью можно распределять запросы по серверам баз данных в зависимости от их содержания, например:

- запросы на чтение будут передаваться на один сервер,
- а запросы на запись — на другой.

Алгоритмы и методы балансировки

В числе целей, для достижения которых используется балансировка, нужно выделить следующие:

1. **Справедливость**: нужно гарантировать, чтобы на обработку каждого запроса выделялись системные ресурсы и не допустить возникновения ситуаций, когда один запрос обрабатывается, а все остальные ждут своей очереди;
2. **Эффективность**: все серверы, которые обрабатывают запросы, должны быть заняты на 100%. Желательно не допускать ситуации, когда один из серверов простаивает в ожидании запросов на обработку (сразу же оговоримся, что в реальной практике эта цель достигается далеко не всегда);

Алгоритмы и методы балансировки

3. **Сокращение времени выполнения запроса:** нужно обеспечить минимальное время между началом обработки запроса (или его постановкой в очередь на обработку) и его завершения;

4. **Сокращение времени отклика**: нужно минимизировать время ответа на запрос пользователя.

Алгоритмы и методы балансировки

А так же желательно, чтобы алгоритм балансировки обладал следующими свойствами:

1. **Предсказуемость**: нужно четко понимать, в каких ситуациях и при каких нагрузках алгоритм будет эффективным для решения поставленных задач;
2. **Равномерная загрузка ресурсов системы**;
3. **Масштабируемость**: алгоритм должен сохранять работоспособность при увеличении нагрузки.

Вычислительные кластеры

Вычислительные кластеры

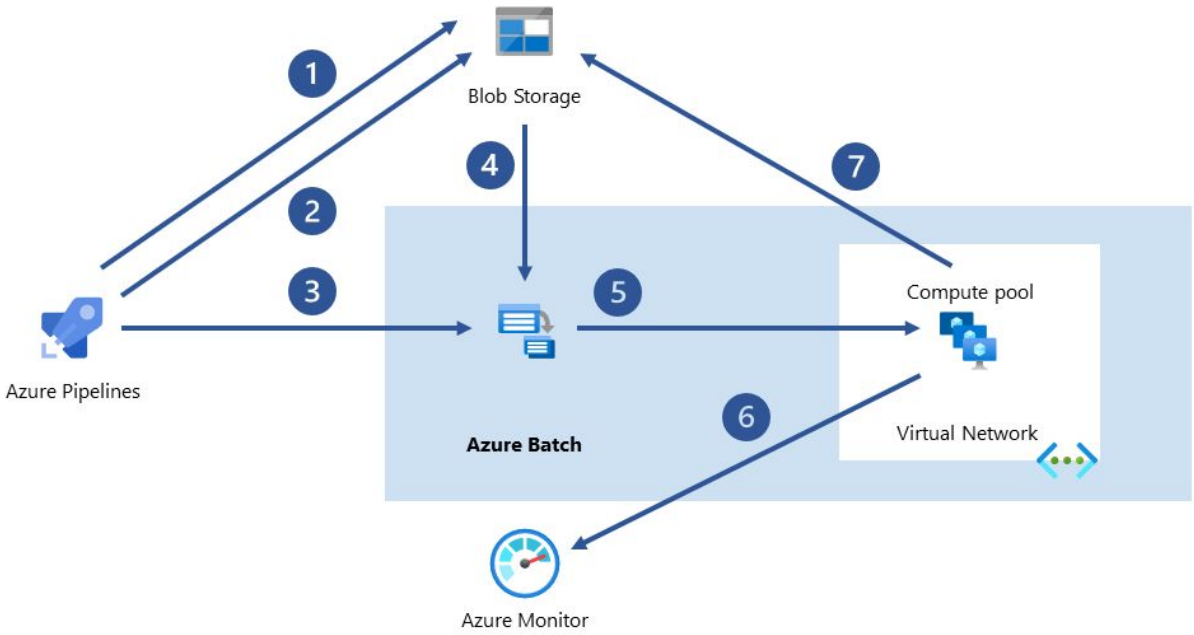
HPC кластеры (High-performance computing cluster) используются:

- в вычислительных целях, в частности, в научных исследованиях;
- расчетных задачах.

Вычислительный кластер представляет из себя **массив серверов** (вычислительных узлов или так называемых нодов), объединенных коммуникационной сетью и размещенных в отдельной стойке.

Наиболее распространенным является использование однородных кластеров, где все узлы абсолютно одинаковы по своей архитектуре и производительности.

Вычислительные кластеры



Вычислительные кластеры

Например, Azure pipelines выполняет сборку и тестирование проектов кода и ИНИЦИИРУЕТ задания HPC в пакетной службе Azure.

Служба хранилища Azure содержит данные и исполняемые файлы HPC, используемые в задании.
Пакетная служба Azure планирует задания и задачи в большом количестве узлов, а также управляет всеми ресурсами вычислений.

Вычислительные кластеры

Виртуальные машины на платформе Azure работают как рабочие роли, выполняя задачи вычислений.
Виртуальная сеть обеспечивает IP-подключение между ресурсами вычислений и другими облачными службами, которые выше и выходят за пределы любого машинного взаимодействия INFINIBAND или RDMA.
Azure Monitor собирает метрики производительности и журналы из облачных ресурсов для отчетов, предупреждений и автоматизированного ответа.

Системы распределенных вычислений

Системы распределенных вычислений

Распределённые вычисления — способ решения сложных вычислительных задач с использованием нескольких компьютеров, чаще всего объединённых в параллельную вычислительную систему.
Например:
zookeeper – брокер серверов,
kafka – система обработки и хранения информации.

Системы распределенных вычислений.

Грид-вычисления — это форма распределённых вычислений, в которой «виртуальный суперкомпьютер» представлен в виде кластеров, соединенных с помощью сети, слабосвязанных гетерогенных компьютеров, работающих вместе для выполнения огромного количества заданий (операций, работ).

Системы распределенных вычислений.

В настоящее время выделяют три основных типа грид-систем:

- **Добровольные гриды** — гриды на основе использования добровольно предоставляемого свободного ресурса персональных компьютеров;
- **Научные гриды** — хорошо распараллеливаемые приложения программируются специальным образом (например, с использованием Globus Toolkit);
- **Гриды на основе выделения вычислительных ресурсов по требованию** — обычные коммерческие приложения работают на виртуальном компьютере, который, в свою очередь, состоит из нескольких физических компьютеров, объединённых с помощью грид-технологий.

Stonith/Fencing

Stonith

Stonith — это акроним выражения "Shoot The Other Node In The Head" (застрелить другую ноду в голову — прим.пер.).

Heartbeat использует эту технологию, для гарантии, что предположительно отказавший сервер не будет мешать работе кластера, а именно, не повредит данные разделяемых дисков.

Fencing

Изоляция узла (англ. fencing) — отключение неисправного узла от кластерного хранилища с целью поддержки целостности данных. До тех пор пока узел не будет благополучно изолирован, все операции ввода-вывода в кластере будут приостановлены. Это позволяет снизить вероятность повреждения данных в хранилище.

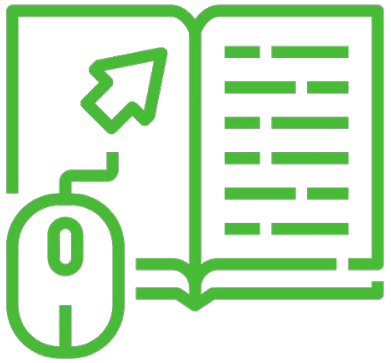
Изоляцию узла проводит специальный демон fenced.

Итоги

Итоги

Сегодня мы рассмотрели:

- что такое кластеризация, из каких компонентов она строится;
- представителей и работу отказоустойчивых кластеров (HA), кластеров с балансировкой нагрузки(Load balancing clusters), вычислительные кластеры (HPC) и системы распределенных вычислений;
- где и как они могут быть использованы.



Домашнее задание

Домашнее задание

Давайте посмотрим ваше домашнее задание.

- Вопросы по домашней работе задавайте **в чате** мессенджера Slack.
- Задачу можно сдавать **по частям**.
- Зачёт по домашней работе проставляется после того, как **приняты задача полностью**.

нетология

Задавайте вопросы и пишите отзыв о лекции!

Александр Зубарев

Александр Зубарев