ОС UNIX. Лекция 5
Обеспечение бесперебойной работы

Михаил Пожидаев

11 марта 2025 г.

Высоконагруженный сервис

Нефункциональные требования к веб-приложению

1. Обеспечение непрерывной работы с устойчивостью к сбоям и атакам.
2. Возможность проведения работ по обслуживанию и обновлений без остановки работы сервиса.
3. Отсутствие архитектурных ограничений для обеспечения масштабирования.
4. Гетерогенность функционала, подразумевающая, что в сервисе присутствуют различные по своей природе, составу или структуре функции.

Микросервисная архитектура

Из каких блоков состоит микросервисная архитектура?

Блоки микросервисной архитектуры:

1. Балансирующий веб-сервер для принятия соединений.
2. Контейнеры с фронт-эндом для обработки запросов.
3. Брокер сообщений для организации взаимодействия контейнеров между собой.
4. База данных для хранения пользовательских данных.
5. Файловое хранилище для хранения пользовательских бинарных данных.
6. Набор микросервисов для выполнения работ.

Rolling updates

Требования к решению задачи «горячих» обновлений

1. Контейнеры, не являющиеся брокером, периодически получают обновления. Чем чаще, тем лучше.
2. Работу сервиса останавливать нельзя, пользователи должны наблюдать непрерывное функционирование с нормальной обработкой запросов.
3. Близкой является задача динамического изменения набора контейнеров-работников для увеличения или уменьшения задействованных вычислительных мощностей.

Процедура

Порядок выполнения обновления

Механизм реализуется в связке с используемым инструментом оркестрации, выполняющим запуск и остановку реплик.

1. Инструмент оркестрации получает команду изменить конфигурацию системы, определяет набор контейнеров для остановки.
2. Контейнеры получают сигнал завершить свою работу (`SIGTERM`), после чего уведомляют брокер о своём новом статусе.
3. Контейнеры завершают выполнение тех задач, которые были в работе на момент получения сигнала.
4. Контейнеры либо добровольно завершают свою работу, либо их убивают по timeout при помощи `SIGKILL`.

Команды и ключи Docker

Основная команда обновления конфигурации:
`docker service update --image`

Пара значимых параметров:

- `update-delay`: задержка между успешными обновлениями контейнеров;
- `stop-grace-period`: задержка после отправление `SIGTERM` до отправления `SIGKILL`.

Остановка

Обработка сигнала завершения работы

Java:

```
Runtime.getRuntime()
.addShutdownHook(new Thread(()->{...});
```

Python с AIOHttp:

```
app.on_shutdown.append(on_shutdown)
```

Обновление контента

Как не разрушить целостность на стороне клиента?

1. **Использование балансирующего сервера на входе.** Соединения принимает обратный прокси-сервер (в наше время почти всегда Nginx) с переменным множеством реплик фронд-енда. Реплика на время обновления исключается из множества для обработки запросов.
2. **Использованеи атомарных операций.** Заменить файл можно без остановки фронт-енда путём использования специальных атомарных операций в файловой системе (например, `renameat2()`).

Nginx

Бесперебойная балансировка запросов

```
stream {
  upstream mysite {
    server web1:80 weight=1;
    server web2:80 weight=1;
  }
  server {
    listen 443 ssl;
  }
}
```

Атомарные операции

1. Поддержка файловых блокировок существенно различается между разными реализациями NFS. В случае Samba информация достаточно противоречива.
2. При этом доступны некоторые атомарные операции (например, `renameat2()` умеет атомарно менять файлы местами).

Исследование Ленардом Поттерингом поддержки файловых блокировок в Linux, включая использование NFS:
http://0pointer.net/blog/projects/locking.html

NFS

Network File System

NFS — протокол сетевого доступа от Sun Microsystems.

1. Работает поверх протокола ONC RPC.
2. Встречается в версиях 3 и 4.
3. Достаточно устойчиво обрабатывает временные потери соединений.
4. Подразумевается единое пространство UID для всех клиентов хранилища.

S3

Amazon Simple Storage Service

Практичное файловое хранилище со следующими возможностями:

- организует плоское пространство имён (в отдельных операциях можно задавать разделитель каталогов);
- позволяет организовать работу статического веб-сервера;
- хранит дополнительную веб-информацию, включая content значения `Content-Type`, `Disposition` и пр.
- не поддерживает блочные операции чтения/записи.

Спасибо за внимание!

Всё о курсе: https://marigostra.ru/materials/unix.html
E-mail: msp@luwrain.org
Канал в Телеграм: https://t.me/MarigostraRu