

Студенческие Проекты

Acronis

СПбАУ, весна 2018

Организационная часть

- Все вопросы можно задавать сюда (Егор): egor-mailbox@ya.ru
- Стипендии
 - Студенты выпускных курсов (4, 6): 0 - 45 т.р.
 - Иначе: 0 - 25 т.р.
- Удалённая работа, т.к. проекты из Москвы

- Ссылка на эту презентацию:

<https://goo.gl/7SDNGH>

Контекст. Система архивации данных

- Файловая система с поддержкой Point In Time (PIT) версий
 - Операции:
 - CREATE/DELETE/MODIFY file in current (most recent) PIT;
 - CREATE/DELETE any PIT
 - READ file in any PIT
- Метаданные хранятся в LSM-деревьях (Log Structured Merge-tree); Такие деревья ориентированы на частые операции вставки
- Данные хранятся:
 - В файле (случай использования архива на локальной файловой системе, в NFS)
 - В объектном хранилище (в облаке): Amazon S3, Google Cloud Storage
- Система написана на Go

1.1 Дедупликация с использованием golomb фильтров

- С точки зрения метаданных, файлы в архиве хранятся экстентами (непрерывными последовательностями байт нефиксированной длины)
- В текущей реализации имеется индекс, отображающий hash в идентификатор экстента (aext ID), хранящийся в LSM дереве
 - Одна запись в таком индексе составляет ~13 байт
 - При размере архива в 1 Петабайт (и максимальном размере одного экстента в 256 КБ) получаем, что индекс будет занимать ~50 ГБ
- **Проблема:** deduplication индекс слишком большой, чтобы помещаться в RAM, из-за чего при записи lookup в этом индексе приводит чтению с диска
- **Задача:** оптимизация (с точки зрения операций с диском) индекса посредством добавления golomb фильтра (модификация bloom фильтра)
- Сложность: 1 семестр, 3-5 курс

1.2 Сборка мусора и compacting

- При удалении Point In Time (PIT) из архива на уровне хранилища появляются области, занятые ненужными данными
- Текущая реализация:
 - *Хранилище - файл в FS*. Используется поддержка разреженных файлов в Unix (FALLOC_FL_PUNCH_HOLE, F_PUNCHHOLE) и Windows (FSCTL_SET_ZERO_DATA) для того, чтобы сказать ОС, что части файла (архива) можно освободить
 - *Хранилище - object storage (GS, S3)*. Объект удаляется из хранилища как только все экстенды, хранимые в рамках этого объекта удалены.
- **Проблема:**
 - *Хранилище - файл в FS*. Некоторые файловые системы (NFS, exFat, ...) не поддерживают разреженных (sparse) файлов
 - *Хранилище - object storage (GS, S3)*. Получаем объекты, которые используются лишь частично.
- **Задача:** реализовать честную сборку мусора (detect garbage -> compact data and delete garbage)
- Сложность: семестр, 3-5 курс

3. Система мониторинга метрик для распределённого хранилища данных

- *Контекст:* распределённое хранилище данных (решение Acronis), для примера можно посмотреть на CEPH
- Во время работы системы происходит сбор всевозможных метрик/статистик о работе системы:
 - Параметры I/O дисков, Сетевые параметры, CPU, Swap, Число запросов, Latency запросов
- **Проблема:** суммарное число снимаемых показателей велико: для каждого сервера, по каждому типу запроса, по каждому сервису и пр. => глазами смотреть и анализировать это трудоёмко
- **Задача:** спроектировать систему анализа/мониторинга получаемых метрик с механизмом оповещений (alerts) в случае обнаружения аномалий (anomaly detection)
- Технологии: Go (предпочтительней), Python
- Сложность: 1 семестр, 3-5 курс

4. Георепликация виртуальных машин QEMU/KVM

- Контекст: развёртывание виртуальных машин пользователей в рамках облачной инфраструктуры Acronis
- При предоставлении высоко доступных (highly available) виртуальных машин требуется защищаться от возможных поломок: вместе с исполняемой (активной) в данный момент VM (*source VM*) существует пассивная (*target VM*): все изменения, происходящие с *source VM* должны консистентным образом переноситься на *target VM*. Таким образом при падении *source VM* можно переключиться на *target VM* прозрачно для пользователя
- **Проблема:** в текущей реализации этого нету :)
- **Задача:** реализовать прототип системы (программы) осуществляющей консистентное реплицирование изменений одной VM на другую для QEMU-KVM виртуальных машин.
- Технологии: Си
- Сложность: 1 семестр, 3-5 курс

5. Реализация life cycle / quotas API объектного хранилища S3 (Amazon)

- Спецификация Amazon S3 Object Storage предполагает предоставление определённых сервисов, в том числе:
 - Quotas: слежение за ограничениями по используемым данным конкретным пользователем,...
 - Life Cycle: возможность удаление объекта по истечению периода времени (expiration and transition actions)
- Эти пункты спецификации нельзя реализовывать в распределённой системе хранения данных тривиальным способом
- Задача: реализовать S3 life cycle / quotas API в рамках распределённого объектного хранилища в Acronis
- Сложность: 1 семестр, 3-5 курс