

Подробности передачи ресурса с удаленного устройства методом, использующим Content-addressable storage

Как `desync`, так и `casync` поддерживают, кроме всего прочего, операции двух типов: деления ресурса на чанки (`make` и `tar` для `desync`, `make` для `casync`) и доставки (`extract` и `untar` для `desync`, `extract` для `casync`). При этом методы доставки позволяют указать место хранения кеша на локальном устройстве и удаленное хранилище чанков. То есть передавая в команду доставки аргументом `caibx/caidx`-файл (индексный файл, указывающий на файл и директорию, соответственно), необходимые чанки будут сначала искаться в локальном хранилище, а только после этого - в удаленном. Таким образом, скачиваться будут только необходимые куски.

То есть использование утилит `desync` и `casync` состоит из двух этапов: разделение исходного ресурса на чанки и загрузка в удаленный репозиторий индексного файла и этих чанков (производится 1 раз для всех клиентов), далее на каждом клиенте вызывается метод доставки, указывая при этом индексный файл, локальное и удаленное хранилища кеша.

Экспериментальная установка

Рассматриваются версии пакета $k, k+1, k+2, \dots, k+m$. Хотим эмпирически измерить время доставки $(k+m)$ -й версии при условии:

- 1) Если способ доставки использует технологию Content-addressable storage, то предварительно клиент хранит кеш от версий $k, k+1, \dots, k+m-1$.
- 2) Иначе $(k+m)$ -я версия замещает $(k+m-1)$ -ю.

Построим для каждого рассматриваемого ресурса графики времени доставки от номера версии m .

Рассматриваемые ресурсы:

1. Директории с кодом проекта

В качестве проекта возьмем ClickHouse.

Как версии проекта возьмем версию репозитория с разницей в месяц, начиная с июня 2024 года. Всего получается 11 версий. Размер одной версии $\approx 220M$.

2. Бинарный файл с данными

В качестве ресурса возьмем `neovim` версий 7.0-10.2, получилось 11 версий. Размер одной версии $\approx 39M$.