Cassandra. Описание алгоритмов.

* 1. Краулинг, скрапинг

# Алгоритм рекраулинга VK

* 1. Для рекраулинга в БД git200\_crawl.sn\_activity фиксируется:
     + last\_date - дата последней активности группы / дата последней активности поста группы
     + upd\_date - дата последнего рекраулинга группы / дата последнего рекраулинга поста
     + fin\_date - дата завершения краулинга/рекраулинга группы (фиксируется для всех постов)

Запись посл. активности группы, отличается от записи посл. активности поста тем, что в первой нет id поста, т.е. эта запись относится к группе в целом.

Автоматически созданный замещающий текст:
souces 
ec773e72.D53d-433D-sscc-s 
ff10d1ff.e8aS-47fD-S4fc-1dB8 
e3D78883-3ed3-481d.a275-7 
fc8Efd7f.e4S8-4fSf.b70c.acd 
ffd2594e.87D1-4734-8DBE-s,7 
1 531 
16 758 516 
16 758 516 
16 758 516 
1€ 758 
16 758 516 
16 758 516 
1€ 758 
114 70€ 21 08 
осчз 
1138SD 2D2D-11-1E 2351 
114 2D2D-1D-31 15:14 
114 2D2D-1D-23 2239 
114 422 2D2D-1D-23 2247:0043 
114 Зн 
ира «ate 
2020-11-17 
2020-11-17 2221Ђ2*аз 
2020-11-17 2221Ђ2*аз 
2020-11-17 2221Ђ2*аз 
2020-11-17 2221Ђ2*аз 
2020-11-17 2221Ђ2*аз 
2020-11-17 2221Ђ2*аз 

* 1. Параметры проектов git000\_cfg.projects для рекраулинга:
     + date\_deep - глубина скачивания. git000\_cfg.projects
     + recrawl\_days\_post - количество дней рекраулинга поста, от даты последней активности поста
     + recrawl\_days\_reply - количество дней рекраулинга ответов на сообщения, от даты сообщения

* 1. Принцип рекраулинга постов

Исходная постановка: при повторном краулинге посты дублировать не нужно, но новые обсуждения пользователей по данному посту нужно скраулить. Наличие новых сообщений пользователей для постов должны учитывать дату поста, дату последней активности поста, дату последнего рекраулинга поста. Т.е. очень старые посты не должны рекраулиться.

Первое ограничение - дата последнего рекраулинга группы upd\_date. Если дата поста меньше данной upd\_date группы и upd\_date группы не пустая, то считается что данный пост уже скраулен и не рекраулится. Как только попался первый ранее скрауленный пост, фетчинг страниц прекращается.

* 1. Принцип рекраулинга сообщений 1-го уровня (reply to post)

В алгоритм краулинга группы требуется передавать список постов (=список рекраулинга), сообщения по которым нужно рекраулить. Для определения данного списка постов учитываются две даты - дата последней активности поста и дата последнего рекраулинга поста.

Пост включается в список рекраулинга до тех пор, пока дата его последнего рекраулинга не выйдет за период от даты последней активности, до даты последней активности + параметр «кол. дней рекраулинга поста». Т.е. условие включения post в рекраулинг сообщений следующее: post upd\_date in (last\_date, last\_date + recrawl\_days\_post)

Для получения списка постов используется функция postgresql git200\_crawl.get\_sn\_activity.

Сообщения новых постов краулятся в полном объеме.

Сообщения старых постов вносятся в базу если дата сообщения >= post upd\_date. Но, с целью проверки наличия новых сообщений второго уровня, данный пост может быть включен в соответствующий список по условиям, описанным в пункте ниже. Рекраулинг сообщений по конкретному посту останавливается, при встрече с первым сообщением 1-го уровня (!), которое уже скраулено (дата >= post upd\_date) и которое вышло за заданный период ожидания ответов на него (см. пункт ниже).

* 1. Принцип рекраулинга сообщений 2-го уровня (reply to reply)

Сообщения 1-го уровня с датой ранее даты последнего рекраулинга поста уже загружены, но не могут не обрабатываться, т.к. на них могут быть даны ответы "reply to reply". Т.е. они повторно не включаются в БД, но включаются в список для проверки появления ответов на них. Для этой цели введен параметр recrawl\_days\_reply означающий количество дней, в течении которых система считает, что на сообщение может быть дан ответ.

Для вычисления берется дата посл. рекраулинга поста, из нее вычитается кол. дней ожидания: и все сообщения 1-го уровня с датой позднее полученной – включаются в список рекраулинга сообщений второго уровня. Т.е. условия включения сообщения в рекраулинг следующее: reply date > post upd\_date - recrawl\_days\_reply

В алгоритм требуется подать словарь: (id\_post : upd\_date - recrawl\_days\_reply)

Требуется учесть, что сообщения 2-го уровня идут в прямом хронологическом порядке, в отличие от сообщений 1-го уровня, которые идут в обратном хронологическом порядке. При этом, сообщения 2-го уровня идут группами – в пределах группы они идут в прямом хронологическом порядке, но сами группы идут в обратном хронологическом порядке. Т.о. вся группа должна обработаться полностью, т.к. новые сообщения будут в конце. При этом сообщения добавляются в базу только при условии, что их дата больше даты последнего краулинга поста (post upd\_date, см. пункт выше). Если при обработке сообщений группы хотя бы одно сообщение вышло за пределы даты ожидания сообщений upd\_date - recrawl\_days\_reply (см. выше), то фетчинг сообщений 2-го уровня для данного сообщения 1-го уровня требуется остановить.

* 1. Повышение устойчивости на случай обрывов

В случае обрывов связи часть дат по группе и по постам группы уже зафиксировалась в БД. Поэтому нет информации полностью ли скраулена группа или нет, полностью ли скраулены ответы на посты или нет. Для решения данной проблемы вводится поле fin\_date, которое содержит дату завершения краулинга группы. Поле заполняется/обновляется по завершении краулинга/рекраулинга группы для всех постов группы и для самой группы.

При получении дат last\_date и upd\_date они не могут быть больше, чем дата fin\_date.