Рассказ

Оператору связи необходимо узнать маржинальность за обработку каждого звонка для оценки эффективности предоставляемых услуг. В процессе маршрутизации звонков коммутаторы создают CDR записи, в которых документируются детали телефонного звонка, проходящего через конкретное устройство. В данных записях недостаточно информации для оперативного формирования соответствующих выводов. Известные инструменты и алгоритмы являются иностранными и недоступными, а полный перебор всех вариантов не позволит поспособствовать изменению траспортировки звонков.

Целью моей работы является разработка программного обеспечения, которое будет загружать CDR коммутаторов из заданного источника, проводить их корреляцию и выгружать информацию о построенных цепочках, изходя из чего можно будет оценить эффективность услуг оператора связи и судить о её возможном повышении.

Для достижение поставленной цели потребуется решить следующие задачи: спроектировать архитектуру программного обеспечения, составить схему базы данных, разработать генератор данных, который будет по заданной структуре сети маршрутизировать случайным образом созданные звонки, разработать оптимальный алгоритм восстановления маршрутов звонков, произвести тесты производительности программы на исследование эффективности работы алгоритма.

Генератор устроен следующим образом. Для начала требуется задать параметры сети и описать каждый коммутатор, а также все связи в сети. Таким образом, полученную сеть можно описать в виде графа, в котором устройства удалены друг от друга на некоторое расстояние. С помощью алгоритма Флойда-Уоршелла можно лёгко вычислить кратчайщие пути между операторами, по которым и будут далее маршрутизироваться звонки. Данные о звонках генерируются случайным образом, после чего каждый звонок маршрутизируется по сети согласно с указанными данными о начальном и конечном операторе. При маршрутизации также вносится некоторая диффузия в данные о длительности звонка и текущем времени, чтобы соответствовать реально картине мира. Также существует некоторая вероятность потери записи на коммутаторе.

После генератора запускается алгоритм восстановления звонков. Из таблицы CDR для каждого звонка выбирается первая CDR запись в цепочке, после чего происходит поиск следующего CDR в пути звонка на следующем коммутаторе, используя формулу на слайде. – различие во времени создания CDR записей, k – коэффициент для степени учёта данного параметра, который выбирается эксперементально, – различие в длительности звонка, m - коэффициент для степени учёта данного параметра, который выбирается эксперементально, q – предельное значение данной метрики, которое не должны превышать CDR, относящиеся к одному звонку и стоящие в цепочке на соседних местах. По итогам такого сопоставления будет восстановлена полная цепочка пути для каждого звонка, включая исходящего и входящего оператора.

Архитектура программного обеспечения выстроена следующим образом: с помощью Docker и Visual Studio Code организуется подключение на удалённый сервер, на котором происходит работа всего алгоритма.

На удалённом сервере основной алгоритм вызывает сначала генератор (если требуется), генератор заполняет базу данных информацией, после чего алгоритм по сгенерированным данным уже формирует цепочки путей звонков.

Логическая модель базы данных выглядит следующим образом: на каждом коммутаторе хранится таблица CDR записей, в которой для каждой записи указывается информация о входящем и исходящем абоненте, длительности звонка, времени создания записи, а также входящем и исходящем транках коммутатора, которые участвовали при маршрутизации данного звонка. На каждом коммутаторе есть несколько транков (виртуальных каналов связи, передающих трафик между сетевыми устройствами), с помощью которых коммутаторы соединяются между собой. Каждый транк содержит информацию об id удалённого коммутатора, с которым осуществляется соединение через данный транк, а также расстояние до удалённого коммутатора.

На слайде приведены примеры восстановленных цепочек. Для каждого звонка указывается: номер звонка, исходящий оператор, цепочка CDR записей, которые соответствуют данному звонку, конечный оператор, которому звонок был передан далее. Если восстановить цепочку не удалось, выводится сообщение “CAN’T RESTORE PASS”.

Перейдём к тестам производительности. На данном графике изображена зависимость процента восстановленных звонков от вероятности потери CDR записи. Тестирование производилось на данных различного размера. Получившая зависимость практически линейная, исключения составляют только маленькие данные, т.к. вклад одного звонка сильно влияет на общие результаты.

Далее исследовалась зависимость процента восстановленных путей от предельного значения метрики для сопоставления двух CDR. Самый сильный рост наблюдается на промежутке от 15 до 30 мсек граничного значения.

Зависимость времени работы алгоритма от размера данных также была исследована. Сильный рост графика объясняется увеличением времени поиска информации в базах данных.

В результате выполнения ВКР было разработано программное обеспечение для восстановления пути звонков внутри коммутаторной сети, а также генератор данных, с помощью которого тестируется основной алгоритм. В качестве улучшения продемонстрированного алгоритма рассматривается сортировка содержания таблиц после их полного заполнения, что позволит выполнять поиск информации за логарифмическую сложность относительно размера данных.