DOI 10.58351/2949-2041.2026.32.3.012

Гергедава Джони Романович, Магистрант
МГТУ им. Н.Э. Баумана
Gergedava Johny Romanovich,
Magister, BMSTU

ПРОГРАММНАЯ СИСТЕМА ВИЗУАЛИЗАЦИИ ГРАФОВ НА ОСНОВЕ ДИАГРАММ САНКИ
PROGRAM SYSTEM FOR VISUALIZING GRAPHS BASED ON SANKE CHART

Аннотация. В статье рассматривается архитектура и принципы построения программной системы, предназначенной для визуализации графовых структур с использованием диаграмм Санки. Представленная система позволяет анализировать сложные взаимосвязи и повышает наглядность интерпретации больших объемов связных данных.

Abstract. The article discusses the architecture and principles of building a software system designed to visualize graph structures using Sanki diagrams. The presented system allows for the analysis of complex relationships and enhances the visualization of large amounts of connected data.

Ключевые слова: Графовые базы данных, ArangoDB, AQL, диаграмма Санки, агрегация данных, визуализация потоков, d3-sankey.

Keywords: Graph databases, ArangoDB, AQL, Sanki chart, data aggregation, flow visualization, d3-sankey.

Агрегирование графовых данных для последующего построения диаграммы Санки сводится к преобразованию множества первичных связей/событий в набор потоков вида source→target с числовой мерой value, где источники и приёмники однозначно сопоставляются узлам. Практические реализации Sankey (например, d3-sankey) допускают задание source и target как строковыми/числовыми идентификаторами, поэтому ключевым требованием является устойчивость идентификаторов узлов и согласованность ссылок в links.

Прямое агрегирование по рёбрам

Подход заключается в группировке исходных рёбер по паре (источник, приёмник) и вычислении агрегата веса (SUM/COUNT/AVERAGE и др.) для каждой пары, что непосредственно формирует связи. В AQL данная операция концептуально соответствует COLLECT по критериям группы и вычислению агрегатов через AGGREGATE, причём выполнение агрегации «на лету» в процессе группировки обычно эффективнее, чем накопление всех значений групп с последующей пост-агрегацией.

Агрегирование по классам/типам узлов

В данном подходе первичные вершины предварительно отображаются в укрупнённые классы (например, сегмент пользователя, категория продукта, тип события), после чего потоки агрегируются уже между классами, а не между конкретными вершинами. Технически это реализуется как группировка по вычисляемым признакам (нескольким ключам группировки в COLLECT) с последующим расчётом value через AGGREGATE, что позволяет контролировать уровень детализации и размерность результата (число nodes/links).

Агрегирование по путям

Подход применяется, когда поток трактуется как последовательность переходов (многошаговый маршрут), а не как одиночное ребро, поэтому исходными наблюдениями становятся пути фиксированной или ограниченной длины, извлекаемые обходом графа. Этот механизм в AQL допускает получение переменной пути вместе с вершинами и рёбрами обхода, что позволяет агрегировать либо целые маршруты, либо переходы между соседними шагами маршрута, после чего приводить результат к парным связям Санки.

Многокритериальная агрегация: контекст, время, окна

На практике потоки часто анализируются в разрезе контекста (канал, сценарий, статус) и/или времени, поэтому критерии группировки расширяются дополнительными ключами, а итоговая Санки-модель строится для каждого среза отдельно (например, «за период», «по сегменту», «по сценарию») [1]. Возможность группировки по нескольким критериям и одновременного вычисления нескольких агрегатов является штатной для COLLECT/AGGREGATE и позволяет формировать согласованные наборы потоков для сравнительного анализа.

Для библиотеки построения диаграмм Санки исходные данные интерпретируются как потоки в ориентированной ациклической сети, поэтому при подготовке данных важна обработка циклов (например, исключение, «разворачивание» по слоям или введение специальных узлов), иначе модель может стать неоднозначной для слоистой раскладки. Следовательно, выбор схемы агрегирования определяется компромиссом между интерпретируемостью (укрупнение классов уменьшает «шум»), полнотой (путь сохраняет семантику маршрута) и вычислительной стоимостью.

ArangoDB предоставляет набор средств для извлечения и агрегирования графовых данных на языке AQL, что позволяет формировать из исходного графа агрегированные потоки, пригодные для последующего представления в формате диаграммы Санки. В рамках рассматриваемой задачи принципиально важно, что обработка делится на два этапа: (1) получение «сырых» наблюдений (рёбер или путей) и (2) свёртка этих наблюдений в агрегированные переходы вида source→target с метрикой value.

Средства извлечения данных опираются на механизм обходов графа, позволяющий возвращать не только текущие элементы обхода (вершину и ребро), но и объект пути, содержащий массивы vertices и edges, то есть структуру маршрута целиком; это является основой для построения потоков по многошаговым цепочкам.

Для задач построения потоков ключевым является вариант «COLLECT... AGGREGATE...», поскольку он формирует агрегаты инкрементально в ходе группировки и оказывается более эффективным, чем подходы, требующие накопления всех значений групп с последующей пост-агрегацией.

Для удобства дальнейшего проектирования запросов и интерпретации результатов целесообразно выделить основные элементы AQL, используемые в задаче «граф → потоки Санки»:

1) Traversals: получение рёбер и/или путей на глубине min...max в заданном направлении OUTBOUND|INBOUND|ANY, с возможностью вернуть vertices и edges;
2) PRUNE и OPTIONS: раннее отсечение ветвей обхода и настройка параметров обхода;
3) COLLECT / AGGREGATE: группировка результатов по ключам;
4) WITH COUNT / INTO / KEEP: подсчет количества элементов в группе и управление тем, какие значения переносятся в группу, что используется для контроля объема промежуточных данных.

Инструменты визуализации диаграмм Санки различаются по степени «встроенности» в аналитический контур и по тому, какой формат входных данных считается базовым. Для целей настоящей работы существенны не столько средства отрисовки, сколько требования к структуре данных (узлы/связи, идентификаторы, метрика веса), поскольку именно они определяют правила преобразования результатов агрегирования в ArangoDB в потоковую модель.

Библиотека d3-sankey ориентирована на визуализацию направленного потока между узлами в ациклической сети и принимает структуру nodes и links, где для каждой связи задаются source, target и числовое value. Входные source и target могут задаваться как объектными ссылками, так и строковыми/числовыми идентификаторами, что удобно при передаче данных в JSON и при экспорте результатов из СУБД. В d3-sankey значение узла

интерпретируется как сумма входящих link.value, а ширина связи вычисляется пропорционально link.value, что фиксирует прямую зависимость визуализации от выбранной агрегированной метрики [2].

В Plotly Санки диаграмма задаётся через две части – node и link, при этом связи определяются массивами link.source, link.target, link.value. Источник и приёмник в Plotly указываются индексами, соответствующими позициям узлов (например, меток) в массиве node.label, то есть на уровне данных требуется стабильное сопоставление «идентификатор узла → индекс». Подобная модель удобна для интеграции в отчёты и интерактивные дашборды, однако в контексте исследования агрегирования ключевым остаётся тот факт, что семантика потоков полностью задаётся входными source/target/value, а не «выводится» визуализатором [3].

BI-инструменты и облачные отчётные платформы (например, Looker Studio) обычно описывают Санки через измерения и метрику: требуется указать измерение-источник, измерение-приёмник и числовую метрику веса, которая определяет толщину связи. Вес трактуется как агрегированная метрика и может быть получена, в частности, с применением функций COUNT (), SUM () или AVG () к исходному набору данных. Такой подход упрощает построение диаграмм в рамках отчётности, но переносит критически важное решение (что именно агрегировать и по каким ключам) в слой подготовки данных, то есть напрямую в область исследуемых схем агрегирования [4].

Для последующих разделов работы из данного обзора следуют практические требования к результатам агрегирования:

1) выходной набор должен однозначно формировать links как пары source-target с числовым value, так как именно value определяет ширину связи;

2) должна быть обеспечена устойчивая идентификация узлов, поскольку визуализаторы опираются индексы узлов;

3) вес потока должен быть заранее определен как агрегат, поскольку BI-подходы к Санки трактуют вес как агрегированную метку.

Разработка схем агрегирования графовых данных ArangoDB – формализация графовой модели.

Исходные данные задаются ориентированным графом G= (V,E), где

V – множество вершин, E ⊆ V × V – множество ориентированных рёбер. Каждому ребру e ∈ E сопоставляются начало from (e) ∈ V и конец to (e)∈V, а также набор атрибутов A (e).

В ArangoDB ребро является документом edge collection и содержит специальные поля _from и _to, которые ссылаются на идентификаторы документов вершин, тем самым реализуя отображения from (e) и to (e).

Для перехода к потоковой модели вводится функция веса w: E → R≥0, задающая вклад ребра в поток (например, w (e) = 1 для подсчёта событий или

w (e) = amount (e) для суммирования величины).

Тогда агрегированный поток между двумя вершинами u,v ∈ V определяется формулой (1).Именно значения $F(u,v)$ формируют толщину связей в диаграмме Санки, так как в d3-sankey каждая связь имеет числовое значение, а ширина связи пропорциональна этому значению.

$$F(u,v) = \sum_{\in E:\, from(e)=u, to(e)=v} w(e) \qquad (1)$$

Для визуализации в модели Санки требуется сформировать два набора: узлы N и связи L. Узлам ставятся в соответствие вершины графа: $N = \{n_v \mid v \in V'\}$, где $V' \subseteq V$ – вершины, участвующие в итоговой диаграмме. Связи формируются из пар (u, v) с ненулевым $F(u,v)$: $L=\{(u,v,F(u,v))\}$.

При этом d3-sankey допускает задавать концы связей не ссылками на объекты узлов, а числовыми или строковыми идентификаторами, что удобно при передаче результата из ArangoDB в JSON.

В таблице 1 представлено описание компонентов.

Таблица 1

Соответствие для формальной модели, ArangoDB и Санки

Компонент	Формально	Представление в ArangoDB	Требование для Санки
Вершина	$v \in V$	Документ в vertex collection	Устойчивая идентификация узла, на которую ссылаются связи
Ребро	$e = (u,v) \in E$	Документ в edge collection с _from, _to	Для каждой связи должны быть определены начало, конец и числовая величина потока
Вес	$w(e) \geq 0$	Числовой атрибут ребра или константа 1	Используется как значение связи; ширина пропорциональна значению
Агрегированный поток	$F(u,v)$ по (1)	Результат агрегации по _from/_to	Значение связи между соответствующими узлами

Так, фиксируется минимальное соответствие между формальной моделью, представлением в ArangoDB и данными для Санки.

В первой схеме агрегирования поток строится напрямую из рёбер: каждое ребро в ArangoDB задаёт направленную связь между двумя вершинами через поля _from и _to, а «сила» связи между одной и той же парой вершин получается свёрткой всех таких рёбер в одно число.

На уровне AQL эта свёртка реализуется группировкой по паре _from/_to через COLLECT и вычислением агрегата через AGGREGATE, который накапливает агрегаты инкрементально и часто более эффективен. Итоговый результат представляет собой список агрегированных переходов «откуда → куда» с числовым значением, который напрямую переводится в набор связей Санки, поскольку d3-sankey ожидает для каждой связи начальный узел, конечный узел и числовое значение, причём узлы могут задаваться идентификаторами, а не объектами (рисунок 2).

Данная схема является базовой и наиболее прямой: она сохраняет детализацию на уровне конкретных вершин и показывает суммарный «поток» между каждой парой (*from→to*).

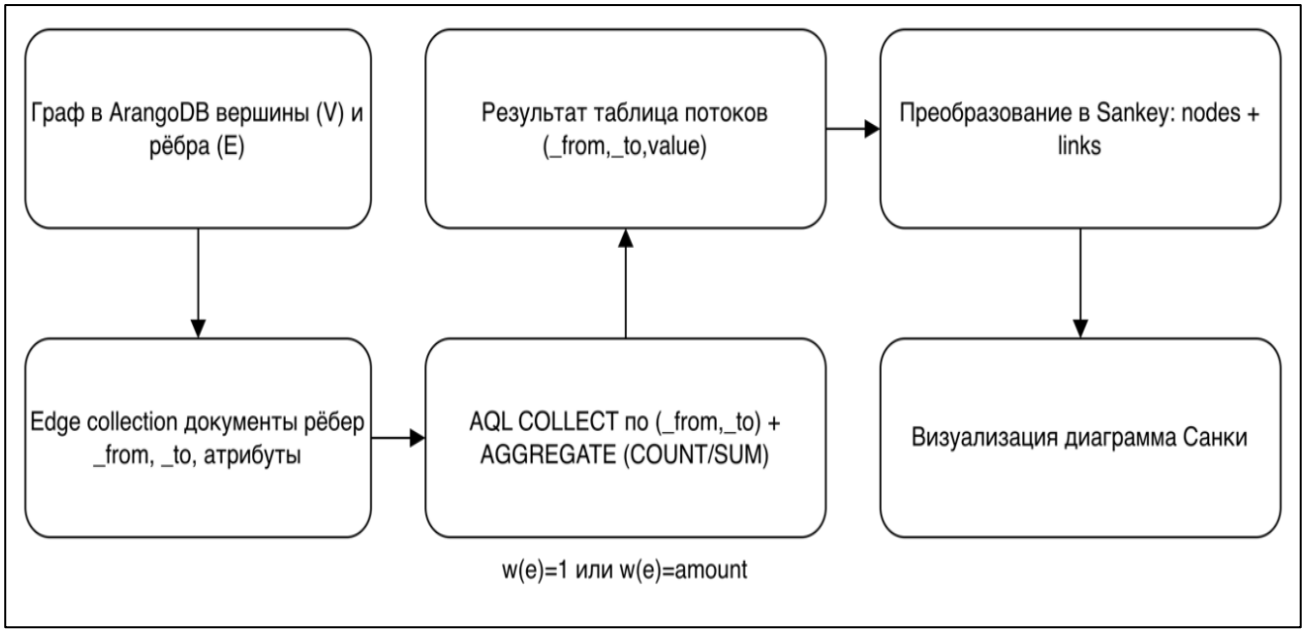

Рисунок 2 – Схема прямого агрегирования по рёбрам

При выборе функции веса *w* (*e*) меняется смысл толщины ребра на диаграмме Санки, поэтому параметр агрегации следует выбирать в соответствии с целью анализа.

На второй схеме исходный граф в ArangoDB рассматривается так же, как и в схеме 1, но вершины предварительно сводятся к типам/классам (атрибут type/class). Далее каждое ребро (from,to) преобразуется в связь между классами: fromClass = class (from), toClass = class (to). После этого в AQL выполняется группировка COLLECT по (fromClass,toClass) и расчёт веса потока через AGGREGATE (COUNT/SUM). На выходе формируется таблица потоков (fromClass, toClass, value), которая затем преобразуется в формат Санки (nodes + links) для визуализации диаграммы Санки (рисунок 3).

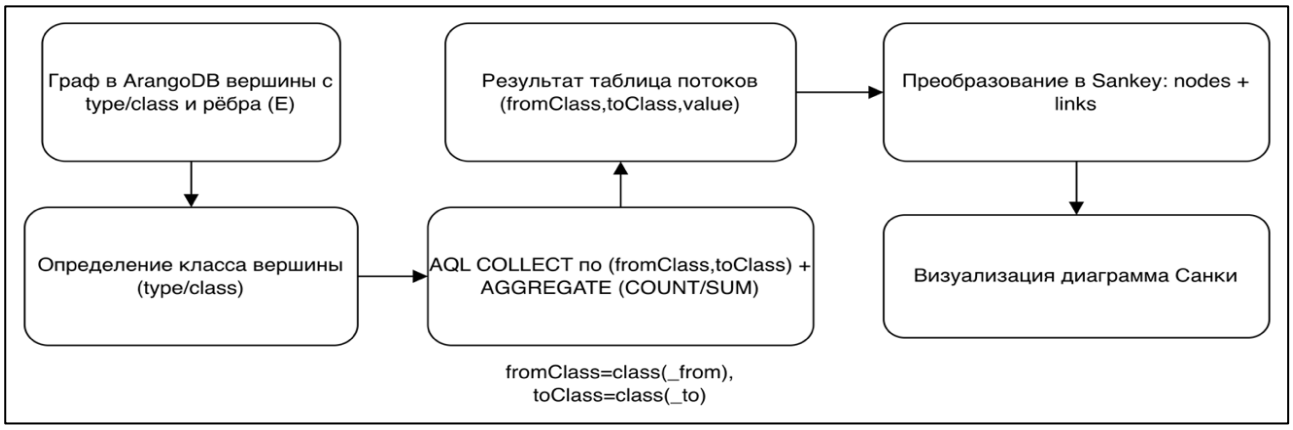

Рисунок 3 – Агрегирование по типам / классам вершин

Агрегирование по типам/классам позволяет существенно сократить число узлов и связей по сравнению с агрегацией на уровне отдельных вершин, поэтому диаграмма Санки получается более «читаемой» и отражает общую структуру взаимодействий между категориями. При этом детализация по конкретным объектам теряется, но для задач обзорной аналитики (выявление доминирующих направлений потоков и ключевых связок между группами) такой уровень обобщения обычно оказывается достаточным.

В третьей схеме исходный граф в ArangoDB используется для извлечения многошаговых цепочек длиной *k* вместо прямого учёта отдельных рёбер. Далее каждый найденный путь раскладывается на переходы между соседними шагами (*vi*→*vi*+1), чтобы получить набор элементарных «переходов по шагам». После этого в AQL выполняется группировка COLLECT по паре (*stepFrom*, *stepTo*) и расчёт веса потока через AGGREGATE (COUNT/SUM) для суммарной оценки повторяемости/величины таких переходов. На выходе формируется таблица потоков (*stepFrom*, *stepTo*, *value*), которая преобразуется в формат Санки (nodes + links) и используется для построения диаграммы Санки (рисунок 4).

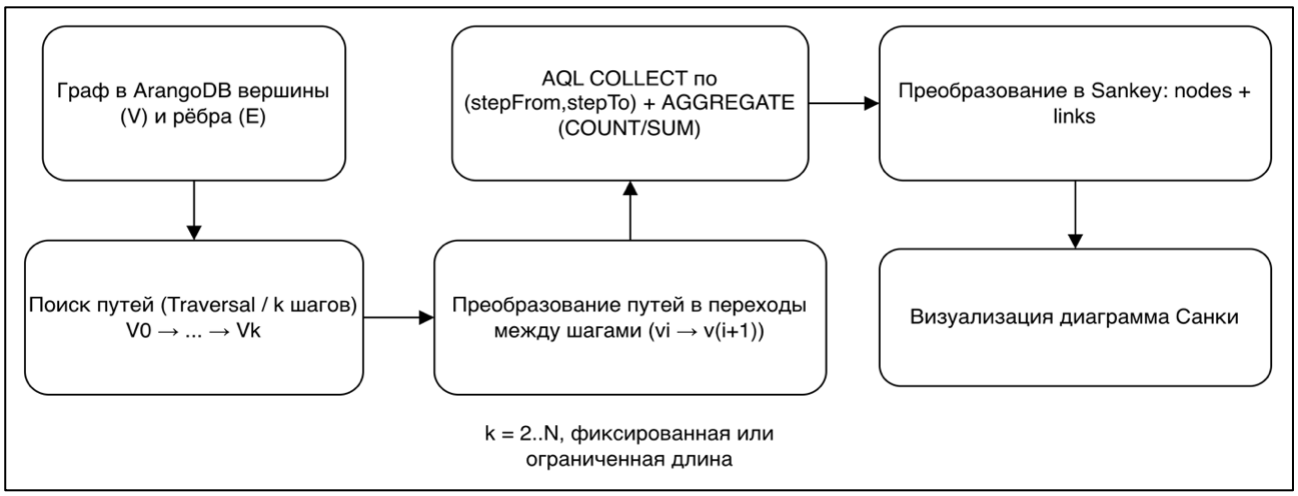

Рисунок 4 – Агрегирование по путям

Параметр k задаёт глубину анализа: увеличение числа шагов позволяет выявлять не только прямые, но и косвенные связи (например, «A влияет на C через B»), однако повышает объём перебора путей и требует ограничения глубины и/или условий отбора при обходе графа. Поэтому на практике для построения диаграммы Санки выбирают фиксированный диапазон k и при необходимости вводят фильтры/ограничения на этапе поиска путей, чтобы итоговая схема оставалась интерпретируемой.

После выполнения агрегации (по рёбрам, по классам или по путям) мы получаем «плоский» результат – таблицу потоков вида *from* → *to* → *value*. Чтобы построить Санки, этот результат нужно привести к двум коллекциям: nodes – список всех уникальных узлов, которые встречаются в полях from/to, и links – список связей, где для каждой строки таблицы создаётся объект {source, target, value}.

Практически сопоставление делается так: сначала собираются уникальные идентификаторы узлов и фиксируется способ адресации. В d3-sankey допускается, что source/target заданы как индекс узла или как строковый id (если настроен nodeId), поэтому удобно сохранять исходные id вершин/классов и не вводить лишние преобразования.

Выводы: В статье выполнен обзор существующих подходов к агрегированию графовых данных (прямое агрегирование по рёбрам, агрегирование по классам/типам и агрегирование по путям), а также рассмотрены средства ArangoDB для обходов графа и агрегаций. Также проанализированы инструменты визуализации потоков (в частности, d3-sankey, Plotly и BI-инструменты) и их требования к структуре входных данных, что позволило зафиксировать практические ограничения и критерии сопоставления результатов агрегации с целевым форматом.

А также разработаны схемы агрегирования графовых данных ArangoDB для построения потоков: схема 1 (прямое агрегирование по рёбрам), схема 2 (агрегирование по типам/классам вершин) и схема 3 (агрегирование по путям с разложением цепочек на переходы между шагами). Для каждой схемы описаны основные этапы обработки и получаемый результат в виде таблицы потоков, после чего приведены правила преобразования к структурам nodes и links, необходимым для построения диаграммы Санки

Список литературы:

1. ArangoDB. Grouping and aggregating data in AQL (Examples and query patterns) [Электронный ресурс]. – Режим доступа: https://docs.arangodb. com/3.11/aql/examples-and-query-patterns/grouping/ (дата обращения: 12.01.2026).

2. Библиотека d3-sankey для построения диаграмм Санки [Электронный ресурс]. – Режим доступа: https://github.com/d3/d3-sankey (дата обращения: 12.01.2026).

3. Plotly.py. Sankey diagram [Электронный ресурс]. – Режим доступа: https://github.com/plotly/plotly.py/blob/main/doc/python/sankey-diagram.md (дата обращения: 12.01.2026).

4. Google Looker Studio. Sankey chart reference [Электронный ресурс]. – Режим доступа: https://docs.cloud.google.com/looker/docs/studio/sank ey-chart-reference (дата обращения: 12.01.2026)