



Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)

КАФЕДРА «Системы обработки информации и управления» (ИУ5)

## **РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ**

***НА ТЕМУ:***

**«Нагрузочное тестирование метаграфового  
хранилища»**

Студент группы ИУ5-34М

\_\_\_\_\_ И. А. Ерохин

Руководитель

\_\_\_\_\_ Ю. Е. Гапанюк

2022 г.

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой ИУ5  
(Индекс)

В.И. Терехов  
(И.О.Фамилия)

«    »      20      г.

## З А Д А Н И Е

### на выполнение научно-исследовательской работы

по теме Нагрузочное тестирование метаграфового хранилища

Студент группы ИУ5-34М

Ерохин Иван Алексеевич  
(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.)  
учебная

Источник кафедра тематики кафедра (кафедра, предприятие, НИР)

График выполнения НИР: 25% к 4 нед., 50% к 8 нед., 75% к 12 нед., 100% к 17 нед.

**Техническое задание:** Разработать и реализовать методику нагрузочного тестирования метаграфового хранилища и проанализировать полученные результаты

### Оформление научно-исследовательской работы:

Расчетно-пояснительная записка на 16 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

Дата выдачи задания «1» октября 2022 г.

Руководитель НИР

Ю.Е. Гапанюк  
(Подпись, дата) (И.О.Фамилия)

Студент

И.А. Ерохин  
(Подпись, дата) (И.О.Фамилия)

## ***Оглавление***

|  |    |
|--|----|
| Введение.....                                | 4  |
| Метаграфовая модель .....                    | 5  |
| Хранение элементов метаграфовой модели ..... | 7  |
| Изменяемость элементов.....                  | 9  |
| Темпоральность.....                          | 9  |
| Схема Данных.....                            | 10 |
| Эксперименты .....                           | 11 |
| Выводы.....                                  | 14 |
| Литература .....                             | 15 |

## ***Введение***

В настоящее время число работ тем или иным способом взаимодействующих со сложными графами, в целом, и с метаграфовой моделью, в частности, стремительно увеличивается. Но инструменты для последующей обработки данных подобных моделей крайне редки. Этим фактом обусловлена необходимость разработки подобных инструментов.

На данный момент времени, нет исследований, определяющих оптимальный способ хранения данных метаграфовой модели. Вследствие чего, крайне остро стоит вопрос о разработке нагрузочного тестирования для определения качества разрабатываемых архитектур хранения данных.

## Метаграфовая модель

Согласно источникам [6], [7], [8] можно выделить следующие особенности построения метаграфа.

Метаграф определяется следующим образом:

$$MG = \langle V, MV, E, ME \rangle$$

$MG$  – метаграф;

$V$  – множество вершин метаграфа;

$MV$  – множество метавершин метаграфа;

$E$  – множество ребер метаграфа,

$ME$  – множество метаребер метаграфа.

Вершина метаграфа характеризуется множеством атрибутов:

$$v_i = \{atr_k\}, v_i \in V$$

$v_i$  – вершина метаграфа;

$atr_k$  – атрибут.

Ребро метаграфа характеризуется множеством атрибутов, исходной и конечной вершиной:

$$e_i = \langle v_S, v_E, \{atr_k\} \rangle, e_i \in E$$

$e_i$  – ребро метаграфа;

$v_S$  – исходная вершина (метавершина) ребра;

$v_E$  – конечная вершина (метавершина) ребра; .

$atr_k$  – атрибут.

Для формулирования понятия метавершины вводится дополнительное понятие фрагмент метаграфа, который в общем виде может содержать произвольные вершины (метавершины) и ребра.

$$MG = \{ev_j\}, ev_j \in (V \cup E \cup MV \cup ME)$$

$MG_i$  – фрагмент метаграфа;

$ev_j$  – элемент, принадлежащий объединению множеств вершин, метавершин, ребер и метаребер метаграфа.

Метавершина метаграфа:

$$mv_i = \langle \{atr_k\}, MG_i \rangle, mv_i \in MV$$

$mv_i$  – метавершина метаграфа;

$atr_k$  – атрибут,

$MG_i$  – фрагмент метаграфа.

Таким образом, метавершина обладает не только свойствами вершины, но и включает в себя вложенный фрагмент метаграфа. Также, благодаря наличию собственных атрибутов и связей с другими вершинами, метавершина увеличивает несводимость собственного понятия к общности элементов входящих в состав метавершины.

Таким образом резюмируя итоги изучения метаграфа можно сказать, что метаграф позволяет естественным образом моделировать сложные иерархические зависимости и может включать в свой состав вершины, метавершины и ребра. Определение вершин соответствует таковому в обычных графовых структурах. Метавершина позволяет выделять фрагмент графа (метаграфа), аннотировать его дополнительными свойствами, проводить к нему ребра, что в свою очередь придает понятию новое качество и увеличивает несводимость понятия к сумме его составных частей. Ребро метаграфа может соединять вершины внутри одной метавершины, вершины между различными метавершинами, метавершины, вершины и метавершины.

## *Хранение элементов метаграфовой модели*

Для хранения вершин, метавершин и ребер используется общая сущность. В этом случае сущность используется для хранения вершины-предиката со стереотипом, который может соответствовать вершине, метавершине или ребру. Одним из преимуществ данного варианта является уменьшение количества сущностей, что позволяет упростить программную реализацию

Для обеспечения программной реализации введем следующие ER-атрибуты «вершины-предиката»:

1. `nodeid` – числовой первичный ключ, используемый для создания схемы данных.
2. `id` – уникальный строковый ключ вершины-предиката.
3. `nodetype` – тип вершины-предиката.

Необходимо отметить, что для «метавершины» и «вершины» не следует вводить различные статусы, так как в случае добавления к вершине «вершин-предикатов» нижнего уровня, данная вершина автоматически превращается в метавершину.

Сущность «вершина-предикат» содержит следующие связи с другими сущностями:

1. Соединения между вершинами, метавершинами, ребрами и метаребрами.
2. Множество атрибутов, принадлежащих метавершине / вершине / ребру / метаребру.

Для обеспечения соединения между вершинами, метавершинами, ребрами и метаребрами введем сущность «точка\_соединения». Для хранения данных будем использовать сущность «тип точки-соединения», также для физической схемы данных будем использовать название «`linktypes`». Таблица содержит:

1. linkid – числовой первичный ключ, используемый для создания схемы данных.
2. nodeid1 – числовой вторичный ключ первой вершины-предиката.
3. nodeid2 – числовой вторичный ключ второй вершины-предиката.
4. id1 – уникальный строковый ключ первой вершины-предиката.
5. id2 – уникальный строковый ключ второй вершины-предиката.
6. linktype – тип связи между первой и второй вершинами-предикатами.

Для хранения атрибутов может быть использована отдельная сущность. Сущность «атрибут», в соответствии с определением, содержит следующие ER-атрибуты:

1. attrid – числовой первичный ключ, используемый для создания схемы данных.
2. nodeid – числовой вторичный ключ, используемый для связи с сущностью «вершина-предикат».
3. id – уникальный строковый ключ вершины-предиката, к которой принадлежит атрибут.
4. system – логическое значение. Если значение истинно, то атрибут считается «системным», то есть предназначен для хранения параметров метаграфовой модели. Если значение ложно, то атрибут предназначен для хранения данных моделируемой предметной области.
5. name – наименование атрибута.

Также предполагается использование системных атрибутов, которые предназначены для хранения параметров элементов метаграфовой модели. К ним можно отнести такие параметры как:

1. Наименование метавершины/вершины/ребра/метаребра.
2. Текстовое описание метавершины/вершины/ребра/метаребра.
3. Признак направленности ребра.



## ***Изменяемость элементов***

Под изменяемостью элементов метаграфовой модели будем понимать способность элементов модели изменяться с течением времени.

Для обозначения изменяемости элементов метаграфовой модели будем использовать традиционные термины, применяемые в функциональном программировании для обозначения изменяемости переменных: изменяемые (mutable) и неизменяемые (immutable).

Метавершины/вершины/ребра/метаребра не могут быть изменяемыми, так как не содержат изменяемых элементов, они могут или присутствовать или отсутствовать в базе данных.

Точки соединения между вершинами, метавершинами и ребрами также не могут быть изменяемыми, они могут или присутствовать или отсутствовать в базе данных (immutable).

Атрибуты могут быть изменяемыми. Мы предполагаем, что имя и тип данных атрибута неизменны. Но значение атрибута может меняться во времени.

## ***Темпоральность***

Под темпоральностью метаграфовой СУБД будем понимать способность отслеживания изменений в метаграфовых данных с течением времени.

Реализация темпоральности в обязательном порядке предполагает наличие шкалы времени. Могут быть использованы следующие варианты работы с временной меткой:

1. Если временная метка не задана явно, то в качестве временной метки используется текущее значение даты и времени сервера (с учетом временного пояса).
2. При добавлении данных создается символическое значение временной метки. Символическое значение ассоциируется с текущим

на момент создания метки значением даты и времени сервера (с учетом временного пояса).

Для хранения символических значений временных меток используется сущность «временная\_метка» (также для физической схемы данных будем использовать название «time\_table»), которая содержит следующие ER-атрибуты:

1. tid - числовой первичный ключ, используемый для создания схемы данных.
2. time\_label – символическое значение временной метки.
3. time – значение временной метки – текущее значение даты и времени сервера (с учетом временного пояса).

### ***Схема Данных***

В результате рассмотренных выше способов представления элементов метаграфовой модели была сформированна схема данных представленных на рисунке 1.

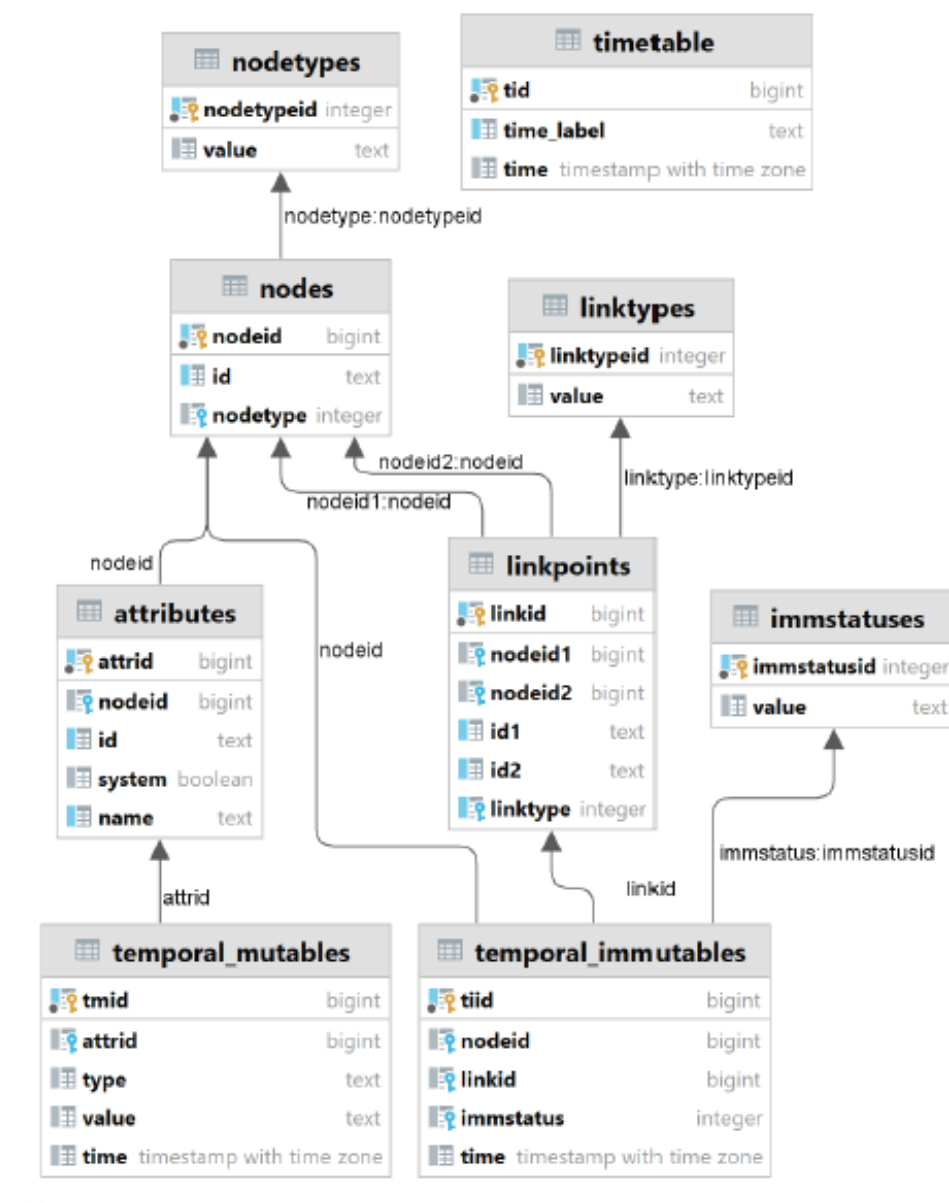


Рис 1 – Схема данных для описания метаграфовой модели.

## Эксперименты

Для того чтобы оценить производительность разработанного хранилища, была проведена серия экспериментов. Методика эксперимента подразумевает масштабирования количества элементов одного типа метаграфовой модели при фиксации количества остальных элементов, с последующим снятием характеристик. Основными характеристиками в данном эксперименте является время загрузки и выгрузки данных. В результате работы была проведена серия из пяти экспериментов в результате

которых были получены данные для изменения количества вершин, метавершин, ребер, метаребер и атрибутов.

Результаты экспериментов представлены в таблице 1. В колонке 2 показаны четыре типа элементов модели в виде радиолокационной диаграммы, количество которых фиксировано в текущем эксперименте. Столбцы 3 и 4 содержат зависимости времени чтения и записи, соответственно, элементов пятого типа, в зависимости от количества элементов.

Эксперименты проводились на компьютере с процессором AMD Ryzen 9 3950X, 16-ядерным процессором 3,50 ГГц и 64 Гб оперативной памяти с использованием PostgreSQL 14.

| No. | Experiment configuration | Load time dependency | Save time dependency |
|-----|--------------------------|----------------------|----------------------|
| 1   |                          |                      |                      |
| 2   |                          |                      |                      |
| 3   |                          |                      |                      |
| 4   |                          |                      |                      |
| 5   |                          |                      |                      |

Таблица 1 – результаты нагрузочного тестирования

## ***Выводы***

Основываясь на результатах проведенных экспериментов, можно сделать следующие выводы:

1. Наиболее важным результатом является то, что длительность загрузки и сохранения находится в линейной зависимости от количества элементов метаграфа.
2. Также видно, что хранилище метаграфов может использоваться с моделями метаграфов, состоящими из нескольких тысяч элементов (т.е. узлов, метавершин, ребер и метаребер).
3. Внезапно запись метаграфа в хранилище обычно происходит быстрее, чем его чтение. Поэтому необходимо найти способ оптимизировать операцию чтения.
4. Временные зависимости от метавершин и узлов выглядят аналогично. Временные зависимости от конечных ребер метаребер также выглядят аналогично. Таким образом, есть хорошая перспектива для эффективного взаимодействия с вложенными структурами.
5. В настоящее время производительность разработанного прототипа хранилища выглядит не очень высокой, около 2000 базовых элементов модели (узлов, метавершин, ребер и метаребер) считываются примерно за минуту, а записываются примерно за 2- 2,5 минуты, при этом скорость обработки атрибутов выглядит несколько лучше. Поэтому, чтобы еще больше повысить производительность, необходимо доработать встроенные механизмы PostgreSQL и провести подробные эксперименты с различными вариантами индексации таблиц.

## *Литература*

1. Basu, A. Metagraphs and their applications [Text] / A. Basu, R. W. Blanning. Springer, 2007. — 172 p.
2. Черненький В.М., Гапанюк Ю.Е., Ревунков Г.И., Терехов В.И., Каганов Ю.Т. Метаграфовый подход для описания гибридных интеллектуальных информационных систем. Прикладная информатика. 2017. № 3 (69). Том 12. С. 57–79.
3. Самохвалов Э.Н., Ревунков Г.И., Гапанюк Ю.Е. Использование метаграфов для описания семантики и прагматики информационных систем. Вестник МГТУ им. Н.Э. Баумана. Сер. «Приборостроение». 2015. Выпуск №1.
4. Гапанюк Ю.Е., Ревунков Г.И., Федоренко Ю.С. Предикатное описание метаграфовой модели данных. Информационно-измерительные и управляющие системы. 2016. Выпуск № 12. С. 122–131.
5. Voloshin, V. Introduction to Graph and Hypergraph Theory [Text] / V. Voloshin. —Nova Science Publishers, 2009. — 287 p
6. Попков, В. К. Математические модели связности. Том 1: Графы и сети [Текст] / В. К. Попков. — Новосибирск : ИВМиМГ (ВЦ) СО РАН, 2000. —174 с.
7. Попков, В. К. Математические модели связности. Том 2: Гиперграфы и гиперсети [Текст] / В. К. Попков. — Новосибирск : ИВМиМГ (ВЦ) СО РАН, 2001. — 180 с.
8. Попков, В. К. Математические модели связности. Том 3: Представления графов [Текст] / В. К. Попков. — Новосибирск : ИВМиМГ (ВЦ) СО РАН, 2002. —169 с.
9. Johnson, J. Hypernetworks in the Science of Complex Systems [Text] / J. Johnson.— Imperial College Press, 2013. — 348 p. — (Series on Complexity Science).

- 10.Черненко В.М., Гапанюк Ю.Е., Ревунков Г.И., Терехов В.И., Каганов Ю.Т. Метаграфовый подход для описания гибридных интеллектуальных информационных систем. Прикладная информатика. 2017. № 3 (69). Том 12. С. 57–79.
- 11.Самохвалов Э.Н., Ревунков Г.И., Гапанюк Ю.Е. Использование метаграфов для описания семантики и прагматики информационных систем. Вестник МГТУ им. Н.Э. Баумана. Сер. «Приборостроение». 2015. Выпуск №1.
- 12.Гапанюк Ю.Е., Ревунков Г.И., Федоренко Ю.С. Предикатное описание метаграфовой модели данных. Информационно-измерительные и управляющие системы. 2016. Выпуск № 12. С. 122–131.
- 13.The structure and dynamics of multilayer networks [Text] / S. Boccaletti [et al.] //Physics Reports. — 2014. — Nov. — Vol. 544, no. 1. — P. 1—122.
- 14.*Aleta, A.* Multilayer networks in a nutshell [Text] / A. Aleta, Y. Moreno // Annual Review of Condensed Matter Physics. — 2019. — Mar. — Vol. 10, no. 1. —P. 45—62.