

Воронова Наталья Михайловна

Старший преподаватель

Владимирский государственный университет имени Александра Григорьевича и
Николая Григорьевича Столетовых

Кованова Александра Сергеевна

Студент

Владимирский государственный университет имени Александра Григорьевича и
Николая Григорьевича Столетовых

Корж Николай Сергеевич

Студент

Владимирский государственный университет имени Александра Григорьевича и
Николая Григорьевича Столетовых

АЛГОРИТМЫ ОЦЕНКИ ПРОИЗВОДИТЕЛЬНОСТИ МОДУЛЯ РАБОТЫ С ДАННЫМИ

Аннотация: Целью работы является разработка алгоритмов оценки производительности модуля работы с данными (модели Single Persistence и Multi Persistence). Критерии оценки – время выполнения запросов и объём используемой памяти. В алгоритмах эмулировалась многопоточность запросов (Thread), для тестирования алгоритмов генерировались множества данных разного объёма. Анализ работы алгоритмов показал, что увеличение объема базы данных и количества одновременных запросов к ней не приводят к ошибкам при использовании модели Single Persistence в отличие от модели Multi Persistence. Модель Multi Persistence работает существенно быстрее модели Single Persistence, но при увеличении размеров базы данных или количества запросов к базе данных, запросы завершаются с ошибкой Out Of Memory.

Ключевые слова: модуль работы с данными, оценка производительности.

Keywords: data processing module, performance evaluation.

Модуль работы с данными предназначен для организации работы с реляционными базами данных в J2EE-приложениях. Основным интерфейсом является StorageManager, который предоставляет API для работы с хранилищами данных в приложениях (рисунок 1).

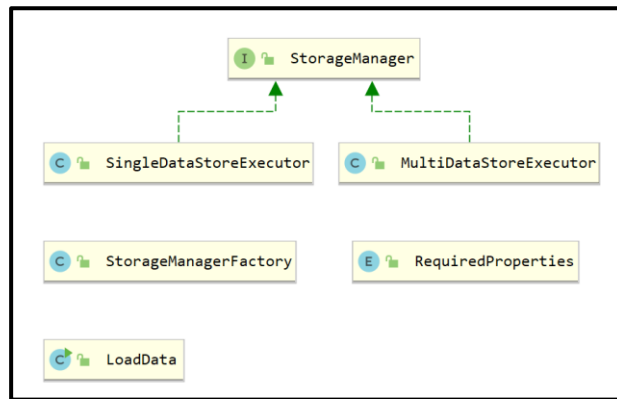


Рисунок 1. Диаграмма классов модуля работы с данными

Модуль реализует две модели доступа к данным – Single Persistence и Multi Persistence. В модели Single Persistence все запросы к данным обрабатываются одним экземпляром **StorageManager** (однопоточный доступ к данным). В модели Multi Persistence запросы к данным обрабатываются разными экземплярами **StorageManager** (многопоточный доступ к данным). Интерес представляет исследование производительности обеих моделей, а именно – как скажется на времени работы и как будет меняться объем используемой памяти при увеличении размера базы данных и при увеличении количества одновременных запросов на получение данных.

Для тестирования алгоритмов производительности модуля работы с данными использовалась тестовая база данных **university**. генерация датасетов производилась в таблице **t_department** (кафедры).

Оценка времени обработки запросов

Для оценки времени обработки запросов к базе данных для моделей Single Persistence и Multi Persistence были разработаны алгоритмы (рисунки 2,3). Замеры времени выполнялись для запросов к таблице **t_department** со 100, 1000, 10000, 100000 записями. Многопоточный доступ эмулировался для 10, 100, 1000, 10000 потоков выполнения (Thread). Результаты для каждого потока записывались в .csv-файл. Результаты экспериментов представлены в таблицах 1,2.

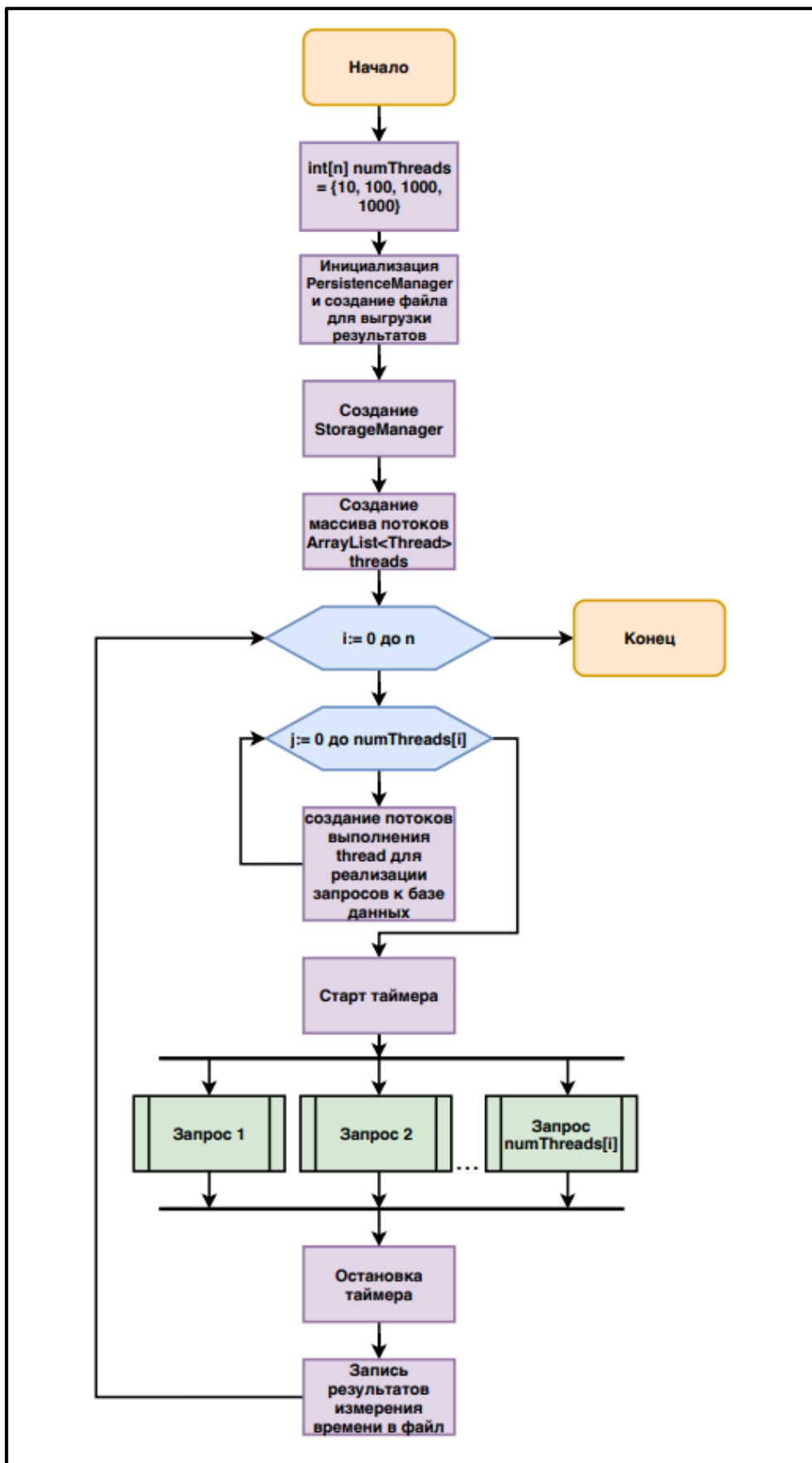


Рисунок 2. Блок-схема алгоритма оценки времени выполнения запроса к базе данных для модели Single Persistence

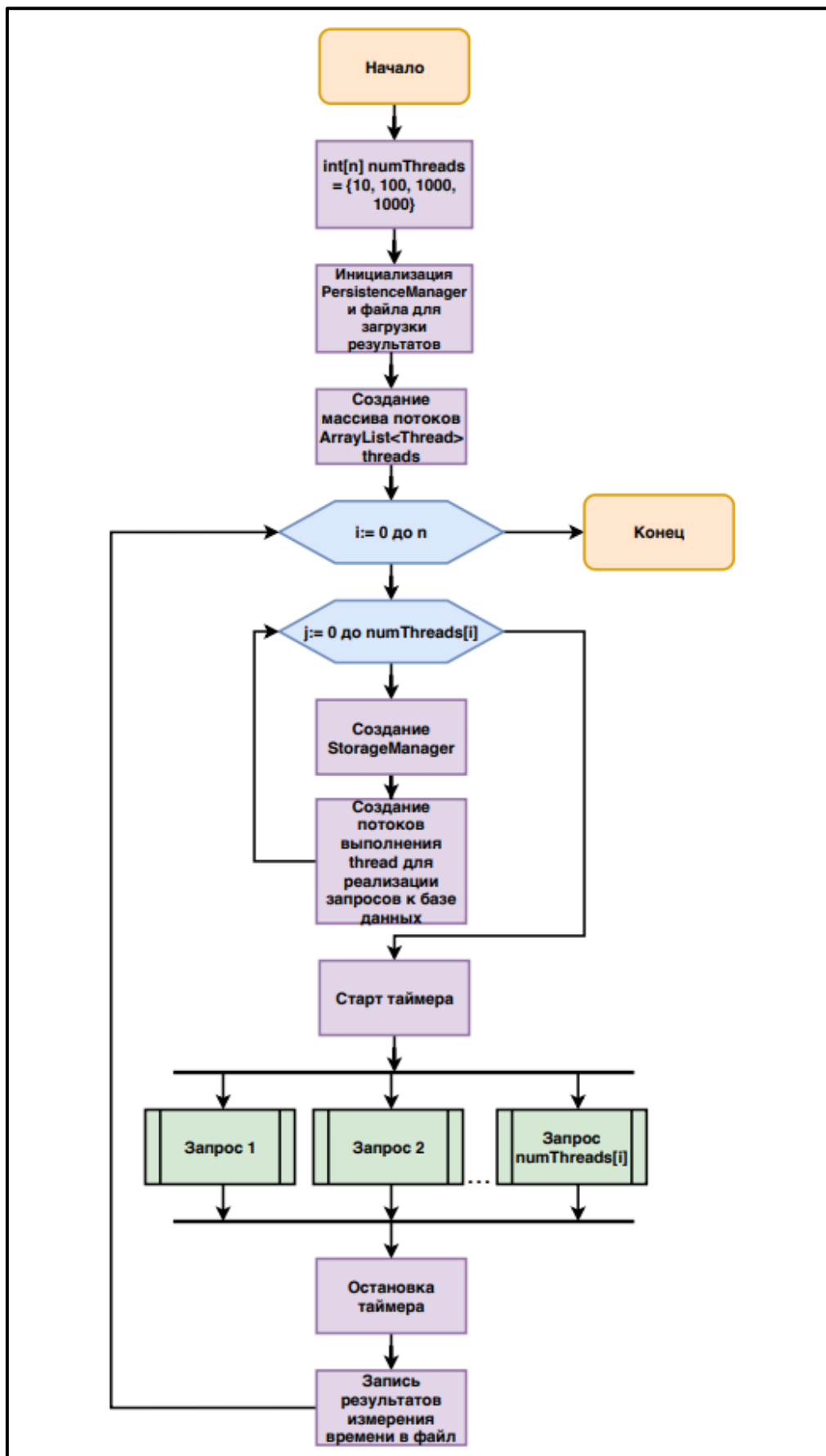


Рисунок 3. Блок-схема алгоритма оценки времени выполнения запроса к базе данных для модели Multi Persistence

Таблица 1.

**Результаты проведения экспериментов по оценке времени выполнения запросов
в базу данных для модели Single Persistence**

Количество записей в таблице t_department	Время работы, с			
	Количество потоков			
	10	100	1000	10000
100	0,2	0,97	6,18	50,31
1000	0,4	2,58	21,41	203,42
10000	2,13	15,743	148,21	1484,93
100000	9,17	82,71	910,15	8040,32

Таблица 2.

**Результаты проведения экспериментов по оценке времени выполнения запросов
в базу данных для модели Multi Persistence**

Количество записей в таблице t_department	Время работы, с			
	Количество потоков			
	10	100	1000	10000
100	0,19	0,38	1,9	11,49
1000	0,32	1,05	5,37	103,91
10000	1,17	4,7	79,69	OutOfMemory
100000	6,61	OutOfMemory	OutOfMemory	OutOfMemory

Оценка объема используемой памяти

Для оценки объёма используемой памяти для моделей Single Persistence и Multi Persistence были модифицированы алгоритмы (рисунки 2,3) – вместо замеров времени в соответствующих блоках выполняется оценка занимаемой памяти.

Измерения используемой памяти выполнялись для запросов к таблице t_department со 100, 1000, 10000, 100000 записями. Многопоточный доступ эмулировался для 10, 50, 100, 200, 400, 600, 800, 1000 потоков выполнения (Thread). Результаты для каждого потока записывались в .csv-файл. Результаты экспериментов представлены в таблицах 3,4. Аббревиатура OOM обозначает OutOfMemory.

Таблица 3.

Результаты эксперимента для модели Single Persistence

Количество записей в таблице t_department	Используемая память, Мб							
	Количество потоков							
	10	50	100	200	400	600	800	1000
100	13,66	39,14	10,49	48,87	30,08	54,72	16,44	17,57
1000	25,39	36,77	54,3	33,98	50	46	21,92	33,95
10000	50,45	94,89	96,73	102,09	113,89	33,69	60,71	103,68
100000	553,65	501,05	402,89	267,13	519,95	283,8	382,1	425,33

Таблица 4.

Результаты эксперимента для модели Multi Persistence

Кол-во записей в таблице t_department	Используемая память, Мб							
	Количество потоков							
	10	50	100	200	400	600	800	1000
100	23,4	18	30,5	81,3	69,4	153,6	276,8	246
1000	46,7	69,4	243,6	443,4	775,8	841,3	1300	987,6
10000	180	676	1187,9	1313,6	1095	1329,5	1415	1447,7
100000	928,9	1 394	OOM	OOM	OOM	OOM	OOM	OOM

Для наглядности полученные результаты экспериментов представлены на графиках. На рисунках 4-7 представлены графики зависимости времени выполнения запросов от количества потоков. На рисунках 8-11 представлены графики зависимости количества, используемой приложением памяти при выполнении запросов от количества потоков.

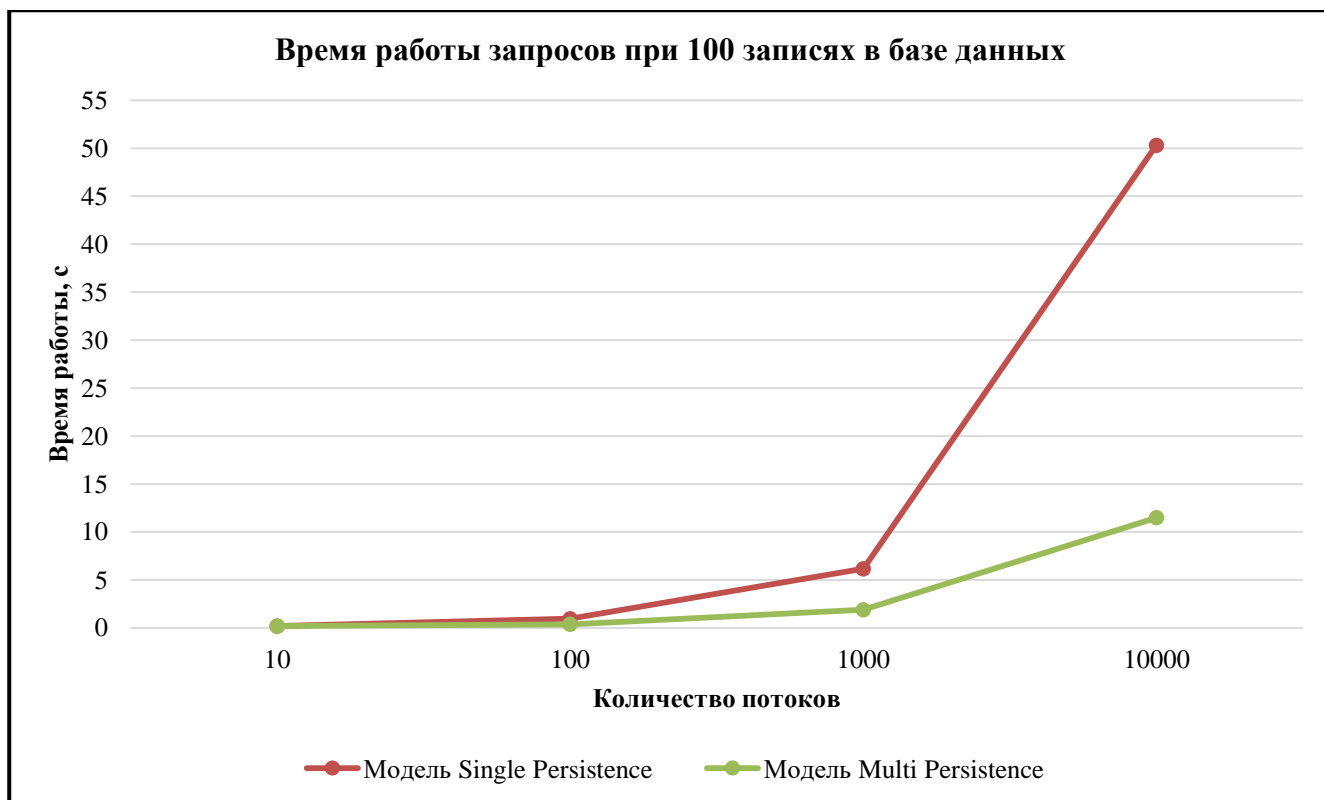


Рисунок 4. Время работы запросов при 100 записях в таблице

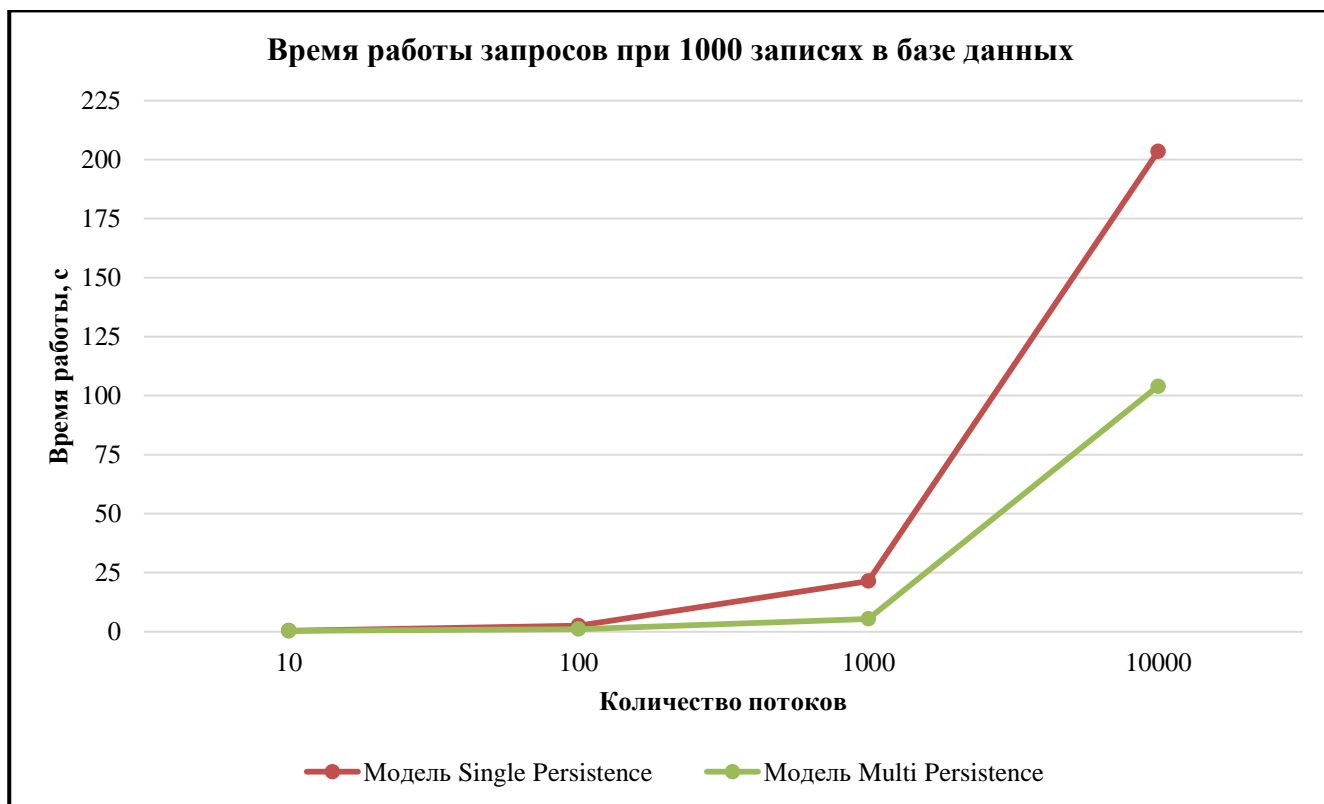


Рисунок 5. Время работы запросов при 1000 записях в таблице

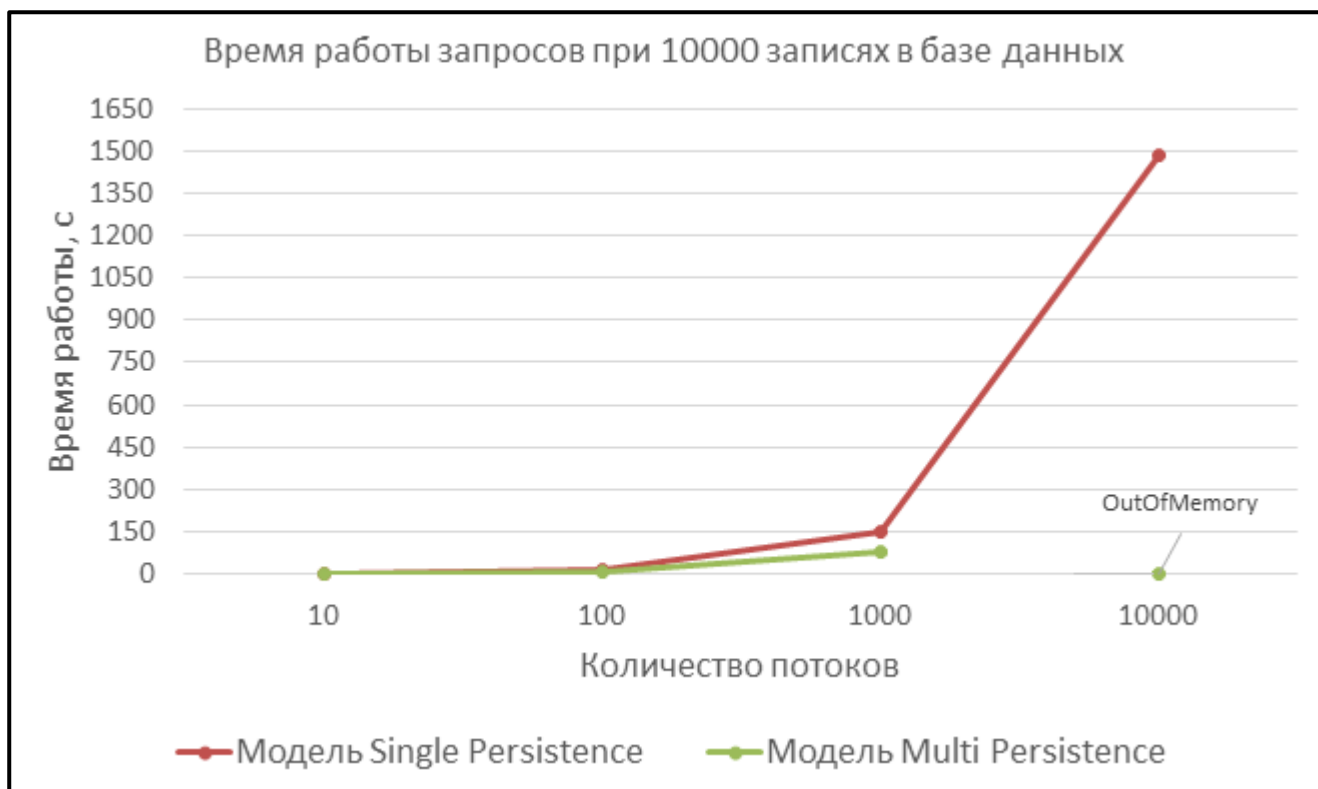


Рисунок 6. Время работы запросов при 10000 записях в таблице

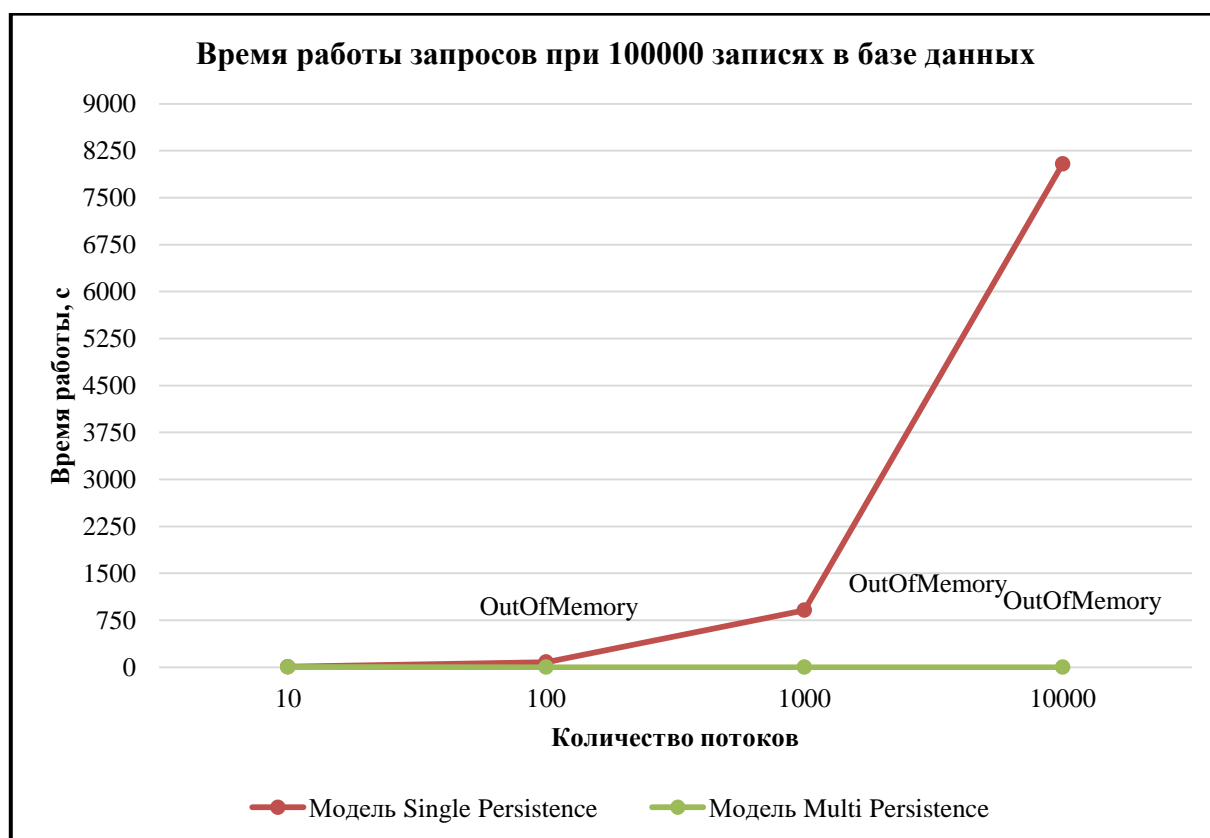


Рисунок 7. Время работы запросов при 100000 записях в таблице

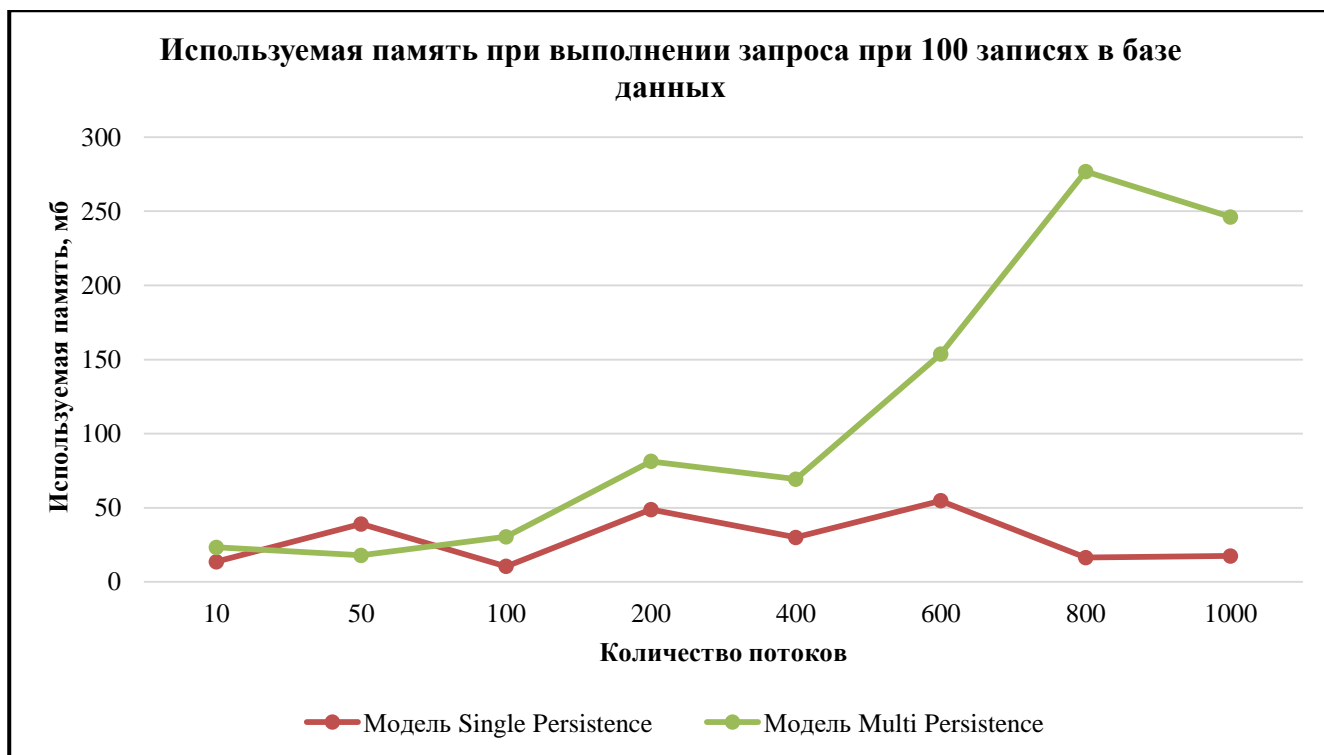


Рисунок 8. Использование памяти при выполнении запросов при 100 записях в таблице

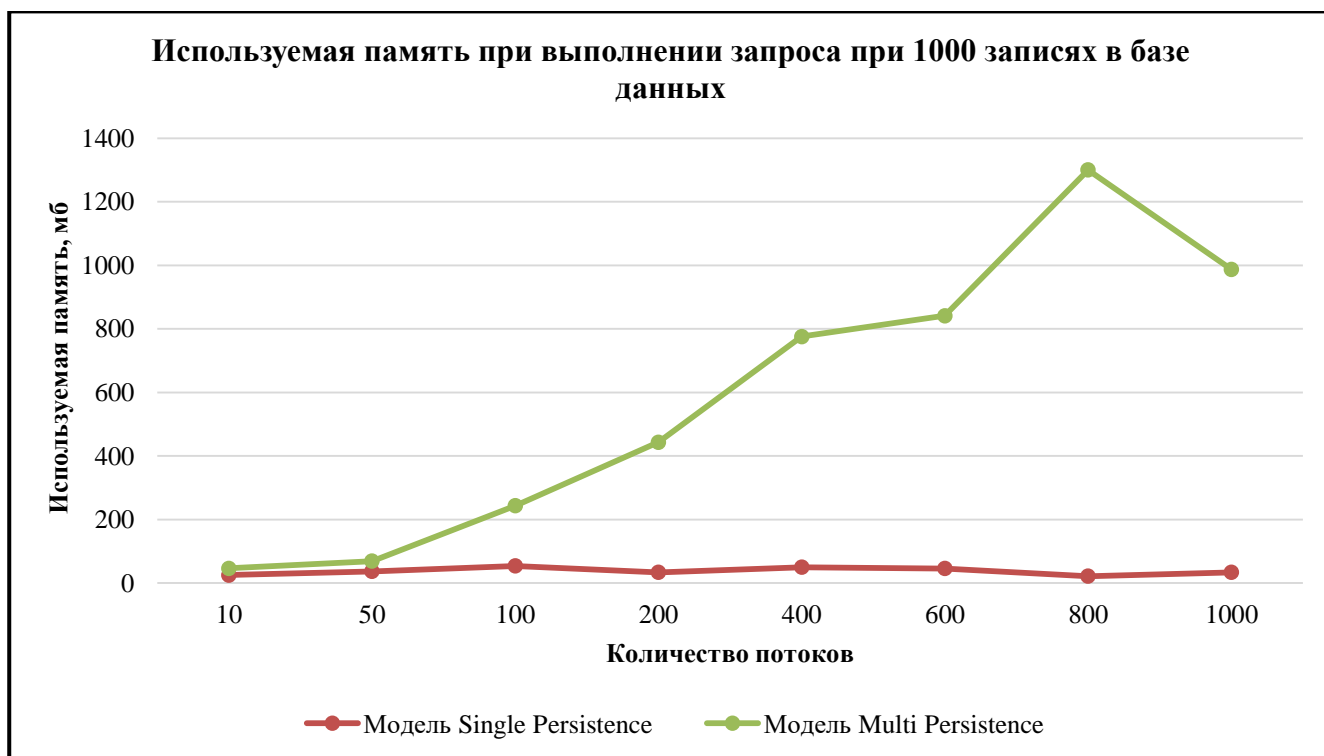


Рисунок 9. Использование памяти при выполнении запросов при 1000 записях в таблице

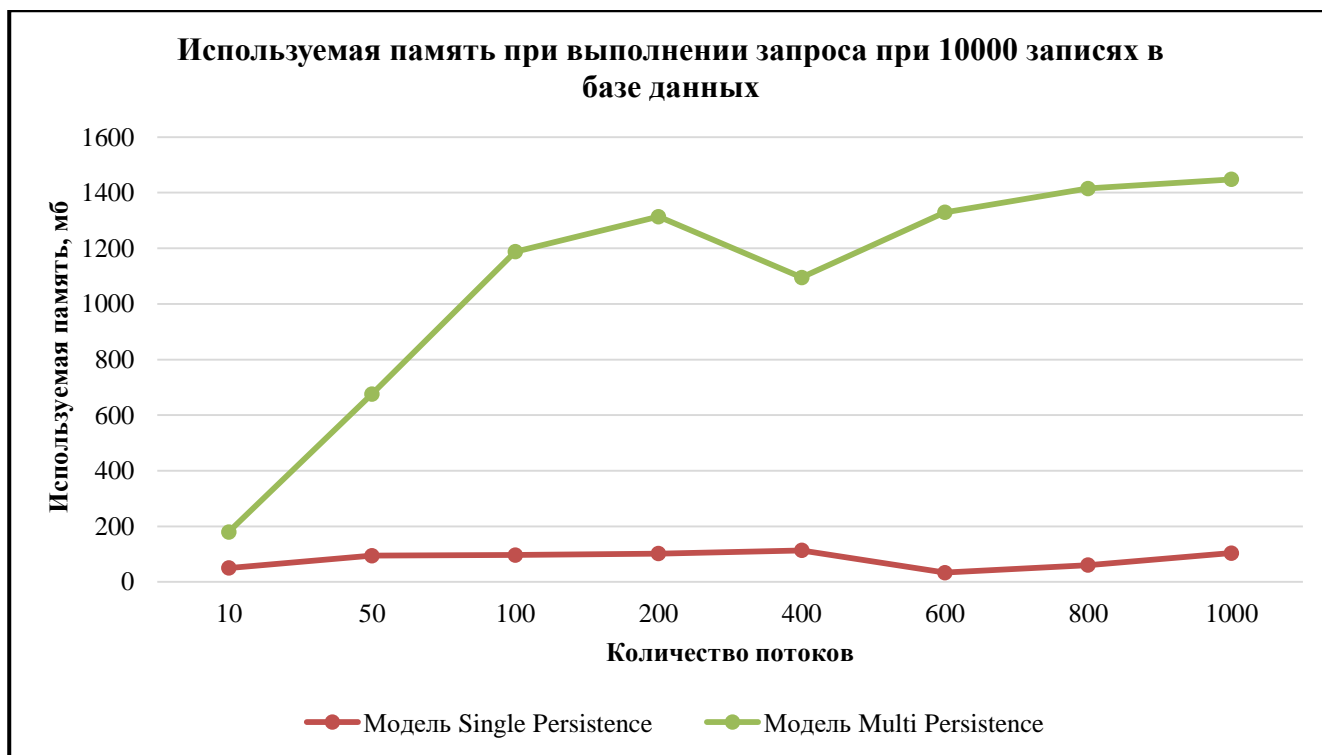


Рисунок 10. Использование памяти при выполнении запросов при 10000 записях в таблице

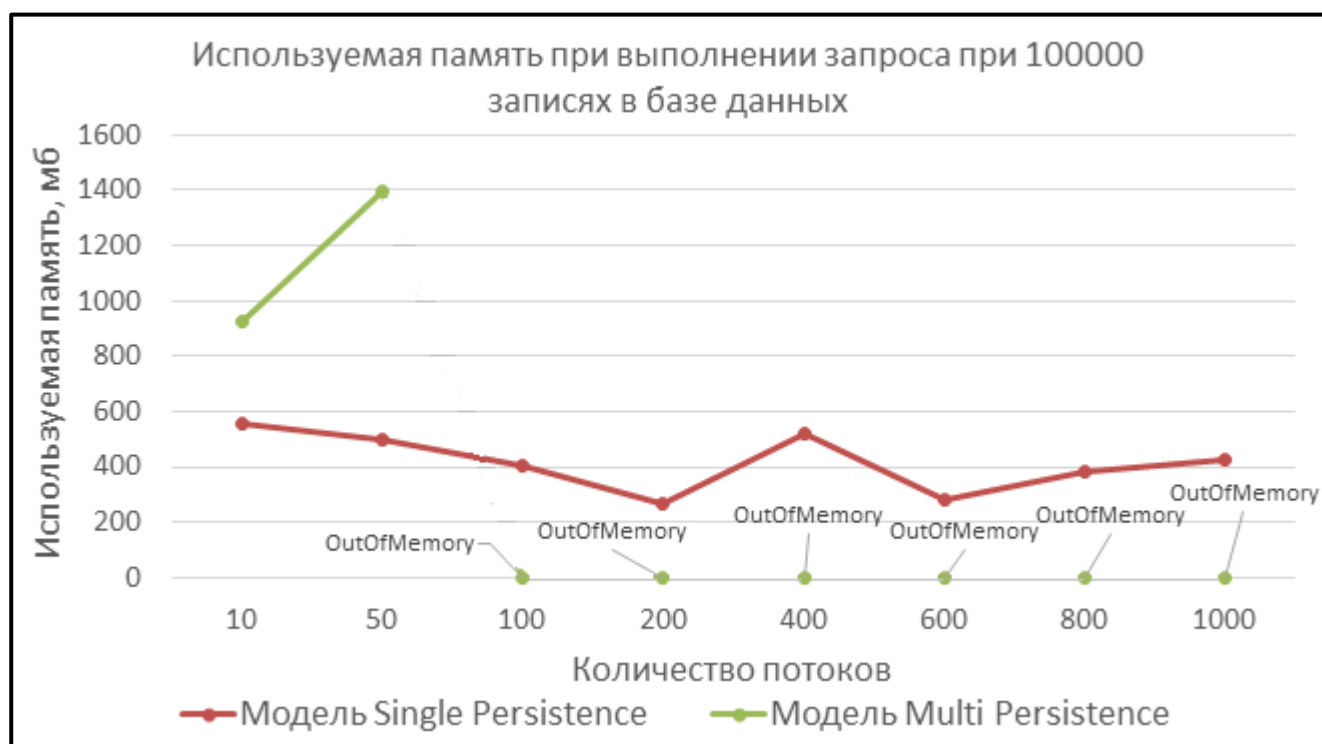


Рисунок 11. Использование памяти при выполнении запросов при 100000 записях в таблице

Обе модели показывают допустимое время ожидания выполнения запросов получения данных при 100 записях в таблице и до 1000 одновременных запросов к базе данных. При

этом время на обработку запросов для модели Multi Persistence значительно меньше (~2 секунды против ~6 секунд для модели Single Persistence при 1000 запросах к базе данных).

С увеличением числа записей до 1000 модель Single Persistence показывает допустимые результаты только при одновременных 100 запросах, в то время как модель Multi Persistence и при одновременных 1000 запросах показывает допустимое время ожидания выполнения запроса ~5 с.

При 10000 записях в таблице модель Single Persistence показывает допустимые результаты только при одновременных 10 запросах, в то время как модель Multi Persistence показывает хорошие результаты при одновременных 100 запросах к базе данных ~5 с. Обе модели показывают долгое время ожидания выполнения запроса при > 1000 одновременных запросов, и запросы данных при 10000 потоках завершаются с ошибкой OutOfMemory.

Модель Single Persistence при 100000 записей в базе данных работает очень медленно, в то время как запросы с моделью Multi Persistence с более чем 10 потоками завершаются ошибкой OutOfMemory.

Для модели Single Persistence не наблюдается рост количества используемой памяти при увеличении числа запросов к базе данных. Для модели Multi Persistence, так как для каждого нового запроса создается свой объект Storage Manager, при увеличении числа запросов наблюдается сильный рост используемой памяти. Использование памяти приложением с моделью Multi Persistence может превышать использование памяти приложением с моделью Single Persistence в 10 и более раз.

При 100000 записях в таблице и более чем 50 одновременных запросов работа приложения, использующего модель Multi Persistence, завершается с ошибкой OutOfMemory.

По результатам работы алгоритмов оценки производительности можно сделать следующие выводы:

- модель Multi Persistence справляется с запросами к базе данных существенно быстрее модели Single Persistence, но при этом требует намного большего количества оперативной памяти (использование памяти приложением с моделью Multi Persistence может превышать использование памяти приложением с моделью Single Persistence в 10 и более раз).
- увеличение объема базы данных и количества одновременных запросов к ней не приводят к ошибкам при использовании модели Single Persistence в отличие от модели Multi Persistence. Для модели Multi Persistence при увеличении размеров базы данных или количества запросов к базе данных, запросы завершаются с ошибкой Out Of Memory.

Литература:

1. Thread – Java Documentation [Электронный ресурс]. URL: <https://docs.oracle.com/javase/7/docs/api/java/lang/Thread.html> (Дата обращения: 15.07.2021)