|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА − Российский технологический университет»**  **РТУ МИРЭА** |

Институт информационных технологий (ИИТ)

Кафедра математического обеспечения и стандартизации информационных технологий (МОСИТ)

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ**

по дисциплине «Распределенные системы управления базами данных»

**Практическое занятие № 5**

|  |  |  |  |
| --- | --- | --- | --- |
| Студент группы ИНБО-01-17 | *ИКБО-11-22, Гришин Андрей Валерьевич* | (подпись) | |
| Преподаватель | *Красников Степан Альбертович* | (подпись) | |
| Отчет представлен | «\_\_\_»\_\_\_\_\_\_\_\_202\_\_г. | |  | |

Москва 2025 г.

**1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

Данные записываются в Cassandra таким образом, чтобы обеспечить полную надежность и высокую производительность. Но в связи с тем, что данная СУБД является распределенной, уходит некоторое время на обработку запросов и передачу информации между узлами. Время обработки запросов называется задержкой. Причинами больших «задержек записи» или “write latency” (при выполнении операций вставки и обновления) могут быть следующие факторы:

⎯ Низкая скорость записи дисков, на которых расположена СУБД;

⎯ Большой размер батча;

⎯ Запись строки большого размера. В идеале размер строки не должен превышать 1 мегабайта;

⎯ Сетевая задержка, когда пропускная способность сети низка;

⎯ Отсутствие свободных потоков для записи;

⎯ Загруженность центрального процессора;

⎯ Работа «сборщика мусора» (garbage collector).

Что касается «задержки чтения» или “read latency”, то тут гораздо больше возможных причин ее возникновения:

⎯ Большой размер раздела (partition). При создании таблиц в Cassandra важно уметь создать правильный ключ раздела, чтобы разделы не были слишком большими;

⎯ Плохой запрос. Сюда входит полное сканирование таблицы, поиск по ключу, который не является ключом раздела (ALLOW FILTERING);

⎯ Количество затронутых sstable;

⎯ Использование в качестве ключей кластеризации слишком уникальных (почта, номер телефона) или полей с малым количеством возможных значений (пол);

⎯ Уровень согласованности. При большом уровне необходимо дождаться ответа бо́льшего количества узлов;

⎯ Большое количество tombstone;

⎯ Большое количество записей в секунду.

**2 ПРАКТИЧЕСКАЯ ЧАСТЬ**

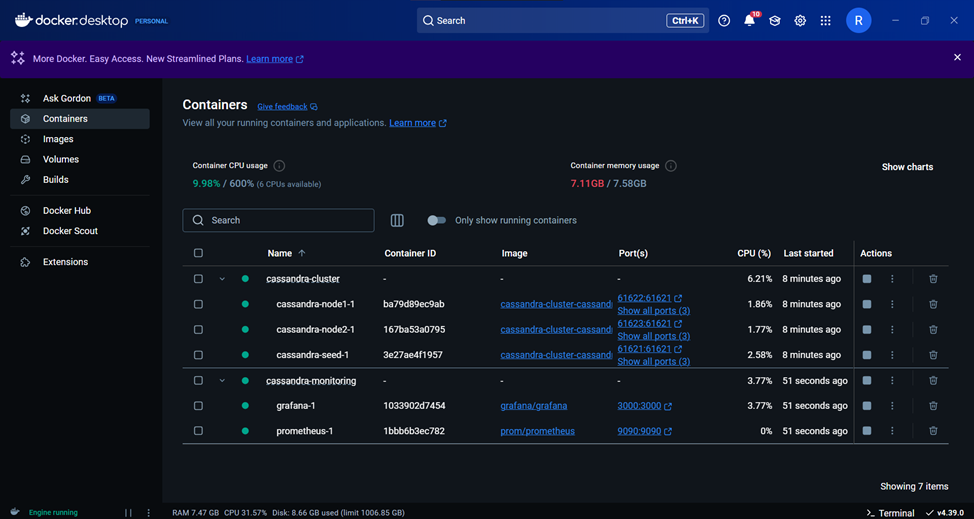


Рисунок 1 – Запуск кластеров

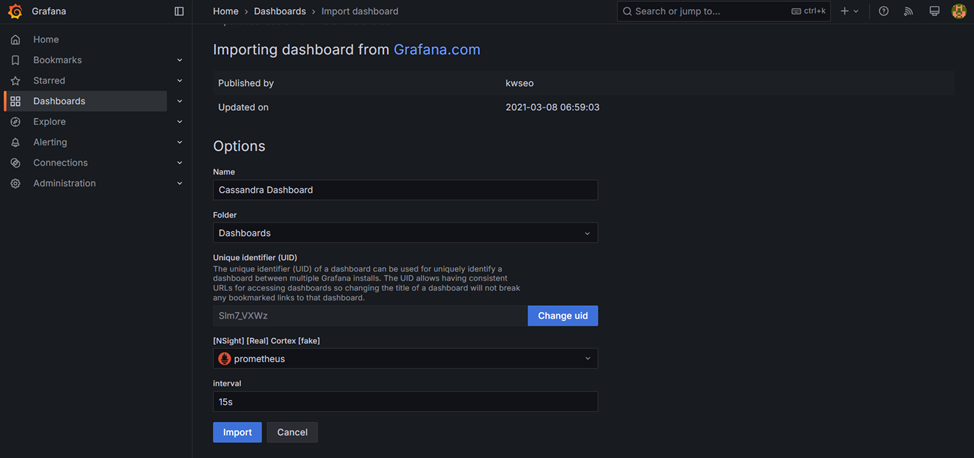


Рисунок 2 – Импорт панели “Cassandra Dashboard”

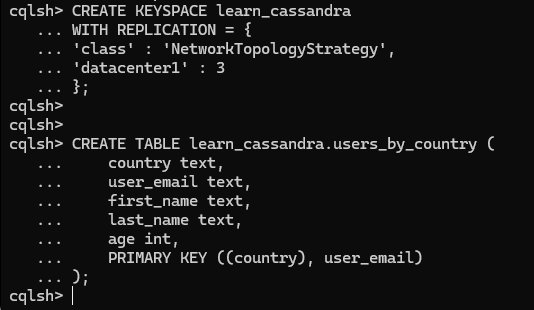


Рисунок 3 – Создание keyspace и таблицы для дальнейшего использования

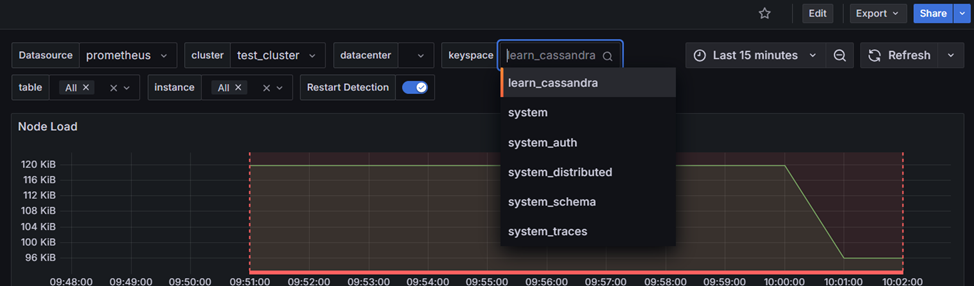


Рисунок 4 – Выбор пространства ключей в Grafana

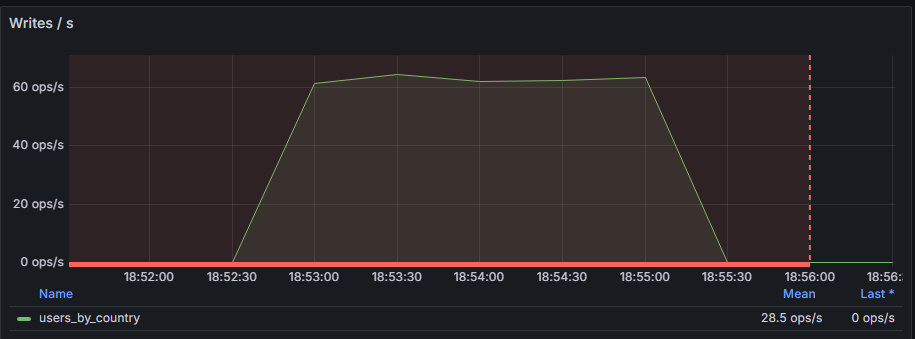


Рисунок 5 – график Writes для однопоточной вставки

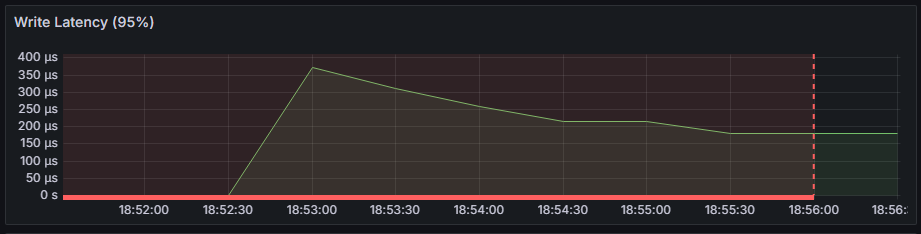


Рисунок 6 – график Write Latency (95%) для однопоточной вставки

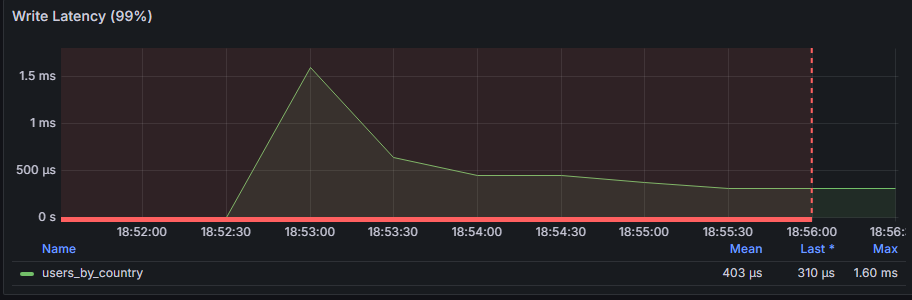


Рисунок 7 – график Write Latency (99%) для однопоточной вставки



Рисунок 8 – график Writess для многопоточной вставки

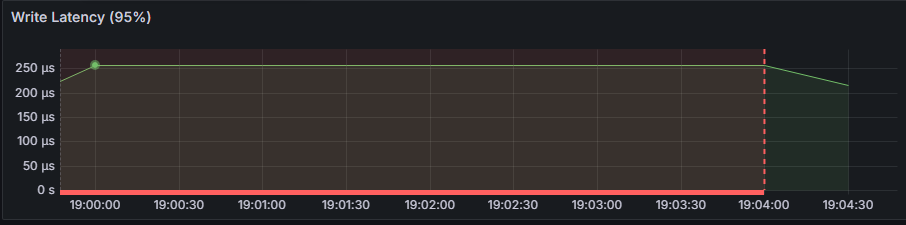


Рисунок 9 – график Write Latency (95%) для многопоточной вставки

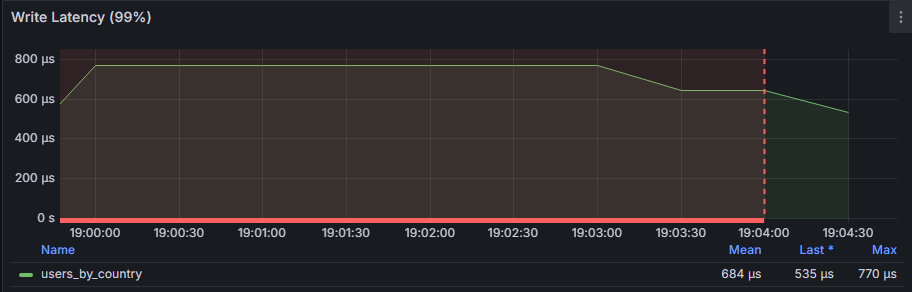


Рисунок 10 – график Write Latency (99%) для многопоточной вставки

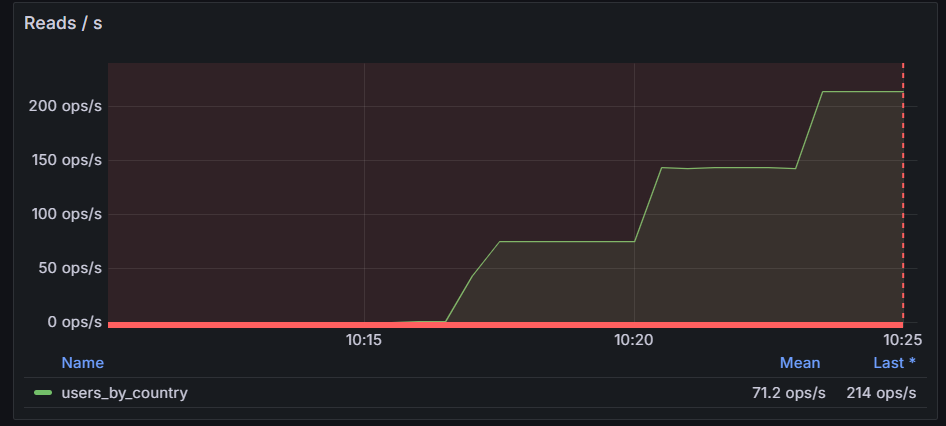


Рисунок 11 – график Reads

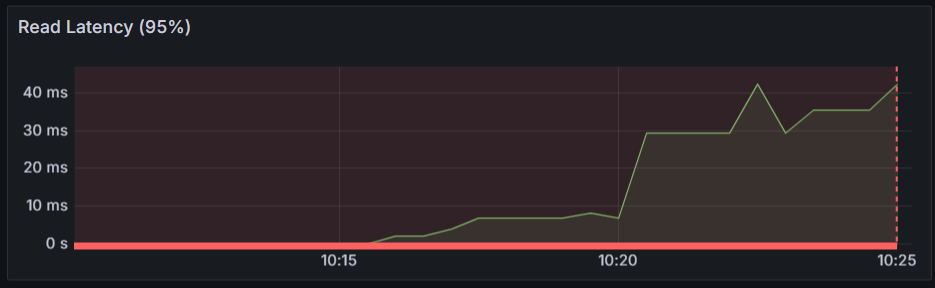


Рисунок 12 – график Read Latency (95%)

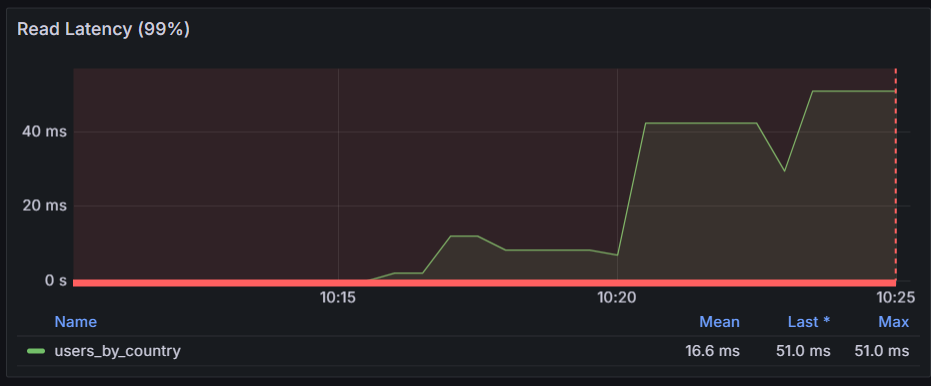


Рисунок 13 – график Read Latency (99%)

**Заключение:**

В ходе данной работы были изучены и протестированы показатели задержки чтения (read latency) и задержки записи (write latency) в распределенных СУБД.

Для эксперимента была настроена Grafana для визуализации метрик, а также создан keyspace и таблица для выполнения операций вставки и чтения данных.

Проведенные измерения позволили оценить производительность системы и выявить влияние различных факторов на задержки обработки запросов. Полученные графики наглядно продемонстрировали изменения показателей в зависимости от нагрузки на систему.

**Вопросы для самопроверки:**

1. **Какие аналоги Prometheus и Grafana можно использовать в качестве временного хранилища данных для статистики?**
   * **VictoriaMetrics** – быстрая и эффективная альтернатива Prometheus с поддержкой PromQL.
   * **InfluxDB** – специализированная TSDB с мощным API для работы с временными рядами.
   * **OpenTSDB** – хранилище временных рядов, работающее поверх HBase.
   * **TimescaleDB** – расширение для PostgreSQL, оптимизированное для временных данных.
2. **Можно ли получить графическую визуализацию по какому-либо параметру, используя только Prometheus?**

Нет, Prometheus не предоставляет встроенных средств графической визуализации. Он поддерживает базовую отрисовку графиков в веб-интерфейсе, но для полноценной визуализации рекомендуется использовать Grafana или другие инструменты.