

## ЛР7. Мониторинг

### 1. Мониторинг в PostgreSQL

Мониторинг баз данных — это комплексная задача, которая включает в себя отслеживание показателей производительности баз данных на различных уровнях: SQL (оптимальность запросов), экземпляр БД, инфраструктура, пользователи.

Виды мониторинга:

- мониторинг доступности БД;
- мониторинг подключенных клиентов и их активности;
- мониторинг работы с данными;
- мониторинг запросов;
- мониторинг фоновых процессов;
- мониторинг системных метрик.

В PostgreSQL существует подсистема накопительной статистики, которая позволяет собирать статистику о работе всех компонентов СУБД и предоставляет интерфейс в виде набора хранимых процедур и представлений для просмотра статистики. В настоящее время в ней отслеживаются обращения к таблицам и индексам как на уровне блоков на диске, так и на уровне отдельных строк. Также подсчитывается общее число строк в каждой таблице, и собирается информация о выполняемых для каждой таблицы действиях очистки и анализа.

Для просмотра текущего состояния системы предназначены несколько предопределённых представлений, которые перечислены в таблице.

Спецификация	Описание
pg_stat_activity	Одна строка для каждого серверного процесса с информацией о текущей активности процесса, включая его состояние и текущий запрос.
pg_stat_replication	По одной строке для каждого процесса-передатчика WAL со статистикой по репликации на ведомом сервере, к которому подключён этот процесс.
pg_stat_wal_receiver	Одна строка со статистикой приёмника WAL, полученной с сервера, на котором работает приёмник.

pg_stat_subscription	Одна строка для подписки, сообщающая о рабочих процессах подписки.
pg_stat_gssapi	Одна строка для каждого подключения, в которой показывается информация об использовании аутентификации и шифровании GSSAPI для данного подключения.
pg_stat_progress_analyze	По одной строке с текущим состоянием для каждого обслуживающего процесса, в котором работает ANALYZE.
pg_stat_progress_create_index	По одной строке с текущим состоянием для каждого обслуживающего процесса, в котором выполняется CREATE INDEX или REINDEX.

В дополнение к рассмотренным есть несколько других представлений, позволяющих просмотреть собранную статистику. Кроме того, на базе существующих функций накопительной статистики можно создать собственные представления.

Спецификация	Описание
pg_stat_archiver	Одна строка со статистикой работы процесса архивации WAL.
pg_stat_database	Одна строка для каждой базы данных со статистикой на уровне базы.
pg_stat_all_tables	По одной строке на каждую таблицу в текущей базе данных со статистикой по обращениям к этой таблице.
pg_stat_all_indexes	По одной строке для каждого индекса в текущей базе данных со статистикой по обращениям к этому индексу.
pg_statio_all_tables	По одной строке для каждой таблицы в текущей базе данных со статистикой по операциям ввода/вывода с этой таблицей.
pg_statio_all_indexes	По одной строке для каждого индекса в текущей базе данных со статистикой по операциям ввода/вывода для этого индекса.

Модуль `pg_stat_statements` предоставляет возможность отслеживать статистику выполнения сервером всех операторов SQL. Этот модуль нужно загружать, добавив `pg_stat_statements` в `shared_preload_libraries` в файле `postgresql.conf`, так как ему требуется дополнительная разделяемая память. Когда модуль `pg_stat_statements` загружается, он отслеживает статистику по всем базам данных на сервере. Для получения и обработки этой статистики этот модуль предоставляет представление `pg_stat_statements` и вспомогательные функции.

Можно мониторить:

- наиболее часто выполняемые запросы;
- самые долгие запросы;
- самые «тяжелые» запросы в плане использования ресурсов;
- запросы, которые возвращают большое число строк и т.д.

## 2. Prometheus

Prometheus – система мониторинга, которая изначально предназначалась для мониторинга контейнерных сред, но со временем возможности системы расширились до приложений, серверов, баз данных и виртуальных машин.

Prometheus предоставляет we-интерфейс с отображением графиков практически по любому сочетанию параметров, но сами графики не очень удобные, поэтому Prometheus используют вместе с Grafana – полноценным инструментом визуализации, который включает встроенную поддержку Prometheus. Grafana предоставляет список панельных модулей, позволяющих создавать визуализации с такими параметрами, как карты мира, тепловые карты, гистограммы, круговые и линейные диаграммы.

Сервер Prometheus (Prometheus Server) – центральный компонент системы мониторинга, чья основная задача — хранить и мониторить определенные объекты. В терминах Prometheus объекты мониторинга называются целевыми объектами (Prometheus targets). Через конечную точку HTTP-сервер Prometheus проверяет статус приложения.

Каждый элемент целевого объекта, который необходимо мониторить, называется метрикой. Prometheus собирает метрики на основе методов `push` и `pull`. В первом методе отслеживаемое приложение отвечает за отправку метрик в систему мониторинга с помощью Push шлюза (Pushgateway).

В методе pull приложение подготавливает метрики, а сервер Prometheus считывает целевые объекты с интервалом, который определяет сбор метрик, и сохраняет их в базе данных временных рядов (Time-Series database, TSDB). Целевые объекты и временной интервал считывания метрик задаются в конфигурационном файле prometheus.yml.

Для мониторинга сторонних систем (таких как сервер Linux, сервер MySQL и т.д.) можно использовать экспортеры. Экспортер—часть программного обеспечения, которое получает существующие метрики от сторонней системы и экспортирует их в формат, понятный серверу Prometheus.

Prometheus можно установить локально, либо воспользоваться Docker – это программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации. Docker позволяет «упаковать» приложение со всем его окружением и зависимостями в контейнер, который затем может быть развернут на системе или отдельном сервере. Пакетный менеджер Docker Compose позволяет описывать и запускать многоконтейнерные приложения.

Для запуска системы мониторинга Prometheus и системы визуализации Grafana в составе пакетного менеджера Docker Compose необходимо:

- 1) Создать файл docker-compose.yml, содержащий все необходимые контейнеры. Пример файла с контейнерами prom/prometheus и grafana/grafana. Предполагается, что конфигурационный файл prometheus.yml находится в каталоге prometheus/ относительно файла docker-compose.yml. Кроме указанных контейнеров, файл может содержать другие контейнеры-экспортеры, осуществляющие мониторинг сторонних систем (в т.ч. СУБД MySQL, PostgreSQL и т.д.).

```
version: '3.1'

volumes:
  prometheus_data: {}
  grafana_data: {}

services:
  prometheus:
    image: prom/prometheus
    volumes:
      - ./prometheus:/etc/prometheus/
      - prometheus_data:/prometheus
    command:
      - '--config.file=/etc/prometheus/prometheus.yml'
      - '--storage.tsdb.path=/prometheus'
      - '--web.console.libraries=/usr/share/prometheus/console_libraries'
      - '--web.console.templates=/usr/share/prometheus/consoles'
```

```

    - '--web.enable-lifecycle'
ports:
  - 9090:9090

grafana:
  image: grafana/grafana
  depends_on:
    - prometheus
  ports:
    - 3000:3000
  volumes:
    - grafana_data:/var/lib/grafana
    - ./grafana/dashboards:/var/lib/grafana/dashboards
    - ./grafana/provisioning:/etc/grafana/provisioning

```

- 2) В конфигурационном файле `prometheus/prometheus.yml` настроить адреса всех экспортеров, которые будут использоваться для мониторинга. Пример настройки конфигурационного файла:

```

global:
  scrape_interval:     15s
  evaluation_interval: 15s
  external_labels:
    monitor: 'db-monitoring'

scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:9090']

  - job_name: 'postgres'
    static_configs:
      - targets: ['HOST:PORT']

```

- 3) Запустить контейнеры командой из каталога, в котором расположен файл `docker-compose.yml`

```
docker-compose -p <имя пакета> up
```

- 4) После запуска Prometheus будет доступен по адресу <http://localhost:9090/>, Grafana - <http://localhost:3000/>

## **ЛР7. Вопросы для контроля**

1. Виды мониторинга СУБД.
2. PostgreSQL. Мониторинг текущего состояния системы.
3. PostgreSQL. Просмотр статистики.
4. Prometheus. Компоненты системы.
5. Prometheus. Возможности.

## **ЛР7. Задание**

1. Используя систему просмотра статистики PostgreSQL:
  - 1.1. Получить информацию обо всех активных подключениях к СУБД.
  - 1.2. Получить информацию о 5 таблицах с наибольшим количеством операций добавления записей.
  - 1.3. Получить информацию о 5 индексах с наибольшим количеством произведённых сканирований.
  - 1.4. Используя модуль `pg_stat_statements`, вывести информацию о самых долгих запросах (с наибольшим общим временем выполнения запроса).
  - 1.5. Используя модуль `pg_stat_statements`, вывести информацию о запросах, вернувших наибольшее число строк.
2. Настроить систему мониторинга Prometheus + Grafana. Подключить как минимум один экспортер метрик из СУБД по выбору (MySQL, PostgreSQL, MongoDB). Настроить панель мониторинга (дашборд) для отображения показателей работы СУБД.