

OTUS

Онлайн-образование

Не забыл включить
запись

Мониторинг

Меня хорошо видно && слышно?

Ставьте плюсы, если все хорошо
Напишите в чат, если есть проблемы

План

- Задачи мониторинга
- Для чего нужен мониторинг
- Паттерны и антипаттерны построения мониторинга
- Типы мониторинга
- Метрики
- Хранение данных
- Сбор данных
- Алертинг

Мониторинг для слабых духом

Ожидание от мониторинга

- Алертинг о сбоях в системе => Сбор информации о текущем состоянии
- Отражение состояния системы/сервера/сервиса/компонента сервиса
- Предупреждения о возможных сбоях и проблемах
- Визуализация и дашборды
- Предоставление информации для мониторинга “второго порядка”
- Отчеты

Для чего нужен мониторинг

- Повышение качества сервиса
- Оперативное реагирование
- Обучение на основе собранных данных
- Сбор обратной связи

Обратная связь

- Мониторинг всего (окружений, пользователей, сторонних сервисов)
- Автоматизация
- Chatops

Надежность имеет значение

Продуктом *НЕ* будут пользоваться если:

- не работает
- работает медленно
- критически важный функционал не исправен

Что мониторить?

- Инфраструктура
- Вспомогательные сервисы
- Качество работы и состояние приложения
- Бизнес метрики

Бизнес метрики vs. Обычные метрики

Антипаттерны в мониторинге

- Одержимость выбором инструментов
- "Специальный" отдел мониторинга
- Мониторинг для галочки
- Использование мониторинга в качестве "костыля"
- Ручное конфигурирование

Хорошие паттерны

- Мониторинг из многих компонентов как правило лучше, чем один монолит
- Мониторинг это не только про оповещения
- Построение мониторинга необходимо начинать с пользовательских метрик
- Если есть возможность, начинать лучше с мониторинга как сервис (Datadog, New Relic, okmeter.io), чем разворачивать свою инфраструктуру
- Постоянное улучшение всех компонентов

Типы мониторинга

WHITEBOX

BLACKBOX

Whitebox

Мониторинг, базирующийся на метриках, которое дает само приложение/сервис:

- через логи
- интерфейсы
- API
- SNMP

Blackbox

Мониторинг, оценивающий внешнее состояние сервиса/системы:

- ping
- HTTP request
- Открытый порт
- Наличие процессов

Метрики

Основные метрики системы:

- LoadAvg, CPU, Net (bps/pps), DISK Load
- потребление Mem/DISK
- “чистота” системных логов: dmesg, messages
- Актуальность состояния резервных копий

Метрики

Метрики сервиса

- “диагностический запрос”, который задействует все или большинство компонентов системы.
- время отклика/обработки запроса
- количество обращений в единицу времени
- количество одновременных обращений

Метрики

Метрика = временной
ряд

Метрики

```
Metrics: CPU, free_mem, net_rssi, battery
```

```
Tags: Host=MyServer, Region=West
```

```
Data:
```

```
2017-01-01 01:02:00 70 500 -40 80
2017-01-01 01:03:00 71 400 -42 80
2017-01-01 01:04:00 72 367 -41 80
2017-01-01 01:05:01 68 750 -54 79
```

Метрики

Характеристики временного ряда:

- Ориентированы на время - каждая запись имеет отметку времени
- Только добавление - данные только добавляются, но не обновляются
- Работа со свежими данными

Исторические данные (TimeSeries)

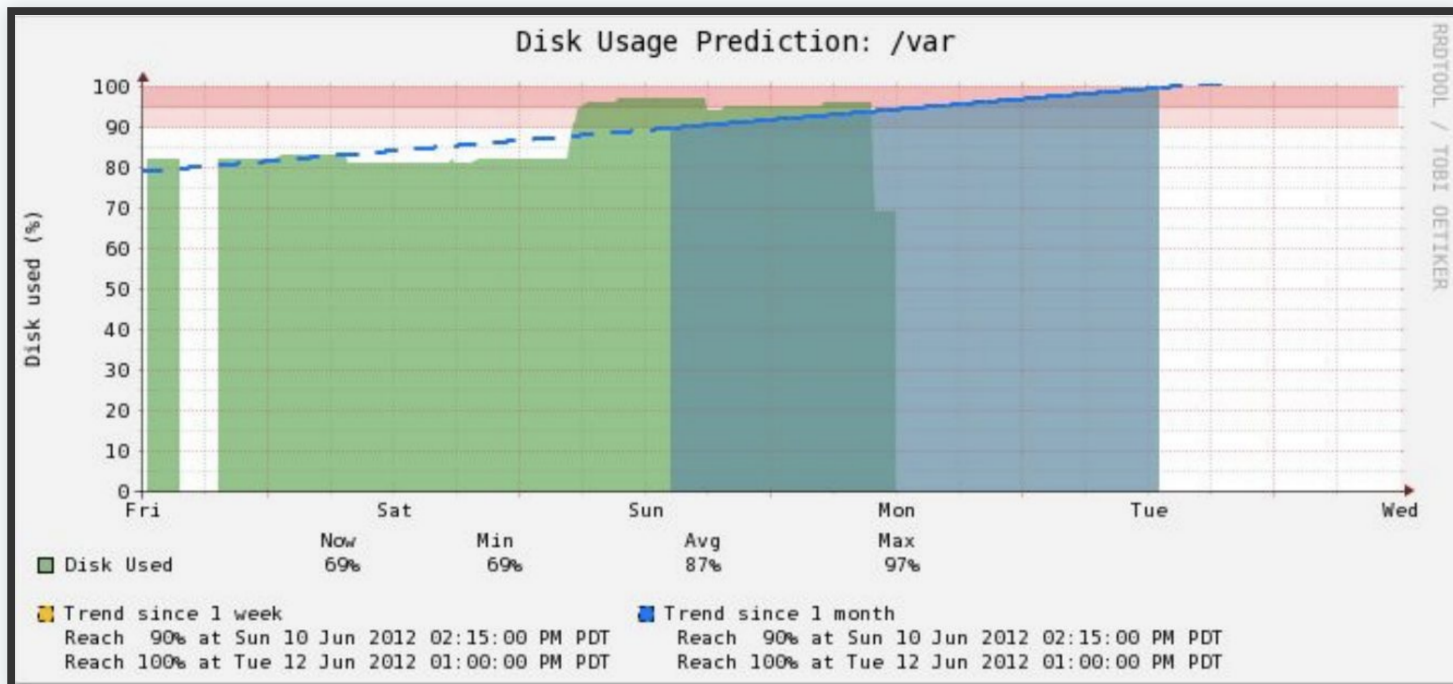
С историческими данными возникает несколько вопросов:

1. Как их собирать?
2. Где их хранить?
3. Сколько их хранить?
4. Как с ними работать?

Для чего нужны исторические данные

1. Анализ проблем
2. Гадание на графиках или Trend monitoring. Тут высокое разрешение не важно, т.к. временные рамки в которых идет “гадание” это недели, месяца и годы.

Пример исторических данных



Хранение исторических данных

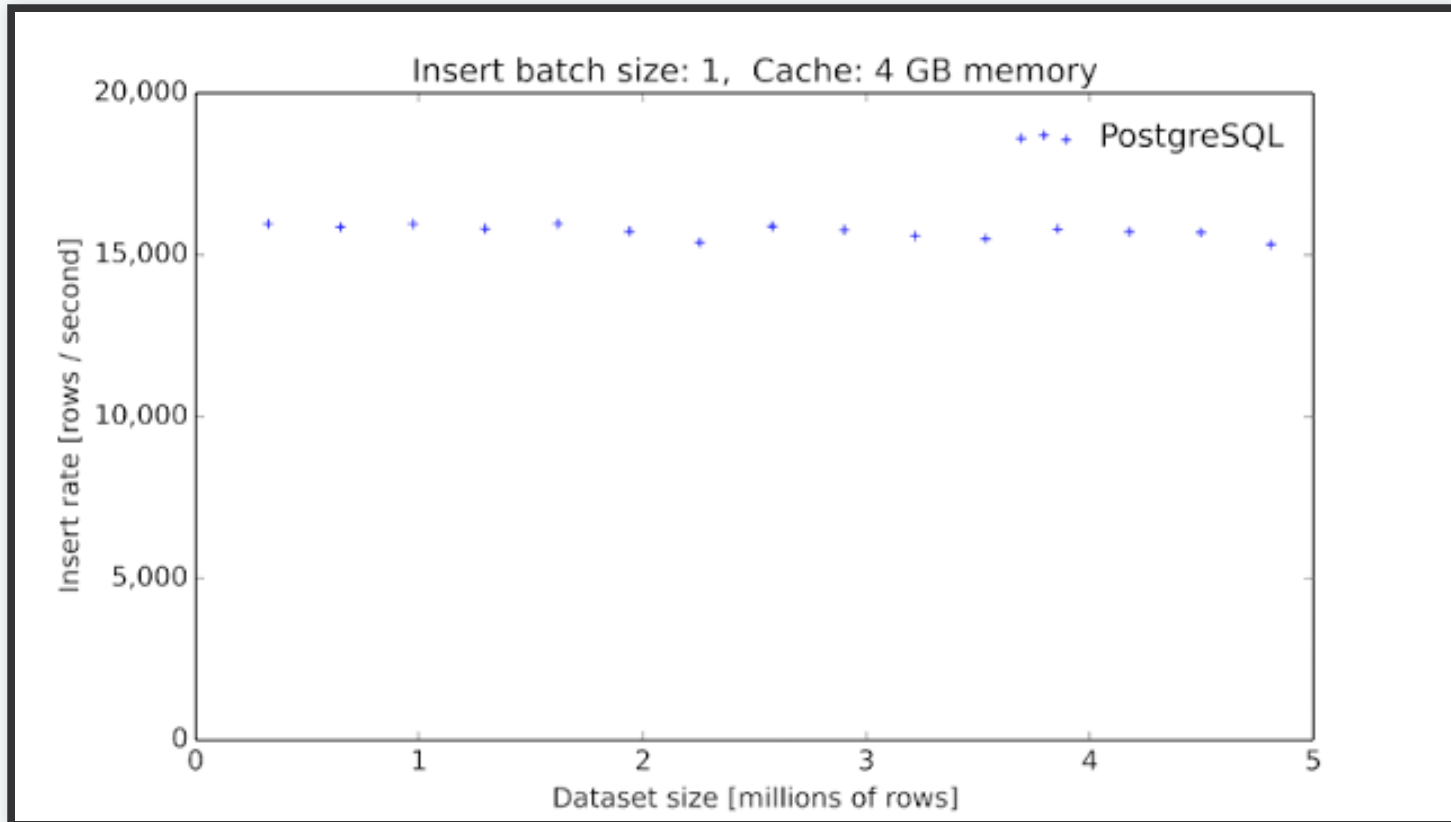
SQL vs. NoSQL

Хранение исторических данных

SQL базы данных:

- MySQL
- Postgresql

Хранение исторических данных: SQL



Хранение исторических данных: SQL

Проблемы:

- Вставка (конкурентная вставка)
- Партиционирование данных
- Выборка данных за длительный период

Хранение исторических данных: NoSQL

Обычные:

- Elasticsearch

NotNoSQL:

- Clickhouse

Специализированные:

- Victoria Metrics
- Thanos
- Cortex

Инструменты мониторинга

- Zabbix - свой агент, свой веб, SQL database
- nagios/icinga - свой агент, свой веб, SQL database
- Graphite - carbon (агент), whisper (tsdb), graphite-web
- cacti, ganglia, collectd, mrtg, munin - база rrdtool
- Netdata - агент и веб
- Prometheus - node_exporter (агент), prometheus (база, веб), alertmanager
- InfluxData - telegraph (агент), influxdb, chronograph (web), kapacitor (alerts)
- StatsD - набор инструментов для сбора метрик
- Grafana - интерфейс для дашбордов и алертинга

Алертинг

- Mail
- SMS
- SIP
- Jabber
- Slack
- Telegram
- Discord

