

Построение моделей временного ряда для описания нагрузочных характеристик в кластере

Студент:

Накорнеева Юлия, МФТИ ФПМИ Б05-253

Научный руководитель:

Саенко Владимир Иванович, доц. каф. ТиПИ



Март, 2025

План доклада

1. Постановка задачи
2. Мотивация и актуальность
3. Обзор существующих подходов
4. Выбор среды разработки
5. Пример сгенерированных данных
6. Текущие результаты

Постановка задачи

- **Контекст**

Имеется система серверов, объединенных в кластер. Для кластера существенно, чтобы его мощность была достаточной для обработки запросов, но при этом не слишком превосходила эту нагрузку для экономии ресурсов. Это создает необходимость в прогнозировании нагрузки для эффективного управления ресурсами и предотвращения перегрузок.

- **Цель**

Построить модель временного ряда для предсказания изменения нагрузки на сервер. Модель необходима для принятия решения об изменении конфигурации сервера.

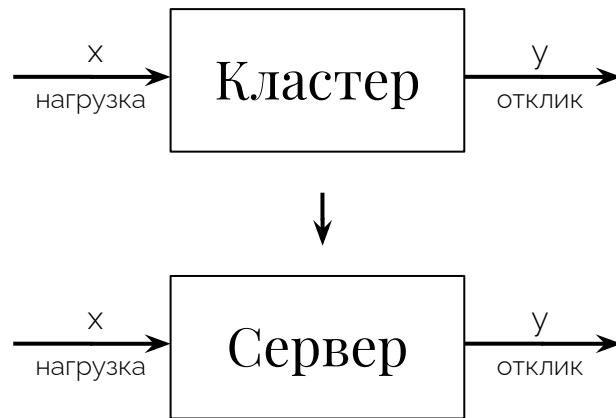


Рисунок 1 – Обобщенная схема приема данных для кластера

Постановка задачи

- Этапы работы:

1. Прием/генерация пакетов
2. Формирование нагрузочной характеристики
3. Предварительная обработка входных данных
4. Анализ данных
5. Формирование рекомендаций и построение моделей.
6. Создание программной системы

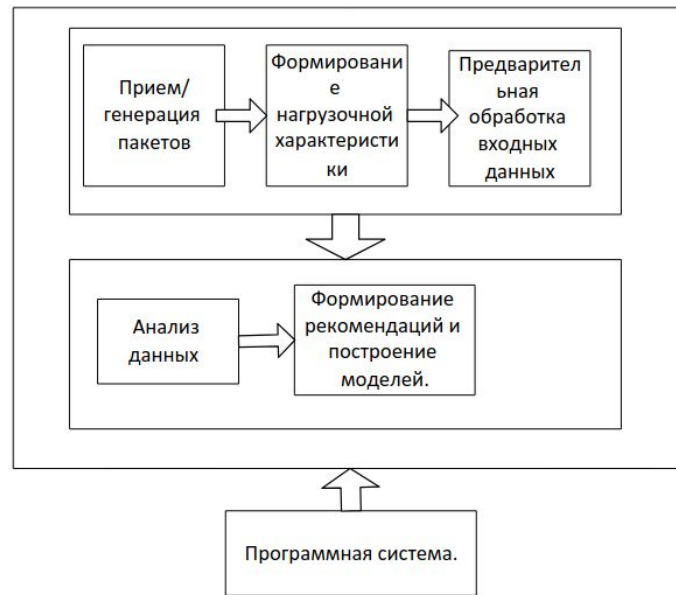


Рисунок 2 – Схема полного цикла подготовки данных

Мотивация и актуальность

На данный момент исследования в области применения временных рядов имеют хорошо разработанную научную базу, однако в контексте нашей задачи нет общего решения. Разработанная модель может быть интегрирована в системы мониторинга и управления конфигурацией серверов. Это необходимо для адаптивного изменения конфигурации в зависимости от имеющейся нагрузки. Ожидается, что такое решение повысит эффективность и отказоустойчивость системы в целом.

Современные компании активно арендуют серверные мощности, и для них крайне важно находить компромисс между затратами на аренду и скоростью ответа на запросы пользователей. В условиях высокой конкуренции клиента легко потерять из-за медленной работы системы, это делает задачу прогнозирования нагрузки особенно важной.

Модели временных рядов:

Модель авторегрессии

Модель скользящего среднего

Экспоненциальное сглаживание

Модели на нейронных сетях

Обзор моделей

Модель авторегрессии

Авторегрессионные модели прогнозирования основываются на модели временных рядов, в которой значения временного ряда в данный момент линейно зависят от предыдущих значений этого же ряда.

Наиболее популярными являются модель авторегрессии:

$$X_t = c + a_1X(t - 1) + a_2X(t - 2) + \dots + a_pX(t - p) + \varepsilon_t$$

где c – вещественная константа, a_1, \dots, a_p – коэффициенты, p – порядок авторегрессии, ε_t – ошибка модели

Обзор моделей

Модель скользящего среднего

Скользящее среднее - это расчет для анализа точек данных путем создания ряда средних значений различных подмножеств полного набора данных:

$$X_t = \frac{1}{q}(X(t-1) + X(t-2) + \dots + X(t-q)) + \varepsilon_t$$

где q - порядок скользящего среднего, ε_t - ошибка прогнозирования.

Обзор моделей

Экспоненциальное сглаживание

Экспоненциальное сглаживание использует взвешенные значения прошлых наблюдений, причем более свежие данные имеют больший вес.

$$S(t + 1) = \alpha X(t) + (1 - \alpha)S(t)$$

где α – коэффициент сглаживания, изменяющейся от нуля до единицы, а $S(1) = Z(0)$ – начальные условия

Обзор моделей


Модели на нейронных сетях

При помощи модели на нейронных сетях возможно моделирование нелинейной зависимости будущего значения временного ряда от его фактических значений и от значений внешних факторов.

Выбор среды разработки

Отдельные инструменты

- MatLab
- Statistica


- 
- Коммерческий
 - Есть поддержка нейронных сетей
 - Возможность интеграции с python
 - Есть визуализация данных

Языки программирования

- C++
- python

Выбор среды разработки

Отдельные инструменты

- MatLab
 - Statistica
- 
- Коммерческий
 - Есть поддержка нейронных сетей
 - Есть визуализация данных
 - Есть макросы для автоматизации

Языки программирования

- C++
- python

Выбор среды разработки

Отдельные инструменты

- MatLab
- Statistica

Языки программирования

- C++
- python

Armadillo и Eigen - библиотеки линейной алгебры

- Высокая производительность.
- Гибкость и более сложная настройка
- Нет визуализации

ROOT Data Analysis Framework - программный инструментарий

- Обработка данных
- Анализ данных
- Визуализация данных
- Хранение данных


Выбор среды разработки

Отдельные инструменты

- MatLab
- Statistica

Языки программирования

- C++
- python

- 
- Простота и удобство использования.
 - Большое количество библиотек (pandas, numpy, scipy, statsmodels, tensorflow, pytorch).
 - Отличная поддержка визуализации (matplotlib, seaborn, plotly).
 - Бесплатное и открытое ПО.

Пример сгенерированных данных (неделя)



Рисунок 3 – график нагрузки данных на сервер в течении недели

Пример сгенерированных данных (сутки)



Рисунок 4 – график нагрузки данных на сервер в течении дня

Пример кода генерации диаграммы процесса для моделирования временного ряда

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Данные
hours_in_week = 24
time = np.arange(0, hours_in_week, 0.2)

# Функция отражающая нагрузку
evening_peak = 72.5 * np.abs(np.sin(np.pi * (time - 6) / 24)) + 5
lunch_peak = 22.5 * np.abs(np.sin(np.pi * (time - 6) / 6))

# Шум
noise = 10 * np.random.normal(size=len(time))

# Итоговые данные
requests = lunch_peak + evening_peak + noise

# Создание DataFrame
data = pd.DataFrame({'Time': time, 'Requests': requests})
```

Рисунок 4 – код генерации процесса

Текущие результаты

- Выполнено изучение моделей временных рядов
- Сделан сравнительный анализ инструментов разработки
- Подготовлен набор сгенерированных данных

Источники

1. Обзор современных моделей и анализа временных рядов
<https://doi.org/10.32362/2500-316X-2020-8-4-7-45>
2. Документация MatLab
<https://docs.exponenta.ru/matlab/index.html>
3. Документация Armadillo
<https://arma.sourceforge.net/docs.html>
4. Документация Eigen
https://eigen.tuxfamily.org/index.php?title=Main_Page

Спасибо за внимание!

Накорнеева Юлия

nakorneeva.ya@phystech.edu

Tg: @yulianakorneeva