

ПРОГНОЗИРОВАНИЕ КОМПЬЮТЕРНЫХ МОЩНОСТЕЙ ДЛЯ СЕРВИСНОЙ ШИНЫ ПРЕДПРИЯТИЯ

Аннотация. В статье описан способ сбора показателей потребления ресурсов Сервисной Шиной Предприятия для решения задачи планирования компьютерных мощностей.

Ключевые слова: Многослойный перцептрон, Прогнозирование, Сервисная Шина Предприятия, ESB, REST, Apache Camel, Spring Boot, Docker,

Планирование мощностей

Одним из ключевых этапов в жизненном цикле программного обеспечения является «Этап развертывания компонентов системы». Существует несколько процессов выполняемых в рамках данного этапа [1]:

- оптимизация плана компонентной конфигурации
- развертывание отдельных программных компонентов
- создание целевой компонентной конфигурации

Проблемным является этап «Оптимизации плана компонентной конфигурации» включающий в себя задачу по определению компьютерных мощностей для программного обеспечения [1].

Задача планирования производственных мощностей имеет несколько возможных методов решения [2]:

- Метод черного планирования мощностей
- Модуль планирования требуемых мощностей
- Кадровое обеспечение производственной линии
- Потребности в кадровом обеспечении гибкой рабочей ячейки

В данной работе мы рассмотрим планирования компьютерных мощностей, для решения, разработанного на базе Сервисной Шиной

Предприятия с использованием метода «Метод черного планирования мощностей».

Метод черного планирования мощностей обычно применяется вручную или с помощью простой программы табличных вычислений. Он может использоваться для определения требуемых мощностей в долгосрочном и среднесрочном периодах. План-график мощностей составляется, включая текущие потребности в мощностях и прогнозируемые, на месяц вперед, чтобы получить необходимые ресурсы.

В данной статье мы рассмотрим способ получения показателей потребления ресурсов Сервисной Шиной Данных, а также время отклика на атомарный запрос, что позволит нам в дальнейшем использовать эти данные для решения задачи «Планирования производственных мощностей» методом «Метод черного планирования мощностей».

- **Сбор показателей потребления мощностей**

В данной статье рассматривается одна из задач интеграции систем, решаемых с помощью Сервисной Шины Предприятия: Uniform Data Access (Унифицированный доступ к данным) [3]

Для получения показателей потребления мощностей Сервисной Шиной Предприятия был выбран один из базовых сценариев ее использования, при задаче предоставления унифицированного доступа к данным: создание Restful сервиса, где источником данных является база данных.

Была построена виртуальная лаборатория, включающая в себя следующие элементы (*Рис. 2*):

- База Данных (PostgreSQL)
- Интеграционное приложение (На базе Сервисной Шины Предприятия Apache Camel)
- Приложение для управление ходом эксперимента [4]
- Приложение для отправки запросов в интеграционное приложение [4]



Рис. 2 Архитектура виртуальной лаборатории

Основными шагами эксперимента являются:

- Установка и запуск клиентских приложений, управляющего, интеграционного и базы данных
- Подготовка данных для эксперимента. Для эксперимента были использованы данные агрегируемые из нескольких источников [5]
- Настройка Управляющего приложения
- Запуск эксперимента
- обработка и сохранения результатов в виде таблицы

Интеграционное приложение принимает на вход сообщения от клиента в виде JSON объекта [4]:

```

{
  "columnsCount": 5,
  "rowsCount": 10,
  "serializeType": "JSON"
}
  
```

Листинг 1 Пример запроса от клиентского приложения

На каждый такой запрос интеграционное приложение запускает фоновый процесс замера показателей ресурсов, создает динамический SQL запрос и исполняет его. В зависимости от параметра «serializeType» интеграционное приложение сериализует данные одним из трех способов:

- JSON
- XML
- CSV

Далее к полученному HTTP телу ответа, добавляются HTTP заголовки с результатами замеров показателей потребления ресурсов.

Клиентское приложение, при получении ответа от интеграционного приложения, отправляет результаты замеров в управляющее приложение. Таким образом все результаты замеров потребления ресурсов по каждому запросу консолидируются в управляющем приложении [4].

При планировании эксперимента были выделены следующие показатели потребления мощностей интеграционным приложением:

- Время отклика на каждый запрос
- Загрузка центрального процессора
- Использование оперативной памяти приложением

Также были выделены следующие факторы влияющие на показатели потребления мощностей:

- `serializeType` - типа сериализации сообщения. Возможные значения: JSON, XML, CSV
- `concurrentlyMonitoringCount` - количество параллельных запросов, отправляемых клиентским приложением в интеграционное
- `rowCount` - количество строк, выбираемых из таблицы базы данных
- `columnsCount` - количество столбцов, выбираемых из таблицы базы данных

По полученным факторам и откликам был составлен следующий план эксперимента:

Таблица 1 План эксперимента

Фактор	Основной уровень	Область определения	Интервал	Примеры значений через запятую
<code>serializeType</code>	-	JSON, XML, CSV	-	JSON, XML, CSV
<code>concurrentlyMonitoringCount</code>	6	[1, 10]	2	2,4,6,8,10
<code>rowCount</code>	6000	[1, 11000]	1000	1000,2000,3000
<code>columnsCount</code>	30	[1,50]	10	10,20,30,40,50

Результаты эксперимента дополнительно были «очищены» от шумов.

Для выявления зависимости откликов от факторов, была составлена таблица корреляции откликов от факторов (Таблица 2).

Таблица 2 Значение коэффициента корреляции отклика от фактора

	Время отклика	Загрузка ЦП	Использования Оперативной памяти
serializeType	-0,32481	-0,15023	0,08665
concurrentlyMonitoringCount	0,61501	0,44862	0,70660
rowCount	0,21164	0,07476	0,01089
columnsCount	0,48537	0,12886	0,00727

- **Прогнозирование мощностей**

Для примера математической модели способной прогнозировать компьютерные мощности была выбрана регрессионная модель: многослойный перцептрон.

Данная модель позволяет построить функциональную зависимость между факторами и откликами.

Для обучения регрессионной модели прогнозирования значения отклика от значений факторов были выбраны следующие параметры [6]:

- Функция активации линейная $f(x) = x$
- Квазиньютоновские методы для обучения многослойного перцептрона

По данным полученным из эксперимента была обучена модель для получения значений:

- Время отклика
- Используемая память
- Загрузка ЦП
- На входе были переданы следующие параметры:
- serializeType - типа сериализации сообщения. Возможные значения: JSON, XML, CSV

- `concurrentlyMonitoringCount` - количество параллельных запросов, отправляемых клиентским приложением в интеграционное
- `rowCount` - количество строк, выбираемых из таблицы базы данных
- `columnsCount` - количество столбцов, выбираемых из таблицы базы данных

Результаты обучения представлены в виде коэффициента детерминации для откликов, полученных с помощью обученной модели и откликов, полученных в ходе эксперимента:

Таблица 3 Результаты обучения многослойного персептрона

Наименование выходного параметра	Значение коэффициента детерминации
Время отклика	0,815
Используемая память	0,374
Загрузка ЦП	0,21

Ниже представлен график «Время откликов на атомарный запрос» полученных при построении регрессионной модели (красный пунктир) и при эксперименте (зеленый пунктир):

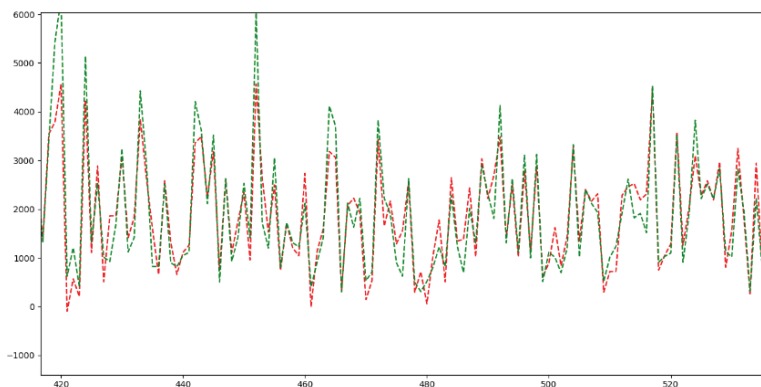


Рис. 3 Сравнение откликов, полученных в ходе эксперимента (зеленый пунктир) и с помощью математической модели (красный пунктир)

По результатам обучения можно сделать следующие выводы:

- Регрессионная модель способна достаточно хорошо прогнозировать время отклика на атомарный запрос.
- Регрессионная модель способна прогнозировать количество используемой памяти приложением. Но имеет достаточно большое отклонение: 51МВ. Требуется результаты эксперимента для большей загрузки, чтобы оценить, как поведет себя отклонение при больших нагрузках на интеграционное приложение.
- Регрессионная модель для загрузки ЦП не может быть смоделирована, так как не учитывает следующие особенности работы приложения: работу «сборщика мусора», оптимизацию приложений «на лету» и другие оптимизационные процессы, происходящие на уровне как Сервисной Шины Предприятия, так и на уровне виртуальной машины JAVA [7].
- Данные, полученные в виртуальной лаборатории, пригодны для построения модели для прогнозирования, что позволяет решить задачу «Задача планирования производственных мощностей имеет» в этапе «Оптимизации плана компонентной конфигурации» [1].

СПИСОК ЛИТЕРАТУРЫ

1. Лаврищева Е.М. Методы и средства инженерии программного обеспечения / Лаврищева Е.М. , Петрухин В.А. // Московский физико-технический институт (государственный университет), 2006. - С. 296
2. Луис Р.. Система канбан. Практические советы по разработке в условиях вашей компании / Пер. с англ. Е.В. Журиной; Под науч. ред. Э.А. Башкардина. — М.: РИА «Стандарты и качество» — 216 с., ил. — (Серия «Бережливое управление»),. 2008
3. Dittrich, K.R. Data Integration - Problems, Approaches, and Perspectives/Patrick Ziegler and Klaus R. Dittrich // Database Technology Research Group. Department of Informatics, University of Zurich. С. 4-5

4. Кабардинский Е.О. Выбор сервисной шины предприятия для разработки интеграционного решения / Кабардинский Е.О., Ивашко А.Г.//Материалы Всероссийской научно-практической конференции молодых ученых с международным участием. В 2-х томах. Главный редактор А.П. Шкарапута. 2017. - С. 95 - 99.
5. Realistic data generator. [Электронный ресурс]. - <https://mockaroo.com/>
(Дата обращения: 03.05.2018)
6. SciKit learn. Multi-layer Perceptron regressor. [Электронный ресурс]. - http://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html
(Дата обращения: 03.05.2018)
7. Memory Management in the Java HotSpot™ Virtual Machine / Sun Microsystems//HotSpot™ Virtual Machine specification. 2006. – С. 4