

При запуске программы появляется окно. Пользователь должен вставить USB-устройство в порт. Когда система обнаруживает факт подключения устройства, программа проверяет, входит ли серийный номер подключенного устройства в список разрешенных.



Рисунок 1 – Сообщения о подключении и отключении портов

Если доступ разрешен, в окне программы появится сообщение «Порты включены!» и будет показан серийный номер устройства (рис. 1). Если доступ запрещен, в окне программы появится сообщение «Порты выключены!» и снова будет показан серийный номер устройства (см. рис. 1). После отключения и извлечения USB-устройства порты будут включены.

**Заключение.** В результате выполнения работы были:

- изучены принципы построения системы безопасности при работе в компьютерном классе;
- изучены принципы функционирования шины USB;
- определены наборы групп пользователей и системных разрешений для работы с реестром ОС в контексте подключения USB-устройств;
- разработано программное обеспечение для предотвращения несанкционированного подключения USB-устройств к персональному компьютеру.

#### Литература

1. Щесняк, Е.Л. Комплексная безопасность высшего учебного заведения / Е.Л. Щесняк, В. Г. Плющиков, В.С. Побыванец.– Москва: Российский университет дружбы народов, 2011. — 768 с.
2. Интегрированная среда разработки Visual Studio [Электронный ресурс]. - 2017. - Режим доступа: <https://msdn.microsoft.com/ru-ru/library/dn762121.aspx>. -Дата доступа: 10.02.2017.

## РЕАЛИЗАЦИЯ МЕТОДА ПЕРЕМЕННЫХ НАПРАВЛЕНИЙ В КОНТЕКСТЕ АЛГОРИТМОВ РАСПАРАЛЛЕЛИВАНИЯ

*Савостеенко Е.В.,*

*магистрант 6 курса ВГУ имени П.М. Машерова, г. Витебск, Республика Беларусь*

*Научный руководитель – Маркова Л.В., канд. физ.-мат. наук, доцент*

Инновационное развитие общества сегодня напрямую связано с решением задач, требующих большого объема весьма сложных вычислений. Нет ни одной области науки и техники, в которой бы не ставились задачи математического моделирования. Это задачи ядерного синтеза, геномной инженерии, тепло и массопереноса, экологических и метеопрогнозов, изучения человека, создания роботов и т.д. В основе вычислений каждой задачи лежат алгоритмы, важнейшим критерием разработки которых является их эффективность [1].

Цель исследования – показать возможность повышения эффективности решения задачи в зависимости от программной реализации вычислительного алгоритма.

**Материал и методы.** Материалом исследования была выбрана двумерная квазилинейная задача теплопроводности. Основные методы исследования – системный подход, анализ литературы, методы ООП и параллельных вычислений, вычислительный эксперимент.

**Результаты и их обсуждение.** Большое количество программного обеспечения, написанного ранее для последовательной (однопроцессорной) вычислительной техники, не обладает необходимым запасом параллелизма и попытки его распараллеливания на уровне распараллеливания отдельных циклов, как правило, не приводят к хорошим результатам. Производительность кода остается низкой.

Другим подходом к разработке параллельных программ является использование модели программирования с распараллеливанием данных, когда в последовательный код вставляются директивы компилятору и распараллеливание происходит на автоматическом уровне. Если программа логически простая и обладает ресурсом параллелизма, этот подход может дать хорошие результаты. Однако рекордных результатов удается получить только при использовании знаний программиста о структуре алгоритма и управлении вручную потоком данных. Это достигается при использовании стиля программирования с передачей сообщений [3].

Одной из основных характеристик параллельного алгоритма является ускорение  $S$ , которое определяется как отношение общего времени прохождения программы для последовательного алгоритма ко времени работы параллельного алгоритма с использованием  $P$  процессоров.

Другой важной характеристикой алгоритма является параллельная эффективность  $E$ , которая определяется как отношение ускорения к числу процессоров, то есть  $E = S / P$ .

Получение близкой к пиковой производительности и высокой параллельной эффективности программ представляет собой сложную задачу.

Возможные причины потери параллельной эффективности следующие [3]:

- 1) стартовое время инициализации параллельной программы,
- 2) дисбаланс загрузки процессоров,
- 3) коммуникационные затраты,
- 4) наличие последовательных частей кода.

Для вычислительного эксперимента был реализован метод переменных направлений для двумерного уравнения теплопроводности. Для получения средних значений расчет выполнялся 5 раз, после чего сетка по обоим направлениям была увеличена на 1000. Всего было произведено 35 итераций. Общее время выполнения эксперимента без распараллеливания: 48 минут, в то время как с распараллеливанием: 30 минут.

Ниже представлена зависимость времени исполнения в миллисекундах от количества узлов сетки.

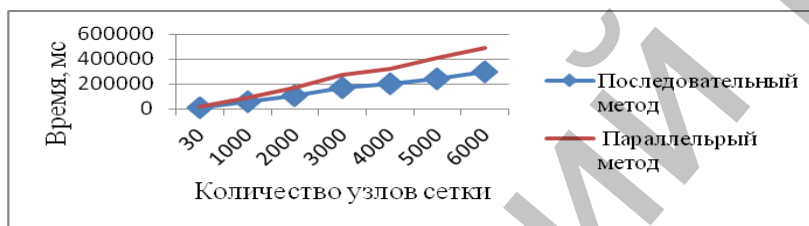


Рис. 1

Эффективность, представляющая отношение времени выполнения задачи с использованием распараллеливания к решению задачи последовательным методом, отображена на графике ниже.

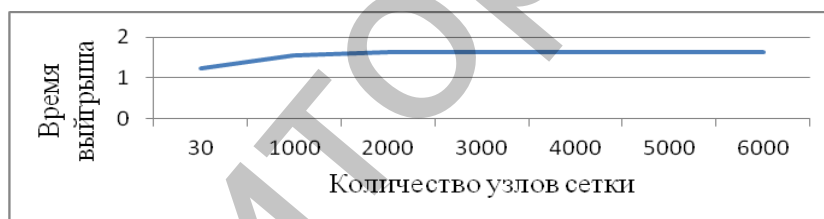


Рис. 2

На рис. 2 видно, что решение задачи с распараллеливанием проходит в 1,63 раз быстрее после 1250 узла сетке.

Все вышеуказанные расчеты были выполнены при помощи класса `TThread`, реализующего работу алгоритма в отдельном потоке. Количество одновременно работающих потоков равнялось 2. Для получения дополнительных результатов и закрепления предыдущих, используем тот же алгоритм, используя 2, 3, 4 и 8 потоков запущенных одновременно, другими словами будет создано 2, 3, 4 и 8 экземпляров класса `TThread` для каждого эксперимента.

Был проведен эксперимент зависимости времени исполнения от количества узлов при использовании различного количества потоков. Так как процессор, на котором выполнялись вычисления, двухъядерный, то никаких существенных изменений во времени работы не произошло. Однако, если создать пул с чрезмерно большим количеством потоков, это может привести к существенным затратам машинных ресурсов на создание объектов в памяти и простоям в процессе очереди выполнения, что в целом приведет к ухудшению эффективности распараллеливания. На графике ниже представлен результат.

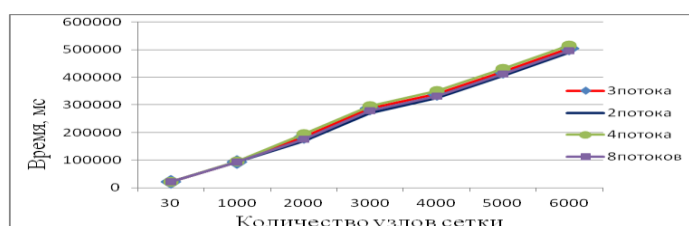


Рис. 3

**Заключение.** При помощи реализованной программы на языке программирования C++ была исследована эффективность использования распараллеливания двумерной квазилинейной задачи теплопроводности. Исследования показали, что использование распараллеливания для решения задачи теплопроводности методом переменных направлений является эффективным и может быть использовано для сокращения времени вычислений. Практическая значимость исследований состоит в том, что результаты могут быть использованы в учебном процессе для студентов ИТ-специальностей.

Литература:

1. Кормен Т. Алгоритмы: построение и анализ. / Пер. с англ. / Т.Кормен и др. – М. : ООО «И.Д.Ви льямс», 2013. – 1328 с.
2. Макконнелл, С. Совершенный код. Мастер-класс / Пер. с англ. / С. Макконнелл. – СПб.: Издательско-торговый дом «Русская редакция», 2005. – 896 с.
3. Анотонов, А.С. Параллельное программирование с использованием технологии MPI: Пособие / Анотонов А.С. – М.: Изд-во МГУ, 2004. – 71 с.
4. Воеводин, В.В. Параллельные вычисления / В.В. Воеводин, Вл.В. Воеводин – СПб.: БХВ-Петербург, 2002. – 608 с.

## АРХИТЕКТУРА СИСТЕМЫ АНАЛИЗА И КОНСОЛИДАЦИИ ДАННЫХ ИЕРАРХИЧЕСКИХ ОБЪЕКТОВ

*Соколовский В.В., Шестаков И.В.,*

*студенты 5 курса ВГУ имени П.М. Машерова, г. Витебск, Республика Беларусь*

*Научный руководитель – Ермоченко С.А., канд. физ.-мат. наук*

На сегодняшний день, ценность и достоверность знаний, полученных в результате интеллектуального анализа бизнес-данных, зависит не только от эффективности используемых аналитических методов и алгоритмов, но и от того, насколько правильно подобраны и подготовлены исходные данные для анализа.

Обычно руководителям проектов по бизнес-аналитике приходится сталкиваться со следующими проблемами. Во-первых, данные на предприятии расположены в различных источниках самых разнообразных форматов и типов – в отдельных файлах офисных документов (Excel, Word, обычных текстовых файлах), в учетных системах («1С: Предприятие», «Парус» и др.), в базах данных (Oracle, Access, dBase и др.). Во-вторых, данные могут быть избыточными или, наоборот, недостаточными. А в-третьих, данные являются «грязными», то есть содержат факторы, мешающие их правильной обработке и анализу (пропуски, аномальные значения, дубликаты и противоречия).

Поэтому, прежде чем приступать к анализу данных, необходимо выполнить ряд процедур, цель которых – доведение данных до приемлемого уровня качества и информативности, а также организовать их интегрированное хранение в структурах, обеспечивающих их целостность, непротиворечивость, высокую скорость и гибкость выполнения аналитических запросов.

Зачастую такая задача решается с применением технологии OLAP и соответствующих коммерческих систем [1]. Однако, помимо высокой стоимости таких решений, они часто оказываются довольно громоздкими и ориентированы на обработку больших объемов данных, что не всегда оправдано в небольших системах.

Целью работы является проектирование архитектуры для системы анализа и консолидации данных иерархических объектов с небольшим объемом информации.

**Материал и методы.** Материалом исследования является набор иерархических объектов и их числовых параметров.

Для создания web-приложения использовались следующие технологии:

- Серверная часть:

- Язык программирования Java, JavaScript.
- Контейнер сервлетов Tomcat, сервер NodeJS.
- Фреймворк Spring, библиотека Hibernate.
- СУБД MySQL.
- БД MariaDB.

- Клиентская часть:

- Языки программирования JavaScript, TypeScript.
- Язык разметки HTML.
- Язык написания каскадных таблиц стилей CSS3.
- Фреймворк AngularJS, библиотека D3.js, библиотека Bootstrap.

Методы исследования – моделирование, анализ, методы объектно-ориентированного и прототипно-ориентированного программирования, ранжирование, тестирование.

**Результаты и их обсуждение.** В основу архитектуры приложения была положена иерархия объектов. Каждый объект в иерархии имеет некий набор параметров. Так как целью работы является разработка системы анализа и консолидации данных без привязки к конкретной предметной области, то и