

Петров А.С., Сидоренко В.Г., РУТ (МИИТ), г. Москва

## Применение параллельных вычислений для решения задач планирования движения поездов метрополитена

Конечной целью автоматизированного построения планового графика движения (ПГД) поездов является получение траекторий движения всех маршрутов при соблюдении:

- заданных параметров и ограничений;
- заданной парности движения, которая может изменяться во времени;
- правил ночной расстановки составов;
- графика оборота электроподвижного состава;
- устойчивости к воздействию возмущающих факторов;
- рациональности с точки зрения выбранных критериев.

Для достижения поставленной цели применительно к процедуре построения ПГД пассажирских поездов на Московском Метрополитене осуществлено ее разбиение на множество процессов, каждый из которых ассоциируется с происходящими в течении рабочего времени на линии метрополитена процессами [1].

С целью реализации процедуры построения ПГД используется детализация операций, происходящих в процессе жизнедеятельности метрополитена. Прямое использование атомарных (узкоспециализированных, простых) операций приводит к излишней сложности алгоритмов построения ПГД.

Существует два способа укрупнения операций: групповые и комплексные операции. Основным отличием групповых операций от комплексных заключается в количестве используемых типов атомарных операций во время обработки данных. Групповые операции используют один тип атомарной операции, многократно повторяя ее для множества подобных объектов. Комплексные операции используют более одного типа атомарных операций. В дополнение к атомарным операциям, комплексные могут использовать в обработке данных групповые [2].

Такой подход к распределению сложности построения ПГД позволяет гибко менять алгоритмическую составляющую любой операции, не вмешиваясь в логику обработки каждого процесса в отдельности. Любые изменения алгоритма конкретной операции скрытым образом вносятся во все операции, которые на ней базируются. Это позволяет при работе над отдельной операцией абстрагироваться от особенностей обработки данных, в которых она задействуется или которые она использует.

Создана одна выделенная операция, называемая базовой. Главное ее назначение заключается в абстра-

гировании методов взаимодействия с любой другой операцией. Базовая операция является шаблоном для организации всех остальных операций.

Обобщение имеющихся знаний об устройстве операций работы над нитками и алгоритмах их автоматизации, многоуровневое исследование процесса автоматизированного построения ПГД позволяет провести анализ возможности ввода параллельных вычислений [3].

Для реализации преимуществ от использования параллельных вычислений необходимо решить проблемы, связанные с синхронизацией работы потоков. Эта задача подразумевает защиту потоков от взаимного влияния на совместно используемые ресурсы. Для решения данной задачи требуется установить атомарные операции, которые влияют на ресурсы в ходе выполнения программного обеспечения. С другой стороны, знание атомарных операций позволяет строить приложения с большей производительностью, так как при написании программного кода появляется возможность избегать обращения к разделяемым ресурсам в одно и то же время из разных потоков. Отличительной чертой задачи автоматизации планирования движения поездов является то, что в ней существует широкое множество атомарных операций, обладающих указанным свойством, что делает актуальным реализацию программного обеспечения средств автоматизации с использованием параллельных вычислений.

В Таблице 1 представлена гипотеза о границах отсутствия влияния параллельно выполняемых однотипных атомарных операций на совместно используемые ресурсы. В настоящее время авторы работают над проверкой выдвинутой гипотезы. Параллельное выполнение однотипных атомарных операций позволит изменить подход к реализации групповых операций. Для реализации комплексных операций важно определить границы отсутствия влияния параллельно выполняемых разнотипных атомарных операций на совместно используемые ресурсы.

Реализация параллельных вычислений требует изучения имеющихся разделяемых данных ввиду наличия опасности использования таких данных разными потоками в одно и то же время. Опасность заключается в их повреждении или неправильной интерпретации одним из потоков, что может привести к недопустимым ошибкам при вычислениях.

Таблица 1. Атомарные действия, выполняемые над нитками

№ п/п	Атомарная операция	Условия отсутствия влияния на совместно используемые ресурсы
1.	Создание	Отсутствие действий, выполняемых над соседними нитками
2.	Удаление	Отсутствие действий, выполняемых над соседними нитками.
3.	Изменение станции начала или конца нитки	Отсутствие действий, выполняемых над нитками, связанными с изменяемой.
4.	Изменение станции начала или конца нитки с удалением предыдущих или последующих ниток	Отсутствие действий, выполняемых над соседними нитками и нитками, время отправления которых с первой станции пути не различается больше, чем на время полного оборота состава.
5.	Перемещение нитки	Отсутствие изменения последовательности прибытия и отправления ниток на все станции пути.
6.	Ввод сверхрежимной стоянки	Отсутствие изменения последовательности прибытия и отправления ниток на все станции пути.
7.	Создание связей между нитками	Отсутствие действий, приводящих к пересечению множеств ниток, задействованных в каждой из выполняемых операций.
8.	Удаление связей между нитками	Нет ограничений
9.	Назначение маршрута на нитку графика	Отсутствие действий, приводящих к назначению одного и того же маршрута на нитки, двигающиеся одновременно.
10.	Определение маневровых перемещений в начале и конце нитки	Отсутствие действий, приводящих к невыполнению ограничений на емкость указателей ночной расстановки.
11.	Удаление маневровых перемещений в начале и конце нитки	Нет ограничений
12.	Удлинение нитки до конечных станций линии	Отсутствие изменения последовательности прибытия и отправления ниток на все станции пути.

С точки зрения программной реализации это означает использование стандартных механизмов многопоточности языка C#. К таким механизмам можно отнести использование заданий, создание выделенных потоков. При этом предполагается отдать предпочтение именно заданиям. Это связано с тем, что они имеют включенные в себя алгоритмы порождения дочерних потоков и их освобождения по мере надобности дополнительных вычислительных ресурсов. В то время как создание выделенного потока вручную может приводить к коллизиям разной степени тяжести, созданию избыточного количества простаивающих потоков, что может приводить к существенному падению производительности приложения и всей Операционной Системы в целом.

Для реализации механизма распараллеливания вычислений с помощью заданий необходимо выполнить ряд синтаксических и интерфейсных требований к реализации алгоритма в виде программного кода. К таким требованиям относятся:

- обособление алгоритма в рамках отдельного метода;
- использование для метода модификатора `static` [4], что подразумевает разделение объявляемой конструкции между всеми создаваемыми экземплярами класса без ссылки на конкретный экземпляр. Другими словами, конструкция должна вызываться на уровне класса, а не на уровне экземпляра [5]; все данные, с которыми оперирует метод, должны быть агрегированы в одном параметре.

При необходимости выполнения оформленного таким образом алгоритма в параллельном режиме производится создание объекта задания, относящегося к классу `Task`. Данный объект имеет возможность:

- инициировать выполнение алгоритма;
- выполнить последствие по завершению метода;
- перевести владельца экземпляра в режим ожидания завершения задания;
- выдать флаг об окончании вычислительной работы;
- отменить задание.

Несмотря на высокую степень автоматизации описанного алгоритма необходимо следить за выдачей на обработку разделяемых данных только одному заданию одновременно. При этом выдаваемые объекты могут иметь ссылки на динамически выделяемые участки памяти, поэтому требуется проверять, что выданный задаче экземпляр не испортит вычисления в смежном потоке над другим экземпляром с ссылкой на те же данные [4].

После определения полного перечня разделяемых данных необходимо произвести их защиту от негативного влияния потоков. Для этого потребуются осуществить сравнение различных способов их хранения с целью минимизации возможных пересечений вычислительных процессов во время исполнения алгоритмов или сравнить различные способы реализации потоковой системы проекта с целью минимизации возможных пересечений. Оставшиеся пересечения необходимо разграничить с помощью механизма синхронизации потоков.

После разграничения данных потребуется формализовать предлагаемое хранение данных, а также организацию взаимодействия между потоками. Сделать это необходимо с целью последующего синтеза получившихся примитивов в цельный проект для получения общего представления о предлагаемой архитектуре нового программного продукта.

Люди, наблюдая за многими явлениями природы, издавна обращали внимание на их самоподобие [6]. Наиболее ярко свойство самоподобия проявляется во фракталах. Свойство самоподобия включает в себя два основных компонента:

- **сдвиг функции** (объекта, явления)  $y(t)$ :  
 $y(t) = y(t + k)$ , где  $k$  – величина сдвига;
  - **масштабирование** (сжатие/растяжение):  
 $y(t) = y(a^m t)$ , где  $a = Const, m = 0, 2, 3, \dots, M$  – коэффициенты масштабирования.
- Следующие выражение объединяет эти условия:  
 $y(t) = y(a^m (t + k))$

По мнению авторов, ПГД также обладает свойством самоподобия. Рассмотрим уровни равномерности при построении ПГД, представленные на Рисунке 1 [1].

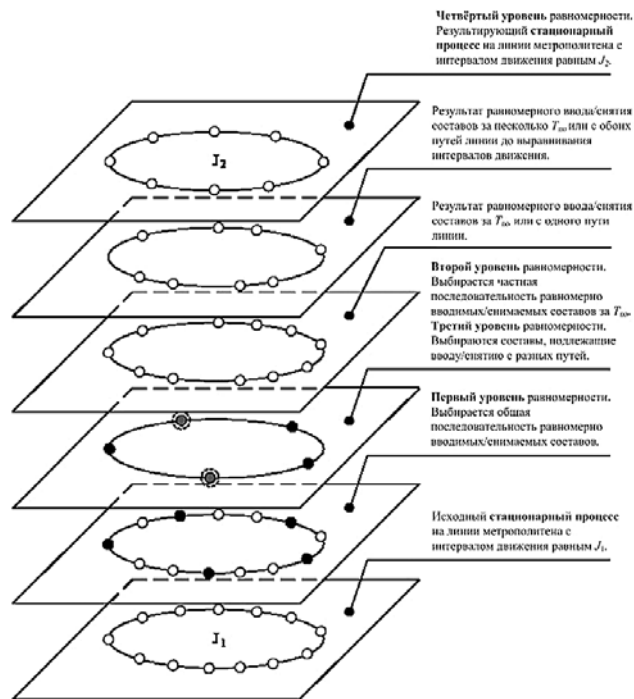


Рис.1. Уровни равномерности при построении ПГД

Переход между двумя стационарными процессами с различной парностью не обладает высокой частотой снятия, что позволит использовать параллельные вычисления при выполнении соответствующих атомарных операций, если гипотезы, выдвинутые в Таблице 1, найдут свое подтверждение.

В общем случае процесс равномерного ввода/снятия составов в течение продолжительного промежутка времени с учетом географии линии может быть описан следующим образом (Рисунок 1):

1. Дано исходное множество  $M$  составов линии (отмечены белыми элементами на окружности), перемещающихся с интервалом движения  $J_1$ .
2. Из множества  $M$  выбирается подмножество составов  $M_{\Sigma}$ , которое соответствует равномерному расположению вводимых/снимаемых составов за весь переходный процесс (отмечены черными элементами).
3. При выполнении  $i$ -го снятия из множества  $M_{\Sigma}$  выбирается множество  $M_{\text{вв}(сн)}$  равномерно расположенных составов, подлежащих вводу/снятию. Если ввод/снятие производится по двум путям или из/в нескольких депо, то из множества  $M_{\text{вв}(сн)}$  выбираются равномерно расположенные составы, подлежащие вводу/снятию по каждому из главных путей или из/в каждого депо (отмечены серыми элементами).
4. После выполнения ввода/снятия из множества  $M_{\Sigma}$  исключаются элементы множества  $M_i$ :  
 $M_{\Sigma} = M_{\Sigma} \setminus M_{\text{вв}(сн)} = M_{\Sigma} \cap M_{\text{вв}(сн)}$
5. Если выполнены не все итерации ввода/снятия, то происходит переход к пункту 3. Повторение операций ввода/снятия продолжается до тех пор, пока из множества  $M$  не будут исключены все элементы множества  $M_{\Sigma}$ .
6. Если все итерации ввода/снятия выполнены, то выполняется выравнивание интервалов движения согласно критерию интервалов движения по отправлению поездов со станций. В этом случае получаем новое множество  $M'$  составов линии, перемещающихся с интервалом движения  $J_2$ .

В нашем случае сдвиг самоподобного процесса равен времени полного оборота состава, а коэффициент сжатия определяется отношением  $J_2/J_1$ . Аналогичная идея была высказана при построении временной модели линии метрополитена [7].

Когда речь идет об исследовании самоподобных процессов, в них выделяют аппроксимирующую и детализирующие составляющие. В нашем случае аппроксимирующую составляющую определяет множество вводимых/снимаемых составов, а детализирующую – изменения, внесенные в график с целью выравнивания интервалов движения по отправлению поездов со станций.

Анализ ПГД, как самоподобного процесса, позволит исследовать вопрос использования разделяемых данных при параллельных вычислениях.

**Литература:**

1. Сафронов, А.И. Построение планового графика движения для метрополитена [Текст] / А.И. Сафронов, В.Г. Сидоренко // Мир транспорта. – 2010. – № 3. – С. 98–105.
2. Петров, А.С. Методика автоматизации построения планового графика движения пассажирских поездов метрополитена [Текст] / А.С. Петров, А.И. Сафронов, В.Г. Сидоренко, М.А. Чжо // Транспорт и образование: актуальные вопросы и тенденции – ЧУПС УрГУПС – 2015. – С. 74 – 80.
3. Сидоренко, В.Г. Обеспечение безопасности движения поездов при построении планового графика движения [Текст] / В.Г. Сидоренко, К.М. Филипченко // Безопасность Движения Поездов: труды XIII научно-практической конференции. — М.: МИИТ, 2012. — С. II–6 — II–7.
4. Рихтер, Д. CLR via C# программирование на платформе Microsoft .NET FRAMEWORK 4.5 на языке C# [Текст] / Д. Рихтер; ООО Питер – СПб. 2013. – 896 с. – ISBN 978-5-496-00433-6.
5. Троелсен, Э. Язык программирования C#5.0 и платформа .NET 4.5 [Текст] / Э. Троелсен; ООО Вильямс – М. 2013. – 1312 с. – ISBN 978-5-8459-1814-7.
6. Мандельброт, Б. Фрактальная геометрия природы [Текст] / Б. Мандельброт; Институт компьютерных исследований – Москва. 2002. – 656 с. – ISBN 5-93972-108-7.
7. Балакина, Е.П. Автоматика выполняет функции диспетчера [Текст] / Е.П. Балакина // Мир транспорта, 2008, №2. – С. 104-109.



Тимченко В.С., ООО «Транс Сити Групп Центр», г. Санкт-Петербург  
Ковалев К.Е., ПГУПС, г. Санкт-Петербург

## **Оценка длительностей занятия приемо-отправочных путей технической станции с учетом количества поездных локомотивов методом имитационного моделирования**

**Введение.** К причинам непроизводительных простоев грузовых поездов на станциях можно отнести:

1. Неравномерное поступление поездов на станции,
2. Ограниченные возможности технических станций по обработке поездов (технический и коммерческий осмотры),
3. Отсутствие локомотивов и готовых к рейсу локомотивных бригад [1],
4. Необходимость отцепки вагонов в текущий отцепочный ремонт (ТОР),
5. Отказы технических средств,
6. Дефицит пропускных способностей на прилегающих к станции перегонах,
7. Недостаточная производительность технических средств,
8. Предоставление длительных «окон» [2],
9. Человеческий фактор.

Перечисленные факторы приводят к дополнительным издержкам на эксплуатационную работу [3], неэффективному использованию подвижного состава и пропускной способности инфраструктуры, наруше-

нию нормативных сроков доставки грузов и необходимости выплаты пени.

Все это обостряется в связи с избытком вагонного парка на основных направлениях сети железных дорог РФ [4], что снижает маневренность технических станций и резервы их перерабатывающей способности, что может привести к остановке работы станций и как следствие «лавинообразному» росту непроизводительных простоев.

Поэлементный факторный анализ [5] продвижения поездопотока на грузонапряженных железнодорожных линиях показал, что более 50 % от времени нахождения грузовых поездов в пути следования составляют непроизводительные простои [6-7].

Проблема непроизводительных простоев обостряется на стыках железнодорожного и других видов транспорта. Так, в работе [8], отмечается: «Проблема взаимодействия в транспортных системах является центральной, так как именно этот аспект приводит к значительным потерям: простои подвижного состава в ожидании обслуживания, нарушение сроков достав-