

Плаксин М.А.

Национальный исследовательский университет Высшая школа экономики (Пермский филиал), г.Пермь, к.ф.м.н., доцент кафедры информационных технологий в бизнесе, mapl@list.ru

«СУПЕРКОМПЬЮТЕРЫ» VS «ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ». «ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ» VS «СОВМЕСТНАЯ ДЕЯТЕЛЬНОСТЬ». КАК ИЗУЧАТЬ ТЕМУ «ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ» В СРЕДНЕЙ ШКОЛЕ?

КЛЮЧЕВЫЕ СЛОВА

Информатика, параллельное программирование, параллельные вычисления, параллельные алгоритмы, суперкомпьютеры, начальная школа, средняя школа, ТРИЗформашка.

АННОТАЦИЯ

Статья посвящена вопросу о включении в школьный курс информатики темы «параллельные вычисления». Упоминается ряд возникающих при этом проблем, рассматривается цель изучения темы, отбор материала, некоторые предложения по методике обучения, механизмы апробации предложенной методики и накопленный опыт. Не затрагивается вопрос о месте этого материала в учебной программе.

Современный этап развития computer science связан с массовым распространением параллелизма вычислений на всех уровнях (многомашинные кластеры, многопроцессорные ЭВМ, многоядерные процессоры).

Массовое распространение параллелизма влечет серьезные последствия, которые еще предстоит выявить и проанализировать. Начнем с перечисления некоторых теоретических проблем.

Современная теория алгоритмов создавалась в расчете на понятие последовательного алгоритма. Каким образом отразится на понятии алгоритма отказ от требования последовательности выполнения шагов?

По крайней мере последние 20 лет понятие «алгоритм» вводилось в школе в неразрывной связке с понятием «исполнитель». Для последовательного алгоритма это естественно. Как быть с алгоритмом параллельным? Его выполняет один исполнитель или группа исполнителей? Для конкретности в качестве примера рассмотрим компьютерную обучающую программу «Танковый экипаж» [11]. В этой программе от учащегося требуется запрограммировать действия экипажа танка, состоящего из трех человек: наводчика, водителя и заряжающего. Каждый из них имеет свою систему команд. Для того, чтобы выполнить боевую задачу (поразить все цели), все члены экипажа должны действовать согласованно. Пример игрового поля программы «Танковый экипаж» см. на рис.1.

Вопрос: надо ли рассматривать этих трех действующих лиц как независимых исполнителей или как три составные части (устройства) одного сложного исполнителя? Для экипажа танка более естественным представляется второй вариант, поскольку ни один персонаж сам по себе выполнить задание не в состоянии. Но как быть, если игра будет усложнена, и боевая задача будет поставлена сразу для двух танков? Для трех танков? Трех членов одного экипажа вполне можно рассматривать как три части одного исполнителя. Но каждый экипаж очевидно является самостоятельным исполнителем. Значит, параллельный алгоритм для нескольких танков будет выполняться сразу группой исполнителей. Получается, что для параллельного алгоритма рассматривать надо обе возможности: выполнение параллельных действий одним исполнителем и группой исполнителей. В случае танкового экипажа границу провести просто. Исполнитель — это тот, кто в состоянии решить поставленную задачу. Этот исполнитель может состоять из нескольких компонент, каждая из которых выполняет некую часть задания, но не может самостоятельно без помощи других компонент выполнить задание целиком. Но всегда ли разделение «целых исполнителей» и частей сложного исполнителя будет также просто — сейчас сказать нельзя.

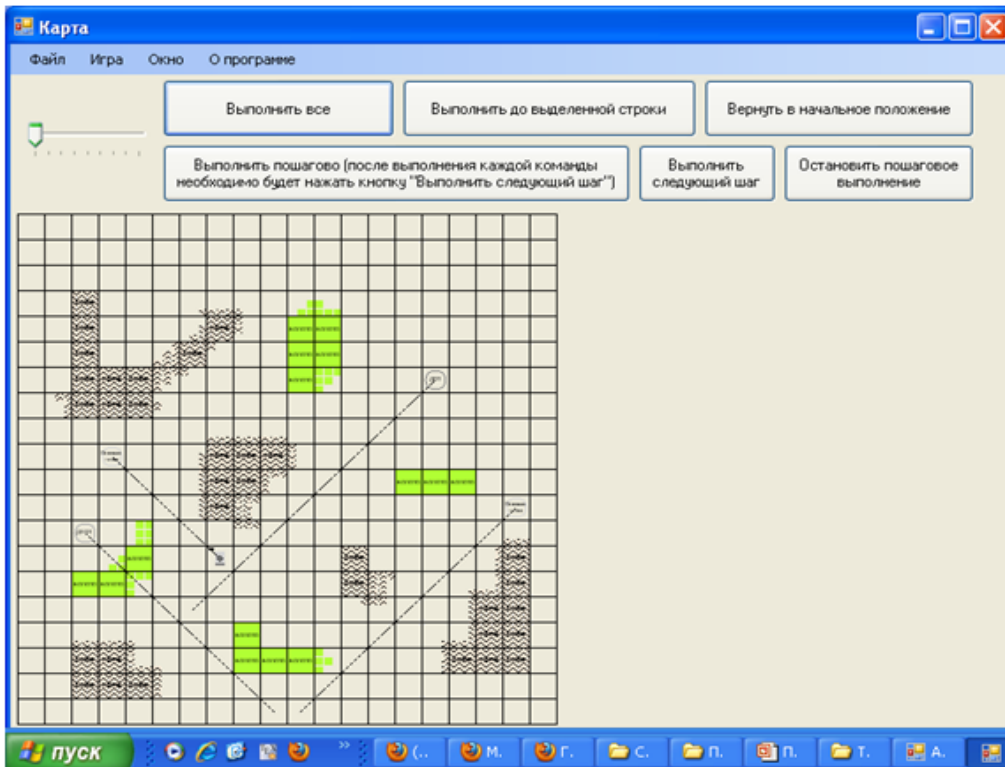


Рис.1. Фрагмент игрового поля программы «Танковый экипаж»

Выделение частей исполнителя, способных к самостоятельным действиям, требует как-то эти части назвать. Причем название должно допускать рекурсию, поскольку действующие части исполнителя сами могут иметь сложную структуру.

Нужно договориться о термине для обозначения группы совместно действующих исполнителей. Термин «команда» не годится, ассоциируется с «системой команд исполнителя» и с «командами центрального процессора». «Коллектив исполнителей»? «Бригада исполнителей»?

#	Наводчик	Водитель	Заряжающий
1	Повернуть орудие по часовой стрелке на 45 гра...	Стоп	Зарядить 1
2	Пауза	Стоп	Зарядить 2
3	Огонь!	Повернуться против часовой стрелки на 90 градусов	Зарядить 1
4	Пауза	Вперед	Зарядить 2
5	Огонь!	Стоп	Зарядить 1
6	Пауза	Стоп	Зарядить 2
7	Огонь!	Стоп	Зарядить 1
8	Пауза	Повернуться по часовой стрелке на 45 градусов	Зарядить 2
9	Пауза	Вперед	Пауза
10	Пауза	Вперед	Пауза
11	Пауза	Вперед	Пауза
12	Пауза	Повернуться по часовой стрелке на 45 градусов	Пауза
13	Пауза	Вперед	Пауза
14	Огонь!	Повернуться по часовой стрелке на 45 градусов	Зарядить 1

Рис.2. Фрагмент программы для «Танкового экипажа» (пример линеек команд)

Требует доработки традиционное понятие «системы команд исполнителя» (СКИ) и само понятие команды. Если мы считаем, что три члена танкового экипажа образуют единого исполнителя, то что считать СКИ этого исполнителя? И что считать командой? Или оставить понятие СКИ для каждого персонажа? То есть это уже не система команд ИСПОЛНИТЕЛЯ, а система команд одной из компонент исполнителя (для которой еще нет названия)?

Понятие команды удобно расширить до «линейки команд». Пример линейек команд танкового экипажа см. на рис.2. Однако понятие «линейки команд» хорошо работает только для линейных алгоритмов. В остальных случаях линейки формируются динамически. Изобразить их в виде наглядной таблицы невозможно.

Среди свойств алгоритмов выделяется новая практически значимая характеристика: способность к распараллеливанию. Уточняющий вопрос — о возможной степени распараллеливания (до какой степени имеет смысл увеличивать количество процессоров при выполнении данного алгоритма).

Отдельный вопрос — методы распараллеливания уже существующих последовательных алгоритмов.

До недавнего времени параллельное программирование было делом небольшого числа высоко квалифицированных системных программистов. Сегодня оно становится частью профессиональной компетенции. Но технология параллельного программирования существенно отличается от традиционного последовательного. В подтверждение этого утверждения вслед за Л.Л. Босовой [2] процитируем крупнейшего российского специалиста в области параллельных вычислений В.В. Воеводина [3, стр.150, 155]:

«... Освоение вычислительной техники параллельной архитектуры ... молодыми специалистами идет с большими трудностями. На наш взгляд, это связано с тем, что знакомство с параллельными вычислениями, как и образование в этой области в целом, начинается не с того, с чего надо бы начинать. К тому же то, с чего надо начинать, не рассказывается ни в каких курсах вообще. Возможность быстрого решения задач на вычислительной технике параллельной архитектуры вынуждает пользователей изменять весь привычный стиль взаимодействия с компьютерами. По сравнению, например, с персональными компьютерами и рабочими станциями меняется практически все: применяются другие языки программирования, видоизменяется большинство алгоритмов, от пользователей требуется предоставление многочисленных нестандартных и трудно добываемых характеристик решаемых задач, интерфейс перестает быть дружественным и т.п. Важным является то обстоятельство, что неполнота учета новых условий работы может в значительной мере снизить эффективность использования новой и, к тому же, достаточно дорогой техники.»

«Важно лишь, чтобы обучающийся как можно раньше узнал, что существуют другие способы организации вычислительных процессов, а не только последовательное выполнение «операция за операцией», что на этих других способах строится самая мощная современная вычислительная техника, что только на такой технике удастся решать крупные промышленные и научные задачи и т.д. Важно, в первую очередь, для того, чтобы как можно раньше обратить внимание обучающихся на необходимость критического отношения к философии последовательных вычислений. Ведь именно с этой философией им приходится сталкиваться на протяжении всего образования как в школе, так и в вузе. И именно эта философия мешает пониманию особенностей работы на вычислительной технике параллельной архитектуры.»

Сегодня нам нужны методики для массового обучения технологии параллельного программирования. Автор данной статьи считает, что в процессе обучения настало время для переворота в отношениях последовательного и параллельного программирования. До сих пор мы сначала учили последовательному программированию, а потом — распараллеливанию последовательных алгоритмов. Сейчас надо ставить вопрос о том, чтобы сразу учить параллельному программированию. А последовательный алгоритм рассматривать как некую часть параллельного алгоритма, которая не требует связи с другими его частями. Как это делать — вопрос открытый. Пока есть некоторые идеи, которые нуждаются в практическом воплощении и апробации. Есть надежда, что через год на следующей конференции можно будет обсудить полученные результаты.

Тридцать лет назад начинающаяся массовая компьютеризация производства потребовала увеличения уровня компьютерной грамотности населения. Это привело к введению в школьную программу в 1985 г. курса информатики. Но курс информатики в советском (затем в российском) исполнении не сводился к «кнопочной информатике» — к освоению технологии работы с пакетами прикладными программ и компьютерными играми. Он начал изменять стиль мышления подрастающего поколения. В первую очередь это касалось алгоритмичности, точности, строгости. Затем курс информатики вообрал в себя элементы логики и системного анализа. Впоследствии все это значительно упростило распространение так необходимого в XXI в. проектного подхода. Сейчас речь идет о том, что в течение следующего десятилетия параллельные алгоритмы должны стать

элементом общей культуры мышления. Вопрос: каким образом скажется на мышлении следующего поколения освоение понятия параллельного алгоритма, к чему приведет перестройка сознания «на параллельный лад»?

Массовое распространение параллельной обработки информации делает актуальным перемещение соответствующих понятий в разряд общедоступных и общекультурных. Знакомство с параллельными алгоритмами должно стать частью грамотности так, как это за последнюю четверть века произошло с базовыми понятиями теории алгоритмов. Сделать это можно только одним путем — включением соответствующих тем в школьный курс информатики. Значит, нужна методика начального знакомства с параллельным программированием на уровне средней школы.

Исторически первая попытка включения тематики параллельных вычислений в школьный курс информатики была сделана еще двадцать лет назад. Двадцать лет назад в курсе под названием «Алгоритмика» [1, 2] был описан исполнитель «Директор строительства», который командовал параллельными действиями нескольких бригад, строящих сооружение из блоков прямоугольной и треугольной формы. Более того, для этого исполнителя была создана программная реализация. Увы! Эта замечательная методическая разработка в середине 90-х оказалась не востребована. Она почти на двадцать лет опередила свое время!

Сегодня положение сложилось так, что тематика параллельных вычислений в средней школе в первую очередь оказалась связана с темой суперкомпьютеров. Именно на суперкомпьютерах акцентируют внимание учащихся авторы различных методических разработок [4, 9, 10, 12, 18], даже тогда, когда в этом нет необходимости. Достаточно сказать, что соответствующий раздел в журнале «Информатика в школе» носит название «Суперкомпьютерное образование в школе». Такая ситуация имеет как положительные, так и отрицательные стороны.

Среди положительных сторон надо назвать:

- интерес, который вызывает в обществе, в том числе, в среде учащихся, тема суперкомпьютеров. Этот интерес повторяет на современном уровне интерес, который полвека назад вызывали большие машины — суперкомпьютеры своего времени;
- организационную поддержку со стороны суперкомпьютерного сообщества. Каждое лето на факультете вычислительной математики и кибернетики МГУ проводится Летняя Суперкомпьютерная Академия [13]. И каждое лето в рамках этой Академии организуется школьный трек для учителей информатики. Обучение проводится бесплатно. Иногородние слушатели обеспечиваются жильем на весьма льготных условиях. На конференции Russian Supercomputing Days в сентябре 2015 г. была организована школьная секция и мастер-класс для учителей информатики. Последовательная организационная работа привела к выявлению и формированию группы учителей, заинтересованных в продвижении данной тематики;
- наличие яркого харизматичного лидера, каковым является Владимир Валентинович Воеводин — доктор физико-математических наук, профессор, член-корреспондент РАН, заместитель директора Научно-исследовательского вычислительного центра Московского государственного университета;
- интерес и поддержку (в том числе, материальную) со стороны российского представительства фирмы Интел и менеджера по стратегическому развитию фирмы Интел Игоря Олеговича Одинцова.

Недостаток «суперкомпьютерного» подхода заключается в зауживании тематики параллельных вычислений. Сами суперкомпьютеры школьникам, как правило, недоступны (разве что в крупных городах на них можно поглазеть на экскурсии). Задачи, на решение которых они нацелены, для школьников слишком сложны и, в большинстве случаев, не имеют непосредственной практической значимости и не представляют практического интереса.

Естественным расширением суперкомпьютерной тематики является изучение параллельного программирования. В настоящее время для выполнения параллельных программ совсем не обязательно иметь суперЭВМ. Достаточно многоядерного процессора или видеокарты с набором графических ускорителей. А это доступно уже почти всем. Из работ в этом направлении отметим кандидатскую диссертацию М.А. Соколовской по методике обучения будущих учителей информатики основам параллельного программирования [18] и опыт Е.Ю. Киселевой по освоению школьниками технологии CUDA [10].

По мнению автора данной статьи сосредоточение внимания на суперЭВМ и параллельном программировании существенно обедняет и усложняет тему параллельных вычислений, отвлекает учащихся от множества важных и доступных вопросов. Целью темы «параллельные

вычисления» в средней школе является не обучение «реальному» параллельному программированию (изучение соответствующих языковых конструкций, языков программирования и технологий), а ознакомление учащихся с соответствующим набором понятий и понимание особенностей параллельной работы. Мир вокруг и внутри нас представляет собой сложную параллельную систему. И эта система сама по себе дает массу материала для освоения понятий и механизмов параллелизма. Никакие сложные искусственные конструкции типа технологий MPI и OpenMP для этого не нужны. Школьная информатика должна воспитать мышление, настроенное на «параллельный лад». А дальше университет пусть закладывает в это мышление профессиональные знания, умения, навыки. В школе акцентировать имеет смысл не знакомство с суперкомпьютерами и изучение параллельного программирования, а освоение механизмов «совместной деятельности», постоянно и широко используемых в жизни.

В курсе предлагается отразить следующие вопросы:

- 1) Совместная работа нескольких исполнителей (копание канавы несколькими землекопами) и распараллеливание «внутри» одного исполнителя при наличии нескольких обрабатывающих устройств (читаю и ем яблоко). В computer science это будут многомашинный комплекс и многоядерный процессор.
- 2) Виды параллелизма: параллелизм истинный и псевдопараллелизм (один процессор выполняет частями несколько программ).
- 3) Исполнители однотипные (землекопы) и разнотипные (экипаж танка).
- 4) Работы однотипные и разнотипные.
- 5) Соотношение «исполнители — работы»: 1 исполнитель — 1 работа, 1 исполнитель — N работ (псевдопараллельное выполнение или истинный параллелизм при наличии нескольких обрабатывающих устройств для разных работ), N исполнителей — 1 работа, N исполнителей — N работ.
- 6) Согласование деятельности исполнителей. Виды согласования: по частям работы, по времени, по результатам деятельности, по ресурсам.
- 7) Ресурсы. Ресурсы разделяемые и неразделяемые, расходуемые и повторно используемые. Утилизация потребленных ресурсов («сборка мусора» в широком смысле).
- 8) Выполнение одной и той же работы одним исполнителем и группой исполнителей. Зависимость скорости работы от количества исполнителей. Зависимость стоимости работы от количества исполнителей. Нелинейный рост скорости работы при росте количества исполнителей. Критический путь. Оптимальное количество исполнителей. Оптимальная загрузка исполнителей. Оптимальный порядок действий. Балансировка нагрузки.
- 9) Конкуренция исполнителей за ресурсы. Блокировка. Клинч (тупик).
- 10) Механизмы согласования действий исполнителей.
- 11) Псевдопараллельное выполнение процессов на компьютере (разделение между исполнителями-процессами одного ресурса — процессора).
- 12) Пригодность алгоритмов к распараллеливанию. Возможная степень распараллеливания. Существование алгоритмов, не поддающихся распараллеливанию.

Отметим, что приведенный список представляет собой частное мнение автора статьи и открыт для обсуждения, дополнения и корректировки. Более того, по мнению автора было бы очень полезно, чтобы «суперкомпьютерное сообщество» сформулировало «социальный заказ» для школы: какие именно знания-умения-навыки оно хочет видеть в выпускниках школы. Чем выпускник школы «суперкомпьютерного мира» должен отличаться от выпускника сегодняшнего? Будет заказ — будет и результат. Свежий пример. В первый день Russian Supercomputing Days-2015 в двух докладах прозвучала мысль, что быстродействие современных суперЭВМ определяется не мощностью процессоров (которая находится в центре внимания публики), а быстродействием оперативной памяти. Именно она становится бутылочным горлышком, пропускная способность которого определяет продуктивность всей системы. В результате на второй день конференции участники учительского мастер-класса обкатывали придуманную автором данной статьи игру, демонстрирующую взаимодействие центрального процессора, оперативной памяти и кэш-памяти.

Порядок и форма изложения материала — вопрос открытый.

Материал должен быть продемонстрирован на примерах, не связанных с работой ЭВМ. Исполнители должны манипулировать материальными объектами.

Как можно большая часть обучения должна носить характер деловых (организационно-деятельностных) игр.

Выполнение этих требований упростит понимание изучаемого материала. Это будет полезно как при использовании данной методики на уроках информатики в школе (в том числе, начальной!), так и при обучении взрослых: учителей информатики и студентов. Школьник, школьный учитель, студент непрофильной специальности смогут остановиться на уровне ознакомления и понимания. Студент-профессионал должен будет сделать следующий шаг и от знакомства перейти к изучению этих механизмов на профессиональном уровне. Но это уже — шаг за пределы методики начального ознакомления с темой.

Работу над подготовкой методики изучения параллельных вычислений автор данной статьи начал в 2013 г. в ходе подготовки конкурса «ТРИЗформашка-2013» и продолжил в последующие годы [6, 7, 8, 17].

«ТРИЗформашка» — межрегиональный Интернет-конкурс по информатике, системному анализу и ТРИЗ. Проводится ежегодно во второй половине марта. Возраст участников — с I класса до IV курса. География — от Владивостока до Риги. Среднее число участников — 100 команд (300 чел.), максимальное — 202 команды (более 600 чел.). Сайт конкурса www.trizformashka.ru.

Тогда, в 2013 г. цель работы была сформулирована следующим образом:

1. В течение двух-трех лет подготовить описание исполнителей, набор игр и задач, связанных с параллельными вычислениями;
2. Предложить их (по частям, ежегодно) участникам конкурса;
3. Проанализировать их реакцию (оценить количество решавших, их возраст, успешность решения, типичные ошибки, обнаруженные неточности в формулировке задач и т.д.).

Конкурс «ТРИЗформашка» оказался удобным инструментом отладки задач, поскольку позволял получить реакцию всех возрастов (от I класса до IV курса), из различных регионов, из различных учебных заведений.

За прошедшие годы был подготовлен следующий набор методических инструментов и площадок для их апробации.

1. Задания на параллелизм, начиная с 2013 г., вошли в конкурс «ТРИЗформашка» (начиная с 2013 г., конкурс имеет подзаголовок «Параллельные вычисления»). Список типов заданий приведен ниже;
2. Подготовлена глава про параллелизм для новой версии учебника информатики для 4 класса [14]. Материал прошел апробацию в 3-х и 4-х классах Лицея №10 г.Перми;
3. Разработана и с 2014 г. используется в конкурсе «ТРИЗформашка» компьютерная игра «Танковый экипаж» [11];
4. Разработан и прошел апробацию ряд игр [16], в которых отражены следующие вопросы:
 - согласование деятельности исполнителей. Различные виды согласования;
 - выполнение одной и той же работы одним исполнителем и группой исполнителей. Зависимость скорости работы от количества исполнителей. Нелинейный рост скорости работы при росте количества исполнителей. Критический путь. Оптимальное количество исполнителей. Оптимальная загрузка исполнителей. Оптимальный порядок действий;
 - ресурсы. Ресурсы разделяемые и неразделяемые;
 - конкуренция исполнителей за ресурсы. Блокировка. Клинч (тупик).

Были предложены и опробованы следующие типы задач [15]:

1. Задачи на виды согласования. (Какие виды согласования существуют в школьной столовой?);
2. Игра «Танковый экипаж». Задание на построение параллельного алгоритма;
3. Исполнитель «Стройка» [5]. Одновременно работающие бригады строят сооружение из горизонтальных и вертикальных балок. Задания включают в себя задания на исполнение указанного алгоритма, на разработку нового алгоритма, на поиск ошибок в заданном алгоритме, на исследование алгоритмов (сравнение сроков строительства по разным алгоритмам, сравнение стоимости строительства, оценка возможности сэкономить за счет перераспределения рабочей силы и др.);
4. Конкуренция за ресурсы. Три поросенка готовят каждый сам себе обед. Для каждого поросенка указано, какие блюда он готовит, какие ресурсы (оборудование, посуда и т.д.) ему для этого нужны и в течение какого времени эти ресурсы должны использоваться. Требуется составить график работы каждого поросенка, если он готовит на кухне один, если они готовят парами, если готовят все трое сразу. Время приготовления пищи должно быть минимизировано;

5. Сетевой график. Дан сетевой график. Требуется изобразить (схематически) сооружение, которое будет построено, определить, сколько дней потребуется для строительства при том или ином числе бригад, какая часть работы будет выполнена к определенному времени;
6. Ярусно-параллельные формы. Планирование работ по разным критериям. Дано задание на работу, производительность работников, правила оплаты. Требуется определить количество работников, нужных, чтобы выполнить работу в заданное время, определить срок работы при заданном количестве работников, определить количество работников, нужное для минимизации стоимости работ;
7. Диаграммы Ганта. Описан текстом план работ по реконструкции цеха: продолжительность и взаимная последовательность действий, требуемые работники. Требуется определить срок сдачи объекта, изменение срока при тех или иных изменениях в рабочей силе, список работников, задействованных на конкретную дату.
8. Согласование повторяющихся работ. Пусть дано задание в минимальный срок изготовить партию приборов, при условии, что каждый прибор должен пройти обработку на разном оборудовании, имеется разное количество оборудования с разной производительностью. Требуется спланировать время начала и работы каждого оборудования, минимизировать простой.

На сегодня имеем в наличии следующие результаты:

1. Сформулирован подход к изучению темы «параллельные вычисления»: идти не от проблем computer science, а «от жизни», делать акцент на «совместной деятельности»;
2. Сформулирован перечень вопросов, которые предлагается отразить в начальном курсе параллельных вычислений;
3. Сформулированы некоторые классы задач. На основании накопленного опыта можно оценить, какого рода задачи стоит придумывать;
4. Подготовлен набор задач названных классов. Задачи прошли апробацию в конкурсах «ТРИЗформашка» за 2013, 2014, 2015 гг. и/или в начальной школе (на занятиях с учениками третьих-четвертых классов лицея №10 г.Перми);
5. Подготовлен набор деловых игр. Игры прошли апробацию в начальной школе и на ряде мероприятий для учителей. В частности, были представлены на школьном треке Летней Суперкомпьютерной Академии ВМК МГУ в 2014 г., на мастер-классе для учителей на Russian Supercomputing Days-2015, на нескольких других конференциях (в том числе, на конференции ИТ-Образование-2015 ассоциации АПКИТ) и других мероприятиях для учителей информатики;
6. Подготовлен набор текстов про параллелизм для учебника IV класса. Тексты прошли апробацию в лицее №10 г.Перми;
7. Подготовлена компьютерная игра «Танковый экипаж». Игра прошла апробацию в конкурсах «ТРИЗформашка» 2014 и 2015;
8. Конкурс «ТРИЗформашка» оправдал себя в качестве апробационной площадки;
9. Сформулирована задача «провести рокировку» в процессе обучения алгоритмизации: учить сразу параллельному программированию, представляя последовательный алгоритм частью параллельного. Есть мысли о том, как можно реализовать эту идею. Есть возможность опробовать эти идея в течение текущего учебного года (на учащихся 4-х — 5-х классов);
10. Есть потребность, желание и возможность продолжать работу.

Литература

1. Алгоритмика: 5-7 классы: Учебник и задачник для общеобразоват. учебных заведений /А.К. Звонкин, А.Г. Кулаков, С.К. Ландо, А.Л. Семенов, А.Х. Шень. — М.: Дрофа, 1996.
2. Босова Л.Л. Параллельные алгоритмы в начальной и основной школе. //Информатика в школе. 2015, №2. С.24-27.
3. Воеводин В.В. Вычислительная математика и структура алгоритмов: 10 лекция о том, почему трудно решать задачи на вычислительных системах параллельной архитектуры и что надо знать дополнительно. чтобы успешно преодолевать эти трудности: учебник. М.: Изд-во МГУ, 2010.
4. Гаврилова И.В. Первое путешествие в «параллельный мир». //Информатика в школе. 2015, №6. С.16-19.
5. Дитер М.Л., Плаксин М.А. Параллельные вычисления в школьной информатике. Игра «Стройка». //Информатика в школе: прошлое, настоящее и будущее.: материалы Всеросс. науч.-метод. конф. по вопросам применения ИКТ в образовании, 6-7 февраля 2014 г. /Перм. гос. нац. иссл. ун-т — Пермь, 2014. — С.258-261.
6. Иванова Н.Г., Плаксин М.А., Русакова О.Л. ТРИЗформашка. //Информатика. N05 (606), 1-15.03.2010. С.3-19.
7. Иванова Н.Г., Плаксин М.А., Русакова О.Л. Задачи на параллельное программирование в конкурсе «ТРИЗформашка-

- 2013». //Информационные технологии в образовании. XXIII Международная конференция-выставка: Сборник трудов. Ч.II. — М.: Издательский отдел факультета ВМК МГУ им.М.В.Ломоносова, 2013. С.9-10.
8. Иванова Н.Г., Плаксин М.А., Русакова О.Л. Конкурс «ТРИЗформашка» как площадка для апробации заданий на параллельное программирование. //Информатика в школе: прошлое, настоящее и будущее.: материалы Всеросс. науч.-метод. конф. по вопросам применения ИКТ в образовании, 6-7 февраля 2014 г. /Перм. гос. нац. иссл. ун-т. — Пермь, 2014. — С.233-236.
 9. Киселева Е.Ю. Реконструкция параллельной модели вычислений на примере задачи суммирования чисел. //Информатика в школе. 2013, №10.
 10. Киселева Е.Ю. Потенциал суперкомпьютерной тематики в проектно-исследовательской деятельности учащихся. //Информатика в школе. 2015, №2. С.20-23
 11. Кучев А.Д., Плаксин М.А. Параллельные вычисления в школьной информатике. Игра «Танковый экипаж». //Информатика в школе: прошлое, настоящее и будущее.: материалы Всеросс. науч.-метод. конф. по вопросам применения ИКТ в образовании, 6-7 февраля 2014 г. /Перм. гос. нац. иссл. ун-т. — Пермь, 2014. — С.241-243.
 12. Логинов А.В. Исторические предпосылки и перспективы суперкомпьютерного образования в школьном курсе информатики. //Информатика в школе. 2015, №2. С.17-19.
 13. Материалы Летней Суперкомпьютерной Академии. <http://academy.hpc-russia.ru/>. [Электронный ресурс] Проверено 10.10.2015.
 14. Плаксин М.А. Информатика: учебник для 4 класса: в 2 ч. /М.А.Плаксин, Н.Г.Иванова, О.Л.Русакова. — М.: БИНОМ. Лаборатория знаний, 2012.
 15. Плаксин М.А. О методике начального знакомства с параллельными вычислениями в средней школе. //Информатика в школе: прошлое, настоящее и будущее.: материалы Всеросс. науч.-метод. конф. по вопросам применения ИКТ в образовании, 6-7 февраля 2014 г. /Перм. гос. нац. иссл. ун-т. — Пермь, 2014. — С.256-258.
 16. Плаксин М.А. Комплекс деловых игр для знакомства с параллельными вычислениями в начальной школе. //Преподавание информационных технологий в Российской Федерации: материалы Тринадцатой открытой Всероссийской конференции «ИТ-Образование-2015» (г.Пермь, 14-15 мая 2015 г.). Пермский государственный национальный исследовательский университет, — Пермь, 2015. С.60-62.
 17. Плаксин М.А., Иванова Н.Г., Русакова О.Л. Набор заданий для знакомства с параллельными вычислениями в конкурсе «ТРИЗформашка». //Преподавание информационных технологий в Российской Федерации: материалы Тринадцатой открытой Всероссийской конференции «ИТ-Образование-2015» (г.Пермь, 14-15 мая 2015 г.). Пермский государственный национальный исследовательский университет, — Пермь, 2015. С. 232-234.
 18. Соколовская М.А. Методическая система обучения основам параллельного программирования будущих учителей информатики.: автореф. дис. ... канд. пед. наук, Красноярск, 2012.