***Хусаинов А.А.**, доктор физико-математических наук, профессор,*

***Чернов А.М.**, магистрант*

(Комсомольский-на-Амуре государственный технический университет)

***Маевская Е.Д.**, студент*

***Романченко А.А.**, студент*

(Амурский гуманитарно-педагогический государственный университет)

МОДЕЛИ ДЛЯ РАСЧЕТА ВРЕМЕНИ РАБОТЫ ВЫЧИСЛИТЕЛЬНЫХ КОНВЕЙЕРОВ

Работа посвящена изучению компьютерной модели асинхронного вычислительного конвейера, построенного с помощью многопоточного приложения. Дано доказательство формулы для нахождения времени работы этого конвейера. Рассмотрена задача об ограниченном конвейере. Предложена формула для нахождения времени обработки данных с помощью ограниченного конвейера. Для исследования ограниченного конвейера применена теория моноидов трасс. Введена ограниченная нормальная форма трассы и построен алгоритм приведения трассы к ограниченной нормальной форме. На основе этого алгоритма разработано программное обеспечение для расчета времени работы ограниченного конвейера.

***Ключевые слова:** вычислительный конвейер, многопоточное приложение, синхронизация, событие, моноид трасс, нормальная форма Фоаты, производительность.*

§1. Компьютерная модель асинхронного вычислительного конвейера

Вычислительные конвейеры широко применяются при создании современных процессоров. Проблема расчета времени работы конвейеров с различ-

Хусаинов А.А., Чернов А.М., Маевская Е.Д., Романченко А.А. Модели для расчета времени работы вычислительных конвейеров // Актуальные проблемы науки: Материалы XXIII Международной научно-практической конференции (11.01.2016). – М.: Издательство «Спутник+» (01.02.2016), 2016. с.83-91.

ными свойствами актуальна. Она связана с нахождением оптимального числа ступеней конвейера. Этому вопросу посвящено много работ, укажем на обзоры [1]-[2]. Один из способов изучения конвейеров – разработка компьютерной модели.

Асинхронный вычислительный конвейер состоит из конечной последовательности функциональных устройств связанных между собой устройствами передачи данных. Эти функциональные устройства называются *ступенями конвейера*, а устройства передачи данных – *каналами*.

В общем случае каждая из ступеней конвейера производит 3 действия: читает данные из входного канала, выполняет некоторую операцию над данными и записывает результат этой операции в выходной канал.

Процесс работы асинхронного вычислительного конвейера можно описать с помощью сети Петри, ее переходы соответствуют ступеням конвейера:

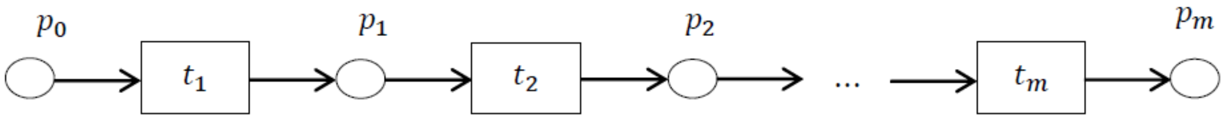

Рис. 1. Сеть Петри вычислительного конвейера

Для исследования задач, связанных с расчетом времени работы конвейеров, мы рассмотрим математическую модель конвейера. Обозначим через $\mathbb{N} = \{0, 1, 2, \dots\}$ множество неотрицательных целых чисел. Пусть Ω – множество символов операций, на которые разлагается обработка элементов входных данных конвейера. Символ операции может быть задан как число, которое является номером этой операции. Будем предполагать, что для каждого символа операции $\omega \in \Omega$ задано число $\tau(\omega) \in \mathbb{N}$, его интерпретацией является время выполнения операции.

Математическая модель асинхронного вычислительного конвейера задается с помощью пятерки $(\mathfrak{P}_m, n, r, w, \alpha)$, где

Хусаинов А.А., Чернов А.М., Маевская Е.Д., Романченко А.А. Модели для расчета времени работы вычислительных конвейеров // Актуальные проблемы науки: Материалы XXIII Международной научно-практической конференции (11.01.2016). – М.: Издательство «Спутник+» (01.02.2016), 2016. с.83-91.

1) $\mathfrak{P}_m$ - показанная на рис.1 сеть Петри с множеством мест $P = \{p_0, \dots, p_m\}$ и множеством переходов $T = \{t_1, \dots, t_m\}$. Число n определяет начальную маркировку этой сети Петри $M(p_0) = n$ и $M(p_i) = 0$, для всех $1 \le i \le m$.
2) $r: T \to \mathbb{N}$ - функция, сопоставляющая каждому переходу $t_i \in T$ время $r(t_i)$ чтения элемента данных из канала p_{i-1}, $w: T \to \mathbb{N}$ - функция, значения которой равны времени записи $w(t_i)$ в канал p_i.
3) $\alpha: T \to \Omega$ – функция, сопоставляющая каждому переходу символ операции, выполняющейся этим переходом.

При $1 \le i \le m$ обозначим через $\tau(t_i)$ время $\tau(\alpha(t_i))$. Если i-я ступень конвейера, при $i \ge 1$, выполняется как последовательный процесс, то время работы этой ступени равно сумме $r(p_{i-1}) + \tau(t_i) + w(p_i)$.

Опишем *компьютерную модель* – программу, имитирующую одновременную работу ступеней конвейера, при которой каждая ступень t_i, $i \ge 1$, ожидает окончания выполнения предшествующей ступени t_{i-1}. Она построена аналогично модели, описанной в [3]. Основное отличие – алгоритм обмена данными через канал.

Ступень конвейера реализуется как поток (нить). Главная программа загружает m потоков. Каждый из потоков сначала определяет свой номер i, $1 \le i \le m$, и затем выполняет n-кратный цикл

```
for (int k=0; k<n;i++)
{
    p[i-1].waitr(); p[i-1]>>x; y=a(x); p[i].waitw(); p[i]<<y;
}
```

Класс, объектами которого являются каналы, имеет несколько отличий от класса, рассмотренного в работе [3]:

1) Буфер канала состоит из одного элемента, тогда как в [3] он содержал очередь элементов.
2) Синхронизация обмена данными между потоком и каналом осуществляется с помощью событий. Для каждого канала определены два события.

Первое событие – можно читать данные из канала, второе событие – можно писать данные в канал.

Тело n-кратного цикла i-го потока состоит из следующих операторов

1) `p[i-1].waitr()` – ожидание появления данных во входном канале;
2) `p[i-1]>>x` – чтение данных из канала в локальную переменную потока;
3) `y=a(x)` – выполнение операции;
4) `p[i].waitw()` – ожидание освобождения выходного канала для записи;
5) `p[i]<<y` – запись результата операции в выходной канал.

§2. Формула для расчета времени обработки входных данных конвейера

Компьютерная модель используется для расчета времени работы следующим образом. К операциям чтения из канала и записи в канал добавляется целое неотрицательное число – задержка операции, в миллисекундах. Вместо операции `y=a(x)` подставляется время выполнения этой операции.

Программа содержит цикл обработки данных. В этом цикле число элементов входных данных n изменяется с некоторым шагом, который может быть установлен пользователем. Начальное значение n равно этому шагу. Конечное значение n не больше 100. Тело цикла начинает выполнение с загрузки потоков. Время отсчитывается после окончания загрузки потоков. Запускается поток, передающий данные в канал p_0, и конвейер начинает работу. В конце тела цикла программа ожидает конца работы последнего потока. После этого заканчивается отсчет времени и данные о времени выводятся.

Обозначим через $T(n)$ время обработки n элементов данных конвейером. Каждый поток будет выполняться как последовательный процесс. Отсюда легко видеть, что время $T(1)$ обработки конвейером первого элемента входных данных будет равно $\sigma = \sum_{i=1}^{m} r(p_{i-1}) + \tau(t_i) + w(p_i)$. Для обработки остальных $n-1$ элементов будет выполняться поток, имеющий максимальное время $\mu = \max\limits_{1 \le i \le m} \{r(p_{i-1}) + \tau(t_i) + w(p_i)\}$. Он будет выполняться последовательно, отсюда вытекает неравенство $T(n) \geq \sigma + (n-1)\mu$. Введем обозначение $\tau_i = r(p_{i-1}) + \tau(t_i) + w(p_i)$.

Предложение 1. Минимальное время обработки n элементов входных данных с помощью асинхронного вычислительного конвейера равно $T(n) = \sigma + (n-1)\mu$, где $\sigma = \sum_{i=1}^{m} \tau_i$, $\mu = \max\limits_{1\leq i\leq m}\{\tau_i\}$.

Доказательство. В силу сделанных выше замечаний, достаточно доказать неравенство $T(n) \leq \sigma + (n-1)\mu$. Для этой цели воспользуемся *таблицей занятости*, аналогичной той, что была рассмотрена в книге Коуги [4] для анализа синхронных конвейеров. Она состоит из m строк, состоящих из одинаковых клеток. Каждая строка посвящена ступени конвейера. Номера столбцов соответствуют начальным моментам единичных интервалов времени. Символ, находящийся в i-й строке и в j-м столбце означает, что в j-й интервал времени выполнялся поток с номером i. Если этот символ равен k, то обрабатывался -й элемент и, значит, выполнялась k-я итерация цикла, содержащегося в потоке. Например, при $m = 4$, $\tau_1 = 2$, $\tau_2 = 3$, $\tau_3 = 1$, $\tau_4 = 2$, мы получаем следующую таблицу занятости при обработке первого элемента:

	0	1	2	3	4	5	6	7	8
t_1	1	1							
t_2			1	1	1				
t_3						1			
t_4							1	1	

Для того, чтобы указать вариант последовательности выполнения потоков, время выполнения которой равно $\sigma + (n-1)\mu$, мы сдвинем таблицу обработки первого элемента вправо, на $\mu = \max\limits_{1\leq i\leq m}\{\tau_i\}$ клеток. Получим таблицу занятости при обработке двух элементов. В частности, для приведенного выше примера, все единицы будут сдвинуты вправо на 3 клетки.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
t_1	1	1		2	2		3	3							
t_2			1	1	1	2	2	2	3	3	3				
t_3						1			2			3			
t_4							1	1		2	2		3	3	

Обработка второго элемента начнется через 3 интервала времени. Затем сдвигаем таблицу еще на $\mu = \max\limits_{1 \le i \le m}\{\tau_i\}$. Для нашего примера, мы получили приведенную выше таблицу занятости при обработке трех элементов. Продолжая это построение, мы получим вариант выполнения потоков, при котором время обработки n элементов будет равно $\sigma + (n-1)\mu$. Поскольку $T(n)$ – минимальное время выполнения, то получаем неравенство $T(n) \le \sigma + (n-1)\mu$. Доказательство завершает установленное выше неравенство $T(n) \ge \sigma + (n-1)\mu$.

§3. Задача об ограниченном конвейере

Многопоточное приложение, построенное на основе описанной выше компьютерной модели конвейера, можно применять для ускорения вычисления значений композиции функций на массивах данных. Оно будет работать наиболее производительно, если будет выполняться под управлением процессоров с симметричным доступом к общей памяти, число которых не меньше, чем число ступеней процессора. Эти процессоры часто называют ядрами, а кристалл, содержащий ядра – многоядерным процессором. Если это условие не выполняется, то это приложение называется ограниченным конвейером. Наша задача – найти формулу для расчета времени обработки n элементов с помощью ограниченного конвейера, работающего под управлением $q \ge 1$ процессоров. Мы получим эмпирическую формулу.

Для решения этой задачи и для проведения экспериментов мы строим компьютерную модель ограниченного конвейера следующим образом. Мы вводим семафор common, первоначальное и максимальное значение которого равно числу процессоров $q \geq 1$. Цикл обработки n элементов с помощью потока, соответствующего ступени конвейера, преобразуется следующим образом.

```
for (int k=0; k<n;i++)
{
     p[i-1].waitr();
     P(common);
     p[i-1]>>x; y=a(x); p[i].waitw();
     V(common);
     p[i]<<y;
}
```

Здесь P(common) и V(common) - операторы уменьшения и увеличения счетчика семафора.

Для измерения времени работы построенной компьютерной модели мы, вместо выполнения вычислительной операция перехода задерживаем поток с помощью подпрограммы $Sleep(\tau(t_i))$. Подпрограмма $Sleep(\tau(t_i))$ позволяет имитировать одновременную работу потоков. Время чтения из канала и записи в канал устанавливаются с помощью операторов $Sleep(r(p[i-1]))$ и $Sleep(w(p[i]))$.

Эксперименты показали, что время обработки n элементов, с помощью ограниченного конвейера, не изменяется, если не изменяются значения суммы $\sigma = \sum_{i=1}^{m} \tau_i$ и максимума $\mu = \max\limits_{1 \leq i \leq m} \tau_i$, где $\tau_i = r(p_{i-1}) + \tau(t_i) + w(p_i)$. Это свойство позволяет перебрасывать время обработки с одних ступеней на другие. Многочисленные эксперименты привели к следующей гипотезе.

Гипотеза. Время обработки n элементов входных данных с помощью ограниченного конвейера, работающего под управлением $q \geq 1$ процессоров, равно

$$T_q(n) = \begin{cases} \sigma + \sigma \left[\dfrac{n-1}{q}\right], & \text{при } 1 \leq q \leq \dfrac{\sigma}{\mu}\ ; \\ \sigma + (n-1)\,\mu\,, & \text{если} \quad q \geq \dfrac{\sigma}{\mu}\ . \end{cases}$$

Здесь $\left[\frac{n-1}{q}\right]$ обозначает наибольшее целое число, не превосходящее дробь $\frac{n-1}{q}$.

На рис. 2 показаны типичные результаты работы компьютерной модели ограниченного конвейера. Пользователь ввел данные о числе ступеней конвейера (устройств), времени выполнения операций, а также времени чтения и записи данных. Они отображаются в таблице

	$i = 0$	$i = 1$	$i = 2$	$i = 3$	$i = 4$
$w(p_i)$		1	0	1	1
$\tau(t_{i+1})$	3	2	5	4	3
$r(p_i)$	0	1	1	0	1

Программа выполнялась при числе процессоров $q = 1,2,3,4,5$. Выведенные графики показывают зависимость времени обработки от числа n элементов входных данных, n изменяется от 1 до 100, с некоторым шагом. Верхний график соответствует числу процессоров $q = 1$. Самый нижний график соответствует значению $q = 5$. Этот график сливается с графиком, полученном при $q = 4$. Маленькими кружками обозначаются точки, полученные с помощью измерения времени. График функции, определенной нашей гипотезой, выведен с помощью линий.

Хусаинов А.А., Чернов А.М., Маевская Е.Д., Романченко А.А. Модели для расчета времени работы вычислительных конвейеров // Актуальные проблемы науки: Материалы XXIII Международной научно-практической конференции (11.01.2016). – М.: Издательство «Спутник+» (01.02.2016), 2016. с.83-91.

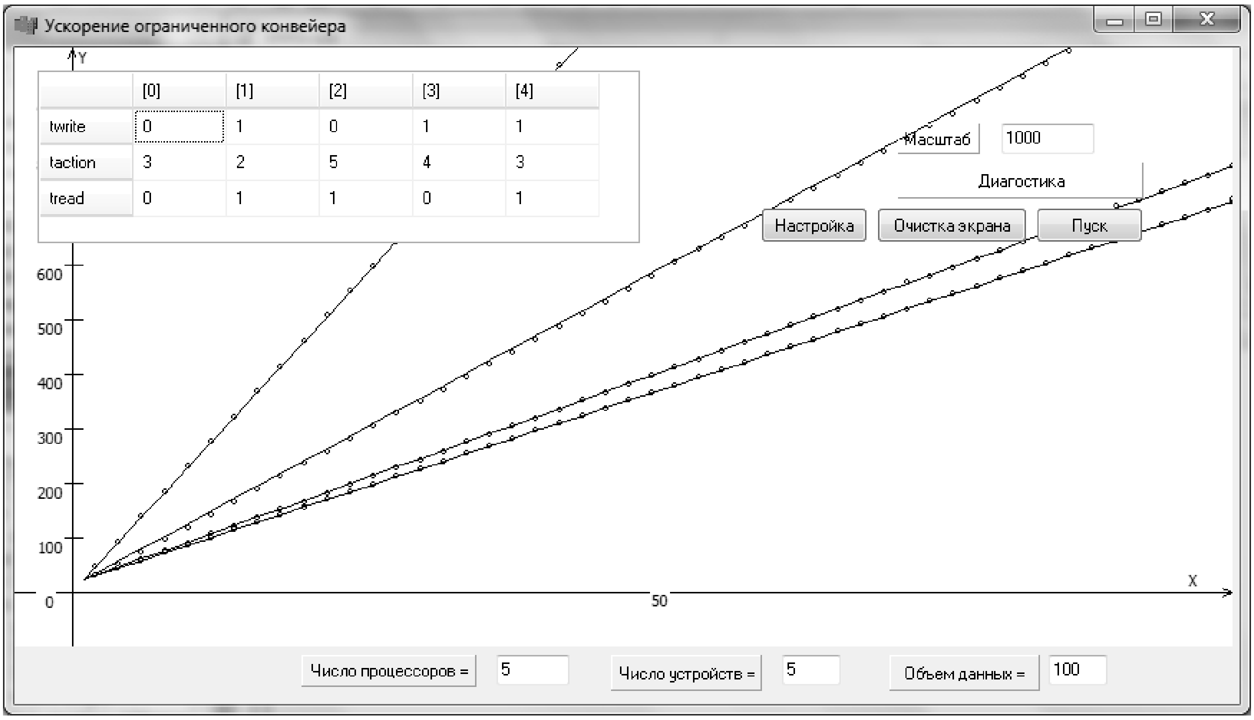

Рис. 2. Результаты расчета времени работы ограниченного конвейера

Графики показывают совпадение значений функций $T_q(n)$ с результатами измерения времени обработки данных с помощью ограниченного конвейера.

§4. Моделирование ограниченного конвейера с помощью трасс

Для расчета времени обработки массива данных с помощью асинхронного вычислительного конвейера применяется метод, описанный в работе [3]. Этот метод можно применять и для вычислительных систем, более общих, чем конвейер. Каждая операция, например, операция чтения, операция записи или вычислительная операция, разлагается в композицию более мелких операций одинаковой продолжительности. Мы будем называть эти мелкие операции *микрооперациями*. Продолжительность микрооперации берется за единицу времени. Микрооперации называются *независимыми*, если они могут выполняться параллельно (одновременно). Пусть E – множество микроопераций, а $I \subset E \times E$ – отношение независимости. Рассмотрим свободный моноид E^*, элементами которого являются слова $a_1 \cdots a_n$, состоящие из букв множества E. Число

n называется длиной слова $a_1 \cdots a_n$ и обозначается через $|a_1 \cdots a_n|$. Единицей 1 этого моноида будет пустое слово, его длина равна нулю. Мы будем говорить, что буквы слова $a_1 \cdots a_n$ упорядочены в алфавитном порядке, если на E задано отношение линейного порядка, и для всех $1 \leq i \leq j \leq n$ имеет место $a_i \leq a_j$. *Моноидом трасс* $M(E,I)$ называется фактор-моноид $E^*/\equiv$ по наименьшему отношению конгруэнтности, удовлетворяющему $ab \equiv ba$ для всех пар $(a,b) \in I$. Моноидам трасс посвящены работы [6]-[7]. Для каждого слова $w \in E^*$ обозначим через $[w] \in M(E,I)$ ее класс конгруэнтности.

Элементы из $M(E,I)$ называются *трассами*. Нормальной формой Фоаты трассы $[w] \in M(E,I)$ называется разложение $[w] = [w_1] \cdots [w_n]$, в котором

1) буквы каждого из слов w_i упорядочены в алфавитном порядке;
2) буквы каждого из слов w_i попарно независимы;
3) при $i \geq 2$ каждая буква слова из w_i зависима с некоторой буквой из слова w_{i-1}.

Трассы $[w_i]$ называются блоками разложения, а их число $h([w]) = n$ называется высотой нормальной формы трассы $[w]$. Для всякой трассы существует единственная нормальная форма Фоаты. Известны алгоритмы нахождения нормальной формы Фоаты для произвольной заданной трассы [7]. Высота нормальной формы Фоаты трассы замечательна тем, что она равна времени выполнения параллельного процесса, соответствующего этой трассе [6].

Заметим, что предложение 1 было доказано в [8] с помощью этого свойства.

Для ограниченного конвейера аналогичного метода расчета времени работы не было, ибо число параллельно работающих микроопераций трассы w_i может оказаться больше, чем число процессоров q.

Чтобы исправить это положение, мы вводим *нормальную q-форму*. Если ввести ее аналогично нормальной форме Фоаты, то возникнут проблемы, связанные с единственностью. Мы поступим иначе. Модифицируем алгоритм приведения нормальной формы Фоаты таким образом, чтобы каждый блок имел не более, чем q букв.

Алгоритм. Задан алфавит E и отношение независимости I на E. Задано некоторое слово w, состоящее из букв алфавита E. Будем строить некоторое разложение этого слова с помощью индукции по длине слова. Будем называть это разложение нормальной q-формой. При $n = 1$ мы получим один блок разложения, состоящий из одной буквы. Предположим, что мы построили разложение $[w] = [w_1] \cdots [w_n]$ в композицию блоков, каждый из которых состоит из попарно независимых букв, причем $|w_i| \leq q$ для всех $1 \leq i \leq n$. Для произвольного $a \in E$ построим разложение слова wa. Его длина будет равна $n + 1$. Возможны следующие случаи:

1. Длина слова w_n равна q, или буква a зависима с некоторой буквой из w_n. В этом случае нормальная q-форма слова wa равна $[w_1] \cdots [w_n][a]$.
2. В противном случае рассматриваем слова $w_{n-1}, w_{n-2}, \dots$. Если существует наибольшее k, такое, что длина слова w_k равна q, или буква a зависима с некоторой буквой из w_k, то нормальная q-форма будет равна $[w_1] \cdots [w_k][w_{k+1}a][w_{k+2}] \cdots [w_n]$.
3. Если для всех $1 \leq i \leq n$ длина слова w_i строго меньше, чем q , и если буква a независима со всеми буквами каждого из слов w_i, то нормальная q-форма равна $[w_1 a][w_2] \cdots [w_n]$.

Число блоков нормальной q-формы слова w будем называтья его q-высотой и обозначать $h_q(w)$.

Предложение 2. Если время выполнения микрооперации принять за единицу, то q-высота слова w будет равна минимальному времени работы параллельного процесса под управлением q процессоров.

Доказательство предложения 2 вытекает из предшествующих рассуждений.

На основе описанного выше алгоритма нами разработано программное обеспечение для нахождения зависимости времени работы ограниченного конвейера от числа процессоров q. На рис.3 показан интерфейс разработанного программного обеспечения.

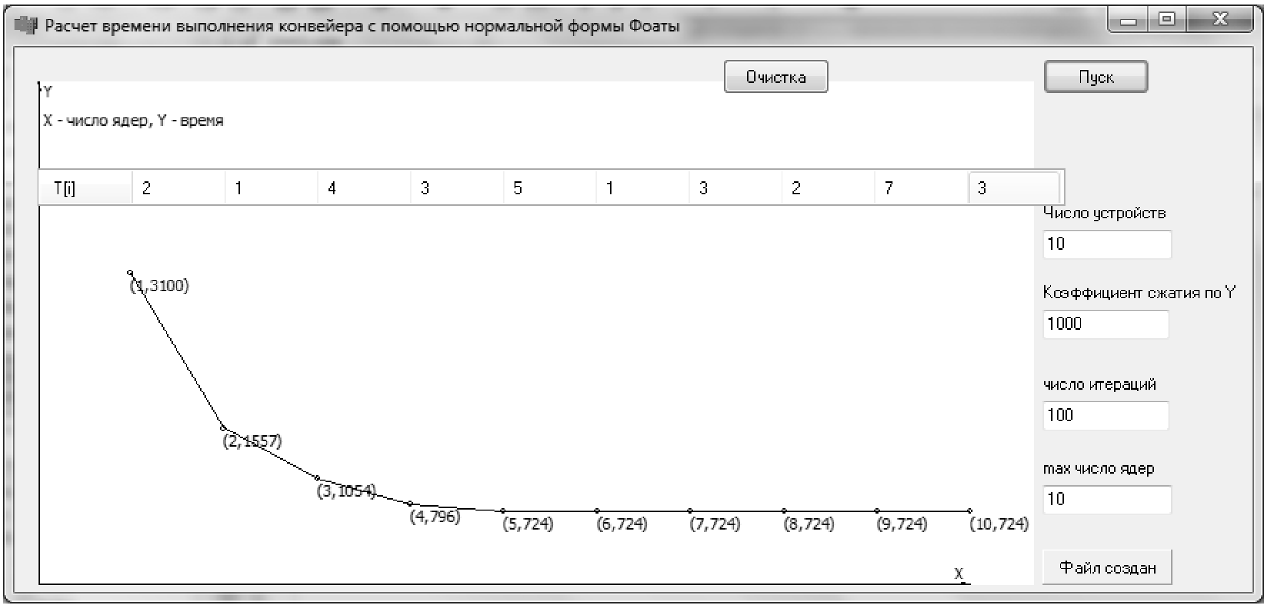

Рис.3. График зависимости времени от числа процессоров

Для каждого перехода вводится суммарная задержка времени $\tau_i = r(p_{i-1}) + \tau(t_i) + w(p_i)$. Программа выводит график зависимости времени обработки ста элементов от числа процессоров q. Крупные точки на графике показывают результаты измерения с помощью q-высоты трассы конвейера. Отрезками показан график зависимости, полученный с помощью сформулированной выше гипотезы.

В заключение отметим, что результаты экспериментов, полученных с помощью многопоточного приложения и результаты экспериментов с программой, основанных на вычислении q-высоты трассы конвейера, совпали со значениями формулы, приведенной выше как гипотеза. Отсюда вытекает, что эта формула может использоваться для расчета производительности вычислительного конвейера. В результате исследований был получен неожиданный для нас результат: минимальное число процессоров, при котором достигается наибольшая производительность, равно наименьшему целому q, удовлетворяющему неравенству $q \geq \frac{\sigma}{\mu}$.

Список использованных источников

1 Hartstein A., Puzak T. R. Optimum power/performance pipeline depth // Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture. – IEEE Computer Society, 2003. P. 117.

2 Yao J. et al. Optimal pipeline depth with pipeline stage unification adoption // ACM SIGARCH Computer Architecture News. V. 35, №. 5. 2007. P. 3-9.

3 Кудряшова, Е. С. Моделирование конвейерных и волновых вычислений / Е. С. Кудряшова, Н. Н. Михайлова, А. А. Хусаинов // Интернет-журнал «Науковедение», 2014 №1 (20) [Электронный ресурс] - М.: Науковедение, 2014. Режим доступа: http://naukovedenie.ru/PDF/56TVN114.pdf

4 Коуги П.М. Архитектура конвейерных ЭВМ: Пер. с англ. – М.: Радио и связь, 1985. – 360 с.

5 Кудряшова, Е. С. Временные оценки и гомоморфизмы асинхронных систем / Е. С. Кудряшова, А. А. Хусаинов // Наука и образование: Электронное научно-техническое издание. №1, 2014. С. 134-149.

6 Diekert, V. Combinatorics on Traces / V. Diekert // Lecture Notes in Computer Science. – Berlin : Springer-Verlag. – 1990. – V. 454. – 169 p.

7 Diekert, V. Partial Commutation and Traces / V. Diekert, Y. Metivier // Handbook of formal languages. – New York : Springer-Verlag. – 1997. – V. 3. – P. 457-533.

8 Кудряшова Е.С. Время работы асинхронного линейного конвейера. XXXVIII Дальневосточная школа-семинар имени академика Е.В. Золотова, 1-5 сентября 2014. – Владивосток, ИАПУ ДВО РАН. С. 410-414.

Хусаинов А.А., Чернов А.М., Маевская Е.Д., Романченко А.А. Модели для расчета времени работы вычислительных конвейеров // Актуальные проблемы науки: Материалы XXIII Международной научно-практической конференции (11.01.2016). – М.: Издательство «Спутник+» (01.02.2016), 2016. с.83-91.