

УДК 004.9

Кудряшова Екатерина Сергеевна

ФГБОУ ВПО «Комсомольский-на-Амуре государственный технический университет»
Россия, Комсомольск-на-Амуре¹
Аспирант, ассистент
E-Mail: ekatt@inbox.ru

Михайлова Наталья Николаевна

ФГБОУ ВПО «Комсомольский-на-Амуре государственный технический университет»
Россия, Комсомольск-на-Амуре
Старший преподаватель
E-Mail: mnataly4217@yandex.ru

Хусаинов Ахмет Аксанович

ФГБОУ ВПО «Комсомольский-на-Амуре государственный технический университет»
Россия, Комсомольск-на-Амуре
Доктор физико-математических наук, профессор
E-Mail: husainov51@yandex.ru

Моделирование конвейерных и волновых вычислений

Аннотация: Предложена компьютерная модель многопроцессорного асинхронного вычислительного конвейера с буферной памятью. Она допускает программную реализацию с помощью многопоточного приложения на языке C++ под управлением операционных систем Windows XP, Windows 7. Эксперименты с компьютерной моделью были проведены на различных процессорах, включая Intel(R) Core(TM) i3-2310M CPU. Роль функциональных устройств играют потоки. Буферная память программно реализована с помощью объектов класса канал, имеющего операции записи и чтения, работающие по алгоритму Дейкстры для решения задачи о производителе и потребителе. Экспериментально установлено, что эта модель может быть применена для исследования производительности и историй параллельного процесса, состоящего из вычислительных операций и операций передачи данных между функциональными устройствами асинхронного конвейера. Приведены графики зависимости ускорения от объема входных данных, полученные с помощью эксперимента. Эти графики подтверждают формулу для расчета ускорения асинхронного конвейера. На основании исследования ускорения с помощью этой компьютерной модели сделаны выводы о том, что ускорение для асинхронного линейного конвейера не зависит от объема буферной памяти. Рассмотрено использование этой модели для изучения явления пузырька в конвейере. Эта модель расширена также для имитации работы волновой системы. Для волновой системы высказана и экспериментально проверена гипотеза о линейной зависимости времени обработки данных от их объема.

Ключевые слова: Вычислительный конвейер; асинхронный конвейер; буферная память; сеть Петри; расчет производительности; каналы ОС Unix; многопоточное приложение; семафор; ускорение параллельного выполнения; задача о производителе и потребителе; трасса; волновая система; объектно-ориентированное программирование.

Идентификационный номер статьи в журнале 56TVN114

¹ 681013 Россия, г. Комсомольск–на–Амуре, пр. Ленина, д. 27

Ekaterina Kudryashova

Komsomolsk-on-Amur State Technical University
Russia, Komsomolsk-on-Amur
E-Mail: ekatt@inbox.ru

Natalya Mikhailova

Komsomolsk-on-Amur State Technical University
Russia, Komsomolsk-on-Amur
E-Mail: mnataly4217@yandex.ru

Ahmet Husainov

Komsomolsk-on-Amur State Technical University
Russia, Komsomolsk-on-Amur
E-Mail: husainov51@yandex.ru

Modeling of pipeline and wave computations

Abstract: A computer model of asynchronous multiprocessor computing pipeline with a buffer storage is proposed. The model allows the software implementation using the multi-threaded application written in C++ running under operating systems Windows XP, Windows 7. The experiments with the computer model were performed on a variety of processors, including the Intel(R) Core(TM) i3-2310M CPU. Threads act as functional devices. The buffer storage is implemented using class channel objects. It has read and write operations, working on the Dijkstra algorithm for solving the producer-consumer problem. Experimentally established that this model can be used to study the performance and the histories of the parallel process, which consists of computing operations and data transfer operations between functional units of asynchronous pipeline. The plots of speedup depending on the number of input data, obtained by experiment, are shown. These plots confirm the formula for calculating the speedup for the asynchronous pipeline. Conclusions are obtained that a speedup for the asynchronous pipeline is independent on the volume of its buffer storage. This model is considered for studying the bubble effect in the pipeline. The model is also extended to simulate a wave system. For the wave system, it is suggested and experimentally tested a conjecture of a linear dependence of the data on their volume.

Keywords: Computational pipeline; asynchronous pipeline; buffer storage; Petri net; performance computation; OS Unix channel; multithreaded application; semaphore; parallel processing speedup; producer-consumer problem; trace; wave system; object-oriented programming.

Identification number of article 56TVN114

Введение

Вычислительная волновая система состоит из функциональных устройств и каналов для передачи данных между этими устройствами. Мы моделируем каналы с помощью каналов ОС Unix [1], а функциональные устройства – с помощью потоков (нитей). Вычислительная волновая система должна удовлетворять следующим условиям:

1. Каждый канал имеет единственное функциональное устройство, считывающее данные из этого канала или единственное устройство, записывающее данные в этот канал (или и то, и другое).
2. Для каждого функционального устройства существует, по крайней мере, один канал, в который это устройство записывает, и, по крайней мере, один канал, из которого оно считывает.
3. Всякая цепь, состоящая из соединений каналов со считывающими из них функциональными устройствами и соединений этих устройств с каналами, в которые они записывают, не имеет циклов.

Таким образом, волновая система является обобщением линейного конвейера в смысле [2] и частным случаем нелинейного конвейера [2]. Волновая система может быть реализована как сеть транспьютеров и составлять вычислительную систему с высокой интенсивностью вычислений и большими объемами входных и выходных данных. Она во многом похожа на систолическую матрицу. Но в волновой системе отсутствует глобальная система синхронизации. Волновая система является асинхронной.

Если канал не имеет устройства, записывающего в него данные, то он называется входным. Если он не имеет устройства, считывающего из него данные, он называется выходным. В начале работы волновой системы, для нахождения ее максимальной производительности, в каждый из ее входных каналов записывается n элементов данных.

Особенно широко применяются асинхронные линейные конвейеры, начиная с уровня обработки запросов в параллельных системах управления базами данных [3] и кончая уровнем вентиляционной реализации [4]. По этой причине мы сначала предлагаем компьютерную модель для исследования линейного конвейера. Мы решаем проблему, связанную с тем, что всякий конвейер является многопроцессорной системой. Мы предлагаем модель, реализуемую на компьютере, имеющем один процессор. Эта компьютерная модель может быть применена для оценки ускорения. Она может быть применена также для описания истории действий, происходящих в вычислительном конвейере.

Каждое функциональное устройство выполняет цикл, включающий чтение данных из входного канала, выполнение операции и запись результата операции в выходной канал. Функциональные устройства имеют различную скорость выполнения. Каналы могут иметь различные времена чтения и записи. Один канал не может работать одновременно с двумя функциональными устройствами. Если каждый канал состоит из одного регистра, а скорости выполнения операций одинаковы, мы получаем синхронный линейный конвейер [2].

Руководствуясь простыми соображениями можно получить формулу для нахождения ускорения вычислений S_p , полученного с помощью синхронного конвейера, состоящего из p функциональных устройств. Поскольку мы не обладаем ни аксиомами, ни правилами вывода для доказательства этой формулы для асинхронного конвейера, мы проверяем ее экспериментально.

Похожая формула рассматривается в [5], где время выполнения операций может быть различным. Но в [5] предполагается, что скорость обмена данными между каналами и функциональными устройствами одинакова, а чтение из канала и запись в этот канал может

происходить одновременно. Мы же возлагаем функции чтения и записи на функциональное устройство. Мы также предлагаем и экспериментально проверяем гипотезу, приводящую к некоторому обобщению этой формулы на волновые системы.

Формула для нахождения ускорения

Рассмотрим конвейер, состоящий из p функциональных устройств и $p + 1$ каналов c_0, \dots, c_p . Пусть u_i - последовательность действий, выполняемых за 1 шаг, $0 \leq i \leq p - 1$. Она состоит из чтения из канала c_i , вычислительной операции a_i и записи в канал c_{i+1} . Время выполнения u_i равно $r(c_i) + \tau(a_i) + w(c_{i+1})$, где $r(c_i)$ обозначает время чтения из канала c_i , $w(c_i)$ - время записи в канал c_i , $\tau(a_i)$ - время выполнения операции a_i , $0 \leq i \leq p - 1$.

Под ускорением $S_p(n)$ мы будем подразумевать отношение $\frac{T_{seq}}{T_p}$ времени T_{seq} последовательного выполнения вычислительных операций над n элементами входных данных к времени T_p параллельного выполнения этих операций с помощью p процессоров. Ускорение можно определить также как отношение $\frac{T_1}{T_p}$.

Одной из наших задач будет экспериментальное подтверждение следующей формулы для расчета ускорения для асинхронного вычислительного конвейера с буферной памятью:

$$S_p(n) = \frac{n \sum_{i=1}^{p-1} \tau(a_i)}{\sum_{i=1}^{p-1} (r(c_i) + \tau(a_i) + w(c_{i+1})) + (n-1) \max_{0 \leq i \leq p-1} \{r(c_i) + \tau(a_i) + w(c_{i+1})\}}. \quad (1)$$

Мы не будем использовать эту формулу для вывода каких-либо утверждений. Ее пока можно считать гипотетической.

Математическая модель

Пусть $\mathbb{N} = \{0, 1, 2, \dots\}$ - множество неотрицательных целых чисел. Для произвольного конечного множества P обозначим через \mathbb{N}^P множество функций $M: P \rightarrow \mathbb{N}$. Сеть Петри называется четверка $(P, T, pre, post, M_0)$, состоящая из произвольных конечных множеств $P = \{p_0, \dots, p_{n-1}\}$ и $T = \{t_0, \dots, t_{m-1}\}$, функций $pre, post: T \rightarrow \mathbb{N}^P$ и функции $M_0: P \rightarrow \mathbb{N}$. Элементы из P называются местами, а из T - переходами. Маркировкой называется функция $M: P \rightarrow \mathbb{N}$. Функция M_0 называется начальной маркировкой.

Сеть Петри наглядно изображается с помощью кругов, прямоугольников и стрелок. Места изображаются в виде кругов, переходам соответствуют прямоугольники. Для всех $t_i \in T$ значения функций $pre, post: T \rightarrow \mathbb{N}^P$ сами будут функциями $pre(t_i): P \rightarrow \mathbb{N}$, $post(t_i): P \rightarrow \mathbb{N}$. Если $pre(t_i)(p_j) = k > 0$, то рисуется k стрелок с началом в круге p_j и с концом в прямоугольнике t_i . Если $post(t_i)(p_j) = k > 0$, то проводится k стрелок от перехода t_i к месту p_j . Кроме того, для указания значений маркировки, рисуются точки в кругах. Если $M(p_i) = k > 0$, то в месте p_i рисуется k фишек.

Состояния сети Петри определены маркировками, которые получаются из начальной с помощью последовательности срабатывания переходов. Переход t_i называется допускающим срабатывание, если для текущей маркировки M для всех $p_j \in P$ имеют место неравенства $pre(t_i)(p_j) \geq M(p_j)$. В этом случае говорят, что маркировка M' получена в результате срабатывания перехода, если для всех $p_j \in P$ верно $M'(p_j) = M(p_j) - pre(t_i)(p_j) + post(t_i)(p_j)$.

Вычислительный конвейер, во входной канал которого записано n входных элементов, можно представить с помощью математической модели в виде сети Петри. На рисунке 1 показана сеть Петри конвейера, состоящего из четырех функциональных устройств и пяти каналов. Входные данные состоят из трех элементов.

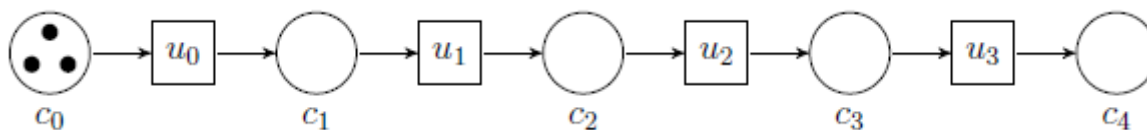


Рис. 1. Конвейер из четырех функциональных устройств

Для каждого функционального устройства u_i , $0 \leq i \leq p - 1$, задано время выполнения. Функциональное устройство выполняет последовательно чтение из канала c_i , вычисление операции a_i и запись в канал c_{i+1} , $0 \leq i \leq p - 1$. Время выполнения обозначим через $\tau(u_i)$. Имеет место равенство $\tau(u_i) = r(c_i) + \tau(a_i) + w(c_{i+1})$. Для сетей Петри вводится отношение независимости, при котором два перехода независимы, если они не имеют общих входных и выходных мест. В сети Петри на рисунке 1 переходы u_0 и u_1 зависимы, поскольку во время записи устройство u_0 должно захватить канал, из которого читает устройство u_1 . Для того, чтобы учесть это правило, мы измельчаем переходы сети Петри на переходы, время работы которых равно 1. Для того чтобы переходы, полученные после измельчения перехода u_i , выполнялись как последовательная подпрограмма, мы добавим сверху место с фишкой, соединенное с первым и последним переходами. В результате для всех $0 \leq i \leq p - 1$ переход u_i будет заменен частью сети Петри, показанной на рисунке 2. Стрелка, входившая раньше в переход u_i , будет теперь входить в $a_{i,0}$, а выходящая из u_i будет выходить из $a_{i,\tau(u_i)-1}$.

Передача данных через каналы будет производиться с помощью алгоритма Дейкстры для решения задачи о производителе и потребителе [6]. В этом алгоритме очередь канала при чтении и записи данных захватывается с помощью бинарного семафора. Отсюда следует, что запись в канал и чтение из него являются зависимыми действиями, в том смысле, что они не могут происходить одновременно.

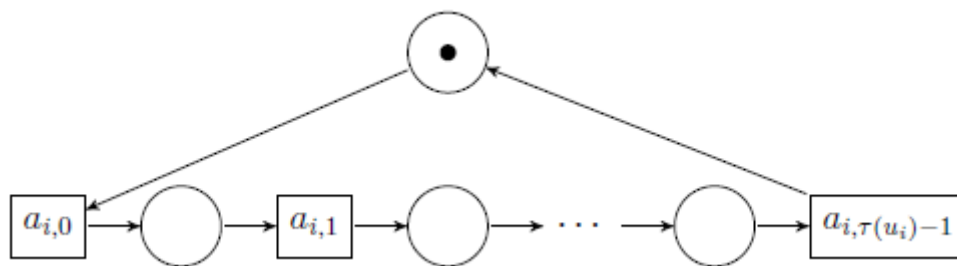


Рис. 2. Измельчение i -ого перехода

Напомним определение трассы, введенное Мазуркевичем [7]. С этой целью рассмотрим параллельный вычислительный процесс, состоящий из операций, принадлежащих некоторому множеству команд E . Если операции могут выполняться одновременно, то будем называть их независимыми. В противном случае они зависимы. Историей [1] называется конечная последовательность операций. Для любого процесса, заданного с помощью конечной последовательности выполняемых операций мы можем переставлять выполнение рядом стоящих независимых операций – результат выполнения процесса не изменится. Поэтому можно отождествлять истории, полученные друг из друга с помощью таких перестановок. Классы историй, полученных с помощью описанного отождествления, называются трассами [7].

Обработка n элементов данных конвейером $u_0 \dots u_{p-1}$ описывается трассой, состоящей из операций $a_{i,j}$, $0 \leq i \leq p-1$, $0 \leq j \leq \tau(u_i) - 1$, со временем выполнения 1:

$$\left(a_{0,0} a_{0,1} \dots a_{0,\tau(u_0)-1} a_{p-1,1} a_{p-1,2} \dots a_{p-1,\tau(u_{p-1})-1} \right)^n,$$

в которой для каждого i попарно зависимы операции $a_{i,0}, a_{i,1}, \dots, a_{i,\tau(u_i)-1}$, принадлежащие u_i , а также для каждого i из диапазона $0 \leq i \leq p-2$ зависимы операции $a_{i,\tau(u_i)-1}$ и $a_{i+1,0}$.

Полученная трасса является математической моделью для исследования всех возможных историй асинхронного конвейера.

Всякой сети Петри соответствует асинхронная система (см. [8]). Ее состояниями служат маркировки. Операции определены переходами, действующими на маркировках с помощью срабатываний. Значит, можно воспользоваться результатами работы [8], в которой было показано, что минимальное время обработки n элементов данных будет равно высоте нормальной формы Фойата этой трассы. Приведение трассы к нормальной форме Фойата аналогично построению канонической ярусно-параллельной нормальной формы в терминологии [9].

В этой математической модели не участвует размер буферной памяти. Она строится из предположения, что каналы имеют достаточно большую буферную память. Одна из наших задач – исследование влияния размера буферной памяти на ускорение. Мы не можем решить эту задачу с помощью математической модели. Поэтому мы разрабатываем компьютерную модель и решаем нашу задачу экспериментально.

Описание компьютерной модели

При построении компьютерной модели мы ориентируемся на сеть Петри, показанную ниже. Местам будут соответствовать каналы $c[0], c[1], \dots, c[p]$.

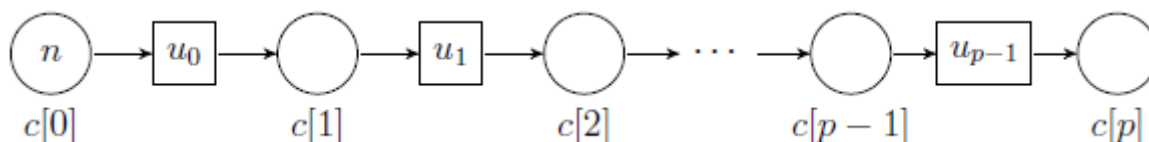


Рис. 3. Конвейер из p устройств

Компьютерная модель будет состоять из p потоков (нитей). Они взаимодействуют с помощью операторов чтения из канала и записи в канал. Перед началом работы n элементов входных данных записывается в канал $c[0]$. Затем загружаются потоки $0 \leq i \leq p-1$:

```

Thr(i)
{
int x;
for (int j=0; j<n; j++)
{
c[i]>>x; Sleep(tau[i]); c[i+1]<<x;
}
}
    
```

Здесь $\tau[i]$ равно $\tau(u_i)$. Пусть $r[i]$ - число, равное времени чтения $r(c_i)$ из канала c_i , а $w[i+1]$ - время записи $w(c_{i+1})$ в канал c_{i+1} , $0 \leq i \leq p - 1$.

Эти потоки будут работать параллельно. Синхронизация достигается с помощью класса канала.

Наша компьютерная модель имитирует многопроцессорную систему. Это становится возможным благодаря использованию подпрограммы `Sleep(t)`, усыпляющей вызвавший ее поток на время t в миллисекундах. Мы проводили отладку программного обеспечения и испытания, используя процессор Intel(R) Core (TM) i3-2310M CPU. Заметим, что программная реализация возможна и на однопроцессорной системе. Но единица времени, с помощью которой мы считаем ускорение, в некоторых случаях должна увеличиваться. Например, для процессора Intel(R) ATOM(TM) CPU N270 эта единица времени должна быть равна 1 секунде, в этом случае аргумент для подпрограммы `Sleep()` надо умножить на 1000.

Класс канала состоит из циклической очереди, определенной как массив целых чисел. Одна из программных реализаций этого класса и примеры использования приведены в учебном пособии [10].

Выполнение оператора $c[i] \gg x$ начинается с ожидания события «канал не пуст». Затем поток захватывает канал $c[i]$, извлекает из него элемент и освобождает канал. Перед освобождением мы добавили оператор `Sleep(r[i])`.

Запись $c[i+1] \ll x$ состоит из ожидания «свободного места» в очереди канала $c[i+1]$. Затем поток захватывает канал $c[i+1]$, записывает в него элемент, выполняет задержку `Sleep(w[i+1])` и освобождает канал.

На экран выводятся графики ускорений, изображенные на рисунке 4. Графики, состоящие из отрезков, получены с помощью формулы (1). Графики ускорений, полученных опытным путем, выводятся точками. Полученные графики почти не отличаются. Это, во-первых, дает подтверждение формулы (1) опытным путем. Во-вторых, мы получаем адекватную модель асинхронного конвейера с буферной памятью.

Гипотетическая формула для нахождения ускорения получена в предположении, что размер буферной памяти достаточно большой. Исследования графиков ускорения при различных размерах буферной памяти показывают, что все эти графики практически совпадают с гипотетическим. Следовательно, производительность асинхронного конвейера почти не зависит от размера буфера. На практике, скорость обмена с одним регистром выше, чем с массивом регистров. Отсюда можно сделать предположение, что асинхронные конвейеры, у которых каналы состоят из одного регистра, являются оптимальными, как для производства, так и для использования.

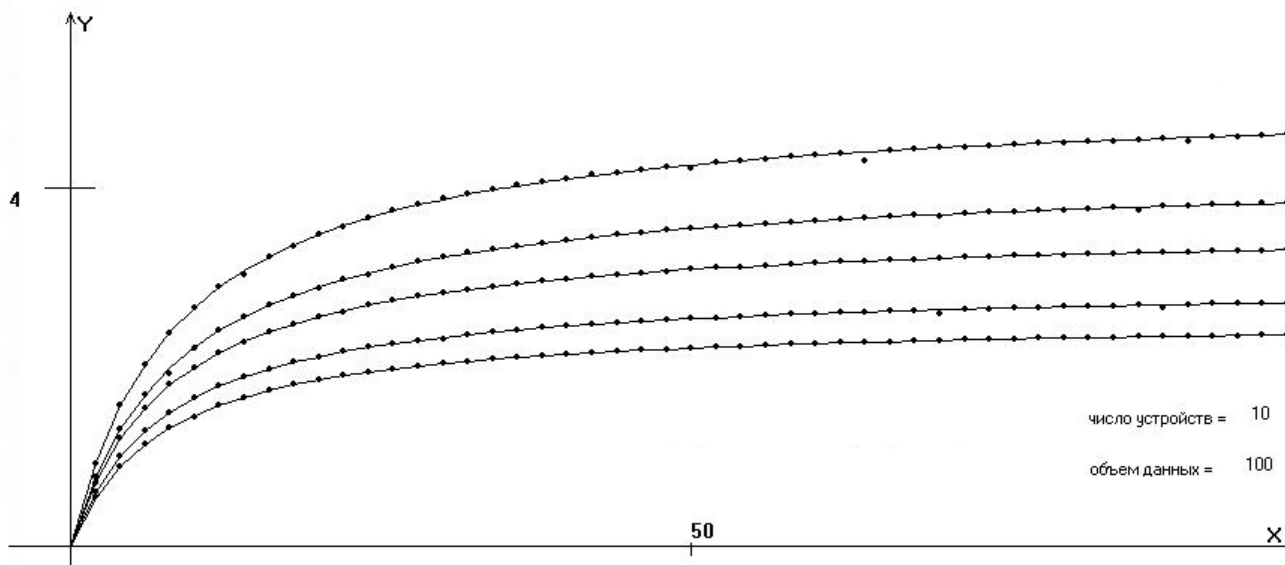


Рис. 4. Графики зависимости ускорений от объема входных данных

Компьютерную модель можно применять также для исследования других свойств конвейера. Например, для явления пузыря, которое имеет место в случае кратковременного замедления действия одного из устройств конвейера. В этом случае на графике ускорения появится скачок:

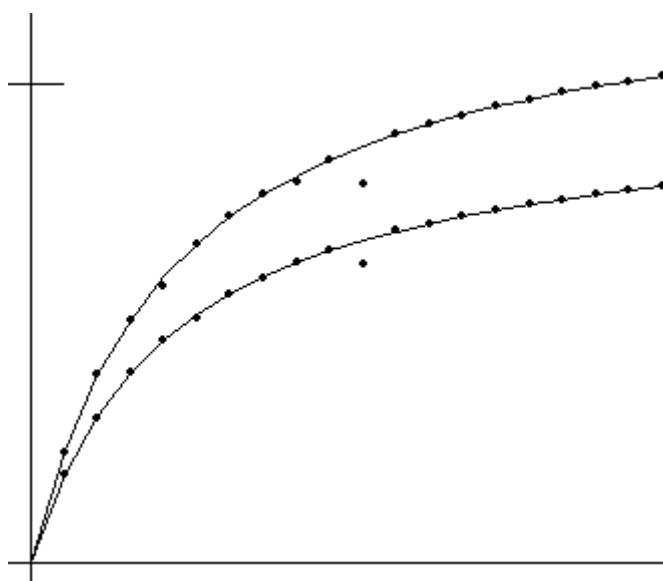


Рис. 5. Явление пузыря

Исследование волновых систем

Переход t сети Петри называется входным для места p , если $post(t)(p) \geq 1$. В этом случае проводится стрелка от t к p . Он называется выходным для p , если $pre(t)(p) \geq 1$. В этом случае проводится стрелка от p к t . Сеть Петри называется ациклической, если в ней нет направленных циклов, состоящих из стрелок $p_0 \rightarrow t_0 \rightarrow p_1 \rightarrow t_1 \rightarrow \dots \rightarrow p_k = p_0$.

Рассмотрим волновые системы [10]. Волновую систему можно рассматривать как параллельный процесс, заданный с помощью ациклической сети Петри, каждое место которой имеет единственный входной переход или единственный выходной переход. Возможно, оно имеет и входной, и выходной переходы. Начальная маркировка для каждого места, не

имеющего входной переход, равна некоторому числу $n > 0$. Переходам соответствуют функциональные устройства, взаимодействующие через буферную память, находящуюся в местах. Термин “волновая” был введен в [11]. Мы рассмотрим в качестве примера волновую систему, имеющую следующую сеть Петри:

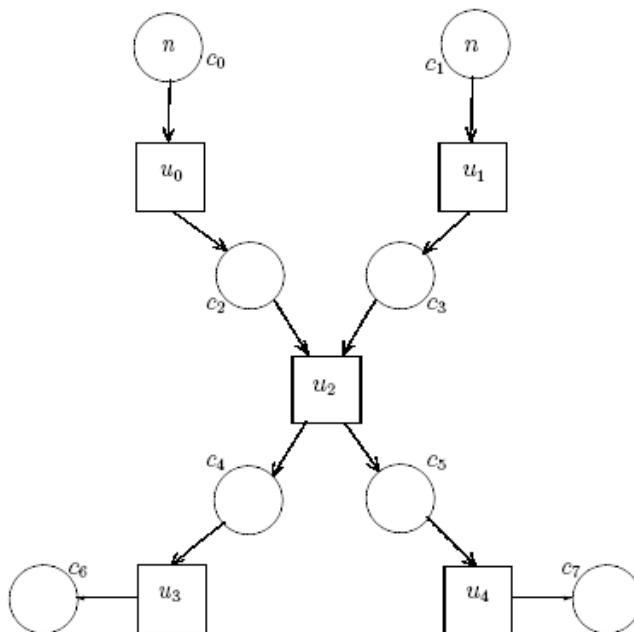


Рис. 6. Сеть Петри волновой системы

Входные данные размещаются в местах, имеющих один выходной переход. На рисунке 6 входные данные размещаются в каналах c_0 и c_1 . Результаты вычислений записываются в каналы, соответствующие местам, имеющим по одному входному переходу. Для данного примера, этими каналами будут c_6 и c_7 .

Компьютерная модель этой волновой системы состоит из потоков, соответствующих переходам ее сети Петри и каналов, соответствующим ее местам. Программная реализация использует описанный выше класс канала. Всякий поток, соответствующий переходу, состоит из цикла. В теле цикла сначала принимаются данные из входных каналов. Затем имитируется выполнение операции с помощью подпрограммы `Sleep()`. После этого передаются данные в выходные каналы. Для волновой системы, показанной на рисунке 6, были получены следующие графики зависимости времени $T_5(n)$ обработки n пар элементов входных данных, где n изменяется от 1 до 99.

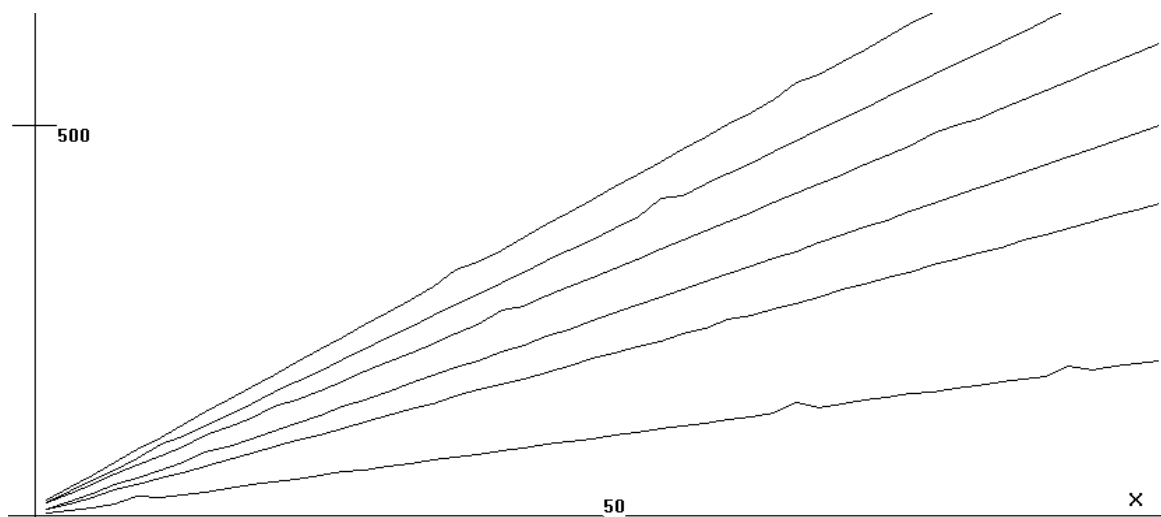


Рис. 6. Графики времени обработки для волновой системы

Время выполнения операций, которые задавались для функциональных устройств, приведены в таблице, первой строке соответствует самый нижний график, второй – следующий и т.д.

Времена обработки функциональных устройств

$\tau(u_0)$	$\tau(u_1)$	$\tau(u_2)$	$\tau(u_3)$	$\tau(u_4)$
2	2	1	0	1
2	2	3	4	3
2	2	3	5	3
6	5	6	5	4
6	5	7	5	4
6	5	7	8	4

Были проведены эксперименты с другими волновыми системами. Во всех случаях графики хорошо приближаются прямым. На основании этого мы выдвигаем следующую гипотезу:

Для всякой волновой системы, состоящей из p функциональных устройств, существуют константы A и B , такие, что время обработки данных объема n , которые передаются через входные каналы волновой системы, можно вычислять по формуле $T_p(n) = A + B(n - 1)$.

В этом случае константа A будет равна $T_p(1)$, а $B = T_p(2) - T_p(1)$. Очень возможно, что $B = \max_{0 \leq i \leq p-1} \tau(u_i)$. Доказательство этой гипотезы значительно облегчит расчет ускорения и производительности волновой системы.

Заключение

Результаты экспериментов позволяют сделать вывод о том, что построенная нами компьютерная модель асинхронного конвейера с буферной памятью подходит для исследования поведения этого конвейера. Исследования привели к выводу, что объем буферной памяти не влияет на ускорение. Мы нашли также возможность моделирования явления пузырьков. Компьютерная модель была также обобщена нами для изучения волновых систем. Мы высказали и экспериментально подтвердили гипотезу о том, что для волновой системы график зависимости времени обработки данных от числа входных элементов описывается с помощью прямой линии.

Данная работа была выполнена в рамках программы стратегического развития государственных образовательных учреждений высшего профессионального образования, грант № 2011-ПР-054.

ЛИТЕРАТУРА

1. Andrews, G. R. Foundations of multithreaded, parallel, and distributed programming. Wesley, University of Arizona, USA, 2000.
2. Цилькер, Б. Я. Организация ЭВМ и систем: учебник для вузов / Б. Я. Цилькер, С. А. Орлов. - СПб.: Питер, 2011. – 668 с.
3. Соколинский, Л. Б. Методы организации параллельных систем баз данных на вычислительных системах с массовым параллелизмом: дис. ... доктора физ.-мат. наук. - Челябинск, 2003. – 247 с.
4. Герасимов, А. А. Способы формального описания асинхронных схем / А. А. Герасимов, П. В. Кустарев // Сборник трудов молодых ученых и сотрудников кафедры ВТ / Под ред. д.т.н., проф. Т. И. Алиева. - СПб: СПбГУ ИТМО, 2010. С. 31-35.
5. Герценбергер, К. В. Аналитическая модель оценки производительности многопроцессорной обработки данных для набора параллельных алгоритмических структур / К. В. Герценбергер, Е. В. Чепин // Бизнес-информатика. 2011. Т.18. №4. С. 24-30.
6. Dijkstra, E. W. Co-operating Sequential Processes. F// Programming Languages. - New York: Academic Press, 1968. P. 65-138.
7. Mazurkiewicz, A. Basic notions of trace theory // Linear time, branching time and partial order in logics and models for concurrency, Lecture Notes in Computer Science. Vol. 354. - Berlin: Springer-Verlag, 1989. P. 285-363.
8. Хусаинов, А. А. Модель для временных оценок параллельных вычислительных систем / А. А. Хусаинов, Е. С. Кудряшова // Информационные технологии XXI века: материалы международной научной конференции (Хабаровск, 20-24 мая 2013 г.). – Хабаровск: изд-во Тихоокеан. гос. ун-та, 2013. С. 203-207.
9. Воеводин, В. В. Параллельные вычисления / В. В. Воеводин, Вл. В. Воеводин. - СПб: БХВ-Петербург, 2002. – 608 с.
10. Хусаинов, А. А. Архитектура вычислительных систем: учеб. пособие / А. А. Хусаинов, Н. Н. Михайлова. – Комсомольск-на-Амуре: изд-во Комсомольского-на-Амуре гос. техн. ун-та, 2006. – 123 с.
11. Водяхо, А. И. Высокопроизводительные системы обработки данных: учеб. пособие для вузов / А. И. Водяхо, Н. Н. Горнец, Д. В. Пузанков. – М.: Высш. шк., 1997. – 304 с.

Рецензент: Амосов Олег Семенович, доктор технических наук, профессор, зав. кафедрой промышленной электроники ФГБОУ ВПО «КНАГТУ».

REFERENCES

12. Andrews, G. R. Foundations of multithreaded, parallel, and distributed programming. - Wesley, University of Arizona, USA, 2000.
13. 2. Cil'ker, B. Ja. Organizacija JeVM i sistem: uchebnik dlja vuzov / B. Ja. Cil'ker, S. A. Orlov. - SPb.: Piter, 2011. – 668 s.
14. 3. Sokolinskij, L. B. Metody organizacii parallel'nyh sistem baz dannyh na vychislitel'nyh sistemah s massovym parallelizmom: dis. ... doktora fiz.-mat. nauk. - Cheljabinsk, 2003. – 247 s.
15. 4. Gerasimov, A. A. Sposoby formal'nogo opisaniya asinhronnyh shem / A. A. Gerasimov, P. V. Kustarev // Sbornik trudov molodyh uchenykh i sotrudnikov kafedry VT / Pod red. d.t.n., prof. T. I. Alieva. - SPb: SPbGU ITMO, 2010. S. 31-35.
16. 5. Gercenberger, K. V. Analiticheskaja model' ocenki proizvoditel'nosti mnogoprocessornoj obrabotki dannyh dlja nabora parallel'nyh algoritmicheskikh struktur / K. V. Gercenberger, E. V. Chepin // Biznes-informatika. 2011. T.18. №4. S. 24-30.
17. 6. Dijkstra, E. W. Co-operating Sequential Processes. F // Programming Languages. - New York: Academic Press, 1968. P. 65-138.
18. 7. Mazurkiewicz, A. Basic notions of trace theory // Linear time, branching time and partial order in logics and models for concurrency, Lecture Notes in Computer Science, Vol. 354. - Berlin: Springer-Verlag, 1989. P. 285-363.
19. 8. Husainov, A. A. Model' dlja vremennyh ocenok parallel'nyh vychislitel'nyh sistem / A. A. Husainov, E. S. Kudrjashova // Informacionnye tehnologii XXI veka: materialy mezhdunarodnoj nauchnoj konferencii (Habarovsk, 20-24 maja 2013 g.). - Habarovsk: izd-vo Tihookean. gos. un-ta, 2013. S. 203-207.
20. 9. Voevodin, V. V. Parallel'nye vychislenija / V. V. Voevodin, Vl. V. Voevodin. - SPb: BHV-Peterburg, 2002. – 608 s.
21. 10. Husainov, A. A. Arhitektura vychislitel'nyh sistem: ucheb. posobie / A. A. Husainov, N. N. Mihajlova. - Komsomol'sk-na-Amure: izd-vo Komsomol'skogo-na-Amure gos. tehn. un-ta, 2006. – 123 s.
22. 11. Vodjaho, A. I. Vysokoproizvoditel'nye sistemy obrabotki dannyh: ucheb. posobie dlja vuzov. / A. I. Vodjaho, N. N. Gornec, D. V. Puzankov. - M.: Vyssh. shk., 1997. – 304 s.