ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ С ПРОГРАММИРУЕМОЙ СТРУКТУРОЙ
(Вычислительные системы)
1982 год Выпуск 94

УДК 519.68:519.71:681.142.2

МОДЕЛЬ КОЛЛЕКТИВА ВЫЧИСЛИТЕЛЕЙ, ОСНОВАННАЯ НА ПРИНЦИПЕ БЛИЗКОДЕЙСТВИЯ

В.А. Воробьев

В качестве математической модели коллектива вычислителей предлагается гомогенная система асинхронных марковских автоматов. Даны способы описания однородной структуры и асинхронного марковского автомата, условия ординарности системы,позволяющие рассматривать функционирование системы как марковский процесс. Приведен пример неординарной системы. Обсуждаются основные направления дальнейших исследований.

I. Цель работы

В работе [I] рассмотрены коллективы вычислителей, основанные на принципах параллельности вычислений, программируемости структуры и конструктивной однородности. Вычислительные средства, отвечающие этим принципам, называют однородными вычислительными системами или вычислительными системами с программируемой структурой.

Настоящая работа выполнена в рамках исследования упомянутого класса вычислительных систем.

Представленная в [I] идея коллективного вычислителя является весьма общей. Конструкторы однородных вычислительных систем имеют более конкретные представления о предмете своих разработок. Если эти представления сформулированы в терминах вычислительной техники (регистр, устройство, схема, канал), то можно говорить о технической модели коллектива вычислителей, если же они достаточно абстрактны и формализованы, то имеет смысл говорить о математической модели.

Кроме того, всякая модель вычислителя связана с соответствующей моделью вычислений. Модель коллектива вычислителей связана с соответствующим представлением о параллельных вычислениях. Поскольку многие аспекты поведения вычислителя реализуются программно и, наоборот, любая программа может быть реализована аппаратурно, с точки зрения внешнего наблюдателя эти две модели трудно различимы.

Предметом настоящей работы является обсуждение существующих технических и математических моделей коллективов вычислителей и соответствующих моделей вычислений. Цель работы - обоснование и построение математической модели однородной вычислительной системы, удовлетворяющей условиям локальности, асинхронности, распре - деленности и децентрализованности. Совокупность этих условий определяет принцип близкодействия в применении к вычислительной технике.

2. Некоторые модели коллективов вычислителей

Исследования коллективного поведения восходят еще к Дж.фон Нейману, предложившему клеточные автоматы. В дальнейшем это направление получило значительное развитие (см., например, [2-5]), но мало повлияло на вычислительную технику. Дело в том, что вычислительная машина должна иметь достаточную сложность, чтобы ин - терпретировать программу, записанную в ее память. Теория автома - тов не располагает ни языком для компактного описания такого объекта, ни средствами для его анализа.

Первая модель [6] коллектива вычислителей является техниче - ской и базируется на инженерном опыте развития вычислительных машин первого и второго поколений. Согласно [6] однородная вычислительная система есть совокупность электронных вычислительных машин, объединенная регулярной и программно настраиваемой сетью связи. Элемент системы называется элементарной машиной (ЭМ).

Системные операции (или команды) позволяют настроить систему связи и реализовать необходимые взаимодействия между ЭМ. Полный набор таких операций включает операции настройки, обмена (передачи и приема), обобщенного условного и безусловного переходов.

В [6] развита также соответствующая модель параллельных вычислений. Согласно этой модели параллельное вычисление есть набор последовательных вычислений над различными частями данных одной задачи. Поскольку все элементы данных должны так или иначе провзаимодействовать между собой в процессе вычислений, последние время от времени приостанавливаются и начинается массовый обмен данными. При этом все выделенные для этого обмена ЭМ должны вступить во взаимодействие о д н о в р е м е н н о. Слово "одновременно" выделено здесь потому, что отражает традиционное понимание обсуждаемой модели параллельных вычислений. Такое понимание исключительно удобно для проблемного программиста,однако в модели коллектива вычислителей нет формального аппарата,позволяющего учесть реальную неодновременность событий, происходящих в различных ЭМ.

В последние годы предпринимаются попытки формализовать техническую модель коллектива вычислителей и учесть при этом реальную асинхронность системы [7], однако требование одновременности взаимодействий противоречит основной идее, ради которой создавалась эта модель, - создание вычислительных систем н е о г р а н и ч е н н о й производительности, т. е. таких систем, производительность которых всегда можно увеличивать,добавив еще один вычислитель. Анализ этого противоречия будет дан ниже.

Другим важным недостатком технической модели коллектива вычислителей является отсутствие средств анализа случайностей в поведении системы. Между тем, при числе ЭМ порядка 10^3- 10^4 сбои и отказы будут происходить настолько часто, что ими уже нельзя пренебрегать. Компенсацией этого недостатка явились работы обобщенные в [1], рассматривавшие коллектив вычислителей как статистический ансамбль.Каждому элементу системы ставится в соответствие марковский процесс, протекающий независимо от остальных. Поведение конечной системы в этом случае также описывается обобщенным марковским процессом. Подробное обсуждение этого подхода содержится в [8].

Для решения задач, связанных с разработкой архитектуры вычислительной системы, построением и анализом операционных систем,гораздо более привлекательна а л г о р и т м и ч е с к а я модель коллектива вычислителей. В основу этой модели положен тот факт, что процесс функционирования любого устройства есть реализация некоторого алгоритма функционирования [9], причем этот алгоритм не зависит от того, как устройство его реализует: построена ли соответствующим образом схема устройства или это устройство выполняет соответствующую программу. Более того, один и тот же алгоритм может быть реализован различными схемами или программами.

Система совместно функционирующих устройств может быть описана системой алгоритмов функционирования [10]. С этой целью язык задания алгоритмов должен содержать средства описания взаимодействий процессов [11].

Нетрудно видеть, что система алгоритмов не зависит от числа реализующих ее устройств. В частности она может быть реализована одним вычислителем в режиме мультипрограммирования. Такая реализация называется квази- или псевдопараллельной и применяется при имитационном моделировании вычислительных систем [10].

Алгоритмическая модель коллектива вычислителей оказалась наиболее продуктивной в течение последних лет. В круг интересов и идей разработчиков попали многочисленные [11] результаты теории асинхронных параллельных вычислений и операционных систем. Эти результаты существенным образом использованы, например, при разработке ядра операционной системы СУММА [12]. В [8,12,13] представлены результаты имитационного моделирования и аналитические исследования алгоритмических моделей коллективов вычислителей. Существенным моментом этого цикла работ является строгое определение понятия однородной системы связей между элементарными вычислителями.

Однородность понимается в том смысле, что в группе автоморфизмов графа связей существует транзитивная подгруппа Aut, сохраняющая отметки ребер. Таким образом, алгоритмической моделью коллектива вычислителей является система

$$A \rightleftharpoons \langle A, E, S, \beta, Aut \rangle, \qquad (1)$$

где A - множество экземпляров алгоритма функционирования элементарной машины, E - множество пар, взаимодействующих алгоритмов и соответственно ребер графа связей, S - множество отметок ребер графа, β - отображение S на E, Aut - уже упомянутая транзитивная подгруппа.

Недостатком алгоритмической модели является тот факт, что для разностороннего анализа поведения коллектива вычислителей она располагает в основном трудоемким методом имитационного моделирования.

В заключение обзора моделей коллектива вычислителей следует упомянуть язык параллельных подстановок [14]. Параллельные подстановки являются обобщением нормальных алгоритмов Маркова, пригодным для описания клеточных автоматов. В [14] рассмотрена синхронная интерпретация параллельных подстановок, в [15] - асинхронная. Результаты этих работ будут в полной мере использованы ниже при построении математической модели коллектива вычислителей.

Рассмотрение имеющихся моделей коллектива вычислителей приводит к необходимости следующих проработок.

1. Подробный анализ причин ограничения производительности систем при увеличении числа элементов. Цель этого анализа - элиминация представлений и ограничений, заведомо приводящих к снижению быстродействия коллектива вычислителей.

2. Разработка математической модели коллектива вычислителей, предназначенной для анализа корректности проектируемых систем и получения характеристик эффективности последних. Эта модель должна быть достаточно простой, а следовательно отражать только наиболее существенные свойства коллектива - взаимодействия между элементарными вычислителями.

3. Принцип близкодействия

В данном разделе дается анализ производительности коллектива вычислителей и указываются условия ее неограниченной наращиваемости. Эти условия в совокупности дают формулировку принципа близкодействия, положенного в основу предлагаемой в данной работе модели.

Увеличение производительности, достигаемое при коллективных вычислениях, происходит за счет распараллеливания. Однако вместе с увеличением степени параллельности растут потери на взаимодействия между элементами коллектива вычислителей. Эти потери складываются из следующих компонент:

- организации единичного взаимодействия,
- преодоления расстояний в системе,
- синхронизации, т.е. обеспечении согласованного во времени исполнения операторов системы $\bar{A}$ алгоритмов функционирования,
- сбора, обработки и передачи управляющей информации.

Если с ростом числа n вычислителей в системе время вычислений убывает обратно пропорционально n, а время организации единичного взаимодействия - константа, то время, затрачиваемое на преодоление расстояний в системе, растет, как $\sqrt[3]{n}$, а все другие потери имеют или полиномиальную или даже экспоненциальную оценку. Таким образом, если рост потерь никак не ограничен, найдется такое число n_0, что при $n > n_0$ коллектив вычислителей начнет терять производительность. Известно, например, что для задач линейной алгебры и математической физики легко строятся параллельные программы с числом ветвей равным числу уравнений системы [6]. При этом n достигнет величин 10^4-10^5 и более. Между тем [12] число n_0 для системы СУММА имеет порядок 10^1-10^2 из-за наличия общего ресурса - канала связи.

Представляется возможным ограничить долю потерь в общем времени работы вычислительной системы последовательным применением принципа б л и з к о д е й с т в и я, т.е. такой организации коллектива вычислителей, при которой выполнение очередного шага алгоритма функционирования $A_i \in A$ системы $\bar{A}$ (или программы, реализуемой i-м вычислителем) зависит только от состояний элементов системы $\bar{A}$, находящихся не далее, чем на некотором фиксированном расстоянии от i-го элемента. Далеко не очевидно, каковы должны быть конструктивные особенности таких систем. Для ответа на этот вопрос конкретизируем принцип близкодействия.

I. Л о к а л ь н о с т ь. Говоря о расстояниях в системе $\bar{A}$ необходимо учесть существование в ней двух метрик: физического пространства и графа связей системы $\bar{A}$. Пусть l_{ij} - расстояние между i-м и j-м элементами системы в метрике графа связей, d_{ij} - расстояние между теми же элементами в физическом пространстве. Потребуем, чтобы расположение элементов в пространстве удовлетворяло следующим условиям:

$$\forall (i,j) \in E\{(l_{ij}=1) \wedge (d_{ij} \gg \max_{(i,j),(k,m)\in E} |d_{ij}-d_{km}|)\}, \tag{2}$$

$$\exists k_0 \forall l_{ij}, l_{km} \{(l_{ij} < k_0) \wedge (l_{km} < k_0) \wedge (l_{ij} > l_{km}) \rightarrow (d_{ij} > d_{km})\}, \tag{3}$$

где i, j, k, m - номера элементов множества A, E - множество связей системы $\bar{A}$, k_0 - положительное целое число. При этих условиях в качестве расстояния между элементами можно брать только расстояния в метрике графа связей системы $\bar{A}$.

Система $\bar{A}$ называется локальной, если каждый очередной шаг алгоритма $A_i \in A$ зависит только от состояния окрестности, имеющей фиксированный радиус $k < k_0$.

Из описания технической модели коллектива вычислителей видно, что соответствующая алгоритмическая модель не будет локальной из-за наличия обобщенных переходов. Кроме того, если рассмотреть подробнее методику крупноблочного распараллеливания [6], то видно, что система $\bar{P}$ параллельных ветвей в большинстве случаев имеет полный граф информационных связей. При реализации системы $\bar{P}$ систе -

мой $\bar{A}$ метрики этих систем не совпадают, что приводит к появлению глобальных схем обмена информацией: трансляционной и трансляционно-циклической. В этих системах предполагается, что некоторая ветвь параллельной программы передает данные одновременно всем остальным.

Таким образом, если даже система $\bar{A}$ локальна, остается еще вопрос организации параллельных вычислений с локальными взаимодействиями, причем такими, чтобы метрики систем $\bar{P}$ и $\bar{A}$ совпадали. Этому требованию отвечают конвейерные схемы, широко применяемые в магистральных вычислительных машинах [16-19]. Для однородных коллективов вычислителей соответствующая возможность показана в [20].

2. Асинхронность. Система называется асинхронной, если порядок срабатывания ее элементов не зависит от сигналов тактового генератора, а определяется вычисляемыми логическими условиями (спусковыми функциями).

Если дан алгоритм функционирования, то соответствующее техническое устройство будет синхронным или асинхронным в зависимости от интерпретации этого алгоритма. В частности, в [9] дан метод синхронной интерпретации алгоритма функционирования. Асинхронные клеточные автоматы рассмотрены в [21,22]. В [6] показано, что с ростом тактовой частоты падает принципиально достижимая производительность синхронного коллектива вычислителей. Отсюда делается вывод, что системы должны иметь низкую тактовую частоту.

Асинхронная интерпретация снимает этот парадокс, но для эффективного построения асинхронных систем спусковые функции должны быть явно выписаны в системе $\bar{A}$ алгоритмов функционирования, иначе сложность технического устройства может резко возрасти. Например, в [15] показано, что асинхронная интерпретация алгоритмов параллельных подстановок втрое увеличивает число состояний системы. Однако, эта оценка приложима, по-видимому, только к рассмотренной в [15] модели и способу асинхронной интерпретации.

3. Распределенность. Система называется распределенной, если в ней нет ресурса (общей шины, памяти, регистра, устройства), используемого всеми элементами системы в режиме разделения времени.

Кроме очевидного нарушения условий локальности общий ресурс накладывает дополнительные системные ограничения на рост производительности. Пусть α - интенсивность потока заявок на ресурс от отдельного элемента, β - интенсивность обслуживания, n - число элементов. Известно, что при

$$L\alpha > \beta \qquad (4)$$

производительность системы достигает насыщения и дальнейшее наращивание числа n приводит только к снижению эффективности использования элементов.

4. Д е ц е н т р а л и з а ц и я. Система децентрализована, если в ней нет общего ресурса (вычислителя, алгоритма, подсистемы), выделенного специально для решения задачи управления коллективом.

За исключением случаев принудительного управления (типа синхронизации от тактового генератора) управляющий орган представляет собой общий ресурс с тем отличием, что с ростом n инстенсивность обслуживания заявок этим органом резко падает и неравенство (4) достигается значительно быстрее. В этом случае применяют иерархическое управление системой.

В полностью децентрализованной системе алгоритмы системы все вместе и каждый в отдельности, оптимизируют некоторую функцию качества функционирования, пользуясь информацией только из окрестности заданного радиуса. Во всяком случае, управляющая иерархическая надстройка должна появляться в проекте системы в самых крайних случаях. Ниже будет предполагаться отсутствие каких-либо специальных управляющих элементов.

4. Математическая модель коллектива вычислителей

Предлагаемая модель коллектива вычислителей удовлетворяет всем условиям близкодействия, т.е. является Локальной, Асинхронной, Распределенной и Децентрализованной системой: ЛАРД-системой. Построим ее формальное описание.

Н о с и т е л ь. Множество L номеров (имен) алгоритмов функционирования системы $\bar{X}$ называется носителем ЛАРД-системы. Будем полагать, что $|L| = n$ - конечно. Иногда удобно считать L счетным, но ниже нам это обобщение не потребуется.

С т р у к т у р а. Рассмотрим однородный граф $G \rightleftharpoons \langle L,E \rangle$, где L - носитель, $E \subseteq L^2$ - множество пар элементов из L, связанных в G.

Пусть Aut - транзитивная подгруппа группы автоморфизмов графа G. Относительно E условимся, что если $(i,j) \in E$, то и $(j, i) \in E$ и пара $[(i,j),(j,i)]$ - ребро графа G. Пронумеруем ребра, выходящие из вершины $l \in L$, в некотором порядке $0, 1, \ldots, (m-1)$, где m - число ребер, инцидентных вершине l графа G. При этом каждое ребро получит две отметки. Потребуем, чтобы подгруппа Aut сохраняла отметки ребер графа G, тогда, согласно [23], G – групп - граф подгруппы Aut с точностью до направлений дуг, а отметки ребер соответствуют образующим $a \in Aut$. Автоморфизмы подгруппы Aut являются сдвигами графа G вдоль ребер и иногда удобно приписать ребрам направления, соответствующие направлению сдвигов $a \in Aut$.

Граф G называется структурой коллектива вычислителей.

Введем следующие определения. Пусть $l \in L$, множество $\partial l \subseteq L$ вершин графа G, находящихся на расстоянии I от вершины l, называется окрестностью или ближайшими соседями вершины l, множество $\partial^k l \subseteq L$ вершин графа G, находящихся на расстоянии не более чем k от вершины l и $l \notin \partial^k l$ окрестностью k-го порядка вершины l. Аналогично для любого $K \subset L$ определяются его окрестности $\partial^i K \subseteq L$, причем $K \cap \partial^i K = \emptyset$, $i \leq j \to \partial^i K \subseteq \partial^j K$.

Как и в [14] будем считать, что существуют рекурсивно вычислимые функции φ_i, φ_i^{-1}, такие, что $\partial l_i = \varphi_i(l)$, $l = \varphi_i^{-1}(\partial l_j)$, где $i, j = 0, 1, \dots, (m-1)$ – отметки ребер, причем в последнем случае отметки одного и того же ребра; $\partial l_i \in \partial l$ – элемент окрестности, соединенный с l-м элементом i-м ребром.

Независимость вида функций нумерации от l – естественное следствие однородности графа G.

Элементарный автомат. В алгоритмической модели (I) коллектива вычислителей каждому элементу $l \in L$ сопоставлен алгоритм $A_l \in A$. Реализация алгоритма A_l в реальном времени называется процессом. Состояние процесса есть совокупность значений переменных множества $Z_l \cup \{I_l\}$, где Z_l – множество переменных l-го алгоритма, а I_l – счетчик номеров операторов алгоритма. Очевидно, что множество состояний процесса практически необозримо, однако при исследовании взаимодействий процессов нет необходимости различать состояния, эквивалентные по своему влиянию на соседей. В соответствии с этим, функциональным элементом предлагаемой ЛАРД-системы является асинхронный вероятностный автомат, связанный с алгоритмом A_l следующим образом.

Разобъем множество состояний процесса на классы эквивалентности по указанному признаку и введем фактор-множество $X_l \rightleftharpoons \{x_i / i = 0, 1, \dots, |X_l| - 1\}$. Множество X_l является алфавитом состояний l-го элементарного автомата предлагаемой модели. Элементарный автомат может находиться в состоянии $x \in X_l$ неопределенное время и перейти затем в любое состояние $y \in X_1$, причем вероятность этого перехода зависит как от x, так и от состояния окрестности из множества $Y_{\partial 1} = X_1^m$, где m - число соседей. Интенсивности переходов элементарного автомата зададим системой $\Lambda \rightleftharpoons \{\lambda_{xy}(y_{\partial 1}) \;/\; x, y \in X_1, y_{\partial 1} \in Y_{\partial 1}\}$.

Вероятность перехода из $x \in X$ в $y \in X$ за время Δt при условии $y_{\partial 1} \in X^m$ равна

$$p_{xy}(y_{\partial 1}) = \lambda_{xy}(y_{\partial 1}) \cdot \Delta t + o(\Delta t). \tag{5}$$

Такое задание $p_{xy}(y_{\partial 1})$ - частный случай, но, во-первых, оно является простейшим, во-вторых, многие вопросы теории ЛАРД-системы не зависят от (5), в-третьих, другие способы задания вероятностей $p_{xy}(y_{\partial 1})$ имеют не больше оснований. Наконец, не обязательна и недетерминированность выбора очередного состояния из X, детерминированный асинхронный автомат является частным случаем.

Будем рассматривать только гомогенные системы, такие, что для всякого 1: $X_1 = X$, $Y_{\partial 1} = X^m$, тогда подгруппа Aut действует не только на структуре, но и на системе элементарных автоматов, связанных этой структурой.

Итак, элементарным автоматом ЛАРД-системы является асинхронный автомат $M \rightleftharpoons \langle X, X^m, \Lambda \rangle$. Выходной алфавит автомата M совпадает с алфавитом его состояний X, причем значение состояния $x \in X$ передается по всем выходам автомата в его окрестность.

В качестве окрестности в некоторых случаях может потребоваться окрестность k-го порядка, тогда входной алфавит 1-го элементарного автомата будет

$$Z = Y_{\partial^k 1} = X^{|\partial^k 1|}. \tag{6}$$

К о н ф и г у р а ц и и. Функция $x(1)$, сопоставляющая каждому $1 \in L$ состояние $x(1) \in X$ (иначе $x(1): L \to X$), называется конфигурацией. Зафиксировав определенный порядок перечисления элементов из L можно рассматривать конфигурации, как упорядоченные n-ки из X. Тогда множество Y всех конфигураций - декартова степень множества X: $Y = X^n$, $|Y| = |X|^n$.

Пусть $K \subset L$ и $x(1): K \to X$; такое сужение функции $x(1)$ называется конфигурацией на K и обозначается x_K, y_K, в частности x_1, есть состояние 1-го автомата - случайная величина называемая также к о м п о н е н т о й. В контекстах, связанных с вероятностными мерами на Y, конфигурация на K понимается, как множество конфигураций на L, совпадающих на K.

Введем операцию объединения на множестве непересекающихся сужений конфигураций. Пусть $K \subset L$, $M \subset L$ и $K \cap M = \emptyset$, тогда $(y_K, y_M) = y_{K \cup M}$ – конфигурация на $K \cup M$, совпадающая с y_K на K и с y_M на M. Будем говорить, что x_M содержит x_K, если $K \subseteq M$, $x_M = (x_K, y_{M \setminus K})$.

Будем называть контекстом конфигурации x_K конфигурацию $x_{\partial K}$ на ∂K. В частности, введенное выше состояние $y_{\partial 1}$ – контекст 1-го автомата. В случае (6) будем говорить о контексте K-го порядка.

Параллельные подстановки. Подстановкой называется выражение вида $(x_K, x_{\partial K}) \to x'_{K'}$, обозначающее, что конфигурация x_K в контексте $x_{\partial K}$ заменяется конфигурацией $x'_{K'}$, где $K' \subseteq K \cup \partial K$. В дальнейшем будем пользоваться стационарными и неймановскими подстановками $\Pi(x, x_{\partial 1})$: $(x_1, x_{\partial 1}) \to y_1, z_1, \ldots, r_1$; $\lambda_{xy}(x_{\partial 1}), \ldots, \lambda_{xr}(x_{\partial 1})$, где список $y_1\ z_1, \ldots, r_1$ – это возможные альтернативы значений 1-го элемента в результате применения $\Pi(x, x_{\partial 1})$, $x, y, z, \ldots, r \in X$, $x_{\partial 1} \in Y_{\partial 1}$; список $\lambda_{xy}(x_{\partial 1}), \ldots, \lambda_{xr}(x_{\partial 1})$ – соответствующие интенсивности реализаций этих альтернативных значений; $\lambda_{xy}(x_{\partial 1}), \ldots, \lambda_{xr}(x_{\partial 1}) \in \Lambda$. В силу однородности индекс 1 можно везде опустить.

Элементарный автомат задается системой подстановок

$$\Pi \doteq \{\Pi(x, x_{\partial 1}) / x \in X,\ x_{\partial 1} \in X^{a}\},$$

при этом некоторые интенсивности $\lambda_{xy} \in \Delta$ могут быть равными нулю, и тогда соответствующие элементы подстановок $\Pi(x, x_{\partial 1})$ или даже целые подстановки опускаются.

Итак, ЛАРД-система задана, если задана ее структура G и элементарный автомат M. Формально ЛАРД-система это двойка $\langle G, M \rangle$ или пятерка $\langle L, E, Aut, X, \Pi \rangle$.

5. Функциональные свойства ЛАРД-систем

Опишем теперь, как задается процесс функционирования ЛАРД-системы.

Интерпретация системы Π. Говорят, что подстановка $(x, x_{\partial 1})$ применима к конфигурации $y \in Y$, если y содержит левую часть $(x, x_{\partial 1})$ подстановки $\Pi(x, x_{\partial 1})$, хотя бы для одного $l \in L$. Конфигурация y', получающаяся заменой соответствующего x_l на одно из альтернативных значений правой части подстановки $\Pi(x, x_{\partial l})$, называется результатом применения подстановки $\Pi(x, x_{\partial l})$ к конфигурации y.

Система подстановок Π применима к конфигурации $y \in Y$, если найдется подстановка $\Pi(x, x_{\partial l}) \in \Pi$, применимая к y.

Пусть $y \in Y$ в момент времени t содержит некоторое множество сужений $S \supseteq \{(x_i, x_{\partial i}), (x_j, x_{\partial j}), \ldots, (x_l, x_{\partial l})\}$, совпадающих с левыми частями подстановок системы Π. В отличие от [17] применение системы Π подстановок в ЛАРД-системе состоит в том, что в любой следующий момент времени $t_1 > t$ любое подмножество сужений из S приводит к срабатыванию соответствующего подмножества подстановок из Π, с любой заменой компонент x_i $x_j, \ldots, x_l$ альтернативными значениями из правых частей срабатывающих подстановок. Получившаяся новая конфигурация $y' \in Y$ называется результатом применения системы подстановок Π к конфигурации y, если множество сработавших подстановок непусто.

Функционирование ЛАРД-системы состоит в том, что, начиная с некоторой исходной конфигурации $y_0 \in Y$, система подстановок применяется к каждому текущему результату $y(t)$ до тех пор пока не встретится конфигурация $y_1(t)$, к которой система Π неприменима. Для обозначения такой конфигурации, завершающей процесс, будем применять термин **т у п и к**.

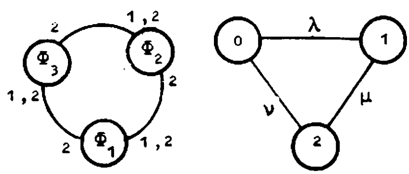
О р д и н а р н о с т ь. Только что описанный способ функционирования ЛАРД-системы вполне адекватен поведению асинхронных технических систем, но крайне неудобен для анализа. Введем класс ординарных ЛАРД-систем. Пусть $Z(y) \subseteq Y$ и любой элемент множества конфигураций $Z(y)$ достижим из начальной конфигурации $y \in Y$ применением системы подстановок Π. Запретим теперь одновременное срабатывание нескольких подстановок, т.е. в каждый момент времени разрешим применение любой, но только одной подстановки. Пусть $Z'(y)$ — множество конфигураций, достижимое из y таким ограниченным применением системы Π.

Множество $Z(y) \subseteq Y$ называется ординарным, если выполнено $Z(y) = Z'(y)$. ЛАРД-система называется ординарной, если для любого $y \in Y$ множество $Z(y)$ достижимых конфигураций ординарно. Срабатывание подстановок в ординарной ЛАРД-системе можно рассматривать как ординарный поток событий, а все функционирование как марковский процесс с системой состояний Y. Тупикам соответствуют по -

глощающие состояния. Интенсивности переходов легко получаются из системы Λ. Естественным образом переносится на ординарные ЛАРД-системы и все другие термины, относящиеся к теории марковских процессов. В частности, множество сообщающихся конфигураций, не содержащее выходов в тупики, будем называть э р г о д и ч е с к и м.

Ординарность обычно предполагается в теории марковских процессов, но в рассматриваемом случае требуются указанные основания, так как если есть логическая возможность одновременного срабатывания нескольких подстановок, то эта возможность рано или поздно реализуется в технической системе. Причина - в конечности времен переходов и, следовательно, в конечной вероятности их совпадения.

Для демонстрации последствий реальной неординарности рассмотрим упрощенную ЛАРД-систему, связанную с проблемой "обедающих философов". На рис. I,а представлена структура G из трех философов (вершины графа G) и трех вилок (ребер графа). Отметки ребер обозначают состояния философа, при которых он захватывает соответствующую вилку. Транзитивная группа Aut, сохраняющая отметки ребер, есть циклическая группа C_3 .



а) б)

Рис. I

Поведение философа, как если бы он был один, задано графом состояний на рис. I,б. В состоянии 0 философ положил вилки и думает, в состоянии I - взял правую вилку, 2 - взял еще и левую вилку и ест спагетти двумя вилками.

Рассмотрим фактор-состояния системы, изображая их конфигурациями-представителями. Остальные элементы фактор-состояния получаются из представителя циклическими сдвигами. Очевидно, что состояния 0I2, 022, II2, I22, 222 физически невозможны, так как два философа не могут есть одной вилкой одновременно, состояние III - тупик, остальные состояния допустимы.

Нетрудно убедиться, что при ординарной интерпретации допустимые состояния образуют эргодическое множество, если принять следующую систему подстановок:

$$\bar{2}00 \to I;\ \lambda,$$
$$00\bar{2} \to I;\ \lambda,$$
$$0I\bar{2} \to 2;\ \mu,$$
$$02\bar{2} \to 0;\ \nu.$$

Для наглядности состояния левого и правого соседей элемента x_1 записаны здесь слева и справа, соответственно. Состояние $\bar{2}$ обозначает 0 или I в целях компактной записи соответствующего множества подстановок.

Размеченный граф состояний соответствующего ординарного марковского процесса представляет собой подграф, изображенный сплошными стрелками на рис.2. Однако, в силу неординарности рассматриваемой ЛАРД-системы подстановка $\bar{2}00 \to I;\lambda$ может сработать одновременно на всех нулевых компонентах конфигураций 000 и I00, что приведет к тупику III. На рис.2 эти переходы изображены штриховой стрелкой. Простейшим выходом из возникшего затруднения является введение в систему подстановки: III $\to$ 0; δ ,описывающей "уступчивость" философа (см. рис.2). Однако уже в системе из четырех философов конфигурация III0 является допустимой и переход вида III0 $\xrightarrow{\delta}$ I0I0 излишество, хотя, может быть и не снижающее эффективность использования вилок.

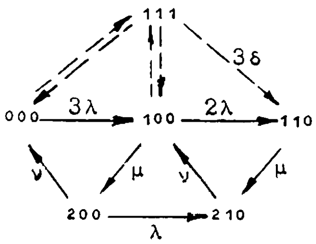


Рис. 2

Итак, если нам удалось обнаружить, что система ординарна, то мы можем применять ограниченное (ординарное) правило срабатывания подстановок системы П и рассматривать поток событий в системе, как ординарный, а весь процесс представлять как марковский. Проверка ординарности, однако, далеко не тривиальная задача.

Э р г о д и ч н о с т ь. Система называется эргодической,если для любой конфигурации $y \in Y = X^n$ множество достижимых конфигураций $Z(y) = Y = X^n$, т.е. граф состояний такой системы сильно связен.

Условие ординарности для эргодической системы выполняется автоматически, следовательно, ее функционирование может быть описано марковским процессом. Финальные вероятности состояний этого процесса задают положительную вероятностную меру $\mu > 0$ на множестве конфигураций.

Пусть $K = \{1, \partial 1_0, \partial 1_1, \ldots, \partial 1_{m-1}\}$, $|K| = m+1$. Фиксируем заданный так порядок элементов множества K, тогда если Y_K - множество всех конфигураций на K , то $|Y_K| = |X^{m+1}| = |X|^{m+1}$. Подстановки П ЛАРД-системы гарантируют эргодичность, если они удовлетворяют следующим условиям.

1) Множество $П_л$ левых частей подстановок системы $П$ есть множество Y_K.

2) Множество сужений на , продуцируемое подстановками системы $П$ из сужений множества Y_K, есть множество $Y'_K = Y_K$.

6. Проблематика теории ЛАРД-систем

Изложенная выше модель коллектива вычислителей сама по себе является интересным объектом математических исследований с широким спектром применений. В контексте данной работы ЛАРД-система построена для целей корректного проектирования и организации функционирования однородных вычислительных систем с программируемой структурой. С этой точки зрения проблематика теории ЛАРД-систем распадается на следующие разделы.

1. Т е о р и я   с т р у к т у р исследует граф $G = \langle L, E \rangle$. Основная задача этого раздела - синтез структур, отвечающих известным требованиям, и анализ свойств получаемых структур [8]. С момента появления публикации [8] теория структур значительно продвинулась [24-26], и в настоящее время получен каталог оптимальных структур с циклической группой автоморфизмов.

2. Т е о р и я   ф у н к ц и о н и р о в а н и я исследует систему подстановок $П$. Задачей этого раздела является, во-первых, установление критериев для таких свойств системы, как ординарность, эргодичность, разложимость, наличие тупиков и, во-вторых, синтез асинхронных автоматов, обладающих заданным поведением в заданной структуре. Пример решения такой задачи был дан в п.2 §5. В целом теория функционирования относится к обширной области параллельного микропрограммирования [27].

3. Т е о р и я   э ф ф е к т и в н о с т и исследует вероятностные меры на множестве Y конфигураций ЛАРД-системы. С вероятностной точки зрения ЛАРД-система представляет собой многокомпонентную случайную систему [28]. Задача раздела - установление критериев эффективности системы, разработка методов расчета или оценки вероятностей заданных множеств конфигураций, исследование свойств случайных полей [29-32], являющихся стационарными состояниями ЛАРД-системы.

4. Т е о р и я   л о к а л ь н о г о   п а р а л л е л ь н о г о   п р о г р а м м и р о в а н и я исследует методы разложения вычислений на параллельные процессы, которые взаимодействуют между собой только в пределах заданной окрестности. В настоящее

время уже ясно, как это можно сделать по крайней мере для тех классов задач, которые решены в [6] методом крупноблочного распараллеливания.

В заключение автор благодарит О.Л.Бандман за плодотворное обсуждение.

Литература

1. ЕВРЕИНОВ Э.В., ХОРОШЕВСКИЙ В.Г. Однородные вычислительные системы. Новосибирск: Наука, 1978. - 320 с.

2. CODD E.F. Cellular Automata.-New York-London: Academic Press,1968. - 122 p.

3. YAMADA H., AMOROSO S. Structural and behavioral equivalences of tesselation automata.- Inform.Contr.,1971, v. 18, N 1, p.1-31.

4. NISHIO H., KOBUCHI Y. Fault Tolerant Cellular Spaces.- J. Comp.Sys.Sci.,1975,v.11, N 2,p.150-170.

5. ALADYEV V. Mathematical theory of homogenous structures and their applications.- Tallinn: Valgus,1980. - 267 p.

6. ЕВРЕИНОВ Э.В., КОСАРЕВ Ю.Г. Однородные вычислительные системы высокой производительности.Новосибирск: Наука,1966, -308 с.

7. МАМЗЕЛОВ И.А. Некоторые вопросы теории модели коллектива вычислителей. -В кн.: Вычислительные системы. Вып. 2. М., Финансы и статистика, 1981. с. 63-68.

8. ВОРОБЬЕВ В.А.,КОРНЕЕВ В.В.Некоторые вопросы теории структур однородных вычислительных систем. -В кн.: Вычислительные системы. Вып. 60. Новосибирск, 1974, с. 3-16.

9. ИЛОВАЙСКИЙ И.В., СИДРИСТЫЙ Б.А. Основы теории проектирования цифровых машин и систем. Новосибирск: Наука, 1976. -128 с.

10. ВОРОБЬЕВ В.А. Моделирование системы параллельных процессов на Р-ЛЯПАСе. -В кн.: Вычислительные системы. Вып. 51. Новосибирск, 1972, с.82-96.

11. LIPTON R.J., SNYDER L., ZALZSTEIN Y. A Comparative Study of Models of Parallel Computation. - In: 15th Annual Symp. on Switching and Automata Theory,1974,p.145-155.

12. ВОРОБЬЕВ В.А.,СЕДУХИН С.Г. Распределенное управление процессами в однородных вычислительных системах. - В кн.: Гибридные вычислительные машины и комплексы. Вып. 1. Киев: Наукова думка, 1979, с. 40-48.

13. ВОРОБЬЕВ В.А., КАШУН И.Н., СЕДУХИН С.Г. Исследование децентрализованных дисциплин взаимодействия между элементарными машинами однородной вычислительной системы. - В кн.: Однородные вычислительные системы и среды. Часть 1. Материалы IУ Всесоюзной конференции. Киев: Наукова думка, 1975, с.36-38.

14. КОРНЕВ Ю.Н., ПИСКУНОВ С.Н.,СЕРГЕЕВ С.Н. Алгоритмы обобщенных подстановок и их интерпретация сетями автоматов и однородными машинами. -Изв. АН СССР. Техн. кибернетика, 1971, №6, с.131-142.

15. БАНДМАН О.Л. Асинхронная интерпретация параллельных микропрограмм. Препринт ОВС-14, Институт математики СО АН СССР, Новосибирск, 1981. - 36 с.

16. RUMBAUGH J. A Data Flow Multiprocessor. - IEEE Trans. on Comp.,1977,v.C-26,N 2,p.138-147.

17. RAMAMORTHY C.V.,LI H . Pipeline architecture.- Computing Surveys, 1977,v.9, N 1,p.61- .2.

18. RUSSEL R.M. The CRAY-1 computer system. - Commun.ACM., 1978, V.21, N 1,p.63-72.

19. THURBER K.J. Parallel processor architectures. Pt 1: General purpose systems. Pt 2: Special purpose systems. - Computer Design,1979,v.18, N 1,p.89-97; N 2,p.103-104.

20. МИШИН А.И., СЕДУХИН С.Г. Вычислительные системы и параллельные вычисления с локальными взаимодействиями. -В кн.: Математическое обеспечение вычислительных систем (Вычислительные системы, 78). Новосибирск, 1979, с. 20-103.

21. GOLZE U. (A-)Synchronous (Non-)Deterministic Cell Spaces Simulating Each Other.- J.Comp.Sys.Sci.,1978,v.17,N 2,p.176-193.

22. PRIESE L. A Note on asynchronous cellular automata.- J. Comp. and Sys.Sco.,1978,v.17, N 2,p.237-252.

23. WAGNER E.G. On connecting modules together uniformly to form a modular computer.- IEEE Trans.on EC, 1966,v.EC-15, N 6, p.863-972.

24. ВОРОБЬЕВ В.А. Простейшие структуры однородных вычислительных систем. -В кн.: Вычислительные системы, вып. 60. Новосибирск, 1979, с. 35-49.

25. WONG C.K., Don COPPERSMITH. A Combinatorial Problem Related to Multimodule Memory Organization.- J.Assoc.comp.Machinery,1974,v.21, N 3,p.392-402.

26. МОНАХОВА Э.А. Синтез оптимальных диофантовых структур. -В кн.: Вопросы теории и построения вычислительных систем. (Вычислительные системы, вып. 80). Новосибирск, 1979, с. 18-35.

27. Методы параллельного микропрограммирования /Анишев П.А., Ачасова С.М., Бандман О.Л., Пискунов С.В., Сергеев С.Н. - Новосибирск: Наука, 1981. - 180 с.

28. ДОБРУШИН Р.Л.Марковские процессы с большим числом локально взаимодействующих компонент - существование предельного процесса и его эргодичность. - Проблемы передачи информации, 1981, т. 7, вып. 2, с. 70-87.

29. PRESTON C. Random fields. - Lect.Notes Math., 1976. V.534. - 200 p.

30. SPITZER F. Markov random fields and Gibbs ensembles.-Amer. Math.Mon.,1971,v.78, N 2,p.142-154.

31. SULLIVAN W.G. Potentials for Almost Markovian Random Fields. - Commun.Math.Phys.,1973,v.33,N 1,p.61-74.

32. MOUSSOURIS J. Gibbs and Markov random system with constraints. - J.Statist.Phys.,1974,v.10,N 1,p.11-33.

Поступила в ред.-изд.отд.
2 июня 1982 года