

## АВТОМАТНАЯ МОДЕЛЬ УПРАВЛЕНИЯ ПАРАЛЛЕЛЬНЫМИ И РАСПРЕДЕЛЕННЫМИ ВЫЧИСЛИТЕЛЬНЫМИ ПРОЦЕССАМИ БОЛЬШОЙ РАЗМЕРНОСТИ

Строится автоматная модель управления распределёнными и параллельными вычислительными процессами. На основе этой модели предлагается метод построения балансного уравнения вычислительных процессов, сложность которого слабо зависит от размера решаемых задач. Обсуждаются направления возможного использования для формирования единого алгоритмического пространства распределённых и параллельных вычислений.

**AUTOMATA MODEL FOR CONTROL OF PARALLEL AND DISTRIBUTED PROCESSES OF LARGE DIMENSION / Yu.S. Zatuliveter (V.A. Trapeznikov Institute of Control Sciences, Profsoyuznaya 65, Moscow 117342, Russia, E-mail: [zvt@ipu.rssi.ru](mailto:zvt@ipu.rssi.ru)).** The automata model for control of distributed and parallel computing processes is under construction. On the basis of this model the method of constructing of the balanced equation of the computing processes which complexity poorly depends on the size of solved tasks is offered. The directions of possible use for formation of uniform algorithmic space of the distributed and parallel computing are discussed.

### 1. Введение

Высокий технический уровень компьютерных устройств связанных сетями предопределяет колоссальный совокупный вычислительный потенциал компьютерной среды (КС). Так, среда WWW связывает уже около 2 млрд. ПК. Только эта среда уже обладает суммарной производительностью более 1-2 Эфлопс ( $10^{18}$  флопс), совокупной оперативной памятью более 1-2 Эбайт ( $10^{18}$  байт) и сотнями экзбайт долговременной памяти. Параллельно с этой средой развиваются сети мобильной связи (более 4,5 млрд. абонентов). В стандартах 3G и 4G они быстро интегрируются в WWW, увеличивая не только вычислительный потенциал КС, но и привнося новые функциональные возможности, связанные с перемещениями вычислительных узлов (универсальные платёжные терминалы, сенсоры, мониторинг, навигация и многое др.).

Однако быстро растущий совокупный вычислительный, функциональный и системообразующий потенциал глобальной КС остаётся нераскрываемым ввиду её крайней *разнородности*, де-факто сложившейся за два десятилетия становления. Стихия разнородности проявляется на всех уровнях аппаратных и программных средств. По мере увеличения масштабов КС и размеров задач непрерывное воспроизводство и накопление разнородных форм представления данных и программ в такой среде становится

фундаментальным препятствием на путях интеграции глобально распределённых процессов и систем, поскольку сложность задач интеграции в условиях изначальной разнородности носит комбинаторный характер [1]. С увеличением количества связанных вычислительных узлов в вычислительных системах, реализуемых в условиях разнородности, трудоёмкость интеграции растёт с ускоряющимся опережением (сверхлинейно), что является проявлением "комбинаторного проклятия" размерности.

Разнородность глобальной КС становится непреодолимым препятствием на путях раскрытия её совокупного потенциала. Однако без его раскрытия невозможно решение практически неограниченного разнообразия возникающих задач полномасштабной переработки экспоненциально растущих потоков и объёмов глобально распределённых данных, сопровождающих функционирование мировой социосистемы в глобальном информационном пространстве [1]. Своевременно неперерабатываемая информация создаёт эффект лавинного нарастания в глобальном информационном пространстве информационных шумов. В условиях растущих шумов неизбежно снижается качество процессов управления, поэтому устойчивое развитие социосистемы становится невозможным [1,2].

Необходимым условием полного раскрытия вычислительного потенциала КС становится *выявление и устранение фундаментальных причин* непрерывного воспроизводства разнородности [1-4]. Без этого невозможно формирование в ней глобального и бесшовно программируемого алгоритмического пространства распределённых и параллельных вычислений [5]. Такое пространство должно охватить всю совокупность компьютерных устройств, связанных глобальными сетями – от смартфонов, ПК и встраиваемых компьютерных устройств до суперкомпьютеров – и обеспечить их согласованное алгоритмическое взаимодействие в едином формате представления данных и программ с бесшовным программированием задач любого размера. [3,4].

Одна из ключевых проблем формирования такого пространства – необходимость создания единой универсальной модели распределённых и параллельных вычислений, эффективно реализуемой в больших и сверхбольших компьютерных средах с сильно связными вычислительными узлами. В перспективе такая модель должна стать единой основой для реализации бесшовного программирования глобально распределённых вычислений, как в распределённых (сетевых), так и высокопараллельных (многопроцессорных) средах.

## 2. Предметная область

Проблематика параллельных вычислений, как самостоятельное направление в компьютерных науках, сформировалась в 60-х годах. В отличие от последовательных вычислений, которые на логическом и инженерном уровнях аксиоматически регламентированы классической моделью универсальных вычислений Дж. фон Неймана, параллелизм предполагает множественность одновременного исполнения операций. При этом допустимый состав множеств исполняемых на каждом шаге действий в условиях машинной среды многовариантен. В связи с этим возникает две, характерные для параллелизма, проблематики:

- определение допустимого, с точки зрения корректности вычислений, состава множеств операций, исполняемых на каждом шаге;
- из допустимых вариантов состава множеств выявление таких, которые обеспечивают как можно более высокую эффективность исполнения в той или иной вычислитель-

ной среде (по наборам таких критериев как время, объёмы памяти, надёжность, энергопотребление, стоимость и др.).

Многовариантные и многокритериальные проблемы параллелизма отличаются высоким уровнем сложности. Математическая особенность таких проблем в том, что они формулируются в многовариантных пространствах и требуют нахождения решений, связанных с преодолением комбинаторной сложности.

Следует отметить, что за прошедшие десятилетия наработано большое количество моделей параллельных вычислений. В отличие от иных компьютерных областей науки и техники математические методы получения и обоснования результатов в данной проблематике играют существенно более значимую роль. Многие известные модели параллельных и распределённых вычислений опираются на математические методы обоснования. К сожалению, далеко не все они отвечают растущим требованиям практической применимости из-за отсутствия средств эффективной технической реализации в рамках существующих технологий массового производства элементной базы и других компьютерных компонентов.

Модели параллельных вычислений и соответствующие им архитектуры начали активно разрабатываться в 70-е годы. С тех пор их наработано огромное количество. Одни создавались как абстрактные формализмы и конструкции вне требований практической реализуемости. Другие имели жёсткую привязку к тем или иным частным структурным, архитектурным или техническим решениям, что, во-первых, предельно сужало диапазоны их использования и, во-вторых, в большинстве случаев не отвечало требованиям практической реализуемости компьютеров и программ в массовых тиражах. Поэтому и они не могут составить основу для нового индустриального мэйнстрима.

Проблематика распределённых и параллельных вычислений [6-9] до сих пор продолжает развиваться по двум слабосвязным направлениям. Теоретические модели параллельных/ распределённых вычислений и, соответственно, практические архитектуры – высокопараллельные компьютерные и распределённые сетевые.

Компьютерные архитектуры лежат в основе многопроцессорных систем, в которых физические расстояния между процессорами стремятся делать как можно меньше, что является одним из основных факторов повышения коммутируемой связности процессоров, скорости их взаимодействия, а значит и вычислительной производительности.

Распределённые архитектуры реализуются многомашинными системами из многих компьютеров, связанных сетями, как локальными, так и глобальными, в которых на системных и прикладных уровнях формируются вычислительные среды, которые в качестве узлов могут включать как стационарные компьютерные устройства, так и мобильные. В распределённых архитектурах расположение и изначально отдалённых компьютеров предопределено их целевыми назначениями, поэтому физические расстояния между компьютерами неустраняемы.

При этом следует отметить, что, несмотря на отмеченные различия, и многопроцессорные, и распределённые архитектуры имеют принципиальное сходство, поскольку реализуют одновременно протекающие (параллельные) процессы, которые обеспечивают согласованное в соответствии с алгоритмами решаемых задач взаимодействие совокупных вычислительных ресурсов их составляющих. До сих пор различия доминировали над сходством, что приводило к непрерывному воспроизводству разнородных, трудно интегрируемых решений. Однако в условиях сверхбольших размеров глобальной КС дальнейшее развитие в условиях разнородности не имеет стратегических перспектив.

До последнего времени единая модель распределённых и параллельных вычислений, которая, отвечала бы, подобно классической модели последовательных вычислений Дж.

фон Неймана, требованиям массового производства компьютеров и программ пока не предложена. Большое разнообразие практически реализованных многопроцессорных архитектур создаёт непреодолимые проблемы тиражирования параллельных компьютеров и программ. В отсутствие общей модели параллельных вычислений современные технологии программирования не могут обеспечивать программную совместимость всего разнообразия многопроцессорных архитектур и распределённых вычислительных сред. Это объясняет тот факт, что, несмотря на растущую активность распространения многоядерных архитектур, доля практического применения высокопараллельного компьютеринга относительно всей КС до сих пор остаётся незначительной.

Для продвижения практически значимых объединяющих моделей требуются такие способы организации и управления распределёнными и параллельными вычислениями в средах сверхвысокой размерности, у которых собственная системная сложность слабо зависит от размеров КС.

### 3. Актуальность исследований

Компьютерная среда (КС), составленная из миллиардов стационарных и мобильных компьютерных устройств (ПК, смартфоны, встраиваемые компьютерные устройства, рабочие станции, суперкомпьютеры и др.), связанных глобальными сетями, стала носителем глобального сильносвязного информационного пространства, свойства которого в предельном случае выражается формулой " всё влияет на всё, а также всё зависит от всего и сразу".

В глобальном сильносвязном пространстве открываются принципиально новые и практически неограниченные возможности системной интеграции распределённых данных, программ, объектов, ресурсов, процессов, сервисов и субъектов. В нем раскрывается и структурируется массовая информационная активность общества, устанавливаются прямые и обратные каналы влияния всех и каждого на разнообразные сферы функционирования социумов. Возникают и актуализируются новые классы задач согласованной переработки глобально распределённой информации [10-14], которые требуют формирования вычислительных сред со сколь угодно большими количествами связанных вычислительных узлов. Такие задачи выдвигают качественно новые требования к системным свойствам КС в части интеграции, масштабируемости и сквозной программируемости вычислительных ресурсов глобальных сетей.

КС концентрирует и накапливает все большую часть информации, сопровождающей функционирование и развитие большого количества малых и больших систем различного назначения. По мере роста и проникновения КС в разные сферы жизнедеятельности вся совокупность глобально распределённой компьютерной информации (данных и программ) глобальных сетей с всё большей полнотой отражает текущее состояние социосистемы в целом и её частей. Компьютерная информация является формализуемым объектом, поэтому потенциально доступна для алгоритмической переработки во всей гамме сетевых вычислительных ресурсов в целях эффективного управления устойчивым развитием социосистем в условиях глобальной сильносвязности информационного пространства [1].

В дальнейшем прогресс социосистем становится напрямую зависимым от системных, функциональных и вычислительных возможностей КС. В сферы её прямого действия попадают не только отдельные системы технического, экономического, организационного, социального, экологического или других профилей, но и всевозможные системы из таких систем. Поэтому проблемы совершенствования системообразующих возмож-

ностей КС, направленные на эффективное использование её совокупного вычислительного потенциала относятся к наиболее актуальным.

С учётом качественно новых требований к КС актуальность исследований в области распределённых и параллельных вычислений и соответствующих вычислительных систем резко возрастает. Тому множество причин, из которых выделим следующие:

- кризис массового компьютеростроения в связи со структурным насыщением одноядерных микропроцессорных архитектур [15], в основе которых лежит классическая модель последовательных вычислений Дж. фон Неймана;
- кардинальное повышение требований к производительности суперкомпьютеров (достижение суперкомпьютерной производительности 1 Эксафлопс и более [16,17]);
- формирование КС с миллиардами вычислительных ядер (от смартфонов до суперкомпьютеров), системообразующий и вычислительный потенциал которой необходимо направить на решение глобально связанных задач, требующих глубокой и полномасштабной интеграции распределённых и многопроцессорных систем с количеством вычислительных узлов  $\sim 10^8$ - $10^{10}$  и более [5,18].

В 00-е годы ситуация с перспективами массового производства компьютеров и программ кардинально изменилась. Системообразующий потенциал классической модели Дж. фон Неймана, лежащей в основе микропроцессорной революции, оказался исчерпанным. Дальнейший рост производительности массово производимых компьютерных устройств в рамках модели последовательных вычислений стал невозможным по следующим причинам:

- структурное насыщение микропроцессорных архитектур (все архитектурные методы наращивания в рамках модели последовательных вычислений скрытого от программистов параллелизма использованы) [15].
- невозможность наращивания рабочей частоты ввиду достижения теплового барьера (на частотах свыше 4 ГГц тепловыделение превышает возможности воздушного охлаждения).

Единственным путём увеличения производительности массовых компьютерных устройств остаётся увеличение явного параллелизма посредством однокристалльных многопроцессорных архитектур. Такие архитектуры должны обеспечивать рост производительности пропорционально количеству сильносвязанных ядер, при том, что рост числа ядер обеспечивается за счёт прогресса СБИС-технологий, которые продолжают увеличивать количество транзисторов на кристалле в соответствии с законом Мура (удвоение каждые 1.8-2 года) [19].

Другая причина роста актуальности параллелизма – кардинальное повышение требований к производительности суперкомпьютеров. Она связана с освоением полного петафлопсного диапазона и выходом на эксафлопсный [16,17]. Вследствие энергетических барьеров на пути наращивания производительности за счёт увеличения рабочих частот, главным методом достижения поставленной цели остаётся кардинальное увеличение размерности многопроцессорных вычислительных сред. В настоящее время количество параллельно работающих ядер в суперкомпьютерах производительностью 1-20 Пфлопс составляет  $\sim 10^4$ - $10^5$ . Для достижения 1 Эфлопс и более требуются вычислительные среды с количеством узлов более  $10^8$ - $10^9$ , т.е. масштабы практически реализуемого параллелизма многопроцессорных сред должны увеличиться на 3-4 порядка.

Таким образом, параллелизм становится одним из главных резервов развития компьютеростроения, причём уже не только рекордного, но массового.

Индустриальным фундаментом реализации массового параллелизма становятся СБИС-технологии глубокого нанометрового диапазона 40-30-20-10-5нм, которые позволяют делать новую элементную базу в виде однокристалльных многопроцессорных компьютеров с количеством сильносвязных ядер от 10К до 16К и более [16,17,20]. С применением такой элементной базы для достижения производительности 1 Эфлопс потребуется  $\sim 10^5$  многопроцессорных чипов, что достижимо в рамках известных методов конструирования суперкомпьютеров.

Сверхбыстрый прогресс полупроводниковых технологий позволяет наращивать массовый многоядерный параллелизм в двух противоположных направлениях:

- повышение пространственной плотности параллелизма (вычислительный параллелизм с компактной реализацией; конструктивная база – СБИС, платы, модули, шкафы);
- расширение покрываемого пространства (параллелизм управления распределёнными системами стационарных и мобильных объектов; конструктивная база – локальные и глобальные сети).

До сих пор эти направления развивались практически независимо. Разработки и исследования моделей распределённых и параллельных вычислений и систем развивались как отдельные, мало пересекающиеся направления, со своими объектами приложения [6-9]. В отсутствие единой модели распределённых и параллельных вычислений в каждом из них сосуществуют многие разнородные подходы с преобладанием частных решений. Это во многом объясняется большими техническими различиями всего разнообразия существующих многопроцессорных и сетевых сред, а также принципов организации вычислений в них.

Однако новые требования к массовости параллелизма вычислительных сред актуализируют поиски общих методов и универсальных моделей, объединяющих способы рассмотрения и решения задач организации массовых параллельных и распределённых вычислений. Наряду с теоретическими аспектами массового параллелизма, они должны включать в себя как высокоэффективные архитектуры [20], так и принципы их программирования, отвечающие требованиям индустриального производства программ для систем с массовым параллелизмом [21].

Для формирования глобального алгоритмического пространства необходима единая модель распределённых и параллельных вычислений.

В данной работе предлагается автоматная модель управления параллельными и распределёнными процессами, которая позволяет формализовать и с единых позиций рассматривать управление распределёнными и параллельными вычислительными процессами в режимах динамического распараллеливания на основе принципов Data Flow [22-24]. Особенность модели – слабая зависимость вычислительной сложности алгоритма управления от размерности задач, определяемой общим количеством вычислительных операций.

#### 4. Постановка задачи

Решаемая задача: разработка общей модели управления параллельными и распределёнными процессами, вычислительная сложность которой не зависит от размеров задачи и вычислительной среды. Управление состоит в динамическом формировании последовательности из максимально возможных множеств параллельно исполняемых операций.

Предполагается, что исходная задача задаётся в виде двудольного информационного графа, в котором направленные дуги связывают вершины двух разных типов – операторные и объектные. Операторные вершины представляют вычислительные действия, объектные – переменные, которые являются аргументами или значениями операторных. Подобная форма представление задач относится к непроцедурным и применяется в моделях Data Flow [22-24].

В данной модели вопросы динамического управления памятью данных, которое обеспечивает повторное использование освобождающихся областей памяти, не рассматриваются.

## 5. Модель

Модель имеет два уровня. Первый – представление вычислительных задач в виде информационных графов. Второй – автоматное представление структуры управления вычислениями с динамическим распараллеливанием.

### 5.1. Представление задач в виде информационных графов

Информационный граф – двудольный ориентированный граф  $G = \langle \mathcal{A} \cup \mathcal{U}, S \rangle$ , где  $\mathcal{A} = \{a_1, a_2, \dots\}$  – множество операторных вершин,  $\mathcal{U} = \{u_1, u_2, \dots\}$  – множество объектных вершин,  $S = S(\mathcal{A}, \mathcal{U}) \cup S(\mathcal{U}, \mathcal{A})$  – множество ориентированных дуг:  $S(\mathcal{A}, \mathcal{U})$  из операторных вершин к объектным,  $S(\mathcal{U}, \mathcal{A})$  – из объектных к операторным.

Операторным вершинам  $\forall a \in \mathcal{A}$  и объектным  $\forall u \in \mathcal{U}$  ставится в соответствие типы вершин:  $b(a) \in \mathcal{B}$  и  $d(u) \in \mathcal{D}$ , соответственно, где  $\mathcal{B} = \{b_1, b_2, \dots, b_m\}$  – множество базисных операторов (операций),  $\mathcal{D} = \{d_1, d_2, \dots, d_n\}$  – множество базисных типов данных. Дуги  $S(\mathcal{U}, \mathcal{A})$  определяют вхождение переменных в качестве аргументов, дуги  $S(\mathcal{A}, \mathcal{U})$  – передачу вычисленных значений (выходных данных) каждого оператора.

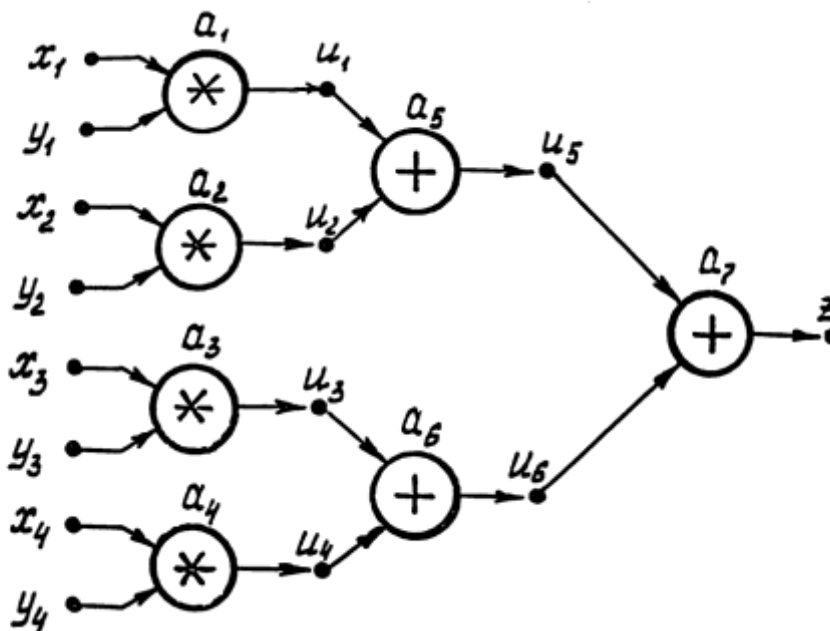


Рис.1. Скалярное произведение векторов размером 4.

Информационные графы взаимно однозначно соответствуют системе формульных выражений в явном виде, в которых объектным вершинам ставятся в соответствие уникальные имена переменных, а также типы данных, а операторным – уникальные идентификаторы операторов, а также типы операций.

На рис. 1 приведен информационный граф для скалярного произведения векторов  $\mathbf{x}=(x_1, x_2, \dots)$  и  $\mathbf{y}=(y_1, y_2, \dots)$ :  $z=\sum_i x_i * y_i$

Информационный граф рис.1 строится по исходной системе формульных выражений в явном виде:

$$(1) \quad \begin{cases} z = u_5 + u_6; \\ u_5 = u_1 + u_2; \\ u_6 = u_3 + u_4; \\ u_1 = x_1 * y_1; \\ u_2 = x_2 * y_2; \\ u_3 = x_3 * y_3; \\ u_4 = x_4 * y_4; \end{cases}$$

## 5.2. Управляющие автоматы [23]

Порядок выполнения действий в информационном графе определяется правилом: каждая операторная вершина может выполняться, если введены или вычислены значения всех её аргументов (входных переменных), а также для вычисленных значений имеются свободные ячейки памяти. Автоматная модель управления реализует это правило.

Введём дополнительные обозначения.

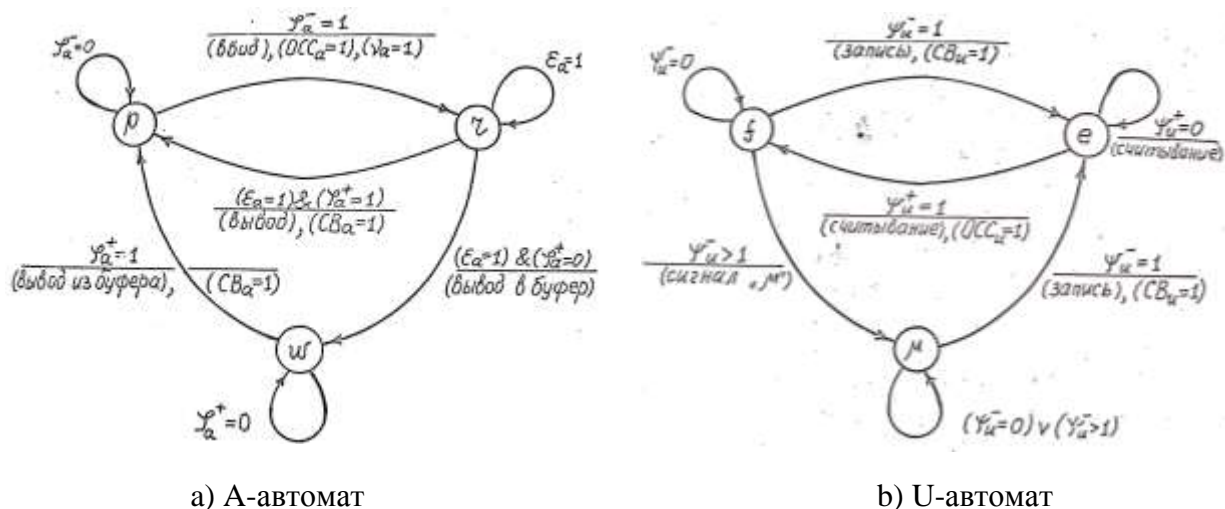
Для  $\forall a \in \mathcal{A}$  ставится в соответствие два кортежа объектных вершин (переменных):

$\mathcal{U}^-(a) = \{u \in \mathcal{U} : \exists S(u, a)\}$  – входной кортеж и  $\mathcal{U}^+(a) = \{u \in \mathcal{U} : \exists S(a, u)\}$  – выходной.

Для  $\forall u \in \mathcal{U}$  ставится в соответствие два кортежа операторных вершин (операций):

$\mathcal{A}^-(u) = \{a \in \mathcal{A} : \exists S(a, u)\}$  – входной кортеж и  $\mathcal{A}^+(u) = \{a \in \mathcal{A} : \exists S(u, a)\}$  – выходной.

Каждой операторной вершине и объектной вершине поставим в соответствие А-автомат (рис.2 а) и U-автомат (рис.2 б), соответственно.





Управляющие автоматы определяют поведение операторных и объектных вершин в ходе исполнения вычислительного процесса.

Автоматы соединены между собой. Каждой дуге информационного графа соответствуют две встречно направленные связи между автоматами. Оба типа автомата в простейшем случае имеют по три состояния.

А-автоматы реализуют управление потоками данных: оператор может быть выполнен, если определены значения его входных переменных. Алфавит состояний А-автомата:  $\{p, r, w\}$ , где  $p$  – оператор пассивен (не все входные значения получены),  $r$  – выполняется,  $w$  – ждёт разгрузки результата.

U-автоматы реализуют принцип защиты значения объекта: новые данные не могут быть записаны в объект до момента полного использования предыдущих значений. Алфавит состояний U-автомата:  $\{f, e, \mu\}$ , где  $f$  – объектная вершина свободна,  $e$  – занята (хранит значение),  $\mu$  – состояние неопределённости из-за конфликта по записи от двух или более операторных вершин.

Переходы автоматов из одного состояния в другое значениями индикаторных функций  $\varphi^-()$ ,  $\varphi^+()$ ,  $\psi^-()$ ,  $\psi^+()$ . В моменты переходов автоматы вырабатывают внутренние управляющие сигналы, которые принимают значения из  $\{0,1\}$ . Они передаются встречно между соседними автоматами по направлению дуг информационного графа. Прямые сигналы (по направлению совпадают с ориентацией дуг) называются сигналами возбуждения (СВ). Обратные – называются сигналами обратного согласования (ОСС). Каждый автомат имеет накопитель СВ и ОСС. На них определяются индикаторные функции. Внутренние сигналы синхронизируют передачу потоков данных в информационном процессе.

Каждый А-автомат помимо внутренних сигналов вырабатывает сигнал запуска своего оператора на выполнение ( $v=1$ ) и принимает от него обратный сигнал завершения работы ( $\varepsilon=1$ ), см. рис. 2а. Это внешние относительно автоматов сигналы связи с информационным графом. Они связывают информационный процесс в графе с управляющим в автоматной сети.

В ходе исполнения вычислительного процесса автоматная сеть обеспечивает автоматическое асинхронное распараллеливание в соответствии с правилом "операторная вершина готова к обработке, как только её аргументы получают значение". Такое правило лежи в основе моделей Data Flow [22, 24] и, как известно, обеспечивает выявление максимально возможного параллелизма.

Важно отметить, что в пространстве сильносвязных алгоритмических взаимодействий СВ определяют прямое направление влияния в соответствии с формулой "всё влияет на всё и сразу". Сигналы ОСС определяют встречное направления в соответствии с формулой "всё зависит от всего и сразу". Автоматная сеть посредством анализа цепочек передачи СВ позволяет рассматривать и формализовывать задачи формирования пространства "угроз", а посредством ОСС – способов противодействия угрозам, в основе которых лежит выявление признаков, которые являются первопричинами угроз (начальные звенья цепочек в нарастании угроз).

### 5.3. Балансное уравнение вычислительного процесса

С помощью автоматной модели носителя управляющего процесса построим аналитическое выражение механизма управления потоками данных.

Множества операторных  $\mathcal{A}$  и объектных  $\mathcal{U}$  вершин на каждом шаге  $j$  с учётом текущего состояния соответствующих автоматов могут быть тождественно представлены в виде разбиений на непересекающиеся подмножества вершин с одинаковым состоянием:

$$(2) \quad \begin{cases} \mathcal{U} \equiv \mathcal{U}^f(j) \cup \mathcal{U}^e(j) \cup \mathcal{U}^\mu(j), \\ \mathcal{A} \equiv \mathcal{A}^p(j) \cup \mathcal{A}^r(j) \cup \mathcal{A}^w(j), \\ j=0,1,2,\dots \end{cases}$$

Отметим, что множество  $\mathcal{A}^r(j)$  на каждом шаге  $j$  определяет множество параллельно исполняемых операторных вершин.

С учётом взаимодействия вершин информационного графа новый состав каждого из множеств (2) с одинаковыми состояниями на каждом шаге определяется следующей системой рекуррентных выражений:

$$(3) \quad \begin{cases} \mathcal{A}^p(j+1) = (\mathcal{A}^p(j) \cup \Delta\mathcal{A}^{wp}(j) \cup \Delta\mathcal{A}^{rp}(j)) \setminus \Delta\mathcal{A}^{pr}(j); \\ \mathcal{A}^r(j+1) = (\mathcal{A}^r(j) \cup \Delta\mathcal{A}^{wp}(j)) \setminus (\Delta\mathcal{A}^{rp}(j) \cup \Delta\mathcal{A}^{rw}(j)); \\ \mathcal{A}^w(j+1) = (\mathcal{A}^w(j) \cup \Delta\mathcal{A}^{rw}(j)) \setminus \Delta\mathcal{A}^{wp}(j); \\ \mathcal{U}^f(j+1) = (\mathcal{U}^f(j) \cup \Delta\mathcal{U}^{ef}(j)) \setminus (\Delta\mathcal{U}^{fe}(j) \cup \Delta\mathcal{U}^{f\mu}(j)); \\ \mathcal{U}^e(j+1) = (\mathcal{U}^e(j) \cup \Delta\mathcal{U}^{fe}(j) \cup \Delta\mathcal{U}^{\mu e}(j)) \setminus (\Delta\mathcal{U}^{ef}(j)); \\ \mathcal{U}^\mu(j+1) = (\mathcal{U}^\mu(j) \cup \Delta\mathcal{U}^{f\mu}(j)) \setminus \Delta\mathcal{U}^{\mu e}(j); \\ j=0,1,2,\dots \end{cases}$$

Состав множеств в левой части (3) определяется как баланс переходных потоков множеств операторов и объектов правой части, которые обозначены  $\Delta\mathcal{A}^{qs}$  и  $\Delta\mathcal{U}^{qs}$ , соответственно. Их автоматы на  $j$  такте переходят из состояния  $q$  в  $s$ , где  $q,s \in \{p,r,w\}$  либо  $q,s \in \{f,e,\mu\}$ .

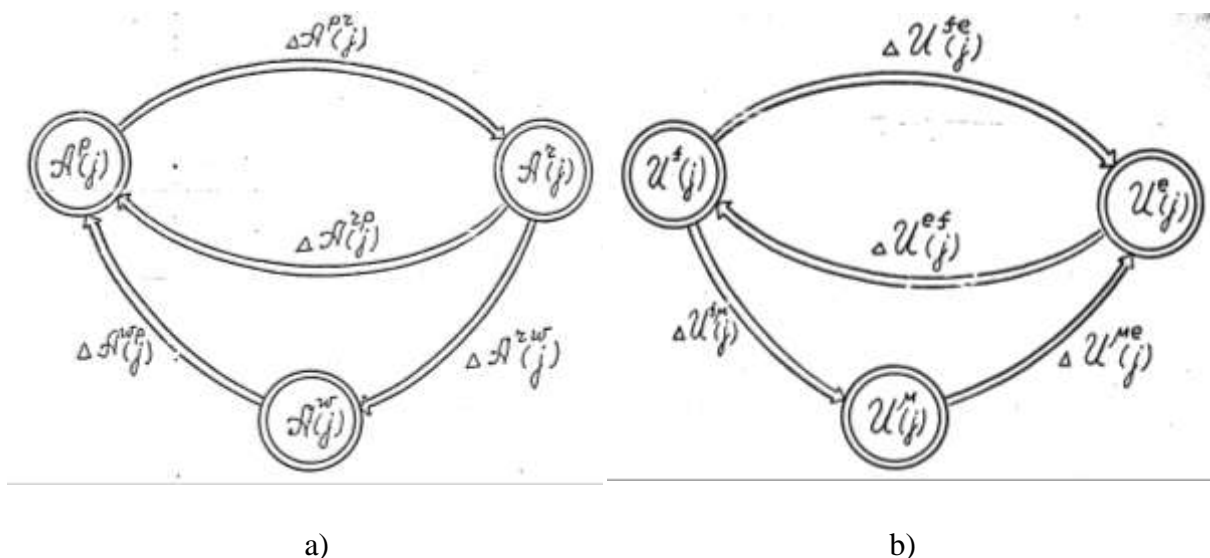


Рис. 3.

На рис. 3 показана диаграмма балансного уравнения. Важное свойство данной модели – состояние процесса на каждом шаге полностью определяется составом множеств вершин с одинаковыми состояниями. Количество таких множеств не связано с количеством вершин в информационном графе задачи и зависит только от сложности автоматов, определяющих состояние вершин в ходе выполнения процесса. Представление параллельных и распределённых вычислительных процессов посредством балансного уравнения позволяет существенно уменьшить размерность пространства состояний процесса по сравнению с числом состояний собственно автоматной сети.

Сложность каждого типа автоматов определяется количеством его различных состояний. В данном рассмотрении – два типа автоматов, алфавит состояний в каждом из них (рис.2) состоит из трёх символов, что даёт диаграмму балансного уравнения с 6 вершинами (по три для каждого из двух типов автоматов).

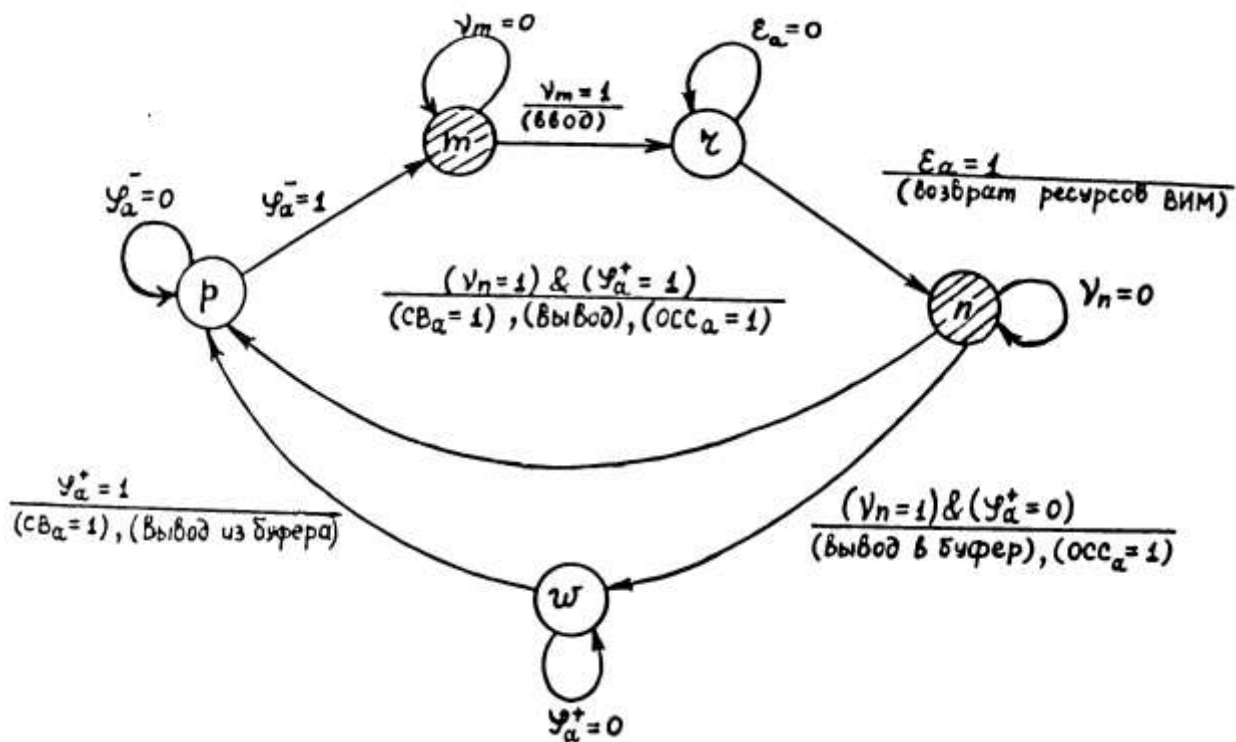


Рис. 4. Автомат текущего состояния операторных вершин с учётом ограниченности вычислительных (вершина  $m$ ) и управляющих (вершина  $n$ ) ресурсов

Очевидно, что предложенный метод обобщается на случаи более сложного поведения вершин графов, что обеспечивается введением автоматов с большим числом состояний. На рис. 4 приведён пример А- автомата для случая с ограниченными вычислительными ресурсами, когда количество готовых к выполнению виртуальных информационных машин (ВИМ) операторов может превышать имеющиеся физические ресурсы, что требует организации эффективной очередности исполнения, которая обеспечит лучшее их использование.

Одним из стратегически важных направлений в исследовании проблем компьютерной глобализации парадигмы управления [1] является формирование в ресурсах глобальной компьютерной среды математически однородного алгоритмического простран-

ства распределённых и параллельных вычислений. Цель исследований – постановка и решение задач сетцентрического управления направленных на выработку общих подходов к обеспечению устойчивого развития мировой социосистемы в сильносвязном пространстве "всё зависит от всего и сразу" [1,4, 5].

Предложенный метод формализации процессов управления на основе балансных уравнений вычислительными процессами благодаря независимости своей сложности от размерности графов может использоваться для решения задач моделирования и управления большими и сверхбольшими распределёнными системами, в том числе и для формализации социосистемы как целостной динамической системы [1]. На рис.5 социосистема представлена посредством балансного уравнения, представляющего взаимодействие двух информационных сред – социальной и компьютерной.

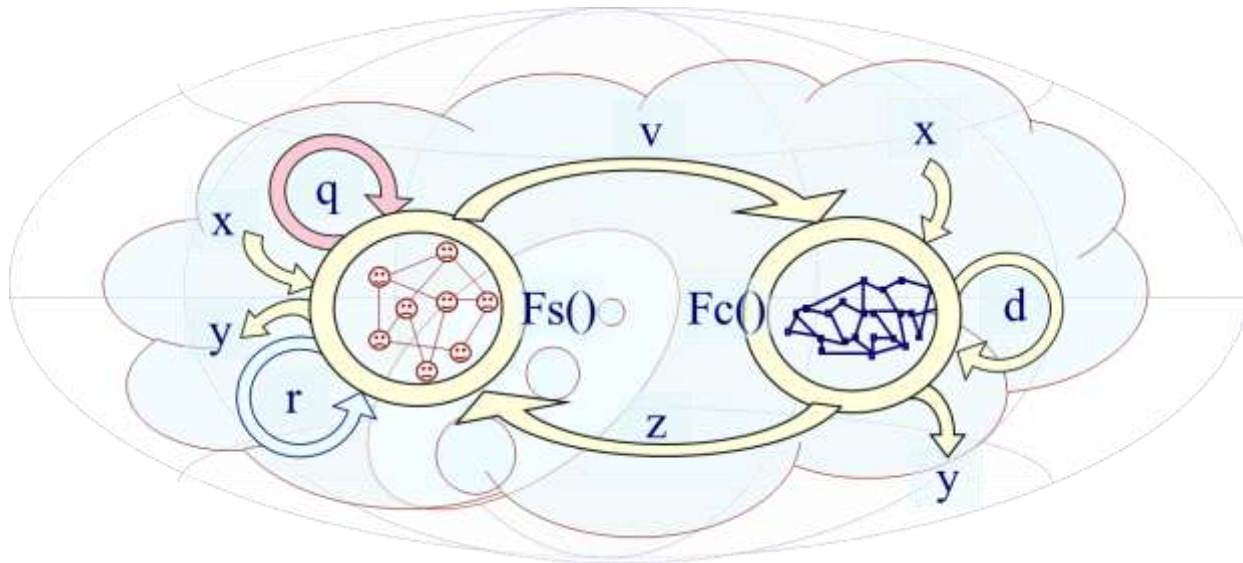


Рис. 5. Балансное представление взаимодействия двух информационных сред – социальной и компьютерной

Соответствующее аналитическое представление имеет вид [1]:

$$\left\{ \begin{array}{l} t_0 = 0; \\ q[t_0] = q_0, r[t_0] = r_0, d[t_0] = d_0, v[t_0] = v_0, z[t_0] = z_0; \\ (y[t_i], v[t_i], q[t_i], r[t_i]) = Fs(q[t_{i-1}], r[t_{i-1}], z[t_{i-1}], x[t_i]); \\ (y[t_i], z[t_i], d[t_i]) = Fc(d[t_{i-1}], v[t_{i-1}], x[t_i]); \\ i = 1, 2, \dots, \end{array} \right.$$

где  $t_i$  – разномасштабное структурированное время,

$X_i, Y_i, Q_i, R_i, D_i, V_i, Z_i$  – множества допустимых значений,

$Fs()$  – метафункция социальной среды,

$Fc()$  – метафункция компьютерной среды.

### Выводы

Новизна модели определяется применением метода балансировки потоков применительно к изменяющимся состояниям основных компонентов параллельных и распределённых вычислительных процессов, что позволяет динамически выявлять максимально возможное распараллеливание. Свойством независимости сложности модели от размера

задач открывает возможности её применения в параллельных и распределённых вычислительных средах с массовым параллелизмом.

#### СПИСОК ЛИТЕРАТУРЫ

1. *Затуливетер Ю.С.* Проблемы глобализации парадигмы управления в математически однородном поле компьютерной информации // Проблемы управления. 2005. -Ч.1: №1, С.1-12, URL: [http://zvt.hotbox.ru/Zatuliveter\\_comp\\_glob\\_contr\\_1.htm](http://zvt.hotbox.ru/Zatuliveter_comp_glob_contr_1.htm). -Ч.II: №2, С.13-23, URL: [http://zvt.hotbox.ru/Zatuliveter\\_comp\\_glob\\_contr\\_2.htm](http://zvt.hotbox.ru/Zatuliveter_comp_glob_contr_2.htm).
2. *Затуливетер Ю.С.* Информационная природа социальных перемен. Серия "Информация и Социум", -М.: СИНТЕГ, 2001. –132с. ISBN 5-89638-046-1. URL: [http://www.ipu.ru/sites/default/files/publications/Zatuliveter\\_InfPrirodaSocPeremen\\_.pdf](http://www.ipu.ru/sites/default/files/publications/Zatuliveter_InfPrirodaSocPeremen_.pdf).
3. *Затуливетер Ю.С.* На пути к глобальному программированию // Открытые системы. 2003. № 3. – С. 46-47. -URL: <http://www.osp.ru/os/2003/03/182704/>.
4. *Затуливетер Ю.С.* Компьютерный базис сетецентрического управления // Российская конференция с международным участием "Технические и программные средства в системе управления, контроля и измерения" (УКИ'10). Труды конференции. Москва, 18-20 октября 2010 г. Учреждение Российской Академии наук Институт проблем управления им. В.А. Трапезникова РАН. С.17-37. -URL: <http://cmm.ipu.ru/proc/Затуливетер%20Ю.С.%20.pdf>.
5. *Затуливетер Ю.С.* EхаScale: на пути к единому пространству распределенных и параллельных вычислений // Научный сервис в сети Интернет: Эксафлопсное будущее: Труды Международной суперкомпьютерной конференции (20-25 сентября 2010г., г. Новороссийск). – М.: Изд-во МГУ, 2011. С.10-14. –URL: <http://agora.guru.ru/abrau2011/pdf/10.pdf>.
6. *Таненбаум Э., М. ван Стеен.* Распределенные системы. Принципы и парадигмы. Издательство: СПб.: Питер, 2003. -877с.
7. *Воеводин В.В., Воеводин Вл.В.* Параллельные вычисления. – СПб.: БХВ-Петербург, 2002. . – 608 с.
8. *Топорков В.В.* Модели распределенных вычислений. Физматлит. 2004. -320 с.
9. *Гергель В.П.* Теория и практика параллельных вычислений. URL: <http://www.intuit.ru/department/calculate/paralltp/>
10. *Weiser Mark.* The Computer for the 21st Century // Scientific American Special Issue on Communications, Computers, and Networks, September, 1991. -URL: <http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html>.
11. *Кротов В.* Интернет вещей // Computerworld. - 2008. - №3. -URL: <http://www.osp.ru/cw/2008/03/4755084>.
12. *Кузьмин И.* Future Combat System – революция или эволюция? -URL: [http://www.3dnews.ru/editorial/future\\_combat\\_system](http://www.3dnews.ru/editorial/future_combat_system) .
13. *Кобец Б.Б., Волкова И.О., Окороков В.Р.* SMART GRID как концепция инновационного развития электроэнергетики за рубежом. -URL: <http://www.transform.ru/articles/html/10it/it000018.article>
14. A Smarter Planet. -URL: <http://www.ibm.com/smarterplanet/us/en/>.
15. *Затуливетер Ю.С.* Компьютерные архитектуры: неожиданные повороты. HARD'n'SOFT. Компьютерный журнал для пользователей, N2, 1996, с. 89-94. URL: [http://zvt.hotbox.ru/p2\\_z1.htm](http://zvt.hotbox.ru/p2_z1.htm).
16. *Harrod Bill.* Future Processing: ExtremeScale. DARPA. September 17, 2009. –URL: [http://www.darpa.mil/tcto/docs/Harrod\\_ACS\\_Brief\\_091709.pdf](http://www.darpa.mil/tcto/docs/Harrod_ACS_Brief_091709.pdf).

17. *Воеводин Вл.В.* Суперкомпьютерный комплекс Московского университета // IV международная конференция Распределенные вычисления и Грид-технологии в науке и образовании. –URL: [http://grid2010.jinr.ru/files/pdf/VVV\\_Dubna\\_2010.pdf](http://grid2010.jinr.ru/files/pdf/VVV_Dubna_2010.pdf).
18. *Амелин К.С., Граничин О.Н., Кияев В.И.* Мобильность или Супервычисления: кто кого? // Суперкомпьютеры, №4, 2010, с.30-33.
19. Прогноз Intel – переход к 5-нм технологиям позволит закону Гордона Мура продержаться еще минимум десятилетие. URL: <http://habrahabr.ru/post/151765/>, <http://dvice.com/archives/2012/09/intel-5nm-proce.php>.
20. *Артамонов С.Е., Затуливетер Ю.С., Фищенко Е.А.* Предпосылки к созданию однокристалльного многопроцессорного компьютера ПС-2000М производительностью 1-10 Tflops // Параллельные вычислительные технологии (ПаВТ'2011): труды международной научной конференции (Москва, 28 марта – 1 апреля 2011 г.). С. 402–410. URL: [http://omega.sp.susu.ac.ru/books/conference/PaVT2011/talks/012\\_zatuliveter\\_talk.pdf](http://omega.sp.susu.ac.ru/books/conference/PaVT2011/talks/012_zatuliveter_talk.pdf), Челябинск: Издательский центр ЮУрГУ, 2011. – 730 с. – URL: <http://omega.sp.susu.ac.ru/books/conference/PaVT2011>. ISBN 978-5-696-04090-5. 5
21. *Затуливетер Ю.С.* Введение в проблему параметризованного синтеза программ для параллельных компьютеров / (Препринт/Институт проблем управления). -М., 1993, 88с.
22. *Dennis J.* Data Flow Supercomputers // Computer. - Vol.13. - No.11. Nov, 1980. P.48-56.
23. *Затуливетер Ю.С.* Алгоритм согласованных асинхронных вычислений. В кн.: Однородные вычислительные структуры и малые ЭВМ: Тез. докл. на Всесоюзном семинаре, г. Звенигород, 1979, с.112-116 (Институт проблем управления, Москва).
24. *Затуливетер Ю.С.* Разработка и исследование методов организации потоковых вычислений в многопроцессорных системах с общим управлением. Диссертация на соискание ученой степени к.т.н., -М., Институт проблем управления, 1984, 190 с.