

## ОРГАНИЗАЦИЯ НЕЧЕТКОГО ЛОГИЧЕСКОГО ВЫВОДА НА ОСНОВЕ МНОГОУРОВНЕВОГО ПАРАЛЛЕЛИЗМА

Р.Н. ПОНОМАРЕНКО

**Аннотация.** Разработан, теоретически обоснован и реализован метод построения иерархических систем нечеткого логического вывода на основе многоуровневого параллелизма, в частности параллелизма второго уровня. Сформулирована концепция многоуровневого параллелизма, введено понятие уровня параллелизма. Сформулирована и доказана теорема, разработан метод теоретической оценки максимально возможного ускорения для систем, построенных на базе параллелизма уровня  $n$ , а также подход к проектированию иерархических нечетких систем на основе многоуровневого параллелизма для графических ускорителей NVIDIA. Спроектирована экспериментальная программная система для иерархических нечетких систем на основе многоуровневого параллелизма и технологии CUDA, имеющих сложные графы зависимостей между блоками нечетких правил. На основе разработанного метода вычислены теоретические оценки максимального ускорения; получены экспериментальные оценки ускорения для иерархических сложных систем нечеткого логического вывода.

**Ключевые слова:** нечеткий логический вывод, многоуровневый параллелизм, ускорение, нечеткие системы типа Такаги–Сугено, теоретическая оценка ускорения.

### ВСТУПЛЕНИЕ

Математический аппарат нечеткой логики [1] зарекомендовал себя в тех системах, где существует неопределенность в той или иной степени, и применяется все шире в ряде направлений: наука, медицина, принятие решений в условиях неопределенности и т. д. В основе нечетких систем лежит нечеткий логический вывод, который позволяет применять нечеткую логику для построения интеллектуальных и экспертных систем, существенно расширяя круг решаемых задач.

При проектировании масштабных систем нечеткого логического вывода главной проблемой является необходимость существенного расширения нечеткой базы знаний, возникающая при увеличении количества входящих посылок (входных переменных). При увеличении количества входных переменных в рамках элементарной нечеткой системы количество правил растет экспоненциально [2]. Количество нечетких правил в пределах элементарной системы нечеткого логического вывода, которое должен написать эксперт [2]

$$N = M^K,$$

где  $K$  — количество нечетких входных переменных элементарной нечеткой системы;  $M$  — количество термов для каждой нечеткой переменной.

Единственным эффективным способом борьбы с количеством правил является декомпозиция на уровне входных переменных. Такие системы называются иерархическими системами нечеткого логического вывода, где в качестве посылок одних систем могут выступать заключения других систем, нижележащих по иерархии [3–6]. Таким образом, с увеличением количества элементарных систем суммарное количество правил при масштабировании иерархической системы увеличивается в арифметической прогрессии, давая возможность строить нечеткие системы с неограниченным количеством входных переменных. Так возник вопрос о повышении производительности вычислений иерархических систем нечеткого логического вывода.

**Цель данной работы** — предоставить теоретическую базу и способы реализации иерархических систем нечеткого логического вывода на основе концепции многоуровневого параллелизма с целью повышения быстродействия таких систем в масштабируемых задачах, имеющих сложные иерархические зависимости между блоками нечетких правил.

За основу взят нечеткий логический вывод типа Такаги–Сугено, однако не исключается (как и не утверждается) распространение представленной теории на нечеткие системы другого типа, базы знаний которых состоят из блоков нечетких правил.

## НЕЧЕТКИЙ ЛОГИЧЕСКИЙ ВЫВОД ТАКАГИ–СУГЕНО

Система нечеткого логического вывода типа Такаги–Сугено основывается на блоках нечетких предикатных правил ЕСЛИ–ТО [7]:

$$R_j : \text{ЕСЛИ } x_1 \text{ есть } A_{1j} \text{ И } x_2 \text{ есть } A_{2j} \text{ И } \dots \text{ И } x_n \text{ есть } A_{nj} \\ \text{ТО } y_j = g_j(x_1, x_2, \dots, x_n), \quad j = 1, 2, \dots, N,$$

где функция нахождения подзаключения правила  $g_j(x_1, x_2, \dots, x_n) = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \dots + \omega_n x_n$  — линейная функциональная зависимость от четких величин входных посылок нечеткой системы;  $N$  — общее количество правил в пределах блока. Вычисление общего заключения нечеткой системы типа Такаги–Сугено имеет вид:

$$y = \frac{\sum_{j=1}^N g_j \prod_{i=1}^{m_j} \mu_{ij}(x_i)}{\sum_{j=1}^N \prod_{i=1}^{m_j} \mu_{ij}(x_i)}, \quad (1)$$

где  $\mu_{ij}(x_i)$  — функция принадлежности входных посылок нечеткому терму;  $T$  —  $T$ -норма, в которой в качестве конъюнкции обычно используется операция нахождения минимума.

## КОНЦЕПЦИЯ МНОГОУРОВНЕВОГО ПАРАЛЛЕЛИЗМА

Любой параллельный алгоритм можно представить в виде ярусно-параллельной вычислительной формы [8]. То есть представить в виде дерева

операций, вершины которого означают множество операций, а дуги – информационные зависимости между операциями (рис. 1). Пусть имеется ациклический ориентированный мультиграф  $G = \{V, E\}$ , где  $V$  — множество операций (множество вершин);  $E$  — множество зависимостей между операциями (множество дуг). Приведем понятие яруса [8].

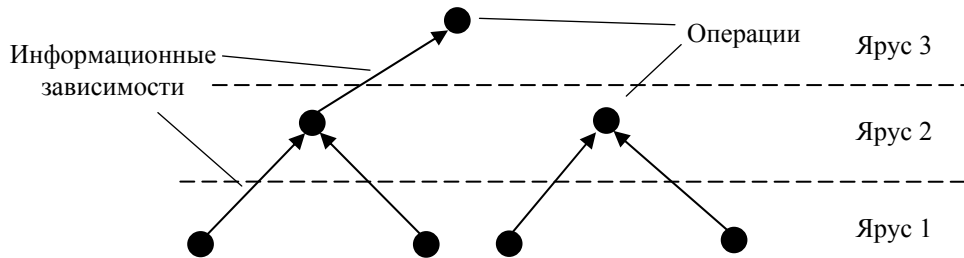


Рис. 1. Ярусно-параллельная форма вычислений

Пусть  $s, q$  — вершины с индексами  $i, j$  соответственно, где  $s$  — операция приемник данных;  $q$  — операция поставщик данных. Тогда, если индексы  $i, j$  номера соответствующих ярусов, то всегда  $i > j$ . В этом случае множество вершин, имеющих одинаковый индекс, является ярусом ярусно-параллельной формы вычислений. Количество систем в пределах одного яруса называется шириной данного яруса. Естественно, чем шире ярусы алгоритма, тем выше его степень параллелизма и, как следствие, возможности для распараллеливания. Степень параллелизма определяет соотношение количества вершин к количеству ярусов алгоритма.

Для обеспечения доступа к вершинам ярусно-параллельной схемы автором разработан метод, позволяющий, используя для хранения графа одномерный массив вершин, получать доступ к любой вершине любого яруса графа. Используемая структура данных состоит из двух одномерных массивов данных: массива вершин и массива номеров вершин.

Представим метод обеспечения доступа к вершинам каждого яруса [9]. Пусть  $N$  — количество вершин графа (количество блоков правил в ярусе),  $s$  — количество ярусов. Тогда, если количество систем в  $i$ -м ярусе  $l_i, i = \overline{1, s}$ , то имеем следующее соотношение:

$$\sum_{i=1}^s l_i = N.$$

Пусть  $V$  — множество вершин;  $M$  — множество номеров вершин графа. Определим для каждого яруса  $i$ -ю группу вершин  $D_i = \{d_p^{(i)}\}, p = \overline{1, l_i}$  на множестве  $V$ . Тогда каждую вершину  $i$ -го яруса на множестве  $V$  можно найти по формуле

$$d_p^{(i)} = M_{z_p^{(i)}}, z_p^{(i)} = p + \sum_{k=1}^i l_k, \quad (2)$$

где  $i$  — номер яруса;  $p$  — номер вершины в  $i$ -м ярусе;  $z_p^{(i)}$  — индекс номера вершины  $p$   $i$ -го яруса на множестве  $M$ ;  $M_{z_p^{(i)}}$  — найденный индекс вершины  $d_p^{(i)}$  на множестве  $V$ .

При этом операции рассматриваются как элементарные, однако операциями чаще всего являются именно блоки операций, примерно равные между собой по количеству атомарных вычислений и контексту логических действий. Концепция многоуровневого параллелизма предполагает рассмотрение вершин графа в качестве сложных операционных блоков. Такие блоки, притом, что имеют подобно элементарной операции один вход – один выход, могут иметь возможности внутреннего распараллеливания. Пример отдельного яруса ярусно-параллельной схемы вычислений, вершины которой рассматриваются как вычислительные блоки и обладают внутренним параллелизмом, показан на рис. 2.

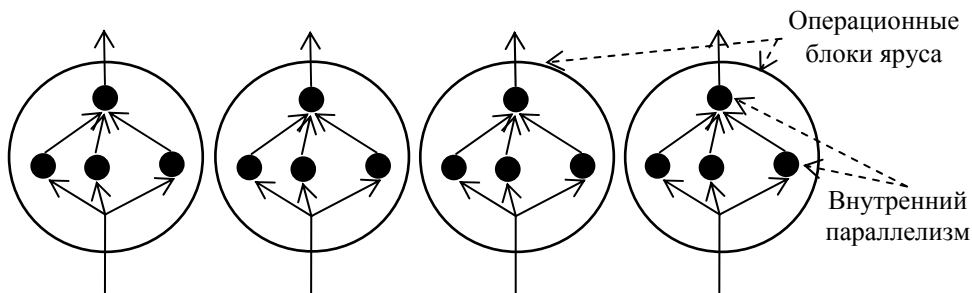


Рис. 2. Пример яруса вычислительной ярусно-параллельной схемы с внутренним параллелизмом

Введем понятие уровня параллелизма. Пусть имеется граф алгоритма  $G = (V, E)$ , где  $V$  — множество операций;  $E$  — множество зависимостей между операциями. В общем виде множество операций  $V = \{v_i\}, i = \overline{1, M}$ , обозначает операции именно элементарные, где  $M$  — количество операций. Однако каждая отдельно взятая операция  $v_i$  также может подразумевать блок операций. То есть можно утверждать, что в этом случае  $v_i = \{Q, W\}$ , где  $Q$  — множество операций блока  $v_i$ ;  $W$  — множество зависимостей операций блока  $v_i$ .

Пусть  $k$  — количество ярусов блока операций  $v_i$ ;  $M$  — количество операций блока  $v_i$ . Тогда можно утверждать, что уровень параллелизма возрастает на единицу, если соблюдаются следующие условия:  $M \geq 2, k < M$ . Иными словами, количество операций блока  $v_i$  должно быть не менее двух, а количество ярусов меньше общего количества вершин множества  $v_i$ . Алгоритм, все операции которого выполняются строго последовательно (последовательный алгоритм), является алгоритмом нулевого уровня параллелизма.

Примем изначально весь алгоритм  $G$  как единый блок операций. Поскольку уровни параллелизма начинают исчисление с нулевого уровня (уровня последовательной программы), то начальный уровень алгоритма  $G$  будет также нулевым. Таким образом, если  $N \geq 2, s < N$ , тогда наблюдается параллелизм первого уровня, а если  $M \geq 2, k < M$ , то параллелизм второго уровня. При дальнейшем исследовании алгоритма количество уровней па-

раллелизма может возрасти до некоего максимального значения. Это значение назовем верхней границей параллелизма и обозначим  $\omega$ .

В работе сформулирована теорема [9] для нахождения теоретической оценки максимально возможного ускорения для вычислительных систем, имеющих параллелизм уровня больше единицы.

**Теорема.** Для параллелизма уровня  $n$ , суммарное максимально возможное ускорение вычисляется как произведение максимально возможных ускорений каждого из уровней:

$$S_P^{\text{sum}} = s_{p_1}^1 s_{p_2}^2 \times \dots \times s_{p_n}^n = \prod_{i=1}^n s_{p_i}^i, \quad (3)$$

где  $S_P^{\text{sum}}$  — суммарное ускорение параллелизма уровня  $n$ ;  $s_{p_i}^i$  — ускорение уровня  $i$  при условии использования на данном уровне количества процессоров равном  $p_i$ .

Следует уточнить, что в формуле (3) присутствует также нулевой уровень  $s_{p_0}^0$ ,  $p_0 = 1$  (последовательный вариант), который в большинстве случаев полагается равным единице и может быть опущен при вычислении теоретической оценки максимального ускорения. Однако теоретически не исключается использование для нулевого уровня значения, отличного от единицы ( $s_{p_0}^0 \geq 1$ ). Таким примером может послужить случай, когда ускорение возможно за счет использования более быстрой памяти. Таким образом, обозначив нулевой уровень, формула (3) примет вид:

$$S_P^{\text{sum}} = s_{p_0}^0 s_{p_1}^1 s_{p_2}^2 \times \dots \times s_{p_n}^n = \prod_{i=0}^n s_{p_i}^i.$$

**Доказательство.** Максимально возможное ускорение в пределах параллелизма первого уровня можно вычислить за законом Амдала [8], который находит теоретическую оценку ускорения с учетом того, какая доля вычислений выполняется строго последовательно:

$$S_P = \frac{1}{\alpha + \frac{1-\alpha}{p}} = \frac{p}{\alpha p + 1 - \alpha}, \quad (4)$$

где  $p$  — количество используемых процессоров;  $\alpha$  — доля вычислений, выполняющихся строго последовательно. При вычислении  $\alpha$  обычно сложную структуру разбивают на вычислительные блоки примерно одинакового размера для нахождения доли блоков, выполняющихся последовательно. При этом, в свою очередь, каждый из блоков может иметь свой внутренний параллелизм, который невозможно достичь между блоками, однако достигается в пределах блока (рис. 2).

Из изложенного следует, что переменная  $\alpha$  будет также иметь коэффициент внутреннего ускорения, который при строго последовательном выполнении операций блока будет равняться единице и не рассматривается. Введем новый коэффициент  $\gamma$  при  $\alpha$ , обозначив его как коэффициент внутреннего ускорения. Тогда формула (4) примет вид

$$S_p = \frac{1}{\frac{\alpha}{\gamma} + \frac{\left(\frac{1-\alpha}{\gamma}\right)}{p}} = \frac{1}{\alpha\gamma^{-1} + \frac{(1-\alpha)\gamma^{-1}}{p}}.$$

Поскольку каждый блок можно рассматривать как отдельную подсистему, то коэффициент  $\gamma$  также будет вычисляться по формуле (4). При этом уровень параллелизма будет увеличен на единицу. Тогда, применяя метод математической индукции для доказательства данной теоремы, получим выражение при параллелизме уровня  $n = 2$  :

$$\begin{aligned} S_p^{\text{sum}} &= \frac{1}{\alpha_1 \left( \frac{1}{\alpha_2 + \frac{1-\alpha_2}{p_2}} \right)^{-1} + \frac{(1-\alpha_1) \left( \frac{1}{\alpha_2 + \frac{1-\alpha_2}{p_2}} \right)^{-1}}{p_1}} = \\ &= \frac{1}{\alpha_1 (S_{p_2}^2)^{-1} + \frac{(1-\alpha_1)(S_{p_2}^2)^{-1}}{p_1}} = \frac{1}{\alpha_1 (S_{p_2}^2)^{-1} p_1 + (1-\alpha_1)(S_{p_2}^2)^{-1}} = \\ &= \frac{p_1}{(\alpha_1 p_1 + (1-\alpha_1))(S_{p_2}^2)^{-1}} = \frac{S_{p_2}^2 p_1}{\alpha_1 p_1 + 1 - \alpha_1} = S_{p_1}^1 S_{p_2}^2 = \prod_{i=1}^n S_{p_i}^i. \end{aligned}$$

Теорема доказана.

## ИЕРАРХИЧЕСКИЕ СИСТЕМЫ НЕЧЕТКОГО ЛОГИЧЕСКОГО ВЫВОДА

Иерархические нечеткие системы строятся на основе ярусно-параллельной схемы вычислений [8] и информационных зависимостей между элементарными нечеткими системами. Под информационной зависимостью имеется в виду зависимость входных параметров одних нечетких систем (приемников данных) от выходных значений других (поставщиков данных).

Представим иерархическую нечеткую систему по аналогии с ярусно-параллельной формой в виде ациклического ориентированного мультиграфа  $G = \{V, E\}$ , где  $V$  — множество элементарных нечетких систем;  $E$  — множество информационных зависимостей между элементарными нечеткими системами. Общая схема иерархических систем нечеткого логического вывода представлена на рис. 3. Прямоугольниками обозначены элементарные нечеткие системы (FS), где верхний индекс — номер яруса; нижний индекс — номер системы в данном ярусе. Типы иерархических структур нечеткого логического вывода приведены в работе [2].

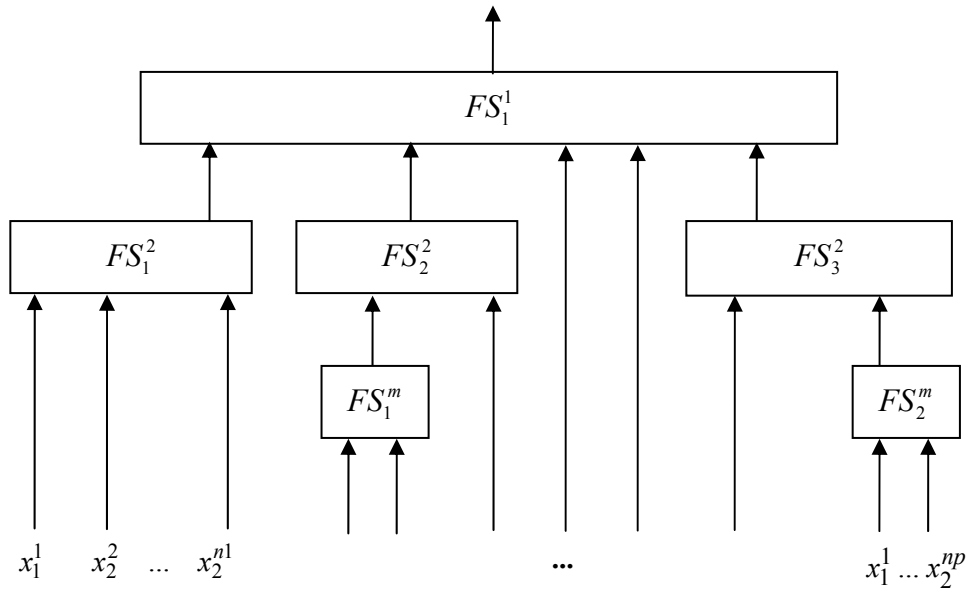


Рис. 3. Схема иерархической системы нечеткого логического вывода

### ПАРАЛЛЕЛИЗМ НЕЧЕТКИХ СИСТЕМ ПЕРВОГО УРОВНЯ

Параллелизм первого уровня в иерархических нечетких системах — это параллелизм на уровне иерархической связки нечетких систем между собой. Под операцией понимается элементарная нечеткая система. Непосредственно параллельное исполнение операций происходит в пределах яруса, т.е. осуществляется последовательная обработка параллельно исполняемых ярусов. Представим всю иерархию элементарных систем множеством  $Y = \{y^d\}$ ,  $d = \overline{1, L}$ , где  $L$  — общее количество нечетких систем. Также представим подмножество множества  $Y$  нечетких систем в пределах яруса в виде  $Q = \{y^q\}$ ,  $q = \overline{1, P}$ , где  $P$  — количество нечетких систем в пределах одного яруса. Тогда в пределах яруса  $Q$  присутствует параллелизм, где доступ к элементарным системам параллельно исполняемого яруса  $Q$  будет происходить в соответствии с (2).

Представим множество потоков для параллелизма первого уровня множеством  $PLI = \{p_t\}$ ,  $t = \overline{1, K}$ , где  $K$  — общее количество доступных потоков первого уровня. Каждый элемент  $p_t$  содержит номера обрабатываемых элементарных нечетких систем. Номера этих систем для каждого потока

$$p_t = \left\{ \begin{array}{l} q, \text{ если } P \leq K \\ t + \sum_r K, r = 0, \dots, \left\lfloor \frac{P}{K} \right\rfloor, \text{ если } P > K \end{array} \right\}. \quad (5)$$

Следовательно, каждый параллельный поток  $p_t$  выполняет вычисления:

$$y^q = \frac{\sum_{j=1}^{N^q} g_j^q w_j^q}{\sum_{j=1}^{N^q} w_j^q},$$

где  $q$  — индекс элементарной нечеткой системы яруса  $Q$ , рассчитанный по формуле (5).

### ПАРАЛЛЕЛИЗМ НЕЧЕТКИХ СИСТЕМ ВТОРОГО УРОВНЯ

Параллелизм второго уровня является ускорителем на уровне элементарной нечеткой системы. Как упоминалось выше, элементарная нечеткая система состоит из блока нечетких предикатных правил  $RB = \{R_j\}, j = \overline{1, N}$ . Нахождение заключения нечеткой системы предполагает последовательную обработку всего множества правил блока. Причем не играет роли последовательность обработки данного множества, каждый элемент которого является независимым. Представим множество потоков для параллелизма второго уровня множеством  $PL2 = \{p_l\}, l = \overline{1, M}$ , где  $M$  — общее доступное количество потоков второго уровня. Каждый элемент  $p_l$  содержит номера обрабатываемых правил. Номера этих правил для каждого потока  $p_l$  определяются согласно формуле:

$$p_l = \left\{ \begin{array}{l} j, \text{ если } N \leq M; \\ l + \sum_k M, k = 0, \dots, \left[ \frac{N}{M} \right], \text{ если } N > M \end{array} \right\}. \quad (6)$$

Следовательно, каждый параллельный поток  $p_l$  выполняет следующие вычисления:

- $\prod_{i=1}^{m_j} \mu_{ij}(x_i)$  — активацию предусловий нечеткого правила;
- $g_j(x_1, x_2, \dots, x_n)$  — нахождение подзаключения нечеткого правила в случае успешной его активации (если  $\min_{i=1, 2, \dots, m_j} \mu_{ij}(x_i) \neq 0$ ),

где  $j$  — индекс правила в блоке нечетких правил, рассчитанный согласно выражению (6).

Важно отметить, что после синхронизированной обработки всего множества нечетких правил следует нахождение заключения по формуле (1), что несколько снижает эффективность параллелизма, так как данная операция выполняется строго последовательно относительно параллельной обработки правил [10]. Нахождение заключения в большинстве случаев полагается на первый поток параллельной вычислительной системы.



## МНОГОУРОВНЕВЫЙ ПАРАЛЛЕЛИЗМ НА БАЗЕ ТЕХНОЛОГИИ NVIDIA CUDA

Многоуровневый параллелизм предполагает, что для каждого из уровней должно быть зарезервировано  $n$  параллельных потоков. Это количество не обязательно должно удовлетворять степень параллелизма, однако от этого количества зависят возможности распараллеливания. Еще одна задача реализации многоуровневого параллелизма — организация взаимодействия потоков. При увеличении уровня параллелизма хотя бы до второго уровня размер блоков операций на этом уровне заметно падает. Исходя из этого, распределенные системы типа MPI, действующие на основе обмена сообщениями между процессорами, являются неэффективными. Время вычислений замедляется учащающейся передачей данных. Также стоит отметить сложную индексацию процессов в распределенных системах.

Технология NVIDIA CUDA относится к технологиям мелкозернистого параллельного программирования с использованием в качестве вычислительного устройства видеокарты. Отметим, что графические ускорители призваны решать масштабные вычислительные задачи, которые имеют высокую степень параллелизма. Главными достоинствами CUDA, отмеченными при реализации многоуровневого параллелизма, являются: наличие общей памяти, трехмерная индексация, большое количество параллельных потоков.

Элементарной вычислительной единицей на графических ускорителях NVIDIA являются нити. Нити собираются в блоки, блоки, в свою очередь, в сетку (рис. 4). Такая вычислительная архитектура, позволяет использовать более мелкие вычислительные единицы для реализации параллелизма последующих уровней. Пример реализации многоуровневого параллелизма второго уровня на базе технологии графических ускорителей NVIDIA проиллюстрирован на рис. 4. Блоки (blocks) в пределах сетки (grid) реализуют параллелизм первого уровня, тогда как нити (threads) — параллелизм второго уровня (рис. 4). Построение иерархических систем нечеткого логического вывода на основе графических ускорителей изложено в работах [9–11].

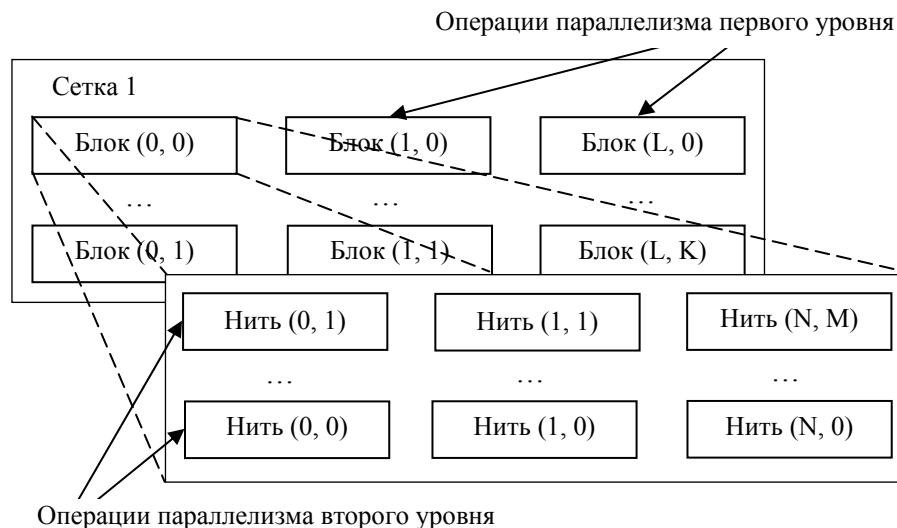


Рис. 4. Организация параллелизма второго уровня на графических ускорителях NVIDIA CUDA

## ИНТЕЛЛЕКТУАЛЬНЫЕ ПРОГРАММНЫЕ СИСТЕМЫ НА ОСНОВЕ МНОГОУРОВНЕВОГО ПАРАЛЛЕЛИЗМА И ТЕХНОЛОГИИ ПАРАЛЛЕЛЬНОГО ПРОГРАММИРОВАНИЯ CUDA

В работах [10–12] автором было разработано и спроектировано несколько иерархических параллельных интеллектуальных систем нечеткого логического вывода с применением разных уровней параллелизма и способов распараллеливания [11]. Среди таких систем целесообразно отметить работы «Интеллектуальная система оценивания качества научных работ, что подаются на конференцию» [11, 12] и «Интеллектуальная система оценивания привлекательности стартапов» [10].

С целью исследования свойств параллельных иерархических нечетких систем следует рассмотреть результаты ускорения иерархических нечетких систем второго уровня параллелизма на основе технологии графических ускорителей NVIDIA CUDA [10]. Для этого была спроектирована экспериментальная программная система, генерирующая иерархию систем нечеткого логического вывода по заданным критериям. Полученные результаты сравним с теоретической оценкой максимально возможного ускорения, воспользовавшись сформулированной теоремой (3). Проведем теоретическую оценку ускорения для параллелизма второго уровня системы на 1000 блоков нечетких правил согласно этой теореме. В качестве эксперимента была сгенерирована система, обладающая следующими свойствами: количество ярусов, зададим равным 20, количество входов для каждой элементарной системы — 3, количество нечетких правил в пределах блока — 27, количество параллельных CUDA-поток для 1-го и 2-го уровней параллелизма выбрано 100 и 27 соответственно.

Согласно формуле (4) вычислим оценку максимально возможного ускорения на каждом из уровней. Далее по формуле (3) найдем суммарное ускорение для всех уровней одновременно:

$$S_P^{\text{sum}} = s_{p_1}^1 s_{p_2}^2;$$

$$p_1 = 100; p_2 = 27;$$

$$\alpha_1 = s / M = 20 / 1000 = 0,02; \alpha_2 = 0,01;$$

$$s_{p_1}^1 = \frac{1}{\alpha_1 + \frac{1 - \alpha_1}{p_1}} = \frac{1}{0,02 + \frac{1 - 0,02}{100}} \approx 33;$$

$$s_{p_2}^2 = \frac{1}{\alpha_2 + \frac{1 - \alpha_2}{p_2}} = \frac{1}{0,01 + \frac{1 - 0,01}{27}} \approx 22;$$

$$S_P^{\text{sum}} = s_{p_1}^1 s_{p_2}^2 \approx 33 \cdot 22 \approx 726,$$

где  $M$  — количество элементарных нечетких систем;  $s$  — количество ярусов графа зависимостей между блоками нечетких правил.

Экспериментальные оценки ускорения иерархических систем нечеткого логического вывода второго уровня параллелизма в сравнении с теоретическими оценками ускорения, а также с экспериментальными оценками ускорения таких же систем для первого уровня параллелизма (использовалась технология параллельных вычислений с распределенной памятью MPI [11, 12]) показаны на рис. 5. Оценки ускорения получены для разного количества элементарных нечетких систем.

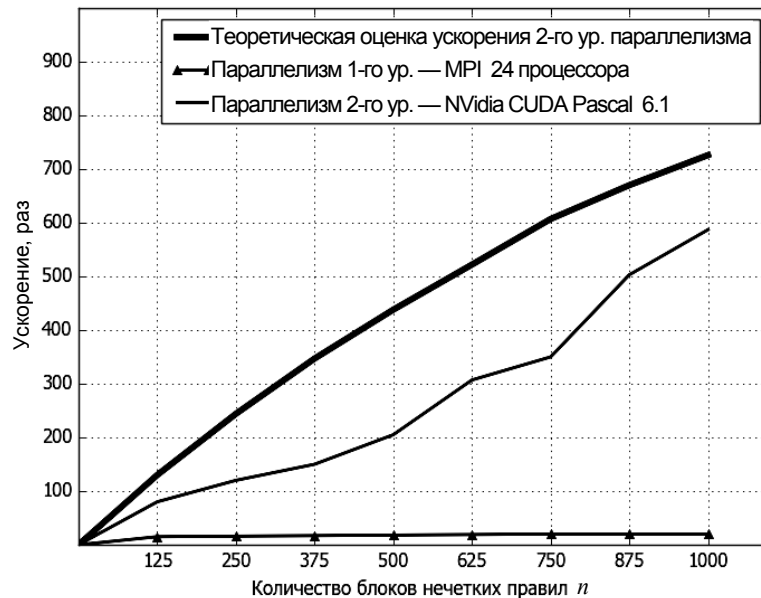


Рис. 5. Сравнительная характеристика теоретической и экспериментальных оценок ускорения для иерархических нечетких систем

Экспериментальная оценка ускорения несколько ниже теоретической, что обусловлено накладными расходами при работе с глобальной памятью графических ускорителей. Однако в целом наблюдается тенденция наращивания продуктивности вычислений с увеличением количества блоков нечетких правил. Таким образом, в пределах 1000 блоков нечетких правил решена проблема скорости вычислений при проектировании сложных иерархических нечетких систем.

## ЗАКЛЮЧЕНИЕ

В работе сформулирована концепция многоуровневого параллелизма на основе внутреннего распараллеливания, которая позволяет значительно ускорить выполнение сложных высоконагруженных вычислений. Введено и обосновано понятие уровня параллелизма. Разработан метод нахождения теоретической оценки ускорения для параллельных вычислений уровня  $n$ , а также метод распараллеливания иерархических нечетких систем до второго уровня параллелизма на примере элементарных нечетких систем типа Такаги–Сугено. Спроектирована экспериментальная программная система для исследования высоконагруженных иерархических систем нечеткого логического вывода на основе многоуровневого параллелизма второго уровня с поддержкой вычислений на графических ускорителях NVIDIA.

Полученные оценки ускорения демонстрируют, что параллельная схема второго уровня параллелизма при одних и тех же экспериментальных данных дает ускорение значительно больше, нежели распараллеливание первого уровня.

## ЛИТЕРАТУРА

1. *Zadeh L.A.* The concept of a linguistic variable and its application to approximate reasoning / L.A. Zadeh // *Information Sciences*. — 1975. — Vol. 8, N 8. — P. 199–249, 301–357.
2. *Wang D.* A survey of hierarchical fuzzy systems (invited paper) / D. Wang, X. Zeng, J.A. Keane // *International journal of computational cognition*. — 2006. — Vol. 4, N 1. — P.18–29.
3. *Єршов С.В.* Принципи побудови нечітких мультиагентних систем в розподіленій середі / С.В. Єршов // *Комп'ютерна математика*. — 2009. — № 2. — С. 54–61.
4. *Yager R.R.* On a hierarchical structure for fuzzy modeling and control / R.R. Yager // *IEEE Trans. Syst. Man Cybern.* — 1993. — N 23. — P. 1189–1197.
5. *Cordon O.* Linguistic modeling by hierarchical systems of linguistic rules / O. Cordon, F. Herrera, I. Zwir // *IEEE Trans. Fuzzy Syst.* — 2002. — N 10. — P. 2–20.
6. *Torra V.A.* review of the construction of hierarchical fuzzy systems / V.A. Torra // *Int. J. Intell. Syst.* — 2002. — N 17. — P. 531–543.
7. *Леоненков А.В.* Нечеткое моделирование в среде MATLAB и fuzzyTECH / А.В. Леоненков. — СПб.: БХВ, 2005. — 736 с.
8. *Воеводин В.В.* Параллельные вычисления / В.В. Воеводин, Вл.В. Воеводин. — СПб.: БХВ-Петербург, 2002. — 608 с.
9. *Пономаренко Р.М.* Моделі паралельних ієрархічних систем для нечіткого логічного виведення / Р.М. Пономаренко // *Комп'ютерна математика*. — 2017. — № 2. — С. 37–45.
10. *Єршов С.В.* Метод побудови паралельних систем нечіткого логічного виведення на основі графічних прискорювачів / С.В. Єршов, Р.М. Пономаренко // *Проблеми програмування*. — 2017. — № 4. — С. 3–15.
11. *Єршов С.В.* Паралельні моделі багаторівневих нечітких систем Такагі-Сугено / С.В. Єршов, Р.М. Пономаренко // *Проблеми програмування*. — 2016. — № 1. — С. 141–149.
12. *Єршов С.В.* Ярусно-паралельна модель обчислень для логічного виведення у нечітких багаторівневих системах / С.В. Єршов, Р.М. Пономаренко // *Комп'ютерна математика*. — 2016. — № 1. — С. 28–36.

*Поступила 03.09.2018*