ISSN 2079-3316 ПРОГРАММНЫЕ СИСТЕМЫ: ТЕОРИЯ И ПРИЛОЖЕНИЯ №4(31), 2016, с. 145–160

УДК 519.713.8+519.718.2+517.977.1

Н. Н. Непейвода

# О некоторых возможностях локальных вычислений в теории систем и базах данных

Аннотация. Применительно к задачам суперкомпьютинга и сверхбольших баз данных рассматривается абстрактная топологическая концепция локальных систем и прямоточной организации вычислений при декомпозиции локальных систем. Она иллюстрируется на примерах и устанавливаются базовые результаты. Устанавливается тесная взаимосвязь абстрактного понятия локальности с конкретным понятием робастности и заодно принципиальные их отличия. Работа предполагается первой в серии работ, посвящённых методам организации вычислений над локальными системами.

*Ключевые слова и фразы:* локальность, расслоение вычислений, стена памяти, робастность.

## 1. Практические и содержательные мотивировки

Эта работа планируется как первая в серии статей, посвящённых вычислениям без промежуточной памяти в процессах, которые поддаются некоторому варианту локализации зависимостей и влияния каждого действия и поэтому могут структурироваться на меньшие узлы, требующие ограниченных ресурсов и, возможно, вычисляемые распределённо. В отличие от других работ, здесь единым аппаратом берутся и разбиение аргументов на подобласти, в том числе перекрывающиеся, и разбиение данных на частичные с последующим соединением результатов, и комбинация этих «горизонтального» и «вертикального» разбиений.

Если вычислять с помощью света, то любая попытка «остановить мгновение» и временно запомнить промежуточные результаты приводит к уменьшению на порядки скорости вычислений и увеличению выделения энергии.

---

Проект проводится при финансовой поддержке РАН, проект 46П «Сверхвысокопроизводительные базы данных».

В аналоговых вычислениях на базе электричества M. B. Pour-El [1] доказала невозможность элемента «задержка». На практике этот элемент реализуется приблизительными схемами, зависящими от конкретной задачи. Таким образом, и здесь результат вычислений прямо отправляется на следующее вычислительное устройство.

«Условный элемент», пропускающий сигнал в одном либо в другом направлении в зависимости от его свойств, возможен, но в данных условиях на втором выходе всё равно будет сигнал, и скорее такой элемент нужно назвать «разделителем», отправляющим разные компоненты сигнала в преобразованном виде по двум выходам. Так действуют, в частности, применяемые в нынешних исследованиях реверсивных вычислений гейты Тоффоли, Фейнмана и Фрадкина [2]. Помимо полезного результата возникает компенсаторный, дающий возможность вернуть вычисления к исходному состоянию. Так что и условных операторов в традиционном смысле тоже нет.

В ходе исследования данной задачи, возникшей в результате поиска новой элементной базы и новых принципов работы для компьютеров, появилось понимание её взаимосвязи также с тематикой распределённых вычислений над сверхбольшими базами данных (СВБД) и теории систем.

Аналоговые и оптические элементы обладают ещё одним важным свойством. Корректность (и даже функционирование) гарантируется лишь в ограниченном диапазоне изменения аргумента и результата. Грубо говоря, сигнал должен быть от $-1$ до 1. Если используются элементы другой природы (в частности, если элементами считаются узлы сети распределённых вычислений над СВБД), то ограничение возникает по величине моделируемой области. Это свойство ограниченности может быть использовано в положительном смысле для обеспечения надёжности и безопасности распределённых вычислений. Каждый узел знает ограниченную область, а место ограничений по диапазону занимают ограничения по слою обрабатываемой информации. Таким образом, даже в своей области небезопасный узел не может восстановить информацию об исходных обрабатываемых данных в СВБД.

Поскольку «зацикливание» сигнала в световых вычислениях даёт сложно описываемые результаты и также замедляет вычисления из-за необходимости времени на выход системы в равновесный режим, и аналогичные эффекты имеются в других аналоговых случаях (например, аналоговые элементы, преобразующие двумерные и трёхмерные векторные поля), а в случае СВБД приводит к неприемлемым затратам времени и ресурсов, ограничимся здесь случаем, когда обратные связи отсутствуют. Число обрабатывающих элементов на каждом пути вычислений при реализации локальной части операции должно быть заранее ограничено константой.

Задача данной статьи: строго описать условия, при которых некоторая определённая структура вычислений позволяет организовать прямоточные расслоённые вычисления. Преимущество данной структуры в том, что она применима для нескольких типов вычислений, в частности, для цифровых вычислений типа сложения, для аналоговых вычислений с варьируемой точностью, для распределённых расслоённых вычислений. Тем самым показано, что прямоточные расслоённые вычисления могут достаточно широко применяться в базах данных и в аппаратуре информационных систем.

## 2. Базовые понятия

Определение 1. *Под прямоточными вычислениями понимаются вычисления, организованные в сеть, обладающие следующими свойствами:*

(1) *Результаты каждого обработчика непосредственно направляются в следующие разработчики.*

(2) *Длина каждого пути (число обрабатывающих элементов) в сети ограничена сверху некоторой константой $K$.*

*Прямоточные вычисления расслоённые, если они построены по следующей структуре: вначале информация разбивается на подструктуры; затем обрабатывающие элементы работают с подструктурами; в конце результат собирается из подструктур.*

При этих условиях вычислительное устройство может быть формализовано как сеть, в вершинах которой стоят преобразователи информации, а дугам приписаны сигналы. Входами устройства являются начальные вершины сети. Выходами — конечные.

Понятие прямоточных вычислений кажется исключительно слабым, намного слабее даже конечных автоматов. Возникает вопрос о принципиальных возможностях их применений.

Область применений их оказалась шире, чем виделось первоначально, и это обосновывается далее. Она тесно связана с понятием локальности системы: когда результат в каждой точке зависит лишь от её ограниченного окружения.

Поэтому наша первая задача: *построить строгую математическую модель локальности преобразований информации, не связанную с конкретным представлением данных (как числа либо автоматы), и с конкретными ограничениями на характер вычислительных элементов и сигналов.*

Инструментом для описания этой модели выбрана топология.

Определение 2. *Пусть заданы хаусдорфовы топологические пространства аргументов данных $\mathcal{A}$, значений данных $\mathcal{V}$, аргументов результата $\mathcal{B}$ и значений результата $\mathcal{W}$. Входным сигналом является функция $\mathcal{A} \to \mathcal{V}$, выходным — $\mathcal{B} \to \mathcal{W}$.*

*Класс допустимых входных функций называется типом аргументов и обозначается $\mathbb{A}$, выходных — типом результатов и обозначается $\mathbb{W}$.*

*Действие (преобразователь информации) — оператор из $\mathbb{A}$ в $\mathbb{W}$*

$$\mathfrak{F} : \mathbb{A} \to \mathbb{W}.$$

Непрерывность или гладкость аргументов и результатов в общем случае не предполагается, поскольку возможны пороговые, фрактальные либо дискретные функции.

Среди окрестностей наших пространств выделяется подкласс ограниченных $\mathcal{L}_S$, замкнутый вниз по вложению и такой, что у любой точки есть ограниченная окрестность. Здесь нижний индекс — обозначение пространства. Ограниченным называется подмножество ограниченной окрестности.

Пример 1. *Рассмотрим несколько примеров задания ограниченных окрестностей.*

(1) *Широко используемое в теории систем понятие равномерной ограниченности на метрическом пространстве $X$ формализуется у нас классом $\mathcal{L}_X(d_0)$ окрестностей диаметра не более $d_0$.*
(2) *Подмножества дискретного пространства $D$ без меры с числом элементов не более $n_0$ образуют класс ограниченных окрестностей $\mathcal{L}_D(n_0)$.*
(3) *Все окрестности метрического пространства можно считать ограниченными. Тогда ограниченным будет любое подмножество конечного диаметра (стандартное понятие ограниченности в анализе).*
(4) *Все конечные подмножества дискретного пространства можно считать ограниченными. Практически это формализует понятие конечной информации для алгоритмов.*

Следующее понятие формализует операторы, различия значений которых на некотором подмножестве не могут быть с гарантией компенсированы значениями аргумента на всём пространстве.

Определение 3. *Влияние действия $\mathfrak{F}$ на подмножестве пространства аргументов $X \subseteq \mathcal{A}$ есть множество всех аргументов, для которых значение действия может отличаться для входных сигналов, совпадающих вне $X$:*

$$\mathcal{I}(X) = \bigcup_{f_1, f_2 \colon \{x \mid f_1(x) \neq f_2(x)\} \subseteq X} \{x \mid \mathfrak{F}(f_1)(x) \neq \mathfrak{F}(f_2)(x)\} \tag{1}$$

*Действие монотонно, если*

$$X \subseteq Y \Longrightarrow \mathcal{I}(X) \subseteq \mathcal{I}(Y) \tag{2}$$

Пример 2. *Большинство используемых на практике действий монотонно в нашем смысле: изменение входных данных на меньшем множестве не может повлечь изменения выходных на большем. Приведём пример немонотонного действия.*

*Пусть имеется большая компьютерная база данных юридически значимых документов, два экземпляра которой хранятся в двух независимых равноправных хранилищах. Возникло подозрение о порче этих баз в результате злоупотреблений либо хакерских атак. Программа проверки целостности сравнивает попарно соответствующие документы из обеих баз, и в случае, если расхождений находится достаточно мало, выдаёт их список для ручной обработки исполнителями, имеющими полномочия на восстановление документов и доступ к их исходникам. Если же расхождений много, она выдаёт сообщение, что базы испорчены, и рекомендацию прекратить ими пользоваться до завершения процедуры полной перегенерации. Здесь большие различия приводят к одинаковым результатам, а меньшие к различным.*

*Другим примером является декодирование повреждённой информации при помощи найденных ошибок в контрольных функциях. Если ошибок мало, исправление ошибок возможно, если же их много, часто приходится констатировать, что файл испорчен.*

Определение 4. *Действие называется локальным, если для каждого аргумента $x$ найдётся его окрестность $\mathcal{D}(x)$ (область зависимости), такая, что если два входных сигнала $f_1$, $f_2$ различаются лишь на $\mathcal{D}(x)$, то выходные сигналы совпадают вне ограниченного множества $\mathcal{J}(x, f_1, f_2)$, включающего $x$.*

*Действие равномерно локально, если выбор $\mathcal{D}(x)$ и $\mathcal{J}(x, f_1, f_2)$ могут быть сделаны не зависящими от $f_1$, $f_2$.*

Таким образом, значения входного сигнала в некоторой точке могут влиять лишь на ограниченное подмножество выходного сигнала. Формулировка «может быть сделано» базируется на свойстве, что любое надмножество множества зависимости само является множеством зависимости.

Лемма 1. *Действие равномерно локально тогда и только тогда, когда для каждой точки существует окрестность, на которой влияние действия ограничено.*

Доказательство. Если $O$ — окрестность точки $x$, такая, что $\mathcal{I}(O)$ ограничено, то её можно принять за $D$ из определения 4, а $\mathcal{I}(O)$ за $\mathcal{J}(x, f_1, f_2)$ для любых $f_1$, $f_2$.

Если $\mathcal{J}(x, f_1, f_2)$ не зависит от $f_1$, $f_2$, то $\mathcal{I}(D) \subseteq \mathcal{J}(x, f_1, f_2)$. $\square$

Для дискретных пространств и монотонных действий этот критерий сводится к ограниченности влияния любого одноточечного множества.

Теперь рассмотрим случай неравномерной локальности и выведем достаточный критерий для возможности построения единого множества зависимости.

Напомним, что по любому направленному множеству можно определить понятие предела. Поскольку множество окрестностей является направленным, то можно задать предел множеств различия по окрестностям точки $x$:

$$\bigcap_{x \in O)} \mathcal{I}(O). \tag{3}$$

Определение 5. *Предел (3) называется влиянием $x$, обозначается $\mathcal{IN}(x)$. Множество*

$$\begin{aligned} \{y \mid \forall \mathcal{O}(y \in \mathcal{O} \Longrightarrow \\ \exists f_1, f_2(\forall z(f_1(z) \neq f_2(z) \& f_1(x) = f_2(x) \Longrightarrow z \in \mathcal{O}) \& \\ \mathfrak{F}(f_1)(x) \neq \mathfrak{F}(f_2)(x)))\} \end{aligned} \tag{4}$$

*называется зависимостью $x$ и обозначается $\mathcal{D}(x)$. Здесь $\mathcal{O}$ — переменная по окрестностям, $f_1$, $f_2$ — по входным данным. Зависимость множества $\mathcal{D}(X)$ — объединение зависимостей его элементов.*

Содержательно это определение означает: $x$ зависит от таких $y$, что изменение входных данных на $y$ может сказаться на результате в $x$, сколь бы ограниченной ни была область этого изменения.

Формально зависимость и влияние определены и для немонотонных действий, но их определение становится для немонотонных действий неустойчивым относительно задания топологии и на самом деле перестаёт быть пределом, поскольку большие окрестности с малым влиянием могут аннулировать влияние малых окрестностей (этот случай иллюстрируют две испорченные базы данных из примера 2).

Лемма 2. *Если действие монотонно, то множества зависимости и влияния не зависят от конкретного выбора открытых множеств, порождающих топологию на пространствах $\mathcal{A}$, $\mathcal{B}$.*

Доказательство. Достаточно заметить, что, если топологии эквивалентны, то для каждой окрестности точки в одной из них имеется вложенная в неё окрестность в другой. □

По этой причине в дальнейшем мы ограничиваемся рассмотрением монотонных действий.

Пример 3. *Рассмотрим примеры локальных и нелокальных действий.*

(1) *Пусть входными данными являются непрерывные векторные поля в многомерном евклидовом пространстве либо на его подобласти. Тогда оператор*

$$\Phi(\alpha)(\vec{x}) = \int\limits_{S(r(\vec{x}),\vec{x})} \cdots \int \alpha(\vec{y}) d\vec{y} \tag{5}$$

*является локальным действием. Здесь $S(a,\vec{x})$ — шар радиуса $a$ с центром $\vec{x}$, $r(\vec{x})$ — непрерывный ограниченный оператор из фазового пространства в $\mathbb{R}^+$. Влиянием множества является объединение шаров $S(r(\vec{x}),\vec{x})$ для всех его точек. Зависимостью точки является шар $S(r(\vec{x}),\vec{x})$.*

(2) *Для действительных чисел, представленных в традиционной позиционной системе, действие сложения нелокально. Влиянием $n$-го знака является бесконечный сегмент $[n,+\infty)$. Его зависимостью является прямое произведение двух бесконечных сегментов $[n,-\infty)\times[n,-\infty)$.*

(3) *При представлении чисел в системе с перекрытием «три половинки» действие сложения становится локальным, если ограниченными считаются окрестности, включающие данный знак, два предшествующих и один следующий* [3], *усиление и обобщение этого результата в* [4].

(4) *Если локальными считать любые конечные сегменты разрядов действительного числа, то в любой системе с перекрытием любая вычислимая функция действительных чисел локальна* [3].

(5) *Для натуральных чисел сложение нелокально. Но если вводится не менее двух дополнительных цифр в каждом разряде* (*Кочуров и Демидов* (*ИПС*) [5], *усиление результата* [6]), *оно становится локальным. Каждая цифра зависит от двух текущих цифр слагаемых и от двух непосредственно предшествующих.*

(6) *Действие восстановления полинома по конечному дискретному множеству значений нелокально, поскольку изменение одного входного значения влияет на значения полинома в окрестности любой точки. Здесь множеством влияния каждого исходного значения является всё пространство кроме точек других значений, множество зависимости исходной точки состоит из самой этой точки, множество зависимости любой другой точки — всё множество исходных данных.*

(7) *Действие нахождения среднего значения данных в базе данных не является локальным.*

В приведённых примерах виден один из способов локализации нелокальных действий: изменение конкретного представления данных (случай чисел). Далее мы рассмотрим другой способ, связанный с разбиением области на подобласти, данных на под-данные.

## 3. Подобласти, переполнение, прямоточность

Рассмотрим теперь взаимосвязь введённой системы понятий с организацией вычислений в больших системах и в распределённых системах с ненадёжными агентами. В качестве модельных задач будем рассматривать задачи нахождения среднего, интерполяции полинома и многомерный интеграл. Немного модифицируем действие (5). Рассмотрим задачу вычисления модуля интеграла

$$\left| \int\cdots\int_{S(r(\vec{x}),\vec{x})} \alpha(\vec{y})d\vec{y} \right|. \tag{6}$$

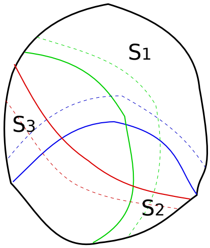



Рис. 1. Пересекающиеся подобласти и зависимости

Пусть отдельные устройства (или агенты при распределённом вычислении) могут посчитать интеграл с нужной точностью лишь на ограниченной части интересующей нас области. Тогда область можно разбить на подобласти, каждая из которых заведомо может быть обработана одним агентом. Но области взаимодействуют: влияние каждой из них распространяется также на часть соседних. Возникает в общем случае неприятный итерационный процесс, расползающийся по аргументам всё дальше и дальше. Но во всех трёх случаях (как и в большинстве реальных задач) есть понятие нуля, и его мы используем для работы с перекрывающейся информацией.

Определение 6. *Область переполнения — часть зависимости области $S$, не входящая в $S$: $\mathcal{D}(S) \setminus S$.*

На области переполнения мы будем подавать нулевые значения на вход обработчика. Таким образом, простейший способ разбиения задачи интегрирования для распределённых вычислений:

> Каждый обработчик получает значения аргумента на некоторой подобласти $S_i$ и множество $D_i$, на котором ему необходимо задать нулевые значения в операторе интегрирования и выдать результат.

На рис. 1 показано разбиение области данных на три пересекающиеся подобласти. Их зависимости обозначены пунктирными линиями соответствующего цвета.

С математической (а не алгоритмической) точки зрения это соответствует тому, что функция $f$ ограничивается на подобласть $S_i$ и вне её продолжается нулями. Но даже в случае весьма устойчивой задачи численного интегрирования такое «ступенчатое» пополнение может привести к неприятным биениям вблизи поверхности разрыва, а в случае применения более тонких методов необходимо обеспечить гладкость каждой частичной функции $f_i$. Для этого необходимо использовать разбиение на пересекающиеся подобласти и на пересечениях компонент гладко опускать функцию до нуля при приближении к границе области. Таким образом, здесь возникает задача корректного разбиения входных данных, а не только областей аргументов. В случае линейного оператора, подобного интегрированию, она решается просто: на пересечении областей $(D_i)_{i \in I}$ нужно обеспечить равенство

$$\sum_{i \in I} F_i(x) = f(x).$$

Таким образом, мы приходим к разбиению функций, подобному используемому в анализе и стандартных численных методах. Его предпосылками оказываются локальность действий и линейность соответствующих функционалов.

Нелинейная часть вычислений — нахождение модуля — может быть проделана после того, как аддитивно собраны результаты всех частичных вычислений и, поскольку области влияния и зависимости каждого аргумента здесь одноточечные, лишь для нужных аргументов.

Теперь рассмотрим задачу нахождения среднего арифметического очень большого числа данных при условии, что части этого массива данных приходится передавать ненадёжным агентам. Чтобы решить эту задачу, обобщим её до нахождения взвешенной суммы. Тогда естественно передавать каждому агенту пары из веса данного и данного. В этом случае такую пару легко исказить, сделав быстрое умножение одного из её членов на $\alpha$, а другого на $\alpha^{-1}$. Далее, можно аддитивно разбить некоторые данные, и всё равно правильный результат будет получен простым суммированием частичных данных.

Таким образом, мы видим, что здесь возникает комбинация разбиения данных и разбиения областей, и опять-таки на пересечении областей данные должны обязательно разбиваться. Более того, нелокальную задачу удалось свести к нескольким локальным. Далее, за счёт разбиения данных можно преодолеть ограничения на диапазон величин, если наши вычислительные агенты ограничены по диапазону.

Заметим, что для вычисления других характеристик, например, медианы, требуется более тонкое сведение нелокальной задачи к локальным.

И, наконец, задача построения полинома. Если произвольным образом аддитивно разбить значения полинома в исходных точках,

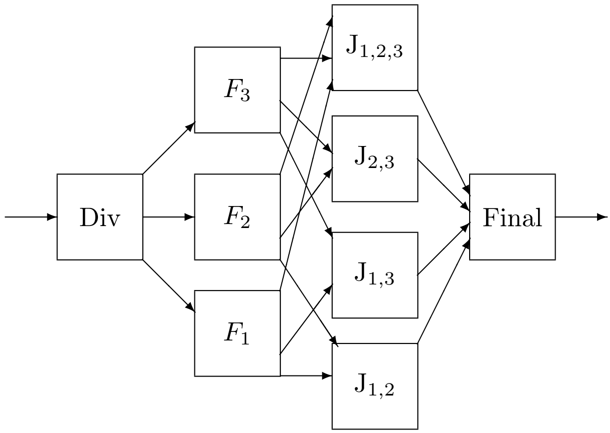



Рис. 2. Вычислительный процесс над локальными агентами

то сумма полученных значений в любой точке будет давать искомое значение результирующего полинома. А вот разбить область аргументов на подобласти здесь не получается, можно лишь упростить частичные вычисления, задав большинство из исходных значений 0.

## 4. Взаимосвязь с прямоточностью

Рассмотрим взаимосвязь введенных концепций с прямоточностью. Заметим, что ни в одном из наших случаев не возникает необходимости накапливать промежуточные данные. Результаты, полученные процессором, разбивающим информацию, могут прямо передаваться агентам, а результаты агента — процессору, собирающему итоговую информацию. Заметим также, что технология map-reduce (см. напр. [7]) является частным случаем наших разбиений (это подробнее рассматривается в следующей статье).

Видно, что, если действия локальные, возникает общая схема прямоточных вычислений, если каждый вычислительный агент работает на своей подобласти и подобласти пересекаются (как минимум, по влиянию). На рис. 2 показана организация вычислительного процесса для разбиения 1. Блок Div разделяет входные данные на потоки, направляемые в подобласти. При этом данные, попавшие в пересечения, разбиваются. Блоки $\mathrm{F}_i$ вычисляют частичные действия. Блоки

$\mathrm{J}_{i,j,k}$ объединяют результаты соответствующих переполнений. Блок Final объединяет всё в конечный результат. Чтобы несколько менее загоромождать схему, каналы прямой передачи данных из областей в результат не изображаются.

На этой фигуре видно, что даже три области, находящиеся в соотношении общего вида, усложняют структуру схемы и повышают её сложность. Поэтому возникает общая стратегия: по мере возможности направлять зависимости в одну сторону, чтобы исключать тройные пересечения.

## 5. Соотношения с другими направлениями

Понятие локальности действия является важнейшим в теории систем, но в нынешней формулировке оно привязано к мере.

Прежде всего, заметим, что обыкновенные дифференциальные уравнения, удовлетворяющие условию единственности решения, локальны, если не имеется управления. Поведение системы детерминируется значением аргумента в исходной точке. Марковские и обобщённые марковские цепи и процессы [8] представляют менее вырожденный вариант локальности. Следующее значение определяется лишь конечным набором (конечной окрестностью) предшествующих.

В теории управления приближается к локальности понятие гиперустойчивости [9], но принципиальная разница в том, что гиперустойчивость направлена на уменьшение различий в поведении системы после возмущения, а не на исчезновение их. В книге [10] понятия ещё приближаются к нашим, но по-прежнему требуетсмя уменьшение различий и всё привязано к числам и к оси времени. Ещё ближе к нашим понятия, применяющиеся в задачах идентификации и анализа систем [11–13], но здесь пространство результатов заранее сильно ограничивается априорными предположениями о структуре системы.

А робастность [14] и локальность взаимосвязаны исключительно тесно. Оформим это в виде точного результата, обобщив определение из [14], используя наше понятие ограниченности.

Определение 7. *Тривиальное расстояние на пространстве $X$ определяется условным выражением:*

$$\rho(x,y)=\begin{cases}0, & x=y\\ 1, & x\neq y\end{cases}.$$

*Пусть в множестве значений входных данных выделено подмножество нежелательных значений (например, неопределённых либо ошибочных) Действие $\mathcal{F}$ робастно, если пространство значений результата $W$ метрическое и если две функции $f_1$, $f_2$ различаются лишь на некотором ограниченном множестве и $f_2$ принимает на нём нежелательные значения, то для любого $\varepsilon > 0$ найдётся ограниченное множество аргументов результата $K_\varepsilon$, вне которого $\rho(\mathcal{F}(f_1)(x), \mathcal{F}(f_2)(x)) < \varepsilon$.*

Лемма 3. *Любое монотонное локальное действие робастно при тривиальном расстоянии.*

Очевидно.

Обратное верно не всегда, поскольку робастность означает исчезновение эффекта лишь нежелательных аргументов.

Локальность возникает и в дискретных вычислениях, хотя здесь она называется обычно «автоматными вычислениями» и моделируется конечными автоматами.

Поскольку все эти случаи рассматриваются с помощью разных инструментов, полезные находки из одной области неизвестны в другой.

## Заключение

Общие структуры дискретных и непрерывных процессов, представляемых как сети действий без запоминания промежуточных результатов, позволяют применять к ним единые методы структурирования и в дальнейшем оптимизации. На этом пути важно продвигать методы работы с нелинейными локальными операторами и разрабатывать общие методы структуризации линейных нелокальных операторов. Совершенно не разработанной остаётся тема немонотонных действий. Необходима также дальнейшая отвязка от чисел и переход к алгебраическим и топологическим методам.

## Список литературы

[1] M. B. Pour-El. “Abstract computability and its relation to the general purpose analog computer (some connections between logic, differential equations and analog computers)”, *Transactions of the American Mathematical Society*, **199** (1974), pp. 1–28. ↑ 146

[2] A. De Vos. *Reversible Computing: Fundamentals, Quantum Computing, and Applications*, WILEY-WCH Verlag, Weinheim, 2010, 261 p. ↑ 146

[3] Н. Н. Непейвода, И. Н. Григоревский, Е. П. Лилитко. «О представлении действительных чисел», *Программные системы: теория и приложения*, **5**:4((22) (2014), с. 105–120, URL: http://psta.psiras.ru/read/psta2014_4_105-121.pdf ↑ 152

[4] А. Б. Шворин. «Параллельное сложение вещественных чисел в системах счисления с перекрытием», *Программные системы: теория и приложения*, **6**:2 (2015), с. 101–117, URL: psta.psiras.ru/read/psta2015_2_101-117.pdf ↑ 152

[5] Н. Н. Непейвода, Е. В. Кочуров, А. А. Демидов, А. Б. Шворин. «Работа с числами в системах счисления с перекрытием и с переносами», *X конференция «Свободное программное обеспечение в высшей школе»*, Изд-во «Университет города Переславля», Переславль-Залесский, 2015, с. 7–9. ↑ 152

[6] E. Pelantova, M. Svobodova. “Minimal Digit Sets for Parallel Addition in Non-Standard Numeration Systems”, *Journal of Integer Sequences*, **16** (2013), 13.2.17. ↑ 152

[7] S. Sakr, A. Liu, A. G. Fayoumi. “The Family of Mapreduce and Large-scale Data Processing Systems”, *ACM Comput. Surv.*, **46**:1 (2013), pp. 11:1–11:44. ↑ 155

[8] W. J. Stewart. *Probability, Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Modeling*, Princeton University Press, Princeton, 2009, 720 p. ↑ 156

[9] В. М. Попов. *Гиперустойчивость автоматических систем*, Наука. Глав. ред. физико-математической лит-ры, М., 1970, 453 с. ↑ 156

[10] Е. К. Макаров, С. Н. Попова. *Управляемость асимптотических инвариантов нестационарных линейных систем*, Беларус. навука, Минск, 2012, 407 с. ↑ 156

[11] В. Г. Ильичев. «Локальные и глобальные свойства неавтономных динамических систем и их приложение в моделях конкуренции», *Сибирский математический журнал*, **44**:3 (2003), с. 622–635. ↑ 156

[12] Н. А. Бодунов. «Введение в теорию локальной параметрической идентифицируемости», *Дифференциальные уравнения и процессы управления*, 2012, №2, с. 1–137, URL: http://www.math.spbu.ru/diffjournal/pdf/bodunov_book.pdf ↑ 156

[13] В. Б. Живетин, *Введение в теорию риска (динамических систем)*, Риски и безопасность человеческой деятельности, т. **16**, Институт проблем риска, 2016, 156 с. ↑ 156

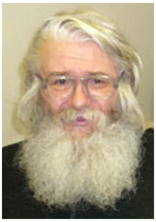
[14] C. Briat, *Linear Parameter-Varying and Time-Delay Systems. Analysis, Observation, Filtering & Control*, Advances in Delay and Dynamics, vol. **3**, Springer Verlag, Berlin–Heidelberg, 2015. ↑ 156

*Пример ссылки на эту публикацию:*

Н. Н. Непейвода. «О некоторых возможностях локальных вычислений в теории систем и базах данных», *Программные системы: теория и приложения*, 2016, **7**:4(31), с. 145–160.

URL: http://psta.psiras.ru/read/psta2016_4_145-160.pdf

*Об авторе:*



**Николай Николаевич Непейвода**

Главный научный сотрудник ИПС РАН, научный руководитель работ. Автор более 200 публикаций по конструктивной математике, логике, информатике

*e-mail:* nepejvodann@gmail.com

Nikolai Nepejvoda. *Local computations in system theory and VLDB*.

Abstract. Concept of intermediate-memory-free organization of computing process over big data processors is considered. An abstract topological notion of local system is introduced. It is considered on examples from various domains. Basic results including interrelation with robustness are stated. This work is planned to be the first in the series devoted to memory-free computations and locality. (*In Russian*).

*Key words and phrases:* locality, distributed computing, wall of memory, robustness.

*References*

[1] M. B. Pour-El. "Abstract computability and its relation to the general purpose analog computer (some connections between logic, differential equations and analog computers)", *Transactions of the American Mathematical Society*, **199** (1974), pp. 1–28.

[2] A. De Vos. *Reversible computing: fundamentals, quantum computing, and applications*, WILEY-WCH Verlag, Weinheim, 2010, 261 p.

[3] N. N. Nepeyvoda, I. N. Grigorevskiy, Ye. P. Lilitko. "New representation of real numbers", *Programmnyye sistemy: teoriya i prilozheniya*, **5**:4((22) (2014), pp. 105–120 (in Russian), URL: http://psta.psiras.ru/read/psta2014_4_105-121.pdf

Supported by RAS (project 46P).

[4] A. B. Shvorin. "Parallel Addition of Real Numbers in Overlaying Numeration Systems", *Programmnyye sistemy: teoriya i prilozheniya*, **6**:2 (2015), pp. 101–117, URL: http://psta.psiras.ru/read/psta2015_2_101-117.pdf

[5] N. N. Nepeyvoda, E. V. Kochurov, A. A. Demidov, A. B. Shvorin. "Using the numbers in numeral systems with overlapping and translations", *X konferentsiya "Svobodnoe programmnoe obespechenie v vysshey shkole"*, Izd-vo "Universitet goroda Pereslavlya", Pereslavl'-Zalesskiy, 2015, pp. 7–9 (in Russian).

[6] E. Pelantova, M. Svobodova. "Minimal Digit Sets for Parallel Addition in Non-Standard Numeration Systems", *Journal of Integer Sequences*, **16** (2013), 13.2.17.

[7] S. Sakr, A. Liu, A. G. Fayoumi. "The Family of Mapreduce and Large-scale Data Processing Systems", *ACM Comput. Surv.*, **46**:1 (2013), pp. 11:1–11:44.

[8] W. J. Stewart. *Probability, Markov chains, queues, and simulation: the mathematical basis of performance modeling*, Princeton University Press, Princeton, 2009, 720 p.

[9] V. M. Popov. *Hyperstability of automatic systems*, Nauka. Glav. red. fiziko-matematicheskoy lit-ry, M., 1970 (in Russian), 453 p.

[10] Ye. K. Makarov, S. N. Popova. *Controlability of asymptotic invariants of unsteady linear systems*, Belarus. navuka, Minsk, 2012 (in Russian), 407 p.

[11] V. G. Il'ichev. "Local and global properties of nonautonomous dynamical systems and their application to competition models", *Siberian Math. J.*, **44**:3 (2003), pp. 490–499.

[12] N. A. Bodunov. "Introduction to the theory of local parametric identifiability", *Differentsial'nyye uravneniya i protsessy upravleniya*, 2012, no.2, pp. 1–137 (in Russian), URL: http://www.math.spbu.ru/diffjournal/pdf/bodunov_book.pdf

[13] V. B. Zhivetin, *Introduction to risk theory (dynamic systems)*, Riski i bezopasnost' chelovecheskoy deyatel'nosti, vol. **16**, Institut problem riska, 2016 (in Russian), 156 p.

[14] C. Briat, *Linear parameter-varying and time-delay systems. Analysis, observation, filtering & control*, Advances in Delay and Dynamics, vol. **3**, Springer Verlag, Berlin–Heidelberg, 2015.

*Sample citation of this publication:*

Nikolai Nepejvoda. "Local computations in system theory and VLDB", *Program systems: Theory and applications*, 2016, **7**:4(31), pp. 145–160. (*In Russian*). URL: http://psta.psiras.ru/read/psta2016_4_145-160.pdf