ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 52 за 2018 г.

ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

Четверушкин Б.Н.,
Якобовский М.В.

Вычислительные алгоритмы и архитектура систем высокой производительности

Рекомендуемая форма библиографической ссылки: Четверушкин Б.Н., Якобовский М.В. Вычислительные алгоритмы и архитектура систем высокой производительности // Препринты ИПМ им. М.В.Келдыша. 2018. № 52. 12 с. doi:10.20948/prepr-2018-52
URL: http://library.keldysh.ru/preprint.asp?id=2018-52

О р д е н а   Л е н и н а
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Р о с с и й с к о й   а к а д е м и и   н а у к

Б.Н. Четверушкин, М.В. Якобовский

Вычислительные алгоритмы и архитектура систем высокой производительности

Москва — 2018

Четверушкин Б.Н., Якобовский М.В.

Вычислительные алгоритмы и архитектура систем высокой производительности

Обсуждается вопрос о влиянии методов прикладной математики на прогресс в архитектуре и производительности вычислительных систем. Рассматриваются особенности вычислительных алгоритмов для решения проблемы отказоустойчивости, актуальной для перспективных систем экзафлопсной и высшей производительности.

Ключевые слова: суперкомпьютер, отказоустойчивость, гиперболические системы уравнений, контрольная точка

Boris Nikolavich Chetverushkin, Mikhail Vladimirovich Yakobovskiy

Numerical algorithms and architecture of HPC systems

The influence of mathematics techniques to the progress in architecture and performance of computing systems is discussed. The properties of computational algorithms aimed for solving the problem of fault tolerance, which is topical for prospective exaflops and super-exaflops systems, are considered.

Keywords: supercomputer, fault tolerance, hyperbolic systems of equations, control point

Работа выполнена при поддержке Российского фонда фундаментальных исследований, проекты 17-07-01604а, 18-01-00587а.

Оглавление

Введение

Очевидным представляется факт, что архитектура существующих и проектируемых в ближайшее время вычислительных систем оказывает непосредственное влияние на развитие алгоритмов, используемых для решения конкретных задач. Так, например, тенденции развития, начиная с восьмидесятых годов 20 столетия, многопроцессорных вычислительных систем, опирающихся на использование систем с архитектурой распределенной памяти, привели к существенному изменению взглядов на параллельные алгоритмы. Одним из основных требований к алгоритмам стала возможность минимизации обмена информацией между процессорами.

Дальнейшее развитие вычислительной техники связано с появлением многоядерных процессоров, использованием графических плат в качестве ускорителей, что привело к ужесточению требования к алгоритмам. На первый план стала выдвигаться логическая простота алгоритма в обязательном сочетании с его эффективностью. Создание алгоритмов, в которых успешно сочетаются эти свойства, представляется далеко нетривиальной задачей.

Хорошо известным удачным примером являются явные схемы для решения параболических уравнений. Они логически просты и идеально адаптируются на архитектуру различных типов многопроцессорных систем. Однако, жесткое условие устойчивости

$$\Delta t \le h^2, \quad (1)$$

где Δt - допустимый шаг по времени, h - характерный шаг пространственной сетки, из-за не недопустимо малого шага Δt не даёт практической возможности реализовать этот алгоритм на подробных пространственных сетках.

В данной работе рассмотрим обратную ситуацию, связанную с влиянием особенностей вычислительных алгоритмов на архитектуру многопроцессорных систем.

Многопроцессорные вычислительные системы гибридной архитектуры

Графические платы разрабатывались и активно применялись в индустрии компьютерных игр для построения изображения. Достоинством графических плат являлась высокая производительность при относительно низкой стоимости и энергопотреблении. Однако эти преимущества применительно к расчету задач, на решение которых были ориентированы существующие в нулевых годах этого столетия суперкомпьютеры, перечеркивались их плохой приспособленностью к использованию ветвящихся алгоритмов. Следует отметить, что логическая простота алгоритмов являлась одной из составляющих успеха многопроцессорных вычислений, независимо от типа используемых компьютерных архитектур. Эта ситуация потребовала усилий по созданию логически простых и, в тоже время, эффективных алгоритмов. Появление таких алгоритмов создало базу для использования графических плат при решении довольно широкого круга практических важных задач.

Первой крупной вычислительной системой в России, использующей графические платы в качестве ускорителя, явилась система К-100 с пиковой производительностью 107 Терафлопс, установленная в ИПМ им. М.В. Келдыша РАН в 2010 году. Каждый вычислительный узел этой системы состоит из двух шестиядерных процессоров Intel Xeon X5670 и трёх ускорителей nVidia Fermi C2050, каждый из которых содержит 448-ядер CUDA. Вычислительные узлы объединены высокоскоростными сетями передачи данных МВС-экспресс и Infiniband [1,2].

Структура используемых алгоритмов [1,3] включала небольшое число операций, ответственных за общую организацию и за логику вычислений (обрабатываемых на однотипных CPU ядрах процессоров общего назначения), и множество однородных арифметических операций, выполняемых на графических платах над большим набором однородной информации.

Вычислительный комплекс К-100 используется для решения достаточно широкого круга задач, в которые, наряду с газодинамическими, входят и задачи, связанные с моделированием переноса излучения и молекулярной динамики. Работы, направленные на расширение круга задач, доступных для решения на подобных гетерогенных системах, ведутся и другими отечественными и зарубежными научными коллективами.

Возможность применения подобного рода ускорителей для решения практически важных задач явились хорошим стимулом для совершенствования графических плат. Они быстро наращивают свою производительность при относительно небольшом энергопотреблении.

Тенденции, связанные с возможностью развития графических плат, а самое главное, с расширением сферы их использования на базе вновь создаваемых алгоритмов, сформировали широко распространённое мнение о том, что будущие вычислительные системы, с производительности нескольких десятков Петафлопс и выше, будут системами с гетерогенной архитектурой.

Обратим внимание на то, что сколько-нибудь успешное создание новой востребованной элементной базы невозможно без учета предыдущего опыта математического моделирования и тенденций развития вычислительных алгоритмов. Наиболее успешным является процесс, когда создание нового процессора происходит параллельно с моделированием на его макете решения достаточно сложных задач.

В следующем параграфе рассмотрим, как с помощью математических моделей вычислительных алгоритмов можно значительно продвинуться в решении проблемы отказоустойчивости, которая актуальна для системы Экзафлопсной и более высокой производительности.

Гиперболизация уравнений механики сплошной среды и проблема отказоустойчивости

Вычислительные системы Экзафлопсной производительности предположительно появятся к уже близкому 2025 году. Они будут состоять из миллионов независимых узлов вычислителей. Такая оценка основана на экстраполяции современных тенденций развития элементной базы вычислительной техники и не учитывает возможность появления новых научных, а также технических прорывов, которые могут коренным образом изменить ситуацию. Вероятность выхода из строя того или иного процессора в этом случае велика. По существующим оценкам, для системы Экзафлопсной производительности, выход из строя хотя бы одного из процессоров будет происходить каждые 20-25 минут. При неизбежном росте производительности свыше 1 Экзафлопса, период бесперебойной работы всей системы будет всё больше сокращаться [4-7].

Ликвидация последствий сбоя процессоров может происходить по следующей процедуре. Регулярно, с некоторым интервалом δ, происходит запись промежуточных данных большой эволюционной задачи. В случае выхода из строя процессора, происходит остановка расчета задачи, выполняется замена испорченного процессора на резервный, и выполняется возврат на момент модельного времени, соответствующий последней записи состояния расчета.

При существующей оценке частоты выхода из строя процессоров, по всей видимости, критической является производительность в один Экзафлопс. Следует отметить, что характерное время записи промежуточных данных, согласно прогнозам так же будет составлять десятки секунд. Если использовать для расчёта задачи вычислительные мощности свыше одного Экзафлопса, то суперкомпьютер фактически будет работать вхолостую, постоянно возвращаясь к одной и той же контрольной записи и заменяя процессоры.

Отметим, что сама процедура замены тоже требует некоторого астрономического времени Δ, логарифмически зависящего от числа вычислительных узлов и не зависящего от числа шагов модельного времени отделяющего момент появления неисправного процессора от момента сохранения последней контрольной точки. Таким образом, её длительность не является критическим фактором, существенно влияющим на возможность эффективного использования суперкомпьютеров.

Неожиданно, выход из тупиковой ситуации оказывается возможным на основе вычислительных алгоритмов. В основе соответствующих методов лежит гиперболизация уравнений механики сплошной среды, основные положения которой кратко изложены в работах [8-9].

Например, уравнения механики сплошной среды могут быть записаны в виде [10]:

$$\frac{\partial Q_i}{\partial t} = div\,\bar{S}_i\,, \tag{2}$$

где Q_i - вектор основных переменных (плотность, импульс, полная энергия), $\bar{S}_i$ – вектор потока субстанции, приводящей к изменению Q_i. Заметим, что для уравнения Навье-Стокса, система (2) является системой параболического типа.

Опираясь на связь между кинетическим и газодинамическим описанием сплошной среды, вместо системы (2) можно записать альтернативную систему:

$$\frac{\partial Q_i}{\partial t} + \frac{\tau}{2}\frac{\partial^2 Q_i}{\partial t^2} = div\,S_i^*. \tag{3}$$

В этом случае газодинамическое уравнение неразрывности для системы (3) будет выглядеть следующим образом:

$$\frac{\partial \rho}{\partial t} + \frac{\tau}{2}\frac{\partial^2 \rho}{\partial t^2} = \frac{\partial}{\partial x_k}\left[-\rho U_k + \frac{\tau}{2}\frac{\partial}{\partial x_p}\left(\rho U_k U_p + p\delta_{kp}\right)\right], \tag{4}$$

где ρ – плотность, U_k – k-ая компонента скорости, p - газодинамическое давление, τ - характерное время между столкновениями молекул газа.

Доказано, что гиперболизированная система (3) для описания течений вязкого теплопроводного газа отличается от соответствующих уравнений Навье-Стокса на члены второго порядка малости по числу Кнудсена $O(Kn^2)$ [8, 11-12]. Эта система получила название квазигазодинамической системы уравнений [8] (QGS-система).

Естественно, отсутствие сколько-нибудь значимых различий результатов численного моделирования с использованием уравнения Навье-Стокса и QGS-системы подтверждается многочисленными вычислительными экспериментами. Заметим, что обобщение этого подхода на моделирование магнито-газодинамических явлений в астрофизике, позволяет получать качественно новые результаты с применением пространственных сеток, состоящих из более чем 10^9 узлов в трёхмерной пространственной постановке [13-14].

QGS-система (3)-(4) является гиперболической системой уравнений [15]. Её гиперболичность открывает возможность построения логических простых и эффективных алгоритмов, хорошо адаптируемых к архитектуре вычислительных систем с экстрамассивным параллелизмом.

Принципиальную возможность построения стратегии, обеспечивающей практически полное исключение влияния возникающих отказов процессоров на время выполнения расчета в целом, рассмотрим на примере пространственных пространственно-одномерного уравнения (5).

$$\frac{\partial^2 f}{\partial x^2} - \frac{1}{C^2}\frac{\partial^2 f}{\partial t^2} = F(x,t). \qquad (5)$$

В отличие от решения уравнения параболического типа, решение гиперболического уравнения на момент времени t зависит от значений искомой функции в некоторый предыдущий момент времени $t-\delta$ не во всей области, где она определена, а от значений только в строго определенной геометрической области пространства содержащей n_1 узлов сетки (рис. 2). Эта область влияния для решения уравнения (5) определяется двумя характеристиками $x-Ct$ и $x+Ct$. Согласно традиционной стратегии, замена неисправного процессора в этом случае выглядит следующим образом [16]. На момент модельного времени t выхода из строя одного из процессоров, расчет во всей области останавливается и происходит возвращение решения на момент последней записи промежуточных данных $t-\delta$.

В соответствии с принципом геометрического параллелизма[1] и характеристикам $x-Ct$ и $x+Ct$ легко определить область, которая, на момент времени $t-\delta$, влияет на решение в зоне ответственности неисправного процессора в момент времени t. Эта область имеет б***о***льшие геометрические размеры, чем область решения, расчет которой выполнялся неисправным процессором. Соответственно, необходимо увеличить число процессоров, которые будут ответственны за решение в выделенной таким образом подобласти на момент времени $t-\delta$. Впрочем, количество дополнительных процессов, вычисляющих решение выделенной области с момента времени $t-\delta$ до момента t, и далее, до момента $t+\delta$, существенно меньше общего количества процессоров, участвующих в расчете в соответствии с принципом геометрического параллелизма [16-18]. Их оценка снизу, не учитывающая затраты на замену процессора и перераспределение данных между

[1] В соответствии с принципом «геометрического параллелизма», широко использующемся при решении эволюционных задач математической физики, каждый процессор ведет расчеты во все моменты времени в своей строго определенной подобности.

процессорами, даётся выражением (6), где $d = 1{,}2$ или 3 – размерность моделируемого пространства.

$$p_d > \frac{2}{d+1}\sum_{i=0}^{d}\alpha^i \tag{6}$$

Здесь, параметр $\alpha = 1 + 2\gamma\frac{k_1}{n_0}$, $\gamma = \frac{C\Delta t}{h}$ - число Куранта, k_1 - число шагов модельного времени, прошедшего с момента записи последней контрольной точки, n_0^d – число расчетных точек, изначально обрабатываемых каждым из процессоров (в начале расчета, до выхода из строя какого-либо процессора).

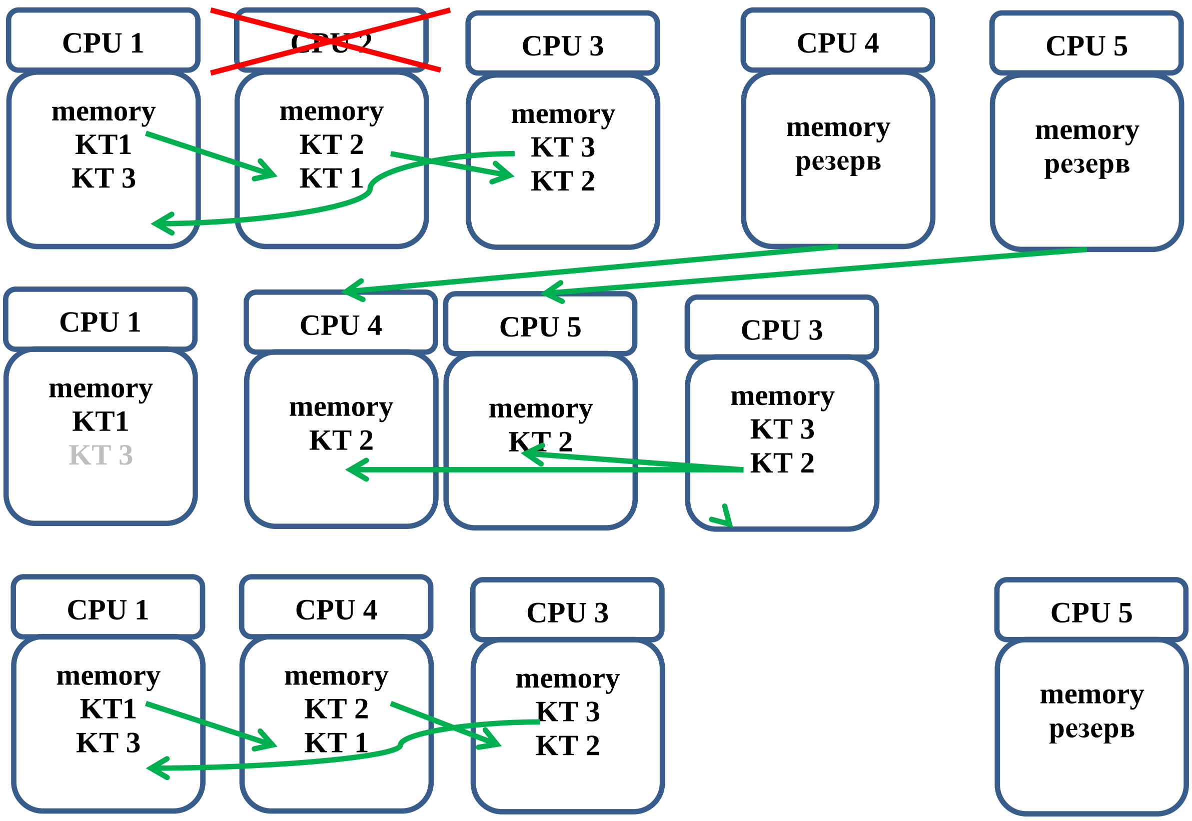

Рис. 1. Общая стратегия обеспечения отказоустойчивости

Опишем предлагаемую стратегию. В ходе выполнения расчета на исправных вычислительных узлах, каждый из них регулярно копирует на некоторый другой вычислительный узел ключевые данные, достаточные для возобновления расчета – локальную контрольную точку. Например, на рисунке 1 приведён пример проведения расчета на 3х процессорах (CPU1, CPU2, CPU3). Вычислительные узлы CPU4 и CPU5 являются резервным и не принимают участия в расчете до тех пор, пока остальные исправны. Узел CPU1 периодически сохраняет локальную контрольную точку (КТ1) - данные, достаточные для возобновления расчета, в свою оперативную память и в оперативную память CPU2. Аналогичные операции выполняют остальные выполняющие расчет вычислительные узлы. В случае выхода из строя, например, CPU2, он заменяется требуемым в соответствии с (6) количеством резервных вычислительных узлов, не принимавших ранее участия в выполнении вычислений. В примере показана замена на два вычислительных узла. Вновь добавленные вычислительные узлы считывают последнюю локальную контрольную точку неисправного вычислительного узла и равномерно распределяют между собой соответствующую вычислительную нагрузку. Далее выполняется восполнение данных, утерянных в результате выхода узла из строя и вычисление данных, необходимых для того, что бы дополнительные процессоры выполнили расчет всех тех шагов модельного времени (с момента t до момента $t+\delta$), которые уже обработали и продолжают обрабатывать остальные процессоры, не прерывавшие своей нормальной работы. Общее увеличение времени счета Δ, обусловленное появлением неисправного процессора, определяется временем замены неисправного вычислительного узла на набор резервных и перестроением соответствующей виртуальной топологии связей вычислительных узлов и временем считывания и перераспределения между ними последней контрольной точки. Легко видеть, что эта задержка не зависит от числа локально потерянных результатов расчета моментов модельного времени от $t-\delta$ до t.

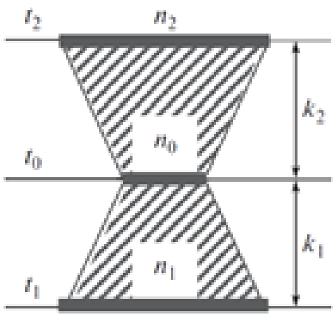

Рис. 2. Область ускоренного расчета

По завершении процесса согласования моментов обрабатываемого модельного времени (когда вычислены данные соответствующие моменту $t+\delta$), данные, обрабатываемые на CPU5, передаются вычислительному узлу CPU4, который продолжает дальнейший расчет, заменяя неисправный CPU2. Вычислительный узел CPU5 возвращается в пул резервных узлов и далее не участвует в расчете до появления следующего неисправного процессора.

Важно подчеркнуть, что во время процедуры восполнения информации, утерянной в результате выхода из строя процессора CPU2 и согласования номеров шагов модельного времени, вычисляемого основными (CPU1 и CPU3) и дополнительными процессорами (CPU4 и CPU5), расчет на оставшихся процессорах (на схеме они не показаны, но их может быть любое количество) не останавливается. На них продолжает расчет шагов модельного времени с момента t до момента $t+\delta$. Увеличим в соответствии с (6), по сравнению с первоначальным, число процессоров в области влияния на момент времени $t-\delta$. Эти дополнительные процессоры, как и процессор, пришедший на смену неисправному, будут предоставлены из резерва заранее предусматриваемого при запуске расчета. Увеличение числа процессов для расчета в области влияния на момент времени $t - \delta$ приведет к уменьшению числа расчетных точек в каждой из подобластей, за вычисления в которой ответственен каждый из дополнительных процессоров. Уменьшение числа обрабатываемых точек приведет к ускорению расчёта данных, которые генерируются в области влияния. Это дает возможность преодолеть к моменту времени $t + \delta$ отставание от основной массы процессоров, которые не останавливали расчёта (рис. 2).

Таким образом, поскольку большая часть процессоров продолжает выполнять вычисления в обычном режиме, даже если один из процессоров вышел из строя, предложенный метод обеспечивает независимость времени выполнения расчета от того, были ли во время вычислений отказы вычислительных узлов, или их не было. Конечно, общее время расчета увеличивается, по сравнению с тем случаем, когда алгоритм не предусматривает никаких действий для поддержки отказоустойчивости. Но, соответствующие накладные расходы регулярным образом добавляются к основной вычислительной работе и не зависят от числа вычислительных узлов, вышедших из строя во время выполнения расчета. Более того, рассмотренный подход обеспечивает выполнение расчета даже в том случае, когда в ходе устранения последствий выхода процессора из строя выйдет из строя ещё один (или более) процессор. Необходимым условием нормальной работы является достаточное расстояние в решетке процессоров между неисправными вычислительными узлами, так же определяемое параметрами (6). Следует отметить, что, несмотря на высокую вероятность (близкую к 1) регулярного выхода процессоров из строя, вероятность выхода из строя пары именно соседних узлов невелика, поскольку она меньше относительно небольшой сегодня вероятности выхода из строя узла в 10-100 Терафлопсной системе.

Сделаем два замечания в связи с предложенным подходом к решению проблемы отказоустойчивости.

Замечание 1. Данный подход, основанный на использовании особенности вычислительных алгоритмов, ни коем образом не отодвигает на второй план проблемы совершенствования качества процессоров и других узлов современных вычислительных систем. Решение проблемы отказоустойчивости следует искать на пути сочетание этих двух направлений.

Замечание 2. Данный подход к решению проблемы отказоустойчивости для гиперболических систем является в достаточной мере очевидным. Менее очевидным является процесс гиперболизации, включая нахождение потоков S_i^* (3). Кроме этого необходим поиск и других направлений решения проблемы отказоустойчивости с помощью вычислительных алгоритмов. Это позволит расширить класс задач, решаемых на системах Экзафлопсной производительности.

Заключение

Прогресс в развитии вычислительной математики, начиная с пятидесятых годов прошлого столетия (времени появления первых ЭВМ), в основном стимулировался развитием вычислительной техники. Достаточно вспомнить алгоритмы, которые появились буквально в течение нескольких лет на стыке сороковых и пятидесятых годов 20 столетия. Их фактически независимое открытие было обусловлено ЭВМ, появившимися в это время в СССР и США. При этом на факт обратного влияния алгоритмов прикладной математики на развитие самой вычислительной техники, практически не обращалось внимание. В данной работе сделана попытка взглянуть на проблему сосуществования алгоритмов прикладной математики и вычислительной техники под другим углом. По мнению авторов данной работы, будущий прогресс вычислительных систем будет определяться, как развитием и появлением средств выполнения вычислений, включая их техническую реализацию и появления новых физических идей, лежащих в их основе, так и развитием новых подходов и идей в прикладной математике. Не исключено, что эти принципиально новые направления вычислительной математики будут широко использовать базовые знания из смежных областей науки физики и механики.

Библиографический список

[1] *Давыдов А.А., Лацис А.О., Луцкий А.Е., Смольянов Ю.П., Четверушкин Б.Н., Шильников Е. В.* Многопроцессорная вычислительная система гибридной архитектуры «МВС-Экспресс». ДОКЛАДЫ АКАДЕМИИ НАУК, 2010, том 434, № 4, с. 459-463.

[2] Гибридный вычислительный кластер K-100/ URL: http://www.kiam.ru/MVS/resourses/k100.html. Сетевой ресурс.

[3] *Горобец А.В., Суков С.А., Железняков А. О., Богданов П.Б., Четверушкин Б.Н.* Расширение двухуровневого распараллеливания MPI+openmp посредством opencl для газодинамических расчетов на гетерогенных системах // Вестник Южно-Уральского государственного университета. Серия Математическое моделирование и программирование. — 2011. — № 9. — С. 76–86.

[4] *Bland W.* // International Journal of High Performance Computing Applications. 2013. Vol. 27. No. 3. P. 244–254.

[5] *Cappello F.* // International Journal of High Performance Computing Applications. — 2009. Vol. 23. No. 3. P. 212–226.

[6] *Cappello F., Geist A., Gropp W., Kale S., Kramer B., Snir M.* // Supercomputing frontiers and innovations. 2014. Vol.1, No. 1. P. 1–28.

[7] *Marc Snir, et al.* // International Journal of High Performance Computing Applications. May 2014. Vol. 28. No. 2. P. 129–173.

[8] *Четверушкин Б. Н.* Кинетические схемы и квазигазодинамическая система уравнений. М.: Макс Пресс, 2004, 328 стр.

[9] *Четверушкин Б. Н.* Пределы детализации и формулировка моделей уравнений сплошных сред // Математическое моделирование. — 2012. — Т. 24, № 11. — С. 33–52.
[10] *Седов Л. И.* Методы подобия и размерности в механике. Наука, 8-е издание. 1977г., 440 с.
[11] *Злотник А. А., Четверушкин Б. Н.* О балансе энтропии для одномерной гиперболической квазигазодинамической системы уравнений. Доклады РАН, 2017, т. 474, №1, с. 22-27.
[12] *Chetverushkin B. N., Zlotnik A. A.* On some properties of multidimensional hyperbolic quasi-gasdynamic systems of equations // Russian Journal of Mathematical Physics, July 2017, Volume 24, Issue 3, pp 299–309, DOI 10.1134/S1061920817030037
[13] *Д'Асчензо Н., Савельев В. И., Четверушкин Б. Н.* Об одном алгоритме решения параболических и эллиптических уравнений // Ж. вычисл. матем. и матем. физ., 55:8 (2015), 1320–1328; Comput. Math. Math. Phys., 55:8 (2015), 1290–1297
[14] *Chetverushkin B., D'Ascenzo N., Saveliev A., Saveliev V.* A kinetic model for magnetogasdynamics // Matem. Mod., 29:3 (2017), 3–15; Math. Models Comput. Simul., 9:5 (2017), 544–553
[15] *Злотник А. А., Четверушкин Б. Н.*, О параболичности квазигазодинамической системы уравнений, ее гиперболической 2-го порядка модификации и устойчивости малых возмущений для них // Ж. вычисл. матем. и матем. физ., 48:3 (2008), 445–472; Comput. Math. Math. Phys., 48:3 (2008), 420–446
[16] *Четверушкин Б. Н., Якобовский М. В.* Вычислительные алгоритмы и отказоустойчивость гиперэкзафлопсных вычислительных систем // Доклады Академии наук. — 2017. — Т. 472, № 1. — С. 1–5. [DOI: 10.7868/S0869565217010042]
[17] *Бондаренко А.А., Якобовский М.В.* Моделирование отказов в высокопроизводительных вычислительных системах в рамках стандарта MPI и его расширения ULFM// Вестник Южно-Уральского государственного университета. Серия «Вычислительная математика и информатика», том 4, № 3, 2015, с. 5-12.
URL: http://www.mathnet.ru/links/76f8ebac383f6603304d157a36123407/vyurv1.pdf
[18] *Bondarenko A.A., Kornilina M.A., Yakobovskiy M.V.* Fault tolerant algorithm for HPC //Supercomputing in Scientific and Industrial Problems. German-Russian. – 2016.
URL: http://www.kiam.ru/SSIP/SSIP_abstracts.pdf#page=9