МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ 2019 год, том 31, номер 4, стр. 17-32

# ПРОИЗВОДИТЕЛЬНОСТЬ ОТЕЧЕСТВЕННОГО ПРОЦЕССОРА ЭЛЬБРУС-8С В СУПЕРКОМПЬЮТЕРНОМ МОДЕЛИРОВАНИИ ЗАДАЧ ВЫЧИСЛИТЕЛЬНОЙ ГАЗОВОЙ ДИНАМИКИ

© *2019 г.* ***А.В. Горобец[1], М.И. Нейман-заде[2], С.К. Окунев[2], А.А. Калякин[2], С.А. Суков[1]***

[1]ИПМ им. М.В. Келдыша РАН, Москва andrey.gorobets@gmail.com
[2]АО "МЦСТ", Москва

Работа выполнена при поддержке Программы фундаментальных исследований Президиума РАН № 26 по приоритетным направлениям, определяемым президиумом РАН, на 2018 год «Фундаментальные основы создания алгоритмов и программного обеспечения для перспективных сверхвысокопроизводительных вычислений».

DOI: 10.1134/S0234087919040026

Исследуется производительность отечественного процессора Эльбрус-8С на расчетах задач вычислительной газовой динамики. Рассматриваются параллельные программные комплексы на основе методов повышенной точности на неструктурированных сетках для численного моделирования турбулентных течений. Описаны особенности архитектуры Эльбрус, а также подходы к адаптации и оптимизации программ. Производительность исследована как для алгоритмов в целом, так и для основных составляющих алгоритмов операций в отдельности. Представлены результаты сравнительного тестирования с различными многоядерными процессорами Intel и AMD.

Ключевые слова: вычислительная газовая динамика, высокопроизводительные вычисления, Эльбрус, CFD, суперкомпьютерное моделирование.

## PERFORMANCE OF ELBRUS-8C PROCESSOR IN SUPERCOMPUTER CFD SIMULATIONS

***A.V. Gorobets[1], M.I. Neiman-zade[2], S.K. Okunev[2], A.A. Kalyakin[2], S.A. Soukov[1]***

[1]Keldysh Institute of Applied Mathematics of RAS, Moscow
[2]MCST, Moscow

The present work is devoted to performance evaluation of a multicore CPU Elbrus-8C in supercomputer computational fluid dynamics (CFD) applications. Parallel simulation codes based on higher-accuracy methods on unstructured meshes for modelling of turbu-

lent flows are considered. Main features of the Elbrus architecture are described, approaches for adaptation and optimization of computing software are presented. Performance has been investigated for the whole algorithms and for their main operations separately. Benchmarking results in comparison with various Intel and AMD multicore CPUs are presented.

Keywords: CFD, HPC, Elbrus processor, supercomputer simulations, performance.

## 1. Введение

Задачи вычислительной газовой динамики являются одним из наиболее распространенных классов суперкомпьютерных приложений. Такие расчеты широко востребованы в различных областях науки и промышленности, в особенности, в авиационно-космической отрасли. Интенсивный рост производительности суперкомпьютеров открывает все более широкие возможности применения численного моделирования. При этом производительность процессоров повышается за счет увеличения как числа ядер, так и числа операций, выполняемых ядром за такт.

Увеличение числа операций с плавающей точкой за такт достигается различными способами. Во-первых, это расширение векторных регистров процессора. Во-вторых, это расширение суперскалярности, то есть увеличение числа исполнительных устройств, которые могут одновременно выполнять независимые по данным операции.

В подавляющем большинстве современных процессоров загрузку множественных исполнительных устройств обеспечивает ядро процессора за счет возможности внеочередного исполнения команд. Дополняет этот механизм поддержка одновременной многопоточности, которая позволяет ядру выполнять инструкции сразу из нескольких потоков. За счет этого увеличивается число поступающих на исполнение независимых команд и повышается коэффициент загрузки исполнительных устройств ядра.

В отличие от большинства распространенных западных аналогов, отечественные процессоры Эльбрус являются представителями архитектуры VLIW (very long instruction word – очень длинная машинная команда). VLIW команда содержит в себе несколько операций для параллельного выполнения суперскалярным ядром процессора. В такой архитектуре задача распределения команд по исполнительным устройствам возложена на компилятор. Ядро процессора Эльбрус-8С имеет шесть 64-битных FMA (fused multiply-add – совмещенное умножение-сложение) устройств, что позволяет выполнять до 12 FLOP (floating point operation – арифметическая операция с числами с плавающей точкой) с аргументами двойной точности 64-битного формата FP64 за такт.

Для оценки производительности процессоров широко используются различные эталонные тестовые алгоритмы – бенчмарки. Наиболее распространенным является бенчмарк LINPACK [1], на котором основан мировой рейтинг суперкомпьютеров TOP500. В этом тесте решается СЛАУ с плотной матрицей прямым методом на основе LU разложения, который имеет вычислительную стоимость O($N^3$) FLOP и потребление оперативной памяти O($N^2$) байт, где $N$ – число неизвестных в системе. Операции с плотными матрицами удобны для процессора, что позволяет получать высокий процент от пиковой производительности, достигающий 90%. Однако данный тест не считается репрезентативным, поскольку он далек по своему паттерну вычислений и доступа к памяти от подавляющего большинства реальных суперкомпьютерных приложений, производительность в которых многократно ниже.

Другой крайностью является тест HPCG [2], в котором СЛАУ с сильно разреженной матрицей решается итерационным методом сопряженных градиентов с предобуславливателем на основе многосеточного метода. Тест имеет одинаковую асимптотику вычислительной стоимости и потребления памяти O($N$). В этом тесте используются весьма проблематичные для процессора операции с разреженными матрицами нерегулярной структуры, которые характерны для расчетов на трехмерной неструктурированной сетке. Таким операциям свойственны крайне низкая вычислительная плотность (порядка 1 FLOP на FP64 аргумент из памяти) и нерегулярный доступ к памяти, сопряженный с частыми кэш промахами. Этот тест намного ближе к реальным приложениям, чем LINPACK, но представляется наоборот излишне жестким по отношению к процессору. Достигаемая на нем производительность в пределах всего нескольких процентов от пика, как правило, в разы ниже, чем в реальных приложениях.

Также существуют различные бенчмарки из наборов репрезентативных операций, характерных для задач трехмерного моделирования, как, например, NAS Parallel Benchmarks [3]. Исследование архитектуры Эльбрус на таком наборе тестов представлено в [4]. Аналогичные результаты для отечественных процессоров КОМДИВ для сравнения можно найти в [5]. Результаты таких тестов более информативны, поскольку включают операции различных типов. Тем не менее такие тестовые операции являются сильно упрощенными, а получаемая производительность даже для близких по структуре алгоритмов зачастую далека от реальных приложений, имеющих гораздо более сложную логику и разнородные вычислительные операции.

В данной работе исследование производительности выполнено именно на примере реальных суперкомпьютерных приложений. Используются про-

граммные комплексы для трехмерного моделирования задач газовой динамики на неструктурированных сетках. Алгоритмы такого класса ориентированы на промышленные задачи. Использование гибридных неструктурированных сеток позволяет моделировать обтекание тел сложной формы, таких как планер летательного аппарата, крыло с механизацией, винт вертолета, вентилятор авиадвигателя и т.д.

Далее будут рассмотрены различные типы алгоритмических операций, оценена производительность процессора Эльбрус относительно западных аналогов Intel и AMD, приведены примеры оптимизации вычислений.

## 2. Программное обеспечение для суперкомпьютерного моделирования

Рассматриваются программные комплексы, предназначенные для расчетов задач аэродинамики и аэроакустики на неструктурированных сетках, которые имеют следующие общие особенности:

1. Низкая вычислительная интенсивность, примерно 1–2 FLOP на байт, при которой производительность главным образом ограничена пропускной способностью памяти. Это заметно выше, чем на операциях линейной алгебры с разреженными матрицами (около 1/8 FLOP на байт), но, с другой стороны, в несколько раз ниже, чем соотношение производительности и пропускной способности памяти современных процессоров (около 10).

2. Большой объем обрабатываемых данных, ассоциированных с элементами расчетной области – узлами сетки, гранями контрольных объемов, сеточными элементами и т.д., многократно превышающий объем кэш памяти процессора.

3. Нерегулярный доступ к памяти, косвенная адресация при доступе к инцидентным элементам расчетной области, что обусловлено дискретизацией на неструктурированной сетке. При таком паттерне доступа существенное влияние оказывает латентность доступа к памяти. Соответственно, от процессора требуется эффективная работа кэш памяти и механизмов сокрытия латентности.

Для распараллеливания используется двухуровневый подход [6], сочетающий параллельные модели с распределенной и общей памятью, на основе стандартов MPI (Message passing interface – интерфейс передачи сообщений) и OpenMP соответственно.

**2.1. Программный комплекс NOISEtte.** Основной областью приложений NOISEtte [7] является суперкомпьютерное моделирование задач аэродинамики и аэроакустики, характерных главным образом для авиационно-космической отрасли.

Течение вязкого идеального газа описывается системой уравнений На-

вье–Стокса. Для численного решения используется конечно-объемный метод на основе экономичных схем повышенной точности с квазиодномерной реконструкцией [8]. Интегрирование по времени осуществляется неявной схемой с линеаризацией по Ньютону. Для решения СЛАУ с матрицей Якоби используется предобусловленный метод Bi-CGSTAB [9]. NOISEtte имеет комбинированное MPI+OpenMP распараллеливание. Более подробно параллельный алгоритм и показатели параллельной эффективности в расчетах на различных суперкомпьютерах представлены в [10]. Для анализа производительности в алгоритме были выделены следующие основные ресурсоемкие операции.

**Расчет конвективных потоков** представляет собой цикл по множеству внутренних граней ячеек, ассоциированных с ребрами сетки. Для расчета потока через грань *f* между ячейками вокруг узлов *i* и *j* (рис. 1) по заданным в узлах значениям переменных (плотность, компоненты вектора скорости, давление) выполняется реконструкция значений "слева" и "справа" от центра грани, точки *L* и *R*. Для этого используется интерполяционная конструкция вдоль прямой *e*, проходящей через узлы *i* и *j*, которая включает узлы из трех уровней соседства. В трехмерном случае шаблон состоит из 14 узлов. Затем для наборов значений *L* и *R* решается вычислительно-интенсивная задача Римана о распаде разрыва для нахождения результирующего потока через грань *f*, а также рассчитывается вклад грани в коэффициенты приближенной матрицы Якоби.

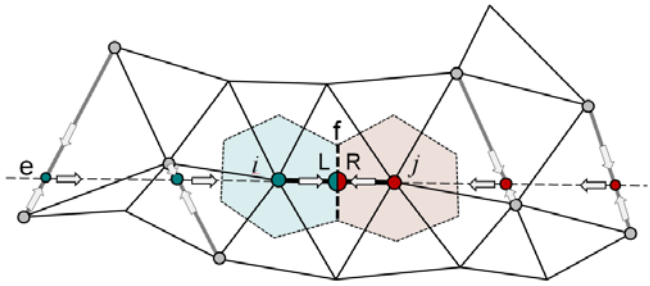


**Рис.1.** Шаблон схемы расчета потока через грань *f* между ячейками *i* и *j*.

Особенностью данной операции является выборка данных из узлов сетки путем косвенной адресации к данным, расположенным в памяти произвольным образом. Также для решения задачи Римана используется сложная программная логика с большим количеством ветвлений и несколькими уровнями вызовов подпрограмм.

На вход для каждой грани из узлов поступают 70 вещественных значений переменных (по 5 из 14 узлов), 10 значений суммарных потоков в узлах

(по 5 из двух узлов), на выходе записываются 10 значений суммарных потоков (по 5 в два узла). Также считываются и записываются 100 значений коэффициентов в матрицы Якоби (4 блока по 25 коэффициентов). Вклад данной операции в общее время вычислений составляет около 20%.

**Расчет вязких потоков** представлен в виде цикла по ребрам графа обратной топологии (граф, в котором вершинам сопоставлены узлы сетки, а ребрам между двумя вершинами соответствует общий сеточный элемент, содержащий соответствующие вершинам узлы). Для каждого ребра также рассчитывается вклад в потоки в узлах и в матрицу Якоби.

На вход из узлов поступают 10 значений переменных и 10 значений суммарных потоков (по 5 из двух узлов), на выходе также 10 значений суммарных потоков и 100 значений коэффициентов в матрицу Якоби в случае неявной схемы. Вклад расчета вязких потоков в общее время составляет около 30%.

**Матрично-векторное произведение** (МВП) является основной ресурсоемкой операцией итерационного метода Bi-CGSTAB, который используется в неявной схеме с линеаризацией по Ньютону для решения СЛАУ с матрицей Якоби. Для представления матрицы используется блочный построчно-разреженный формат CSR (Compressed Sparse Row). Размер блока равен числу физических переменных (5 в трехмерном случае). Число ненулевых блоков в строке зависит от типа сеточных элементов. Для тетраэдральной сетки это число в среднем около 15, а объем данных на вход для каждой строки составляет 375 значений коэффициентов, расположенных в памяти компактно, и 75 значений входного вектора в 15 блоках, расположенных в памяти произвольным образом. В выходной вектор записывается 5 значений. Вклад данной операции в общее время вычислений составляет около 20%.

**2.2. Программный комплекс Tapir.** Tapir предназначен для расчета дозвуковых и сверхзвуковых течений вязкого сжимаемого газа. Дискретизация уравнений Навье–Стокса выполняется на основе метода конечного объема с определением значений сеточных функций в центрах масс элементов неструктурированной сетки. Конвективный поток через грани контрольных объемов определяется с использованием одной из схем (HLLE, HLLC, AUSM, схема Роу и т.д., см. [11]) решения задачи Римана о распаде произвольного разрыва. Повышение точности пространственной аппроксимации достигается за счет полиномиальной реконструкции. Коэффициенты линейных или квадратичных полиномов определяются на основе дискрет-

ного аналога формулы интегрального представления градиента или метода наименьших квадратов. Интегрирование по времени выполняется явным методом. Более подробное описание алгоритма и его программной реализации для различных платформ можно найти в [12]. Для замеров были выбраны следующие основные ресурсоемкие операции.

**Расчет градиентов** выполняется в цикле по ячейкам. На вход для каждой ячейки поступает от 40 (для тетраэдра) до 56 (для гексаэдра) вещественных значений: 5 физических переменных в ячейке, по 5 переменных из соседних ячеек, по 3 геометрических коэффициента для самой ячейки и ее соседей. Геометрические коэффициенты ячейки расположены в памяти линейно, а для косвенной адресации к значениям в соседних ячейках используется целочисленный массив, хранящий адреса соседних ячеек. На выходе записывается 15 значений градиентов. Данная операция имеет низкую вычислительную интенсивность, близкую к МВП. Вклад данной операции в общее время составляет 10–20%.

**Расчет потоков –** это основная операция алгоритма, представленная в виде цикла по граням ячеек. Для каждой грани сначала определяются реконструированные значения переменных с двух сторон от центра масс грани, затем вычисляются конвективный и вязкий потоки, для чего вызываются соответствующие вычислительно-емкие подпрограммы. На вход поступают 12 вещественных значений параметров грани (координаты центра масс, площадь, компоненты нормали и т.д.) и 2 целочисленных значения адресов инцидентных грани ячеек. Из этих двух ячеек поступает 46 вещественных значений: по 3 координаты центра массы, по 5 значений переменных и по 15 значений градиентов. На выходе записывается 5 значений результирующего потока через грань. Эта операция имеет более высокую вычислительную стоимость, вклад в общее время составляет 60–80%.

## 3. Особенности архитектуры процессора Эльбрус-8С

Восьмиядерный процессор Эльбрус-8С принадлежит семейству архитектур VLIW. Каждое ядро может исполнять до 25 различных элементарных операций в одном такте. Структура широкой команды (ШК) позволяет разместить до 6 арифметико-логических операций. Соответственно, для вещественной арифметики имеется 6 арифметико-логических устройств с полностью конвейеризированными устройствами умножения и сложения, позволяющих выполнять до 12 FLOP за такт с аргументами двойной точности. Также имеется одно частично конвейеризированное устройство деления и квадратного корня, позволяющее запускать одну из этих операций один раз

в 2 такта. Более подробно описание архитектуры представлено в [13, 14].

Подсистема памяти имеет 4 канала DDR3 1600 и обеспечивает предельный темп работы 51 GB/s. Иерархия кэш-памяти представлена кэш:

– данных первого уровня, 64 KB на каждое ядро, 4-way;

– инструкций первого уровня, 128 KB на каждое ядро, 4-way;

– второго уровня, 512 KB на каждое ядро, 4-way;

– третьего уровня, инклюзивным, общим для 8 ядер, 16 MB, 16-way.

До 4 процессоров можно объединять в NUMA-систему, в которой для связи используются каналы пропускной способностью до 8 GB/s в каждом направлении. Более подробное описание работы подсистемы памяти процессора представлено в [15].

Традиционно для архитектур VLIW обеспечение производительности исполнения кода во многом переносится на этап оптимизирующей компиляции. Система программирования включает оптимизирующий компилятор lcc собственной разработки для языков C/C++/Fortran, предназначенный для выявления параллелизма операций и планирования широких команд, а также использования аппаратных возможностей процессоров Эльбрус. При этом для удобства использования обеспечена совместимость с системой программирования GNU Compiler Collection версии 5.5.0 по опциям и основным расширениям стандартов языков.

## 4. Адаптация программы к архитектуре Эльбрус

Основными особенностями VLIW архитектуры являются:

– повышенные требования к возможности инлайн-подстановки процедур с целью расширения контекста для поиска параллелизма операций;

– более высокий эффект от межмодульной (или полнопрограммной) оптимизации по сравнению с другими архитектурами;

– двухфазная компиляция с генерацией профиля путем тестового запуска задачи и использованием этого профиля при повторной компиляции;

– более высокая отзывчивость к разрешению конфликтов между операциями обращения к памяти.

Анализ производительности кодов NOISEtte и Tapir на вычислительном комплексе Эльбрус-801 показал ряд схожих проблем:

– из-за объёма данных кэш не может полностью обеспечить быстрый доступ, есть проблема с блокировками по времени чтения из памяти;

– есть плохо разрешаемые зависимости между указателями;

– в горячих участках кода присутствуют циклы с малым переменным числом итераций, не позволяющие провести ни оптимизацию полной раскрутки, ни эффективную программную конвейеризацию;

– есть межмодульные вызовы процедур небольшого размера, инлайн-подстановка которых была бы полезна.

Для борьбы с этими негативными эффектами были использованы следующие настройки компилятора:

– опция -fcache-opt, включающая режим конвейеризации с увеличенным расстоянием от операций чтения до момента использования их результатов;

– опции -frestrict-all, -frestrict-params, устанавливающие спецификатор restrict на локальные указатели и параметры процедуры;

– режим работы межмодульной оптимизации -fwhole в сочетании с расстановкой атрибутов функций __attribute((always_inline));

– опции -fprofile-generate и -fprofile-use для двухфазной компиляции;

– подкачка данных в кэш в сочетании с опцией -fset-ld-delay=13 для выдерживания дистанции при получении данных из кэш;

– опция выбора целевой архитектуры -mcpu=elbrus-v4;

– стандартные флаги оптимизации -O4 -ffast -ffast-math.

Также были внесены точечные модификации исходного кода в наиболее важных с точки зрения производительности участках. Большинство модификаций было связано с подстановкой подкачки данных путем обращения к встроенной функции __builtin_prefetch, с помощью чего снижаются потери на ожидание поступления данных из памяти.

Для помощи компилятору в разрешении зависимости между указателями в некоторых местах добавлены ключевые слова __restrict.

Выполнены модификации кода для устранения зависимостей при доступе к локальным данным. Пример устранения конфликта между двумя фрагментами кода, мешавшего обоим фрагментам выполняться одновременно, приведен в табл.1.

Выполнена точечная ручная модификация малоитерационных циклов с целью выравнивания числа итераций для последующей автоматической полной раскрутки циклов на этапе компиляции. Пример модификации циклов приведен в табл.2. Устранение условного выхода из цикла сделало его доступным для полной раскрутки. Перестановка порядка обхода n-k в гнездах необходима для разрыва зависимостей после раскрутки гнёзд и более плотной упаковки кода внешнего цикла.

Изменения в кодах NOISEtte и Tapir суммарно затронули всего несколько десятков строк. Для кода NOISEtte изменения отдельных фрагментов дали ускорение на архитектуре Эльбрус около 1.3 раз. Для кода Tapir выполнены модификации всех ресурсоемких операций, что дало ускорение

более 2 раз. При этом изменения не оказали заметного эффекта на процессорах Intel, имеющих внеочередное исполнение команд.

**Таблица 1.** Пример устранения зависимости между участками кода.

| Исходная версия | Модифицированная версия |
|---|---|
| `double dXYZ[3];`<br>`for(j=0; j<3; ++j)`<br>`  dXYZ[j]=fMC[j]-CEL[j];`<br>`for(j=0; j<5; ++j)`<br>`  ql[j]=QL[j]+dQL[j][0]*dXYZ[0]`<br>`  +dQL[j][1]*dXYZ[1]`<br>`  +dQL[j][2]*dXYZ[2];`<br>`//зависимость по dXYZ между частями`<br>`for(j=0; j<3; ++j)`<br>`  dXYZ[j]=fMC[j]-CER[j];`<br>`for(j=0; j<5; ++j)`<br>`  qr[j]=QR[j]+dQR[j][0]*dXYZ[0]`<br>`  +dQR[j][1]*dXYZ[1]`<br>`  +dQR[j][2]*dXYZ[2];` | `{double dXYZ[3];`<br>`  for(j=0; j<3; ++j)`<br>`  dXYZ[j] = fMC[j]-CEL[j];`<br>`  for(j=0; j<5; ++j)`<br>`  ql[j]=QL[j]+dQL[j][0]*dXYZ[0]`<br>`  +dQL[j][1]*dXYZ[1]`<br>`  +dQL[j][2]*dXYZ[2];}`<br><br>`{double dXYZ[3];`<br>`  for(j=0; j<3; ++j)`<br>`  dXYZ[j]=fMC[j]-CER[j];`<br>`  for(j=0; j<5; ++j)`<br>`  qr[j]=QR[j]+dQR[j][0]*dXYZ[0]`<br>`  +dQR[j][1]*dXYZ[1]`<br>`  +dQR[j][2]*dXYZ[2];}` |

**Таблица 2.** Пример модификации циклов.

| Исходная версия | Модифицированная версия |
|---|---|
| `//цикл с переменной границей`<br>`//от 4 до 6 (посредством break)`<br>`for(int j=0; j<6; ++j){`<br>`  int ic = IC[i*6+j];`<br>`  if(ic == -1) break;`<br>`  for(int n=0; n<5; ++n)`<br>`  for(int k=0; k<3; ++k)`<br>`  dQ[n][k] += Q[n]*kXYZ[k];`<br>`}` | `//цикл с постоянной границей`<br>`//лишние слагаемые зануляются`<br>`double C=1.0;`<br>`for(int j=0; j<6; ++j){`<br>`  int ic = IC[i*6+j];`<br>`  if(ic == -1){ic=0; C=0.0;}`<br>`  for(int k=0; k<3; ++k)`<br>`  for(int n=0; n<5; ++n)`<br>`  dQ[n][k] += C*Q[n]*kXYZ[k];`<br>`}` |

## 5. Сравнение производительности

В тестовых расчетах моделируется течение вязкого сжимаемого газа на небольших сетках, число элементов в которых примерно соответствует обычной характерной нагрузке на процессор в расчетах больших задач на кластерных системах. Тесты кодом NOISEtte выполняются на неструктурированной сетке, состоящей из 119 тыс. узлов и 679 тыс. элементов. Используется неявная схема по времени с линеаризацией по Ньютону и схема EBR5 повышенной точности по пространству [8]. Для решения задачи распада разрыва используется схема Roy [16].

Для кода Tapir выбрана сетка из 273 тыс. узлов и 445 тыс. элементов. Используется схема с полиномиальной реконструкцией (линейные полино-

мы) по пространству и явная схема Эйлера первого порядка по времени. Для распада разрыва используется схема Роу [16].

Измеряются следующие алгоритмические операции (см. раздел 2):

– *NOISEtte Total* – расчет кодом NOISEtte, весь алгоритм;

– *NOISEtte CFlux* – операция расчета конвективных потоков;

– *NOISEtte VFlux* – операция расчета вязких потоков;

– *NOISEtte SpMV* – МВП с разреженной матрицей в решателе;

– *Tapir Total* – расчет кодом Tapir, весь алгоритм;

– *Tapir Grad* – операция расчета градиентов сеточных функций;

– *Tapir Flux* – операция расчета потоков.

Для сравнительного тестирования (табл.3) выбраны два близких аналога, работающих с памятью DDR3, AMD Opteron 6276 и Intel Xeon E5-2650v2 (Ivy Bridge), и новые процессоры, работающие с памятью DDR4: Intel Xeon E5-2683v4 (Broadwell) и Intel Xeon Platinum 8160 (Skylake).

**Таблица 3.** Основные характеристики рассматриваемых процессоров: число ядер, тактовая частота, ГГц, пиковая производительность, GFLOPS, пропускная способность памяти, GB/s, энергопотребление, Вт, техпроцесс, нм.

| CPU | ядер | ГГц | GFLOPS | GB/s | Вт | нм |
|---|---|---|---|---|---|---|
| Эльбрус-8С | 8 | 1.3 | 125 | 51 | 80 | 28 |
| AMD Opteron 6276 (Interlagos) | 16 | 2.3 | 147 | 51 | 115 | 32 |
| Intel Xeon E5-2650v2 (Ivy Bridge) | 8 | 2.6 | 166 | 60 | 95 | 22 |
| Intel Xeon E5-2683v4 (Broadwell) | 16 | 2.1 | 294 | 77 | 120 | 14 |
| Intel Xeon Platinum 8160 (Skylake) | 24 | 2.1 | 1612 | 128 | 150 | 14 |

**Параллельное ускорение** показывает, как ускоряется расчет в многопоточном режиме относительно последовательного исполнения на том же процессоре. Результаты ускорения на 8 ядрах представлены в табл.4. Ускорение на процессоре Эльбрус-8С (5–6 раз) в целом хорошо соответствует западным аналогам. Skylake демонстрирует одинаково высокое ускорение на всех операциях благодаря более мощной подсистеме памяти, имеющей 6 каналов DDR4.

**Сравнение одиночного ядра** показывает соотношение производительности в последовательном режиме с процессором Эльбрус-8С, скорость работы которого принята за единицу (табл.5). На коде NOISEtte разница с процессорами Intel составила около 3 раз. В пересчете на такт это соответствует разнице около полутора раз. Относительно AMD проигрыш составил около 1.4 раз. На коде Tapir разница с Intel составила около полутора раз. Таким образом, на данном приложении Эльбрус-8С демонстрирует более

высокую производительность на такт, чем аналоги Intel. Ядро AMD оказалось медленнее Эльбруса примерно в полтора раза.

**Таблица 4.** Параллельное ускорение, 8 нитей OpenMP на 8 ядрах.

| Операция | Эльбрус-8С | AMD 6276 | Intel Ivy Bridge | Intel Broadwell | Intel Skylake |
|---|---|---|---|---|---|
| NOISEtte CFlux | 6.4 | 6.1 | 6.3 | 5.9 | 7.7 |
| NOISEtte VFlux | 6.7 | 5.3 | 5.5 | 5.6 | 7.5 |
| NOISEtte SpMV | 4.4 | 1.7 | 3.7 | 4.7 | 7.0 |
| **NOISEtte Total** | **5.8** | **3.5** | **5.2** | **5.4** | **7.2** |
| Tapir Grad | 1.8 | 3.4 | 4.4 | 4.3 | 7.2 |
| Tapir Flux | 6.7 | 5.3 | 6.8 | 6.2 | 7.6 |
| **Tapir Total** | **5.0** | **4.6** | **6.2** | **5.8** | **7.5** |

**Таблица 5.** Производительность относительно Эльбрус-8С: одно ядро.

| Операция | AMD 6276 | Intel Ivy Bridge | Intel Broadwell | Intel Skylake |
|---|---|---|---|---|
| NOISEtte CFlux | 1. 40 | 3.43 | 3.63 | 2.97 |
| NOISEtte VFlux | 1.20 | 2.75 | 2.53 | 1.80 |
| NOISEtte SpMV | 0.90 | 2.14 | 1.73 | 1.82 |
| **NOISEtte Total** | **1.27** | **2.82** | **2.80** | **2.28** |
| Tapir Grad | 0.52 | 1.13 | 1.30 | 0.84 |
| Tapir Flux | 0.63 | 1.50 | 1.67 | 1.31 |
| **Tapir Total** | **0.68** | **1.58** | **1.75** | **1.34** |

**Сравнение производительности на 8 ядрах** показывает производительность вычислений в многопоточном режиме на одинаковом с Эльбрус-8С числе ядер. Результаты по соотношению с Эльбрусом показаны в табл.6. Сравнивая с табл.5, можно отметить, что на коде NOISEtte соотношение изменилось в пользу Эльбруса. Для кода Tapir разница с Intel составила около 2 раз. Эльбрус обогнал AMD на обоих кодах.

**Таблица 6.** Производительность относительно Эльбрус-8С: 8 ядер.

| Операция | AMD 6276 | Intel Ivy Bridge | Intel Broad-well | Intel Skylake |
|---|---|---|---|---|
| NOISEtte CFlux | 1.34 | 3.42 | 3.33 | 3.60 |
| NOISEtte VFlux | 0.94 | 2.25 | 2.13 | 2.02 |
| NOISEtte SpMV | 0.34 | 1.80 | 1.85 | 2.88 |
| **NOISEtte Total** | **0.77** | **2.55** | **2.63** | **2.87** |
| Tapir Grad | 1.01 | 2.81 | 3.22 | 3.48 |
| Tapir Flux | 0.50 | 1.53 | 1.54 | 1.51 |
| **Tapir Total** | **0.62** | **1.95** | **2.01** | **2.00** |

**Сравнение для всего процессора** показывает производительность в многопоточном режиме на всех ядрах (табл.7). Эльбрус-8С обгоняет на коде Tapir 16-ядерный процессор AMD. При этом 16- и 24-ядерные процессоры Intel за счет высокого OpenMP ускорения заметно прибавили относительно 8-ядер, разница с Эльбрусом составила уже примерно 3–7 раз.

**Производительность вычислений** была измерена путем подсчета числа арифметических операций в исходном коде. В качестве примера операции, сильно ограниченной пропускной способностью памяти, была выбрана операция NOISEtte SpMV, имеющая наименьшую вычислительную интенсивность – около 0.2 FLOP на байт, что примерно в 10 раз ниже, чем у кода Tapir. Результаты представлены в табл.8, в которой также приводится процент от теоретического пика устройств.

**Таблица 7.** Производительность относительно Эльбрус-8С: весь процессор.

| Операция | AMD 6276 | Intel Ivy Bridge | Intel Broadwell | Intel Skylake |
|---|---|---|---|---|
| NOISEtte CFlux | 2.56 | 3.42 | 6.41 | 10.25 |
| NOISEtte VFlux | 1.87 | 2.25 | 2.25 | 5.72 |
| NOISEtte SpMV | 0.61 | 1.80 | 2.53 | 4.86 |
| **NOISEtte Total** | **1.39** | **2.55** | **4.17** | **6.82** |
| Tapir Grad | 0.81 | 2.81 | 3.63 | 7.25 |
| Tapir Flux | 0.88 | 1.53 | 2.90 | 4.13 |
| **Tapir Total** | **0.89** | **1.95** | **3.30** | **4.94** |

**Таблица 8.** Производительность, GFLOPS и процент от пика.

| | | Эльбрус-8С | AMD 6276 | Intel Ivy Bridge | Intel Broadwell | Intel Skylake |
|---|---|---|---|---|---|---|
| **NOISEtte SpMV** | 1 ядро | 1.0, 6.5% | 0.9, 9.9% | 2.2, 10% | 1.8, 9.5% | 1.8, 2.7% |
| | 8 ядер | 4.4, 3.6% | 1.5, 2.1% | 8, 4.8% | 8.2, 5.6% | 12.8, 2.4% |
| | Процессор | 4.4, 3.6% | 2.7, 1.8% | 8, 4.8% | 11.2, 3.8% | 21.6, 1.3% |
| **Tapir Total** | 1 ядро | 3.1, 20% | 2.1 , 23% | 4.9, 23% | 5.4, 29% | 4.1, 6.1% |
| | 8 ядер | 15.5, 12% | 9.5, 13% | 30, 18% | 31, 21% | 31, 5.8% |
| | Процессор | 15.5, 12% | 13.7, 9% | 30, 18% | 59, 17% | 77, 4.7% |

Из результатов видно, что на SpMV достигается заметно меньший процент от пика, поскольку пиковая производительность процессора многократно превосходит пропускную способность памяти (см. табл.3). Также можно отметить очень низкий процент от пика у Skylake, имеющего учетверенное число операций на такт по сравнению с предшественником Broadwell. Можно сделать вывод, что на данном классе алгоритмов, в основном из-за ограничений пропускной способности памяти, удвоение FMA устройств и расширение векторных регистров не дало ускорения.

**Параллельное ускорение на 4-процессорном модуле Эльбрус-804** (процессоры с частотой 1.2 ГГц) показывает эффективность вычислений в условиях обмена данными по межпроцессорным соединениям, имеющим пропускную способность 8 GB/s. Для данного тестирования использовались более подробные сетки, чем в случае одиночного процессора. Результаты для MPI+OpenMP режима вычислений представлены в табл.9. Из результатов видно, что OpenMP ускорение выше для более подробной сетки (по сравнению с табл.4). Это связано с тем, что на подробной сетке число кэш промахов выше, за счет чего один поток в меньшей степени задействует пропускную способность памяти, что позволяет получить большее ускорение в многопоточном режиме.

Параллельная эффективность при переходе с одного процессора на 4 в режиме 8 OpenMP нитей на 1 MPI процесс составила около 100% (с точностью до погрешности измерений). Это подтверждает достаточность пропускной способности межпроцессорных связей для данного типа приложений. При этом использование 4 процессоров в режиме 32 нитей OpenMP (без какого-либо учета NUMA факторов при выделении памяти) показало заметно меньшую параллельную эффективность. Например, ускорение кода Tapir на 4 процессорах относительно 1 процессора составило 1.8 раз, что соответствует эффективности 45%. Такая ситуация характерна и для многопроцессорных модулей с процессорами Intel, поэтому такой режим вычислений на практике нецелесообразен и более подробно не исследовался.

**Таблица 9.** Ускорение и параллельная эффективность до 32 ядер Эльбрус-8С.

| Код | Сетка, ячеек | 1 MPI x 8 OpenMP | | 4 MPI x 8 OpenM | | 32 MPI | |
|---|---|---|---|---|---|---|---|
| NOISEtte | 1 млн | 6.8 раз | 85% | 27 раз | 84% | 27.6 раз | 86% |
| Tapir | 3 млн | 7 раз | 87% | 28 раз | 88% | 28.5 раз | 89% |

**Заключение**

Производительность вычислений на 8-ядерных процессорах Эльбрус-8С была исследована на реальных приложениях вычислительной газовой динамики. Использовались два программных комплекса для моделирования сжимаемых течений на неструктурированных сетках, NOISEtte [7] и Tapir [12]. Рассматривались несколько моделей процессоров Intel Xeon, от моделей 5-летней давности до наиболее современных. Эльбрус ожидаемо оказался медленнее. Проигрыш по производительности ядра составил в среднем 2.6 раза для кода NOISEtte и 1.5 раза для кода Tapir. Это представляется хорошим результатом, учитывая, что тактовая частота Эльбрус-8С примерно вдвое ниже. В пересчете на такт Эльбрус не уступает Intel. Кроме того, проигрыш около двух раз на данном классе приложений является типичным даже для процессоров AMD, основного конкурента Intel. По производительности процессора проигрыш относительно Intel на коде NOISEtte со-

ставил от 2.5 раз против 8-ядерного Ivy Bridge до 6.8 раз против 24-ядерного Skylake, а на коде Tapir от 2 до 5 раз, соответственно. Для сравнения был рассмотрен 16-ядерный AMD Opteron 6276 примерно 5-летней давности, соответствующий по времени Intel Ivy Bridge. На коде NOISEtte Эльбрус-8С оказался медленнее 16-ядерного AMD всего в 1.4 раза, а на коде Tapir даже быстрее на 12%.

Можно отметить, что у процессоров Intel на данном типе приложений не растет производительность ядра. В то же время ядро Эльбрус-8С быстрее примерно в полтора раза относительно предыдущего поколения Эльбрус-4С. Это позволяет надеяться, что с выходом нового процессора Эльбрус-16С отставание еще сократится. Ожидается, что следующая модель будет иметь 16 ядер, работающих на частоте 2 ГГц, и до 8 каналов памяти DDR4-2666, что может увеличить пропускную способность памяти более чем в 3 раза. Также большой вклад в рост производительности может внести дальнейшее совершенствование оптимизирующего компилятора.

Данная статья является продолжением работы, представленной в [17].

Авторы благодарят Б.Н. Четверушкина и А.К. Кима за помощь и внимание к данной работе.

СПИСОК ЛИТЕРАТУРЫ

1. *Dongarra J.* The LINPACK Benchmark: An explanation // ICS 1987: Supercomputing. Lecture Notes in Computer Science, v.297, p.456–474.
2. *Dongarra J., Heroux M.A., Luszczek P.* HPCG Benchmark: a New Metric for Ranking High Performance Computing Systems // Technical Report, Electrical Engineering and Computer Science Department, Knoxville, Tennessee, UT-EECS-15-736, 2015. URL: http://www.hpcg-benchmark.org/
3. *Bailey D., Barszcz E., Barton J.T., Browning D.S., Carter R.L., Dagum D., Fatoohi R.A., Frederickson P., Lasinski T.A, Schreiber R., Simon H., Venkatakrishnan V., Weeratunga K.* The NAS Parallel Benchmarks // International Journal of High Performance Computing Applications, 1991, 5, p.63–73.
4. *Тютляева Е.О., Конюхов С.С., Московский А.А., Одинцов И.О.* Оценка потенциала использования платформы Эльбрус для высокопроизводительных вычислений // Суперкомпьютерные дни в России, 2016, с.373–385;
   *Tyutlyayeva Ye.O., Konyukhov S.S., Moskovskiy A.A., Odintsov I.O.* Otsenka potentsiala ispol'zovaniya platformy El'brus dlya vysokoproizvoditel'nykh vychisleniy // Superkomp'yuternyye dni v Rossii, 2016, s.373–385.
5. *Богданов П.Б., Сударева О.Ю.* Производительность процессоров КОМДИВ на ряде типовых расчётных задач // Информационные Технологии и Вычислительные Системы, 2017, т.4, с.104–111;
   *Bogdanov P.B., Sudareva O.Yu.* Proizvoditel'nost' protsessorov KOMDIV na ryade tipovykh raschotnykh zadach // Informatsionnyye Tekhnologii i Vychislitel'nyye Sistemy, 2017, t.4, s.104–111.
6. *Горобец А.В.* Параллельная технология численного моделирования задач газовой динамики алгоритмами повышенной точности // ЖВМиМФ, 2015, т.55, №4, с.641–652; англ. пер: *Gorobets A.V.* Parallel technologies for solving CFD problems using high-accuracy algorithms // Comput. Math. Math. Phys., 2015, v.55, Issue 4, p.638–649.

7. *Абалакин И.В., Бахвалов П.А., Горобец А.В., Дубень А.П., Козубская Т.К.* Параллельный программный комплекс NOISETTE для крупномасштабных расчетов задач аэродинамики и аэроакустики // Вычислит. методы и программ., 2012, t.13, с.110–125;
*Abalakin I.V., Bakhvalov P.A., Gorobets A.V., Duben' A.P., Kozubskaya T.K.* Parallel'nyy programmnyy kompleks NOISETTE dlya krupnomasshtabnykh raschetov zadach aerodinamiki i aeroakustiki // Vychislitel'nyye metody i programmirov., 2012, t.13, c.110–125.
8. *Бахвалов П.А., Козубская Т.К.* О построении реберно-ориентированных схем, обеспечивающих точность на линейной функции, для решения уравнений Эйлера на гибридных неструктурированных сетках // ЖВМиМФ, 2017, т.57. № 4, с.92–111;
англ. пер: *Bakhvalov P.A.*, *Kozubskaya T.K.* Construction of edge-based 1-exact schemes for solving the Euler equations on hybrid unstructured meshes // Comput. Math. Math. Phys., 2017, v.57, Issue 4, p.680–697
9. *Van der Vorst H.A*. A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems // SIAM Journal on Scientific and Statistical Computing, 1992, 13 2, p.631–644.
10. *Gorobets A.* Parallel Algorithm of the NOISEtte Code for CFD and CAA Simulations // Lobachevskii Journal of Mathematics, 2018, 39 4, p.524–532.
11. *Toro E.F*. Riemann Solvers and Numerical Methods for Fluid Dynamics // Springer-Verlag Berlin Heidelberg, 2009. DOI: 10.1007/b79761
12. *Gorobets A., Soukov S., Bogdanov P.* Multilevel parallelization for simulating turbulent flows on most kinds of hybrid supercomputers // Computers and Fluids, 2018, In Press. https://doi.org/10.1016/j.compfluid.2018.03.011
13. *Kozhin A.S., Polyakov N.Y., Alfonso D.M., Demenko R.V., Klishin P.A., Kozhin E.S., Slesarev M.V., Smirnova E.V., Smirnov D.A., Smolyanov P.A., Kostenko V.O., Gruzdov F.A., Tikhorskiy V.V., Sakhin Y.K*. The 5th Generation 28nm 8-Core VLIW Elbrus-8C Processor Architecture // Proceedins of the 2016 International Conference on Engineering and Telecommunication EnT-2016. Moscow, 2016, p.85–89.
14. *Альфонсо Д.М., Деменко Р.В., Кожин А.С., Кожин Е.С., Колычев Р.Е., Костенко В.О., Поляков Н.Ю., Смирнова Е.В., Смирнов Д.А., Смольянов П.А., Тихорский В.В*. Микроархитектура восьмиядерного универсального микропроцессора «Эльбрус-8С» // Вопросы радиоэлектроники, 2016, № 3, сер. ЭВТ, с.6–13;
*Al'fonso D.M., Demenko R.V., Kozhin A.S., Kozhin Ye.S., Kolychev R.Ye., Kostenko V.O., Polyakov N.Yu., Smirnova Ye.V., Smirnov D.A., Smol'yanov P.A., Tikhorskiy V.V.* Mikroarkhitektura vos'miyadernogo universal'nogo mikroprotsessora «El'brus-8C» // Voprosy radioelektroniki, 2016, № 3, ser. EVT, s.6–13.
15. *Кожин А.С., Нейман-заде М.И., Тихорский В.В*. Влияние подсистемы памяти восьмиядерного микропроцессора «Эльбрус-8С» на его производительность // Вопросы радиоэлектроники, 2017, № 3, сер. ЭВТ, с.13–21;
*Kozhin A.S., Neyman-zade M.I., Tikhorskiy V.V*. Vliyaniye podsistemy pamyati vos'miyadernogo mikroprotsessora «El'brus-8S» na yego proizvoditel'nost' // Voprosy radioelektroniki, 2017, № 3, ser. EVT, s.13–21.
16. *Roe P.L.* Approximate Riemann Solvers, Parameter Vectors and Difference Schemes // J. Comput. Phys., 1981, 43, N2. 357–372.
17. *Горобец А.В., Нейман-заде М.И., Окунев С.К., Калякин А.А., Суков С.А*. Производительность процессора Эльбрус-8С в суперкомпьютерных приложениях вычислительной газовой динамики // Препринт ИПМ им. М.В. Келдыша, № 152. – М.: 2018. DOI:10.20948/prepr-2018-152;
*Gorobets A.V., Neiman-zade M.I., Okunev S.K., Kaliakin A.A., Sukov S.A*. Proizvoditelnost protsessora Elbrus-8S v superkompiuternykh prilozheniiakh vychislitelnoi gazovoi dinamiki // Preprint IPM im. M.V. Keldysha, № 152. – M.: 2018. DOI:10.20948/prepr-2018-152.

Поступила в редакцию 09.08.2018
После доработки 09.08.2018
Принята к публикации 22.10.2018