# ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ МОДЕЛИ ЗЕМНОЙ СИСТЕМЫ ИВМ РАН

**В.В. Калмыков**

**Введение**

Как отмечалось на Всемирном саммите по моделированию и прогнозированию климата [1], существует общее согласие о том, что гораздо более высокое разрешение моделей основных компонент (атмосфера, океан, лед, суша) является главной предпосылкой для реалистичного представления климатической системы и актуальности прогнозов. В настоящее представляется, что перспективная модель Земной системы должна иметь горизонтальное разрешение около 1 км.

Прогнозирование климата является одной из самых вычислительно требовательных задач в науке за счет огромных размеров данных и числа операций над ними. Наряду с созданием моделей отдельных компонент Земной системы, становится чрезвычайно важной роль инструмента, организующего их совместную работы — *каплера (*от англ. coupler). К его основным задачам относится синхронизация моделей, интерполяция данных между различными сетками компонент и работа с файловой системой. Кроме того, при со создании совместной системы возникает проблема объединения кода отдельно разрабатываемых моделей с делегированием общих для них функций каплеру. От уровня абстракции архитектуры системы зависит простота подключения новой компоненты, прозрачность работы в ансамбле и степень изменений кода при модификации или замене некоторой модели.

Существующие сегодня системы совместного моделирования можно разделить по следующим критериям: поддерживается ли полный или частичный цикл жизни модели, является ли каплер параллельным или выполняется на одном процессе, какая схема используется для работы с файловой системой, привязана ли структура к определенным моделям Земной системы или представляет только общие для них функции, имеет ли программа вид единого исполняемого файла или использует возможность одновременного запуска нескольких и т.д.

Теоретической основой наших разработок стал пакет MCT (Model Coupling Toolkit) и построенный на его основе каплер для модели NCAR CESM (The National Center for Atmospheric Research Community Earth System Model). Анализ работ [2],[3] по возможностям каплера предоставил знание того, что должна уметь современная программа такого типа.

К немногочисленным недостаткам CESM следует отнести ее размер. Даже отдельный каплер (CESM driver + MCT) занимает около 40000 строк кода, отдельно же требуется подключить библиотеку PIO (Parallel I/O)[6], и, если необходимо, ESMF (Earth System Modeling Framework) [7]. Система написана скорее для предопределенного набора компонент и внедрение новой модели требует нетривиальных изменений и работы с внутренними структурами. Добавление новой сетки хоть и описано в руководстве, все равно требует самостоятельного построения интерполяционных весов для нее [8].

Производительность совместной версии CESM высокого разрешения составляет около 3 лет расчетного времени за сутки реального (скорость 5 моделируемых лет в день традиционно считается минимумом для проведения долгосрочного моделирования климата и достигается пока только на моделях грубого разрешения). Последние тесты показали, что вычислительные затраты каплера CESM составляют довольно значимые 20 процентов [5].

Другой лидер направления — система OASIS (Ocean Atmosphere Sea Ice Soil). Ее самая популярная версия OASIS3 [9] используется до сих пор многими научными группами по всему миру. Тем не менее, она содержит последовательный каплер, что является очевидным узким местом системы — как с точки зрения ограничений по памяти, так и с точки зрения глобальных коммуникаций.

Новая версия 2012 года — OASIS-MCT [11] решает проблему последовательной интерполяции с помощью процедур пакета MCT, выполняющихся параллельно на подмножестве ядер каждой компоненты. Таким образом в OASIS-MCT отсутствует отдельный процесс каплера. На официальном сайте проекта [10] указано, что система по-прежнему содержит систему ввода-вывода через мастер процесс, что очевидно ограничивает ее использование для больших сеток. Даже если будет организован параллельный ввод-вывод, то решение с подмножеством сервисных процессов из числа процессов модели кажется не лучшим, ведь в этом случае все ядра, рассчитывающие модель (которые неявно синхронизированы через внутрикомпонентные обмены) будут ожидать завершения самой медленной операции работы с файловой системой.

Достоинством системы OASIS является ее минимальное вмешательство в код модели. С другой стороны, пользователь должен сам следить за организацией пересылок между компонентами во избежании тупиковых ситуаций, что требует определенных усилий при наличии в системе нескольких компонент [11].

Отталкиваясь от опыта разработки вышеперечисленных моделей, мы решили создать свою компактную систему, которая с одной стороны будет удовлетворять общепринятым предложениям [12] для более простой интеграции с ними в будущем, а с другой — реализующей более эффективно некоторые

критические алгоритмы, автоматически контролирующей периоды событий системы и практически не затрагивающей внутренний код модели пользователя.

**Общая конфигурация**

Разрабатываемый программный комплекс состоит из трех основных частей, представляющих этапы жизни любой физической модели:

1. довычислительного блока построения интерполяционных весов и подготовки начальных данных
2. основного вычислительного блока - каплера и интерфейсов к нему
3. блока визуализации

На довычислительном этапе реализованы инструменты построения интерполяционных весов в SCRIP-формате (Spherical Coordinate Remapping and Interpolation Package [13]) и добавления пользовательских данных в netCDF-файл начальных условий с использованием трехмерной интерполяции на основе пакета CDO (Climate Data Operators, [14]).

Каплер написан на Fortran 2003, состоит всего из 4000 строк кода и поддерживает произвольное число моделей на логически-прямоугольных сетках. В его функционал входит синхронизация моделей, параллельная интерполяция, сохранения диагностических полей и интегральных характеристик, контрольных точек, подкачки данных эксперимента в процессе счета. Работа с файловой системой реализована параллельно на основе пакета NetCDF4/HDF5 [15].

Каждое ядро каплера взаимодействует только с определенным подмножеством ядер компоненты, из чего следует как локальность данных, так и локальность коммуникаций. Схема работы совместной модели для 4 ядер каплера и трех параллельных компонент приведен на Рисунке 1.

В конце счета есть возможность создания высококачественной графики и анимации прямо на кластере средствами скриптов Python, использующих модули PynNGL/PyNIO.

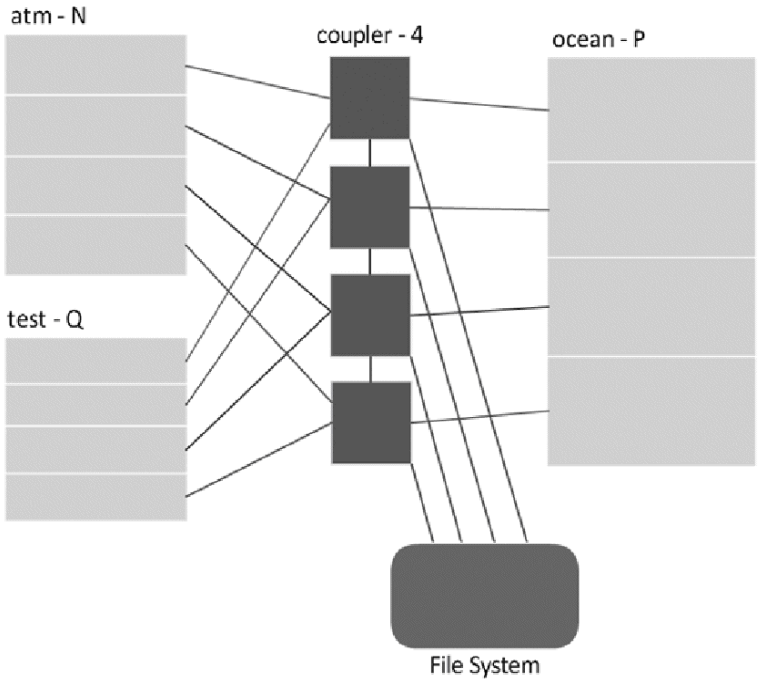


Рис.1 Схема работы совместной системы для 4-ядерного каплера и трех компонент

Совместная система запускается в виде единого исполняемого файла, главная программа которого вызывает интерфейсы моделей, например, *init_grid, init_data, make_step, finalize.* Для работы в совместной системе пользователю достаточно определить свой производный класс, наследующий определенный в системе базовый класс компоненты *comp*. Помимо готовых к использованию методов (например, *send_data*), наследуются и абстрактные интерфейсы, которые не существуют для общего случая и должны быть определены в производном классе в каждом конкретном случае компоненты. Именно эти интерфейсы будет вызывать главная программа, и от их определения зависит то, что именно будет делать компонента на инициализации, какие массивы она зарегистрирует для различных событий, что будет выполнять в течение вычислительного шага и т. д. Такой вид системы позволяет практически не затрагивать код подключаемой модели и обмениваться различными версиями в виде библиотек, описывая компоненту в виде черного ящика с несколькими предопределенными методами.

Кроме того, головная программа хранит лишь указатель на базовый класс *comp*, который, используя полиморфную ссылку, вызывает в коде одни и те же процедуры, фактически обращаясь к методам разных производных классов. Такой подход позволяет ограничить пользователя от любых изменений в коде, за пределами его производного класса. Таким образом, при добавлении новых компонент, ни каплер, ни главная программа не потребуют изменений.

На этапе инициализации известна информация обо всех периодах действий в системе. Это позволяет каплеру построить временную цепочку событий. При наступлении определенного события (например, отправить данные), нет необходимости в синхронизации между различными ядрами каплера и компонентами (например, в ситуации, когда две модели одновременно готовы записывать диагностическую информацию). Цепочка событий позволяет компонентам асинхронно сбрасывать данные, продолжая счет. При досрочном завершении каплером обработки предыдущего события, он заранее инициализирует прием данных для обработки следующего.

Все посылки данных со стороны компоненты пакуются в общий буфер и используют отложенные операции (комбинация MPI_SEND_INIT на инициализации и MPI_STARTALL в процессе счета) для более эффективной отправки. Чтение-запись в массивы производится по переданным на этапе регистрации адресам и уже не требуют вызовов со стороны пользователя — данные действия будут производиться автоматически на основе определенных при старте периодов.

**Интерполяция**

Алгоритм интерполяции использует файлы весов, построенных на довычислительном этапе с помощью пакета SCRIP.

Сначала компонента асинхронно отправляют данные, причем подмножество ядер каждой компоненты работает только со своим ядром каплера. Непосредственное умножение на веса происходит уже в коммуникаторе каплера и реализовано параллельно для двух случаев — обмена на стороне источника и на

стороне получателя, как обсуждается в работе [2]. В первом случае схема интерполяции в общих словах выглядит как «получи все необходимые данные от других ядер каплера и умножь на веса», во втором - «посчитай на локальной области частичные суммы, получи оставшиеся от соседей, сложи их». В обоих случаях число операций одинаково, разница заключается лишь в количестве обмениваемых ячеек. В случае интерполяции, например, с модели океана с разрешением 3600x1800 на более грубую сетку атмосферы 640x400 имеет смысл использовать вторую схему, в обратном направлении — первую.

Все необходимые операции посылок и приема инициализируются в начале счета (MPI_SEND_INIT, MPI_RECV_INIT) и используются на этапе счета как отложенные. Кроме того, сначала вычисляются и отправляются ячейки, необходимые соседям, потом происходит счет в основной локальной области данного ядра для ячеек, не требующих обменов, и только в конце принимаются недостающие данные, и процесс интерполяции завершается. Таким образом, происходит перекрытие вычислений и коммуникаций, что, в сочетании с отложенными MPI-операциями определяет высокую эффективность алгоритма.

Результаты пинг-понг теста на суперкомпьютере “Ломоносов” (характеристики в Приложении) приведены для двух экспериментов совместной модели мирового океана и атмосферы. Условия теста, как в [4] и [9], выглядят как обмен полями между двумя компонентами с отключенной физикой. В нашем тесте модель океана каждые 2 часа отправляла модели атмосферы 3 поля и каждый час принимала от нее 8 полей. Сетка океана — триполярная (3600x1800), атмосферы — широтно-долготная (720x360). Процесс интерполяции состоит из трех частей: сбора данных от компоненты источника, непосредственное умножение на матрицу весов с обменами внутри коммуникатора каплера и распределение данных компоненте получателю. Длина теста составляла 10 модельных дней, что соответствует 240x8 интерполяциям атмосфера-океан, и 120x3 — океан-атмосфера. Данные получены с использованием стандартного Fortran Intel Compiler без дополнительного тюнинга.

Первый тест был проведен для различных комбинаций ядер океана и атмосферы и 4 размеров коммуникатора каплера: 1, 5, 10, 20 (Рис. 2). Горизонтальная ось соответствует сумме ядер океана и атмосферы: 400+100, 800+200, 1600+400, 3200+800, вертикальная — времени выполнения в секундах. Результаты показывают, что увеличение коммуникационных затрат на фазы сбора и распределения данных является относительно небольшим (для параллельного случая) и даже 20 ядер каплера для 3200 и 800 ядер океана и атмосферы проходят тест за разумные 15,5 секунд. Время выполнения последовательного алгоритма растет быстрее и составляет значительные 205 секунд для того же размера океан-атмосфера.

Условия для второго теста остались прежними, за исключением того, что теперь, размеры коммуникаторов океана и атмосферы фиксируются 1600 и 400 ядрами соответственно (Рис 3.) Размер каплера варьируется от 1 до 40 ядер. Результаты показывают хорошую масштабируемость процедуры параллельной интерполяции. Больший размер каплера не имеет смысла при данных размерах стеки, потому что даже 6,4 секунды за 10 дней теста является очень хорошим результатом, и это время, скорее всего, будет перекрыто дисбалансом производительности реальных моделей.

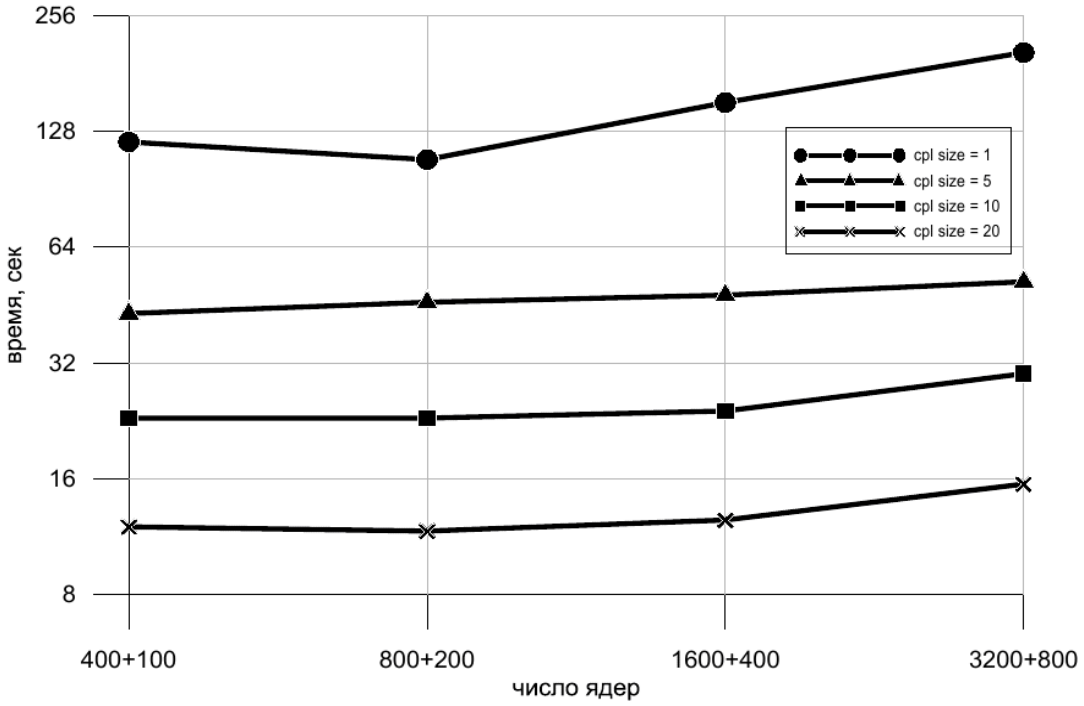


Рис. 2. Время в секундах (ось y) работы процедуры интерполяции между моделями океана (3 поля 3600×1800

каждые 2 часа) и атмосферы (8 полей 720×360 каждый 1 час) в течение 10 дней в зависимости от количества ядер, используемых в моделях (ось x). Результаты приведены для различных размеров каплера (1, 5, 10, 20 ядер).

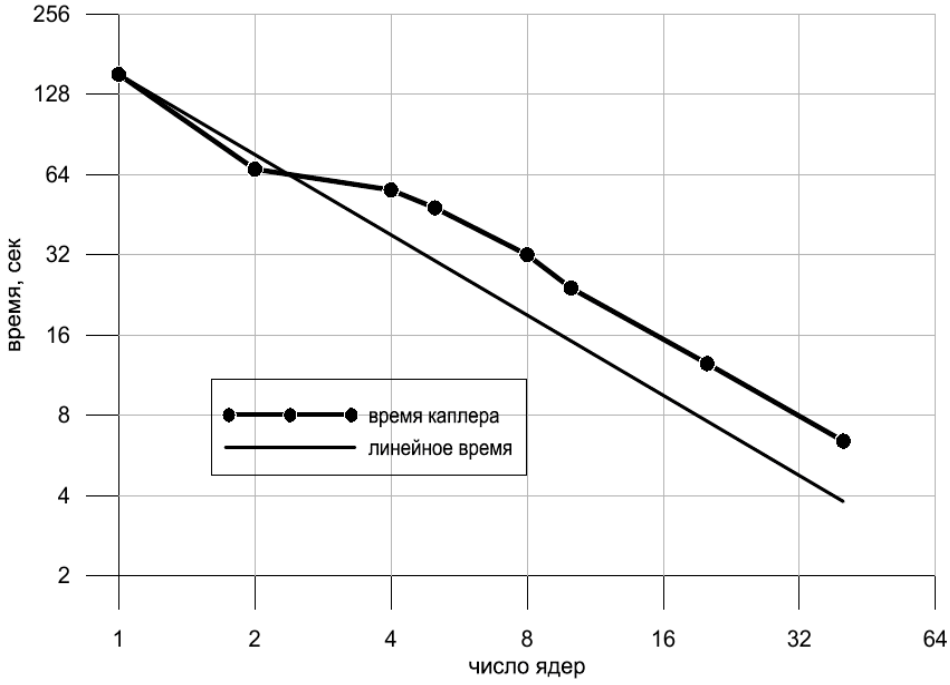


Рис. 3. Время в секундах (ось y) работы процедуры интерполяции между моделями океана (3 поля 3600×1800 каждые 2 часа) и атмосферы (8 полей 720×360 каждый 1 час) в течение 10 дней для различных размеров каплера (ось x) и фиксированного размера компонент (1600 и 400 ядер соответственно)

**Ввод-вывод**

Как известно, существуют три стратегии работы с файловой системой (Рис. 4): через мастер-процесс (сбор данных на одном ядре с последующей записью в файл), прямая (каждый процесс пишет в отдельный или общий файл) и через делегатов (компромиссный вариант – только подмножество процессоров параллельно работает с файлом).

Первый вариант самый простой для реализации, но содержит очевидное узкое место, связанное с глобальными коммуникациями, полным отсутствием масштабируемости и возможной нехваткой памяти узла при передаче больших массивов данных. Например, только один трехмерный массив двойной точности для океана с сеткой 3600x1800x49 занимает около 2,6 гигабайт, что является критическим значением для размера оперативной памяти ядра большинства современных архитектур.

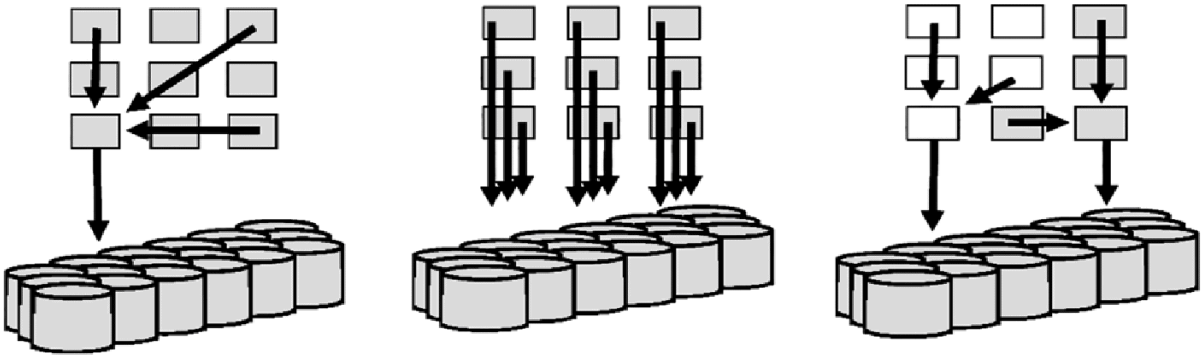

Рис. 4. Схемы реализации ввода-вывода: через мастер-процесс, прямая, делегатная

Вторая стратегия по-прежнему относительно легко реализуется, но содержит два существенных недостатка. Во-первых, как и в случае записи в один, так и в случае записи в разные файлы, возникает перегрузка канала вычислительный узел-память. Например, одновременно обратившиеся в памяти 7000 ядер океана будут ожидать в очереди, так как в параллельном режиме запись возможна лишь для небольшого числа процессов - от одного до нескольких десятков, в зависимости от установленного оборудования. Во-вторых, при записи в различные файлы, возникает необходимость постпроцессинга в конце счета - объединение отдельных

данных в единый файл. Для больших экспериментов это может занять длительное время, сравнимое с непосредственными вычислениями.

Наконец, вариант с делегатами позволяет выделить под работу с дисками небольшое, поддерживаемое для параллельного доступа число процессоров и, кроме того, уменьшить нагрузку при сборе-распределении данных. Такая схема используется в нашем каплере. В результате, как сохранение контрольных точек и диагностики, так и подкачка файловых данных в процессе счета происходит параллельно через процессы каплера, представляющие делегатов. Преимущества метода заключаются в локальности данных и коммуникаций, достигающихся за счет того, что процесс каплера взаимодействует только с непосредственными подчиненными из компоненты, и именно этими локальными частями он обменивается с файловой системой.

Кроме того, поскольку все периоды событий определены заранее, на этапе инициализации система строит всевозможные отложенные операции для межпроцессорных обменов в рамках работы системы ввода-вывода. Таким образом, выделение памяти под буферы приема-посылки, создание отложенных MPI-вызовов и подготовка netCDF файлов выполняется один раз на инициализации.

Система поддерживает следующие события ввода-вывода: прочитать/сохранить контрольную точку, сохранить диагностику, сохранить интегральную диагностику, прочитать данные из файла в процессе счета. В отсутствие других компонент, каплер может использоваться как эффективная внешняя система ввода-вывода.

Несмотря на стабильную работу блока для рабочих разрешений модели океана (3600x1800x50), было интересно протестировать его на перспективных размерах сеток. Работа с огромными массивами данных позволяет определить степень локализации памяти каплера. В Таблице 1 приведены некоторые характеристики сохраненной контрольной точки (4 трехмерных массива и 5 двумерных) для разрешений 3600x1800x50 (базовое), 5400x2700x50, 7200x3600x50. Дополнительно была включена интерполяция с тестовой моделью атмосферы (горизонтальное разрешение 720x360) для моделирования максимально возможной загрузки памяти каплера.

Таблица 1. Характеристики контрольной точки океана для больших сеток.

| 3D-массив | Размер файла, гб | Число ядер каплера | Число ядер океана |
|---|---|---|---|
| $3600\times1800\times50$ | 5,2 | 10 | 800 |
| $5400\times2700\times50$ | 11,7 | 50 | 1800 |
| $7200\times3600\times50$ | 20,8 | 80 | 4000 |

Очевидно, что скорость работы с файловой системой - это очень нестабильная величина, зависящая от установленного оборудования и даже загруженности компьютера в данный момент. Поэтому, мы были больше заинтересованы в возможности такой записи, а не ее масштабируемости. Стоит отметить, что в реальном приложении редкое сохранение контрольных точек может быть полностью совмещено с вычислениями за счет асинхронной отправки данных каждым ядром компоненты.

**Заключение**

В работе представлен оригинальный пакет программ для совместной модели Земной системы ИВМ РАН. Ключевая его часть, каплер, имеет небольшой для программ такого типа размер и решает основные задачи параллельного объединения моделей: синхронизации, интерполяции и ввода-вывода. Архитектура системы полностью независима от кода отдельных компонент и представляет собой единый исполняемый файл, головная программа которого вызывает предопределенные интерфейсы физических моделей и каплера.

Довычислительный блок помогает пользователю построить интерполяционные веса и подготовить начальные данные, используя пакеты SCRIP и CDO. Скрипты Python (PyNGL/PyNIO) позволяют создавать высококачественные рисунки непосредственно на суперкомпьютере.

Было проведено два теста на эффективность работы системы интерполяции. Первый тест показал, что время выполнения параллельного алгоритма лишь незначительно растет при увеличении коммуникационных нагрузок. Из второго теста следует линейная масштабируемость алгоритма для различного числа ядер каплера и фиксированных размеров коммуникаторов компонент. Даже 15.5 секунд для тестовой 10-дневной интерполяции на 20 ядрах каплера является быстрым результатом, хотя это значение и не является пределом.

Тесты системы ввода-вывода подтвердили стабильную работу каплера для огромных массивов данных. Четыре трехмерных и пять двумерных массивов были успешно записаны в качестве контрольной точки для различных размеров сеток высокого разрешения.

В итоге, мы имеем компактный и быстрый каплер для совместной модели Земной системы ИВМ. В качестве первого этапа, он был использован в качестве эффективной системы ввода-вывода вихреразрешающей глобальной модели Мирового Океана [17], [18]. Кроме того, были проведены первые 50 летние эксперименты с форсингом CORE2 [16], за счет использования возможности каплера подкачки данных во время счета для создания файловой компоненты атмосферы.

Дальнейшая работа будет вестись в трех направлениях. Во-первых, будут проведены расширенные тесты системы интерполяции для перспективных размеров сеток. Кроме того, интересно будет установить оптимальные размеры каплера для различных событий работы с файловой системой. Во-вторых, необходимо добавить алгоритмы сжатия для уменьшения размера хранимых файлов. Наконец, планируется закончить объединение вихреразрешающей модели Мирового океана и полулагранжевой модели атмосферы ПЛАВ [19]. В рамках совместной системы океан-атмосфера-каплер неизбежно возникнут интересные задачи оптимизации производительности .

**Благодарности**

Эта работа была частично поддержана грантами РФФИ (12-05-01155-а, 13-05-01141-а, 12-05-31317), проектом фундаментальных исследований Президиума РАН, грантом Министерства образования и науки по договорам № 8344 (17/08/2012) и 8328, стипендией Президента РФ для аспирантов № 136 (28/02/2013).

**Приложение**

Суперкомпьютер «Ломоносов» [20] Московского государственного университета им. Ломоносова имеет несколько различных разделов в зависимости от размера памяти на узел, ее вида и типа процессора. Мы использовали раздел regular4 с 4160 узлами, состоящих из 2 x Intel Xeon 5570 Nehalem процессоров (2×4 ядра) на 12 Гб памяти (всего 33 280 ядер). Вычислительные узлы суперкомпьютера объединены коммуникационной сетью QDR InfiniBand с пропускной способностью до 40 Гбит/с. Многоуровневая система хранения данных состоит из трех частей: быстрого хранилища (500 Тбайт) на основе параллельной файловой системы lustre для проведения расчетов, основного хранилища (312 Тбайт), предназначенного для хранения рабочих данных задач пользователей и хранилища архивных данных (580 Тбайт).

ЛИТЕРАТУРА:

1. World Modelling Summit for Climate Prediction (2009) Report of the Workshop held in Reading, UK, 6-9 May 2008. WCRP-131, WMO/TD-No. 1468
2. Craig, A.P., R. Jacob, B.G. Kauffman (more) , 2005: CPL6: The new extensible, high performance parallel coupler for the Community Climate System Model. The International Journal of High Performance Computing Applications, 19, 309-327, DOI: 10.1177/1094342005056117.10
3. Larson J.W., Jacob R. and Ong E. (2005) The Model Coupling Toolkit: a new Fortran90 toolkit for building multiphysics parallel coupled models. International Journal of High Performance Computing Applications 19(3): 277292. DOI10.1177/1094342005056115.
4. Anthony P. Craig, Mariana Vertenstein, Robert L. Jacob: A new flexible coupler for earth system modeling developed for CCSM4 and CESM1. IJHPCA 26(1): 31-42 (2012)
5. Dennis, J.M., M. Vertenstein, P.H. Worley (more) , 2012: Computational perfor- mance of ultra-high-resolution capability in the Community Earth System Model. The International Journal of High Performance Computing Applications, 26, 5-16, DOI: 10.1177/1094342012436965.
6. Dennis, J. M., J. Edwards, R. Loy, R. Jacob, A. A. Mirin, A. P. Craig, and M. Vertenstein, 2012: An application-level parallel I/O library for Earth system models. International Journal for High Performance Computer Applications, 26, 43-53.
7. Collins, N., G. Theurich, C. DeLuca, M. Suarez, A. Trayanov, V. Balaji, P. Li, W. Yang, C. Hill, and A. da Silva (2005). Design and Implementation of Components in the Earth System Modeling Framework. International Journal of High Performance Computing Applications, Volume 19, Number 3, pp. 341-350.
8. CESM1.1 User guide, http://www.cesm.ucar.edu/models/cesm1.1/cesm/doc/usersguide/book1.html
9. Valcke, S.: The OASIS3 coupler: a European climate modelling community software, Geosci. Model Dev. Discuss., 5, 2139-2178, doi:10.5194/gmdd-5-2139-2012, 2012.
10. OASIS project official website, https://verc.enes.org/oasis
11. OASIS-MCT User guide, https://verc.enes.org/oasis/oasis-dedicated-user-support1/documentation/oasis3-mct-user-guide
12. Valcke, S., Balaji, V., Craig, A., DeLuca, C., Dunlap, R., Ford, R. W., Jacob, R., Larson, J., O'Kuinghttons, R., Riley, G. D., and Vertenstein, M.: Coupling technologies for Earth System Modelling, Geosci. Model Dev., 5, 1589-1596, doi:10.5194/gmd-5-1589-2012, 2012.
13. Jones, P. W., 1998: A Users Guide for SCRIP: A Spherical Coordinate Remapping and Interpolation Package. , Los Alamos National Laboratory, Los Alamos, NM.
14. https://code.zmaw.de/projects/cdo/
15. http://www.unidata.ucar.edu/software/netcdf/
16. http://data1.gfdl.noaa.gov/nomads/forms/mom4/COREv2/CIAF_v2.html
17. Ибраев Р.А., Калмыков В.В., Ушаков К.В., Хабеев Р.Н., 2011. Вихреразрешающая 1/10о модель Мирового океана. Экологическая безопасность прибрежной и шельфовой зон и комплексное использование ресурсов шельфа: Сб. научн. тр. Вып. 25, том 2 / НАН Украины, МГИ, ИГН, ОФ ИнБЮМ. Редкол.: Иванов В.А. (гл. ред.) и др. – Севастополь, 2011. – с. 476. Ил. 244. Табл. 23., 30-44.

18. Ibrayev R.A., 2001: Model of enclosed and semi-enclosed sea hydrodynamics. Russ. J. Numer. Anal. Math. Modelling, 16(4), 291-304.
19. Tolstykh M.A., 2003. Variable resolution global semi-Lagrangian atmospheric model, Russian J. Num. An. & Math. Mod., 18(4), 347–361.
20. http://parallel.ru/cluster/lomonosov.html