

УДК 004.942

ПРОГРАММНЫЙ КОМПЛЕКС СОВМЕСТНОГО МОДЕЛИРОВАНИЯ СИСТЕМЫ ОКЕАН–ЛЕД–АТМОСФЕРА–ПОЧВА НА МАССИВНО-ПАРАЛЛЕЛЬНЫХ КОМПЬЮТЕРАХ

В. В. Калмыков¹, Р. А. Ибраев²

Представлен Программный Комплекс Совместного Моделирования системы Океан–Лед–Атмосфера–Почва (ПКСМ-ОЛАП) на массивно-параллельных компьютерах. Приведен краткий обзор исторического развития технологий совместного моделирования и современного состояния данного направления. Описывается архитектура ПКСМ-ОЛАП. Ключевая часть ПКСМ-ОЛАП носит название каплер (от англ. coupler) и предназначена для объединения компонентов в рамках единой MPI-программы. Основные задачи каплера заключаются в синхронизации работы компонентов совместной модели, обмене данными между моделями, интерполяции и работе с файловой системой. Проведено тестирование вычислительных характеристик совместной модели Мировой океан–глобальная атмосфера под управлением ПКСМ-ОЛАП. Результаты тестов для ключевых функций каплера (обменов данными с параллельной интерполяцией и работы с файловой системой) позволяют говорить о высокой производительности каплера. Работа частично поддержана РФФИ (гранты 12-05-01155-а, 13-05-01141-а, 12-05-31317), проектом фундаментальных исследований Президиума РАН, грантом Министерства образования и науки по договору № 8344 (17/08/2012) и № 8328, стипендией Президента РФ для аспирантов № 136 (28/02/2013). Статья рекомендована к публикации Программным комитетом Международной научной конференции “Научный сервис в сети Интернет: все грани параллелизма” (<http://agora.guru.ru/abrau2013>).

Ключевые слова: модели земной системы, массивно-параллельные приложения, параллельный ввод-вывод, большие задачи, каплер, MPI.

1. Введение. Как отмечалось на Всемирном саммите по моделированию и прогнозированию климата [23], существует общее согласие о том, что гораздо более высокое, по сравнению с существующим, разрешение моделей основных компонентов (атмосфера, океан, лед, суша) является главной предпосылкой для реалистичного представления климатической системы и актуальности прогнозов. На современном уровне знаний представляется, что перспективная модель земной системы должна иметь горизонтальное разрешение около 1 км. Сложность физических и химико-биологических процессов в земной системе и в ее основных компонентах и большой объем данных о состоянии системы ставит задачу прогноза климата Земли в ряд самых вычислительноемких в науке.

Наряду с созданием моделей отдельных компонентов земной системы становится чрезвычайно важной роль инструмента, организующего их совместную работу, — *каплера* (от англ. coupler). К его основным задачам относится синхронизация работы моделей, интерполяция данных между различными сетками компонентов и работа с файловой системой. Системы для совместного моделирования разрабатывались как часть моделей атмосфера–океан–лед с начала 1990-х гг. (см. обзор [27]). Наиболее развитые в настоящее время каплеры — это ESMF (Earth System Modeling Framework) [3], MCT (Model Coupling Toolkit) [12], CPL NCAR CESM (Coupler for The National Center for Atmospheric Research Community Earth System Model) [5, 6] и OASIS (Coupler for Ocean Atmosphere Sea Ice Soil) [16, 18].

Архитектура каплера в значительной степени определяется выбором стратегии работы совместной системы.

В ранних версиях совместных моделей физические компоненты запускались в виде отдельных программ в MPMD-режиме (Multiple Program Multiple Data), коммуникация между которыми происходила через разделяемый файл, например как в OASIS-1 и в климатических моделях INMCM3.0 [24] и

¹ Московский государственный университет им. М. В. Ломоносова, факультет вычислительной математики и кибернетики, Ленинские горы, д. 1, стр. 52, ГСП-1, 119991, Москва; аспирант, e-mail: vvk88@mail.ru

² Институт вычислительной математики РАН, ул. Губкина, д. 8, 119991, Москва; гл. науч. сотр., e-mail: ibrayev@mail.ru

INMCM4.0 [25] Института вычислительной математики РАН. В этом случае отсутствует отдельный компонент каплера, который неявно представлен как набор модельных процедур для записи данных в файл и их интерполяции на другую сетку.

Следующим этапом развития становится появление отдельного компонента каплера, имеющего вид мастер-процесса, как в системе OASIS-3, или дополнительно использующего SMP-параллелизм (Symmetric Multiprocessing), как, например, в каплере CCSM cpl3 [5]. Структура совместной системы по-прежнему выглядит как MPMD-запуск отдельных моделей, причем теперь с ними параллельно работает сервисный компонент-каплер, функции коммуникации с которым вызывают модели. Поскольку в системе нет единого управляющего файла, вся синхронизация должна быть выполнена пользователем вручную.

Для больших потоков информации между моделями оба способа объединения не обеспечивают необходимую скорость расчета системы и неминуемо становятся “узким местом” как с точки зрения ограниченной памяти одного ядра, так и с точки зрения глобальных коммуникаций. Следующее поколение каплеров имеет уже полноценный MPP-параллелизм (Massively Parallel Processing), хотя по-прежнему запускается в режиме нескольких исполняемых файлов [5, 19]. Полностью параллельный вид каплера позволяет избежать вычислительных проблем при увеличении размера сеток и частоты обменов потоками.

Неудобство ручной синхронизации моделей в MPMD-режиме приводит к появлению в 2011 г. модели CESM 1.0 [7], которая имеет вид единого исполняемого файла и содержит высокоуровневый драйвер, вызывающий некоторые общие интерфейсы компонентов (`init`, `run`, `finalize` и др.) и каплера с заданными при старте периодами. Архитектура управляющей программы может предполагать различный порядок вычислений моделей системы: это или последовательные вычисления различных моделей на одних и тех же процессорах, или параллельная работа моделей на непересекающихся множествах процессоров. Данный подход уже требует некоторой реорганизации модели и представление ее в интерфейсном виде, понятном драйверу.

Наконец, оригинальное решение было предложено в 2012 г. в версии OASIS-MCT [16], запускающейся в MPMD-режиме без отдельного компонента каплера, чьи функции выполняются параллельно на множестве процессов каждой модели. Данное решение является развитием первоначальной архитектуры с отсутствием центрального каплера за счет использования сервисных процедур на стороне каждого компонента.

Другой особенностью совместной модели является режим работы с файловой системой. В ранних версиях она осуществлялась каждой моделью самостоятельно. При появлении отдельного компонента каплера стало возможным делегировать эту общую для всех функцию ему. Очевидно, что схема с мастер-процессом (OASIS-3, OASIS-MCT) была ограничена размером памяти одного ядра и скоростью записи, поэтому начались активные разработки параллельных механизмов работы с файловой системой. Кроме того, появление огромного количества данных экспериментов делало необходимым создание единого геофизического формата данных, которым в результате стал netCDF [14]. Именно его неофициальная параллельная версия pnetcdf [20] используется, например, в каплере системы CCSM. Последняя версия netCDF4 использует параллельные интерфейсы другого популярного формата HDF5 [9], и подключение отдельной библиотеки pnetcdf уже не требуется.

Итак, критическими характеристиками каплера являются его

- 1) архитектура (последовательная, параллельная, с высокоуровневым драйвером или в виде набора процедур);
- 2) схема работы с файловой системой (параллельная или последовательная);
- 3) поддержка полного или частичного цикла жизни модели (формирования начальных данных, непосредственного счета и обработки, анализа и визуализации) и привязка структуры к определенным моделям земной системы;
- 4) удобство использования; при создании совместной системы возникает задача объединения кода отдельно разрабатываемых моделей; от уровня абстракции архитектуры системы зависит простота подключения нового компонента, прозрачность работы в ансамбле и степень изменений кода при модификации или замене некоторой модели.

Как уже отмечалось, наиболее развитые сегодня проекты — это CESM (с каплером на основе MCT и ESMF) и OASIS.

К немногочисленным недостаткам CESM следует отнести ее размер. Даже отдельный каплер (CESM driver + MCT) занимает около 40 000 строк кода, отдельно же требуется подключить библиотеку PIO (Parallel I/O) [8] и, если необходимо, ESMF (Earth System Modeling Framework, 500 000 строк кода) [3]. Система написана скорее для предопределенного набора компонентов, 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 [2].

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% [7].

Самая популярная версия OASIS3 [18] 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 г. — OASIS-MCT [16] решает проблему последовательной интерполяции с помощью процедур пакета MCT, выполняющихся параллельно на подмножестве ядер каждого компонента. Таким образом, в OASIS-MCT отсутствует отдельный процесс каплера. На официальном сайте проекта [15] 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 избежание тупиковых ситуаций, что требует определенных усилий при наличии в системе нескольких компонентов [16].

Отталкиваясь от опыта разработки вышеперечисленных моделей, мы решили создать свою компактную систему, которая, с одной стороны, будет удовлетворять общепринятым предложениям конференции Earth System Modelling [22] 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.

Представляемый в настоящей статье Программный Комплекс Совместного Моделирования системы Океан–Лед–Атмосфера–Почва на массивно-параллельных компьютерах (ПКСМ-ОЛАП) опирается на некоторые идеи и методы, предложенные в работах [5, 12], в частности по использованию пакета SCRIP для построения интерполяционных весов и самого алгоритма параллельного машинга. В нашей версии алгоритм был модифицирован за счет отложенных MPI-операций и перекрытия вычислений и обменов.

Комплекс ПКСМ-ОЛАП развивает идею высокоуровневого драйвера: за счет использования абстрактных интерфейсов главная программа и каплер становятся полностью независимыми от количества подключаемых моделей — для работы в совместной системе пользователю достаточно создать производный класс своего компонента. Такой подход делает систему независимой от конкретных моделей в отличие, например, от пакета CESM.

Для работы с файловой системой реализована полностью параллельная делегатная схема на основе библиотеки netCDF4/HDF5.

Кроме того, комплекс ПКСМ-ОЛАП поддерживает полный цикл жизни модели: подготовку начальных данных и довычислительное построение интерполяционных весов, собственно организацию вычислений совместной модели (каплинг) и визуализацию результатов с помощью Python-скриптов.

Следует отметить, что мы изначально разрабатывали ПКСМ-ОЛАП для совместных моделей высокого пространственного разрешения, при этом над нами не довлел набор старых алгоритмов и библиотек. Это позволило создать современный, компактный и в то же время гибкий программный комплекс.

2. Общая конфигурация программного комплекса. Разрабатываемый программный комплекс состоит из трех основных частей, представляющих этапы жизни любой физической модели:

- довычислительного блока построения интерполяционных весов и подготовки начальных данных,
- основного вычислительного блока — каплера и интерфейсов к нему,
- блока визуализации.

На довычислительном этапе реализованы инструменты построения интерполяционных весов в SCRIP-формате (Spherical Coordinate Remapping and Interpolation Package, [11]) и добавления пользовательских данных в файл начальных условий с использованием трехмерной интерполяции на основе пакета CDO (Climate Data Operators, [1]).

Ядро системы, каплер, написан на языке Fortran-2003, состоит всего из 4000 строк кода и поддерживает произвольное число моделей на логически-прямоугольных сетках. В его функционал входит син-

хронизация моделей, параллельная интерполяция, сохранение диагностических полей и интегральных характеристик, контрольных точек, подкачки данных эксперимента в процессе счета. Работа с файловой системой реализована параллельно на основе библиотеки NetCDF4/HDF5 [14].

Архитектура взаимодействий каплера и компонентов совместной модели построена так, чтобы максимально избежать глобальных связей. Каждое ядро каплера взаимодействует только с определенным подмножеством ядер компонента, из чего следует как локальность данных, так и локальность коммуникаций. Схема работы совместной модели для четырех ядер каплера и трех параллельных компонентов приведена на рис. 1.

В конце счета есть возможность создания высококачественной графики и анимации прямо на кластере средствами скриптов Python, использующих модули PyNGL/PyNIO [21].

3. Архитектура совместной системы. Совместная система запускается в виде единого исполняемого файла, главная программа которого вызывает интерфейсы моделей, например, инициализации расчетной сетки (`init_grid`), инициализации данных модели (`init_data`), интегрирования уравнений (`make_step`), завершения счета (`finalize`). Для работы в совместной системе пользователю достаточно определить свой производный класс, наследующий определенный в системе базовый класс компонента `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 компонента. Именно эти интерфейсы будет вызывать главная программа, и от их определения зависит то, что именно будет делать компонент на инициализации, какие массивы он регистрирует 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 компонент 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 не позволяет пользователю вносить любые изменения в код за пределами его производного класса. Таким образом, при добавлении новых компонентов ни каплер, ни головная программа не потребуют изменений.

На этапе инициализации пользователю необходимо задать информацию обо всех периодах действий в системе. Это позволяет каплеру построить временную цепочку событий. При наступлении определенного события (например, принять 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 компонента 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 — эти действия будут производиться автоматически на основе определенных при старте периодов.

4. Обмен данными между модельными компонентами, интерполяция. Одной из важнейших функций каплера, влияющих на общее время выполнения программы, является организация обменов данными между модельными компонентами совместной системы. В каплере ПКСМ-ОЛАП при наступлении события отправки данных из одного модельного компонента (*источник*) в другой модельный компонент (*получатель*) источник асинхронно отправляет данные, причем подмножество ядер каждого компонента работает только со своим ядром каплера.

Непосредственное умножение на веса происходит уже в коммутаторе каплера и реализовано параллельно для двух случаев — на декомпозиции источника и получателя, как это обсуждается в работе [5].

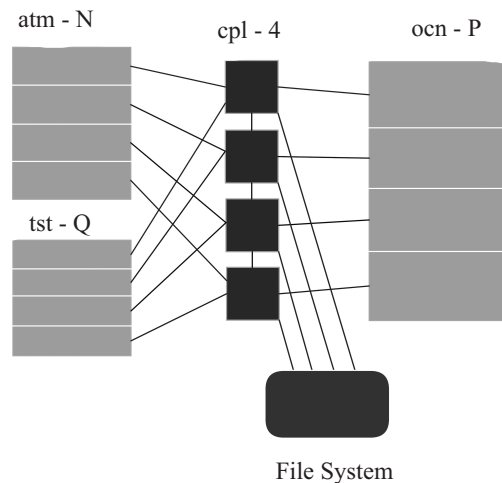


Рис. 1. Схема работы совместной системы для 4-ядерного каплера и трех компонентов

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 3600×1800 на более грубую сетку атмосферы 640×400 имеет смысл использовать вторую схему, в обратном направлении — первую.

Алгоритм интерполяции использует файлы весов, построенных на довычислительном этапе с помощью пакета SCRIP. Таким образом, медленная операция построения весовых файлов (более часа для подробных сеток) производится для определенных размеров сетки одновременно, а во время запуска системы данные просто считываются ядрами каплера из netCDF-файлов.

Все необходимые для выполнения интерполяции операции посылки и приема инициализируются в начале счета (с помощью MPI-процедур `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 обменов, что в сочетании с отложенными MPI-операциями определяет высокую эффективность алгоритма.

Результаты пинг-понг теста на суперкомпьютере “Ломоносов” приведены для двух экспериментов совместной модели Мирового океана и атмосферы. Условия теста, как в [6] и [18], выглядят как обмен полями между двумя компонентами с отключенной физикой, т.е. без решения уравнений океана и атмосферы, но с активными обменами между ними. В нашем тесте модель океана каждые 2 часа отправляла модели атмосферы три поля и каждый час принимала от нее восемь полей. Сетка океана — триполярная (разрешение 3600×1800), атмосферы — широтно-долготная (разрешение 720×360). Процесс интерполяции состоит из трех частей: сбора данных от компонента источника, непосредственное умножение на матрицу весов с обменами внутри коммуникатора каплера и распределение данных компоненту-получателю. Тест проводился на 10 модельных дней, что соответствует 240×8 интерполяциям атмосфера–океан и 120×3 — океан–атмосфера. Данные получены с использованием стандартного Fortran Intel Compiler без дополнительного тюнинга.

Суперкомпьютер “Ломоносов” [13] Московского государственного университета им. М. В. 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 $2 \times$ 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.

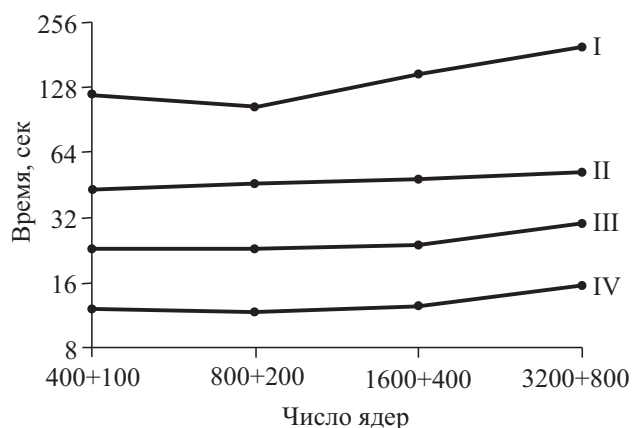


Рис. 2. Время выполнения теста для фиксированного коммуникатора каплера размера 1(I), 5(II), 10(III), 20(IV) ядер и различных размеров коммуникаторов компонентов океана и атмосферы

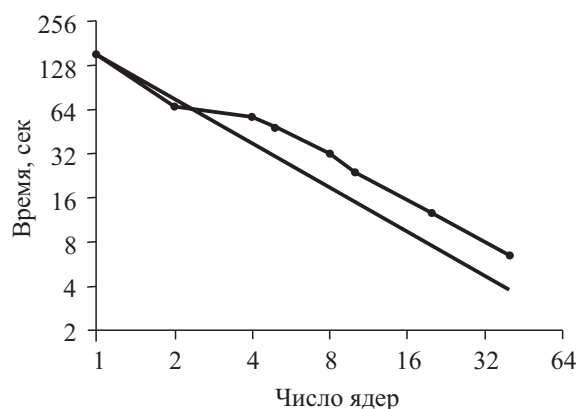


Рис. 3. Время выполнения теста для фиксированного размера коммуникаторов компонентов океана и атмосферы и различных размеров коммуникатора каплера

Первый тест был проведен для различных комбинаций размеров коммуникаторов океана и атмосферы и четырех размеров коммуникатора каплера: 1, 5, 10, 20 ядер (рис. 2, кривые I, II, III, IV соответственно). Горизонтальная ось соответствует количеству ядер для моделей 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): 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.

Во втором тесте размеры коммуникаторов моделей океана и атмосферы фиксируются равными 1600 и 400 ядер соответственно (рис. 3). Размер каплера варьируется от 1 до 40 ядер. Результаты показывают хорошую масштабируемость процедуры параллельной интерполяции. Большой размер каплера не имеет смысла при данных размерах сетки, 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.

5. Работа с файловой системой. 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).

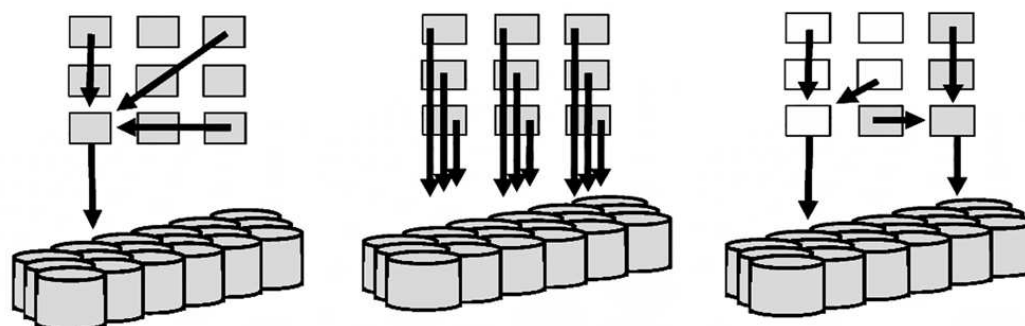


Рис. 4. Схемы реализации ввода-вывода: через мастер-процесс, прямая, делегатная

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 модели Мирового океана с сеткой $3600 \times 1800 \times 49$ занимает около 2.6 гигабайт, что является критическим значением для размера оперативной памяти ядра большинства современных архитектур.

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 к внешней 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 происходят параллельно через процессы каплера, представляющие делегатов. Преимущества метода заключаются в локализации данных и коммуникаций, достигающихся за счет того, что процесс каплера взаимодействует только с непосредственными подчиненными из компонента, 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-отправки, создание отложенных MPI-вызовов и подготовка netCDF файлов выполняется один раз на этапе 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 компонентов каплер может использоваться как эффективная внешняя система ввода-вывода.

Несмотря на стабильную работу блока для рабочих разрешений модели Мирового океана ($3600 \times 1800 \times 50$), было интересно протестировать его на перспективных размерах сеток. Работа с огромными массивами данных позволяет определить степень локализации памяти каплера. Были проведены успешные тесты памяти каплера на сохранение контрольной точки (4 трехмерных массива и 5 двумерных) для разрешений $3600 \times 1800 \times 50$ (базовое), $5400 \times 2700 \times 50$, $7200 \times 3600 \times 50$. Размер файла для последнего разрешения составляет около 20 Гб. Дополнительно была включена интерполяция с тестовой моделью атмосферы (горизонтальное разрешение 720×360) для моделирования максимально возможной загрузки памяти.

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. Следует отметить, что в реальном приложении редкое сохранение контрольных точек может быть полностью совмещено с вычислениями за счет асинхронной отправки данных каждым ядром компонента.

6. Заключение. В работе представлен оригинальный пакет программ для совместной модели Мировой океан–лед–глобальная атмосфера–почва. 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 компонентов 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 на основе пакетов SCRIP и CDO. Скрипты Python (PyNGL/PyNIO) позволяют создавать высококачественные рисунки непосредственно на кластере.

На суперкомпьютере “Ломоносов” 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 коммутаторов компонентов. Даже 15.5 секунд для тестовой 10-дневной интерполяции на 20 ядрах каплера представляется быстрым результатом, хотя это значение и не является пределом.

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

Таким образом, разработана архитектура и основа компактного и быстрого программного комплекса совместного моделирования системы Мировой океан–лед–глобальная атмосфера–почва. В качестве первого этапа комплекс был использован как эффективная система ввода-вывода вихреразрешающей глобальной модели Мирового океана [10, 26]. Кроме того, были проведены первые 50-летние эксперименты с форсингом CORE2 [4] за счет использования возможности каплера подкачки данных во время счета для создания файлового компонента 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 и получить для них графики масштабируемости. Во-вторых, необходимо добавить алгоритмы сжатия для уменьшения размера хранимых файлов. Наконец, планируется закончить объединение вихреразрешающей модели Мирового океана [26] и полулагранжевой модели атмосферы ПЛАВ [17].

СПИСОК ЛИТЕРАТУРЫ

1. Официальный сайт проекта, 2013 (<https://code.zmaw.de/projects/cdo/>).
2. CESM1.1 User Guide, 2012 (<http://www.cesm.ucar.edu/models/cesm1.1/cesm/doc/usersguide/book1.html>).
3. Collins N., Theurich G., DeLuca C., Suarez M., Trayanov A., Balaji V., Li P., Yang W., Hill C., da Silva A. Design and implementation of components in the Earth System Modeling Framework // Int. J. of High Performance Computing Applications. 2005. **19**, N 3. 341–350.
4. Официальный сайт проекта CORE2, 2013 (http://data1.gfdl.noaa.gov/nomads/forms/mom4/COREv2/CIAF_v2.html).
5. Craig A.P., Jacob R., Kauffman B.G. CPL6: The new extensible, high performance parallel coupler for the Community Climate System Model // Int. J. of High Performance Computing Applications. 2005. **19**. 309–327.
6. Craig A.P., Vertenstein M., Jacob R.L. A new flexible coupler for earth system modeling developed for CCSM4 and CESM1 // Int. J. of High Performance Computing Applications. 2012. **26**, N 1. 31–42.

7. *Dennis J.M., Vertenstein M., Worley P.H.* Computational performance of ultra-high-resolution capability in the Community Earth System Mode // *Int. J. of High Performance Computing Applications*. 2012. **26**. 5–16.
8. *Dennis J.M., Edwards J., Loy R., Jacob R., Mirin A.A., Craig A.P., Vertenstein M.* An application-level parallel I/O library for Earth system models // *Int. J. of High Performance Computing Applications*. 2012. **26**. 43–53.
9. Официальный сайт проекта (<http://www.hdfgroup.org/HDF5/>).
10. *Ibrayev R.A.* Model of enclosed and semi-enclosed sea hydrodynamics // *Russ. J. Numer. Anal. Math. Modelling*. 2001. **16**, N 4. 291–304.
11. *Jones P.W.* A User's Guide for SCRIP: A Spherical Coordinate Remapping and Interpolation Package. Los Alamos: Los Alamos National Laboratory, 1998.
12. *Larson J.W., Jacob R., Ong E.* The model coupling toolkit: a new Fortran90 toolkit for building multiphysics parallel coupled models // *Int. J. of High Performance Computing Applications*. 2005. **19**, N 3. 277–292.
13. Официальный сайт, 2013 (<http://parallel.ru/cluster/lomonosov.html>).
14. Официальный сайт проекта (<http://www.unidata.ucar.edu/software/netcdf/>).
15. Официальный сайт проекта, 2013 (<https://verc.enes.org/oasis>).
16. OASIS-MCT User's Guide, 2013 (<https://verc.enes.org/oasis/oasis-dedicated-user-support-1/documentation/oasis3-mct-user-guide>).
17. *Tolstykh M.A.* Variable resolution global semi-Lagrangian atmospheric model // *Russian J. Numer. Anal. Math. Modelling*. 2003. **18**, N 4. 347–361.
18. *Valcke S.* The OASIS3 coupler: a European climate modelling community software // *Geoscience Model Development. Discuss*. 2012. **5**. 2139–2178.
19. *Redler R., Valcke S., Ritzdorf H.* OASIS4 — a coupling software for next generation Earth system modelling // *Geoscience Model Development*. 2010. **3**. 87–104.
20. Официальный сайт проекта, 2012 (<http://trac.mcs.anl.gov/projects/parallel-netcdf>).
21. Официальный сайт проекта, 2013 (<http://www.pyngl.ucar.edu/>).
22. *Valcke S., Balaji V., Craig A., DeLuca C., Dunlap R., Ford R.W., Jacob R., Larson J., O'Kuinghtons R., Riley G.D., Vertenstein M.* Coupling technologies for Earth system modelling // *Geoscience Model Development*. 2012. **5**. 1589–1596.
23. World Modelling Summit for Climate Prediction. Report of the Workshop held in Reading. UK. WCRP-131, WMO/TD-1468. 2008.
24. *Володин Е.М., Дианский Н.А.* Моделирование изменений климата в XX-XXII столетиях с помощью совместной модели общей циркуляции атмосферы и океана // *Изв. РАН. Физика атмосферы и океана*. 2006. **42**, № 3. 291–306.
25. *Володин Е.М., Гусев А.В., Дианский Н.А.* Воспроизведение современного климата с помощью совместной модели общей циркуляции атмосферы и океана INMCM4.0 // *Известия РАН. Физика атмосферы и океана*. 2010. **46**, № 4. 448–466.
26. *Ибраев Р.А., Калмыков В.В., Ушаков К.В., Хабеев Р.Н.* Вихреразрешающая $1/10^\circ$ модель Мирового океана // *Экологическая безопасность прибрежной и шельфовой зон и комплексное использование ресурсов шельфа*. 2011. **25**, № 2. 30–44.
27. *Толстых М.А., Ибраев Р.А., Володин Е.М., Ушаков К.В., Калмыков В.В., Шляева А.В., Мизяк В.Г., Хабеев Р.Н.* Модели глобальной атмосферы и Мирового океана: алгоритмы и суперкомпьютерные технологии. М.: Изд-во Моск. ун-та, 2013.

Поступила в редакцию
23.08.2013
