

МОДУЛЬ ТРАНСПОРТНОГО УРОВНЯ ЛОГОС МИП ДЛЯ РАСЧЕТА СВЯЗАННЫХ ЗАДАЧ

А. В. Дьяков, А. Г. Надуев, А. Д. Черевань, Д. А. Кожаяев

ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской обл.

Введение

Модульная интеграционная платформа (МИП), входящая в состав пакета программ ЛОГОС [1], используется для ряда задач, в числе которых подготовка и проведение связанных расчетов комплексных мультидисциплинарных задач математического моделирования, когда две и более совместно рассчитываемые методики обмениваются данными в обе стороны.

В состав модульной интеграционной платформы входит модуль транспортного уровня Cooperative Computation Framework (CCF), который обеспечивает взаимодействие между математическими методиками ЛОГОС в процессе расчета связанной задачи. CCF характеризуется следующими возможностями:

- наличием программного интерфейса пользователя для обеспечения взаимодействия с математическими методиками ЛОГОС;
- поддержкой протоколов MPI [2] и файлового обмена для передачи информации между математическими методиками ЛОГОС;
- передачей сеточных величин с использованием интерполятора между методиками на каждом шаге обмена;
- обменом командами между математическими методиками ЛОГОС.

Схема использования CCF

На рис. 1 приводится общая схема использования CCF.

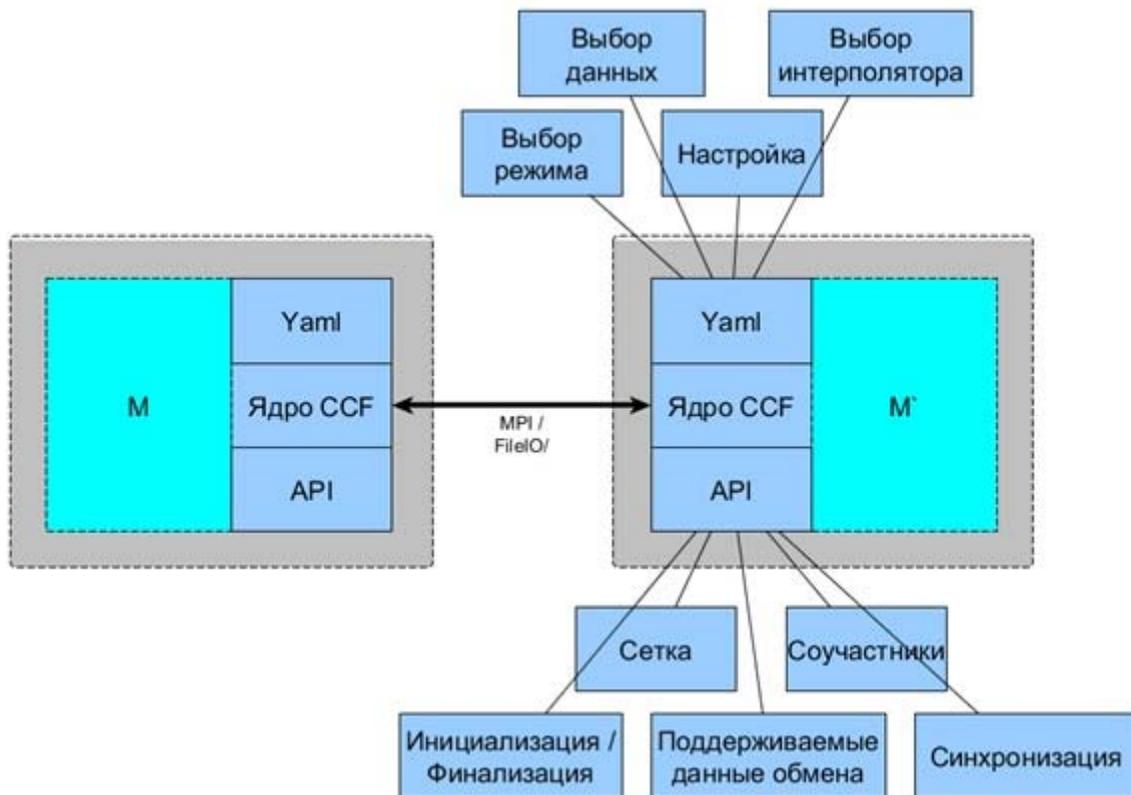


Рис. 1. Схема использования CCF

Данная схема предполагает следующее:

- конфигурирование совместного расчета производится через конфигурационный файл формата YAML [3], который заполняется в графическом конфигураторе (Интеграторе);
- внедрение CCF в методики осуществляется с помощью программного интерфейса пользователя CCF API;
- за обмен данными между методиками отвечает ядро CCF.

Программный интерфейс пользователя

Чтобы обеспечить расчетным методикам ЛОГОС взаимодействие друг с другом с использованием различных транспортных уровней МИП предоставляет программный интерфейс пользователя CCF API. CCF API предполагает следующее:

- интеграцию в связываемые методики для использования ими транспортного слоя;
- обеспечение поддержки внедрения в программные коды методик на языках C, C++, Fortran;
- наличие возможности собранным с поддержкой CCF методикам запускаться в зависимости от конфигурации запуска как в составе связанной или сопряженной задачи, так и независимо в качестве отдельного счетного модуля.

Для использования транспортного слоя CCF в методиках разработчикам необходимо:

- включить в свои файлы исходного кода заголовочный файл, содержащий объявления функций CCF API;
- линковать своё приложение с динамической или статической CCF библиотекой.

Библиотека прикладного программного интерфейса CCF API предоставляет разработчикам методик базовый набор функций, обеспечивающий основную функциональность CCF API, которые делятся на следующие группы:

- функции управления совместным расчетом. К ним относятся: методы, которые инициализируют и завершают совместный расчет (CCF_Init(), CCF_Finalize() и др.), методы синхронизации (CCF_Synchronize(), CCF_ForceSynchronize()), работы с модельным временем (CCF_SetSynchronizationDeltaTime(), CCF_SetStartTime() и др.), а также с участниками обмена;
- функции установки геометрии счетной модели. К ним относятся функции, которые обеспечивают интерфейс передачи информации о сеточной геометрии для запущенной методики (CCF_SetMesh3D(), CCF_SetSurface3D() и др.);
- функции работы с данными.

Среди них:

методы, которые проверяют наличие именованных данных у участников обмена (CCF_CheckDataPresence()),

добавляют публикуемые данные как буфер (CCF_AddData()),

инициируют дополнительный обмен информацией о доступных данных у всех участвующих методик (CCF_PublishData()),

методы, которые предоставляют доступ к полученным данным, как с предварительной обработкой, так и без, а также возвращают размер этих данных;

- функции подписок и запросов.

Такие функции предназначены для работы с подписками и запросами на публикуемые данные;

- функции для управления загрузкой интерполятора.

Кроме базового набора функций, библиотека включает в себя определенные разработчиками типы данных (тип идентификатора совместного расчета, тип запроса данных и т. п.) и перечисления, которые определяют тип сетки, коды команд, принимаемые и обрабатываемые при входе в синхронизацию участников и т. д.

Ядро CCF

Основными классами ядра CCF являются:

- ccf_base_participant – базовый класс участника совместного расчета, от которого наследуются классы ccf_mpi_participant, ccf_file_participant для связывания по интерфейсу MPI и последовательно через файл, соответственно, как показано на рис. 2.

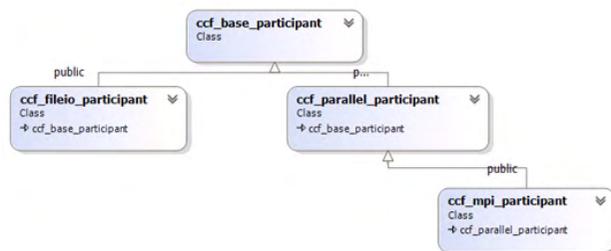


Рис.2. Наследники класса ccf_base_participant

Данный класс обеспечивает инициализацию совместного расчета, владеет объектами межпроцессного взаимодействия, хранит описание участников совместного расчета, отслеживает продвижение участника по модельному времени, обеспечивает публикацию данных, организует доступ к хранилищу полученных данных и предоставляет доступ к выбранному интерполяционному интерфейсу;

- ccf_base_ipc – базовый класс механизма межпроцессного взаимодействия (interprocess-communication – IPC) участников совместного расчета, от которого наследуются классы ccf_mpi_ipc и ccf_file_ipc, как показано на рис. 3.

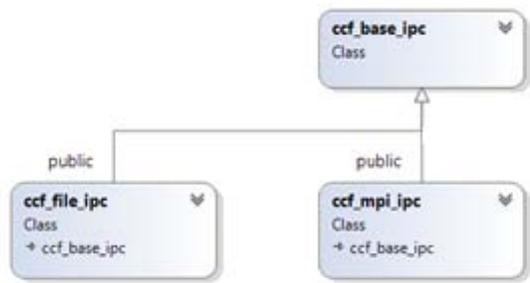


Рис. 3. Наследники класса `ccf_base_ipc`

Данный класс предоставляет универсальный механизм передачи данных от одной методики в другую, добавляет полученные данные в хранилище `ccf_data_pool`, обеспечивает синхронизацию с использованием выбранного типа межпроцессного взаимодействия;

- `ccf_data_pool` – класс-хранилище данных, полученных в момент синхронизации от соседних участников совместного расчета. Класс обеспечивает регистрацию запросов и подписок на данные соседних методик, хранение получаемых данных в процессе межпроцессного взаимодействия;
- `ccf_interpolator_interface` – интерфейсный класс интерполяционных методов (сопоставление соседей, расчет и применение интерполяционных коэффициентов).

Схема управления связанным расчетом

Управление связанным расчетом выполняется по следующей схеме:

- чтение конфигурационного файла формата YAML, если такой файл представлен в системе. Конфигурационный YAML-файл позволяет, задать режим работы CCF, выбрать требуемый интерполятор данных (FSI, Remap), задать физические величины, подлежащие обмену;
- инициализация совместного расчета из конфигурационного файла формата YAML с помощью функции `CCF_InitFromConfig()`, если файл был прочитан на шаге 1, или с использованием параметров по умолчанию;
- инициализация требуемого модуля в соответствии с переданными параметрами и в зависимости от режима совместного расчета: параллельного или файлового обменов с помощью функции `CCF_Init()`:
 - проверка функцией `CCF_Initialized()` инициализации совместного расчета с помощью CCF;
 - обмен данными на каждом шаге;
 - завершение участия в совместном расчете с помощью функции `CCF_Finalize()`.

Режим MPI обменов CCF

Для параллельного высокоскоростного взаимодействия методик, участвующих в совместном рас-

чете, в составе модуля взаимодействия реализован режим обмена данными и командами с использованием интерфейса передачи сообщений MPI.

Инициализация совместного расчета многодисциплинарной задачи с использованием транспортно-го слоя CCF может быть условно разбита на следующие последовательные стадии:

- разделение общего счетного поля между методиками и назначение каждой методике уникального MPI коммуникатора;
- регистрация всех разделяемых ресурсов методик;
- подписка методиками на ресурсы других методик;
- обмен сетками и поиск соседей для обмена данными.

Передача данных от одной методики другой в ходе расчета совместной мультифизической задачи может быть условно разбита на следующие последовательные шаги:

- при получении управления модуль CCF в соответствии со своими внутренними условиями определяет, что на текущем модельном шаге должна быть активирована передача данных от одной методики другой;
- публичные данные исходной методики в момент MPI-синхронизации передаются на соседние MPI-процессы целевой методики;
- на MPI процессе целевой методики поступившие данные хранятся в хранилище (пуле) данных CCF до тех пор, пока целевая методика не запросит их оттуда при помощи заданной функции-обертки.

Пример схемы передачи данных от одной методики другой в ходе совместного расчета с использованием MPI представлен на рис. 4.

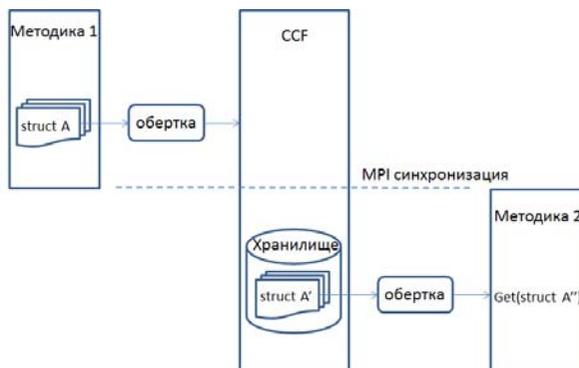


Рис. 4. Пример схемы передачи данных от одной методики другой в ходе совместного расчета с использованием MPI

Режим обмена CCF посредством файловых операций

Методики, во время проведения мультидисциплинарного расчета, могут выполняться на разных узлах локальной сети на различных программных платформах. Для управления совместным расчетом

в такой ситуации целесообразно выбирать связывание с передачей данных через файлы как наиболее гибкий и универсальный способ обеспечения связи между методиками.

Связывание счетных методик через файловый обмен использует их входные и выходные файлы обмена данными. Интегрированный в методику модуль совместного расчета считывает значения входных параметров из файла обмена в момент инициализации методики. По завершении расчета методики модуль совместного расчета записывает значения выходных параметров, и тем самым передает их дальше по цепочке связанным методикам.

Для работы двух счетных модулей в режиме файлового обмена требуется:

- использование CCF API в счетном модуле;
- подготовка двух конфигурационных файлов формата YAML;
- последовательный запуск подзадач, при котором первая подзадача на этапе завершения автоматически сохраняет требуемые данные в файл обмена, а вторая подзадача на этапе инициализации автоматически считывает и интерполирует на собственную сетку данные из файла обмена.

В качестве формата файла обмена используется собственный формат ExLGS – адаптированная реализация формата LGS, используемого в методиках ЛОГОС Аэрогидродинамика для сохранения результатов расчета. Формат ExLGS реализован как параллельный формат и позволяет процессам, использующим CCF API, одновременно записывать свои публикуемые данные в файл обмена. При этом вводится ограничение на превышение 40 одновременных писателей, чтобы излишне не замедлять файловую систему и сетевой интерфейс доступа к ней. Такое же ограничение накладывается и на параллельное чтение файла обмена ExLGS. В дальнейшем количество одновременных писателей/читателей должно быть вынесено на уровень конфигурации совместного расчета в режиме файлового обмена.

Использование интерполяторов

Для расчета значений сеточных величин ячеек целевой расчетной сетки по значениям сеточных величин заданного количества геометрически наиболее близких ячеек исходной расчетной сетки используются интерполяторы.

Программный интерфейс API CCF позволяет работать с интерполяторами двух типов:

- поверхностный интерполятор FSI (FluidStructureInteraction), который позволяет связывать подзадачи через общую интерфейсную поверхность;
- объемный интерполятор Remap, который позволяет связывать подзадачи, рассчитываемые в общем или пересекающемся объеме.

Используемая для связывания методик в режиме MPI-обменов библиотека FSI-интерполятора

«ccf_i_fsi» разработана на базе FSI-интерполятора из библиотеки UnionLib.

При переносе данных с одной сетки на другую посредством файлового обмена используется разработанный программный модуль «ccf_i_remap» на базе интерполятора Remap.

Интерполяторы реализованы на языке программирования C++ с использованием контейнеров STL [4] и специализированной библиотеки работы с поисковыми K-мерными деревьями в виде динамических библиотек, содержащих API интерполятора известного вида.

Обмен командами

Для согласования тактики счета между методиками служат обмены командами. Управляющие команды для согласования тактики счета и обмена состояниями методик посылаются и принимаются методиками в общем случае на каждой итерации совместного расчета и обеспечивают:

- согласование времени следующей точки синхронизации для обмена данными;
- обработку различных ситуаций, а именно:
- повторный расчет итерации;
- синхронизацию чтения/записи данных для перезапуска совместной задачи;
- аварийный останов методик и совместного расчета.

Заключение

В докладе описан модуль транспортного уровня CCF, который позволил обеспечить взаимодействие между собой математическим методикам ЛОГОС в процессе расчета связанной задачи, представлены основные возможности и принципы работы данного модуля.

Литература

1. ЛОГОС [Электронный ресурс]: Пакет программ инженерного анализа суперкомпьютерного моделирования – РФЯЦ – ВНИИЭФ, [2017-2018] – Режим доступа: <http://logos.vniief.ru>.
2. MPI Documents [Electronical resource]: Official version of MPI Standard MPI-2.2 / Bill Gropp, University of Illinois. – [S.l.], 2009. Mode of access: <https://mpi-forum.org/docs/mpi-2.2/mpi22-report.pdf>.
3. YAML Ain't Makeup Language [Electronic resource]: Data serialization standard / Evans Clark . – [S.l.], YAML 10, 2009 – Mode of Access: <http://yaml.org/spec/1.2/spec.html>.
4. Standard Template Library [Electronic resource]: software library for C++ programming language. – Niskayuna: GE Research, [1994-2017] – Mode of Access: <http://cppreference.com>.