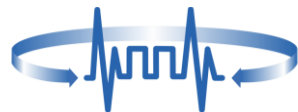


Интеграционная платформа для решения
крупномасштабных научных и инженерных
задач: модель выполнения, архитектура,
диспетчеризация

Доклад о работе по проекту РФФИ № 15-29-07043

Прохоров А. А. Назаренко А. М. Пересторонин Н. О.



DATAADVANCE

Постановка задачи



- Расчет является интеграционным – состоит из некоторого количества интегрированных программных расчетных средств, которые работают в заданной последовательности и обмениваются данными. Решение задачи состоит из решения слабосвязанных в смысле интенсивности взаимодействия подзадач.
 - Пример: МКЭ-расчет: САПР (CAD) → сеткопостроитель → препроцессор → солвер → постпроцессор.
- Расчет является итеративным – некоторые расчетные средства управляют работой других расчетных средств.
 - Пример: оптимизатор управляет процессом МКЭ-расчета
- Расчет является распределенным – некоторые расчетные средства не могут быть развернуты на облачной инфраструктуре и должны работать на ПК пользователя.
 - Примеры: лицензионные ограничения (расчетный код работает только на ПК пользователя); расчет содержит ПО, работающее под управлением разных операционных систем, привязка к инфраструктуре.

Разработка технологии создания облачной программной платформы, предназначенной для организации интеграционных, итеративных и распределенных расчетов.

Модель представления схемы расчета



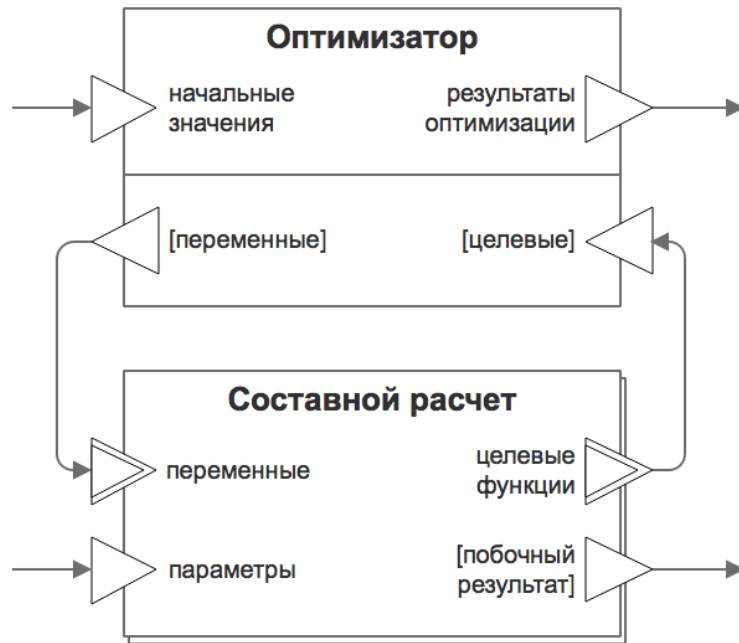
Модель схемы на примере итеративно-пакетного расчета

- Модель потока данных: блоки, порты, связи
- Ветвление и циклы
- Циклы описывают итеративные процессы: блок, управляющий циклом, явно определяет его начало и конец
- Пример: итеративно-пакетный расчет; обработка параметров цикла
- Автоматическое повторение параметров и накопление побочных результатов при итеративном и пакетном расчетах
- Составной блок: иерархия схемы; рабочая директория; пакетный расчет; кеширование данных; обработка ошибок времени исполнения



Модель схемы на примере итеративно-пакетного расчета

Циклы и пакетные расчеты определяют номер итерации, используемый при формировании иерархии контекстов хранения данных (файлов) расчетных средств



- Оптимизатор/
 - Расчет #0/
 - *Данные ПО*
 - ...
 - Расчет #1/
 - ...
 - Расчет #K/
 - Расчет #K+1/
 - ...
 - Расчет #N/
 - ...

- Назаренко А. М., Прохоров А. А. Типичные схемы комплексных инженерных и научных расчетов // Распределенные вычисления и Грид-технологии в науке и образовании. – 2016.
- Назаренко А. М., Пересторонин Н. О., Прохоров А. А. Управление файлами в рамках модели потоков данных для распределенных вычислений // Научный вестник МГТУ ГА. – 2016. Том 19. № 05. – С. 161-172.
- Назаренко А. М., Прохоров А. А. Иерархическая модель потока данных с автоматическим управлением файлами для инженерных и научных приложений // Национальный суперкомпьютерный форум. – 2015.
- Nazarenko A. M., Prokhorov A. A. Hierarchical Dataflow Model with Automated File Management for Engineering and Scientific Applications // Procedia Computer Science. – 2015. – Т. 66. – С. 496-505.
- Назаренко А. М., Прохоров А. А. Автоматизация многодисциплинарных инженерных расчетов с помощью потока работ // Национальный суперкомпьютерный форум. – 2014.
- Назаренко А. М., Прохоров А. А. Модель потока работ для автоматизации многодисциплинарных инженерных расчетов // Распределенные вычисления и Грид-технологии в науке и образовании. – 2014.
- Назаренко А. М., Прохоров А. А., Сухорослов О. В. Автоматизация инженерных расчетов с помощью потоков работ. // Информационные технологии и системы. – 2014.

Архитектура облачной платформы



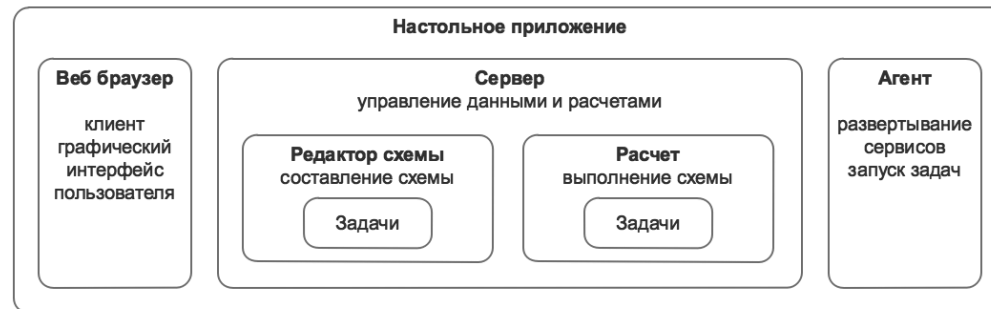
Общая схема архитектуры решения

- Варианты развертывания

- облачное (приложение как сервис, AaaS)
- настольное
- гибридное

- Некоторые особенности

- Повторное использование компонентов и общая схема работы во всех вариантах развертывания
- Совместная работа над одним расчетом
- Изоляция: процессы, контейнеры
- Интеграция расчетных средств, работающих под разными ОС



- Цели
 - Максимальная «автономность» реализации блоков
 - Максимальное упрощение разработки блоков
- Решение – интеграция на уровне протокола
 - Блок – отдельная программа, реализующая определенные HTTP-интерфейсы
 - Задача сама реализует свой HTTP-based GUI
 - На каждого пользователя по экземпляру блока
 - Блок работает с файловой системой для обмена данными с Платформой
 - Стандартные компоненты предоставляются блоку Платформой в виде веб-сервисов:
 - Ввод данных
 - Работа с файлами
- Как?
 - Симметричный двусторонний протокол основанный на websocket
 - Композиция UI от разных источников с помощью стека элементов iframe

- Давыдов А. В., Назаренко А. М., Пересторонин Н. О., Прохоров А. А. Гибридная настольно-облачная платформа для исследования пространства параметров // Программные продукты и системы. – 2016. – №. 2 (114).
- Davydov A. V., Morozov S. M., Perestoronin N. O., Prokhorov A. A., Saratov A. A. The First Full-Cloud Design Space Exploration Platform // Deutschsprachige NAFEMS Konferenz. – 2016.
- Davydov A. V., Morozov S. M., Perestoronin N. O., Prokhorov A. A. A First Full-Cloud Design Space Exploration Platform // Simulation Process and Data Management (SPDM), Computational Fluid Dynamics (CFD) – Beyond the Solve. – 2015. – P. 90-95.
- Назаренко А. М., Прохоров А. А. Архитектура системы управления потоками работ с возможностью облачного и настольного развертывания // Национальный суперкомпьютерный форум. – 2015.

Планирование и диспетчеризация



- Расчеты длительные и ресурсоемкие
- Расчетные схемы относительно просты – десятки блоков
- Сложность быстро растет при увеличении объемов данных
- Длительность работы отдельных задач плохо предсказуема
- Неизвестно количество итераций в циклах
- Нет практической возможности переноса вычисления между вычислительными ресурсами без потерь

- Выполнен обзор
 - Статические методы дают лучший результат, однако требуют выполнения двух условий:
 - Качественных оценок времени выполнения (не можем дать таких оценок)
 - Надежного резервирования временных окон на ресурсах (не можем на это рассчитывать)
 - На практике широко распространены динамические методы (Pegasus, Taverna, CLAVIRE)
 - Подавляющее большинство изученных алгоритмов рассчитаны на DAG-схемы. Есть подходы, основанные на преобразовании схемы к DAG (ASKALON Grid Environment...)
- Идеи
 - Использовать полную структуру схемы: априори известны ресурсоемкие задачи расчета; часто можно утверждать, что итераций будет много; схемы относительно простые; задача, управляющая циклом, никогда не работает одновременно с «телом» цикла, ...
 - Планировать не «к лучшему», а «от плохого» – не минимизировать время, а избегать фатальных конфликтов

- Назаренко А. М., Прохоров А. А. Планирование выполнения композитных приложений в гетерогенной среде // Распределенные вычисления и Грид-технологии в науке и образовании. – 2016.
- **Назаренко А. М., Прохоров А. А. Методы планирования выполнения композитных приложений в распределенных вычислительных средах // Национальный суперкомпьютерный форум. – 2016.**

ИТОГИ



- Проведен анализ специфики поставленных задач и сформированы требования
- Разработана модель представления (составления и интерпретации) схемы расчета: итеративные расчеты; схема хранения данных интегрируемых программных средств; встроенная обработка ошибок
- Предложена архитектура облачной программной платформы: поддерживаются различные схемы развертывания; задачи развертываются (в т. ч. на ПК) даже на этапе настройки расчета
- Предложенная модель потока работ и архитектура платформы создают предпосылки к расширению системы – минимизируются усилия на интеграцию новых расчетных средств в платформу, при этом не ограничивается функциональность – есть собственный GUI (веб интерфейс), более того, отсутствуют ограничения на используемые технологии и языки программирования
- Выполнен обзор методов планирования и диспетчеризации задач, ведется поиск подходящего

- Разработка подробной спецификации протокола жизненного цикла задачи
- Выбор или разработка языка описания расчетной схемы: Python DSL, YAWL, EasyFlow, ...
- Выбор конкретных технологий и создание открытой экспериментальной реализации
 - Динамическое проксирование от веб-браузера пользователя к HTTP-серверу задачи
 - Управление контейнерами: OpenStack, Docker swarm, Kubernetes ...
 - Синхронизация файлов между вычислительными узлами: librsync, распределенная файловая система с явной процедурой захвата и освобождения файлов, ...
- Возможно стоит подумать о больших данных
 - Hadoop HDFS
 - pNFS

Благодарности

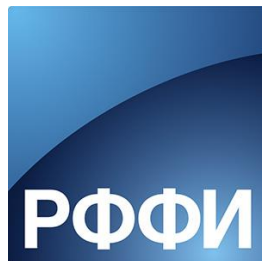
Работа выполнена при финансовой поддержке РФФИ в рамках научного проекта
№ 15-29-07043

Условия для выполнения работ по проекту предоставлены ООО «ДАТАДВАНС»
www.datadvance.net

Центр распределенных вычислений ИППИ РАН

www.distcomp.ru

Афанасьев Александр Петрович Сухорослов Олег Викторович



DATAADVANCE

Спасибо за внимание!

Прохоров Александр Александрович

+7 903 624 90 39

prokher@gmail.com

Научный сотрудник, ИППИ РАН

Начальник отдела разработки ПО, ООО «ДАТАДВАНС»
