

Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ Н.Э.
БАУМАНА»
(МГТУ им. Н.Э. БАУМАНА)



Громов Иван Сергеевич

Программная реализация удалённого запуска графоориентированных решателей систем инженерного анализа

Научный руководитель: доцент РК-6, к.ф.-м.н.,

Соколов Александр Павлович

ПРЕЗЕНТАЦИЯ (курсовая работа)

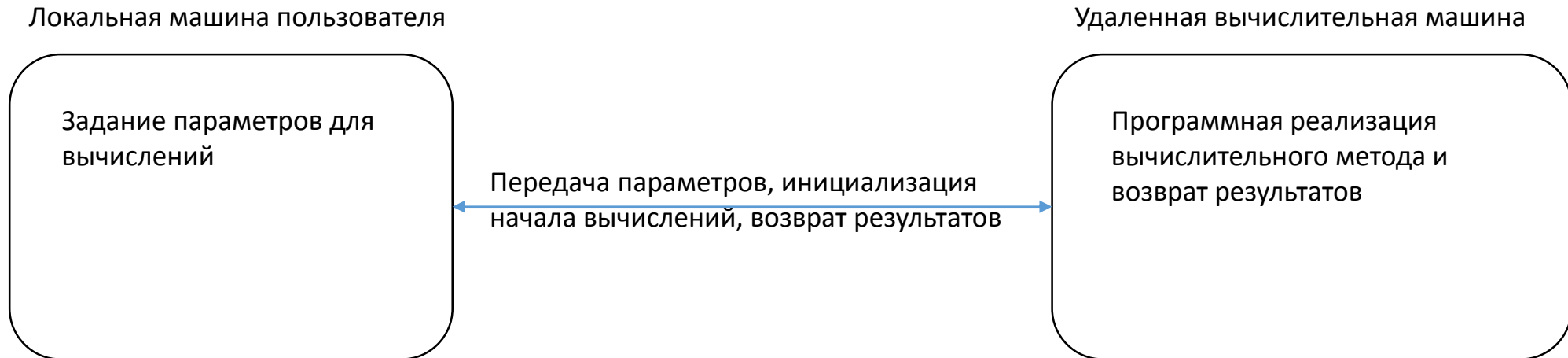
Программная реализация удалённого запуска графоориентированных решателей систем
инженерного анализа

Москва, 2018

Введение

- Актуальность:

При решении сложных ресурсоёмких задач зачастую вычислительных мощностей локального персонального компьютера уже не хватает и возникает потребность в использовании многопроцессорных удалённых машин. Задача направлена в целом на задействование разработанной в МГТУ им. Н.Э. Баумана специальной технологии построения программных реализаций сложных вычислительных методов, основанной на применении понятий теории графов. Технология позволяет систематизировать процесс разработки программных реализаций сложных вычислительных методов.



Введение

Цель и содержание лекции

Объект исследований: Реализация удалённого запуска графоориентированных решателей систем инженерного анализа.

Цель работы: Обеспечить возможность удалённого запуска GBSE решателей.

Задачи работы:

1. Провести обзор литературы по теме: «Технологии и методы удалённого запуска процедур и функций на высокопроизводительных вычислительных системах».
2. Разработать схему архитектуры подсистемы, обеспечивающей удалённый запуск графоориентированных решателей.
3. Создать тестовую функцию системы, с помощью которой можно будет осуществить запуск решателя, реализованного с использованием графоориентированного подхода.
4. Обеспечить поддержку привязки конкретного решателя к кнопке "Обработать" произвольной функций системы (AI).
5. Провести тестирование и отладку полученного решения.

Постановка задачи

Результат анализа литературы

В результате аналитического обзора литературы был изучен механизм вызова удаленных процедур(сокр. RPC – Remote Procedure Call), выявлены проблемы и трудности реализации методов удалённого вызова процедур и исследованы уже существующие программные реализации этих методов.

1. Механизм вызова удаленных процедур:

1. Передача данных и управления внутри программы, размещённой на физически разных машинах.
2. Асимметричность, т.е. одна из сторон обязательно является инициатором взаимодействия.
3. Синхронность, т.е. выполнение вызывающей процедуры приостанавливается с момента передачи запроса удалённой вычислительной машине и возобновляется только после завершения вызываемой процедуры.
4. Связность, т.е. RPC реализуем в системах, удалённые компоненты которых интерактивно связаны.

2. Проблемы реализации RPC:

1. Т.к. процедуры выполняются на разных машинах, имеющих различные адресные пространства – возникают проблемы при передаче параметров и результатов. Это означает, что параметры RPC не должны содержать указателей на ячейки памяти.
2. RPC обязательно должен использовать нижележащую систему связи, при этом это не должно отражаться ни на определении процедур, ни на самих процедурах.
3. Т.к. в реализации RPC участвуют две машины, то задействуется несколько исполняющих процессов – следовательно, возникают проблемы, при прерывании одного из них.
4. Могут возникнуть проблемы, связанные с неоднородностью языков программирования и неоднородностью операционных сред в части структур данных и структур вызова процедур.

Постановка задачи

Результат анализа литературы

3. Наиболее распространенные реализации RPC (В основном основаны на реализации функций – стабов, которые «упаковывают» данные, необходимые для передачи в определённую структуру и передают её другой машине, которая в свою очередь определяет необходимый стаб для выполнения и передаёт данные ему. Основное различие существующих реализаций в структурах передачи данных.):
 1. DCE/RPC – бинарный протокол, на базе различных сетевых протоколов.
 2. JSON-RPC – текстовый протокол на базе HTTP
 3. Sun RPC – бинарный протокол на базе TCP, UDP
 4. XML RPC – текстовый протокол на базе HTTP
 5. Java RMI – набор JAVA библиотек, реализующих RPC

Постановка задачи

Концептуальная постановка задачи

Для использования в нашей системе был выбран подход на основе бинарного протокола на основе сетевого протокола TCP. (как вариант можно использовать boost/mpi).

Результатом выполнения задачи должна стать реализация возможности «запускать» решатели графов из web-клиента, на удаленном вычислительном сервере. Клиент должен, путём ввода входных данных, формировать aINI файл, передаваемый на сервер приложений. Там файл должен быть прочитан и на его основе сформирована структура типа AnyMap. Затем, сервер приложений должен создать объект типа «граф» и инициировать его выполнение на вычислительном сервере и передать ему объект AnyMap, после чего ждать ответа от вычислительного сервера и на его основе сформировать ответ для web-клиента и отправить его.

Архитектура программной реализации



Программная реализация

Были разработаны прототипы функций на языке C++, реализующих RPC:

- Функция на стороне сервера приложений, принимающая на вход ссылку на объект класса AnyMap и Id решателя, и иницирующая запуск решателя на стороне вычислительного сервера, после чего ожидающая ответа от сервера вычислений и отправляющая его обратно web-клиенту.
- Функция на стороне вычислительного сервера, работающая в режиме ожидания инициации запуска решателя, после чего связывающая входные данные с нужным решателем, и возвращающая результат работы решателя серверу приложений.
- Функция на стороне web-клиента, передающая серверу приложений файл в формате aINI и ожидающая ответа от него.

Тестирование и отладка

- T.B.D.

Анализ результатов

- T.B.D.

Заключение

На текущем этапе выполнения курсовой работы была проанализирована и декомпозирована поставленная задача, выполнен аналитический обзор литературы, произведен выбор решения, подлежащего дальнейшей реализации, сформирован прототип архитектуры решения поставленной задачи, спроектированы и прототипированы программные функции, реализующие необходимый функционал. Осталось завершить программную реализацию, провести тестирование и отладку, анализ результатов и подготовку документации.

Заключение

Спасибо за внимание!