

Новосибирский национальный исследовательский государственный университет
Факультет информационных технологий
Кафедра параллельный вычислений

Разработка и реализация системы хранения и применения программных модулей для базы активных знаний

Выполнил студент гр. 15202

Артюхов Алексей Андреевич

Руководитель

д.т.н., зав. Каф. ПВ ФИТ НГУ, г.н.с. ИВМиМГ СО РАН

Малышкин В.Э.

Соруководитель

Перепёлкин В.А.

Новосибирск, 2019 г.

Актуальность автоматического применения знаний

- Для освоения пассивных знаний человеку требуется много времени
- Если знание не используется, то со временем оно утрачивается
- Программу, реализующую знание, в общем случае, невозможно автоматически модифицировать
- Компьютер с трудом и не всегда понимает неформальное описание знаний

Обзор

Онтологический подход:

- Удобно накапливать, хранить и передавать знания
- В общем случае по онтологии невозможно синтезировать прикладную программу

Логический синтез программ:

- Описать полную непротиворечивую теорию предметной области крайне затратно
- Методы машинного доказательства теорем являются вычислительно сложными и не позволяют получать прикладную программу с заданными нефункциональными свойствами

Обзор

Структурный синтез программ:

- Структура связей между используемыми в программе понятиями
- Удобная для обработки информация ↔ удобная для восприятия информация
- Оптимизация при выводе программы и явное описание нефункциональных свойств
- Низкая проработанность подхода на практике

Цель

Предлагается создать систему - Базу Активных Знаний (БАЗу), которая:

- Сохраняла бы **информацию о подпрограммах и о связях между ними, в активной форме**
- **Автоматически** синтезировала бы **оптимальную** прикладную программу

Задачи

- Разработать язык для внесения знаний в базу;
- Разработать формат хранения знаний;
- Разработать архитектуру БАЗы;
- Реализовать БАЗу;
- Составить вычислительную модель предметной области для тестирования системы;
- Провести тестирование базы активных знаний на предмет соответствия требованиям;

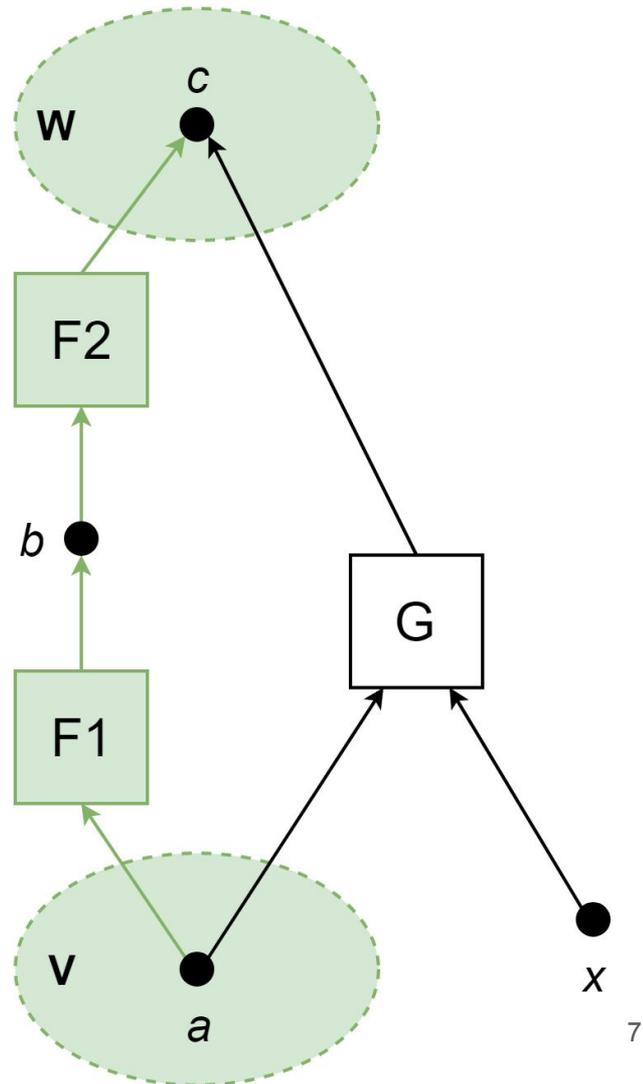
Необходимые определения

Вычислительная модель - это двудольный ориентированный граф, где вершины в одной доле соответствуют операциям, а в другой — переменным. Дуга графа, соединяющая операцию с переменной, определяет переменную как результат этой операции, а противоположно направленная дуга определяет переменную как входной параметр.

Постановка задачи состоит из кортежа переменных которые прикладная программа получает на вход, т.е. входных переменных, кортежа переменных которые будут являться результатом работы программы, т.е. выходных переменных, и вычислительной модели.

План вычислений - частично упорядоченное подмножество операций, где отношение порядка согласовано с информационными зависимостями и определяется как “вычисление А предшествует вычислению Б”.

Оптимальный план - план вычислений оптимизированный по определённому параметру.



Применяемый подход

Знания о подпрограммах хранятся в **операциях вычислительных моделей**.

Для добавления знаний используется **специальный язык**, на котором можно описать вычислительную модель, её операции и переменные.

Для применения накопленных знаний используются планы вычислений (ЧУМ операций).

Получив постановку задачи БАЗа выводит оптимальный план вычислений.

Предлагаемый язык описания ПВМ

Язык должен описывать простые вычислительные модели.

Описание разделено на 4 типа:

- описания вычислительных моделей
- описания операций
- описания структурных операций
- описания переменных

Все описания состоят из строк вида “ключ=значение”.

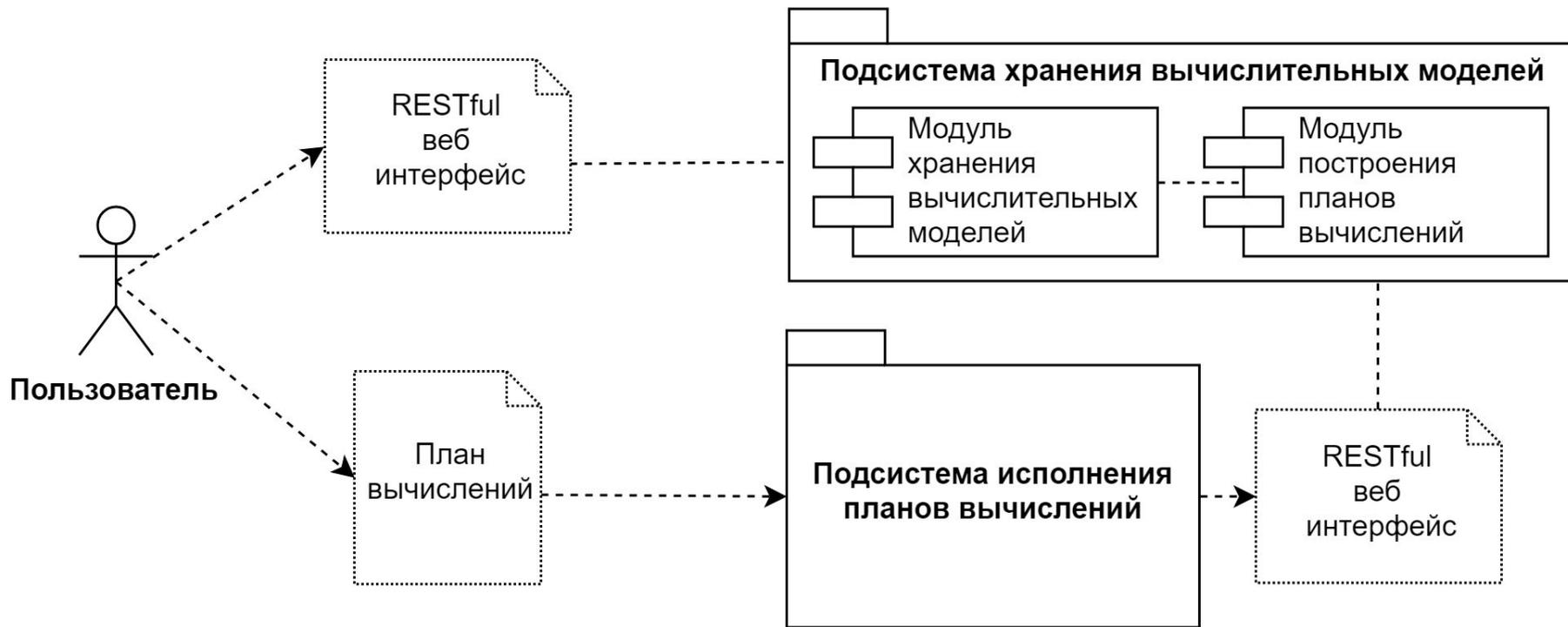
```
Name=MagneticInduction
Description=
Type=PythonScript
```

```
ComputationalModel=CM1
```

```
InputParameters=permeability,
magneticFieldIntensity
InputParameterTypes=Double, Double
InputParametersMapping=permeability,
magneticFieldIntensity
```

```
ResultList=magneticInduction
ResultTypes=Double
ResultParametersMapping=magneticIndu
ction
```

Архитектура



Модуль хранения

Соединён с базой данных, в которой хранятся вычислительные модели.

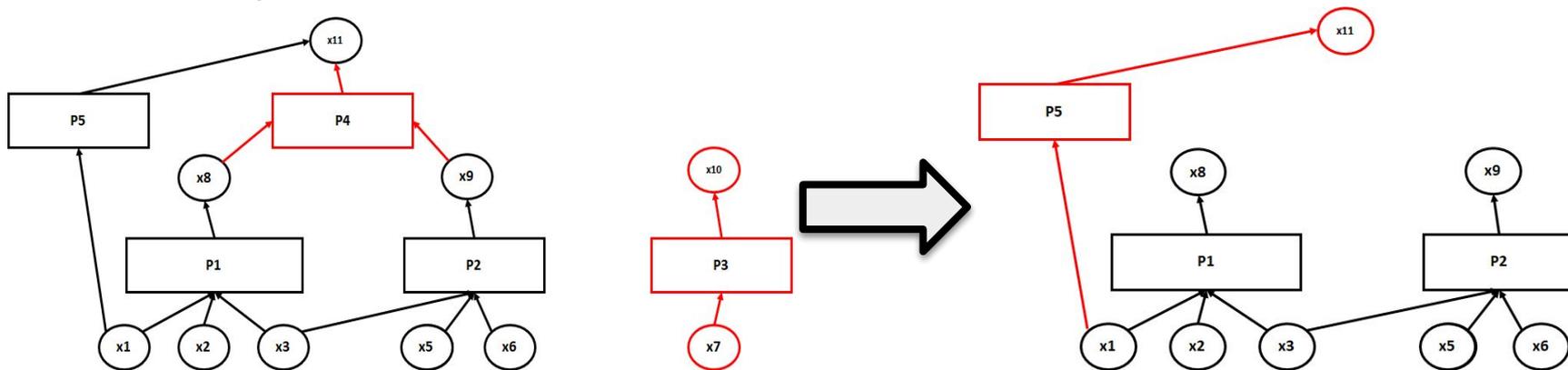
Система плагинов, для расширения набора поддерживаемых типов операций.

Перенаправляет запросы с постановкой задачи в модуль вывода плана вычислений.

Модуль вывода планов вычислений

3 стадии вывода: вычисляемые операции, необходимые операции, линейное упорядочивание подграфа.

В алгоритме вывода предусмотрено использование определённого критерия оптимизации.



Исполнение плана вычислений

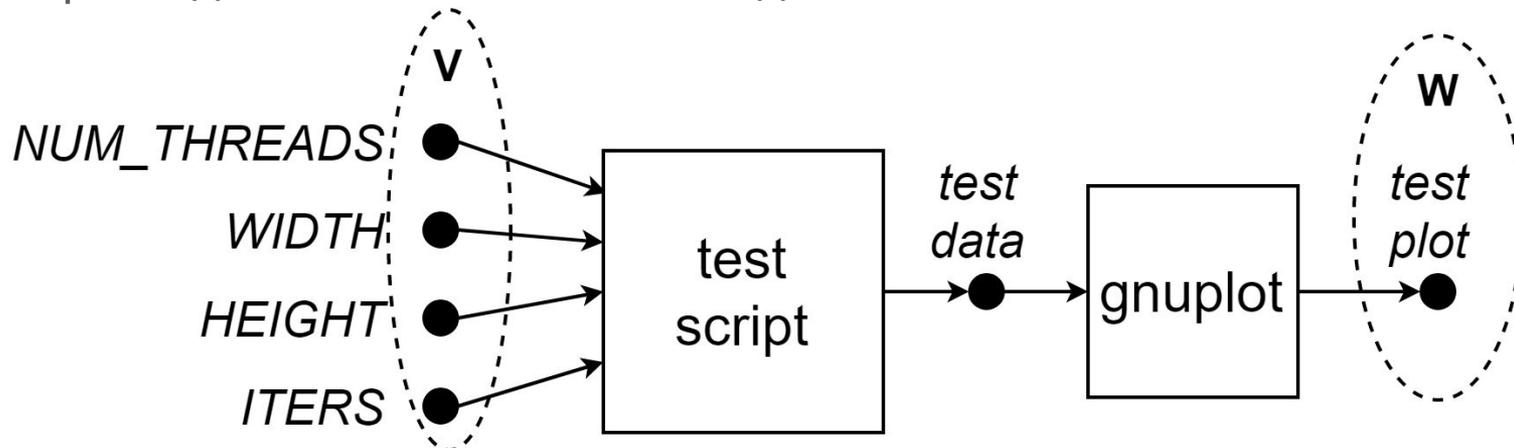
На исполнение могут быть взяты только те подпрограммы, для которых уже вычислены все входные переменные.

При исполнении плана система берёт первую подготовленную подпрограмму и отправляет её на исполнение.

При таком подходе вычисления будут идти практически последовательно, даже если в плане вычислений есть независимые части.

Тестовые вычислительные модели

Прикладная вычислительная модель:

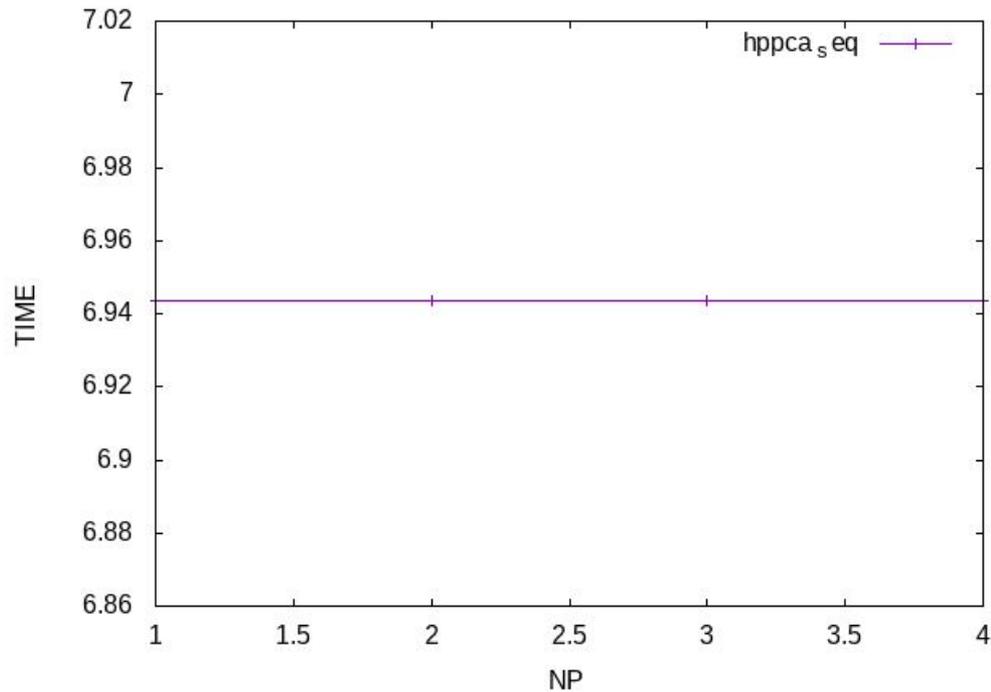


Синтетическая вычислительная модель:

- Порядка 100 операций связанных в сетку 10x10
- У каждой операции определены два нефункциональных параметра
- Операции разделены на две группы по главной диагонали

Тестирование

ITERS=100, .count=1, HEIGHT=2000, WIDTH=2000, .computer=alexey-VirtualBox, FGCC



Заключение

- Применена теория структурного синтеза программ для решения проблемы активных знаний
- Реализована система для накопления и автоматического применения знаний
- Проведено тестирование системы на показательной предметной области
- Работа участвовала в МНСК-2019 и была оценена дипломом 3-й степени

Планы на будущее

Добавить поддержку структурированных операций в модуль хранения и в модуль исполнения

Добавить в **модуль исполнения** плана вычислений возможность **выбора планировщика**

Реализовать **удобный пользовательский веб-интерфейс** для добавления знаний в БАЗу