ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Разработка и реализация алгоритмов вывода оптимального плана тестирования фрагментированных программ

Научный руководитель:
д.т.н., профессор,
заведующий кафедры ПВ ФИТ НГУ
Малышкин В.Э.

Выполнил:
Бедарев Н. А., 14204 гр.
Кафедра параллельных вычислений ФИТ НГУ

Соруководитель:
б/с, старший преподаватель,
Кафедра параллельных вычислений ФИТ
Перепелкин В.А.

Рецензент:
к.т.н., зав. лабораторией,
ИСИ СО РАН
Загорулько Ю. А.

Новосибирск, 2018 г.

Проблема

При использовании вычислительных кластеров пользователь сталкивается с множеством рутинных вспомогательных действий, таких как:

1. Передача различных файлов с локального компьютера на кластер и обратно (исходные файлы программ, данные необходимые для расчета, результирующие файлы работы и т.д.)
2. Установка вспомогательного программного обеспечения на кластер
3. Постановка задачи в очередь на исполнение, опрос очереди о статусе задачи и т.д.

Все эти действия, в своем большинстве, связаны с обилие деталей, носящих технический характер и требующих от пользователя знаний, лежащих за пределами его прикладной области.

Также некоторые задачи могут решаться с помощью различных наборов операций, которые обладают различными нефункциональными свойствами и, как следствие, являются более предпочтительными в определенных обстоятельствах.

Обзор. Системы автоматизации рутинных действий

На сегодняшний день существует множество систем автоматизации вспомогательных действий, таких как:

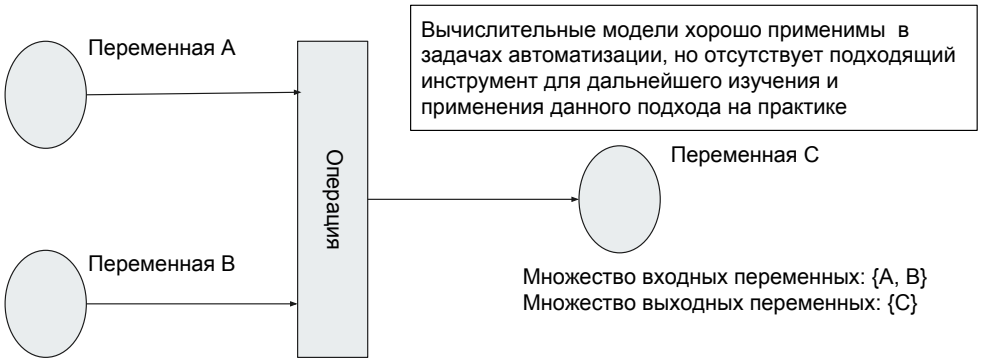
- Qmake/Cmake
- GNU make
- Apache Ant и т.д.

Минусы:

1. Отсутствует механизм описания альтернативных способов получения промежуточных результатов
2. Отсутствует механизм позволяющий описывать и в дальнейшем учитывать нефункциональные свойства операций
3. Отсутствует поддержка работы с удаленными вычислителями

Обзор. Вычислительные модели

Вычислительная модель[1] (ВМ) — это двудольный ориентированный граф, включающий операции и переменные, связанные дугами информационных зависимостей, и описывающий схему вычислений значений одних переменных из других.



Вычислительные модели хорошо применимы в задачах автоматизации, но отсутствует подходящий инструмент для дальнейшего изучения и применения данного подхода на практике

1. Victor Malyshkin. Active Knowledge, LuNA and Literacy for Oncoming Centuries // LNCS, Vol. 9465. 2015. P. 292-303.

Цель и задачи

Создание инструмента, позволяющего применять математический аппарат вычислительных моделей для автоматизации рутинных действий.

Поставленные задачи:

1. Проектирование системы. Разработка синтаксиса языка описания вычислительных моделей.
2. Разработка алгоритмов работы системы.
3. Реализация системы – интерпретатора вычислительных моделей.
4. Тестирование работоспособности системы.

Формулировка требований к системе

1. Возможность описывать вычислительные модели
   a. Простота описания
   b. Расширяемость
2. Поддержка многовариантности получения результата
3. Заменяемость алгоритма построения сценария, который в дальнейшем будет выполняться интерпретатором.

Проектирование системы. Разработка синтаксиса

- В данной работе используется расширенное понятие вычислительной модели, которое было введено в работе Прокопьевой А.В., с помощью введения понятия атрибутов операций.
- Для описания атрибутов операций вводится понятие “меток”. Метка – пара (наименование, значение).
- Под каждой переменной вычислительной модели мы будем понимать именованное состояние, которое задает состояние автоматизируемого процесса

На основе языка был разработан синтаксис описания вычислительных моделей.

Вычислительная модель описывается в текстовом файле с расширением *.cm. Строки вычислительной модели имеют следующий формат:

<входные_состояния>;<выходные_состояния>;<операция>#<метки>

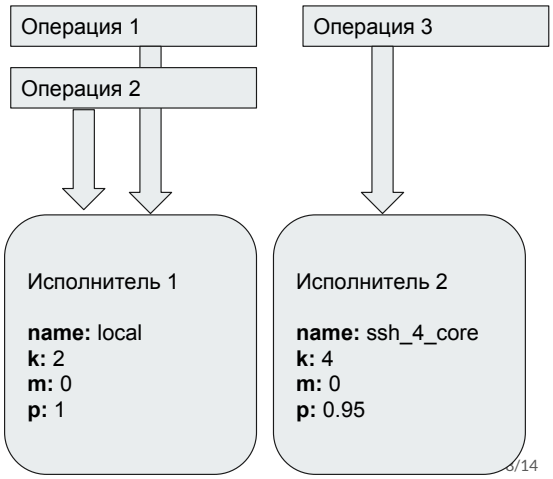
Метки имеют формат: <название>=<значение>

Проектирование системы. Исполнитель

В связи с тем, что исполнение операций на различных вычислителях имеет свою специфику, была разработана абстракция Исполнителя, отвечающая за исполнение и контроль исполнения операций.

Исполнитель задается следующими характеристиками:

- Название
- **k** – коэффициент производительности вычислителя
- **m** – коэффициент дополнительной нагрузки на вычисления
- **p** – вероятность успешного завершения выполнения команды



Проектирование системы. Формализация нефункциональных свойств

Предлагается привести множество нефункциональных свойств операций к одному интегральному весу.

Для это разработан интерфейс “весов”. Весы – объект, который производит отображение множества меток на на множество вещественных чисел.

Далее на основе формулы F(k, m, p, w) рассчитывается вес операции.

F(k, m, p, w) = (w/k + m)/p, где:

- w – вес операции, полученный с помощью весов.

Алгоритм построения сценария

Были разработаны два алгоритма построения сценария работы интерпретатора:

1. Переборный алгоритм, находящий оптимальный сценарий работы.
2. “Быстрый” алгоритм, на основе алгоритма Дейкстры, дающий не всегда оптимальный, но на практике обычно близкий к нему результат.

Алгоритм реализуется заменяемым программным модулем, что позволяет разрабатывать и встраивать другие алгоритмы, ориентированные на те или иные классы задач

Реализация

Была разработана система на языке Java, по причине его широкого распространения и кроссплатформенности. Для работы системы необходима версия Java 8.

Для удобства работы был введен дополнительный функционал в виде:

- Циклов (for)
- Определений (define)
- Включений (include)

Поддержаны метки отвечающие за:

- Установку корректного кода возврата команды.
- Установку весов операций, задающих оценочную занимаемую оперативную память (memory) или оценочное процессорное время (time).
- Маркировку временных файлов, которые необходимо удалить после завершения работы.

Тестирование

1. Для тестирования работы системы были разработаны тесты, показывающие:
   - Соответствие работы системы и описанию вычислительной модели
   - Оптимизацию сценария работы по заданным нефункциональным требованиям
2. Произведено тестирование на прикладной задаче по установке программного комплекса LuNA, разработанной Прокопьевой А.В. в качестве одной из задач выпускной квалификационной работы
   - В ходе тестирования были выявлены недочеты реализации, требующие доработки. В частности, необходима поддержка обработки ситуаций некорректного выполнения команд

Апробация работы

1. 54-я Международная Научная Студенческая Конференция (МНСК 2016 г.)
2. XVII Всероссийская конференция молодых ученых по математическому моделированию и информационным технологиям (2016 г.)
3. 55-я Международная Научная Студенческая Конференция (МНСК 2017 г.)
4. Девятая Сибирская конференция по параллельным и высокопроизводительным вычислениям, 10-12 октября 2017 г., Томск, Россия
5. 56-я Международная Научная Студенческая Конференция (МНСК 2018 г.)

Заключение

1. Был разработан инструмент, позволяющий применять математический аппарат вычислительных моделей из теории структурного синтеза программ для автоматизации рутинных действий
2. Был разработан синтаксис описания вычислительных моделей
3. В качестве модулей интерпретатора были разработаны и реализованы модули исполнителей и весов
4. Разработаны алгоритмы вывода сценария, в виде заменяемых модулей системы