

Разработка и реализация алгоритмов исполнения фрагментированных программ с заданным поведением

Выполнил: Тренин С. А.
ФИТ НГУ, гр. 15205

Руководитель: д.т.н., проф. Малышкин В. Э.
зав. каф. ПВ ФИТ НГУ

Новосибирск 2019



Введение

Высокоуровневое программирование снижает порог вхождения в технологии, тем самым облегчая разработку программ для таких сложных отраслей, как численное моделирование. Но для широкого класса задач высокоуровневое программирование малоприменимо в силу невысокой производительности.

Основная причина невысокой производительности – фундаментальная проблема выбора поведения (конкретного варианта исполнения программы из множества допустимых вариантов, отличающихся нефункциональными свойствами).



Обзор существующих решений

Применяются следующие подходы:

1. Введение в язык специальных операторов и директив, задающих прямое управление. При этом уровень программирования снижается (SMP Superscalar, ProActive Parallel Suite).
2. Сужение предметной области, на которую ориентирована система (PaRSEC, TBLAS).
3. Использование обоих вышеуказанных подходов (LuNA, Legion, Julia).



Система LuNA

Система автоматизации генерации параллельных программ на основе технологии фрагментированного программирования. Управление поведением вынесено отдельно (изменение поведения не затрагивает функциональные свойства программы) и автоматизировано, что делает систему удобной для исследования вопросов управления поведением.



Результаты обзора существующих решений

Дальнейшее развитие специализированных операторов и директив вносит вклад в автоматизацию выбора поведения. Расширение же области применения специализированных языков, как видно на примере Julia, порождает в нём новые специализированные операторы.

Проблема выбора достаточно хорошего поведения полностью не решена ни в одной из существующих систем. Таким образом, разработка новых средств управления поведением и поддерживающих их алгоритмов является актуальной задачей.



Постановка цели и задач

Цель – разработка средств задания поведения и системных алгоритмов для системы LuNA, поддерживающих эти средства.

Задачи:

1. Выбрать средства управления поведением во фрагментированных программах.
2. Разработать алгоритмы, реализующие выбранные средства в системе LuNA.
3. Провести эксперименты на конкретных тестах.



Необходимые определения

В технологии фрагментированного программирования (ФП) исполняющаяся программа состоит из множества ФВ, связанных информационными связями.

Фрагмент данных (ФД) – переменная единственного присваивания.

Фрагмент вычислений (ФВ) – операция единственного срабатывания, для исполнения которой необходим определённый набор входных ФД.

Если один ФВ вырабатывает некоторые ФД, необходимые другому ФВ, то эти ФВ будут информационно зависимыми.



Предлагаемое решение

В результате анализа были выявлены основные аспекты исполнения программ, для которых требуются средства управления поведением:

1. Распределение фрагментов вычислений по вычислительным узлам.
2. Динамическая балансировка нагрузки.
3. Сборка мусора.
4. Выделение и распределение ресурсов.

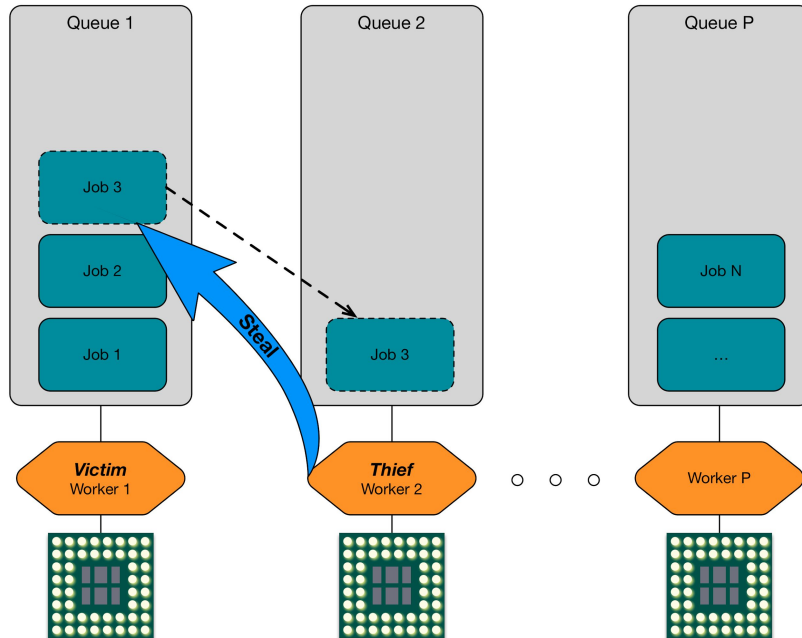


Динамическая балансировка нагрузки

На данный момент в системе LuNA реализована динамическая балансировка методом Rope of Beads, который реализует алгоритм диффузионного типа с линейной структурой. Такая балансировка в первую очередь ориентирована на устранение дисбаланса, возникающего сразу на многих узлах, но плохо подходит для устранения точечного дисбаланса, когда значительный дисбаланс возникает на отдельных узлах.

В результате анализа существующих методов балансировки был выбран метод WorkStealing, так как он относится к другому типу балансировок и устраняет точечный дисбаланс лучше, чем Rope of Beads.

Метод WorkStealing

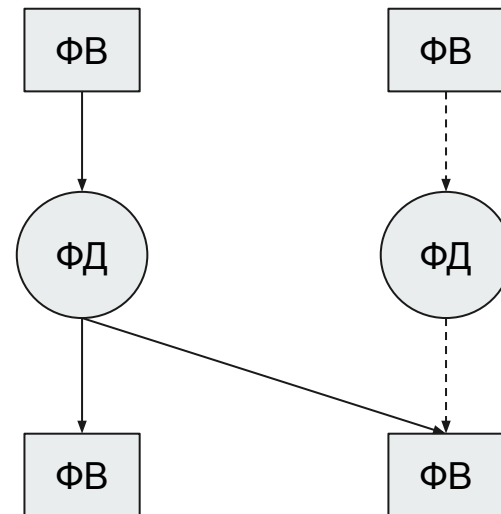


Управление распределением ресурсов

На данный момент все вычисленные ФД перераспределяются run-time системой по узлам. Перед исполнением ФВ запрашивает с прочих узлов все недостающие ему ФД, чтобы, исполнившись, породить ФВ и ФД. ФВ удаляются после исполнения, ФД – сборщиком мусора.

Предложенные средства:

1. Кэширование ФД на тех узлах, где они были выработаны или запрошены.
2. Перевычисление выработанных ФД из ФВ на узле вместо отправки ФД, для экономии коммуникаций.





Порядок исполнения фрагментов вычислений

На данный момент в системе LuNA порядок вычислений устанавливается информационными зависимостями.

Предложенные средства:

1. Система приоритетов для ФВ (по умолчанию 0, можно задавать целое значение). При прочих равных в рамках узла первым исполнится ФВ с большим приоритетом.
2. Установка строгого порядка исполнения для подмножества ФВ путём добавки фиктивного ФД, передаваемого из одного ФВ в другой.



Список предложенных средств

1. Распределение фрагментов вычислений по вычислительным узлам.
 - a. Приоритеты
 - b. Строгий порядок
2. Динамическая балансировка нагрузки.
 - a. WorkStealing
3. Сборка мусора.
4. Выделение и распределение ресурсов.
 - a. Кэширование
 - b. Перевычисление

Таким образом, мной были выбраны три основных пути управления поведением, ещё не реализованные в системе LuNA, наиболее важные для дальнейшего исследования вопросов задания поведения и исполнения фрагментированных программ.



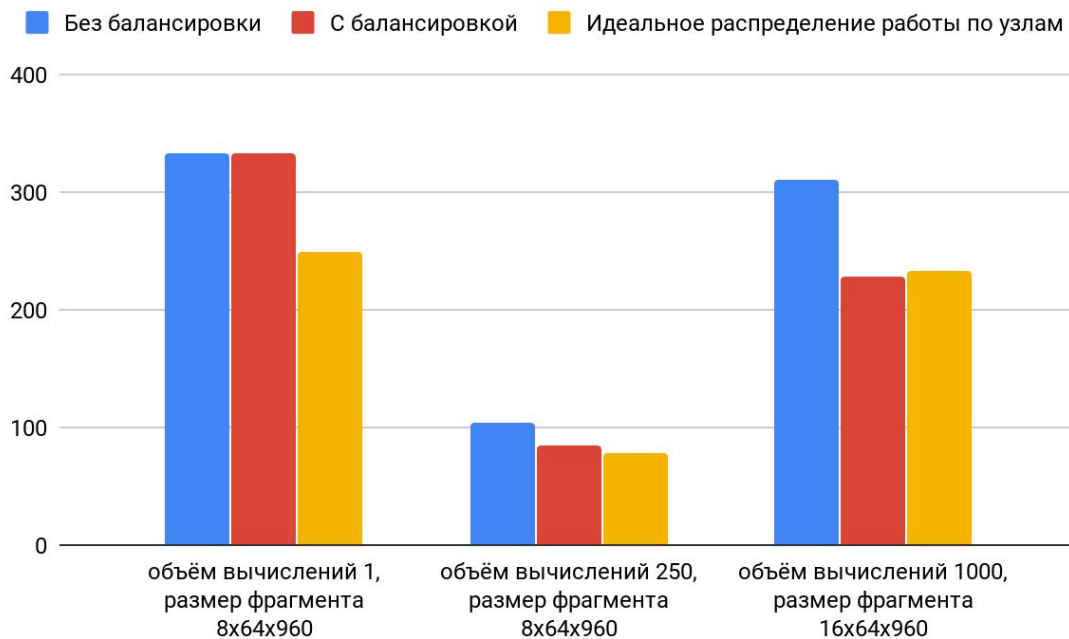
Тестирование

Для тестирования введённой динамической балансировки нагрузки по методу WorkStealing была взята программа на языке LuNA, решающая модельную задачу, реализующую итерационный метод с параметризованным объёмом вычислений в узлах сетки.

Запуск производился на кластере МВС-10П Межведомственного Суперкомпьютерного Центра. Задача запускалась на 4 вычислительных узлах с 2 потоками исполнения на каждом. При этом, чтобы наглядно продемонстрировать эффект динамической балансировки, один из узлов изначально не был загружен работой.

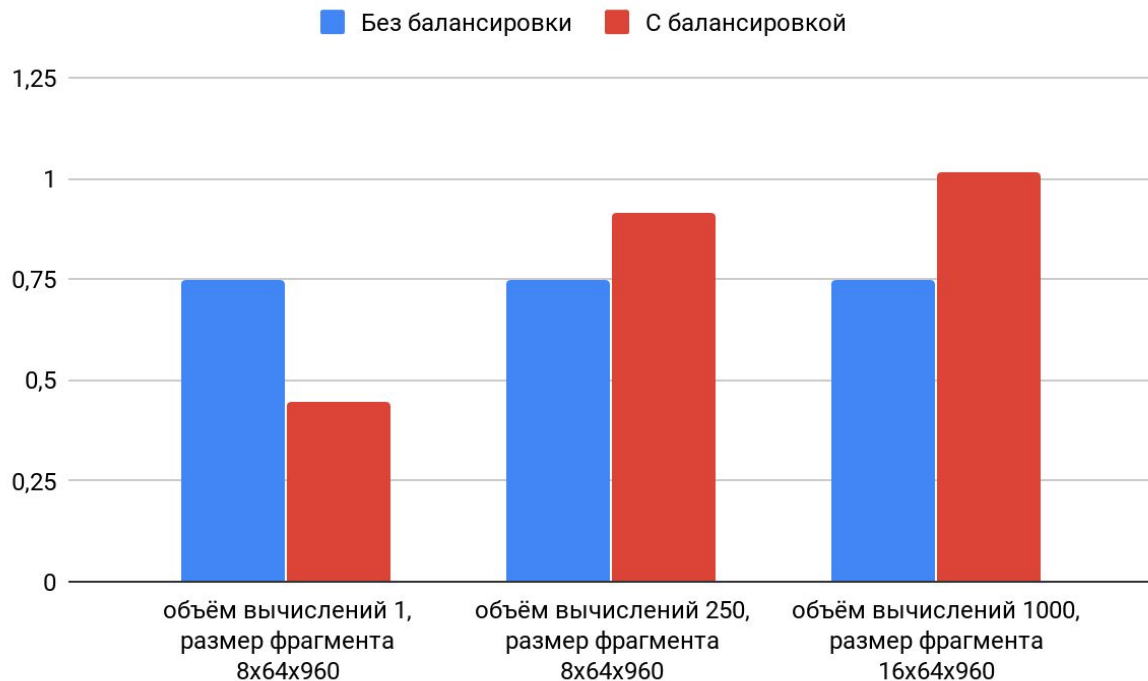
Тестирование

На графике представлено абсолютное время исполнения программы в секундах для различного объёма вычислений с включённой балансировкой, без неё и для идеального случая (все узлы равномерно загружены работой).



Тестирование

На графике представлено отношение времени выполнения программы к идеальному времени (когда все узлы равномерно загружены работой) для различного объёма вычислений.





Апробация работы

Участие в школах и школах-конференциях, проводимых лабораторией синтеза параллельных программ ИВМиМГ СО РАН, кафедрой ПВ НГУ и кафедрой ПВТ НГТУ:

1. XXVIII Зимняя школа по параллельному программированию (Новосибирск, 30.01–03.02.2017).
2. Летняя международная XXIX молодёжная Школа-конференция по параллельному программированию (Новосибирск, 03.07–14.07.2017).
3. Летняя международная XXXI молодёжная Школа-конференция по параллельному программированию (Новосибирск, 02.07–13.07.2018).
4. XXXII Зимняя школа по параллельному программированию (Новосибирск, 28.01–01.02.2019).

Очный доклад на Международной научной студенческой конференции 2019 г. на тему “Автоматизация динамической балансировки нагрузки методом WorkStealing для фрагментированных программ на языке LuNA”.

Публикации по теме:

1. Тренин, С. А., Чмиль, А. В. Автоматизация динамической балансировки нагрузки методом WorkStealing для фрагментированных программ на языке LuNA // Информационные технологии : Материалы 57-й МНСК 14–19 апреля 2019 г. [Электронный ресурс]. – Новосибирск: ИПЦ НГУ, 2019. – С. 112. – Режим доступа: https://issc.nsu.ru/upload/2019/Информационные_технологии.pdf



Заключение

Защищаемые положения:

1. Разработаны средства управления поведением во фрагментированных программах.
2. Разработаны алгоритмы, реализующие выбранные средства в системе LuNA.
3. Часть алгоритмов реализована в виде модулей системы LuNA.
4. Проведено экспериментальное исследование реализованных алгоритмов.

В результате проведённой работы на базе системы LuNA были разработаны средства задания поведения и добавлены системные алгоритмы, поддерживающие эти средства.

Далее планируется улучшение выбранных алгоритмов на основе профилирования. Кроме того, планируется дальнейший подбор параметров балансировки (возможно, с дальнейшей автоматизацией выбора этих параметров на основе профилирования).