**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Саратовский государственный технический университет**

**имени Гагарина Ю.А.»**

Институт прикладных информационных технологий и коммуникаций

Направление «Информационные системы и технологии»

Кафедра «Прикладные информационные технологии»

**ЗАЧЕТНАЯ РАБОТА**

по дисциплине «Основы грид - вычислений»

Выполнил студент группы

Б1ИФСТ41

очной формы обучения

Акутин Артем Сергеевич

Номер зачетной книжки 161211

Проверил: доцент кафедры ПИТ

к.т.н. Ермаков А.В.

Саратов 2020

*Описание задачи:*

Задача коммивояжера

В качестве исходных данных задается матрица смежности графа, описывающая узлы графа (города) и вес ребер (стоимость дорог между городами. Необходимо найти дорогу с наименьшей суммой ребер, при этом проходящую через все города и возвращающуюся к исходному городу. Задача заключается в поиске самого выгодного маршрута, проходящего через все города. Маршрут может проходить каждый город не один раз, стартовый город может быть выбран произвольно.

*Решение:*

### Указание на способ распределения задачи

Способ разделения состоит в следующем: граф представляет собой набор вершин, соединенных дорогами. Итого, у нас есть множество вершин. Путь представляет собой определенную перестановку всех вершин, содержащую все эти вершины. К примеру, если у нас есть множество вершин{ A, B, C, D, E }, то путь может выглядеть как: ABDCE

В данном случае, в нашей программе, которая генерирует задачи, мы должны определить некоторый параметр, который будет варьироваться в зависимости от сложности задачи и размера графа.

Задача будет разделена следующим образом: какую-то часть всего пути программа будет считать сама. Назовем это “головой” перестановки. А все остальные перестановки с фиксированной головой и минимальный путь для данных перестановок будет посчитан при помощи грид-системы. К примеру, если учитывать вышеприведенное множество, то мы можем взять размер головы равный 2 вершинам, создав начало пути как AB, а остальные перестановки с началом AB просчитать на грид-системе (ABCDE, ABDEC, ABEDC и т.д.)

Голова перестановки и будет являться нашим варьируемым параметром, в зависимости от размера графа мы будем увеличивать начальный размер пути или уменьшать. На графе же будут считаться все пути с заданными начальными вершинами

### Мотивация обеспечения эффективности распределения

Грид-вычисления - подход интересный, но далеко не всегда он будет эффективным. В грид - вычислениях узким местом является время пересылки по сети, время разделения задачи на подзадачи. Поэтому в первую очередь нужно определить при каком количестве вершин, имеет смысл разделение задачи на подзадачи

Ниже приведены результаты небольшого исследования, показывающее как растет время выполнения полного перебора для решения задачи коммивояжера:

3 вершины: 87 мс

4 вершины: 105 мс

5 вершин: 107 мс

6 вершин: 150 мс

7 вершин: 478 мс

8 вершин: 2456 мс

9 вершин: 80593 мс

10 вершин: 290346мс

11 вершин: 1445329 мс

По результатам видно, что при 9 вершинах время значительно увеличивается. Поскольку асимптотическая сложность полного перебора для решения задачи коммивояжера может быть выражена следующей зависимостью:

O((N/2)!),

где N - количество вершин. Экстраполируя, можно понять, что время будет лишь существенно увеличиваться

При разделении задачи на подзадачи способом, который описан в п.1, мы получим следующую асимптотическую зависимость:

O((N/2 \* k)! + (k/2)!),

где k - количество начальных вершин в перестановке (голова перестановки из п.1)

Мы видим, что зависимость явно положительная. Итого, при при количестве вершин более 9 и менее 66 имеет смысл разделение задачи на подзадачи и применение грид - вычислений (при количестве вершин меньше 9 задача быстро решается на 1 узле, при количестве вершин более 65 решение полным перебором не представляется возможным, поскольку задача коммивояжера относится к классу трансвычислительных задач).

Предположим, у нас есть граф, размерностью в 11 вершин. Исходя из приведенных выше чисел, полный перебор для поиска гамильтонового цикла с кратчайшей длиной пути (решение задачи) займет 1445329мс. Если мы возьмем размер головы перестановки в 4 элемента, то мы получим, что на узле, который генерирует задачи, нам надо затратить время в 105 мс, а на узлах, решающих задачи - 478мс. Поскольку ПО, которое занимается генерацией подзадачи, умеет работать с грид-системой асинхронно (после генерации задача мгновенно отправляется в грид-систему, не блокирую генерацию следующей задачи), то на решение при помощи грид узла мы потратим следующее время:

105мс + 478мс + K \* T,

где K - количество подзадач ( тут оно равно 4! = 12),

Т - время пересылки по сети

Время пересылки по сети - параметр зависящий не от нас, а от нагрузки сети в текущем сегменте интернета. Возьмем очень неоптимистичный вариант развития, когда время пересылки по сети займет 10000мс. Таким образом, задача коммивояжера для графа из 11 вершин решенная при помощи грид-системы займет 105 + 478 + 12\*10000 = 120583, что **на порядок меньше** времени, которое будет затрачено для решения задачи на 1 узле. И это если все задачи будут решаться блокируюшим образом! А асинхронный способ только ускорит этот процесс

Исходя из приведенных выше посылок, смею сделать вывод, что решение задачи коммивояжера при помощи грид вычислений имеет смысл при графе с кол-вом вершин более 9 и менее 66, при это голова перестановки должна быть размером не более 9 вершин. Параметр может быть варьируемым в зависимости от мощности железа, на котором работает основная программа, и нужд конкретного пользователя.

### Описание развернутой распределенной вычислительной среды

### Скриншоты выполнения примера задачи