МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ Кировоградского национального технического университета

Механико-технологическом

Кафедра ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Параллельные и распределенные вычисления

курс лекций

с элементами кредитно - модульной системы организации учебного процесса

для студентов дневной и заочной формы обучения по направлению подготовки 6.050102 «Компьютерная инженерия»

Составитель:

Доцент Смирнова Н.В.

Кировоград 2015

ВВЕДЕНИЕ

Для решения многих задач (прогноз погоды, задачи гидро- и газодинамики, квантовой химии, астрономии, спектроскопии, биологии, ядерной физики) необходима высокая производительность и высокая скорость передачи информации по каналам связи, большие объемы оперативной и постоянной памяти "памяти, которые не могут обеспечить обычные вычислительные средства. Одним из путей обеспечения таких требований является организация параллельных и распределенных вычислений и соответствующих технических средств их реализации.

Причем, эффективность параллельной обработки зависит как от производительности компьютеров, так и от размеров и структуры памяти, пропускной способности каналов связи, используемых языков программирования, компиляторы, операционных систем, численных методов и других математических исследований. Такой широкий объем параметров требует проведения исследований на различных уровнях: на уровне распараллеливания алгоритмов, создание специальных языков программирования, компиляторы, многопроцессорных систем, неоднородных систем, кластеров и систем, распределенных на больших территориях.

Целью изучения дисциплины является усвоение основных методов и алгоритмов организации параллельных и распределенных вычислений, принципов построения соответствующих структур, приобретение начальных практических навыков проектирования таких средств.

В ***результате изучения*** курсу студент повинен знати основні методи, алгоритми і засоби паралельної та розподіленої обробки інформації, засоби програмування на паралельних та розподілених структурах, склад апаратних засобів та програмного забезпечення обчислювальних систем з елементами паралельної та розподіленої обробки і класи мов програмування високого рівня для них; вміти виконувати елементарні вправи з розпаралелення задач та алгоритмів, проводити розрахунки параметрів процесорів, проектувати окремі вузли.

В розділі курсу “ ***Організація паралельних обчислень*** ” розглядаються такі основні питання:

- основні поняття про паралельні обчислення;
- методи оцінки продуктивності паралельних алгоритмів;
- розробка паралельного алгоритму;
- структури зв’язку між процесорами;
- основні класи паралельних комп’ютерів;
- схеми паралельних алгоритмів задач;
- мови паралельного програмування.

Тема №1: Основные понятия о параллельные вычисления

вопрос:

1. Понятие о параллельных и распределенные вычисления

2. Области применения и задачи параллельной обработки

3. Модели параллельных вычислений. Конвейеризация и параллелизм

4. Средства для проведения параллельных вычислений

5. уровни распараллеливания

6. параллельные операции

7. Основные принципы параллелизма (распараллеливания)

8. Классификация структур параллельной обработки Упражнения и задания к теме №1

1. Понятие о параллельных и распределенные вычисления

В зависимости от предметной области применения есть много определений терминов, характеризующих параллельные и распределенные вычисления. На основе анализа литературных источников и вариантов практической реализации можно так определить эти сроки:

параллельные вычисления - вычисления, поддерживаются на математическом, алгоритмическом, программном или аппаратном уровне (на всех или нескольких) и обеспечивают возможность параллельного выполнения задачи.

В [1] под термином "параллельные вычисления" понимается совокупность вопросов, относящихся к созданию ресурсов параллелизма в процессах решения задач и гибком управлению реализаций этого параллелизма с целью достижения наибольшей эффективности вычислительной техники.

распределенные вычисления - вычисления, которые поддерживаются стандартными или закрытыми протоколами обмена и независимыми аппаратными средствами (компьютеры, серверы), представляемых пользователю единственным вычислителем, пригодным для решения сложной задачи.

Относительно использованных ресурсов можно утверждать: в основном параллельные структуры реализуемых на специализированных процессорах, распределены структуры - на универсальных (стандартных) компьютерах (серверах), которые объединены в сети различного типа.

2. Области применения и задачи параллельной обработки

Есть круг вычислительных задач, для решения которых необходимы мощные вычислительные ресурсы, чем те, которые могут обеспечить обычные компьютеры или системы. В таких задачах необходимо обеспечить: *сверхвысокую быстродействие, большой объем оперативной памяти, большое количество информации, передаваемой обработку и хранение большого объема информации.* При наличии хотя бы одного из приведенных требований использования параллельной обработки оправдано.

примеры:

1. Сложные, многомерные задачи, которые необходимо решить в течение достаточно ограниченного времени, требуют обеспечения *сверхвысокой производительности,* например - задачи прогноза погоды. Область развязку (атмосфера) разбивается на отдельные пространственные зоны. Причем, для обеспечения прогноза на определенном периоде времени, вычисления в каждой зоне повторяется много раз. Если объем зоны равен 1 км ^{3}, то для моделирования 10 км атмосферы необходимо 5х10 ^{8}

таких зон. Предположим, что вычисления в каждой зоне требует 200 операций с плавающей точкой, тогда за один временной шаг необходимо выполнить 10 ^{11} операций с плавающей точкой. Для того, чтобы прогнозировать погоду с передбачуваннистю 10 дней с 10-ти минутным шагом компьютеру с производительностью 100 Mflops (10 ^{8} операций с плавающей точкой в секунду) необходимо 10 ^{7} секунд или более 100 дней. Для того, чтобы произвести расчет за 10 мин., Необходим компьютер производительностью 1.7 Tflops (1.7X10 ^{12} операций с плавающей точкой в секунду).

2. К категории задач, требующих *большого объема оперативной памяти,* относятся, например, задачи гидро- и газодинамики из расчета течений с учетом различных физических и химических процессов. Такие задачи являются, как правило, многомерными, и расчет по каждому из направлений требует оперативной памяти более 10 Гбайт. В квантовой химии неэмпирические расчеты электронной структуры молекул требуют вычислительных затрат, пропорциональных

N^4 или N^5, где N условно характеризует число молекул.

3. требование *обеспечения большого количества передаваемой* характерна для задач гидро- и газодинамики с меняющимися граничными условиями, когда вычислительный алгоритм постоянно требует подведения новой информации, и задач экономической оптимизации, описывающих поведение системы, которая погружена в среду с непрерывно изменяющимися свойствами, от которых зависит состояние системы.

4. проблема *обработки и хранения большого объема информации* характерна для задач астрономии, спектроскопии, биологии, ядерной физики.

3. Модели параллельных вычислений. Конвейеризация и параллелизм

конвейеризация - метод, обеспечивающий совокупность различных действий за счет их разбиение на подфункции со смещением во времени выполнением. конвейерный устройство разрабатывают с отдельных ступеней, время срабатывания которых в идеальном случае должен быть одинаковым.

параллелизм - метод, обеспечивающий выполнение различных функций путем их разбиения на подфункции с одновременным выполнением во времени.

пример:

Вариант структурной схемы выполнения выражения $\sum_{i=1}^{N} ba_i^*$ при параллельной обработке приведен на рис.1.1.

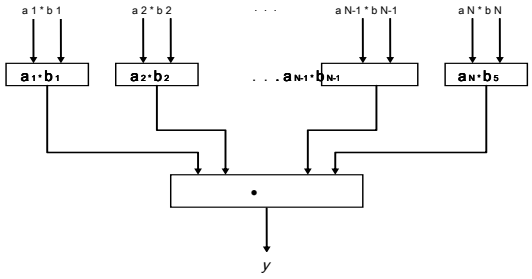

рис.1.1

4. Средства для проведения параллельных вычислений

1. Аппаратные средства:
   - средства для проведения вычислений (вычислительная техника):
     - вычислительная техника, собранная из стандартных комплектующих;
     - вычислительная техника, собранная из специальных комплектующих;
   - средства визуализации;
   - средства для хранения и обработки данных.

2. Программные средства:
   - программные средства общего назначения (операционные системы: стандартные библиотеки, языки программирования, компиляторы, профайлеры, отладчик и т.п.);
   - специальные программные средства: библиотеки (PVM , MPI) средства объединения ресурсов (Dynamite , Globus и др.)

5. Уровни распараллеливания

Классификация параллельности по уровням, отличающиеся показателями абстрактности распараллеливания задач приведена в табл.1.1. Чем "ниже" уровень параллельности, тем более детальным, малоелементнишим будет распараллеливания, что касается элементов программы (инструкция, элементы инструкции и т.д.).

Таблица 1.1 Уровни параллельности

зернистость	уровне	объект обработки	пример системы
крупноблочных	программный	Работа / Задача	мультизадачного ОС
	процедурный	процесс	MIMD-система
Дрибноблокова	уровень формул	инструкция	SIMD-система
	Бит-уровень	В рамках инструкции	Машина фон Ноймана

Методы і конструктивы данного уровня ограничиваются только этим уровнем і не могут быть распространены на другие уровни.

программный уровень

На этом уровне одновременно (или по меньшей мере с временным разделением) выполняются комплексные программы (рис.1.2). Компьютер, выполняющий эти программы *не обязательно должен иметь параллельную структуру,* достаточно, чтобы на нем была установлена многозадачная операционная система (например, реализована как система *распределения времени).* В этой системе каждому пользователю, согласно его приоритета, планировщик (*Scheduler}* выделяет отрезок времени разной длительности. Пользователь получает ресурсы центрального процессорного блока только в течение короткого времени, а затем становится в очередь на обслуживание.

В том случае, когда в системе недостаточное количество процессоров для всех пользователей (или процессов), что, как правило, наиболее вероятно, в системе моделируется параллельное обслуживание пользователей с помощью "квазипаралельних" процессов.

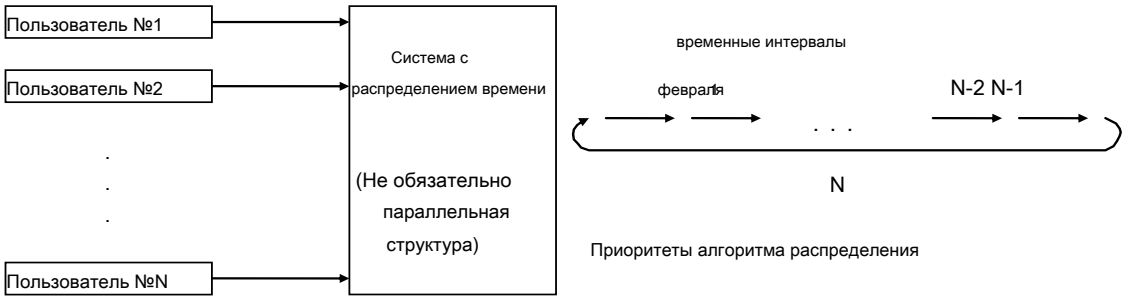

Рис.1.2. Параллельность на программном уровне

уровень процедур

На этом уровне различные части ("процессы") одной и той же программы выполняются параллельно; количество операций обмена данными между процессами должна быть минмальною.

Основное применение процедурного уровня распараллеливания - общее параллельное обработки информации, где применяется деление решаемой проблемы на параллельные задачи - части, которые решаются многими процессорами с целью повышения вычислительной производительности (пример - рис.1.3.).

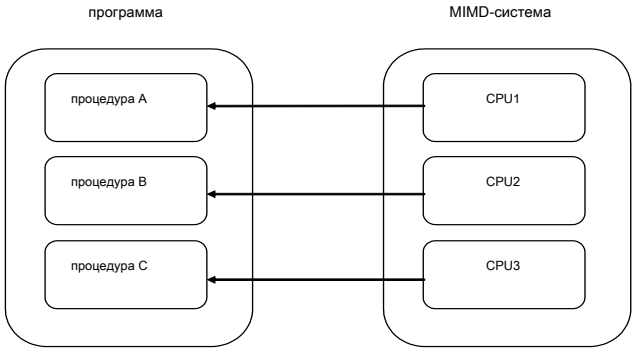

Рис.1.3. Параллельность на уровне процедур

Уровень формул (арифметических выражений)

Арифметические выражения выполняются параллельно покомпонентно, причем в значительно более простых синхронных методах. Если, например, речь идет о добавлении матриц (рис.1.4.), То операция синхронно распараллеливается просто потому, что на каждом процессору исчисляется один (или несколько) элемент матрицы.

при применении *n * n* процессорных элементов можно получить сумму двух матриц размерностью *n * n* за время выполнения одной операции сложения (за исключением времени, необходимого на выполнение операций чтения и записи данных). Этому уровню присущи средства векторизации так называемой *параллельности данных.* Почти каждому элементу данных здесь подчиняется свой процессор, благодаря чему те данные, что в машине фон Ноймана были пассивными. превращаются в «активные вычислительные устройства".

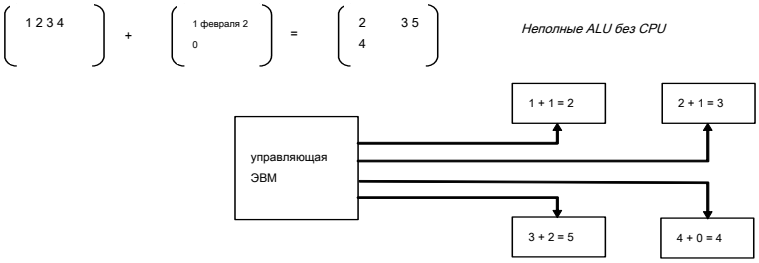

Рис.1.4. Параллельность на уровне операций (формул)

Бит - уровень

На этом рівни происходит параллельное выполнение битовых операций в пределах одного информационного слова (рис.1.5). Параллельность на уровне битов можно найти в любом работающем процессоре. Например, в 8-разрядном арифметико-логическом устройстве побитовая обработка выполняется параллельными аппаратными средствами.

```
          01011101
AND       11011000
          01011000
```

Рис.1.5.Паралельнисть на уровне битов

6. Параллельные операции

Другой вид параллельности возникает из анализа математических операций над отдельными элементами данных или над группами данных. различают *скалярные данные,* операции над которыми выполняются последовательно, і *векторные данные,* операции над которыми выполняются параллельно.

Простые операции над векторами, например сложение двух векторов, могут выполняться синхронно і параллельно. В этом случае каждому элементу вектора подчиняется один процессор. При сложных операциях, таких как формирование частных сумм, построение

эффективного параллельного алгоритма является сложной. Различают одномисцеви (монадной) и двомисцеви (диадного) операции, характерные примеры для которых приведены ниже.

Одномисцеви операции

а). Скаляр -> скаляр — последовательное выполнение

пример 9-> 3 — *"Корень"*

б). Скаляр -> вектор — Розмноження числової величини

Приклад 9->(9,9,9,9) — *"Broadcast"*

в). Вектор -> скаляр — Редукція вектора в скаляр

Приклад (1,2,3,4)->10 — *"Складання"*

(із припущенням, що довжина вектора не змінюється)

г-1) Локальна векторна покомпонентна операція.

Приклад : (1,4,9,16) -> (1,2,3,4) — *"Корінь"*

г-2) Глобальна векторна операція з перестановками.

Приклад: (1,2,3,4) -> (2,4,3,1) — *"Заміна місць компонент вектора"*

г-3) Глобальна векторна операція (часто складається з простих операцій)

Приклад: (1,2,3,4) -> (1,3,6,10) — *"Часткові суми"*

Двомісцеві операції

д) (скаляр, скаляр)-> вектор — Послідовне виконання

Приклад: (1,2)->3 — *"Скалярне додавання"*

е) (скаляр, вектор)->вектор — Покомпонентне застосування операцій над скаляром і вектором

Приклад: (3,(1,2,3,4))->(4,5,6,7) — *"Додавання скаляра з вектором"*

Операція виконується як послідовність операцій б) та є) додавання векторів:

(3,3,3,3)
+
(1,2,3,4)
(4,5,6,7)

є) (вектор, вектор) -> вектор — Покомпонентне застосування операції над двома векторами

Приклад: ((1,2,3,4),(0,1,,3,2)) -> (1,3,6,6) — *"Додавання векторів"*

7. Основні принципи паралелізму (розпаралелення)

Розпаралелити кожну задачу можна не єдиним способом. Тому доцільно розглянути алгоритми можливого розпаралелення типових задач незалежно від конкретної програмної і платформенної реалізації. В загальному випадку процедура розпаралелення розіб'ється на 3 етапи:

- розбиття задачі на незалежні під задачі;
- призначення конкретних процесорів для виконання кожної під задачі;
- збирання результатів роботи окремих процесорів.

8. Классификация структур параллельной обработки

Есть разные системы классификации параллельных систем, базирующихся на разнотипных главных признаках классификации. Многие из них базируется на классификации М. Флинна (1966 г.), Где параллельная обработка выполняется на SIMD и MIMD архитектурах.

В современных классификационных системах архитектуры, которые попадают в один класс, отличаются по количеству процессоров, природе и топологии связи между ними, по способу организации памяти, по технологии программирования и другим признакам.

К примеру, *А.Базу (*A.Basu) считает, что любую параллельную вычислительную систему можно однозначно описать последовательностью решений, принятых на этапе ее проектирования, а сам процесс проектирования представить в виде дерева. Классификация по Базу приведена на рис.1.6.

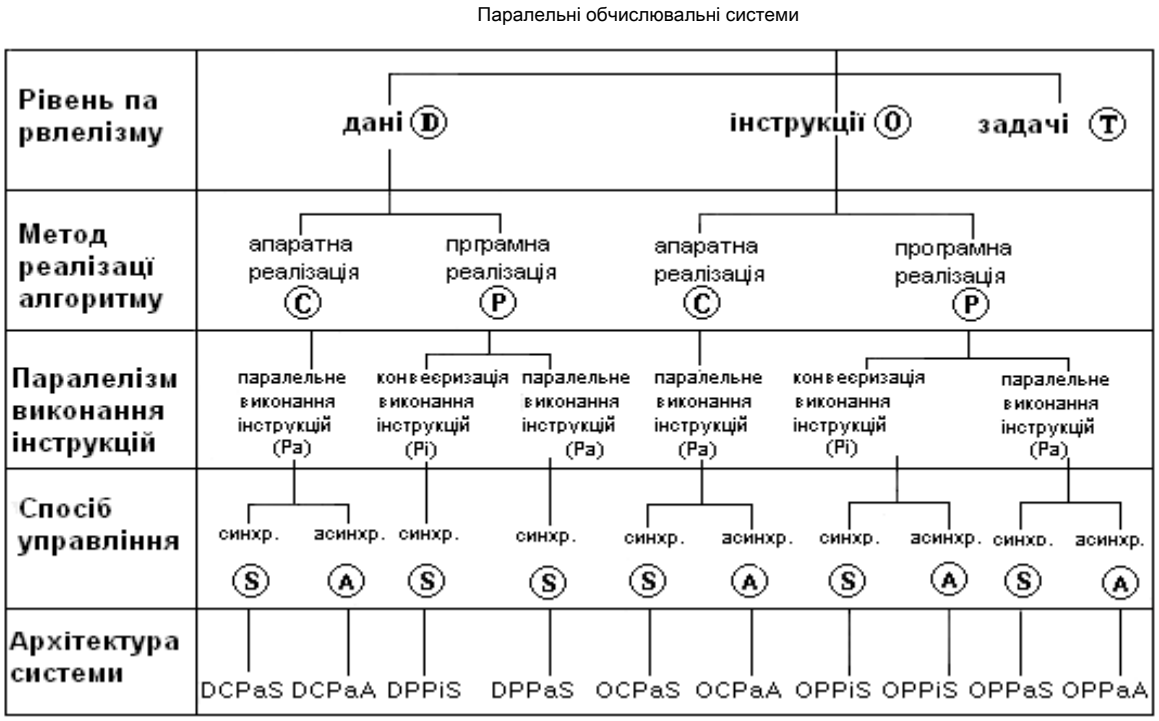

Рис.1.6. Классификация по Базу

Р.Дункан визначає такий набір вимог на який може спиратися класифікація (див.рис.1.7): З класу паралельних машин повинні бути виключені ті, у яких паралелізм закладений лише на найнижчому рівні, включаючи:

- конвеєризацію на етапі підготовки і виконання команди (instruction pipelining), тобто часткове перекриття таких етапів, як дешифрація команди, обчислення адрес операндів, вибірка операндів, виконання команди і збереження результату;
- наявність в архітектурі декількох функціональних пристроїв, що працюють незалежно, зокрема, можливість паралельного виконання логічних і арифметичних операцій;
- наявність окремих процесорів вводу/виводу, що працюють незалежно і паралельно з основними процесорами.

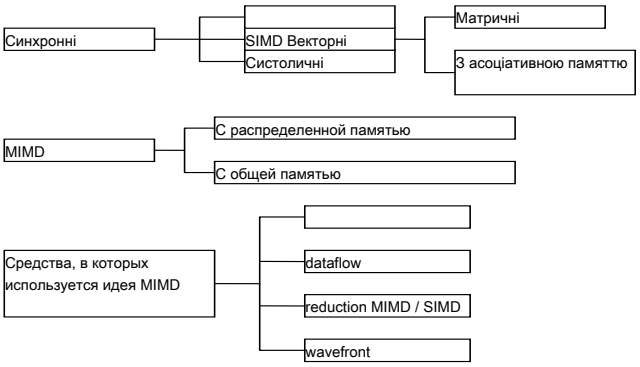

Рис.1.7 Требования к классификации по Р. Дунканом

Дункан дает неформальное определение параллельной архитектуры, а именно: *параллельная архитектура* - это такой способ организации вычислительной системы, при котором допускается, чтобы множество процессоров (простых или сложных) может работать одновременно, взаимодействуя при необходимости друг с другом.

Рассмотрим три вида архитектур, в которых используют нетрадиционные модели вычислений.

Dataflow - используют модель, в которой команда может выполнятся сразу же, как только вычисленные необходимые операнды. Таким образом, последовательность выполнения команд определяется зависимостью по данным, может быть выражена, например, в форме графа.

Модель вычислений, применяемая в *reduction* машинах заключается в следующем: команда становится доступной для выполнения тогда и только тогда, когда результат ее работы нужно другой, доступной для выполнения, команде в качестве операнда.

Wavefront array архитектура сочетает в себе идею систолической обработки данных и модель вычислений, используемый в *dataflow.* В данной архитектуре процессоры объединяются в модули и фиксируются связи, по которым процессоры могут взаимодействовать друг с другом. Однако, в противовес ритмичной работе систолического массивов, данная архитектура использует асинхронный механизм связи с подтверждением (handshaking), из-за чего "фронт волны" вычислений может менять свою форму по мере прохождения по всем процессорах.

Е.Кришнамарфи для классификации параллельных вычислительных систем предлагает использовать четыре характеристики, подобные характеристик классификации А.Базу:
степень гранулярности, способ реализации параллелизма, топология и природа связи, способ управления процессорами.

На основе выделенных четырех характеристик можно определить наиболее известных классов архитектур в данной классификации (см. Табл.1.2).

Таблица 1.2

Тип архитектуры Гранулярность		Реализация параллелизма	Связь процессоров Способ	управления
Векторноконвеерни компьютеры	На уровне данных	аппаратная	Простая топология со средней связностью	синхронный
классические мультипроцессоры	На уровне задач	комбинированная	Простая топология со слабой связностью	асинхронный
матрицы процессоров	На уровне данных	аппаратная	Двумерные массивы с сильной связностью	синхронный
систолические массивы	На уровне данных	аппаратная	Сложная топология с сильной связностью	

Несмотря на то, что классификация Е. Кришнамарфи построена только на четырех признаках, она позволяет выделить и описать такие "нетрадиционные" параллельные системы, как Систолическое массивы, машины типа dataflow и wavefront.

Заслуживают внимания классификации *Хандлер (ерлангерська схема) [*8], *Р. Хокни [*1].

Упражнения и задания к теме №1

1. Как соотносятся между собой классификации *хендлера* и *Флинна?*

2. Какие параметры архитектуры параллельного компьютера надо знать пользователю для создания эффективных программ? Предложите классификацию компьютеров, опирающаяся на выделенные параметры.

3. Можно ли организовать параллельные вычисления для решения задачи нахождения наименьшего числа из трех чисел ?, из четырех чисел? Если да, то каким образом?

4. Приведите пример применения параллельной обработки для любой области науки и техники. Какие из требований параллельной обработки там используются?

5. Есть ли принтер средством для организации параллельной обработки?

6. Необходимо перемножить матрицу A ($n_1 * n_2$) на матрицу B ($n_2 * n_3$). Разработайте функциональную схему устройства, обеспечит быстрее выполнения данной операции.

7. Почему в конвейерных устройствах продолжительность срабатываний отдельных ступеней делают одинаковыми?

8. Один рабочий может вырыть яму размером 1x1x1 $м^3$ за час. За сколько времени бригада из 5, 10, 20 рабочих выроют яму размером 2x2 $м^2$ и глубиной 1м, если производительность всех рабочих одинакова?

9. Постройте график зависимости выполнения работы от количества рабочих в бригаде.