Белорусский государственный университет
информатики и радиоэлектроники”

Факультет информационных технологий и управления

Кафедра интеллектуальных информационных технологий

**Отчёт по лабораторной работе №2 по курсу «МРЗвИС» на тему:
«Реализация модели решения задачи на ОКМД архитектуре»**

Выполнил
студент группы 821703: Щур А. А.

Проверила: Орлова А.С

Минск 2020

Постановка задачи

Реализовать и исследовать модель решения на ОКМД архитектуре задачи вычисления матрицы значений, вычисляемой по следующим правилам:

$$c_{ij} = \underset{k}{\tilde{\wedge}} f_{ijk} * (3 * g_{ij} - 2) * g_{ij} + \left(\underset{k}{\tilde{\vee}} d_{ijk} + \left(4 * \left(\underset{k}{\tilde{\wedge}} f_{ijk} \,\tilde{\circ}\, \underset{k}{\tilde{\vee}} d_{ijk}\right) - 3 * \underset{k}{\tilde{\vee}} d_{ijk}\right) * g_{ij}\right) * (1 - g_{ij})$$

$$f_{ijk} = (a_{ik} \tilde{\to} b_{kj}) * (2 * e_k - 1) * e_k + (b_{kj} \tilde{\to} a_{ik}) * \left(1 + \left(4 * (a_{ik} \tilde{\to} b_{kj}) - 2\right) * e_k\right) * (1 - e_k)$$

$$d_{ijk} = a_{ik} \tilde{\wedge} b_{kj}$$

Где:

$$\underset{k}{\tilde{\wedge}} f_{ijk} = \prod_k f_{ijk}$$

$$\underset{k}{\tilde{\vee}} d_{ijk} = 1 - \prod_k (1 - d_{ijk})$$

$$\underset{k}{\tilde{\wedge}} f_{ijk} \,\tilde{\circ}\, \underset{k}{\tilde{\vee}} d_{ijk} = \max\left(\left\{\underset{k}{\tilde{\wedge}} f_{ijk} + \underset{k}{\tilde{\vee}} d_{ijk} - 1\right\} \cup \{0\}\right)$$

$$a_{ik} \tilde{\to} b_{kj} = \max\left(\{1 - a_{ik}\} \cup \{b_{kj}\}\right)$$

$$b_{kj} \tilde{\to} a_{ik} = \max\left(\{1 - b_{kj}\} \cup \{a_{ik}\}\right)$$

$$a_{ik} \tilde{\wedge} b_{kj} = \min\left(\{a_{ik}\} \cup \{b_{kj}\}\right)$$

Описание модели

Для реализации поставленной задачи была разработана программа, производящая вычисление по заданным правилам, имеющая возможность параметрического задания времени счёта (длины) операций различных типов.

T1 – время выполнения программы на одном процессорном элементе. Вычисляется путём подсчёта количества вызовов той или иной операции, а затем получение значение умножается на время данной операции. Данное действие повторяется для всех операций и в конце все значения суммируются. **Tn** – время выполнения программы на n-количестве процессорных элементов. необходимо установить зависимости между выполняемыми операциями. Вычисляется схожим путём, что и **T1**, за исключением поиска операций, которые можно считать на различных процессорах. Время выполнения такой операции считается следующим образом, а именно находится количество вызовов данной операции и делится на количество процессорных элементов. **Ky** – коэффициент ускорения равен **T1/Tn**. e – эффективность равна **Ky/n**. **D** - коэффициента расхождения программы, **D = Lsum/Lavg. Lsum** - суммарная длина программы и равна **Tn**. **Lavg** - средняя длина программы. Вычисляется путём подсчета количества вызовов операций на различных ветвях выполнения программы. Имея, количества вызовов операций, выполняющихся на ветвях программы, и их время выполнения, считаем данную величину.

Исходные данные:

1. p, m, q – размерность матриц;
2. n – количество процессорных элементов в системе;
3. ti – время(длина) выполнения операции над элементами матриц.
4. Матрицы A, B, E, G заполненные случайными числами в диапазоне [-1;1].

1.Вопросы:

1. Проверить, что модель создана верно: программа работает правильно

Исходные данные:

```
[ -0.959 -0.533  ]
[ -0.666 -0.5   ]

[ 0.169 0.724   ]
[ 0.478 0.358   ]

[ -0.038-0.536 ]

[ 0.705 -0.855  ]
[ 0.281 -0.173  ]
```

2. Результат:

```
[ 2.84073 -33.008  ]
[ 1.99371 -4.77547  ]
```

3. Проверка:

$$d_{00k} = \min(\{a_{0k}\} \cup \{b_{k0}\})$$

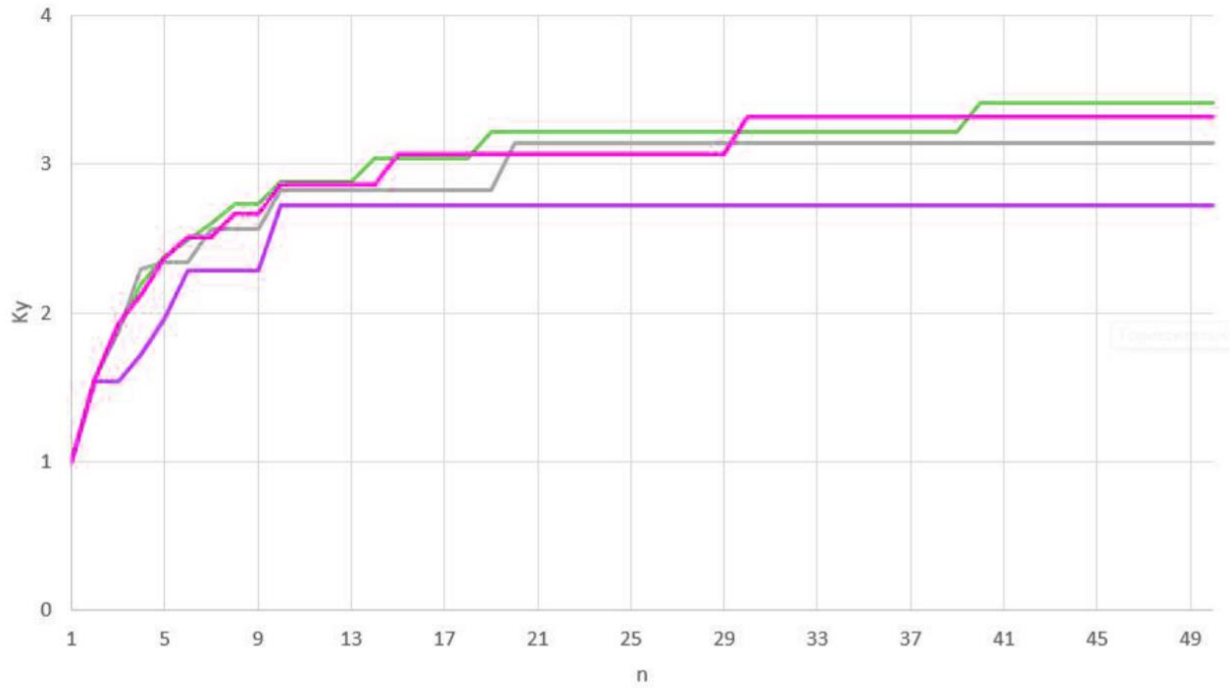
$$f_{00k} = \max(\{1 - a_{0k}\} \cup \{b_{k0}\})) * (2 * e_k - 1) * e_k + \max(\{1 - b_{k0}\} \cup \{a_{0k}\}) * (1 + \left(4 * max\left(\{1 - a_{0k}\} \cup \{b_{0j}\}\right) - 2\right) * e_k) * (1 - e_k)$$

$$C_{00} = \prod_k f_{ijk} * (3 * 0.705 - 2) * 0.705 + \left(\left(1 - \prod_k (1 - d_{ijk})\right) + \left(4 * \left(max\left(\left\{\prod_k f_{ijk} + \left(1 - \prod_k (1 - d_{ijk})\right)\right\} \cup \{0\}\right)\right) - 3 * \left(1 - \prod_k (1 - d_{ijk})\right)\right) * 0.705\right) * (1 - 0.705) = 2.840$$

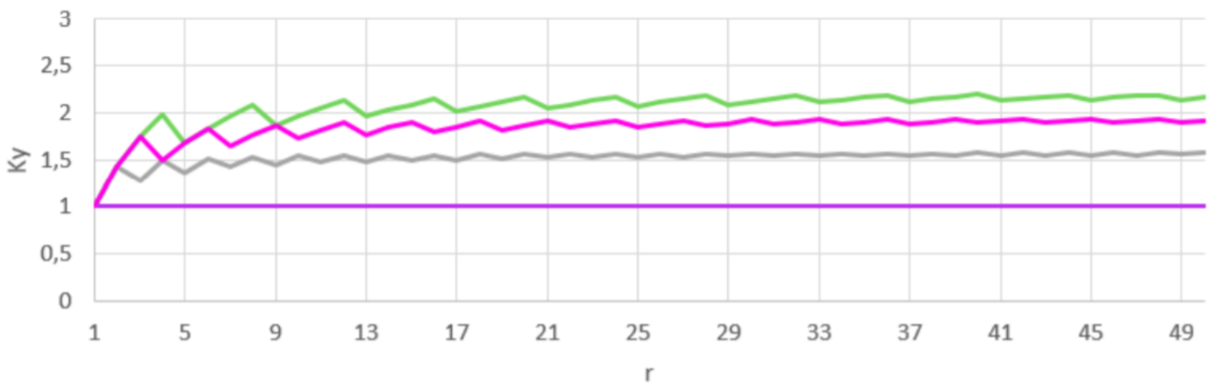
4. Ответ:

Модель создана верно

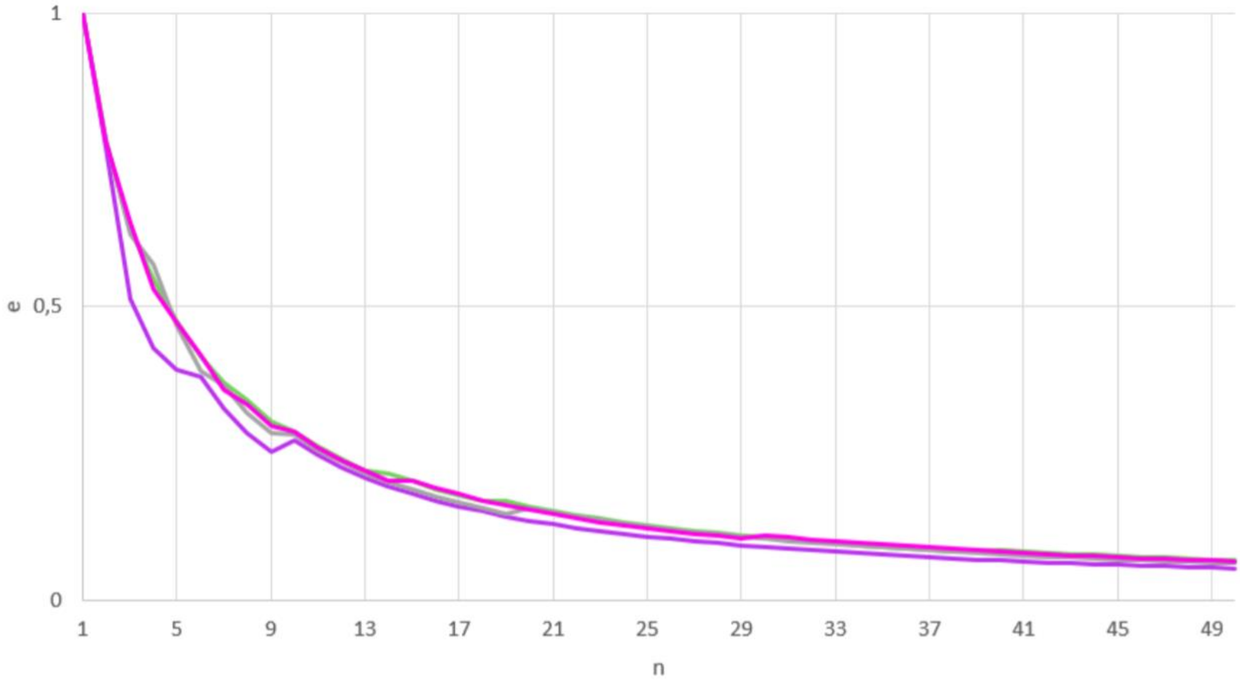
2. Построить графики и объяснить на них точки перегиба и асимптоты.



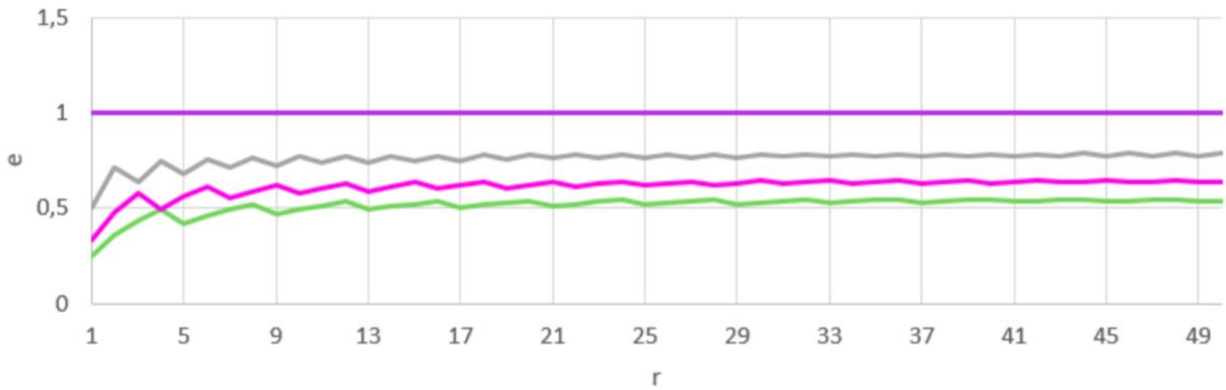
Асимптотой графика является прямая. Данная прямая параллельна оси абсцисс, ордината всех точек этой прямой равна значению коэффициента ускорения при n = r. Связано это с тем, что как только количество процессорных элементов становится больше ранга задачи, в вычислениях участвуют только r процессорных элементов, остальные никак не используются.



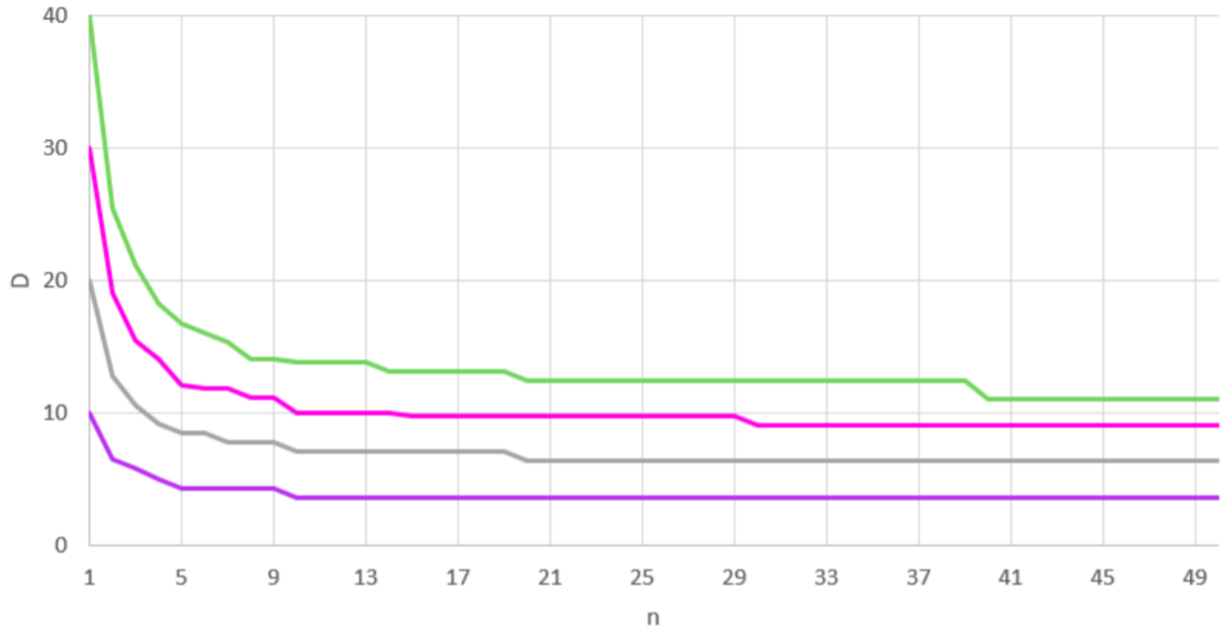
Асимптотой графика является прямая, параллельная оси абсцисс, а ордината всех точек этой прямой равна значению коэффициента ускорения при n = r. Точками перегиба являются те точки, в которых r кратно n. Связано это с тем, что при таких значениях r, все процессорные элементы одновременно задействованы в вычислениях.



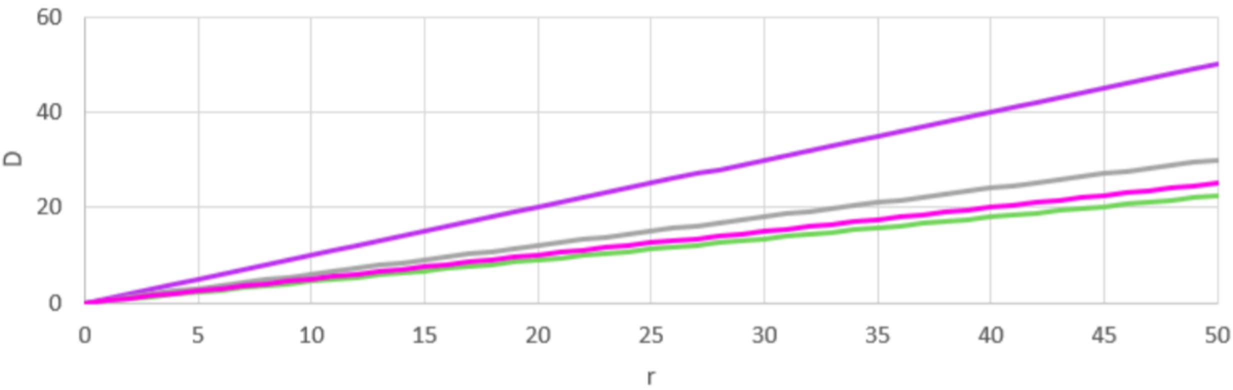
Асимптотой графика является прямая e = 0. Связано это с тем, что как только n становится равным r, рост коэффициента ускорения прекращается, а n продолжает увеличиваться.



Асимптотой графика является прямая e = 1. Точками перегиба являются те точки, в которых r кратно n. Связано это с тем, что при таких значениях r, все процессорные элементы одновременно задействованы в вычислениях.



Асимптотой графика является прямая, параллельная оси абсцисс, а ордината всех точек этой прямой равна значению коэффициенту расхождения программы при n = r. Связано это с тем, что как только количество процессорных элементов становится больше ранга задачи, в вычислениях участвуют только r процессорных элементов, остальные никак не используются.



Асимптотой графика является функция D=k*r+b. При n=1: k=1 b=0, при n=2: k=0.6 b=1, при n=3: k=0.5 b=1, при n=4: k=0.45 b=0.5.

3. Спрогнозировать, как изменится вид графиков при изменении параметров модели. Если модель позволяет, то проверить на ней правильность ответа.

1) Увеличивая n, Ку(n) увеличивается. Рост значения Ку(n) наблюдается до тех пор, пока количество процессорных элементов не становится равным рангу задачи. После этого коэффициент ускорения не изменяется. Увеличивая r, Ку(r) увеличивается скачкообразно.

2) Увеличивая n, e(n) уменьшается. Увеличивая r, e(r) растёт скачкообразно.

3) Увеличивая n, D(n) уменьшается. Падение значения D(n) наблюдается до тех пор, пока количество процессорных элементов не становится равным рангу задачи. После этого коэффициент расхождения программы не изменяется. Увеличивая r, D(r) растёт.

Вывод

В результате выполнения лабораторной работы была реализована модель вычисления матрицы значений на ОКМД архитектуре. Данная модель была проверена на работоспособность и правильность получаемых результатов. С помощью графиков, построенных в результате выполнения лабораторной работы, были изучены зависимости коэффициента ускорения, эффективности и коэффициента расхождения программы от количества процессорных элементов и ранга задачи.