

Министерство образования Республики Беларусь

**Учреждение образования
“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»*

Выполнили
студенты группы
821701

Залесский А. А.
Киселёв Н. В.

Проверил

Крачковский Д. Я.

Минск 2020

Цель:

Реализовать и исследовать модель решения на ОКМД архитектуре задачи вычисления матрицы значений.

Постановка задачи:

Дано: сгенерированные матрицы A, B, E, G , заданных размерностей pxm, mxq, lxm, pxq, mxx и qxm соответственно со значениями в диапазоне $[-1;1]$.

$$c_{ij} = \bigwedge_k f_{ijk} * (3 * g_{ij} - 2) * g_{ij} + (\bigvee_k d_{ijk} + (4 * (\bigwedge_k f_{ijk} \circ \bigvee_k d_{ijk}) - 3 * \bigvee_k d_{ijk}) * g_{ij}) * (1 - g_{ij})$$
$$f_{ijk} = (a_{ik} \rightsquigarrow b_{kj}) * (2 * e_k - 1) * e_k + (b_{kj} \rightsquigarrow a_{ik}) * (1 + (4 * (a_{ik} \rightsquigarrow b_{kj}) - 2) * e_k) * (1 - e_k)$$
$$d_{ijk} = a_{ik} \wedge b_{kj}$$

Вариант индивидуального задания:

$$17. \bigwedge_k f_{ijk} = \prod_k f_{ijk}$$
$$\bigvee_k d_{ijk} = 1 - \prod_k (1 - d_{ijk})$$
$$\bigwedge_k f_{ijk} \circ \bigvee_k d_{ijk} = \max \left(\left\{ \bigwedge_k f_{ijk} + \bigvee_k d_{ijk} - 1 \right\} \cup \{0\} \right)$$
$$a_{ik} \rightsquigarrow b_{kj} = \min \left(\left\{ 1 - a_{ik} + b_{kj} \right\} \cup \{0\} \right)$$
$$b_{kj} \rightsquigarrow a_{ik} = \min \left(\left\{ 1 - b_{kj} + a_{ik} \right\} \cup \{0\} \right)$$
$$a_{ik} \wedge b_{kj} = \max \left(\left\{ a_{ik} + b_{kj} - 1 \right\} \cup \{0\} \right)$$

Получить: C – матрицу значений соответствующей размерности pxq ; в случае необходимости доопределить всеобщности(\forall) или существования(\exists) условие исходной задачи кванторами самостоятельно.

Описание модели:

Была реализована модель решения на ОКМД архитектуре задачи вычисления матрицы значений. Возможность самостоятельно устанавливать все параметры, необходимые для работы модели, позволяет детально исследовать разработанную модель, установить зависимости между вышеуказанными параметрами.

- T_1 – время выполнения программы на одном процессорном элементе. Данный параметр вычисляется следующим образом: подсчитывается количество вызовов той или иной операции, а затем полученное значение умножается на время данной операции. Данное действие повторяется для всех операций, в итоге все значения суммируются.

- T_n – время выполнения программы на n -количестве процессорных элементов. Параметр вычисляется схожим путём, что и T_1 : осуществляется поиск операций, которые можно считать на различных процессорах. Для подсчета времени на выполнение такой операции находится количество вызовов данной операции и делится на количество процессорных элементов.
- K_y - коэффициент ускорения равен $\frac{T_1}{T_n}$.
- e - эффективность равна $\frac{K_y}{n}$.
- D - коэффициент расхождения программы, $D = \frac{L_\Sigma}{L_{cp}}$. Где, L_Σ - суммарная длина программы и равна T_n . L_{cp} - средняя длина программы. Вычисляется путем 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:

- p, m, q – размерность матриц;
- n – количество процессорных элементов в системе;
- t_i – время выполнения i операции над элементами матриц;
- матрицы A, B, E, G , заполненные случайными вещественными числами в диапазоне $[-1;1]$.

Результаты счёта и времена их получения:

```

Input m,p,q,n
1 2 3 4
A:
-0.1805
-0.913

B:
-0.2829  0.3291  0.8415

E:
-0.0485

G:
 0.8605 -0.7203 -0.0933
-0.3441 -0.1441  0.9415

C:
          0          0 -0.00538264
          0 -0.0976003 -0.673667

Parameters:
T1= 240
Tn= 84
Ky= 2.85714
e= 0.714286
Lsum= 84
Lavg= 35
D= 2.4

```

Построение графиков:

Обозначения:

$K_y(n, r)$ – коэффициент ускорения;

$e(n, r)$ – эффективность;

$D(n, r)$ – коэффициент расхождения программы;

n – количество процессорных элементов в системе (совпадает с количеством этапов конвейера);

r – ранг задачи (количество объектов, которые в процессе решения задачи могли бы обрабатываться параллельно);

Графики строятся на одном наборе сгенерированных данных, постепенно уменьшая размеры матриц, в масштабе, отражающем характерные особенности соответствующих зависимостей.

Зависимость коэффициента ускорения КУ от количества процессорных элементов n

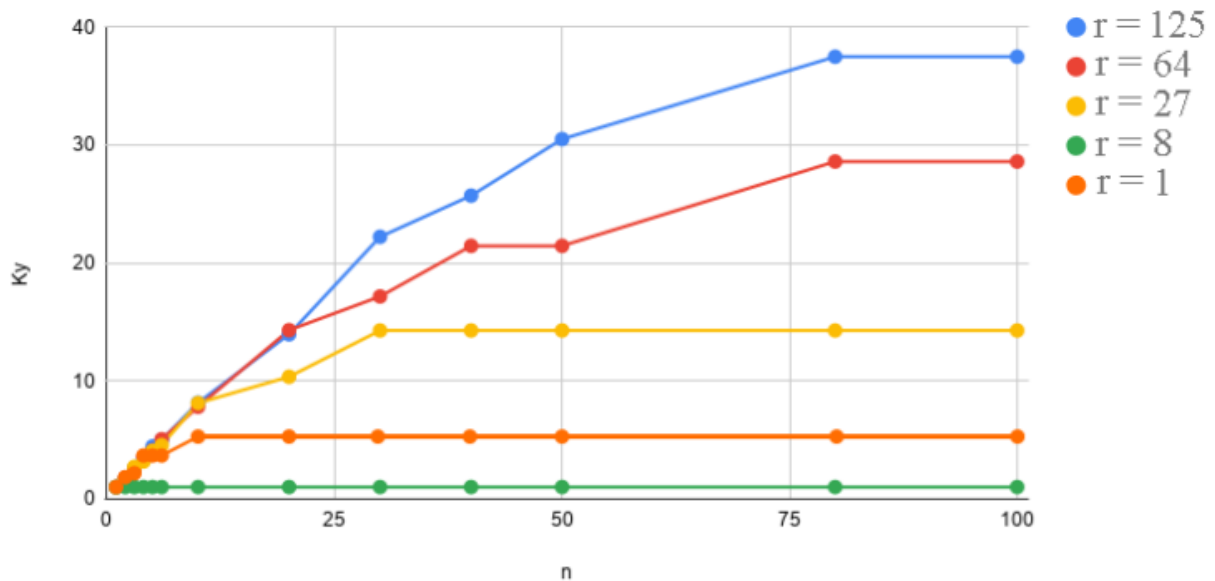


График 1. График зависимости коэффициента ускорения K_y от количества элементов n

Зависимость коэффициента ускорения K_u от ранга задачи r

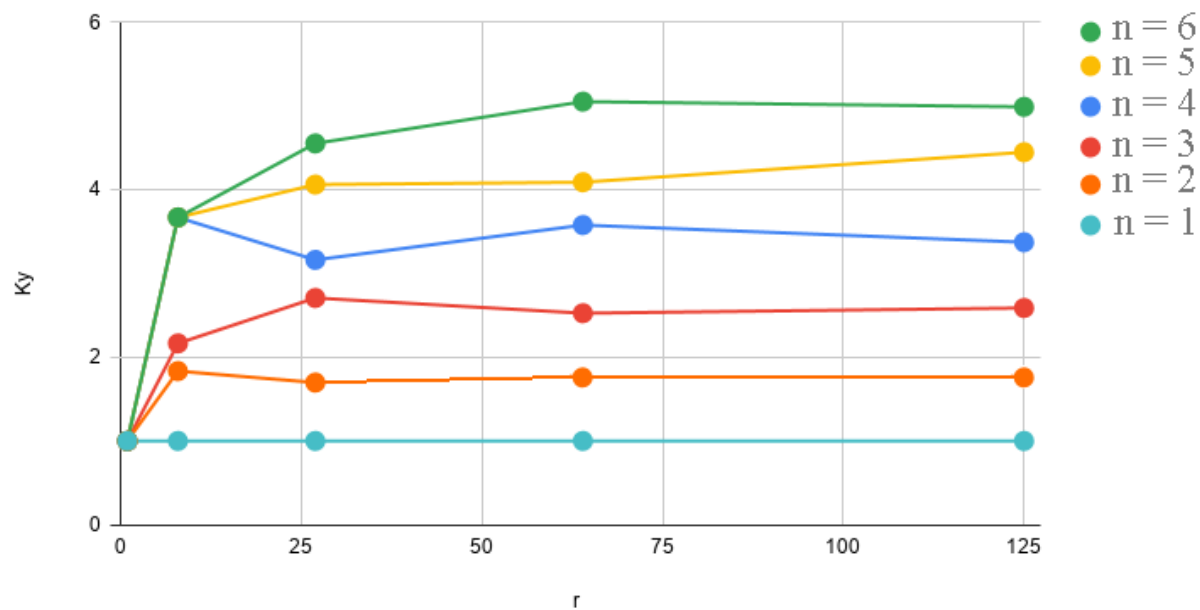


График 2. График зависимости коэффициента ускорения K_u от ранга задачи r

График зависимости эффективности e от количества процессорных элементов n

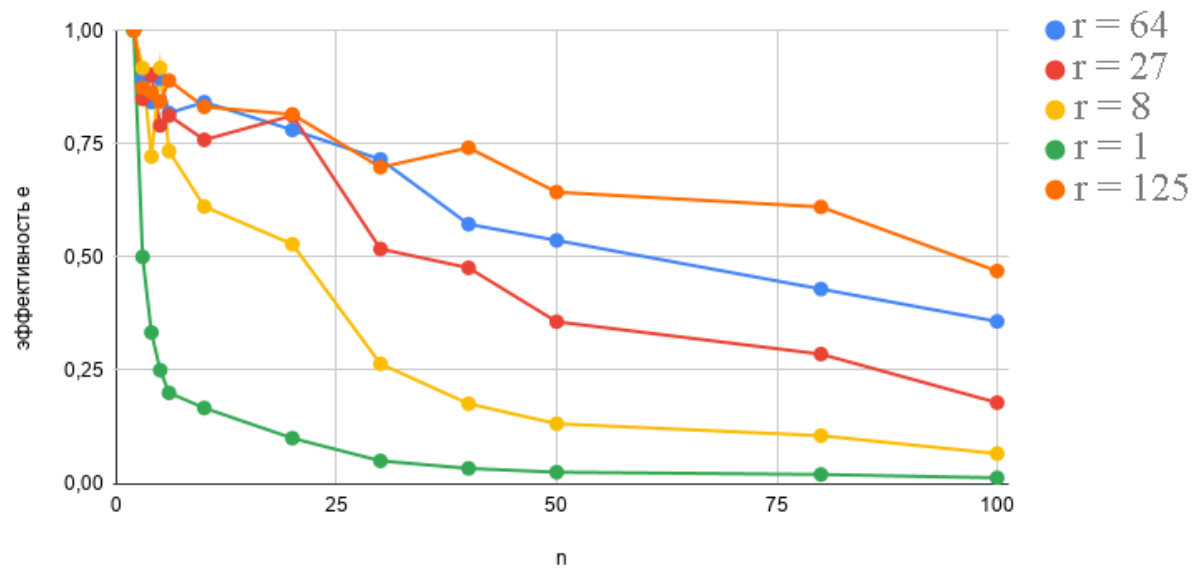


График 3. График зависимости эффективности e от количества элементов n

Зависимость эффективности e от ранга задачи r

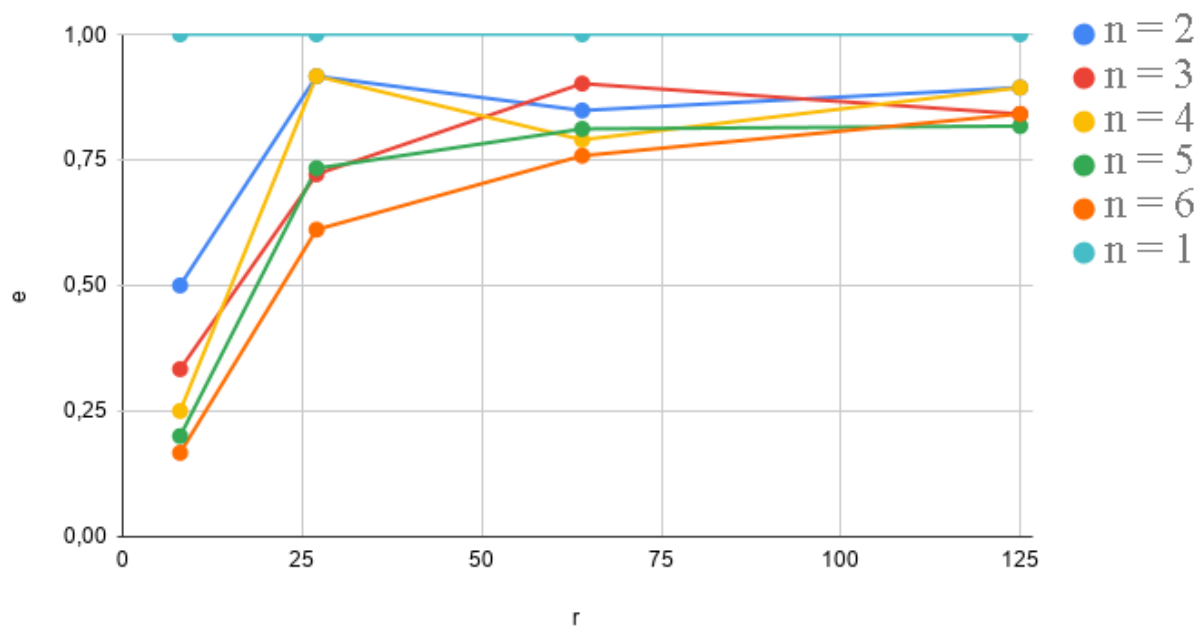


График 4. График зависимости эффективности e от ранга задачи r

Зависимость коэффициента расхождения D от количества процессорных элементов n

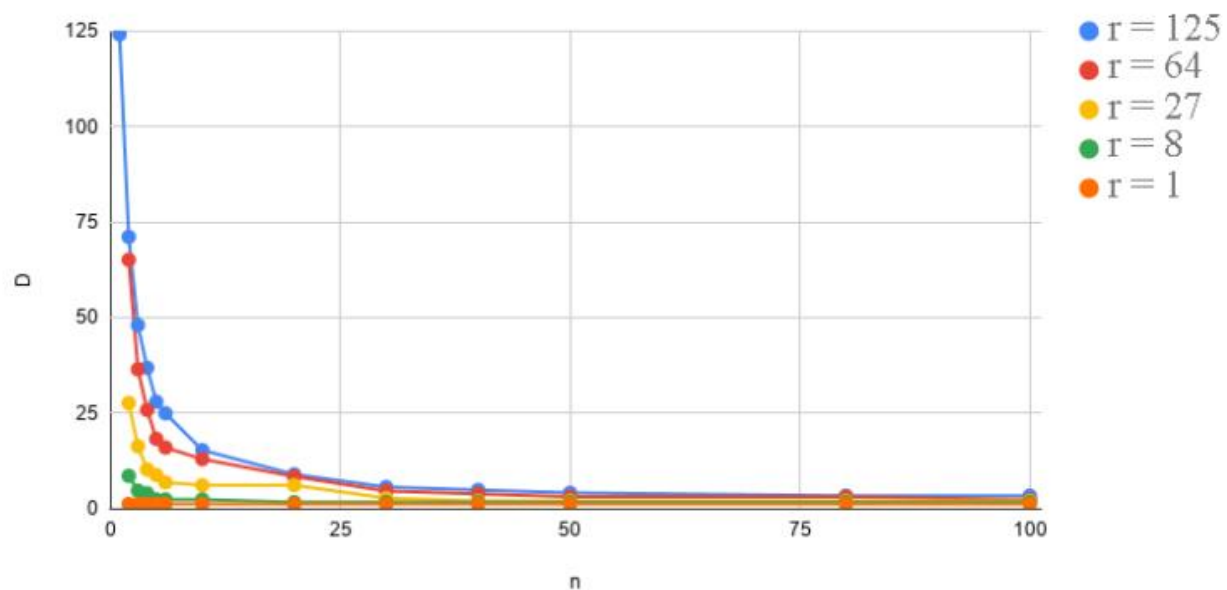


График 5. График зависимости коэффициента расхождения программы D от количества элементов n

Зависимость коэффициента расхождения D от ранга задачи r

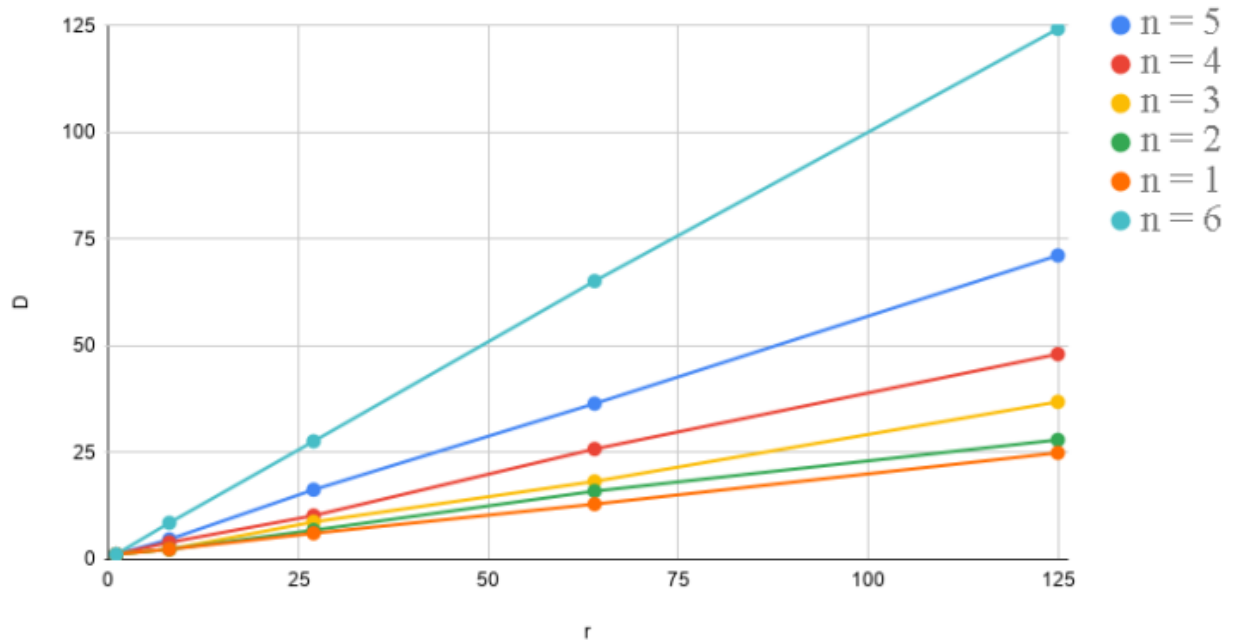


График 6. График зависимости коэффициента расхождения программы D от ранга задачи r

Ответы на вопросы:

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

Проверка правильности работы программы:

| Исходные данные | | | |
|-----------------|---|-----------------------------------|---|
| Время операции | | Другие данные | |
| Сумма | 1 | m | 1 |
| Разность | 1 | p | 2 |
| Произведение | 1 | q | 3 |
| Деление | 1 | количество процессорных элементов | 4 |
| Сравнение | 1 | | |

| | | | | | | | | |
|--|---------|--|--|---------|---------|---------|---------|---------|
| <u><i>A (p x m)</i></u> <table><tr><td>-0,1805</td></tr><tr><td>-0.913</td></tr></table> | -0,1805 | -0.913 | <u><i>B (m x q)</i></u> <table><tr><td>-0.2829</td><td>0.3291</td><td>0.8415</td></tr></table> | -0.2829 | 0.3291 | 0.8415 | | |
| -0,1805 | | | | | | | | |
| -0.913 | | | | | | | | |
| -0.2829 | 0.3291 | 0.8415 | | | | | | |
| <u><i>E (l x m)</i></u> <table><tr><td>-0.0485</td></tr></table> | -0.0485 | <u><i>G (p x q)</i></u> <table><tr><td>0.8605</td><td>-0.7203</td><td>-0.0933</td></tr><tr><td>-0.3441</td><td>-0.1441</td><td>-0.9415</td></tr></table> | 0.8605 | -0.7203 | -0.0933 | -0.3441 | -0.1441 | -0.9415 |
| -0.0485 | | | | | | | | |
| 0.8605 | -0.7203 | -0.0933 | | | | | | |
| -0.3441 | -0.1441 | -0.9415 | | | | | | |

| | | | | | | | | |
|--|------------|-------------|---|---|-------------|---|------------|-----------|
| <i>Полученные данные:</i> | | | | | | | | |
| <u>$C (p \times q)$</u> <table border="1"> <tr> <td>0</td><td>0</td><td>-0.00538264</td></tr> <tr> <td>0</td><td>-0.0976003</td><td>-0.673667</td></tr> </table> | | | 0 | 0 | -0.00538264 | 0 | -0.0976003 | -0.673667 |
| 0 | 0 | -0.00538264 | | | | | | |
| 0 | -0.0976003 | -0.673667 | | | | | | |

Программа работает верно.

2. Объяснить на графиках точки перегиба и асимптоты:

Для графика зависимости коэффициента ускорения (K_y) от количества элементов (n):

Асимптотой графика, исходя из значений графика, является прямая, параллельная оси абсцисс, то есть прямая, заданная при $n = r$. Точки перегиба появляются тогда, когда ширина векторного параллелизма становится кратной числу процессорных элементов, при достижении этого значения коэффициент ускорения перестает расти.

Для графика зависимости коэффициента ускорения (K_y) от ранга задачи (r):

Асимптотой является прямая $K_y = n$, такого значения она достигает в точках, где ширина векторного параллелизма становится кратной числу процессорных элементов. При фиксированном значении процессорных элементов и при устремлении ранга задачи к бесконечности, ОКМД архитектура будет работать быстрее не более, чем в n раз по сравнению с последовательной системой.

Для графика зависимости эффективности (e) от количества элементов (n):

Прямая $e = 0$ будет являться асимптотой. Так как задача с фиксированным рангом содержит фиксированное количество операций, которые необходимо выполнить, а эффективность показывает долю работы одного процессорного элемента, то при большом количестве процессорных элементов эффективность стремится к 0

Для графика зависимости эффективности (e) от ранга задачи (r):

Прямая $e = 1$ будет являться асимптотой, а точками перегиба – точки, где ширина векторного параллелизма становится кратной числу процессорных элементов.

Для графика зависимости коэффициента расхождения программы (D) от количества элементов (n):

При увеличении количества элементов, значение расхождения программы стремится к 1.

Для графика зависимости коэффициента расхождения программы (D) от ранга задачи (r):

При увеличении ранга задачи, значение расхождения программы увеличивается.

3. Спрогнозировать как изменится вид графиков при изменении параметров модели;

если модель позволяет, то проверить на ней правильность ответа;

| Семейства графиков | Изменения вида графика |
|---|--|
| <i>Зависимость коэффициента ускорения (K_y) от количества элементов (n)</i> | При увеличении количества процессорных элементов, возрастет значение коэффициента ускорения. Пиковые значения зафиксированы в точках, где ширина векторного параллелизма становится равной числу процессорных элементов, в этих точках $K_y = n$. |

| | |
|---|--|
| <i>Зависимость коэффициента ускорения (K_y) от ранга задачи (r)</i> | При увеличении ранга, возрастает значение коэффициента ускорения, до момента пока ширина векторного параллелизма не становится равной числу процессорных элементов. Далее при увеличении, коэффициент ускорения остается постоянным. |
| <i>Зависимость эффективности (e) от количества элементов (n)</i> | При увеличении количества процессорных элементов, снижается значение эффективности |
| <i>Зависимость эффективности (e) от ранга задачи (r)</i> | При увеличении ранга, возрастает значение эффективности. Пиковые значения зафиксированы в точках, где ширина векторного параллелизма становится кратной числу процессорных элементов. |
| <i>Зависимость коэффициента расхождения программы (D) от количества элементов (n)</i> | При увеличении количества процессорных элементов, возрастает коэффициент расхождения программы |
| <i>Зависимость коэффициента расхождения программы (D) от ранга задачи (r)</i> | При увеличении ранга задачи, снижается значение коэффициента расхождения программы |

Вывод:

В результате выполнения лабораторной работы была реализована и исследована ОКМД модель для решения задач вычисления матрицы значений. Реализованная модель была проверена на работоспособность и правильность получаемых результатов. Данная модель позволяет ускорить процесс вычисления результата для числовых векторов, по сравнению с последовательной системой. Были исследованы характеристики конвейерной архитектуры: коэффициент ускорения, коэффициент расхождения программы и эффективность.