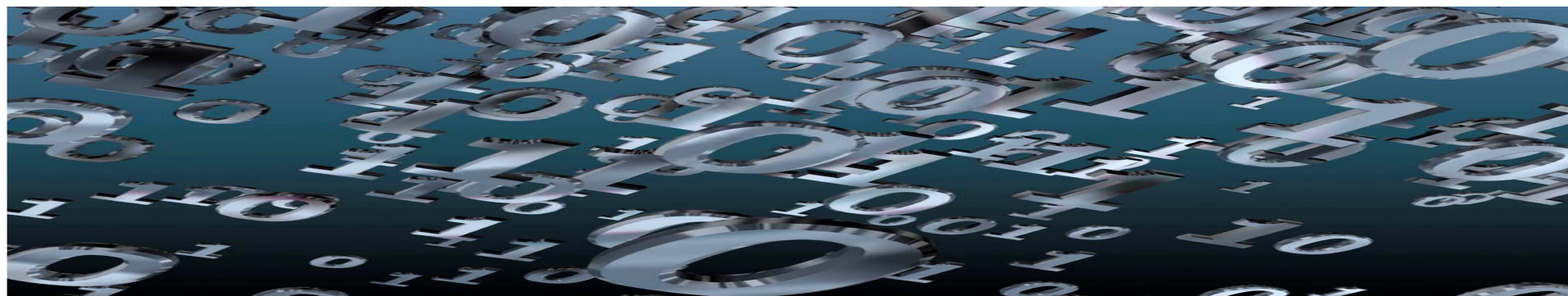


УЛУЧШЕНИЕ ЛОКАЛЬНОСТИ КОМПОЗИЦИИ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ ЯЗЫКА ADA

Л.С. Киркорова, С. И. Киркоров,
Научно-производственное предприятие
МедиаСкан,
Республика Беларусь



Положения, выносимые в работе

1. Улучшение локальности алгоритмов перемножения трёх матриц и алгоритма метода матричной прогонки.
2. Достаточные условия для характеристики локальности данных композиции алгоритмов.

Локализация данных

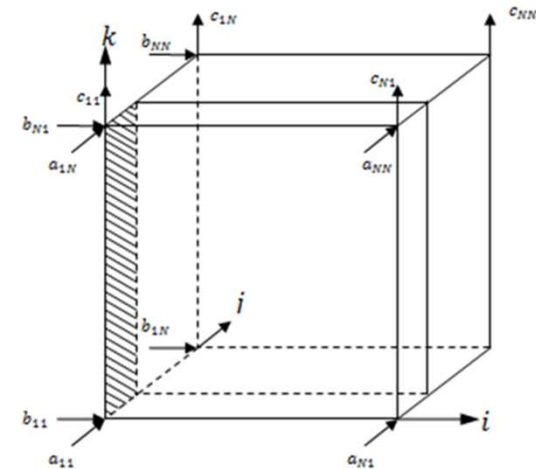
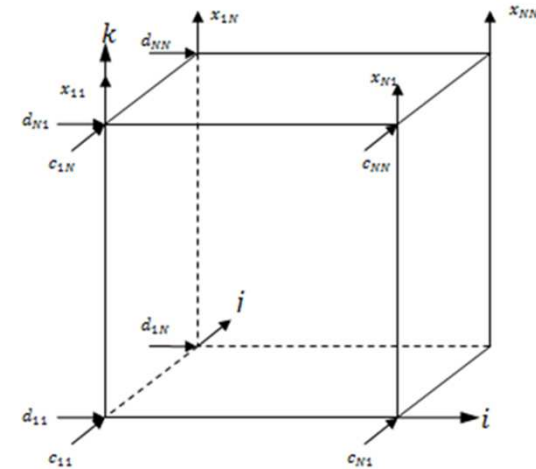
- Локальность – это вычислительное свойство алгоритма, отражающее степень использования памяти с быстрым доступом.
- Локализация данных – это улучшение локальности данных.
- Улучшить локальность означает реорганизовать вычисления таким образом, чтобы добиться более частого использования к памяти с быстрым доступом.

Задача работы

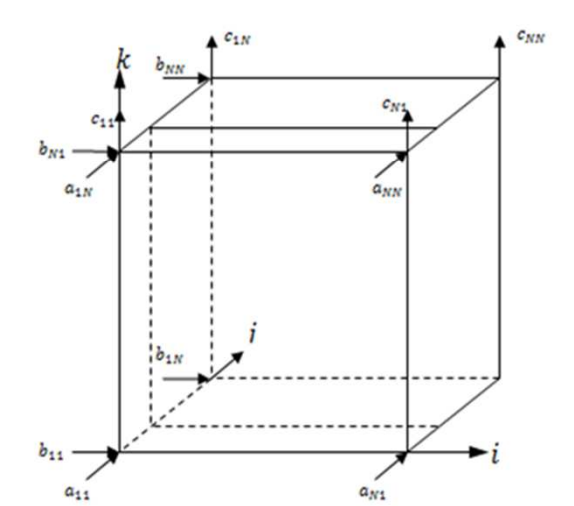
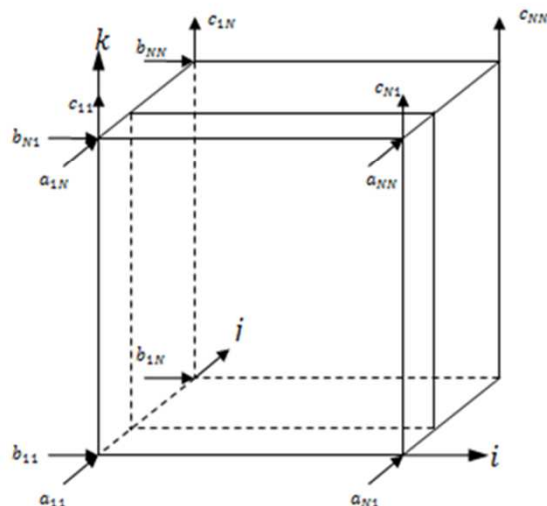
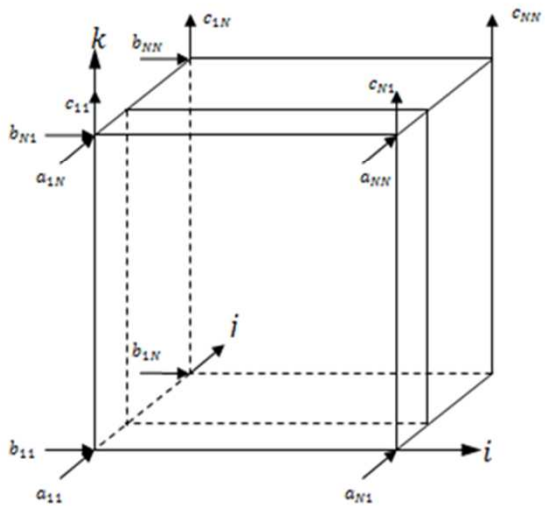
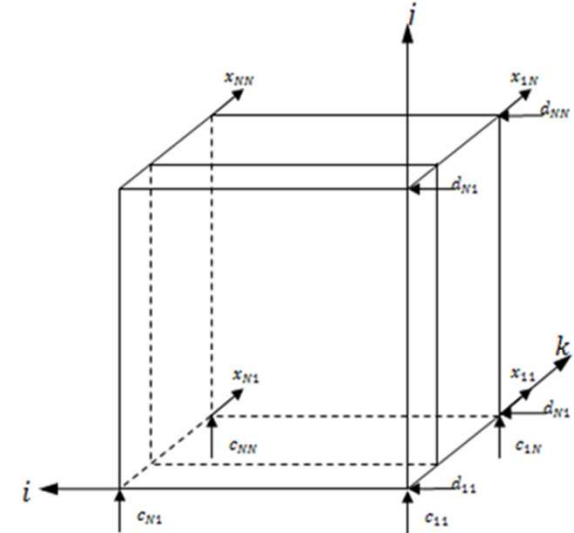
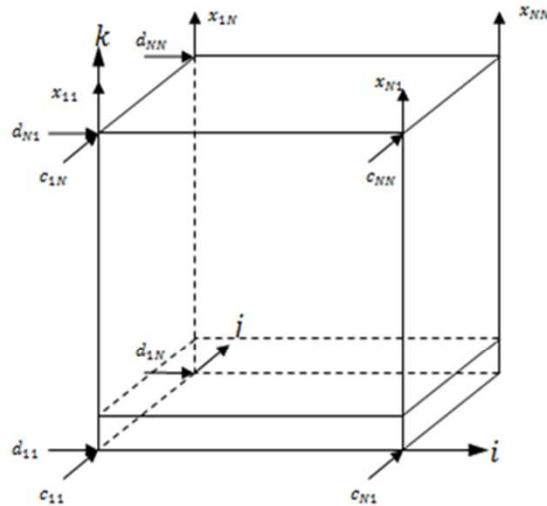
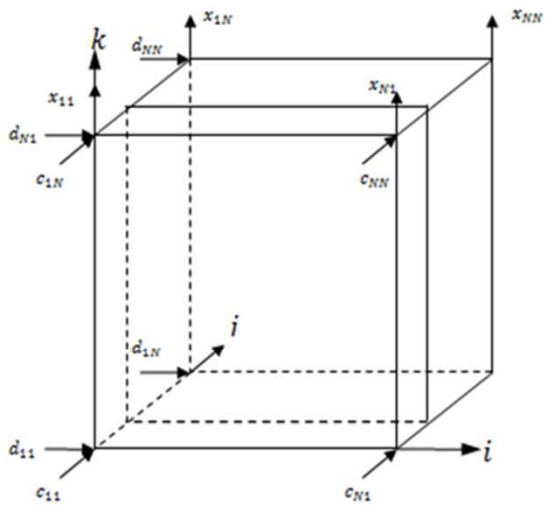
- Одна из часто используемых стратегий получения параллельных вычислительных процессов – отображение на один процессор множества подряд идущих итераций некоторого цикла.
- Подход к композиции параллельных алгоритмов: каждый алгоритм отображается на параллельный компьютер сам по себе.
- Задача – получить условия на эти отображения, обеспечивающие хорошую локальность «стыковочных» данных.

Алгоритм перемножения трёх матриц

```
do i = 1, N
  do j = 1, N
    do k = 1, N
      S1:  $c(i, j) = c(i, j) + a(i, k) * b(k, j)$ 
    enddo
  enddo
enddo
do i = 1, N
  do j = 1, N
    do k = 1, N
      S2:  $x(i, j) = x(i, j) + c(i, k) * d(k, j)$ 
    enddo
  enddo
enddo
```



Алгоритм перемножения трёх матриц



Алгоритм перемножения трёх матриц

```

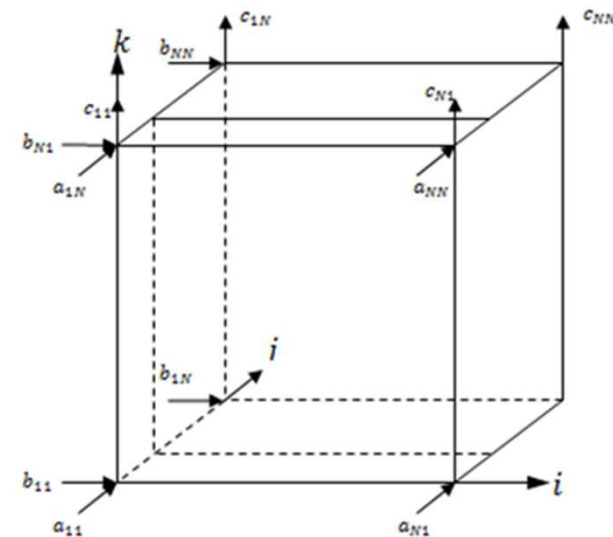
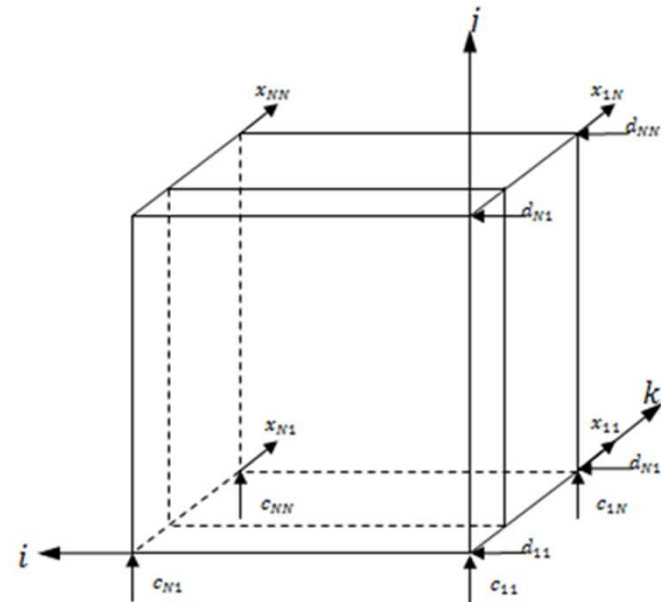
If 1 <= jgl <= Q2
do i = 1, N
  do j = 1 + (jgl - 1)r2, min(jgl r2, N)
    do k = 1, N
      S1: c(i, j) = c(i,j)+a(i,k)*b(k,j)
    enddo
  enddo
enddo

```

```

enddo
If 1 <= kgl <= Q3
do i = 1, N
  do j = 1, N
    do k = 1 + (kgl - 1)r2, min(kgl r2, N)
      S2 : x(i, j) = x(i,j)+c(i,k)*d(k,j)
    enddo
  enddo
enddo

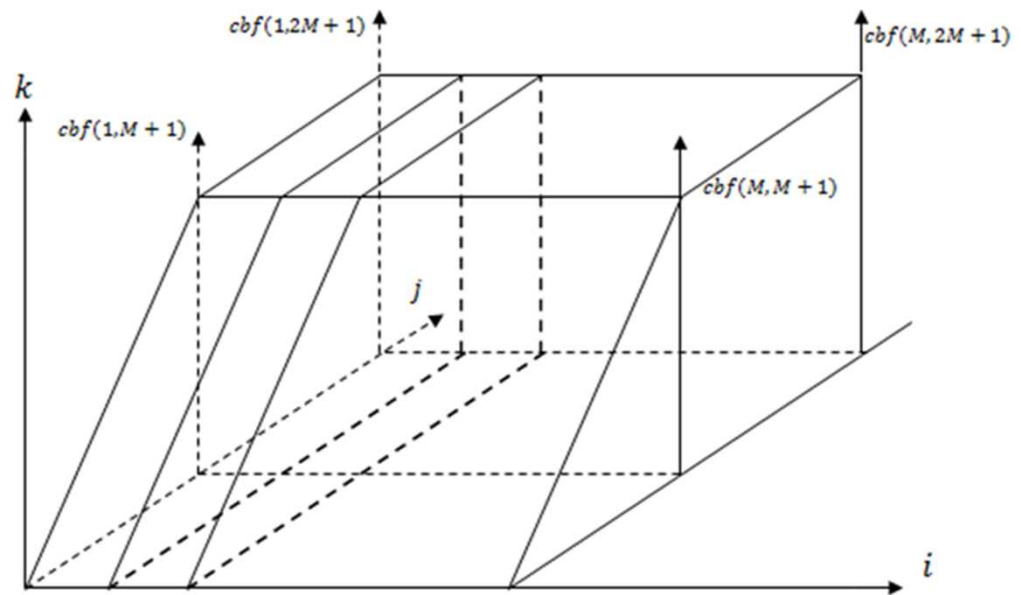
```



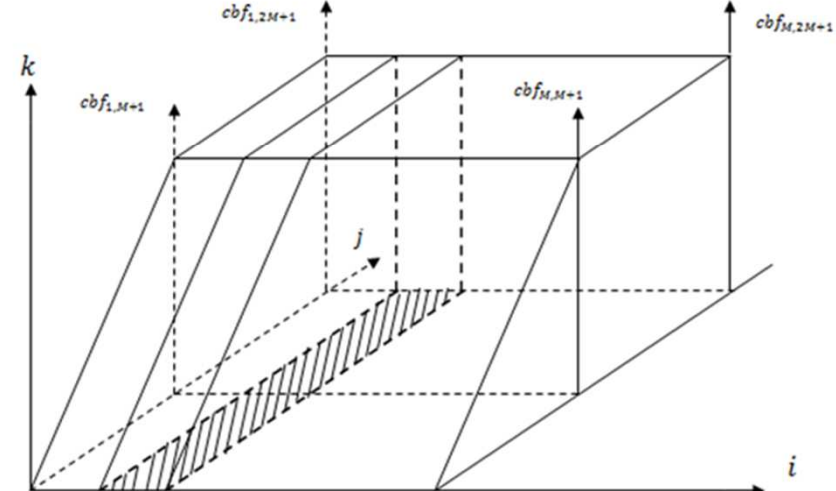
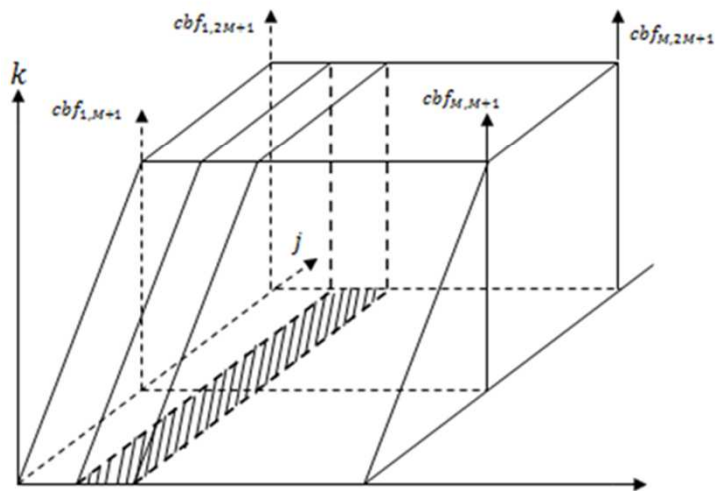
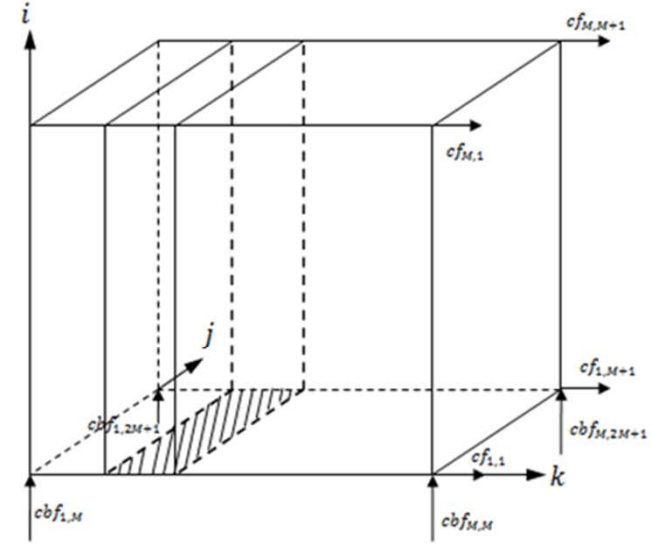
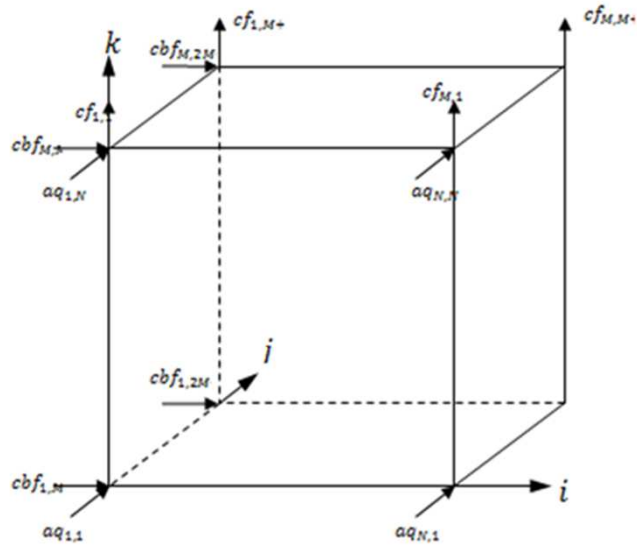
Алгоритм метода матричной прогонки

```
do k = 1, M
  S1: l(1) = 1/cbf(1,k)
  do p = 1, M
    S2: l(1) = 1/cbf(1,k)
  enddo
  do j = k+1, 2M+1
    S3: ur(i, j) = cbf(l, j) * l(1)
    do i = 2, M
      S4: cbf(i-1, j) = cbf(i, j)
          + l(i) * ur(j)
    enddo
    S5: cbf(M, j) = ur(j)
  enddo
enddo
```

```
do i = 1, N
  do j = 1, N
    do k = 1, N
      S6: c(i, j) = c(i, j) + a(i, k) * b(k, j)
    enddo
  enddo
```



Алгоритм метода матричной прогонки



Достаточные условия локальности данных композиции алгоритмов

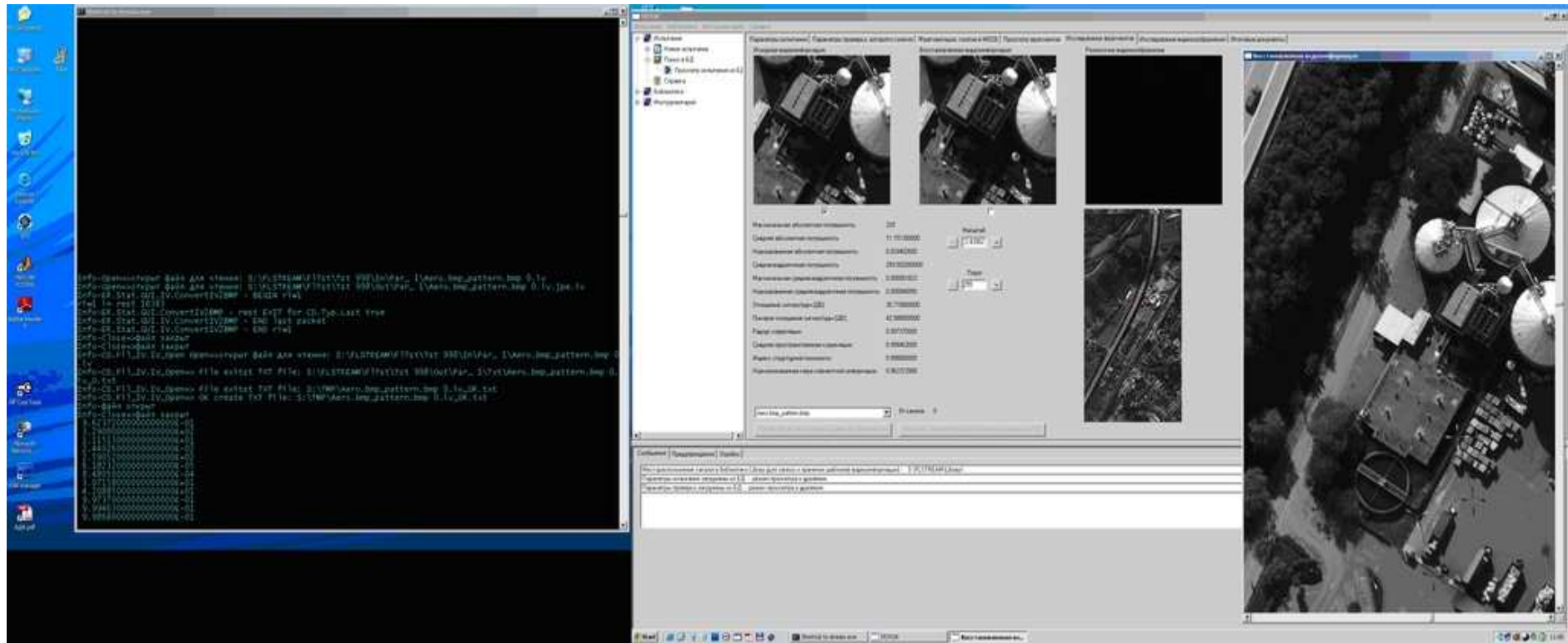
Теорема. Пусть элемент массива a_l определяется на вхождении $(a_l, S_\alpha, 1)$ и используется на вхождении (a_l, S_β, q) в правой части оператора . Определение и использование данных происходят в одном вычислительном процессе, если выполняются следующие условия:

$$(\Phi_{\alpha,\beta})_{\zeta_{\vartheta\alpha}} = e_{\zeta_{\vartheta\beta}}, \quad (1)$$

$$r_{\zeta}^{\vartheta\alpha} = r_{\zeta}^{\vartheta\beta}, \quad (2)$$

$$(\Psi_{\alpha,\beta})_{\zeta_{\vartheta\alpha}} N - \varphi_{\zeta_{\vartheta\alpha}}^{(\alpha,\beta)} - m_{\zeta}^{\vartheta\alpha} = -m_{\zeta}^{\vartheta\beta}. \quad (3)$$

Пример разработки ПО на языке Ada в которой использовались предварительные результаты работы



Программное обеспечение испытательного стенда аппаратуры сжатия видеoinформации полностью написано на языке программирования Ada [3], [4]

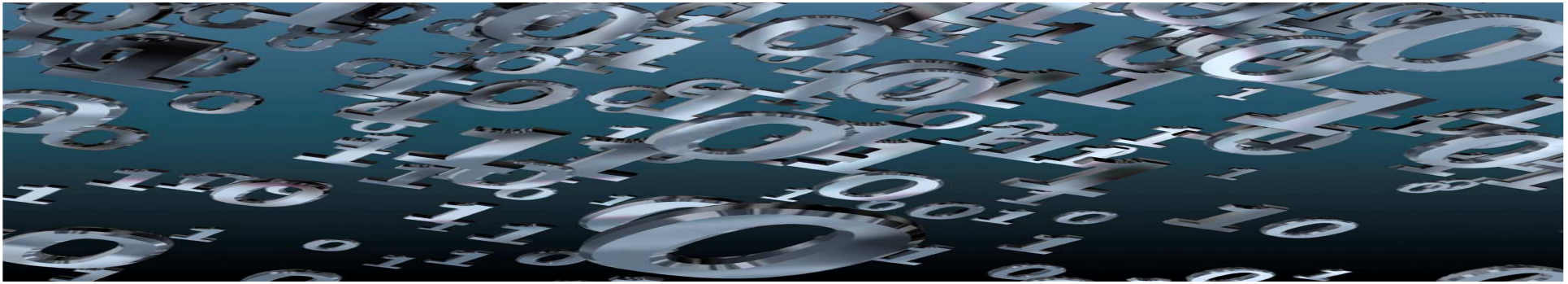


Заключение

Таким образом, получены достаточные условия отображения параллельных алгоритмов, которые обеспечивают хорошую локальность «стыковочных» данных. Исследована локальность данных, определяемых в одном, а используемых в другом гнезде циклов. Мы показали на двух примерах, что теорема работает и решение, полученное геометрическим путём [9], совпадает с решением, полученным теоретическим путём. Полученные результаты могут использоваться при автоматизации распараллеливания алгоритмов на языке Ada.

ЛИТЕРАТУРА

1. Воеводин В. В., Воеводин Вл. В. *Параллельные вычисления*. - СПб.: БХВ-Петербург, 2002. - 600с.
2. Лиходед, Н.А. *Методы распараллеливания гнезд циклов: Курс лекций / Н.А. Лиходед*. – Мн.: БГУ. 2007. – 100 с..
3. *Ada 95 Language Reference Manual ANSI/ISO 8652.1995-std: [Electronicresource]*. – <http://www.adapower.com/rm95/index.html>.
4. Куркоров С. И., Куркорова Л. С. *Параллельные алгоритмы математических моделей: исследование локальности и применение языка Ada*. Ж-л: ВІСНИК Харківського національного університету імені В.Н. Каразіна, №863 Серія: Математичне моделювання. Інформаційні технології. Автоматизовані системи управління, Випуск 12, стр. 129-142, Харків, 2009
5. Lim A.W., Liao M.S. *Blocking and array contraction across arbitrary nested loops using affine partitioning*. *Proceedings of the ACM SIGPLAN Symposium on Principles and Practice of Programming Languages*, 2001.
6. Boulet P., Darte A., Risset T., Robert Y. *(Pen)-Ultimate Tiling Integration*, *The VLSI J.* 1994. Vol. 17. P. 33-51.
7. Лиходед, Н.А. *Характеристика локальности параллельных реализаций многомерных циклов / Н.А. Лиходед // Доклады НАН Беларуси. 2010 – Т54. №1 – С.26-32.*
8. Лиходед, Н.А. *Обобщенный тайлинг / Н.А. Лиходед // Доклады НАН Беларуси. 2011.*
9. Куркорова Л.С. *Улучшение локальности композиции параллельных алгоритмов: Магистерская диссертация, Институт подготовки научных кадров НАН Беларуси, 2011.*



СПАСИБО ЗА ВНИМАНИЕ