

Отчёт по теоретическому заданию в рамках курса
«Суперкомпьютерное моделирование и технологии»

Исходный фрагмент и описание информационной структуры

В качестве условия задачи выступает фрагмент программы на языке C, листинг которой приведён в Приложении 1. Требовалось выполнить исследование информационной структуры этого фрагмента, то есть выявить имеющиеся в ней зависимости по данным и их характер, после чего составить описание информационной структуры на языке разметки Algotang. Итоговый листинг описания структуры фрагмента на языке Algotang получился вот таким:

```
<?xml version = "1.0" encoding = "UTF-8"?>
<algo>
  <params>
    <param name = "N" type = "int" value = "6"></param>
    <param name = "M" type = "int" value = "5"></param>
  </params>
  <block id = "0" dims = "1">
    <arg name = "i" val = "2..N+1"></arg>
    <vertex condition = "" type = "1">
      <in src = "i - 2"></in>
    </vertex>
  </block>
  <block id = "1" dims = "2">
    <arg name = "i" val = "2..N+1"></arg>
    <arg name = "j" val = "2..M+1"></arg>
    <vertex condition = "" type = "1">
      <in src = "i - 2, j - 2"></in>
    </vertex>
  </block>
  <block id = "2" dims = "3">
    <arg name = "i" val = "2..N+1"></arg>
    <arg name = "j" val = "2..M+1"></arg>
    <arg name = "k" val = "1..N - 1"></arg>
    <vertex condition = "(j == 1) and (k == 1)" type = "1">
      <in bsrc = "1" src = "i, M"></in>
      <in bsrc = "0" src = "i"></in>
    </vertex>
    <vertex condition = "(j > 1) or (k > 1)" type = "1">
      <in src = "i - 1, j, k"></in>
    </vertex>
  </block>
</algo>
```

Значение внешних параметров, определяющих верхние пределы циклов в исходном фрагменте, взяты небольшими (5 и 6), что не получить громоздкий информационный граф, но достаточными для выявления базовых свойств информационной структуры фрагмента.

Информационный граф фрагмента и его свойства

В соответствии с инструкциями к системе AlgoLoad из-под логина *ustmc2020ss023* в систему было загружено описание информационной структуры из предыдущего пункта. В результате в окне просмотра оказалась следующая визуализация информационного графа:

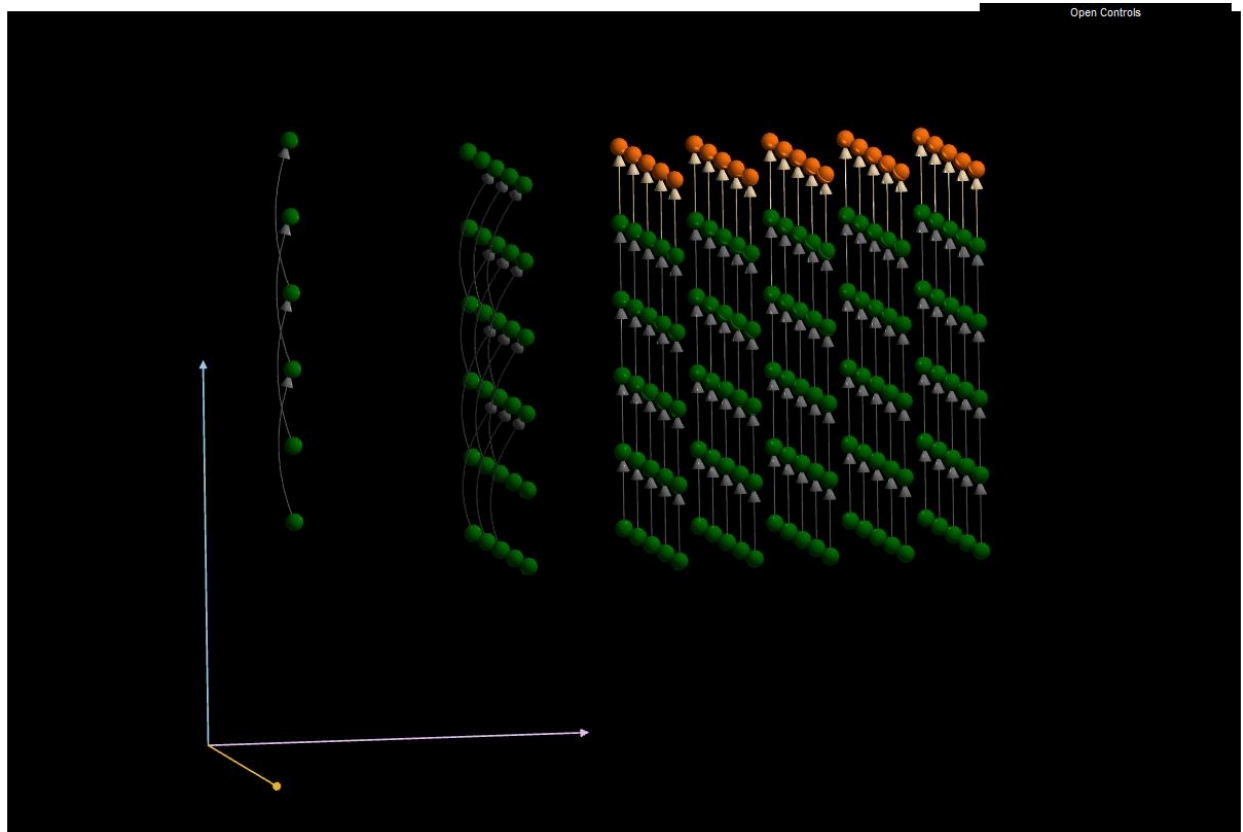


Рисунок 1 Визуализация информационного графа

Также системой были выданы некоторые из базовых свойств информационного графа для указанных в описании значений внешних параметров – число вершин, длина критического пути и ширина ЯПФ. Эти же свойства для произвольного значения внешних параметров, а также остальные свойства, требуемые в задании, получены ручным анализом исходного фрагмента, описания на языке *algolang* либо результирующей визуализации. Полный список свойств выглядит следующим образом:

- 1) Число вершин в информационном графе для заданных значений внешних параметров равняется 186. Анализ визуализации позволяет выписать число вершин C для произвольного значения M и N выражается формулой $C = N + (N^2) * M$
- 2) Длина критического пути в графе для заданных значений параметров – 6. Из визуализации понятно, что в общем случае она равна N .
- 3) Ширина канонической ЯПФ W оказалась равной 45. Используя переключение уровней ЯПФ в визуализации, возможно определить, что в общем случае она задаётся формулой $W = 2 * (N + M - 1) + M * (N - 1)$
- 4) Максимальная глубина вложенности циклов равна 3, что очевидно следует из исходного фрагмента программы.
- 5) Согласно визуализации, в данном информационном графе присутствует три различных типа дуг.
- 6) При этом длинные дуги отсутствуют.

- 7) В графе присутствует 3 области регулярности (4, если считать вершины, из которых вообще не исходит дуг в силу их близости к границам диапазонов блоков).

Приложение 1

Листинг исходного фрагмента на C

```
for (i = 2; i <= n + 1; ++i) {
    C[i] = C[i - 2] * e;
}
for (i = 2; i <= n+1; ++i) {
    for(j = 2; j <= m+1; ++j) {
        B[i][j] = B[i-2][j-2];
    }
}
for (i = 2; i <= n+1; ++i) {
    A[i][1][1] = B[i][m] + C[i];
    for (j = 2; j <= m+1; ++j) {
        for(k = 1; k <= n-1; ++k) {
            A[i][j][k] = A[i][j][k] + A[i-1][j][k];
        }
    }
}
```

Приложение 2

Листинг исходного фрагмента на C, дополненный требуемыми в задании директивами OpenMP

```
for (i = 2; i <= n + 1; ++i) {
    C[i] = C[i - 2] * e;
}
for (i = 2; i <= n+1; ++i) {
    for(j = 2; j <= m+1; ++j) {
        B[i][j] = B[i-2][j-2];
    }
}
for (i = 2; i <= n+1; ++i) {
    A[i][1][1] = B[i][m] + C[i];
    #pragma omp parallel for
    for (j = 2; j <= m+1; ++j) {
        #pragma omp parallel for
        for(k = 1; k <= n-1; ++k) {
            A[i][j][k] = A[i][j][k] + A[i-1][j][k];
        }
    }
}
```