Московский Государственный Университет имени М.В. Ломоносова Факультет вычислительной математики и кибернетики

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

Выполнил: Лазарев Владимир Александрович, 628 группа

Вариант: 231

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 Algolang. Итоговый листинг описания структуры фрагмента на языке Algolang получился вот таким:

Значение внешних параметров, определяющих верхние пределы циклов в исходном фрагменте, я решил взять небольшим, что не получить громоздкий информационный граф, но достаточным для выявления интересных свойств.

<?xml version = "1.0" encoding = "UTF-8"?>

<algo>

<params>

<param name = "N" type = "int" value = "5"></param>

<param name = "M" type = "int" value = "7"></param>

</params>

<block id = "0" dims = "1">

<arg name = "i" val = "2..N+1"></arg>

<vertex condition = "" type = "1">

<in src = "i + 1"></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 + 1, 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"></arg>

<vertex condition = "(j == 1) and (k == 1)" type = "1">

<in bsrc = "0" src = "i + 1"></in>

</vertex>

<vertex condition = "(j > 1)" type = "1">

<in src = "i - 1, j, 1"></in>

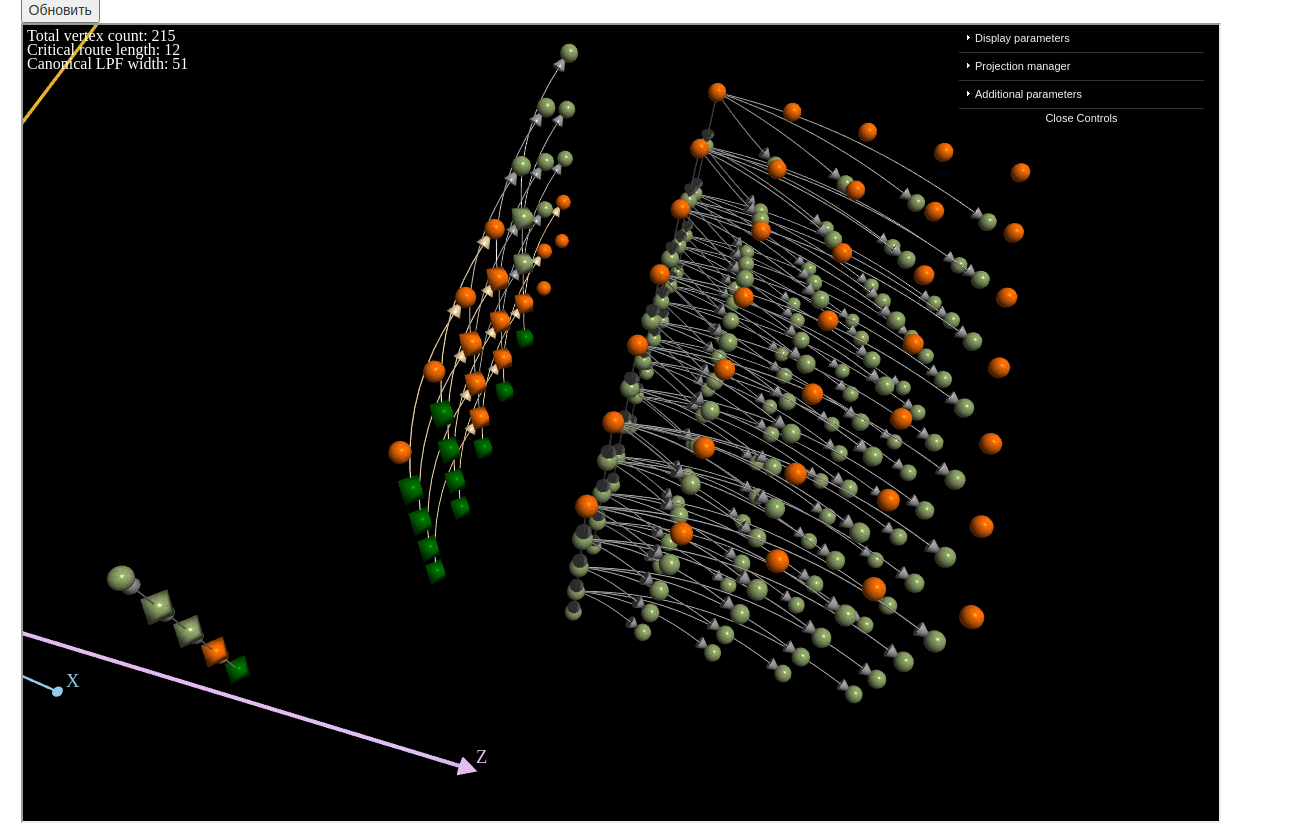
</vertex>

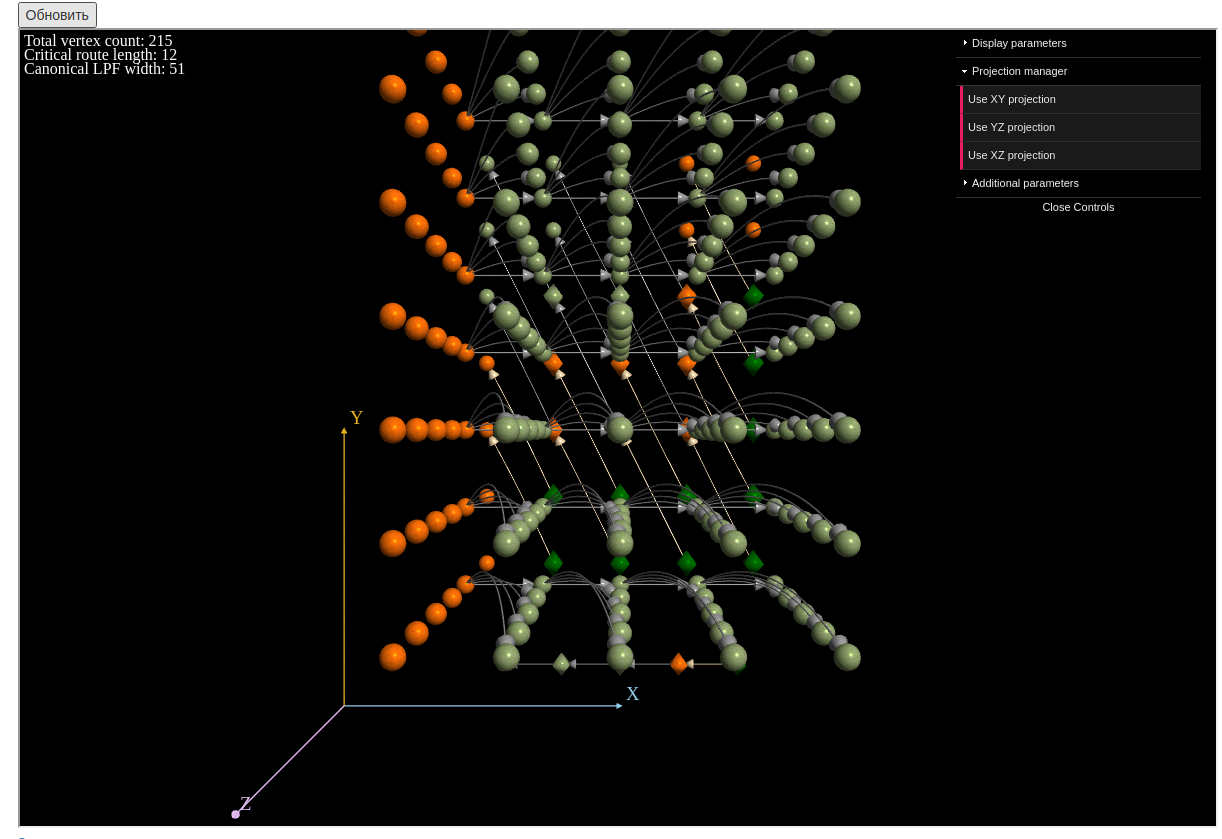
</block>

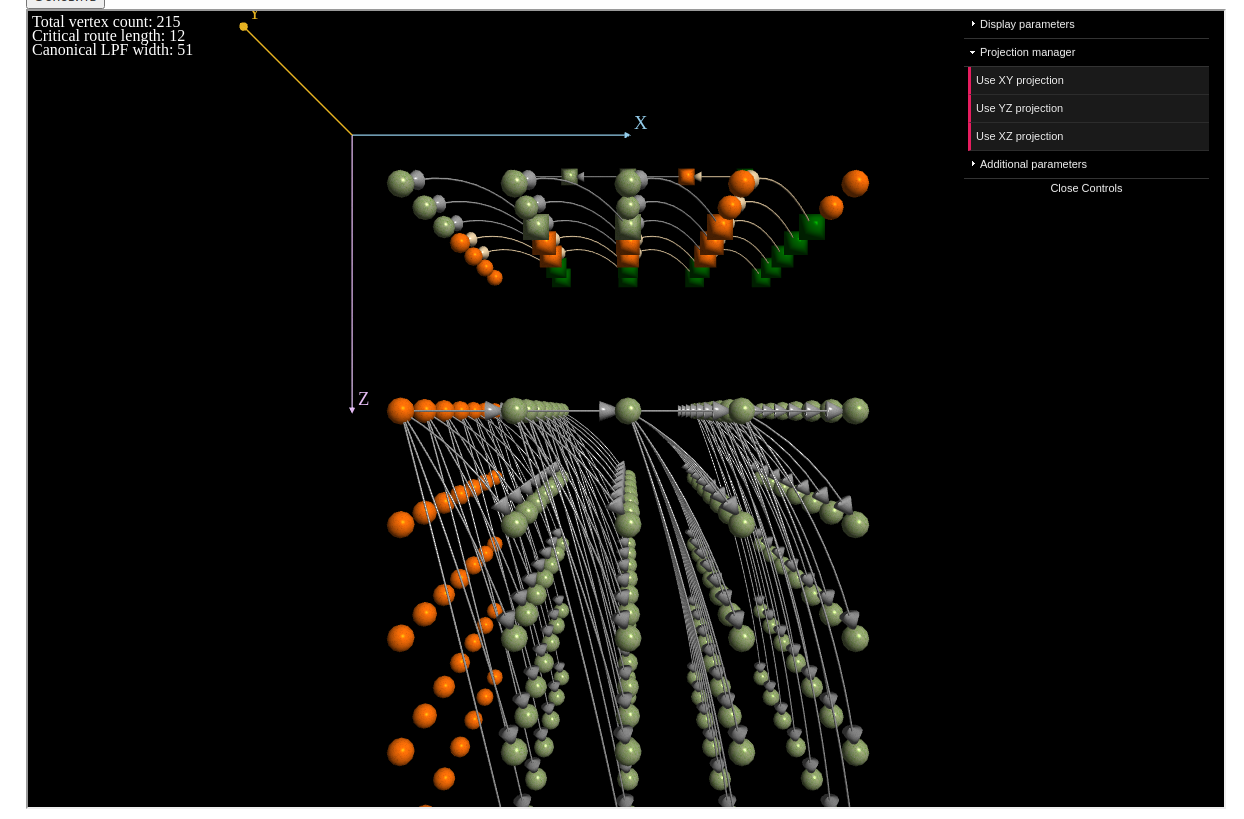
</algo>

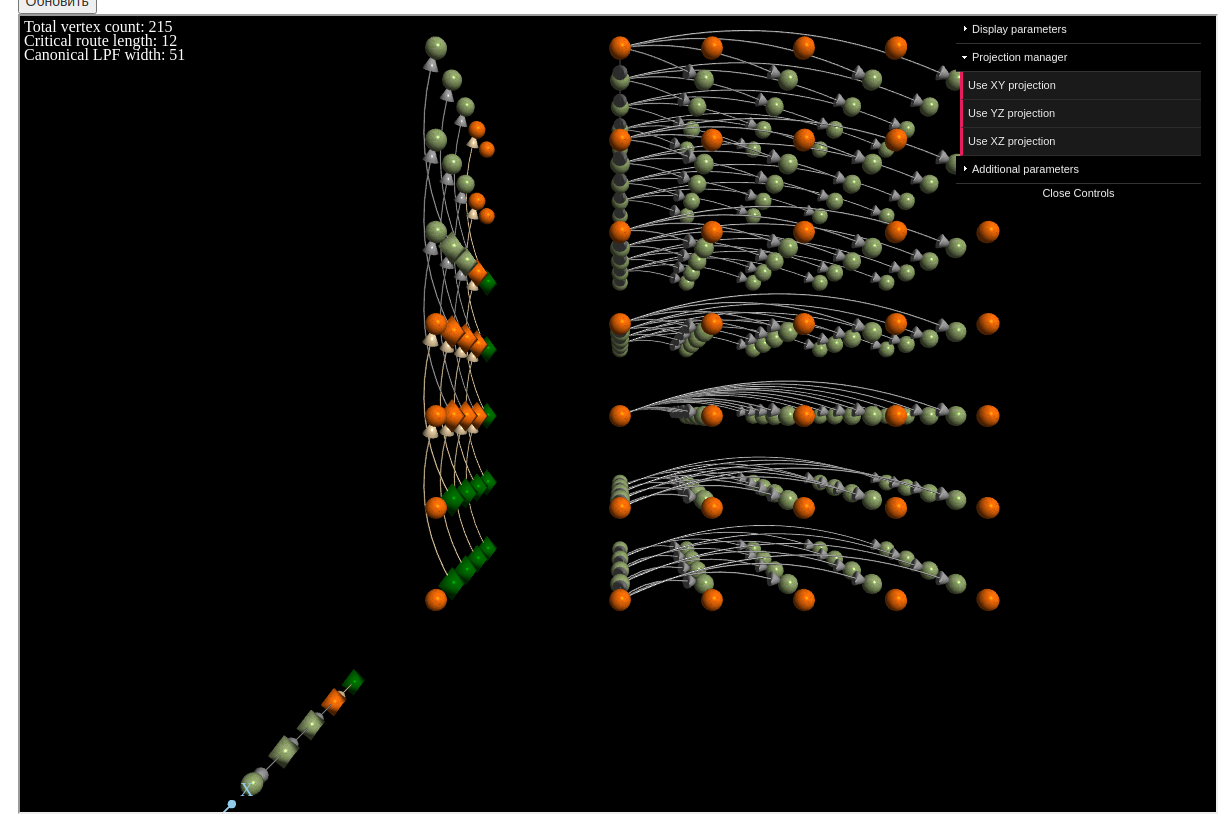
Информационный граф фрагмента и его свойства

В соответствии с инструкциями к системе Algoload я зашёл в неё под своим логином *ucmc2020ss231* и загрузил в систему описание информационной структуры из предыдущего пункта. В окне просмотра оказалась следующая визуализация информационного графа:

Рисунок 1. Визуализация графа

Рисунок 2. Проекция oXY

Рисунок 3. Проекция oXZ

Рисунок 4. Проекция oYZ

Базовые свойства:

* число вершин для данного случая: 215. Формула для общего случая: *n\*n\*m + n\*m + n;*
* длина критического пути в графе равна 5. Для общего случая – *n;*
* ширина канонической ЯПФ равна 51. Формула для общего случая: *2 \* (n + m — 1) + m \* (n — 1) + 1;*
* максимальная глубина вложенности циклов — 3;
* в данном графе присутствует 5 различных типов дуг;
* длинные дуги есть, в данном случае их 5 штук. В общем – *n.*

Приложение 1:

Листинг исходного фрагмента на C:

for (i = 2; i <= n+1; ++i) {

C[i] = C[i + 1] + D[i];

}

for (i = 2; i <= n+1; ++i) {

for(j = 2; j <= m+1; ++j) {

B[i][j] = B[i + 1][j - 2] + C[i];

}

}

for (i = 2; i <= n+1; ++i) {

A[i][1][1] = C[n + 1];

for(j = 2; j <= m+1; ++j) {

for(k = 1; k <= n; ++k) {

A[i][j][k] = A[i - 1][j][1] + A[i][j][k];

}

}

}

Приложение 2:

Фрагмент с разметкой параллельных циклов с использованием директивы OpenMP *#pragma omp parallel for*:

for (i = 2; i <= n+1; ++i) {

C[i] = C[i + 1] + D[i];

}

for (i = 2; i <= n+1; ++i) {

for(j = 2; j <= m+1; ++j) {

B[i][j] = B[i + 1][j - 2] + C[i];

}

}

for (i = 2; i <= n+1; ++i) {

A[i][1][1] = C[n + 1];

#pragma omp parallel for

for(j = 2; j <= m+1; ++j) {

#pragma omp parallel for

for(k = 1; k <= n; ++k) {

A[i][j][k] = A[i - 1][j][1] + A[i][j][k];

}

}

}