Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«Самарский государственный аэрокосмический университет

имени академика С.П. Королева

(национальный исследовательский университет)»

Факультет информатики

Кафедра технической кибернетики

Отчет по лабораторной работе № 1 по дисциплине

«Грид-технологии и облачные вычисления»

на тему

Реализация топологии «бабочка» для преобразования Фурье

Вариант № 2.6

Выполнили студенты Краснов А.П., Становова К.В.

Группа 6227М

Руководители работы Проценко В.И., Серафимович П.Г.

САМАРА 2014

содержание

[Задание 3](#_Toc407061516)

[Описание предметной области 3](#_Toc407061517)

[Описание инструментария 4](#_Toc407061518)

[Описание архитектуры 4](#_Toc407061519)

[Диаграмма потоков данных 5](#_Toc407061520)

[Приложение А 6](#_Toc407061521)

[Приложение Б 7](#_Toc407061522)

[Список использованных источников 9](#_Toc407061523)

# Задание

Реализовать топологию «бабочка» для Фурье преобразования произвольного N. Комментарии: perl mixed mode.

# Описание предметной области

Быстрое преобразование Фурье (БПФ) – алгоритм быстрого вычисления дискретного преобразования Фурье (ДПФ). Спектр последовательности  длины , , имеет следующий вид:

 (1)

где  – фазовый множитель, который обладает следующими свойствами:

1. , ;
2. ;
3. ;
4. ;

, , ,



Последняя операция называется «бабочка», она является основой приведенного метода.

Почти так же вычисляется обратное БПФ, с той лишь разницей, что результат, полученный по формуле, аналогичной выражению (1), делится на . Схема работы метода представлена на рисунке 1.

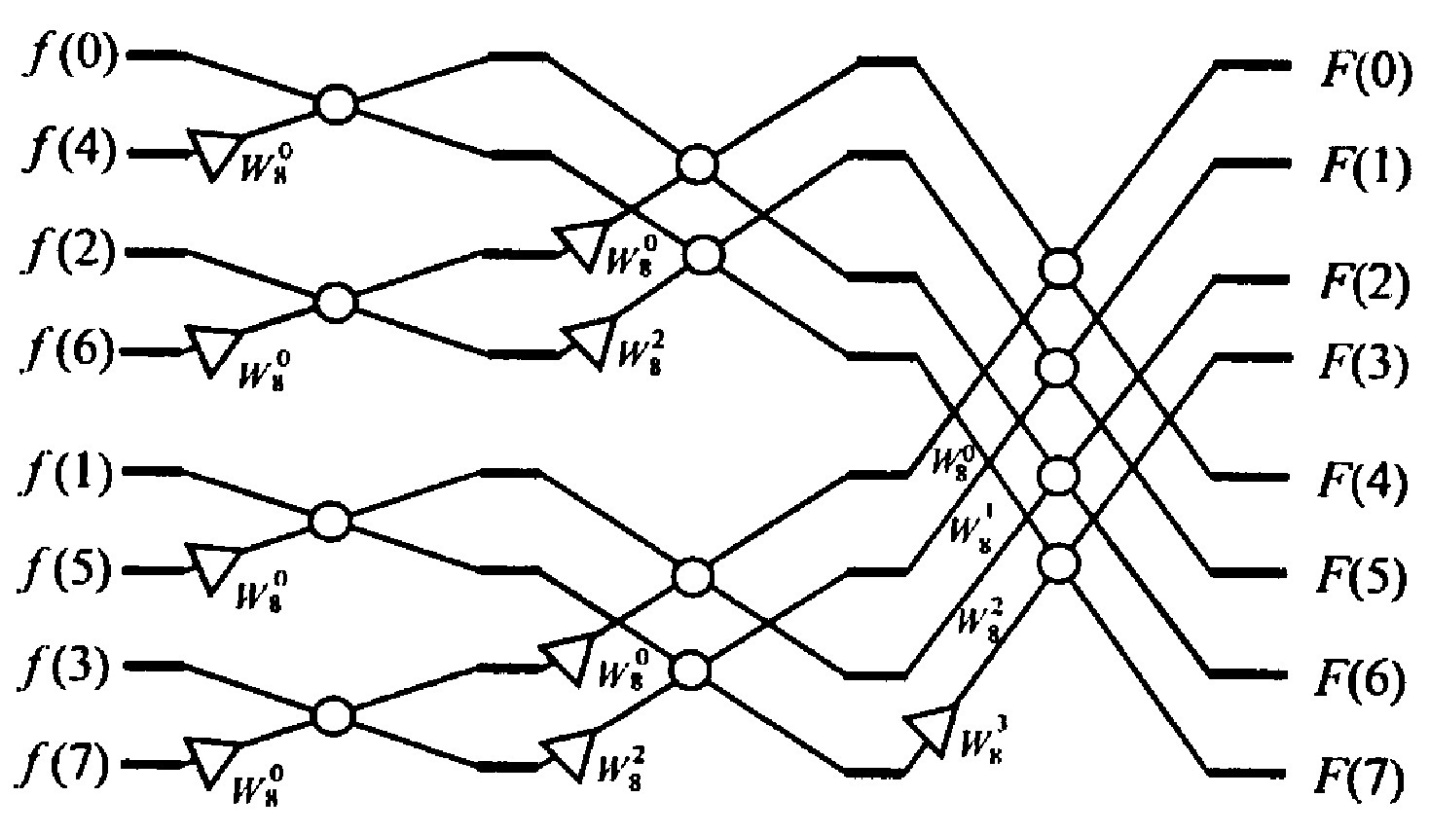


Рисунок 1 – Схема работы быстрого преобразования Фурье

# Описание инструментария

IBM InfoSphere Streams – вычислительная платформа, предназначенная для потоковой обработки данных. Данная программа установлена на виртуальной машине с операционной системой RedHat. В ней используется язык программирования IBM Streams Processing Language (IBM SPL). Среда разработки – InfoSphere Streams Studio. На рисунке 2 представлен пример работы со средой разработки.

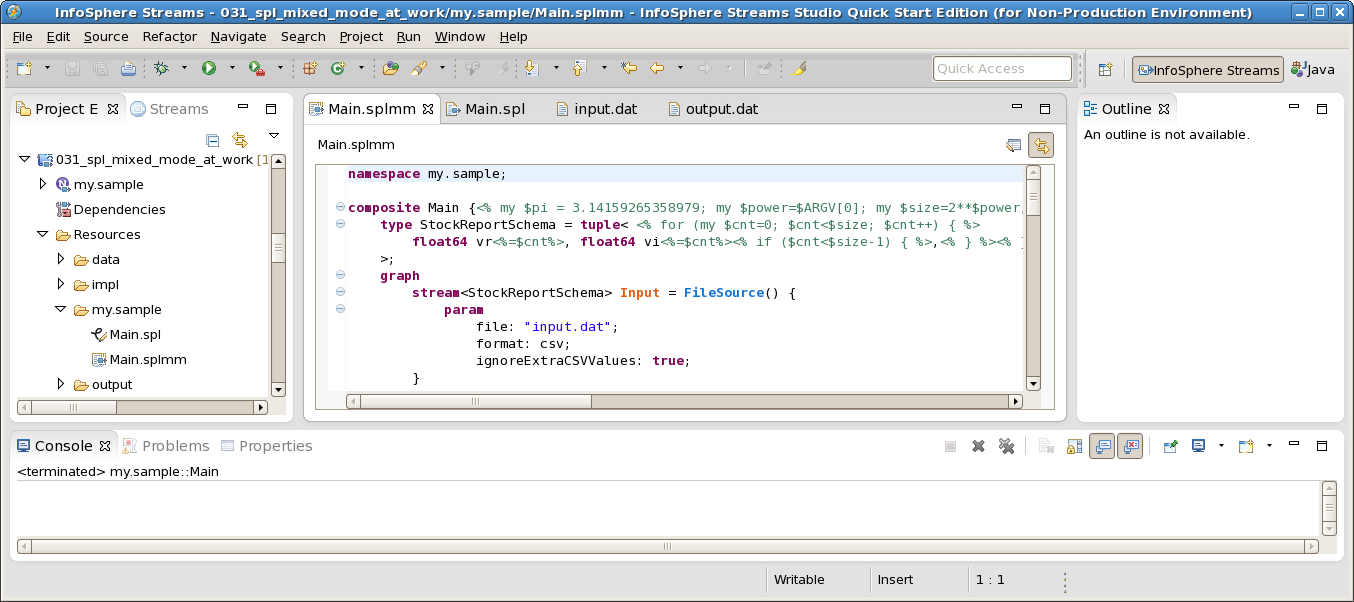


Рисунок 2 – Пример работы со средой разработки IBM InfoSphere Streams Studio

# Описание архитектуры

В данной программе используется Perl mixed mode – вставки в SPL кода на языке Perl. Этот режим применяется для генерации однотипного SPL-кода в зависимости от значений аргументов запуска программы.

Оператор Input = FileSource() считывает файл и передает в выходной поток числа, записанные в этом файле. Числа записаны в виде вещественной и мнимой частей, разделенных запятыми, как и сами числа. Используемые параметры: file – имя входного файла, format – его формат, ignoreExtraCSVValues – опция, позволяющая игнорировать пробелы, отступы и т.п.

Операторы Inverse1 = Custom(Input) и Inverse<%=$i%> = Custom(Inverse<%=$i-1%>) необходимы для осуществления двоично-инверсной перестановки входных значений.

Операторы Output1 = Custom(Inverse<%=$power-1%>) и Output<%=$i%> = Custom(Output<%=$i-1%>) применяются для реализации операций «бабочка», которые приводят к конечному результату.

Оператор FileWriter1 = FileSink(Output<%=$power%>) печатает рассчитанные данные в выходной файл. Используемый параметр: file – имя выходного файла.

В качестве входного аргумента программы используется параметр препроцессора mixed-mode (см. рисунок 3), равный степени, в которую нужно возвести 2, чтобы получить размер обрабатываемых последовательностей. Для примера выберем аргумент равным 4, что соответствует 16 элементам в каждой последовательности.

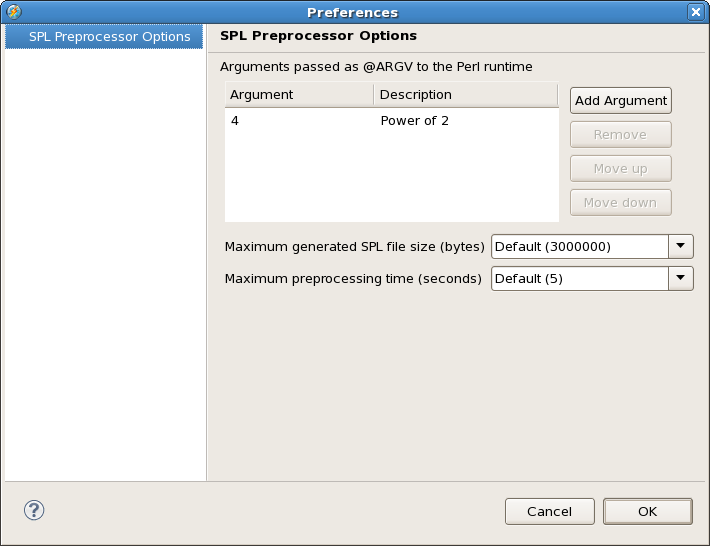


Рисунок 3 – Опции препроцессора mixed mode

# Диаграмма потоков данных

На рисунке 4 представлена диаграмма потоков данных (Data Flow Diagram), описывающая работу представленной программы для последовательностей длины 16.

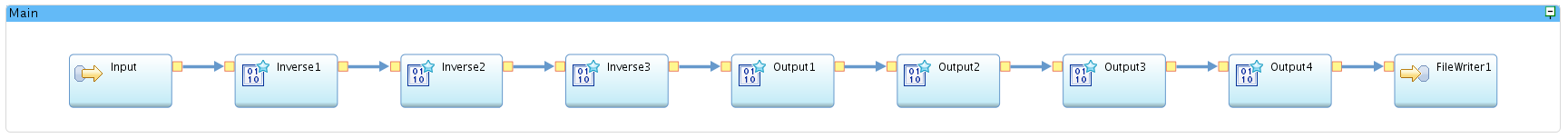


Рисунок 4 – Диаграмма потоков данных

Приложение А

Входные данные (файл input.dat).

1, -1.5, 2, -1.3, 3, -1.1, 4, -0.9, 5, -0.7, 6, -0.5, 7, -0.3, 8, -0.1, -1, 0.1, -2, 0.3, -3, 0.5, -4, 0.7, -5, 0.9, -6, 1.1, -7, 1.3, -8, 1.5

-3, 5.4, 4, 2.0, 4.5, 2.1, -1.3, 2, 7, 0.1, -9.5, 4, 4.8, -5.4, 3.2, 2.9, 8.1, 8, -5.6, 9, -2, 3.3, -4.6, 5.8, 1.1, 1.7, 1.2, -3.4, 5.6, -2.3, 4.3, 3.1

Выходные данные (файл output.dat).

0,0,-24.3178855564894,-51.8733949212586,-3.86274169979695,-1.60000000000001,4.36560197089175,-16.5660576266549,-1.6,-1.6,7.48445148715747,-8.28178637919298,-0.66274169979695,-1.6,8.64217408229603,-3.58912367379654,0,-1.6,9.27869365791086,0.389123673796615,0.662741699796954,-1.6,9.62262312849917,5.08178637919304,1.6,-1.6,9.15474041142117,13.3660576266549,3.86274169979695,-1.59999999999999,-8.23039918168701,48.6733949212584

17.8,38.3,5.23504499855365,-18.1715416665943,32.9395490323255,30.3187337521542,-34.5716323812111,-5.01364153882163,-1.90000000000002,29,-14.0113126053036,-24.2430310813825,-21.6861435713737,2.35649711574551,-15.9699947852062,22.8263461660638,34.4,-12.5,-26.3924043114344,-4.20246987042342,-12.7395490323255,8.68126624784581,-0.833244942316792,0.865618464786066,2.50000000000002,6,-15.6313280818157,12.6170426184003,-10.5138564286262,5.04350288425445,13.374872108734,-5.47832309202818

Приложение Б

Текст программы на языке SPL Mixed Mode (файл Main.splmm).

namespace my.sample;

composite Main {<% my $pi = 3.14159265358979; my $power=$ARGV[0]; my $size=2\*\*$power; %>

type StockReportSchema = tuple< <% for (my $cnt=0; $cnt<$size; $cnt++) { %>

float64 vr<%=$cnt%>, float64 vi<%=$cnt%><% if ($cnt<$size-1) { %>,<% } %><% } %>

>;

graph

stream<StockReportSchema> Input = FileSource() {

param

file: "input.dat";

format: csv;

ignoreExtraCSVValues: true;

}

stream<StockReportSchema> Inverse1 = Custom(Input) {

logic

onTuple Input: {

StockReportSchema srs = {<% for (my $k=0; $k<2\*\*($power-1); $k++) { %><% my $b=$k+2\*\*($power-1); my $c=2\*$k; my $d=$c+1; %>

vr<%=$k%> = Input.vr<%=$c%>,

vi<%=$k%> = Input.vi<%=$c%>,

vr<%=$b%> = Input.vr<%=$d%>,

vi<%=$b%> = Input.vi<%=$d%><% if ($d<$size-1) { %>,<% } %><% } %>

};

submit(srs, Inverse1);

}

}

<% for (my $i=2; $i<$power; $i++) { %>

stream<StockReportSchema> Inverse<%=$i%> = Custom(Inverse<%=$i-1%>) {

logic

onTuple Inverse<%=$i-1%>: {

StockReportSchema srs = {<% for (my $j=0; $j<2\*\*($i-1); $j++) { %><% for (my $k=0; $k<2\*\*($power-$i); $k++) { %><% my $a=2\*\*($power-$i+1)\*$j+$k; my $b=$a+2\*\*($power-$i); my $c=2\*(2\*\*($power-$i)\*$j+$k); my $d=$c+1; %>

vr<%=$a%> = Inverse<%=$i-1%>.vr<%=$c%>,

vi<%=$a%> = Inverse<%=$i-1%>.vi<%=$c%>,

vr<%=$b%> = Inverse<%=$i-1%>.vr<%=$d%>,

vi<%=$b%> = Inverse<%=$i-1%>.vi<%=$d%><% if ($d<$size-1) { %>,<% } %><% } %><% } %>

};

submit(srs, Inverse<%=$i%>);

}

}

<% } %>

stream<StockReportSchema> Output1 = Custom(Inverse<%=$power-1%>) {

logic

onTuple Inverse<%=$power-1%>: {

StockReportSchema srs = {<% for (my $j=0; $j<2\*\*($power-1); $j++) { %><% my $a=2\*$j; my $b=$a+1; %>

vr<%=$a%> = Inverse<%=$power-1%>.vr<%=$a%> + Inverse<%=$power-1%>.vr<%=$b%>,

vi<%=$a%> = Inverse<%=$power-1%>.vi<%=$a%> + Inverse<%=$power-1%>.vi<%=$b%>,

vr<%=$b%> = Inverse<%=$power-1%>.vr<%=$a%> - Inverse<%=$power-1%>.vr<%=$b%>,

vi<%=$b%> = Inverse<%=$power-1%>.vi<%=$a%> - Inverse<%=$power-1%>.vi<%=$b%><% if ($b<$size-1) { %>,<% } %><% } %>

};

submit(srs, Output1);

}

}

<% for (my $i=2; $i<=$power; $i++) { %>

stream<StockReportSchema> Output<%=$i%> = Custom(Output<%=$i-1%>) {

logic

onTuple Output<%=$i-1%>: {

StockReportSchema srs = {<% for (my $j=0; $j<2\*\*($power-$i); $j++) { %><% for (my $k=0; $k<2\*\*($i-1); $k++) { %><% my $p=2\*$pi\*$k\*2\*\*($power-$i)/$size; my $a=2\*\*$i\*$j+$k; my $b=$a+2\*\*($i-1); %>

vr<%=$a%> = Output<%=$i-1%>.vr<%=$a%> + ((float64) <%=cos($p)%>) \* Output<%=$i-1%>.vr<%=$b%> + ((float64) <%=sin($p)%>) \* Output<%=$i-1%>.vi<%=$b%>,

vi<%=$a%> = Output<%=$i-1%>.vi<%=$a%> - ((float64) <%=sin($p)%>) \* Output<%=$i-1%>.vr<%=$b%> + ((float64) <%=cos($p)%>) \* Output<%=$i-1%>.vi<%=$b%>,

vr<%=$b%> = Output<%=$i-1%>.vr<%=$a%> - ((float64) <%=cos($p)%>) \* Output<%=$i-1%>.vr<%=$b%> - ((float64) <%=sin($p)%>) \* Output<%=$i-1%>.vi<%=$b%>,

vi<%=$b%> = Output<%=$i-1%>.vi<%=$a%> + ((float64) <%=sin($p)%>) \* Output<%=$i-1%>.vr<%=$b%> - ((float64) <%=cos($p)%>) \* Output<%=$i-1%>.vi<%=$b%><% if ($b<$size-1) { %>,<% } %><% } %><% } %>

};

submit(srs, Output<%=$i%>);

}

}

<% } %>

() as FileWriter1 = FileSink(Output<%=$power%>) {

param

file: "output.dat";

}

}

Список использованных источников

1. IBM Knowledge Center [Интернет-ресурс]. URL: <http://www-01.ibm.com/support/knowledgecenter/>, дата доступа: 01.12.2014.
2. БПФ по основанию 2 с прореживанием по времени [Интернет-ресурс]. URL: <http://www.dsplib.ru/content/thintime/thintime.html>, дата доступа: 01.12.2014.
3. Методы компьютерной обработки изображений / Под ред. В.А. Сойфера. М.: ФИЗМАТЛИТ, 2003. 784 с.