

**中国科学院软件研究所**

**工作周报**

**姓名： 赵士杰**

**时间：2025年7月20日**

目录

[1. 工作任务 3](#_Toc203968965)

[1.1. 学术资料阅读 3](#_Toc203968966)

[1.2. 实践环节 3](#_Toc203968967)

[1.3. 本周出勤情况 3](#_Toc203968968)

[2. 工作进展 3](#_Toc203968969)

[2.1. 学术资料阅读进展 3](#_Toc203968970)

[2.2. 实践环节进展 3](#_Toc203968971)

[3. 学术资料阅读笔记 4](#_Toc203968972)

[4. 下周计划 4](#_Toc203968973)

[4.1. 学术资料阅读 4](#_Toc203968974)

[4.2. 实践环节 4](#_Toc203968975)

[5. 参考文献 4](#_Toc203968976)

# 工作任务

## 学术资料阅读

无

## 实践环节

完成Flink的编译。

进行GeoFlink具体代码的优化，完成RVV的C程序的编写。

进行交叉编译。

尝试优化流程。

## 本周出勤情况

本周出勤5天：7.14，7.15，7.16，7.17，7.18。

# 工作进展

## 学术资料阅读进展

## 实践环节进展

1. 进行Flink的再次编译
2. 完成对于RVV实际优化的C代码的编写

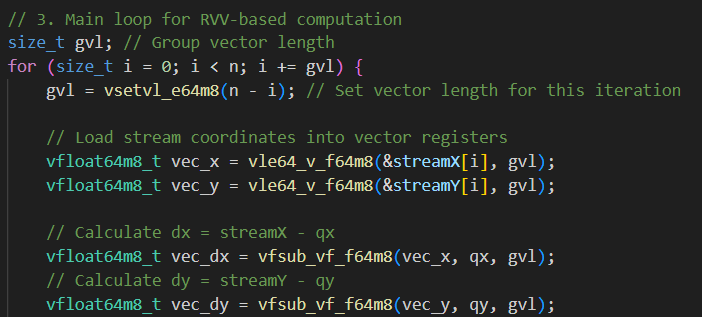
性能瓶颈位于 windowBased 方法内部的 apply 函数中，其中 DistanceFunctions.getDistance 在一个循环内被反复调用。

修改引入以下变化：

①批量处理：不再逐点计算距离，而是先将窗口内的所有点收集到一个列表中。

②数据准备：将点的 X 和 Y 坐标分别提取到两个 double 数组中。

③JNI 调用：调用 RvvDistanceCalculator.calculateDistances，通过一次函数调用，利用底层的 RVV 指令并行计算出所有点到查询点的距离。



1. 进行GeoFlink的编译

因为性能受限，采用交叉编译的方式进行。

# 学术资料阅读笔记

无

# 下周计划

## 学术资料阅读

无

## 实践环节

寻找新的值得优化的库。

等待配置好RVV环境之后进行测试。

# 参考文献

无