

**中国科学院软件研究所**

**工作周报**

**姓名： 赵士杰**

**时间：2025年07月27日**

目录

[1. 工作任务 3](#_Toc204568849)

[1.1. 学术资料阅读 3](#_Toc204568850)

[1.2. 实践环节 3](#_Toc204568851)

[1.3. 本周出勤情况 3](#_Toc204568852)

[2. 工作进展 3](#_Toc204568853)

[2.1. 学术资料阅读进展 3](#_Toc204568854)

[2.2. 实践环节进展 3](#_Toc204568855)

[2.2.1 替换qemu内核使支持RVV 3](#_Toc204568856)

[2.2.2完成环境配置 3](#_Toc204568857)

[2.2.3 完成SpatialFlink的编译、测试、issue提交 4](#_Toc204568858)

[3. 学术资料阅读笔记 6](#_Toc204568859)

[4. 下周计划 6](#_Toc204568860)

[4.1. 学术资料阅读 6](#_Toc204568861)

[4.2. 实践环节 6](#_Toc204568862)

[5. 参考文献 6](#_Toc204568863)

# 工作任务

## 学术资料阅读

## 实践环节

* 完成RVV的相关配置
* 完成Flink的再次编译
* 完成有关优化后SpatialFlink的编译、测试、issue询问

## 本周出勤情况

出勤5天：7.21，7.22，7.23，7.24，7.25

# 工作进展

## 学术资料阅读进展

## 实践环节进展

### 2.2.1 替换qemu内核使支持RVV

根据群内文档，成功完成qemu中内核的替换，并且成功验证其支持RVV。

Qemu内核替换因为各种原因炸了三次（悲），产出对于如何恢复内核炸掉之后的QEMU的文档一份。



### 2.2.2完成环境配置

完成Flink的编译。

Kafka的编译过程中网络中断，导致编译受阻，因此还在进行编译。最初准备使用网上的jar包直接部署。在网上下载jar包并部署后，可以正常启动，但是消息传输出现问题，无法正常接受，经查TCP层正常，Kafka协议层出现问题（详见层次分析）。当前暂时将问题定位于BiSheng JDK在RISC-V架构上的Kafka客户端协议兼容性问题，导致网络I/O层面出现故障。遂后续选择自己编译。

### 2.2.3 完成SpatialFlink的编译、测试、issue提交

**编译：**

在本地WSL配置交叉编译工具链，完成对于支持RVV的SpatialFlink的交叉编译，并成功验证编译。

**性能测试：**

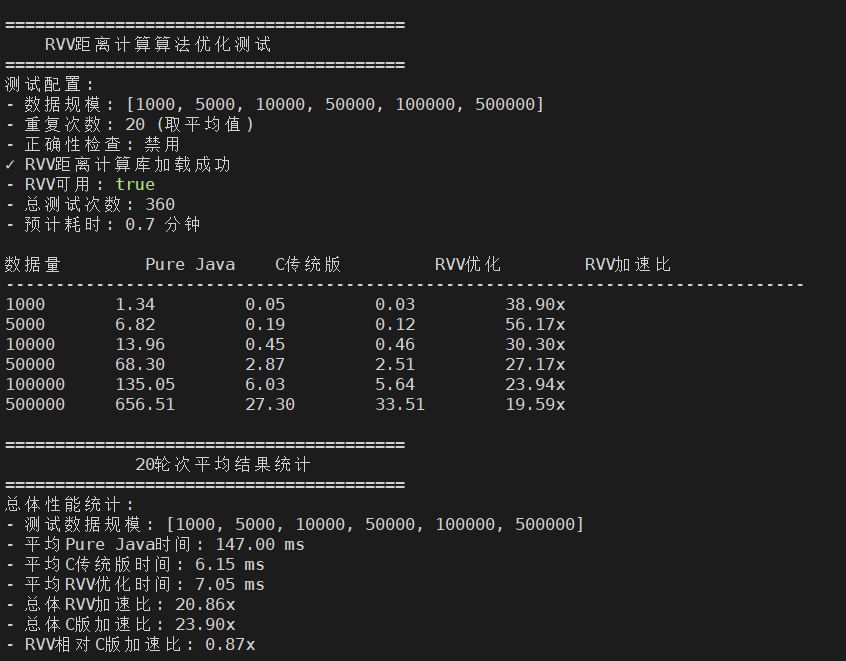
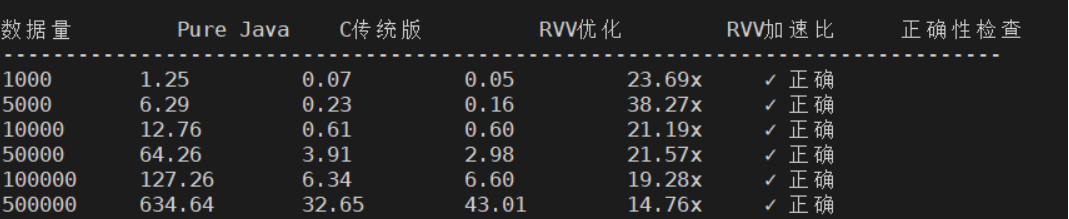
因为Kafka的问题，因此当时无法进行全套的功能测试和性能测试，于是便把优化的代码抽出来，以单独的程序段的方式进行性能测试。将JAVA版本、C版本和RVV版本进行对比，查看性能优化。提高约有20-30倍

发现问题：测试时，C语言版本和RVV版本对于JAVA版本都有极大的性能提升，但是比较C语言版本和RVV版本时，发现C语言版本反而比RVV版本还快。

问题解决：经人提醒，认为可能是最后的结果校验出现了问题，分割导致校验过多拖慢了RVV的速度。于是进行尝试，在第一遍测试保证数据正确性的情况下，关闭数据校验。果然发现在普遍情况下正常了：相同数据量下RVV版本的代码运行快于C语言版本。

发现问题：但是又注意到，在数量大的随机数据集（500K）上，C语言版本的程序反而比RVV的快了一些？

问题解决：问题目前没有解决，下周有时间的话查看一下，准备开组会的时候问一下。



**功能测试：**

因为Kafka的编译受阻，执行脚本的时候中间链接断了，因此，暂未编译成功，为了不影响进度，决定另辟蹊径。

创建独立的测试程序作为测试入口点，完全脱离Flink和Kafka环境，使用main函数进入，手动创建输入数据和查询点，模拟Flink的数据，并且直接调用对应的方法，对比在原本JAVA方法和新的RVV方法的结果。

功能测试基本完成，没有错误。可能下周等Kafka编译好的时候会再测一下。

**issue提交：**

提交issue，询问是否接受有关RVV优化的PR：https://github.com/aistairc/SpatialFlink/issues/13

# 学术资料阅读笔记

QEMU内核错误恢复方法（见上）。

# 下周计划

## 学术资料阅读

无

## 实践环节

寻找更有价值更出名的库进行优化。

# 参考文献

无