

# 研究生考核报告

报告人：牟向翊

课题组：触发与数据获取组

导师：李飞

专业：计算机应用技术

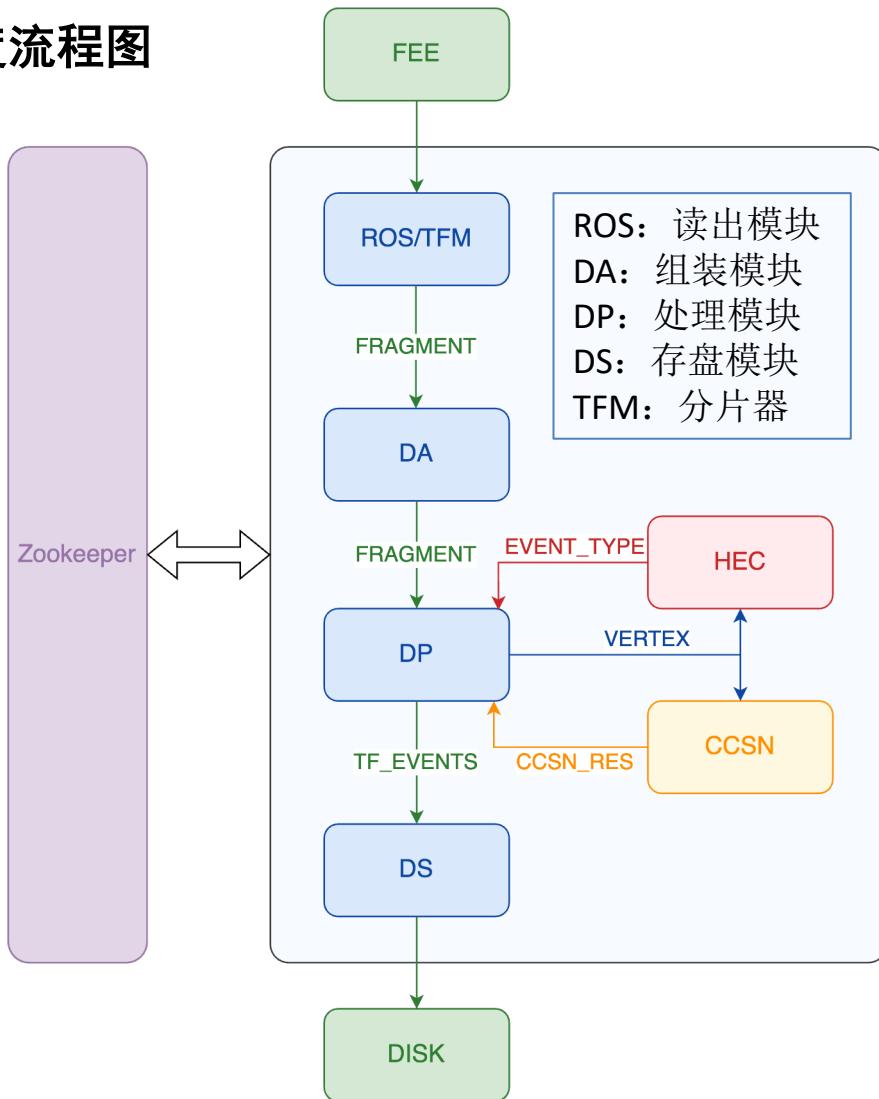
时间：2025.09.05

# 报告主要内容

- ◆ 数据流自调度算法的实现
- ◆ 数据流的高可用升级
- ◆ TT探测器的集成

# 数据流调度优化

## 调度流程图



## DFM功能

记录调度信息→使用ZK记录  
进行任务调度→**自调度算法**  
数据同步→TFM进行同步

## 自调度算法

每个组件自行决定任务调度  
通过ZK中的信息进行调度  
不依赖单节点的判断，有利于高可用的实现

## Zookeeper

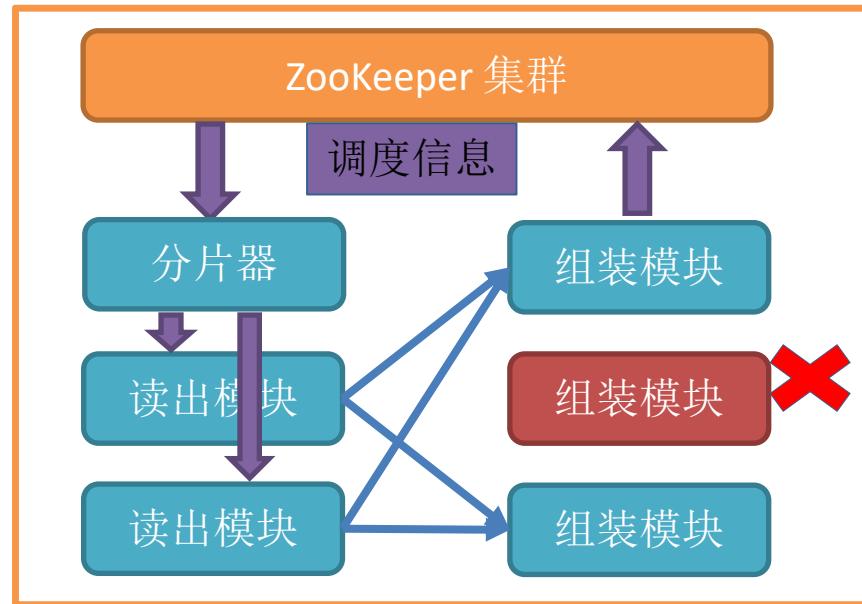
保存各个节点的状态  
提供高可用支持  
临时节点，进程崩溃后自动退出，不影响调度  
Zookeeper集群的升级

## 总结

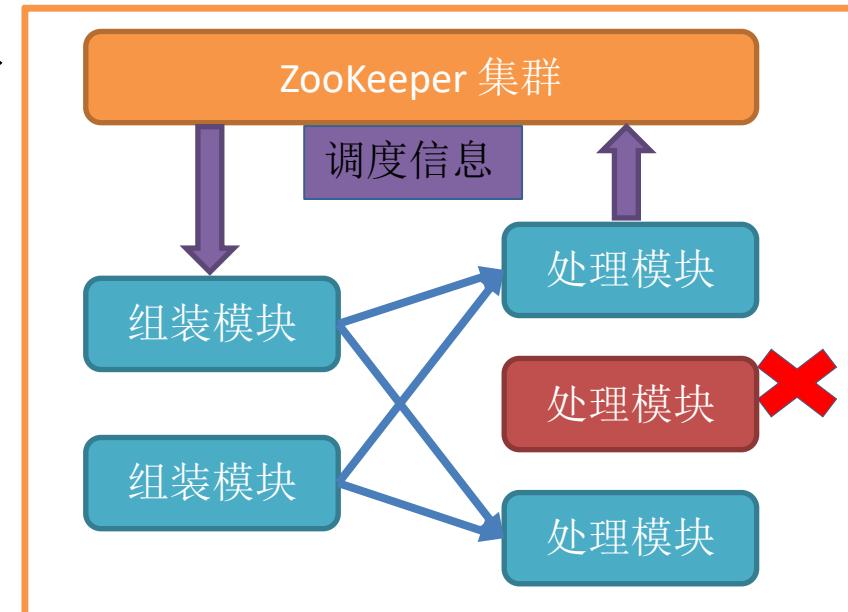
简化了调度过程  
提高了横向扩展能力  
关注了可用性

# 数据流调度优化

读出模块->  
组装模块



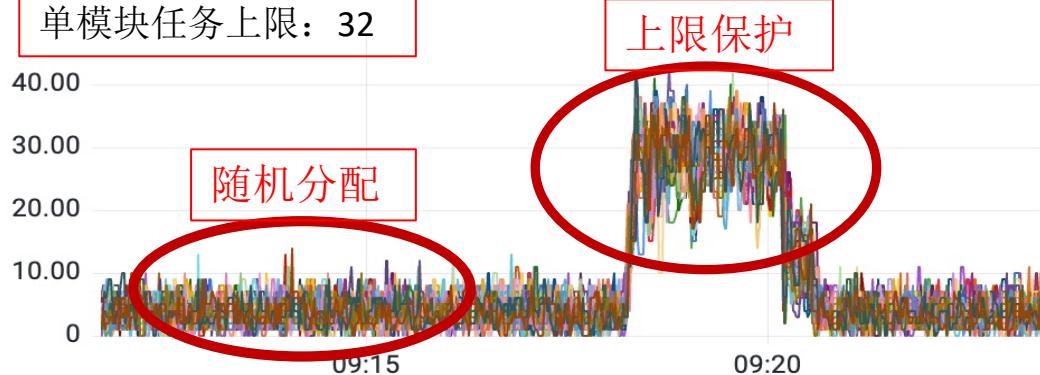
组装模块->  
处理模块



处理模块数量: 40  
单模块任务上限: 32

处理模块任务数量

上限保护



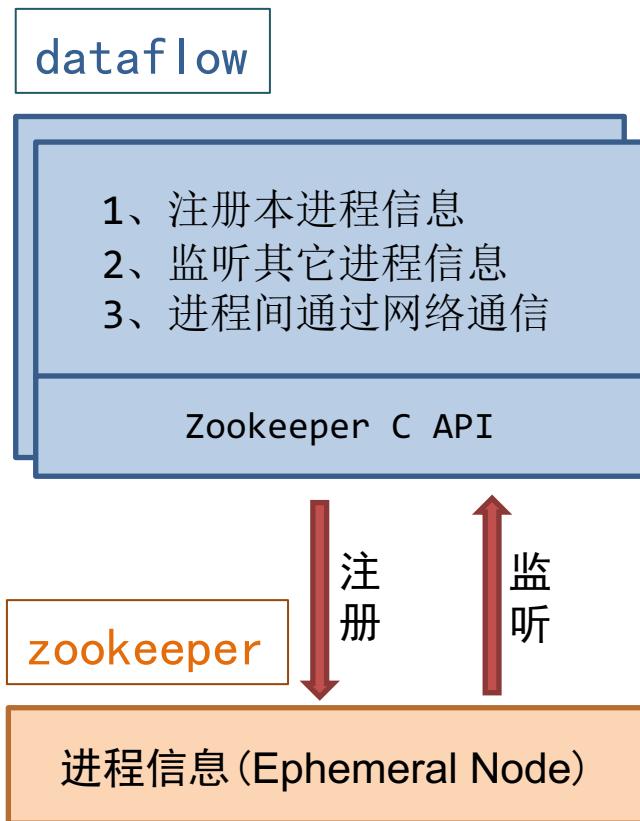
随机分配  
所有节点的硬件相同  
能够实现负载均衡  
上限保护  
延迟导致的任务数量超出上限

InCast问题

Zookeeper访问次数优化  
 $N_{task} \times N_{DA} \rightarrow N_{task}$

# 数据流的高可用升级

## 进程信息的动态更新



通过将**进程信息**注册到第三方服务zookeeper。

所有进程监听zk上的节点信息，实现数据流进程**动态的退出和加入集群**。

Zk中的进程信息为**临时节点**，进程崩溃后，zk会**自动删除**该节点。

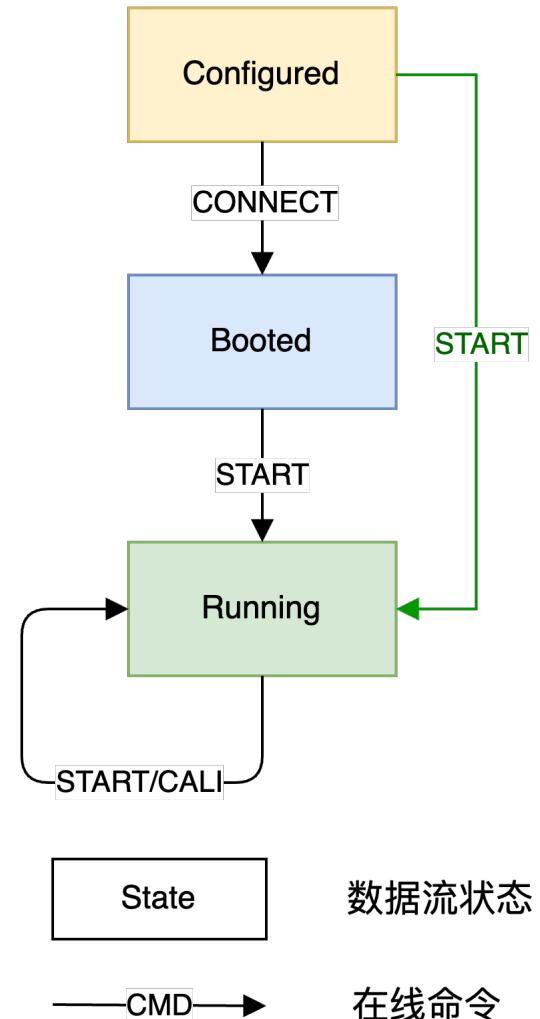
## 状态机添加新路径

故障进程退出集群后，k8s自动重启进程，该进程进入Configured状态，此时的在线命令是START，该进程无法进入Running状态

添加Configured状态到Running状态的新路径，保证加入集群的进程能顺利进入Running状态。

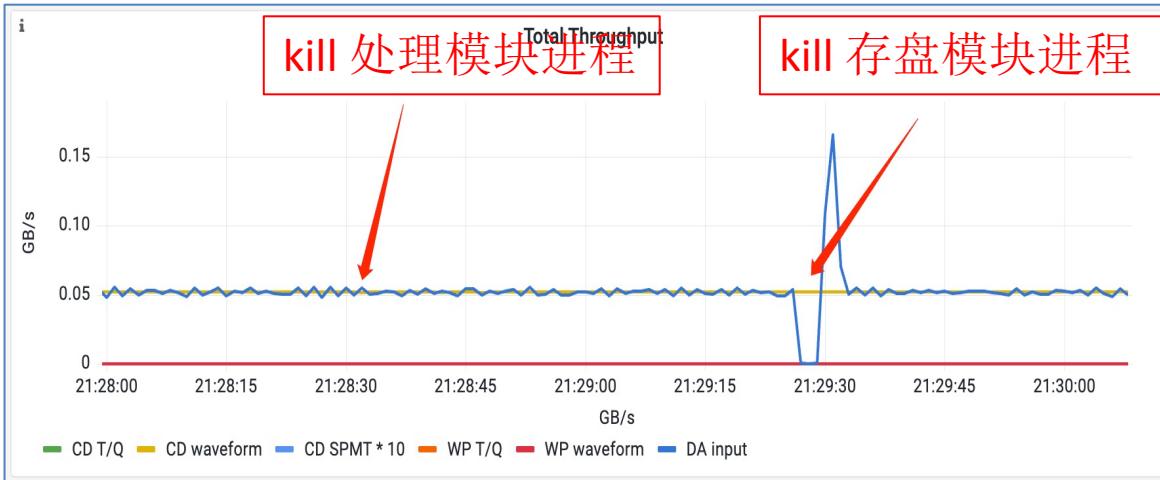
Booted: 注册回掉函数监听其他节点的信息

Running: 调整zmq接收线程在所有对象初始化后启动

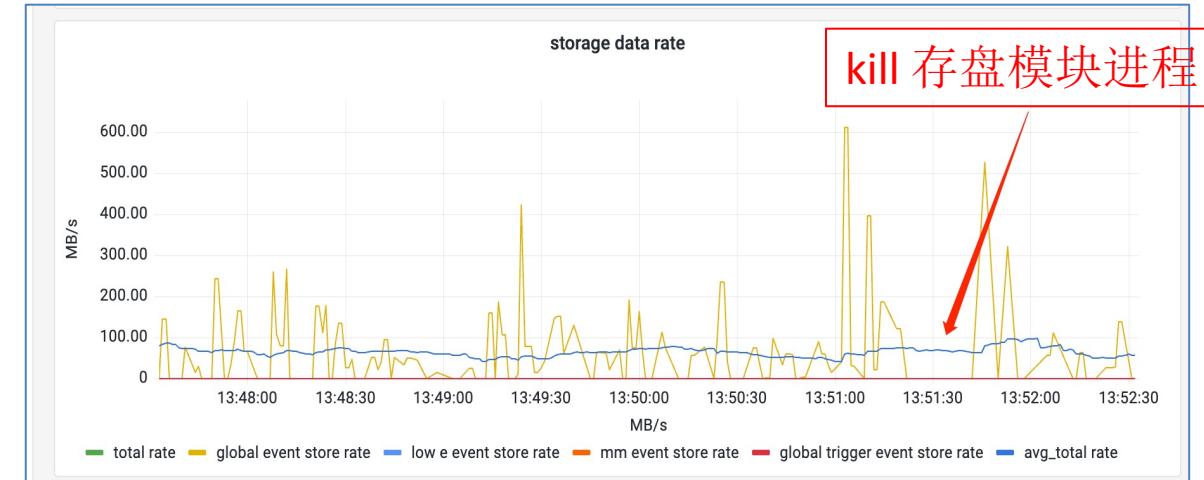


# 数据流的高可用升级

## 模拟数据源测试



## 真实电子学测试



单节点存盘

处理模块

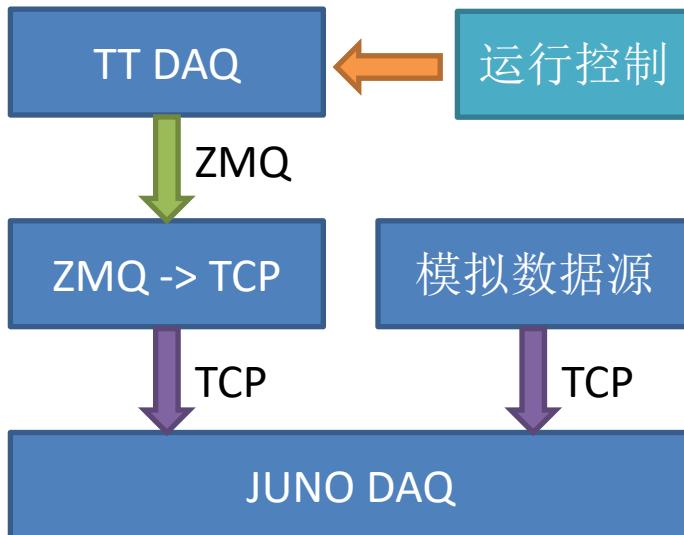
存盘模块/stream1

存盘模块/stream2

在模拟数据源测试和真实的电子学测试中，手动kill掉数据流的进程后，k8s会重启相应的pod，数据流仍然能够正常运行。

- 1、由于存盘模块是单节点存盘，故障出现后数据流会出现卡顿。
- 2、如果存盘模块所在的节点发生故障，k8s可能无法重启。

# TT探测器的集成



ZMQ转TCP

TT DAQ的数据格式为ZMQ  
ROS使用TCP连接读取数据  
考虑在ROS中支持ZMQ

## ROS插件的开发

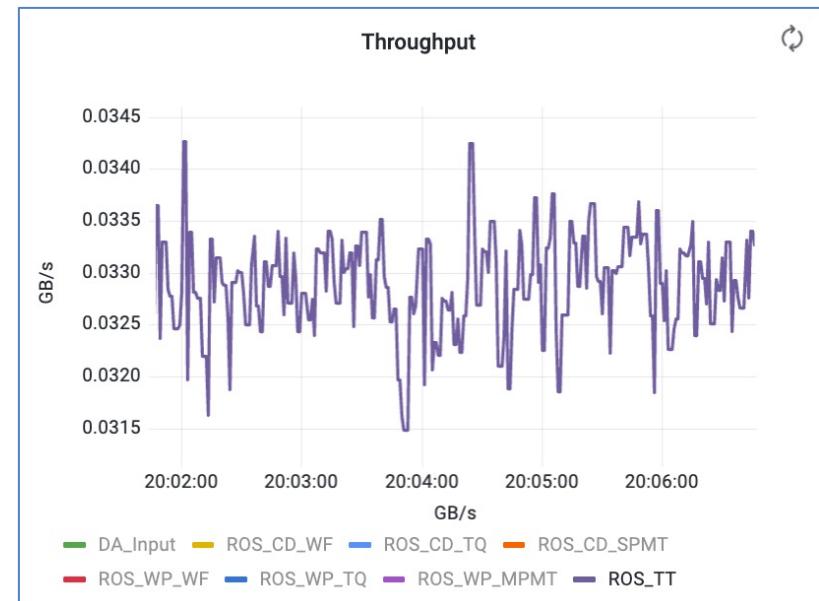
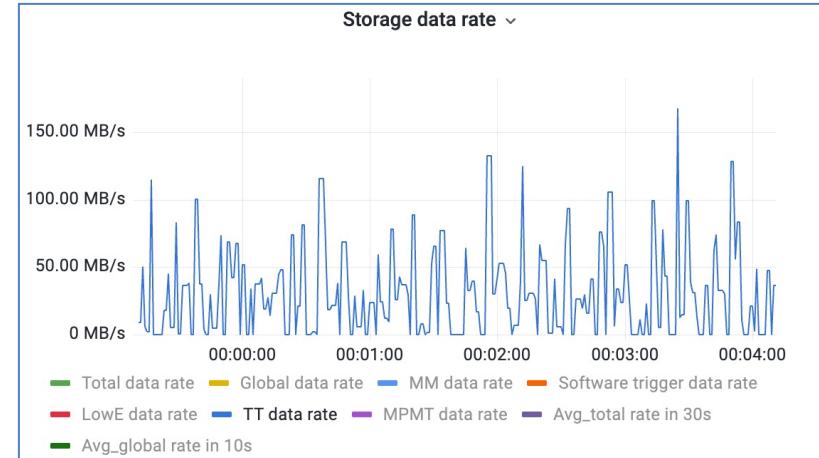
新ROS使用插件对不同类型的数据进行预处理  
开发了TT的预处理插件  
在预处理中对TT的数据进行检查

## 模拟数据源

使用基于配置的模拟数据源，用于调试

## 运行控制

通过HTTP对TT进行自动控制





# **Thank You for Your Attention!**