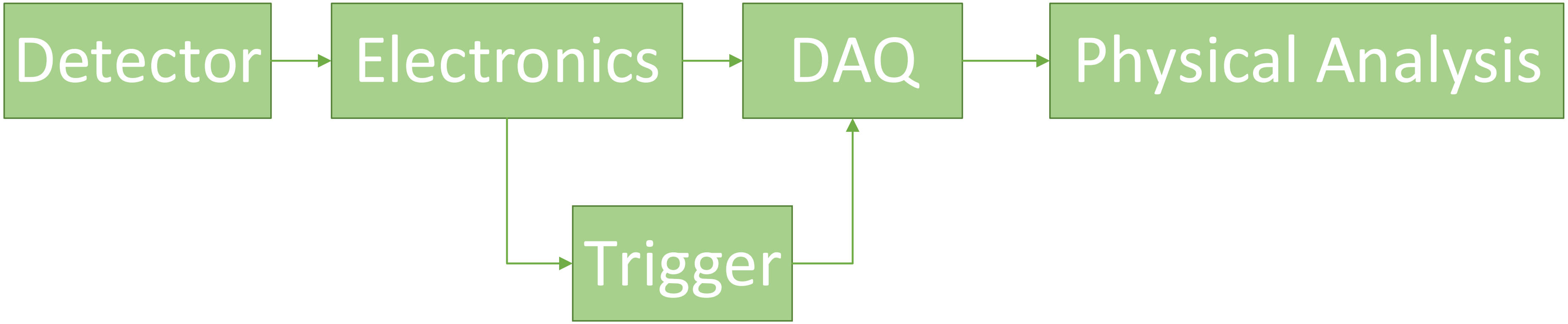
Brief Introduction of TaskRouter

Lu Xiaoxu
2017 03 23

Brief Introduction



Basic Framework of a High Energy Physical Experiment

DAQ Responsibilities:

Software Trigger

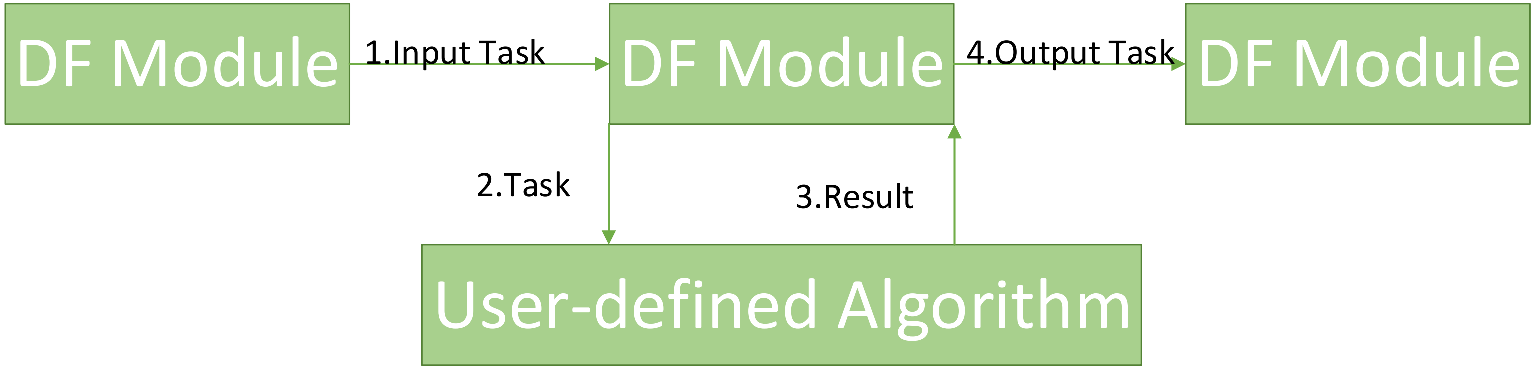
Data Acquisition

Data quality monitoring

Data compression

System mornitoring

Brief Introduction


DF Module

1.Input Task

DF Module

4.Output Task

DF Module

2.Task

3.Result

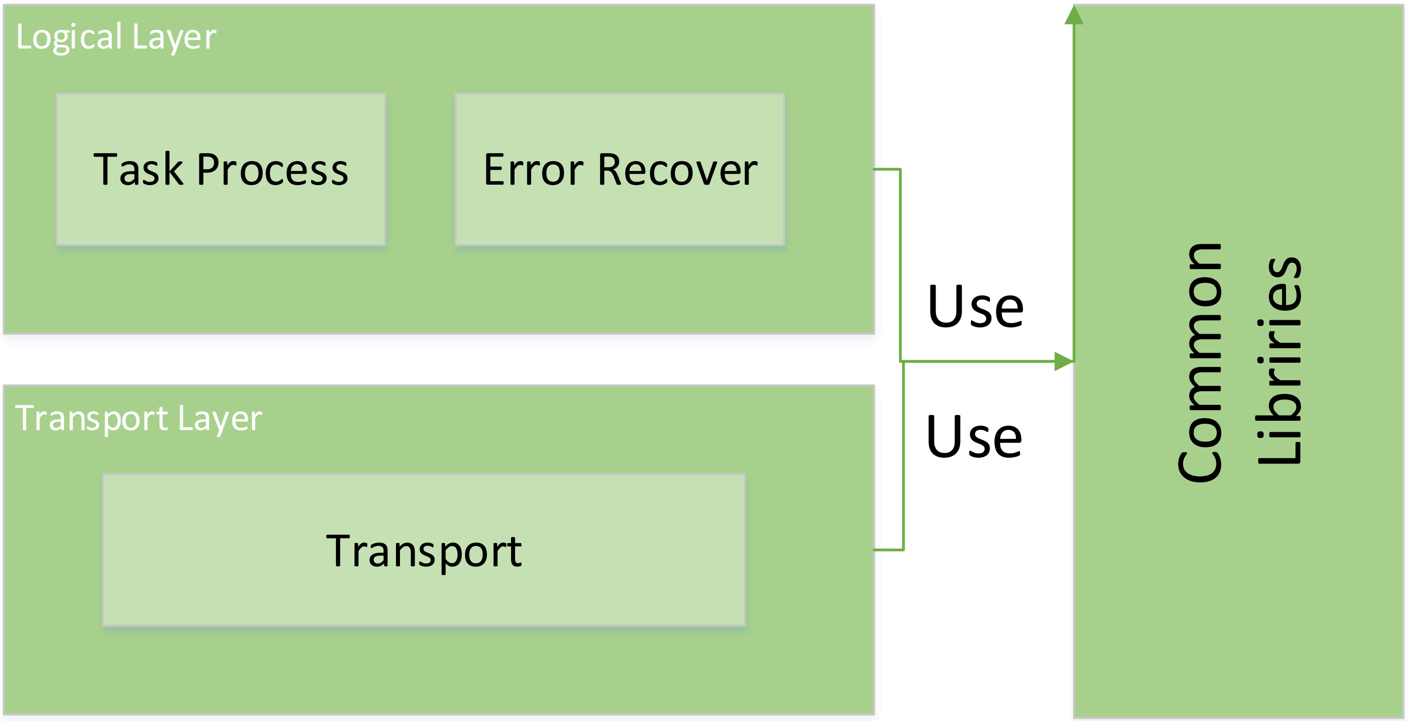
User-defined Algorithm

Basic Function of TaskRouter

Main challenges

1. Deployment and monitoring of DF modules
2. Load balance when using distributed architecture
3. Failures detection and recovery
4. Design of processing procedure
5. Implementation of processing algorithm

Basic Design of TaskRouter



TaskRouter Layers and Components

Details of TaskRouter

Transport Component

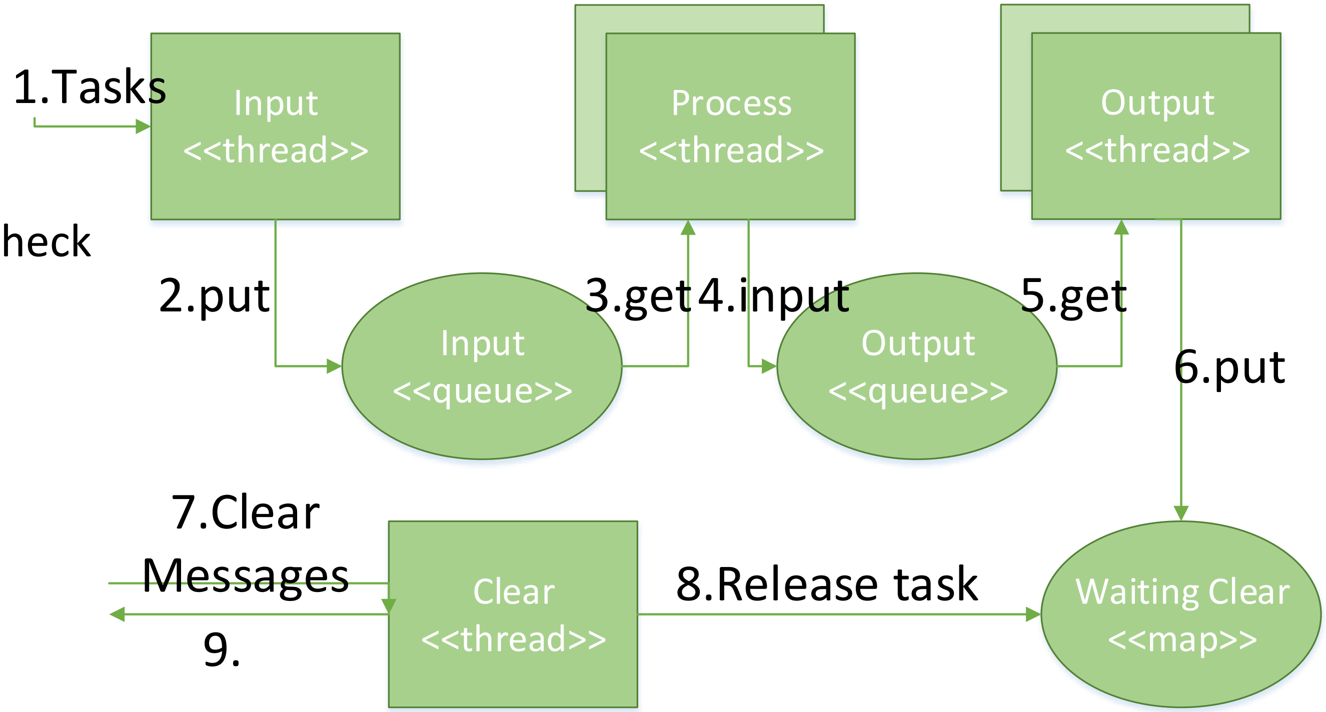
1.Initialize

2.wait for connections & check a connection pool

3.receive data

4.check timeouts & throw exceptions

Task Process Component



Working Threads in the Task Process Component

Details of TaskRouter

PU Classification and Error Recovery Component

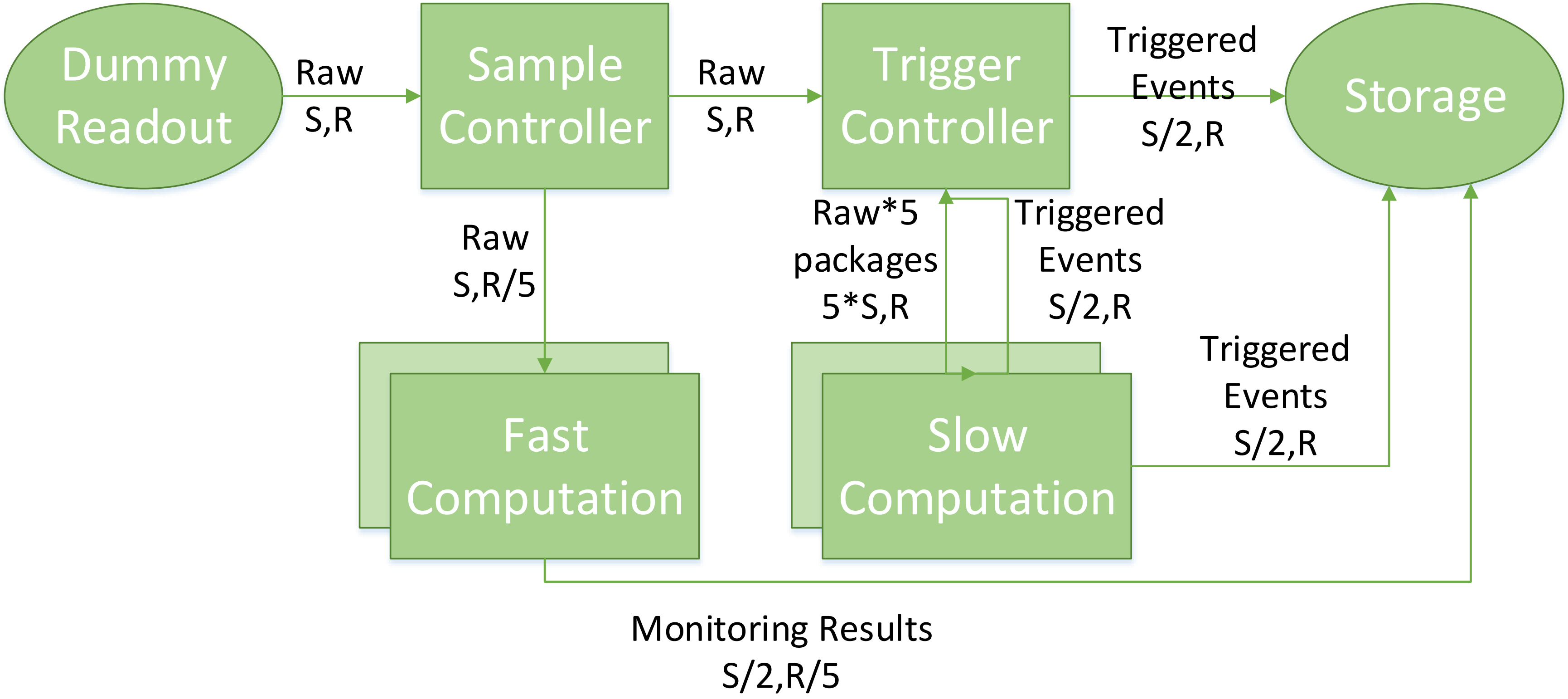
Dispatch PU: for tasks that cannot be processed on different nodes
Compute PU: for tasks that can be done in parallel

Way to detector error: ping message does not arrive within a timeout cycle

Error recovery methodology:
Previous-level Pus check tasks in the Waiting Clear buffer. Put failed task back to the Output Queue to be reassigned.

Common Library
Configuration
Pre-allocated buffer management
Shared Queue

A Prototype



Data flow of an online processing prototype

Test Results

		Output Data Rate(GB/s)	Dt^2 Avg(ms)	DT Max(ms)
Case 1	TC1→SC	1.2	2.9	14.1
	SC2→TC	0.12	1.6	14.8
Case 2	TC1→SC	1.4	46.7	97.0
	SC2→TC	0.14	5.4	22.6

Throughput and Dispatch Delay

Summary

The core design of TaskRouter has been done. This framework can be used for development of flow based online processing. Users can define mostly what they need according to their demands, and things left will be done by the framework.

A prototype online processing software has been developed based on it, and that software works well, with over 1GB/S as the throughput for a single PU, while the dispatch time is under control. Besides, this software can automatically recover from any single-point failures with no data lost.

Still more work, like run control and etc., need to be done in the future.

THANK YOU!