

protoDUNE Raw Data Organization

Kurt Biery

DUNE DAQ Meeting

29 April 2019

Introduction

Roland did some preliminary testing of long TPC readout windows (e.g. ~ 1 sec) at PDSP. I did some additional testing to try to break the data into manageable-size chunks.

- “manageable” == storable in ROOT branches + avoiding limitations on the size of data that can be loaded into memory for offline

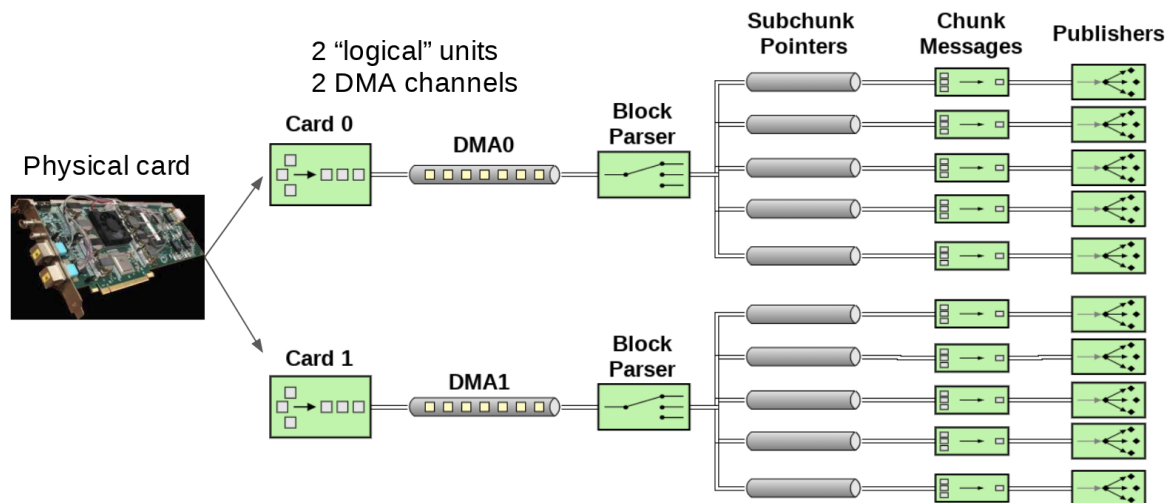
A few of us have had some initial discussions of how to move forward in this area.

The purpose of this talk is to share some of the background information and discuss a couple of details.

Description

At protoDUNE SP, a representative slice of the system has...

- One APA; 5 WIBs; one FELIX card; 10 links; 10 *artdaq* BoardReaders; 10 data fragments per trigger/event



All six APAs:

Begin processing the 2nd record. run: 7613 subRun: 1 event: 61 at 22-Apr-2019 18:32:34 CEST

PRINCIPAL TYPE: Event

PROCESS NAME	MODULE_LABEL..	PRODUCT INSTANCE NAME	DATA PRODUCT TYPE.....	PRODUCT FRIENDLY TYPE	SIZE
DAQ.....	daq.....	TIMING.....	std::vector<artdaq::Fragment>	artdaq::Fragments....	...1
DAQ.....	daq.....	ContainerFELIX.....	std::vector<artdaq::Fragment>	artdaq::Fragments....	..20
DAQ.....	daq.....	ContainerTPC.....	std::vector<artdaq::Fragment>	artdaq::Fragments....	..40
DAQ.....	TriggerResults	art::TriggerResults.....	art::TriggerResults..	...-
DAQ.....	daq.....	ContainerCTB.....	std::vector<artdaq::Fragment>	artdaq::Fragments....	...1

More description

In Roland's 1-sec readout window test, each data fragment was approximately 900 MB in size...

- So, the data size from a full APA would be around 9 GB
- The first issue was that we couldn't store more than one fragment per ROOT branch. (Each ROOT branch is limited to 1 GB)

Begin processing the 2nd record. run: 7613 subRun: 1 event: 61 at 22-Apr-2019 18:32:34 CEST

PRINCIPAL TYPE: Event

PROCESS NAME	MODULE_LABEL..	PRODUCT INSTANCE NAME	DATA PRODUCT TYPE.....	PRODUCT FRIENDLY TYPE	SIZE
DAQ.....	daq.....	TIMING.....	std::vector<artdaq::Fragment>	artdaq::Fragments....	...1
DAQ.....	daq.....	ContainerFELIX.....	std::vector<artdaq::Fragment>	artdaq::Fragments....	..20
DAQ.....	daq.....	ContainerTPC.....	std::vector<artdaq::Fragment>	artdaq::Fragments....	..40
DAQ.....	TriggerResults	art::TriggerResults.....	art::TriggerResults..	...-
DAQ.....	daq.....	ContainerCTB.....	std::vector<artdaq::Fragment>	artdaq::Fragments....	...1

One ROOT branch, 2 FELIX-based APAs, would be ~18 GB for a 1-second readout window

Still more description

Offline processing of the data has reasons for further limiting the size of each ROOT branch (Heidi has mentioned a desired max size of 200 MB).

This implies

- Separating the data products in the event geographically, for sure
- Providing additional separation if we ever want to read out TPC windows longer than ~200 msec

Discussions so far

Proposals:

- Break the problem down into two steps: geographic separation (fairly easy to do now) and time-interval separation (requires more work).
- Bundle the data appropriately in the FELIX+BoardReader part of the system so that the EventBuilders only have to assign the appropriate Product Instance Names.
- Use information in the WIB headers to provide the keys to the geographic mapping between data fragment and appropriate Product Instance Name.

Short-term geographic splitting?

Tom and others have suggested a first step of splitting the data by APA:

- ProductInstanceName = “ContainerFELIX<%03d; APA_number>”
- This would only support windows of ~20 msec

Do we want to go further than this right away?

- Giovanna pointed out that a suffix based on APA (crate) number [%03d], slot number [%d], and link number [%d] is straightforward to do now
- This would support windows of ~200 msec

(I’ve prototyped code changes that support a range of options that include both of these.)

Variable numbers of data products

How much variability in the Product Instance Names do we want to support? (I believe that Tom has prototyped initial ideas for handling split data...)

As we think ahead to adding time-based splitting, the question of variable number of data products event-by-event, within a run, comes up.

Will this be a problem for offline?

Time-based splitting?

Proposal: base this on a configurable time interval

How should the interval be represented in the Product Instance Name?

This will need code changes in the FELIX and *artdaq* parts of the system...

Next steps?

Agree on plan for geographic splitting today.

Test that plan soon at protoDUNE.

What forum should we use for further discussion of time-based splitting at protoDUNE?

Schedule a discussion of how to handle long readout windows for DUNE? Will this be part of the 3 day working meeting this summer to work out the DUNE data model?