Software Plan

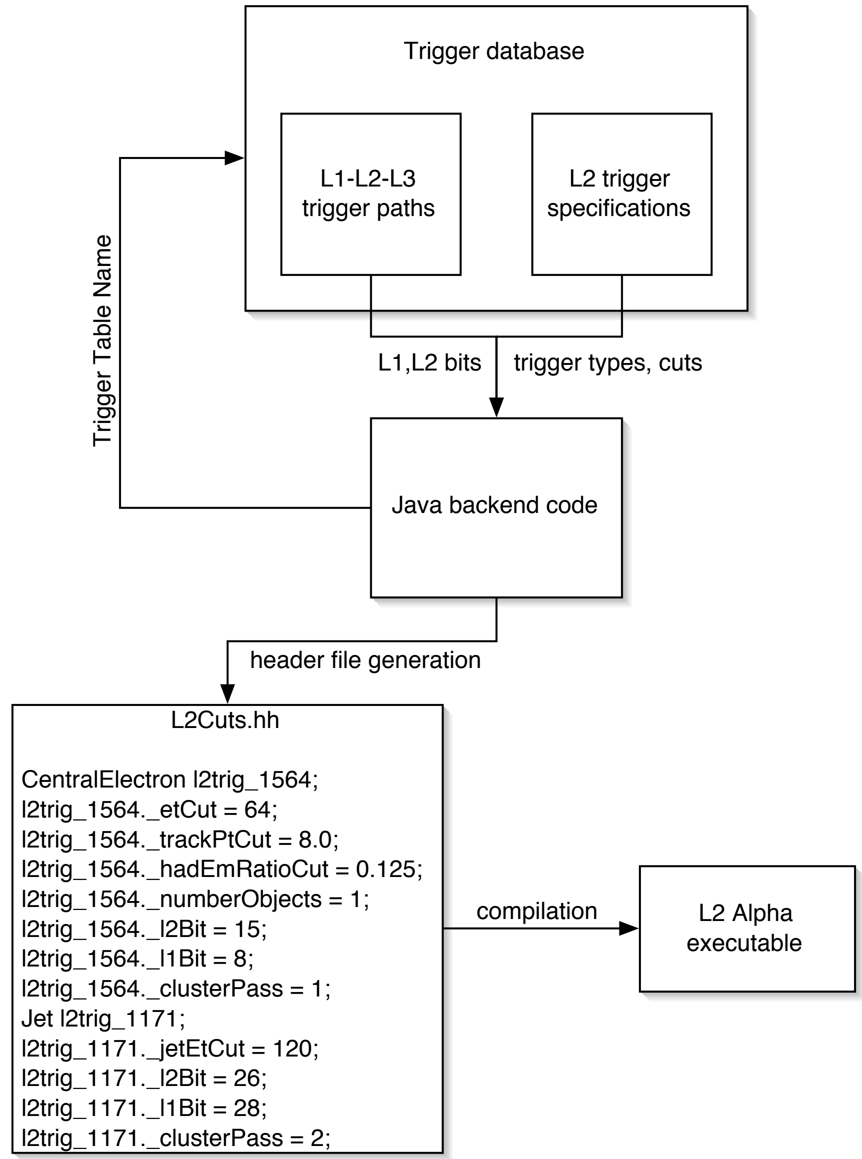
Thomas Wright
University of Michigan

Level 2 Review
August 1, 2002

- Level 2 Software Overview
- Timing Measurements
- Near-Term Improvements
- Projecting to 4E32
- Conclusions

Level 2 Software Overview

- The Level 2 Alpha executable is built specifically for each trigger table
- The trigger database is used to generate header files for each table
- Trigger Algorithm code is written in C++
  - Can run algorithms in offline
  - Easily portable to new architectures



Level 2 Alpha Executable

The Level 2 Alpha executable is an infinite loop:

- Wait for data to finish loading
- Check for next L1A
- Configure DMA, assert START_LOAD for next event
- Unpack data
- Run trigger algorithms
- Error checking
- Send decision to TS
- Wait for TS global decision
- If L2A, build TL2D bank
- Finish TS handshake
- Back to the top

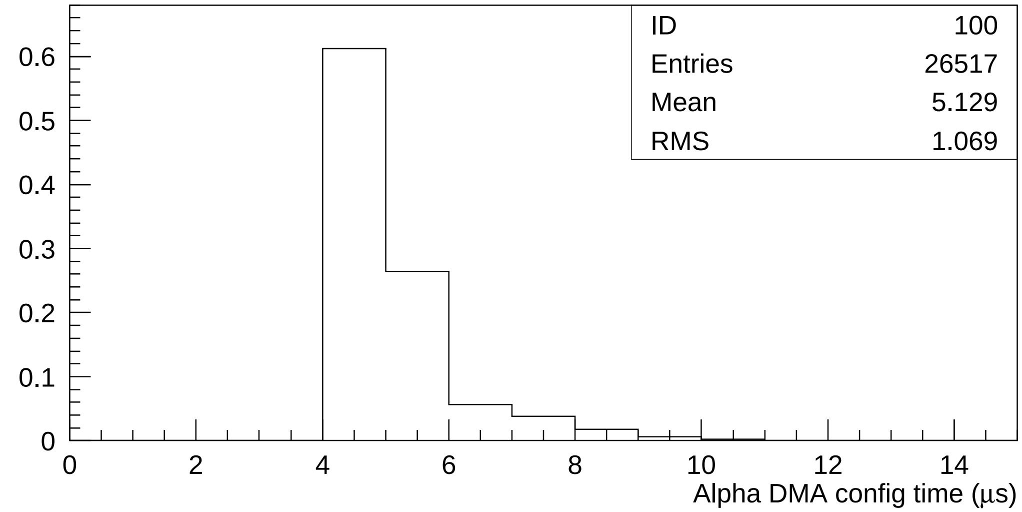
Timing Measurements

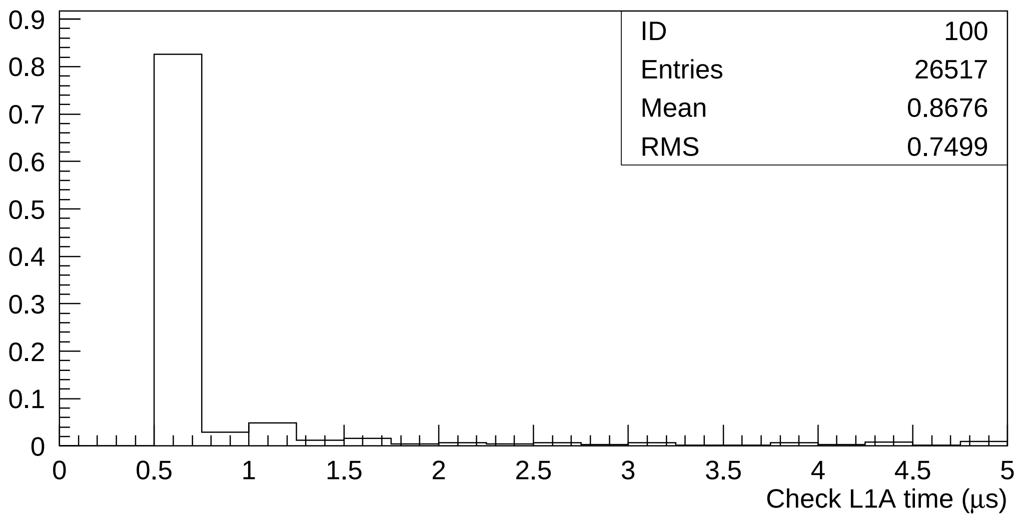
- Timing measurements are made during real running, using the 500 MHz Alpha timer
- To obtain unbaised measurements, a two-stage buffer is used so that on each L2A, the timings for the current and previous L1A are saved
- Also buffered are the L1 bits and the numbers of trigger objects from the previous L1A
- For results in this talk, run 148648 with trigger table PHYSICS_1_02_v-1 was used
- For minimum-bias occupancies, runs using PHYSICS_1_01_v-7 and which had all detector subsystems active were used

Timing Results

Alpha DMA config

- Set up addresses for DMA transfers
- Read amount of data sent
- Reduce by $\sim$3 μs by eliminating read





Check for L1A

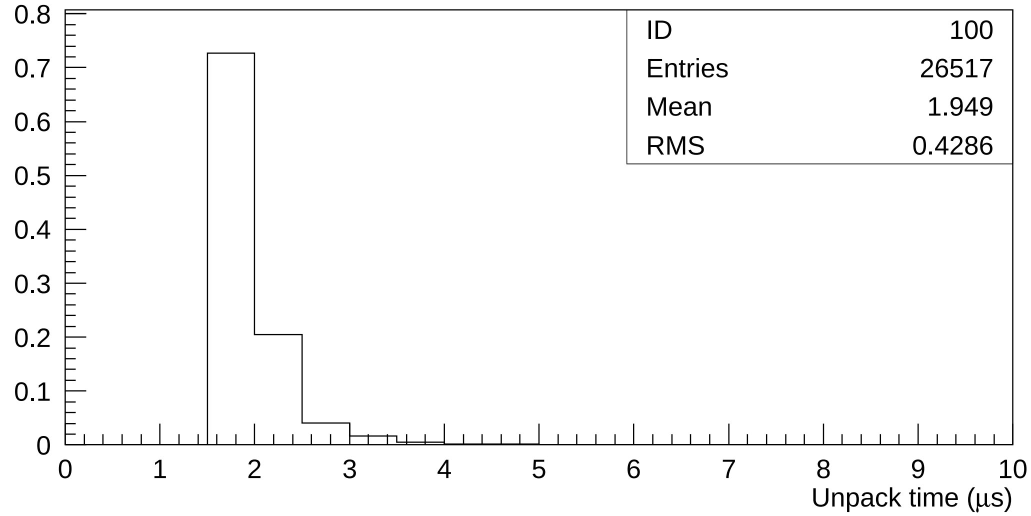
- Read PCI register
- Assert START_LOAD

These operations cannot be performed in parallel with data loading

Timing Results

Unpack data

- L1 bits
- L1 scalers

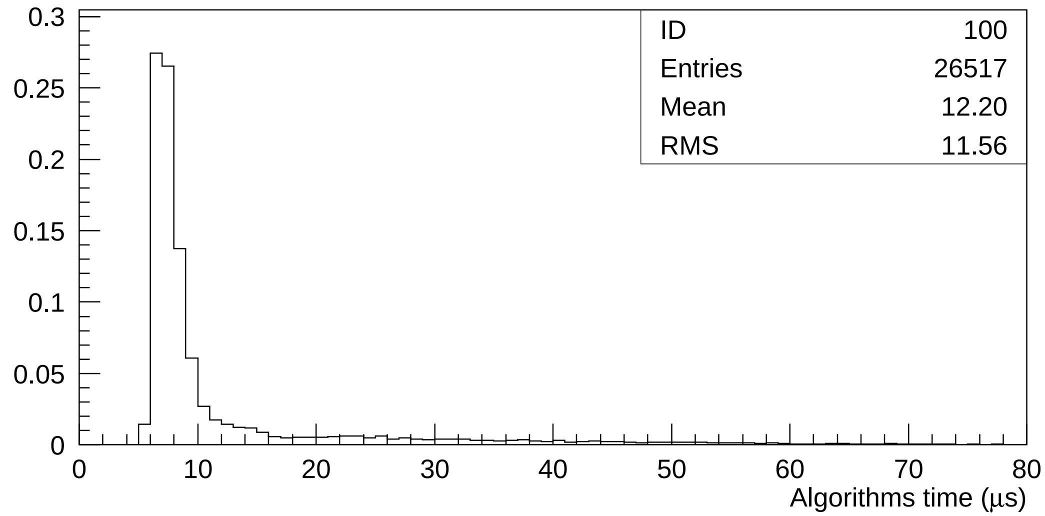


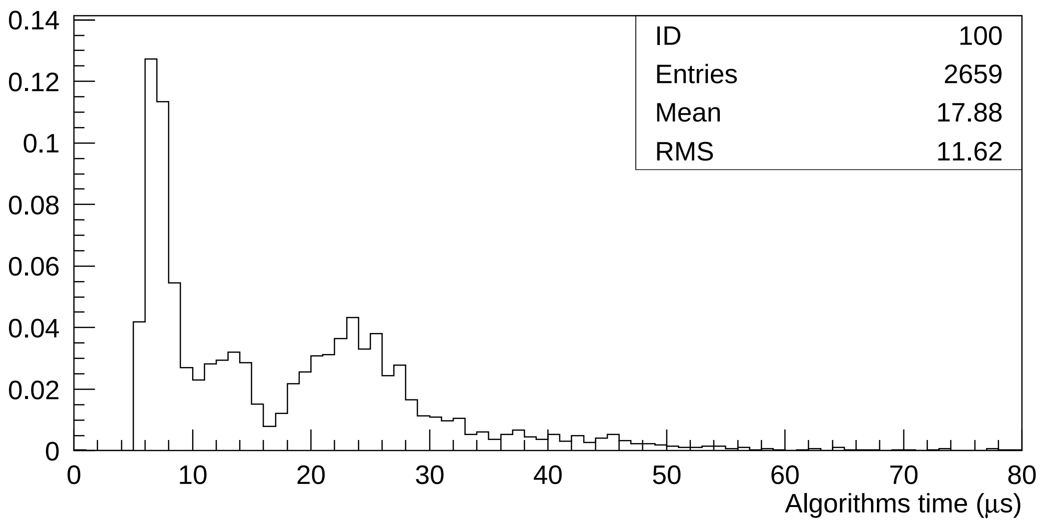
- In the past we always unpacked all of the XFT tracks, SVT tracks, and clusters
- Now only unpack each "on demand", if we run an L2 trigger than needs it
- This saves lots of time on two-track triggers - only need to unpack SVT block
- At high luminosity this will not be as big an effect
- Keep in mind when interpreting algorithm timings that they including unpacking whatever data are needed

Timing Results

Run algorithms

- without compiler optimization, average was ∼40 μs
- threshold can be reduced by 1-2 μs by relocating L1 prerequisite check





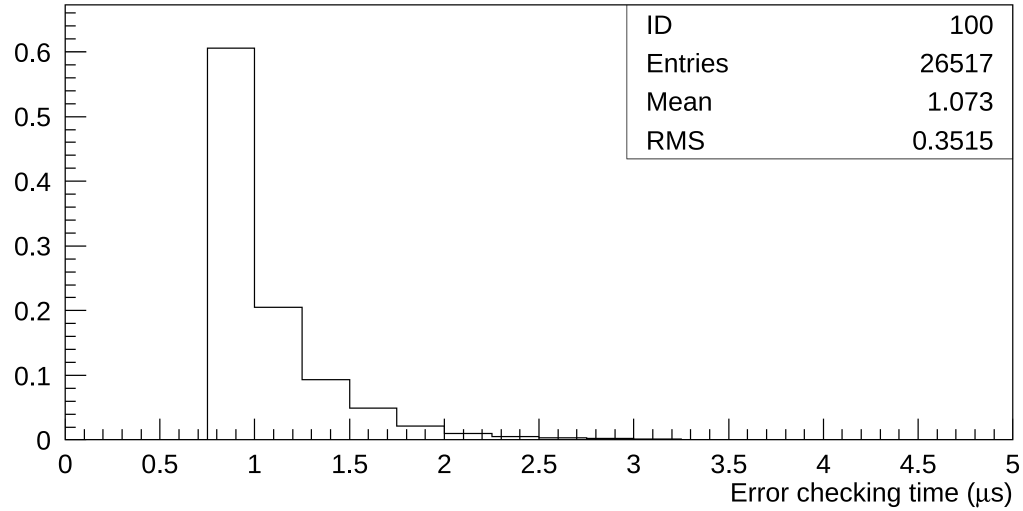
Here is how it would look without:

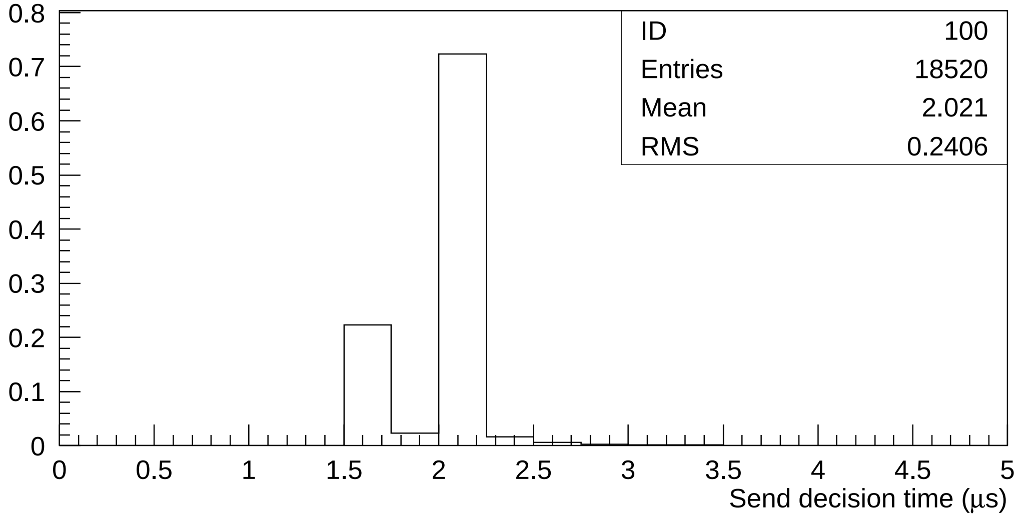
- two-track
- $J/\psi \rightarrow e^+e^-$
- diffractive
- 3 GeV track

Timing Results

Error checking

- Check # of L1 words
- Check buffer numbers





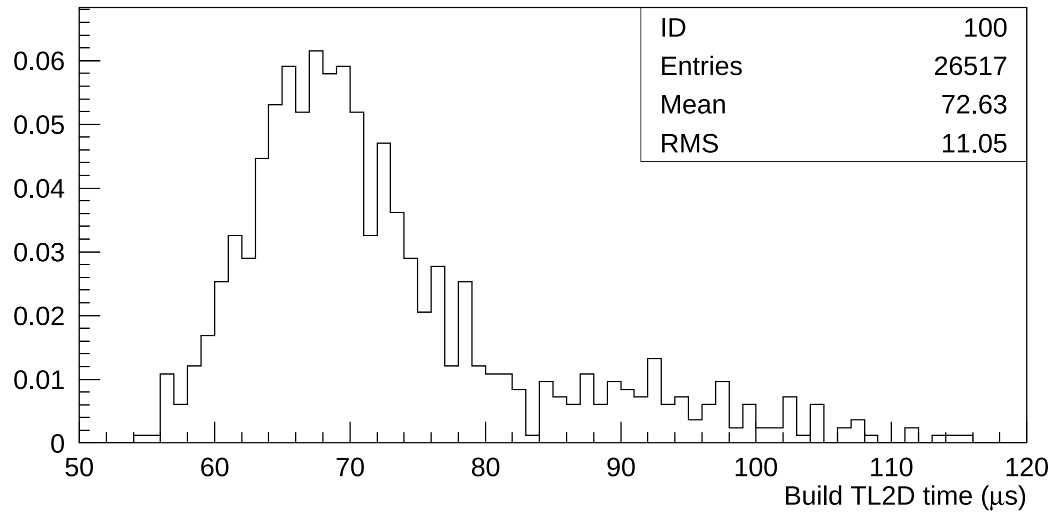
Send decision

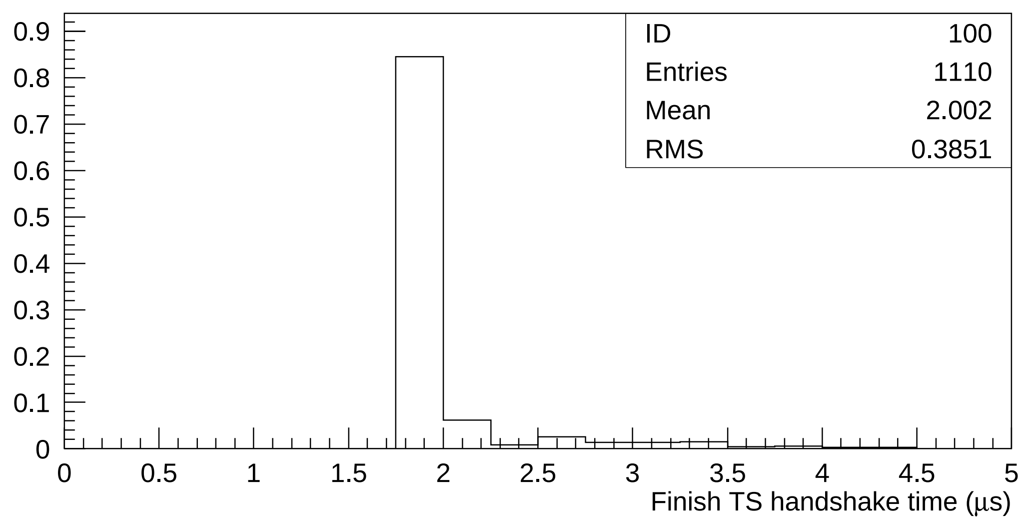
- Send decision to TS
- Wait for global L2 decision
- Not understood why this takes 2 μs

Timing Results

Build TL2D

- Only on GL2A
- Was ~120 μs without compiler optimization
- ~50 μs is RECES readout, can save 0.5% deadtime



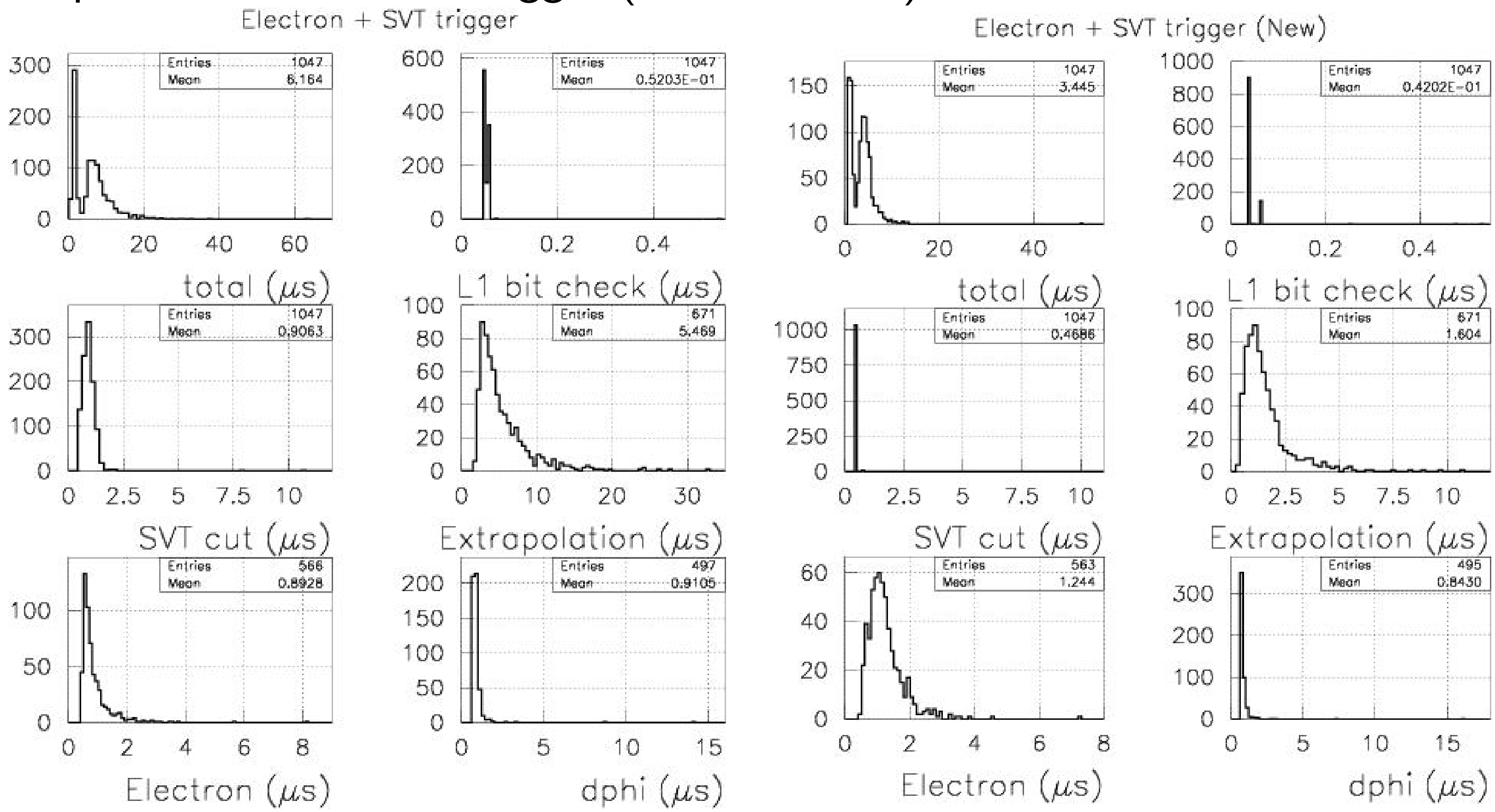


Finish TS handshake

- Also not understood where the 2 μs is going

Trigger Algorithm Optimization

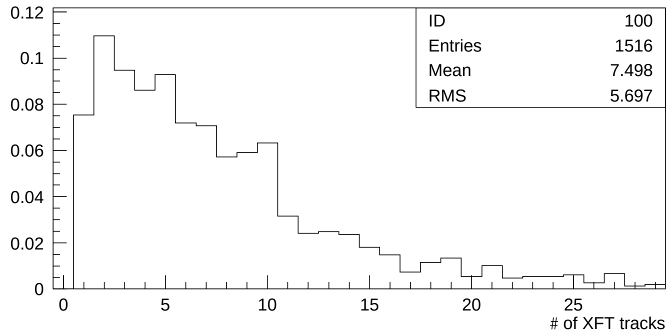
Example: Electron+SVT trigger (Masa Tanaka)

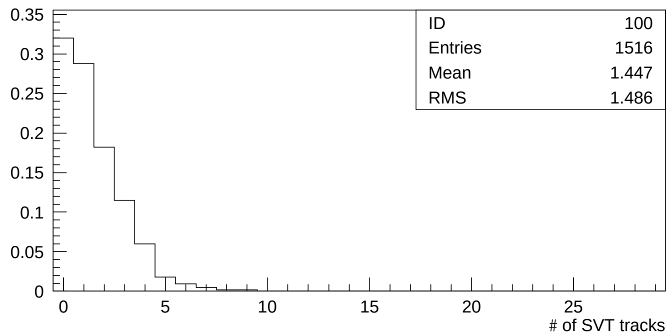


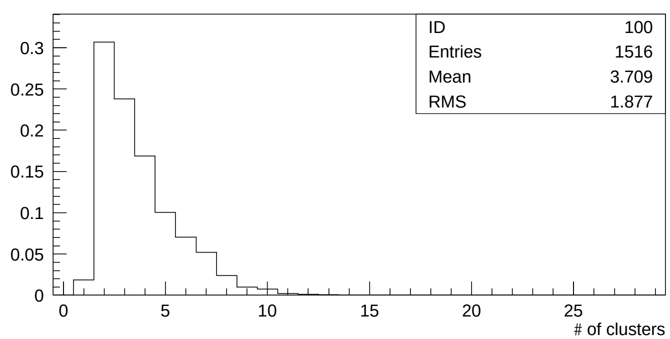
- Not much effort into this so far
- Can do a much better job caching intermediate results
- Difficult to say exactly how much gain is possible

Projecting to 4E32

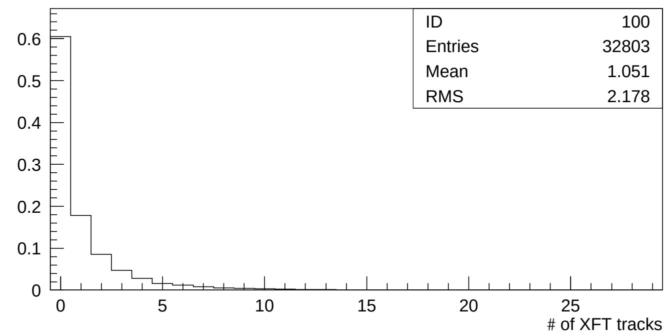
CEM4_PT4 events

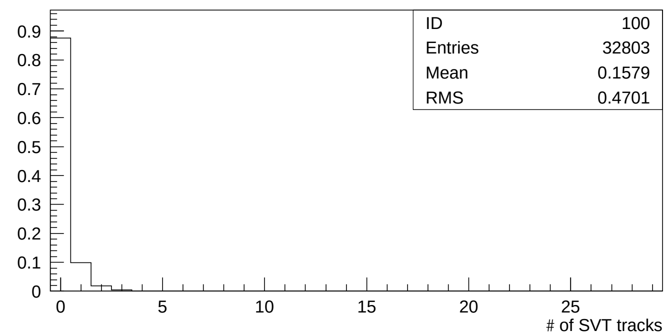


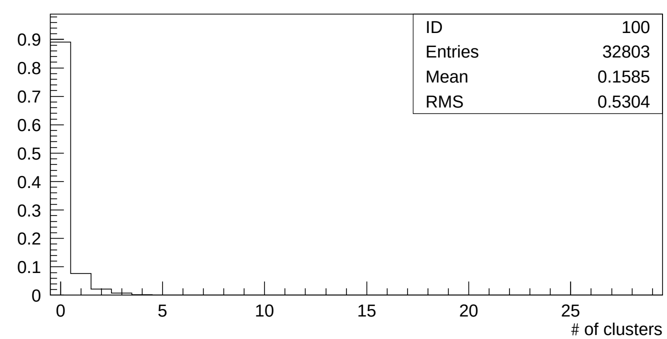




Minimum bias (CLC) events

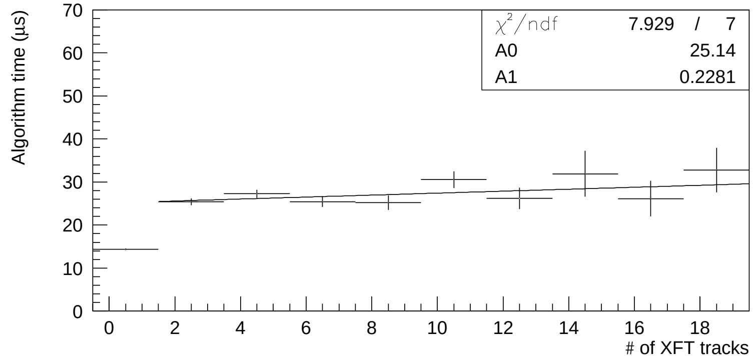


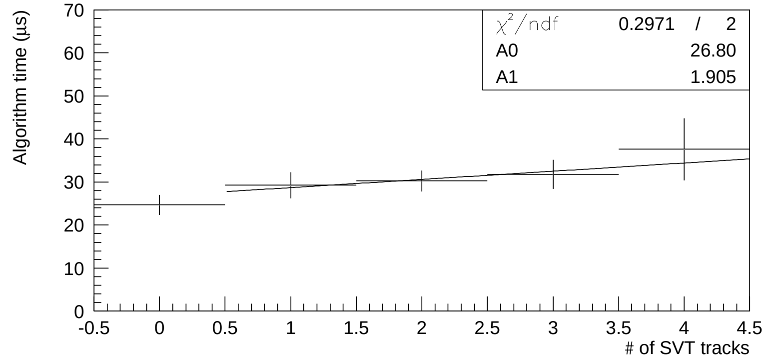


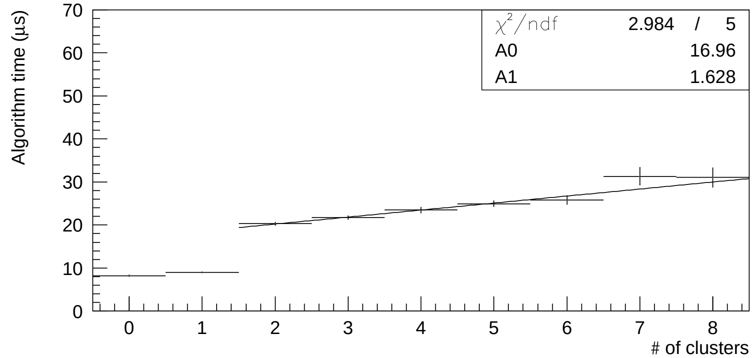


Projecting to 4E32

How much does each additional trigger object add to processing time?







Results:

- 0.23 μs per XFT track
- 1.9 μs per SVT track
- 1.6 μs per cluster

- Removed the same L1 triggers as for previous algorithm timing plot
- These times include unpacking as well as algorithm processing time
- Only possible for events where that object block is unpacked
- This procedure is therefore biased, since we don't always unpack everything

Projecting to 4E32

At 4E32, with one high-pt and 10 minimum-bias events per crossing:

	# additional objects	time per (μs)	total added time (μs)
XFT tracks	11	(0.23)(0.5)	1.3
SVT tracks	1.6	(1.9)(0.5)	1.5
clusters	1.6	1.6	2.6

- Factors of 0.5 account for not always unpacking those blocks
- This does not include effects of increased XFT fake rate at high $\mathcal{L}$ or correlations between the numbers of trigger objects
- For base algorithm time, use the 17.9 μs seen earlier, minus $\sim$4 μs for projected near-term improvements
- Expected algorithm time at 4E32 is then $14 + 1.3 + 1.5 + 2.6 \simeq 20$ μs

Conclusion

Putting it all together (all times in μs):

	current	near-future	4E32
Alpha DMA config	5.2	2.2	2.2
check L1A	0.5	0	0
unpack	1.9	0.5	0.5
algorithms	12.2	9	$\sim$20
error checking	1.1	1.1	1.1
send decision	2	2	2
finish TS handshake	2	0	0
total	24.9	14.8	$\sim$26

- Getting to 15 μs in the near future is a realistic goal
- Maintaining that performance as luminosity increases does not seem possible with current single-Alpha configuration
- Going to multiple Alphas would get back into the 10-15 μs range, but without very much headroom