



# L2 latency, SVT fast abort and some related issues/puzzles

---

Ted Liu, April 21, 2005, Trigger Hardware

This meeting is for Cheng-Ju (outgoing SPL)  
& Vadim (incoming SPL)

Both have done a wonderful job for CDF trigger!

What I really wanted to do is to give a talk about both of them, the wonderful things they have done, but that would take more than one hour...

and I only have 10 mins...☹

So far four trigger SPLs worked on Pulsar:

Peter Wittich, Cheng-Ju, Roberto (SVT RW) and now Vadim

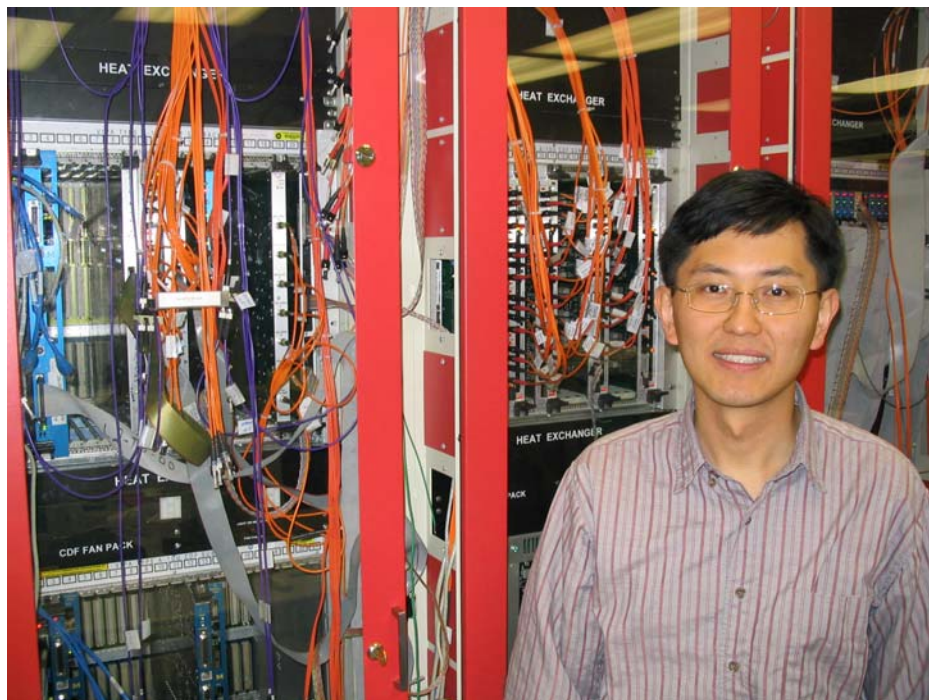


# Cheng-Ju & Vadim

System  
Commission SPL



Performance  
Optimization SPL



Photos taken 1 year ago...when the actual real work for L2 upgrade really started (after building teststand tools, playing with RunIIA muon etc) ...



## Comments: I have only given a few talks at this meeting in the past 5 years...

---

- Looking back, the talks I gave were mostly ONLY about why&how we build a teststand tool for L2...
- later on, some accused me of not telling the whole truth about what Pulsar could possibly do ...

- But so far Pulsars are used mostly as teststand tools  
→ today there are six different Pulsar teststands at different locations: B0/WH/FCC,UC/UIUC/Pisa.

And there are more Pulsars used in teststands than in the system!...(though astrophysicists will use them to study the real heavenly Pulsars → Magic experiment)

Without the teststand capability, there would be no new L2 !

& I was telling the whole truth (essence) about Pulsar back then...☺



# What this talk is about

---

- now Pulsar is in full operation, it is time to carefully study overall L2 performance, and see what we can improve, near term and long term
- This is just an initial study (so far, only ~ one day real work), to show how Pulsar system timing measurements could be useful in understanding/improving the system performance
- Already found **three puzzles** and we know what to do next to understand them and to improve ...
- New SPL will lead the effort to solve the puzzles...

Many slides are for backups (try to put useful info in one place)

Will only use half of the slides due to limited time...



## Introduction

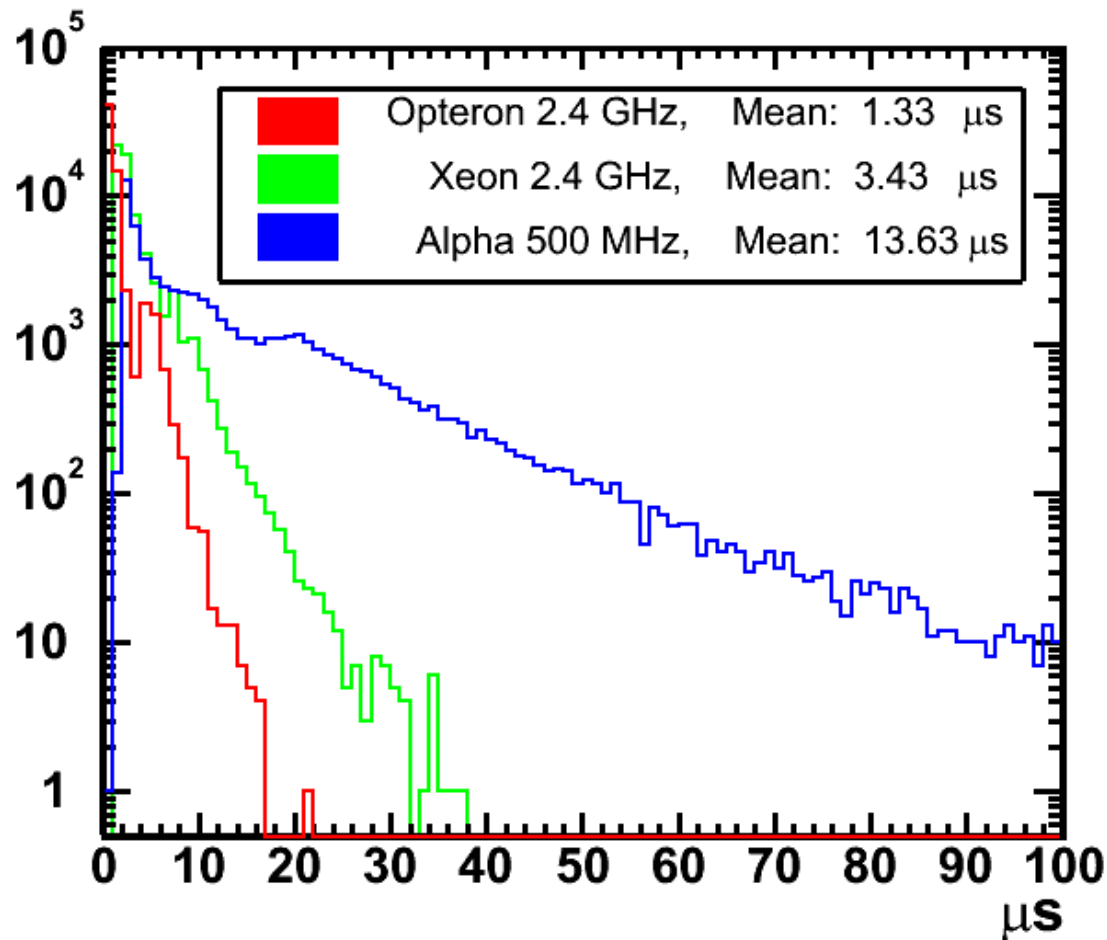
---

- At low luminosity, current overall L2 latency is dominated by SVT processing time with standard trigger table (main motivation to upgrade SVT).
- At high luminosity, with highlumi trigger table, a large fraction of triggers do not need SVT.
- The raw speed of Pulsar L2 system is faster than Alpha, one would naively expect Pulsar system should benefit more from SVT fast abort than Alpha...etc → this was what Roberto asked me to talk about initially... but end up more ...



# Pulsar CPU vs Alpha algorithm timing

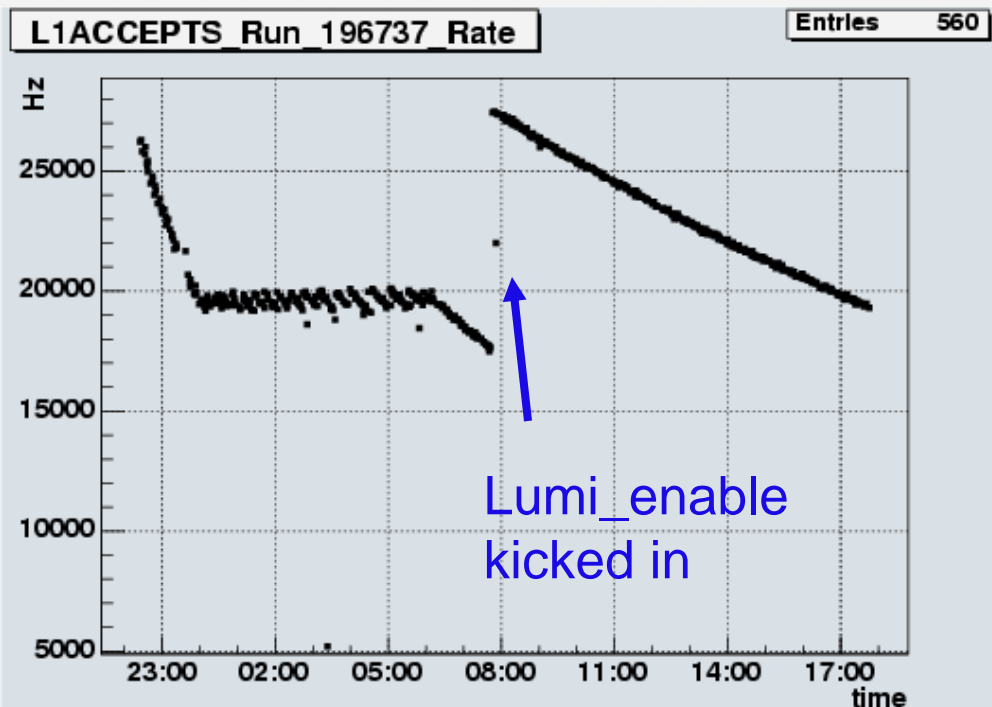
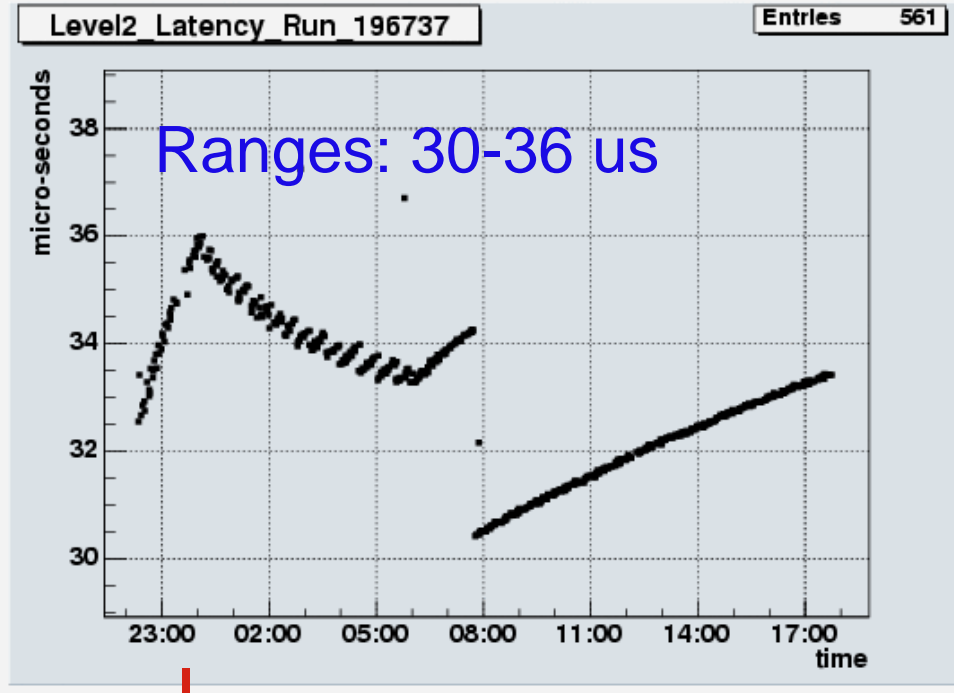
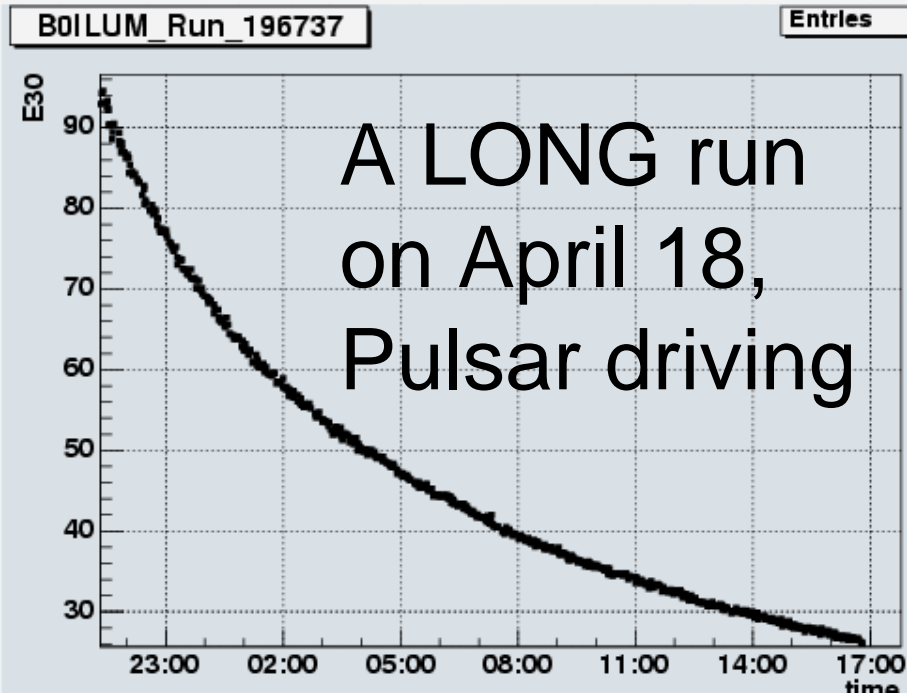
Algorithm Times : Opteron, Xeon & Alpha



- Early study by Kristian etc with real data

- Pulsar  $\sim 1.3 \mu\text{s}$
- Alpha  $\sim 13.6 \mu\text{s}$

A factor of 10 faster



Some Basics:  
Online average L2 latency is  
L1A rate/trigger mix/luminosity  
dependent...

think of it as an “effective latency”  
for a given L1A rate and lumi.  
Just Keep this in mind...

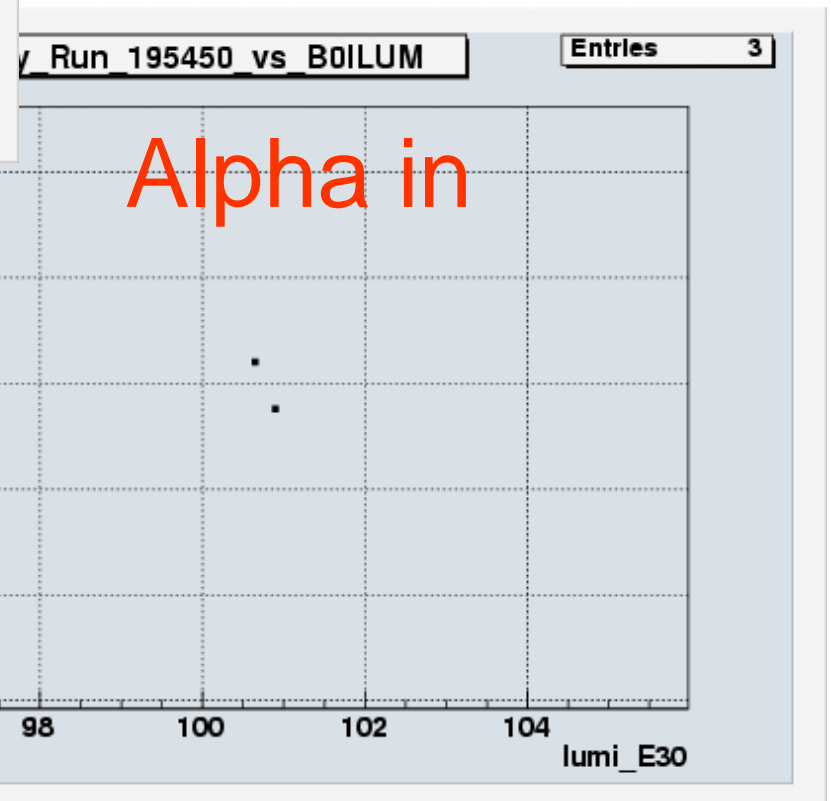
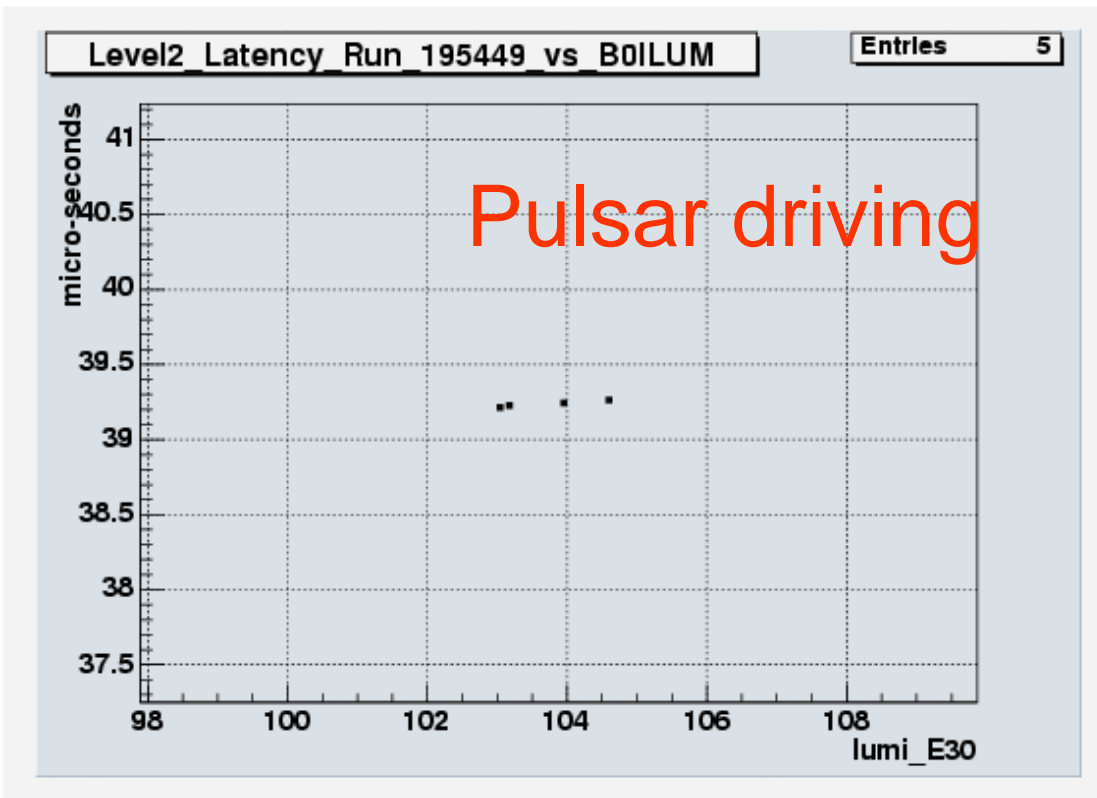


## Timing measurements on March 18

- At high luminosity  $\sim 100 \text{ e}30$ , with highlumi trigger table: PHYSICS\_HIGHLUM\_3\_00[7,613,556], with silicon
- SVT fast abort was enabled

Run	L1A(kHz)	L2A(Hz)	DT(%)	L2 online latency(us)
195449	8.9	280	0.5	39 Pulsar driving
195450	8.5	270	1.3	48 Alpha driving
195451	8.1	275	0.5	39 Pulsar driving

On average, Pulsar is  $\sim 9 \text{ us}$  faster than alpha at high lumi and with High lumi trigger table

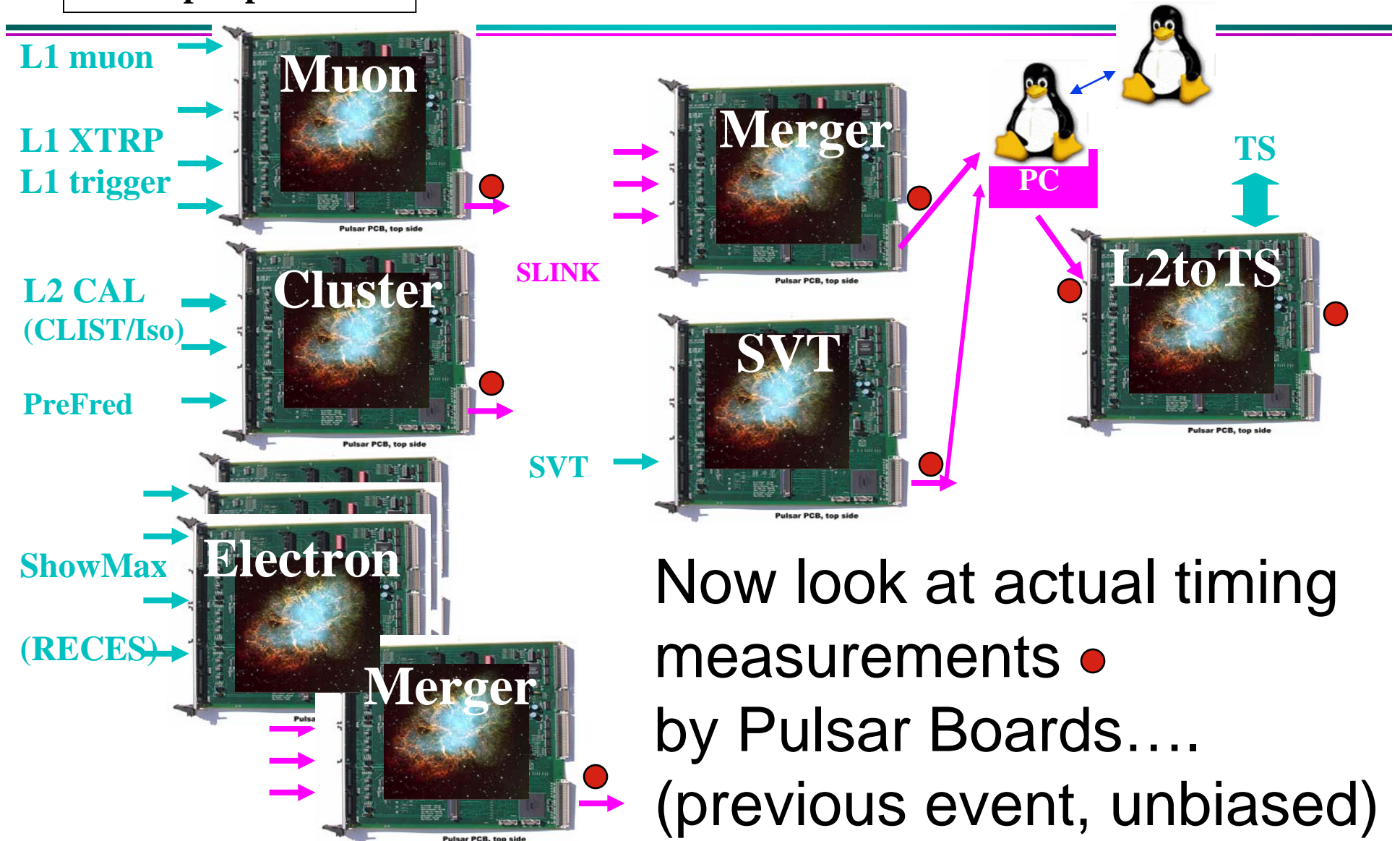


Some details  
from online  
plots

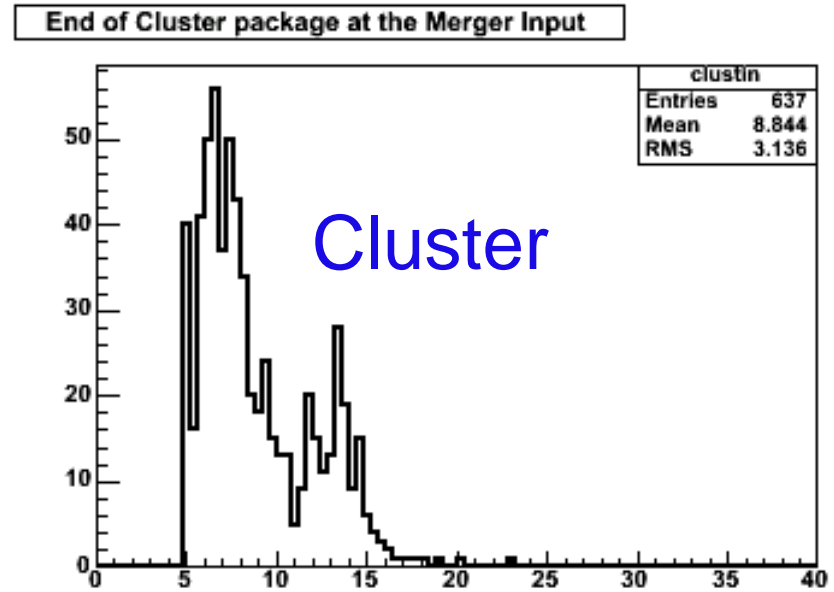
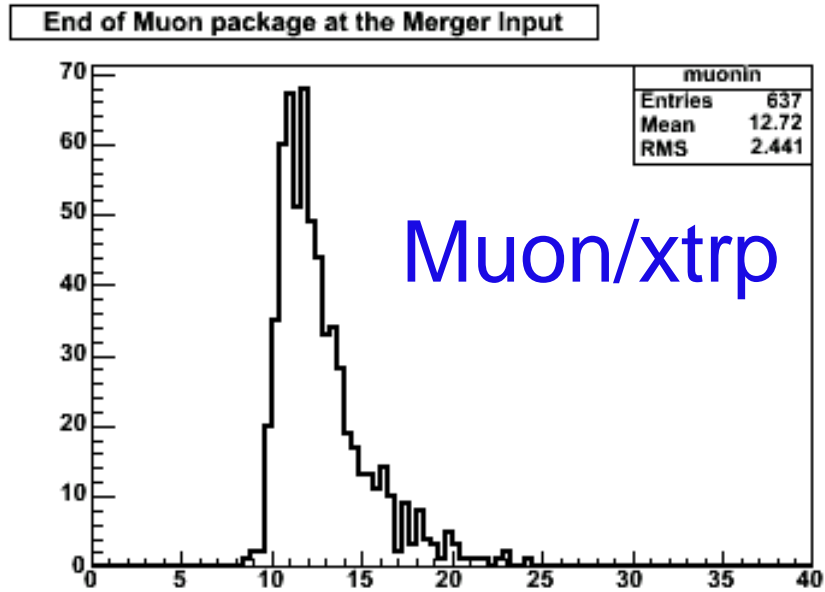


# Current configuration

Pulsar pre-processors

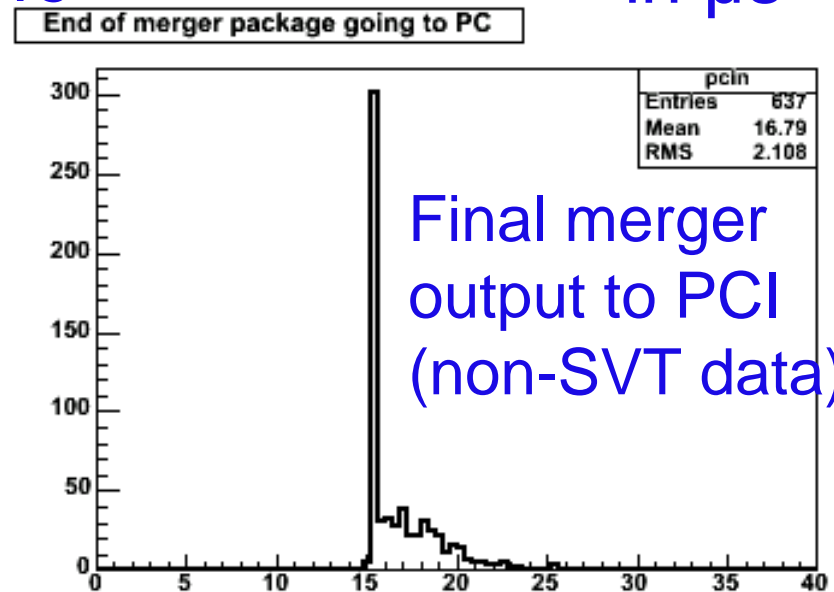
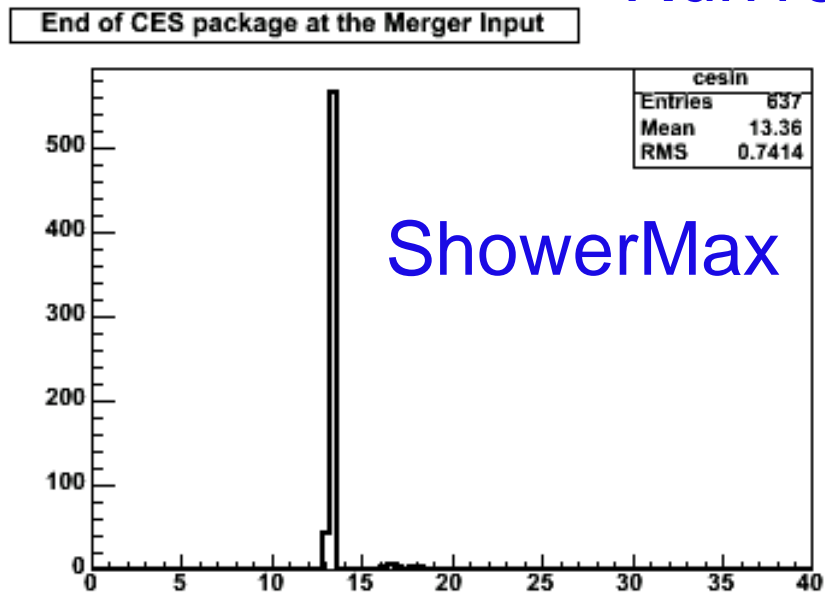


Now look at actual timing measurements ●  
by Pulsar Boards....  
(previous event, unbiased)



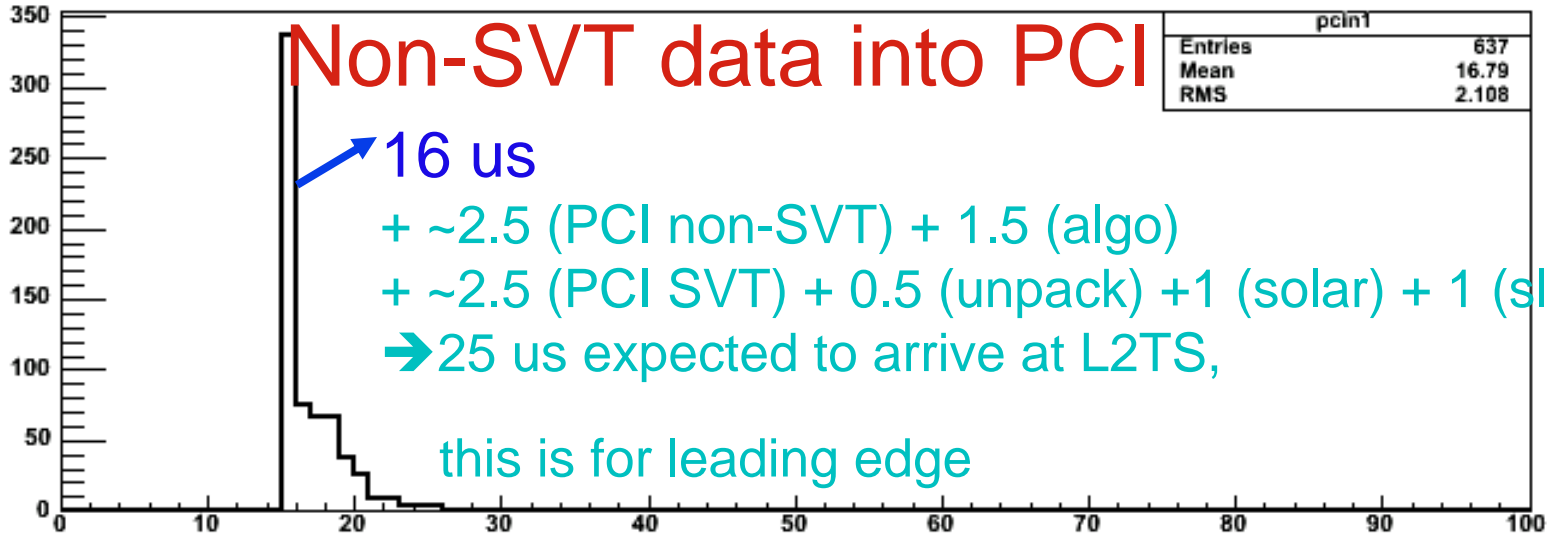
Run195449

in  $\mu\text{s}$



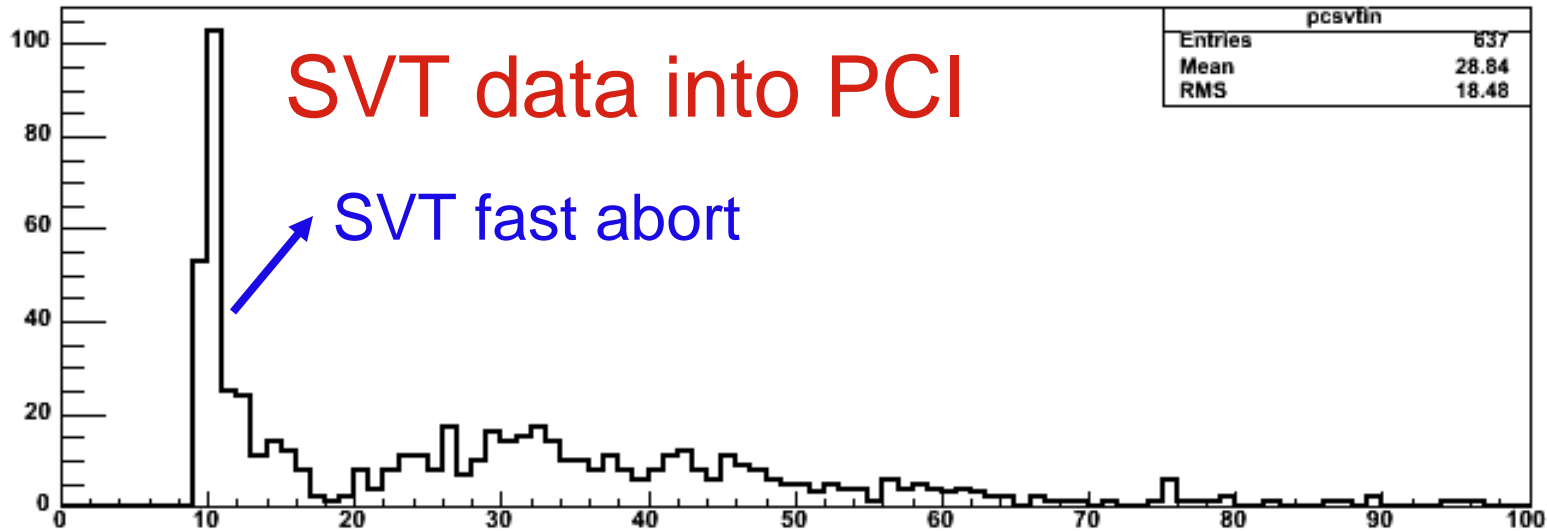
All Pulsar timing plots were made using software tools developed by Vadim/Wojtek/Shawn

End of merger package going to PC

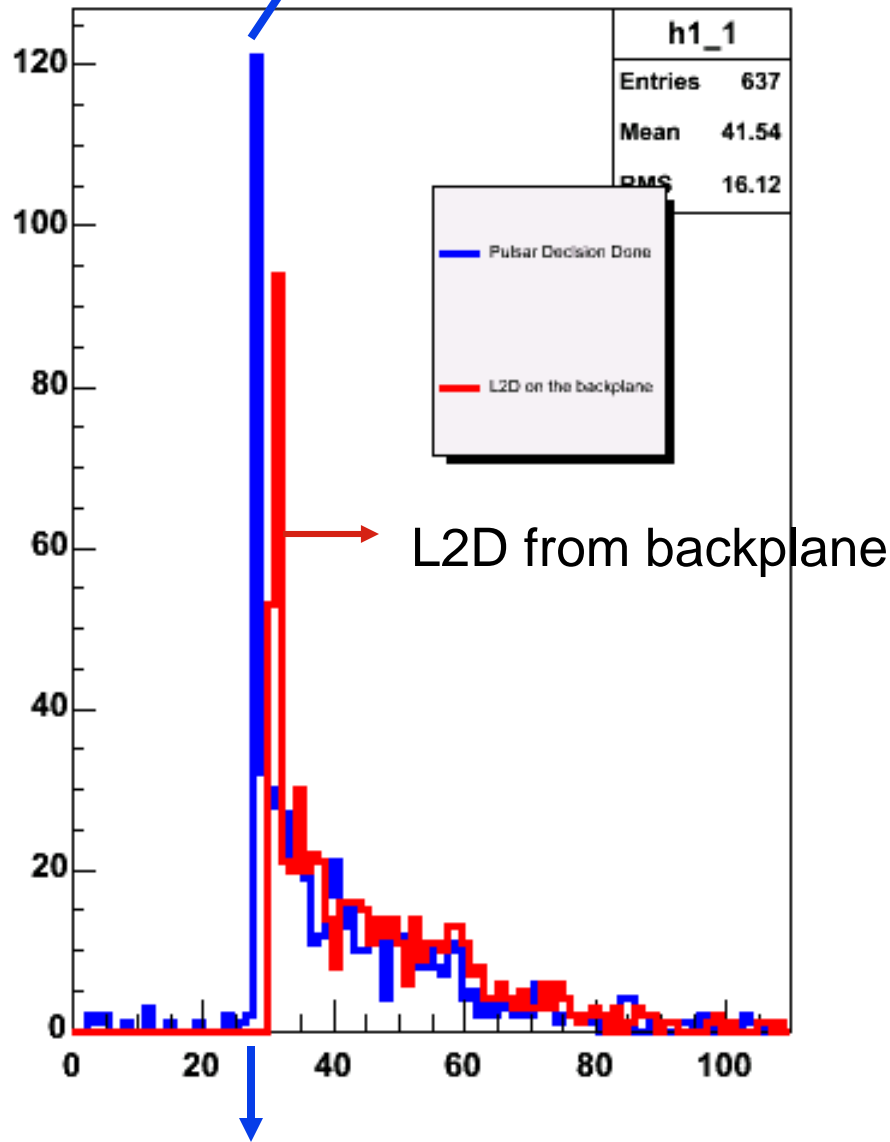


in  $\mu$ s

End of SVT package going to PC



L2D arrive at L2TS input



Where did ~ 3 us spend on?

This is puzzle #1

Need to find it and then shave it

We already know that there is room for improvements for PCI/pipeline handling

28 us observed, expect 25 us → extra ~ 3 us!



## Timing measurement on April 04

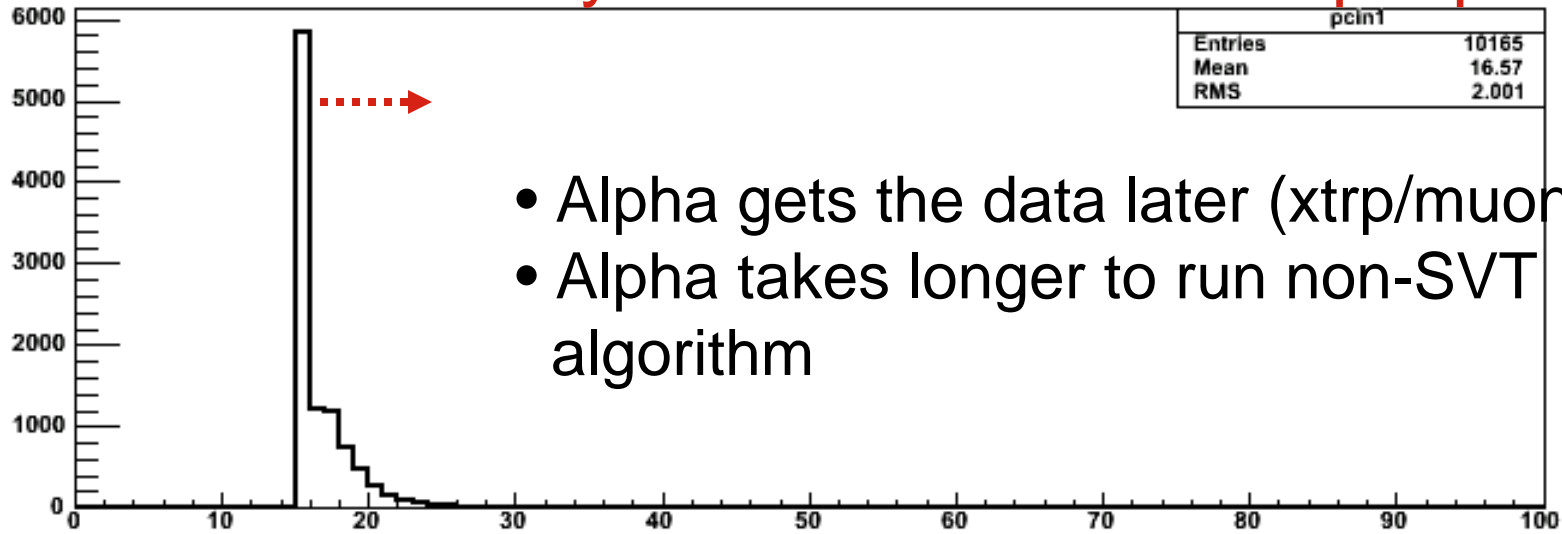
---

- At high luminosity  $\sim 100 \text{ e}30$ , with highlumi trigger table: PHYSICS\_HIGHLUM\_3\_01[1,617,558]
- SVT fast abort disabled

Run	L1A(kHz)	L2A(Hz)	DT(%)	L2latency(us)
196152	9.0	290	2.0	48 Pulsar&Alpha

SVT fast abort disable/enable makes no difference  
for Alpha driving

End of merger package going to PC

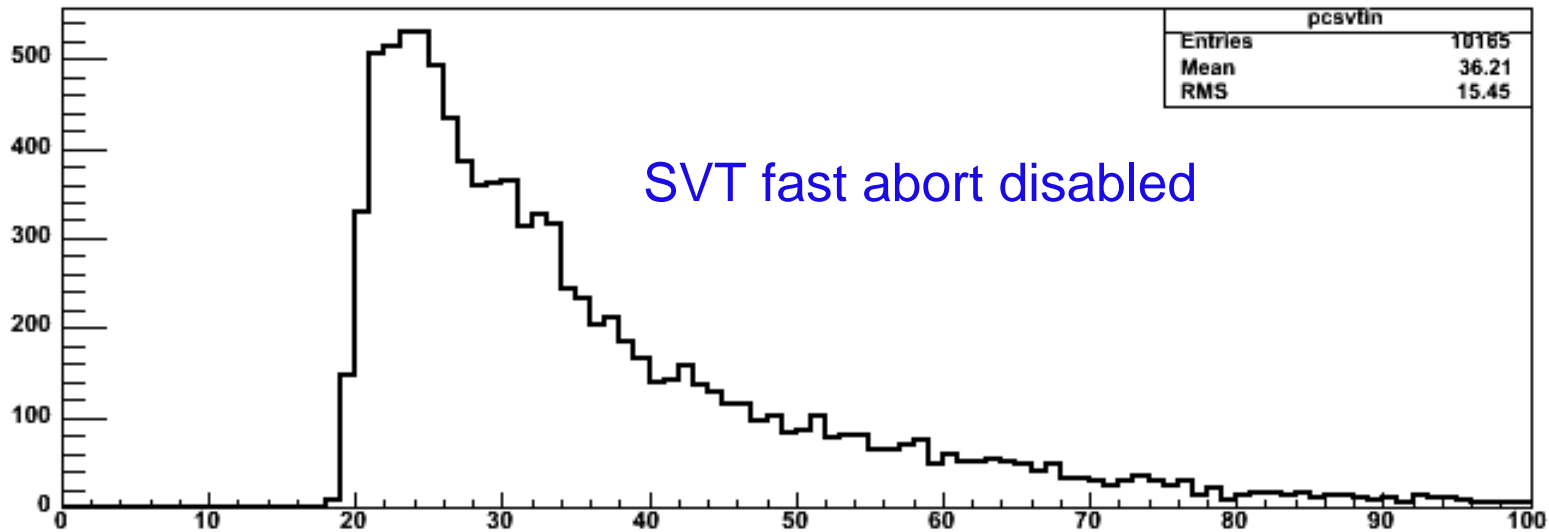


# Why fast abort doesn't help alpha?

- Alpha gets the data later (xtrp/muon)
- Alpha takes longer to run non-SVT algorithm

in  $\mu\text{s}$

End of SVT package going to PC



SVT fast abort disabled



## Timing on April 14/20

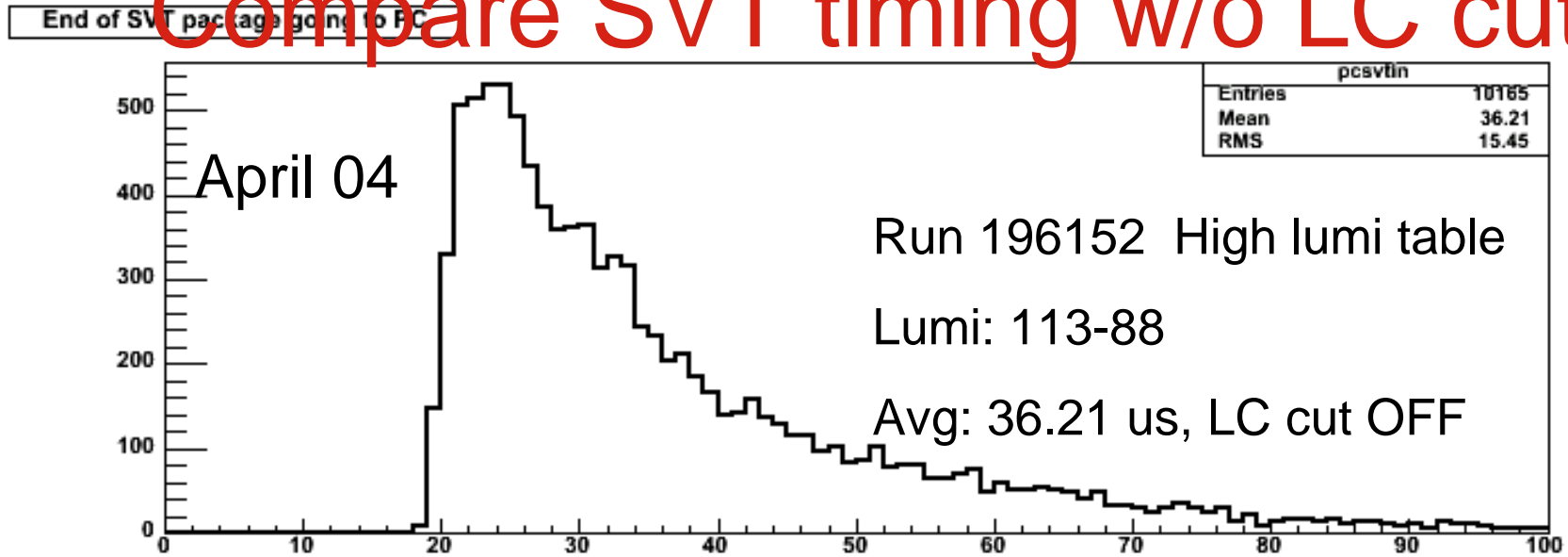
- At high luminosity  $\sim 100 \text{ e}30$ , with highlumi trigger table: PHYSICS\_HIGHLUM\_3\_02[1,640,562] w/ LC cut enabled
- SVT fast abort disabled

Run	L1A(kHz)	L2A(Hz)	DT(%)	L2latency(us)	
196471	10	320	2.0	46	Pulsar driving
196795	10	330	2.0	46	Pulsar driving

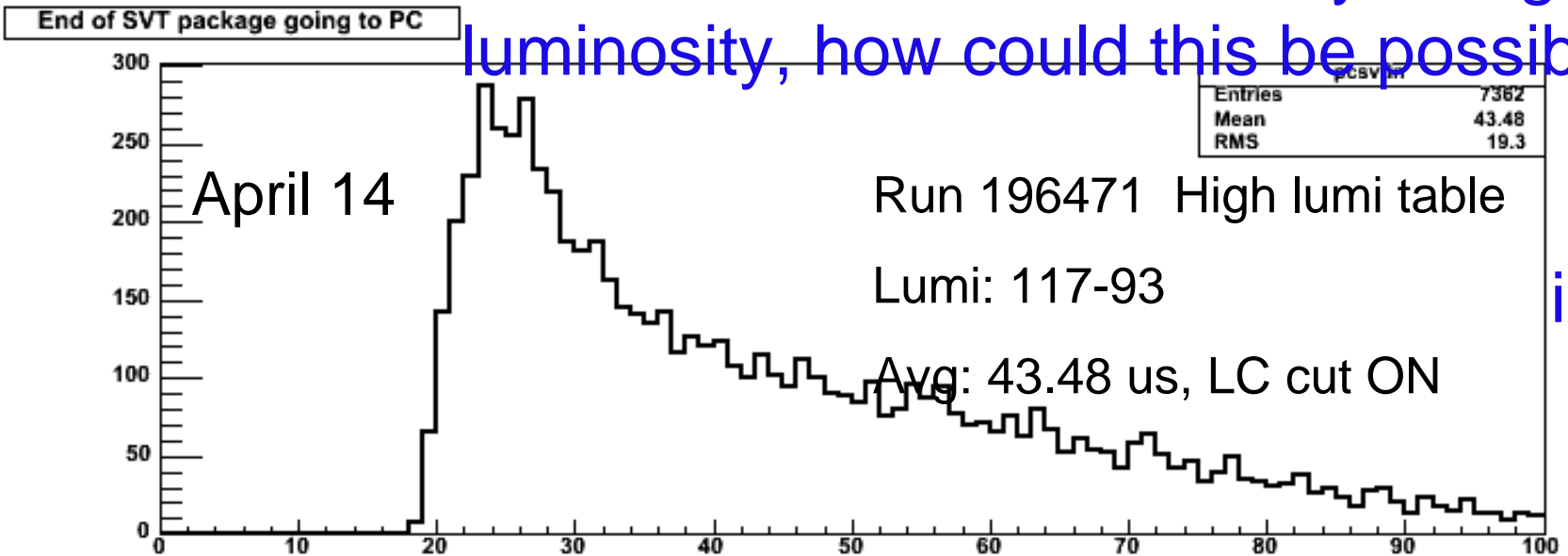
This is  $46 - 39 = 7\mu\text{s}$  longer than before! What caused this?

Differences: SVT LC cut is enabled & fast abort is disabled  
was it due to SVT fast abort disabling ? **Puzzle #2**

# Compare SVT timing w/o LC cut



It seems LC cut “increases” the latency at high luminosity, how could this be possible?

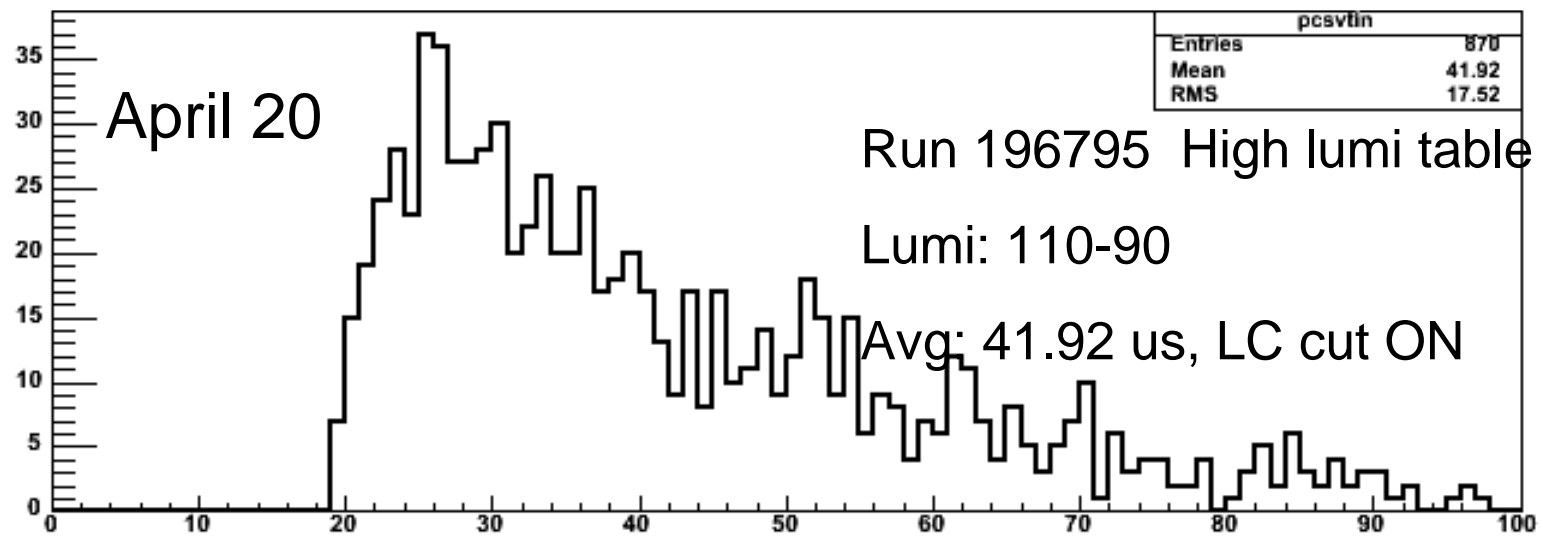
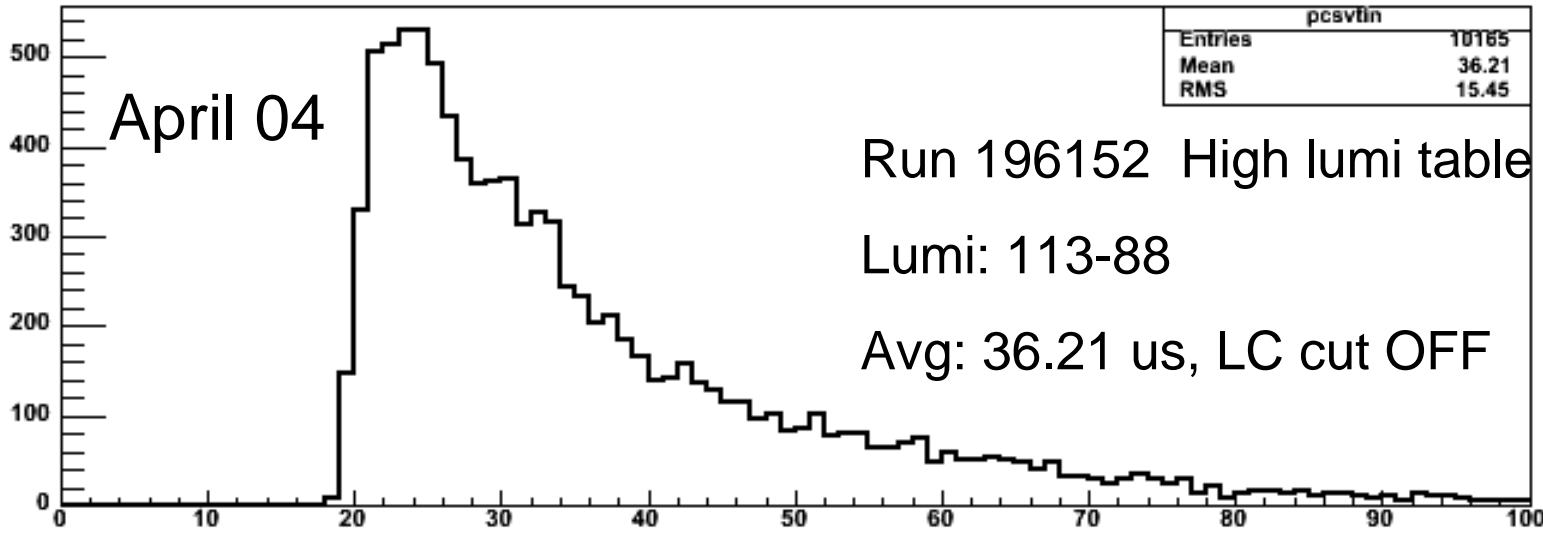


in  $\mu\text{s}$

Jahred: “impossible for LC firmware to do this”

# Compare w recent run on April 20

End of SVT package going to PC

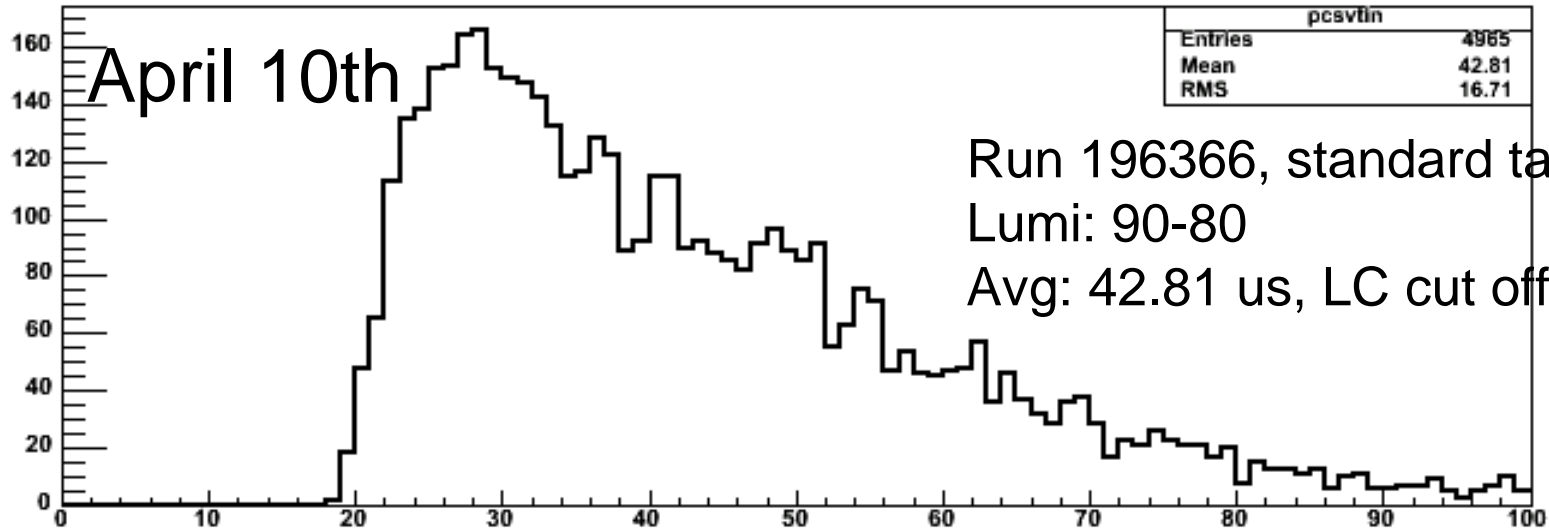


in  $\mu$ s

What else could do this? **Puzzle #3**

# How about at lower lumi?

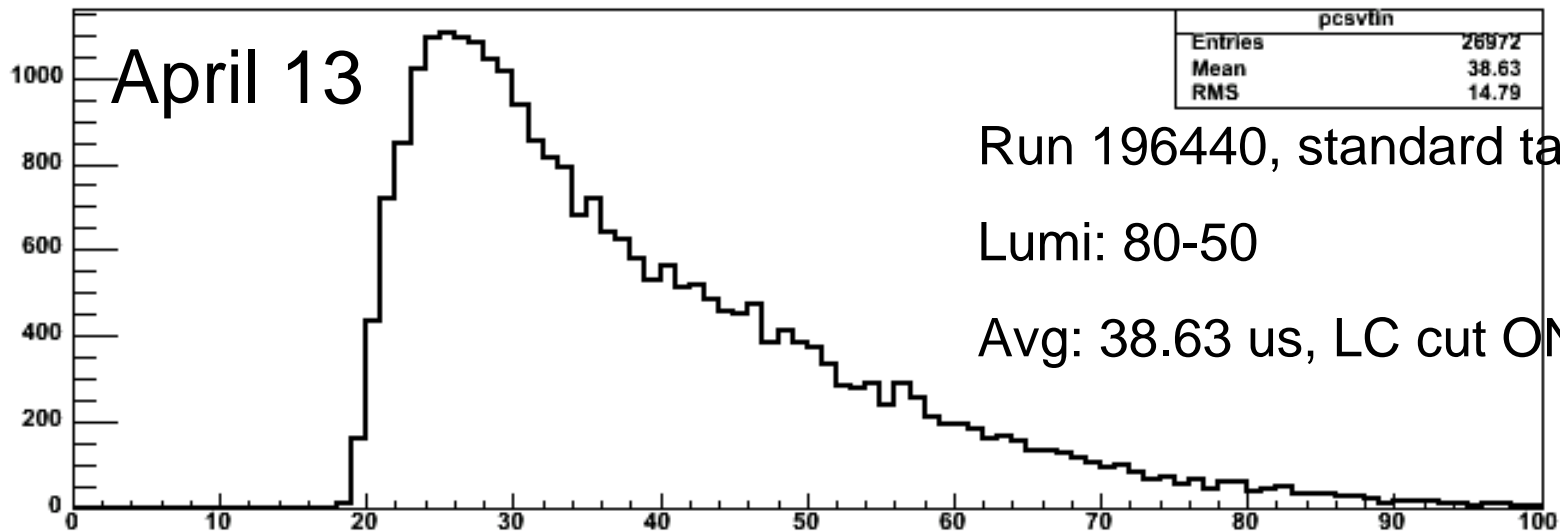
End of SVT package going to PC



Run 196366, standard table  
Lumi: 90-80  
Avg: 42.81 us, LC cut off

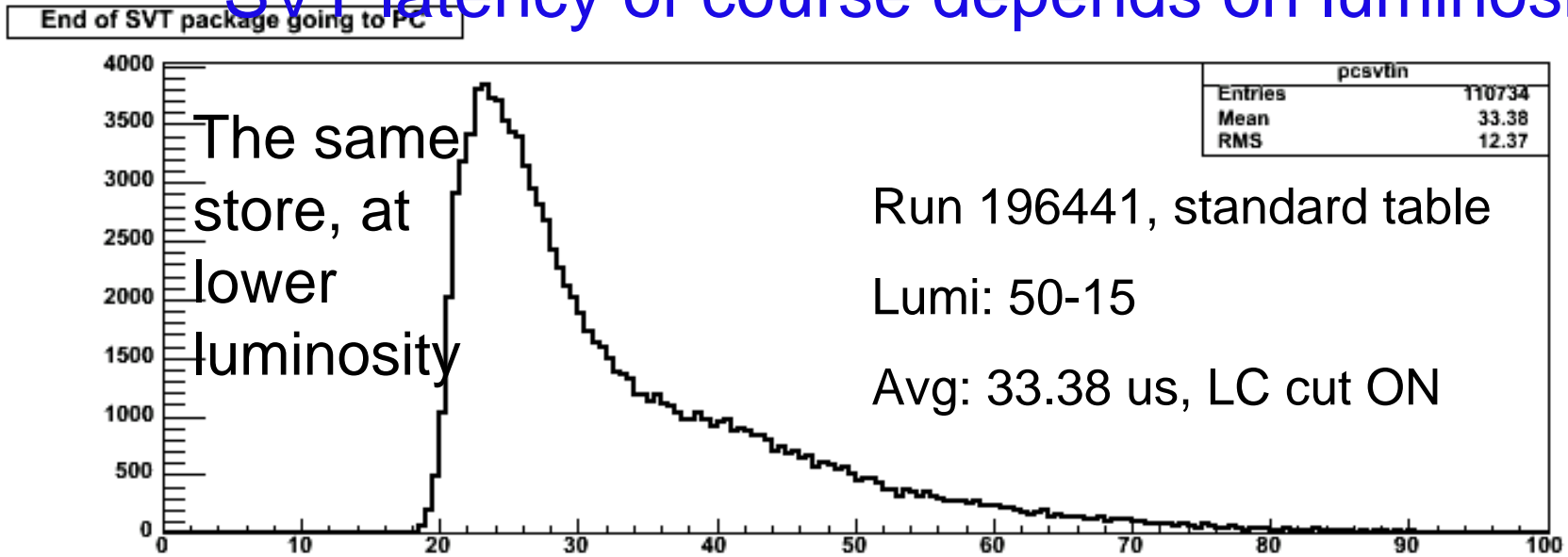
Seem to suggest LC cut reduces latency, but  
luminosity range is also lower ...

End of SVT package going to PC

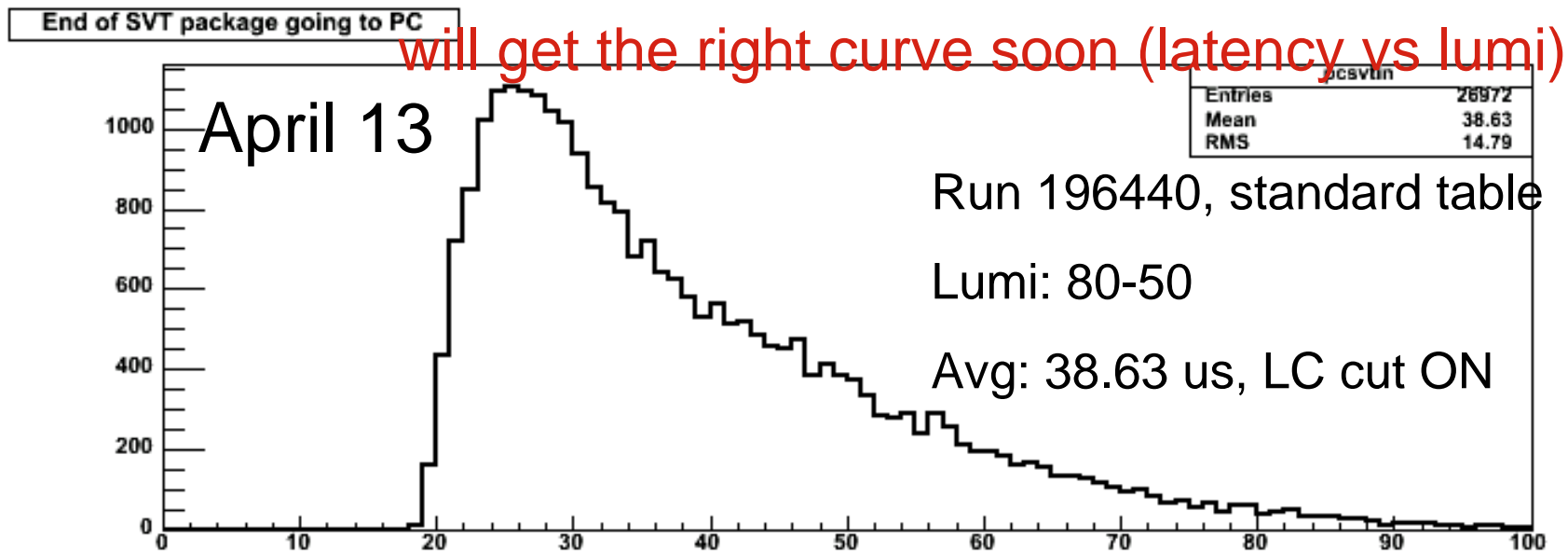


Run 196440, standard table  
Lumi: 80-50  
Avg: 38.63 us, LC cut ON

# SVT latency of course depends on luminosity

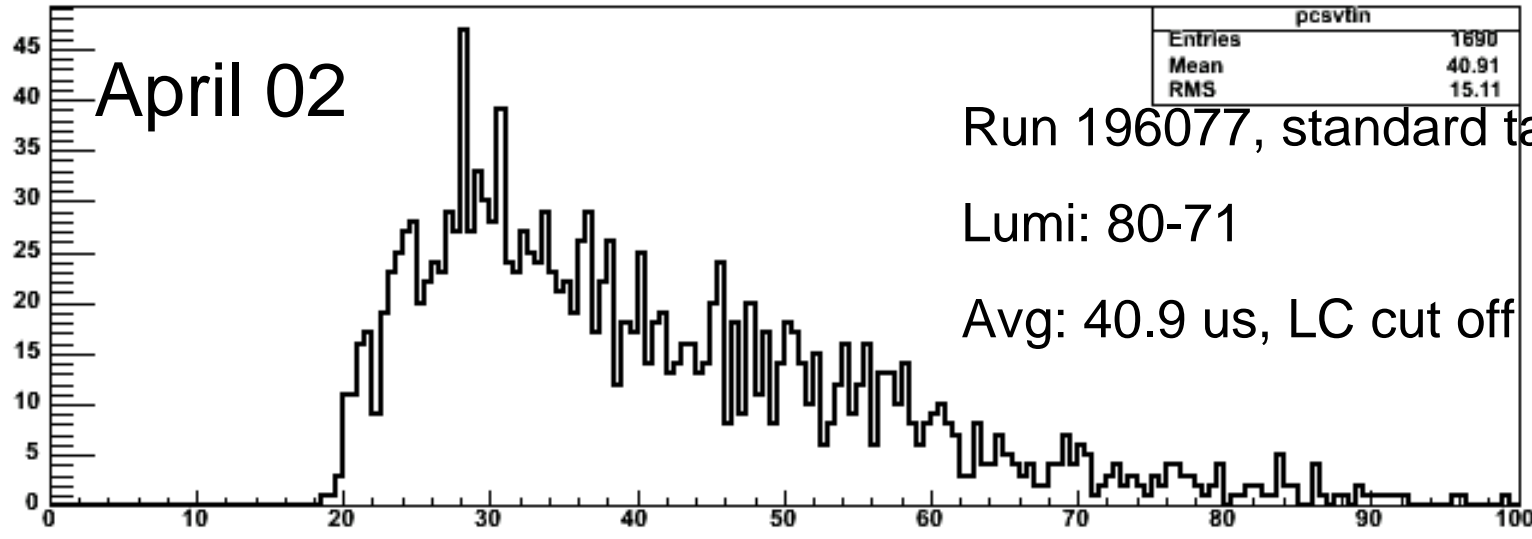


..... seems at rate of  $\sim 1.6 \mu\text{s}$  per  $10e30$  ...likely non-linear.



# Another run to double check ... but

End of SVT package going to PC



April 02

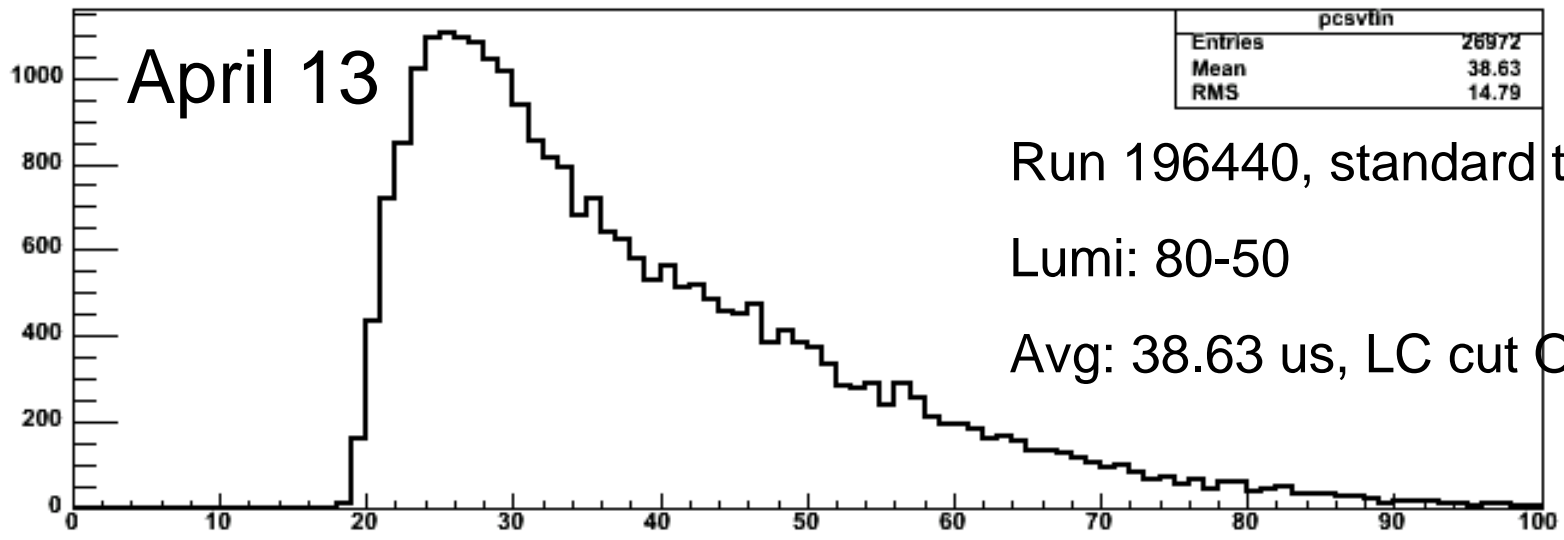
Run 196077, standard table

Lumi: 80-71

Avg: 40.9 us, LC cut off

the luminosity range is still somewhat different... inconclusive for lower luminosity

End of SVT package going to PC



April 13

Run 196440, standard table

Lumi: 80-50

Avg: 38.63 us, LC cut ON



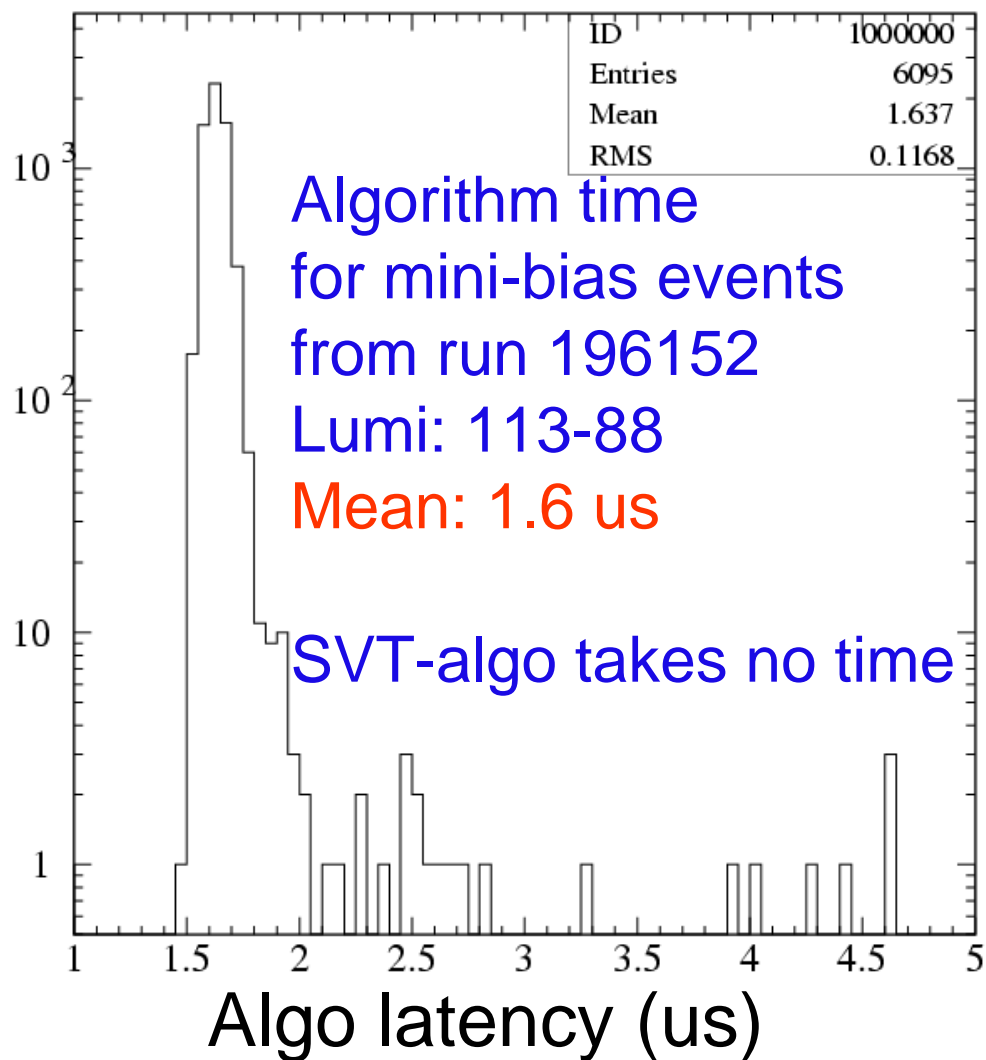
# Solving the three puzzles: could save $\geq \sim 7$ us at high lumi...

---

- **Puzzle #1: extra  $\sim 3$ us in PC/PCI-Pulsar roundtrip**
  - run decision algorithm code offline to study algorithm timing
    - already started: Vadim and Kristian are working on this → DONE!
  - implement timing measurements in decision CPU and save the info in TL2D diagnostic bank for long term studies
    - Daniel will implement this soon
- **Puzzle #2: how much SVT fast abort will help Pulsar?**
- **Puzzle #3: does SVT LC cut hurt overall performance at high luminosity, or something else doing it?**
  - take test runs at high luminosity with highlumi table
  - enable/disable fast abort and LC, one at a time(!)
    - ➔ 1<sup>st</sup> test happened at 3-5 am this morning... but at low lumi: 70-60. didn't learn as much due to other difficulties at control room
  - add luminosity/evt number/etc info in the Pulsar timing ntuple (Vadim)



# Quick feedback from Vadim/Kristian last night



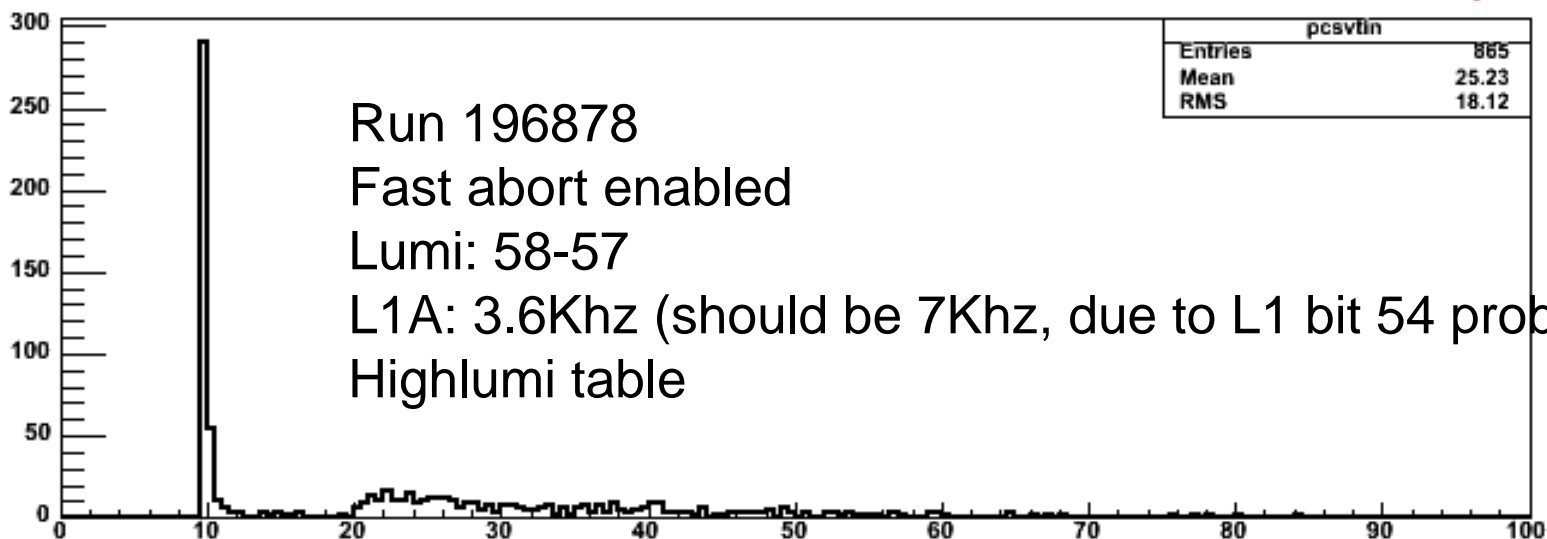
Vadim/Kristian  
quickly striped  
real events from high lumi  
Run 196152, and ran  
the algo code offline  
on Pulsar build node

this was ALL done  
within a few hours (!)  
after they learned about  
**puzzle #1**

Very useful tool later on

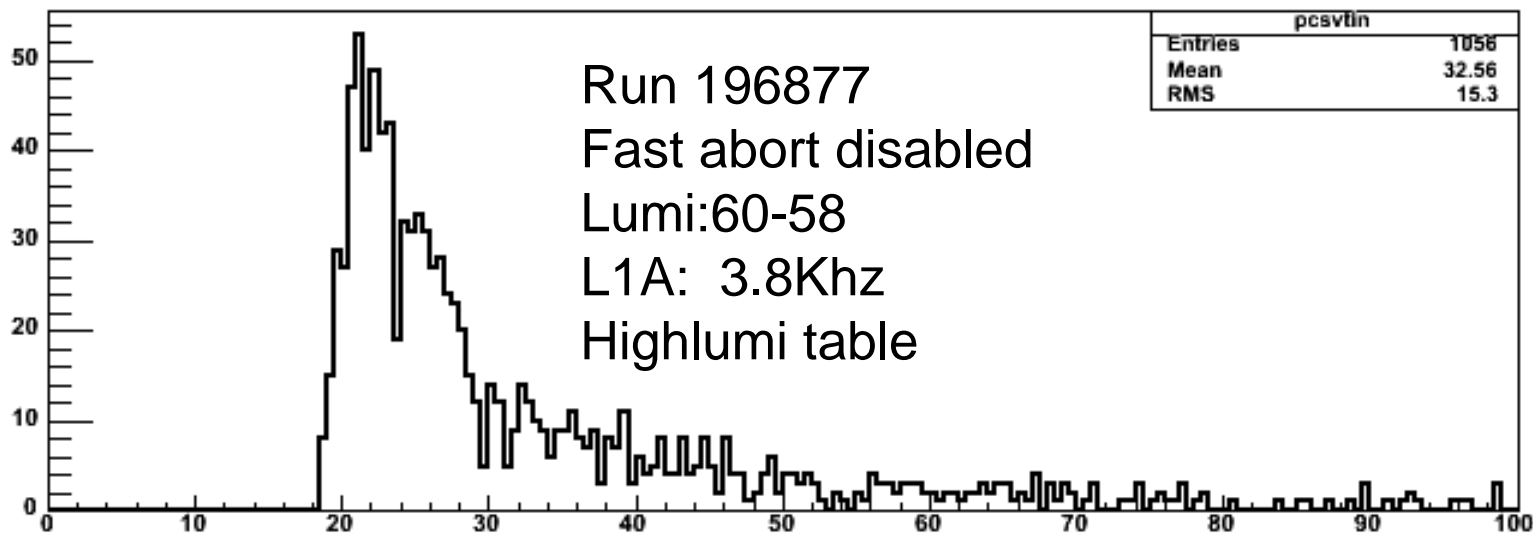
## Fresh beam data this morning

End of SVT package going to PC



## The luminosity was low...

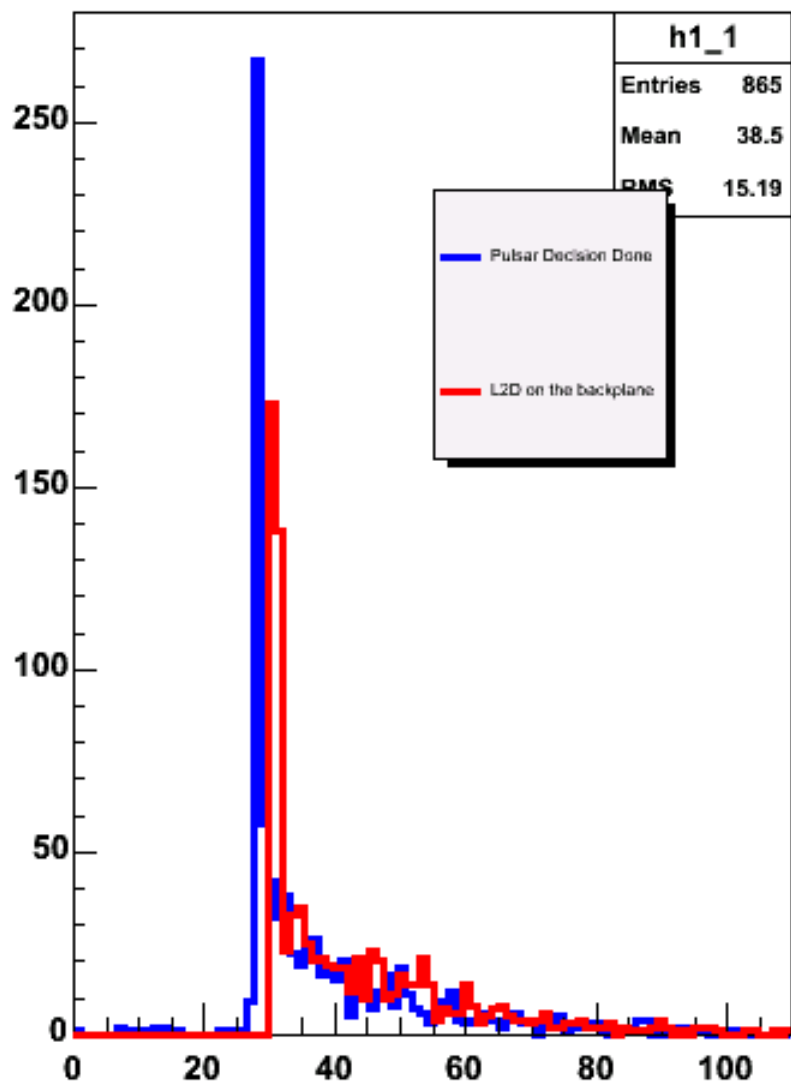
End of SVT package going to PC





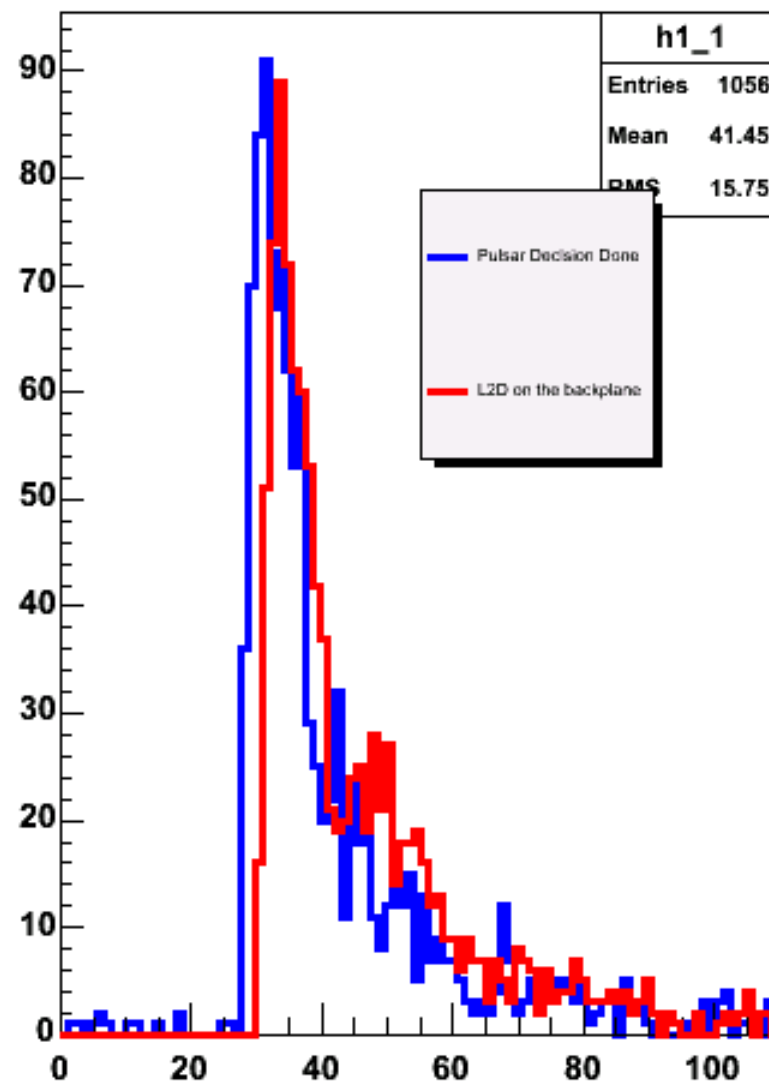
Pulsar timing run 196878

# Fast abort enabled



Pulsar timing run 196877

# disabled

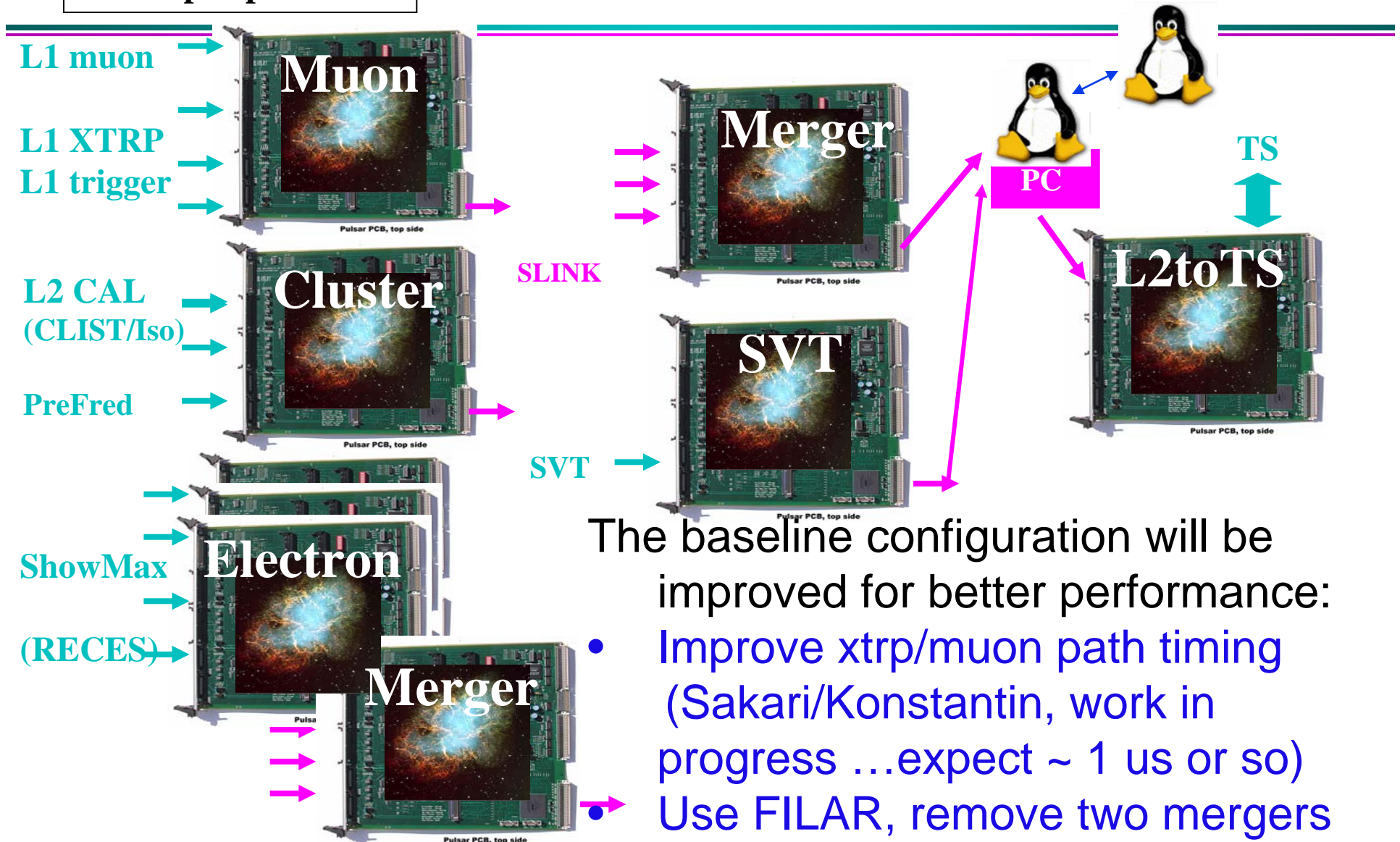


$41.5 - 38.5 = 3$  us gain at low lumi



# Improving the Baseline

## Pulsar pre-processors



The baseline configuration will be improved for better performance:

- Improve xtrp/muon path timing (Sakari/Konstantin, work in progress ...expect ~ 1 us or so)
- Use FILAR, remove two mergers



# Plan for system performance optimization by fall shutdown

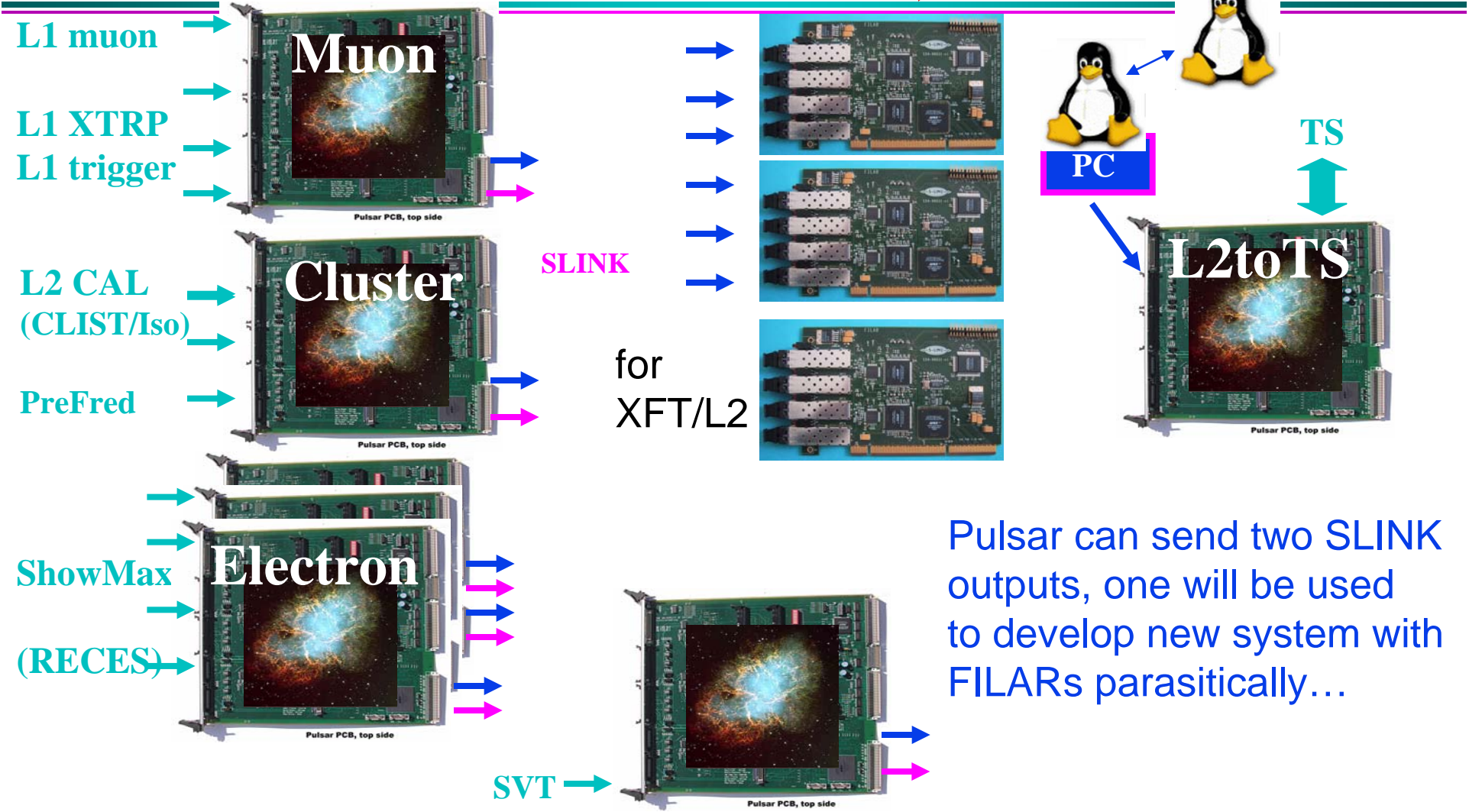
---

- Use FILAR cards instead of S32PCI64
  - Remove the final merger, data directly goes into FILAR
  - Replace Reces merger with another FILAR
  - Also allows us to run SLINK faster: 40 MHz → 60 MHz
  - expect to reduce the non-SVT latency by ~ a few us
  - Run in multi-events request mode, true pipeline inside PC (currently run in single-shot mode and thus semi-pipeline)
  - expect better performance at higher luminosity/L1A rate
  - Have one FILAR in hand, software/test work in progress: Sakari/Kristian/Vadim, later on Konstantin/Daniel
  - plan to order 10 FILARs soon, waiting for management to approve...



# System with FILARs

Pulsar pre-processors

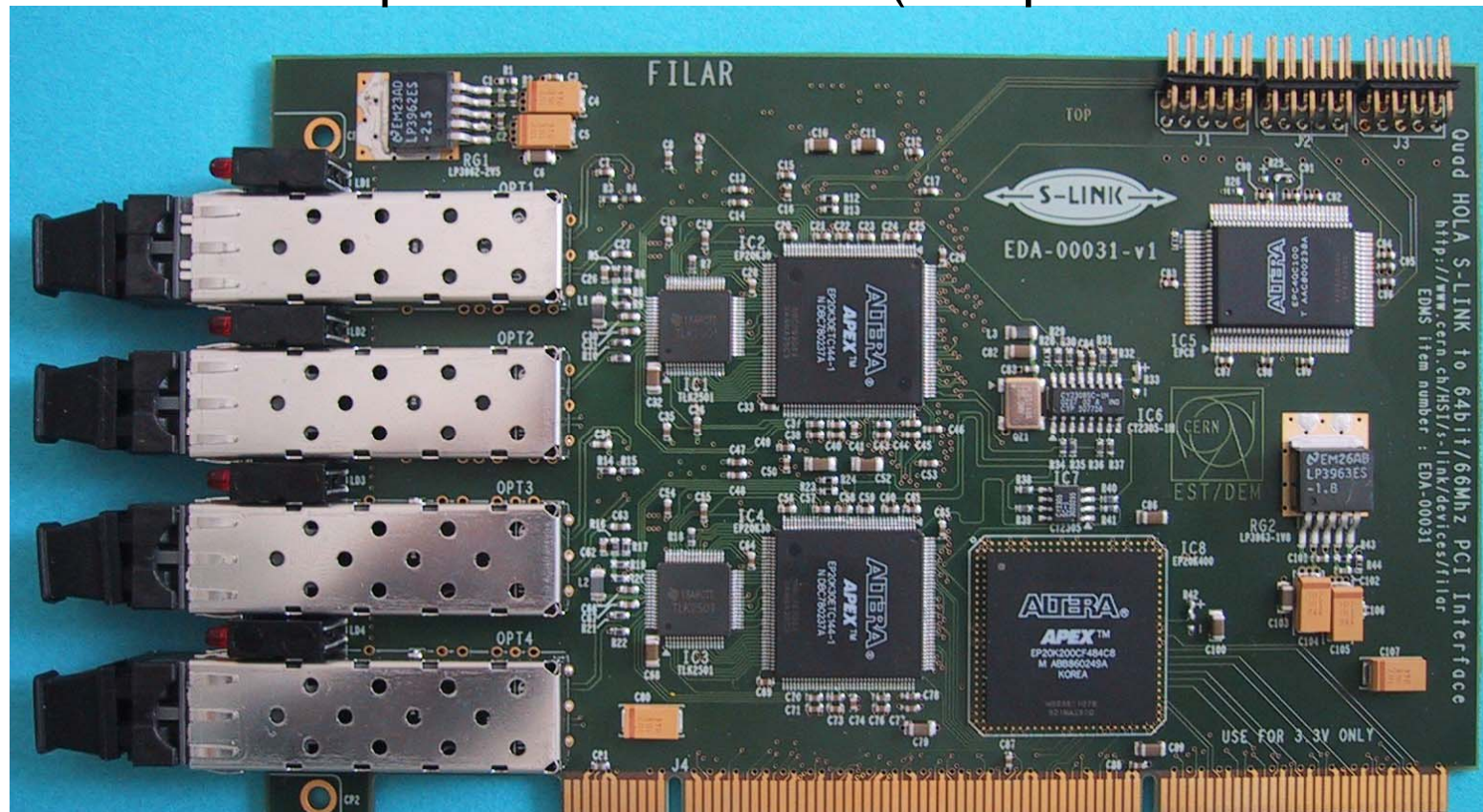


Pulsar can send two SLINK outputs, one will be used to develop new system with FILARs parasitically...



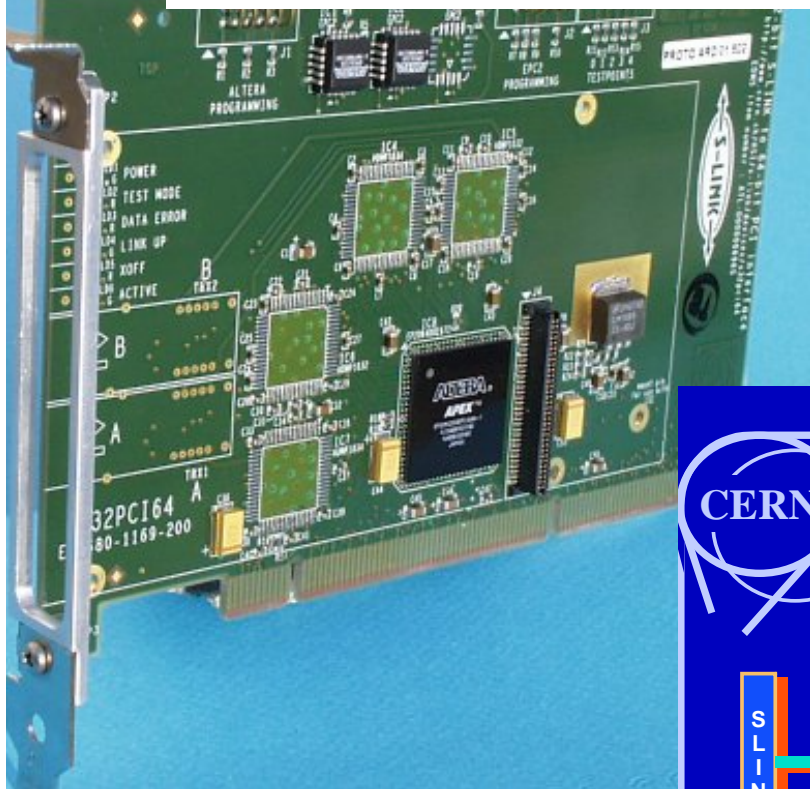
FILAR can improve the system latency  
→ Higher bandwidth, less latency overhead

64-bit/66MHz PCI bus, four 2 Gbit/s S-LINK channels  
Reduced PCI protocol overhead (compare to S32PCI64)

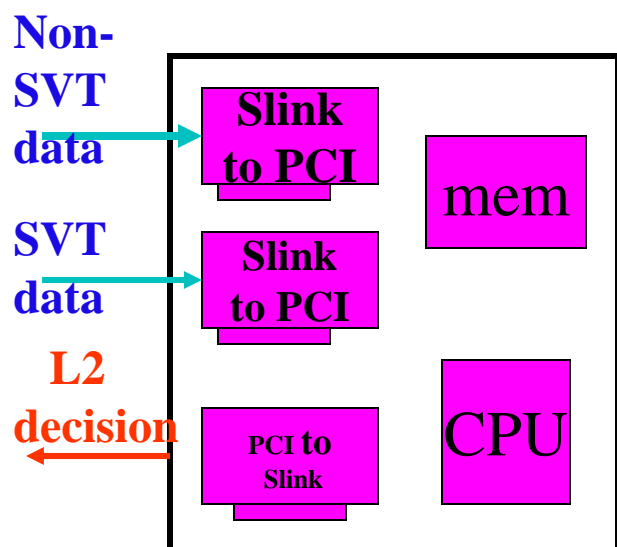
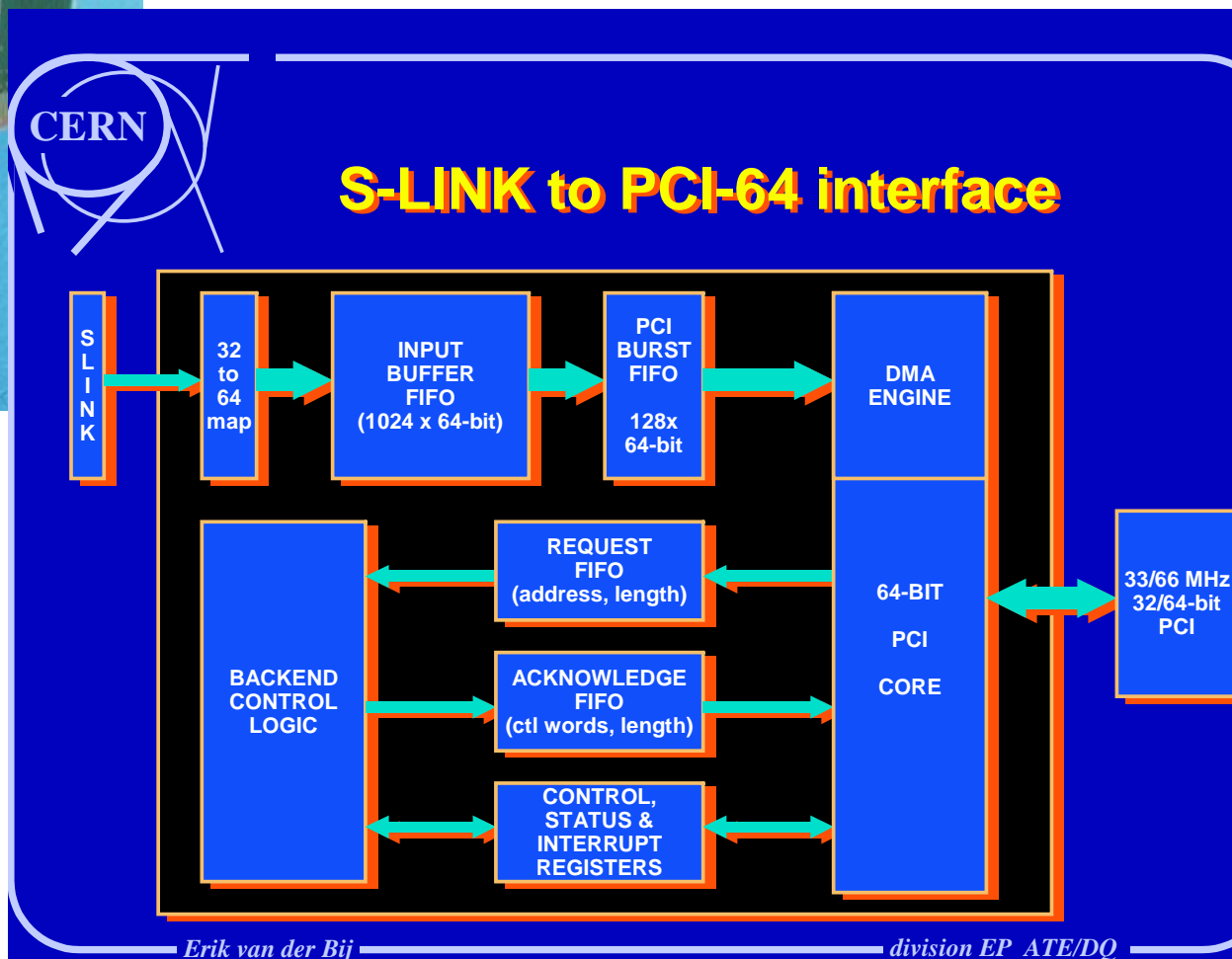


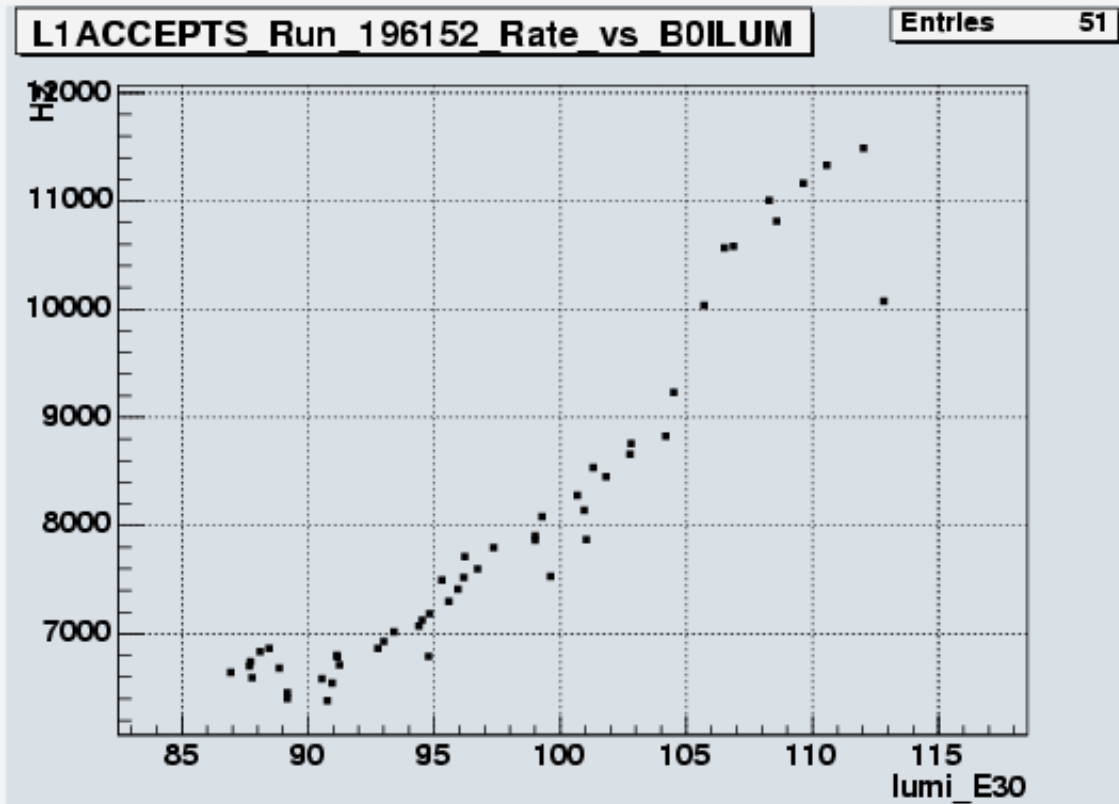
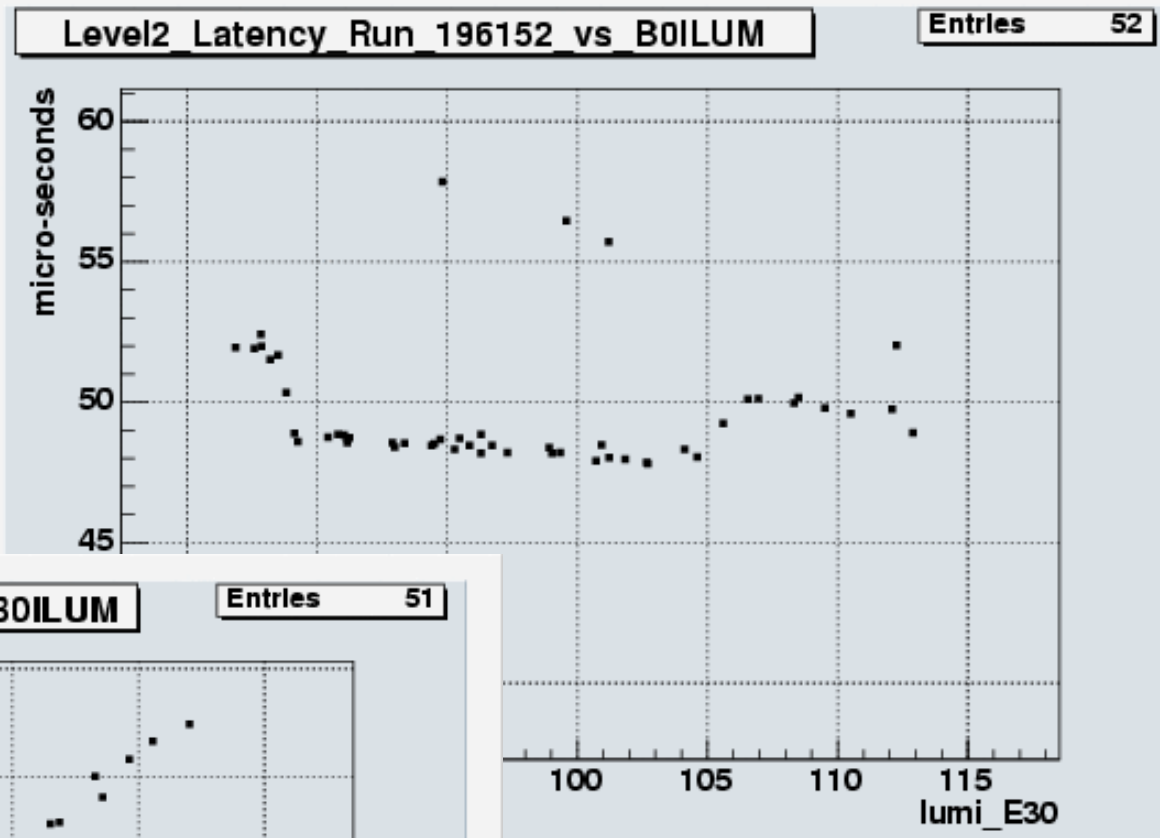
<http://hsi.web.cern.ch/HSI/s-link/devices/FILAR>

# 32-bit SLINK to 64 bit PCI interface card: S32PCI64



- highly autonomous data reception
- 32-bit SLINK, 64-bit PCI bus
- 33MHz and 66 MHz PCI clock speed



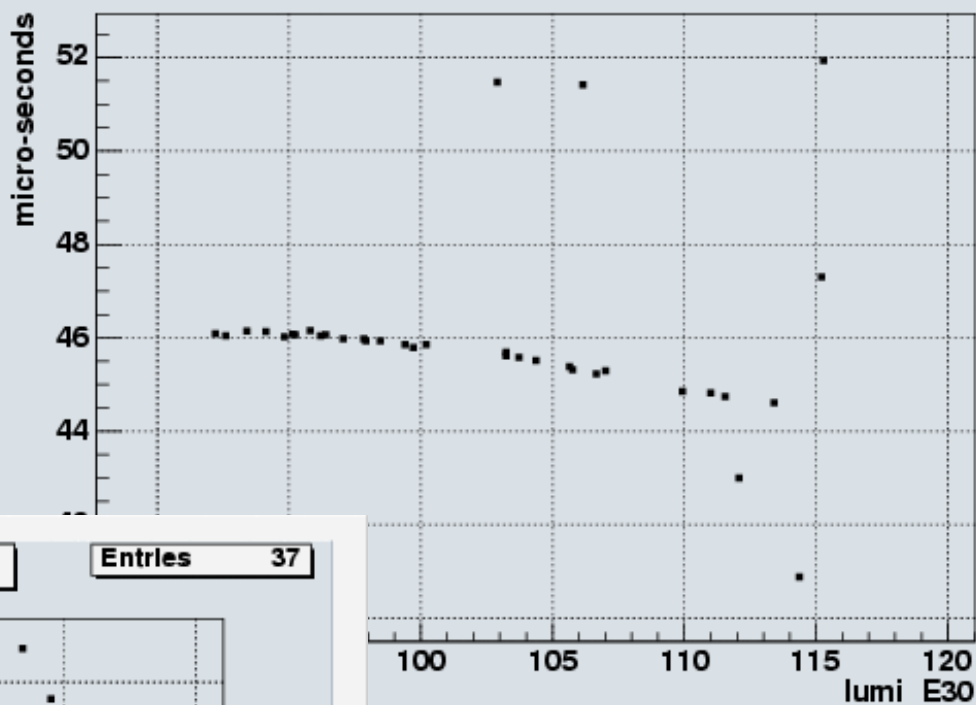


Some details  
from online plots



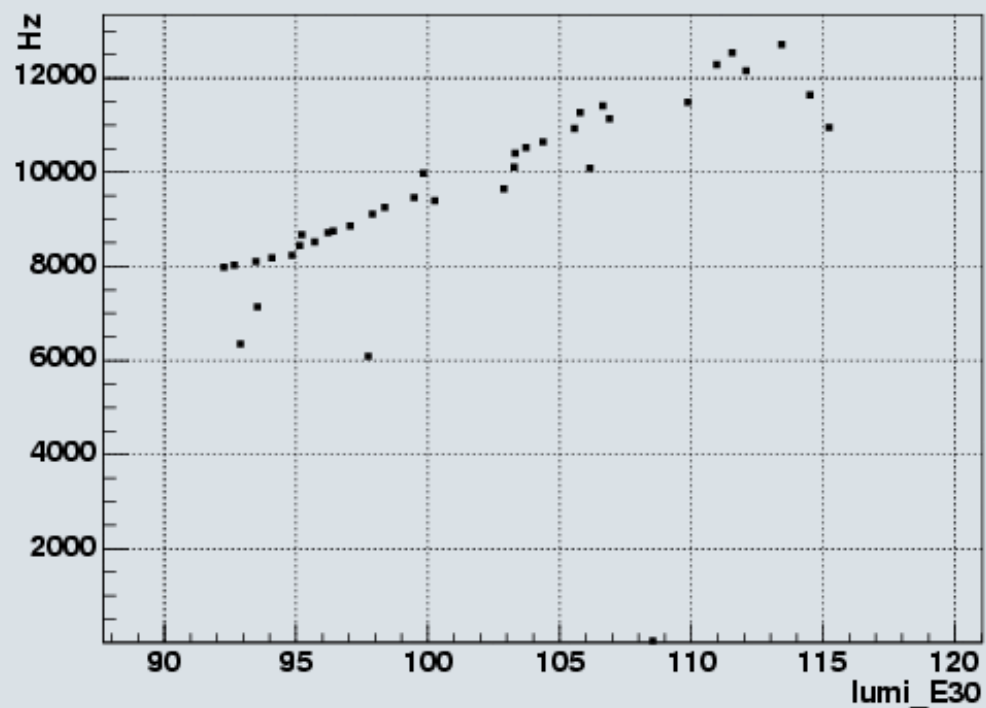
Level2 Latency Run 196471 vs B0ILUM

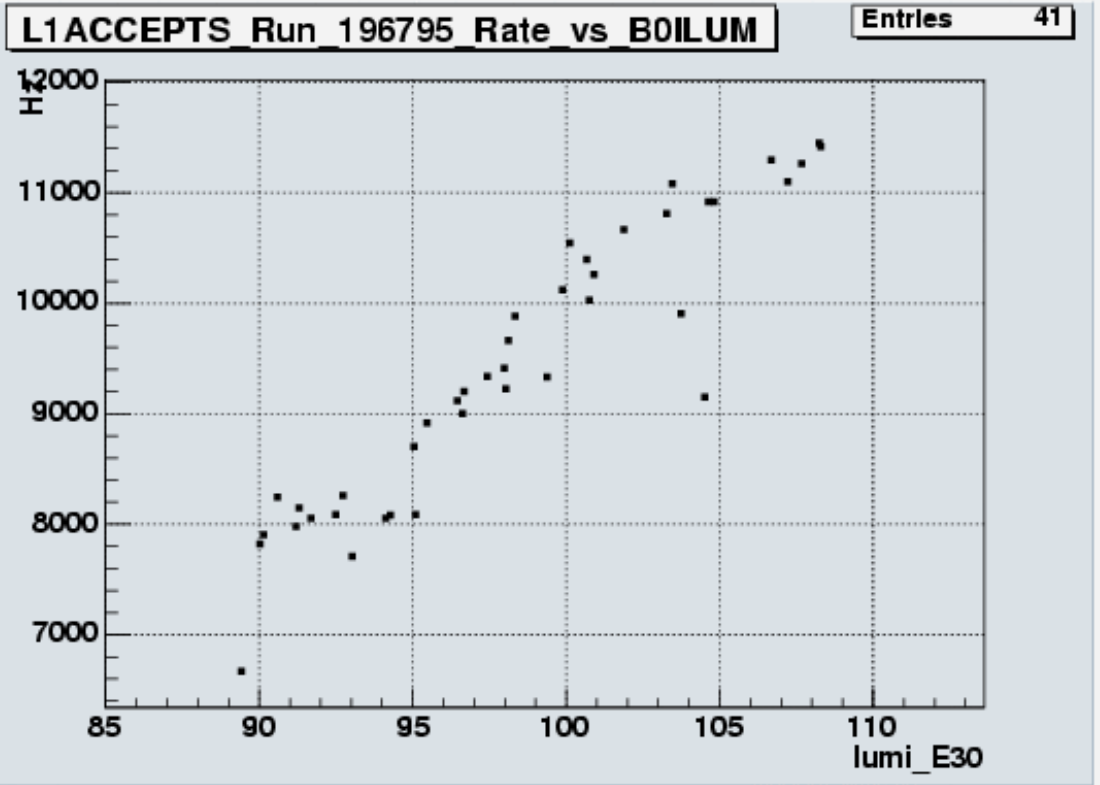
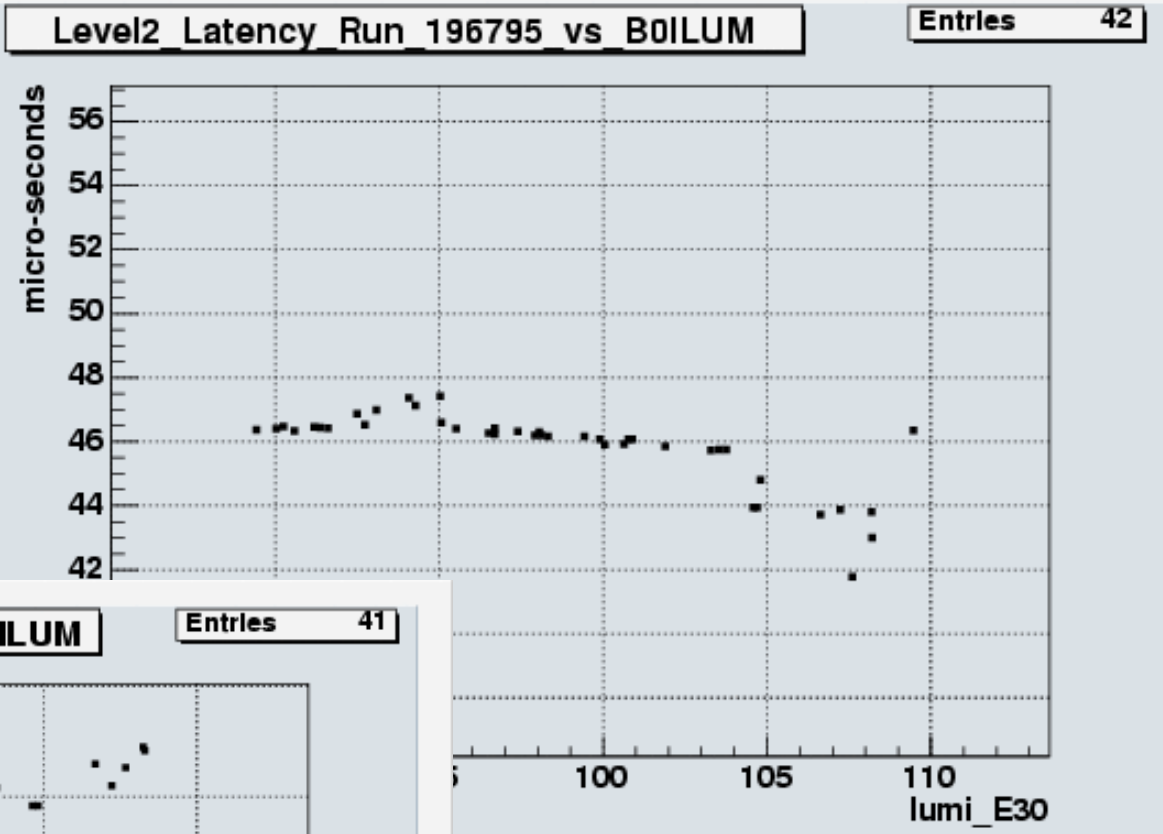
Entries 38



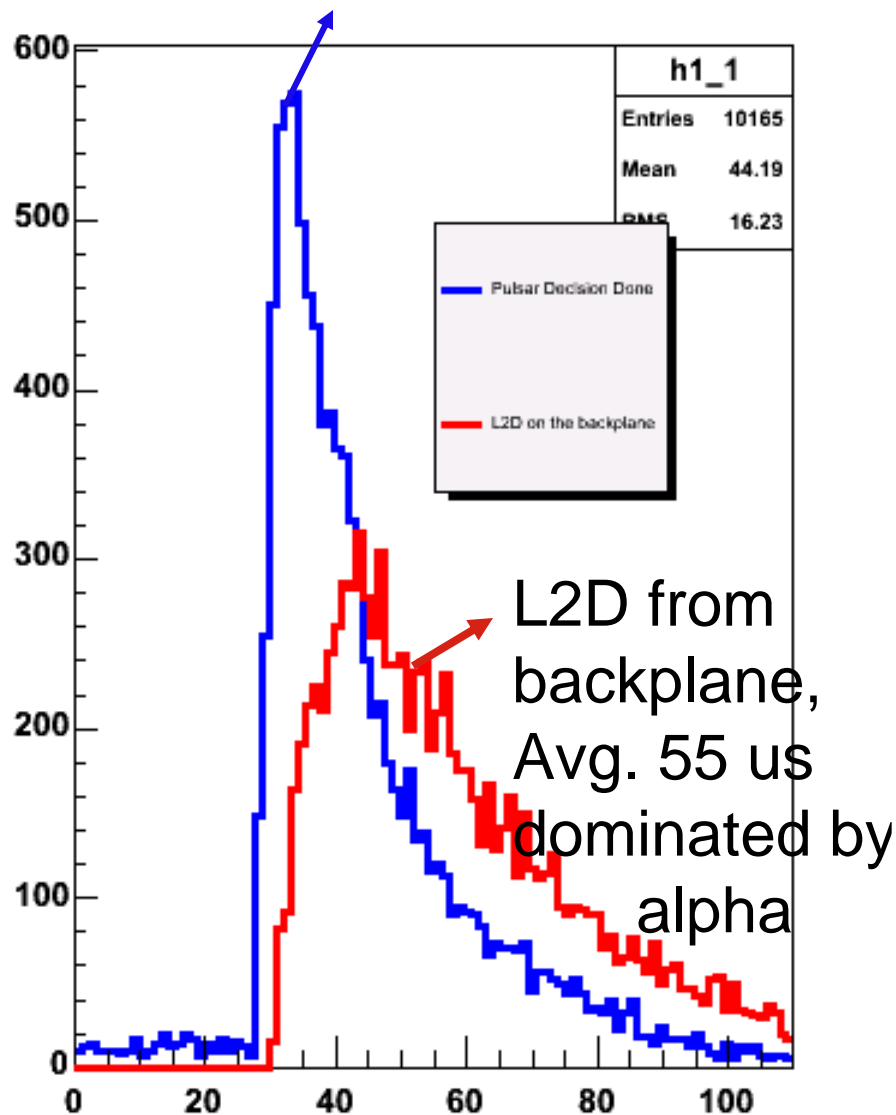
L1ACCEPTS Run 196471 Rate vs B0ILUM

Entries 37





### Pulsar L2D arrive at L2TS input



Note that the time definition here is different from online average

This is time  $t(n)$  between:  
L1A(buffer  $n$ ) to L2D (buffer  $n$ )

Where  $n=0,1,2,3$