Plans for Trigger Software Validation During Running

Trigger Data Quality Assurance Workshop

May 6, 2008

Ricardo Gonçalo, David Strom

Moving to real data

- Trigger Software Validation has so far been concerned with validating trigger software running in offline MC productions
- Need to understand what changes will be needed for LHC exploitation
- Diffuse boundaries between Validation, Online Integration, Monitoring (online/offline)
- A clear definition of tasks is needed to avoid duplication of efforts
- Different activities should work coherently: allow re-use of tools and optimisation of effort

Comparison with data quality monitoring

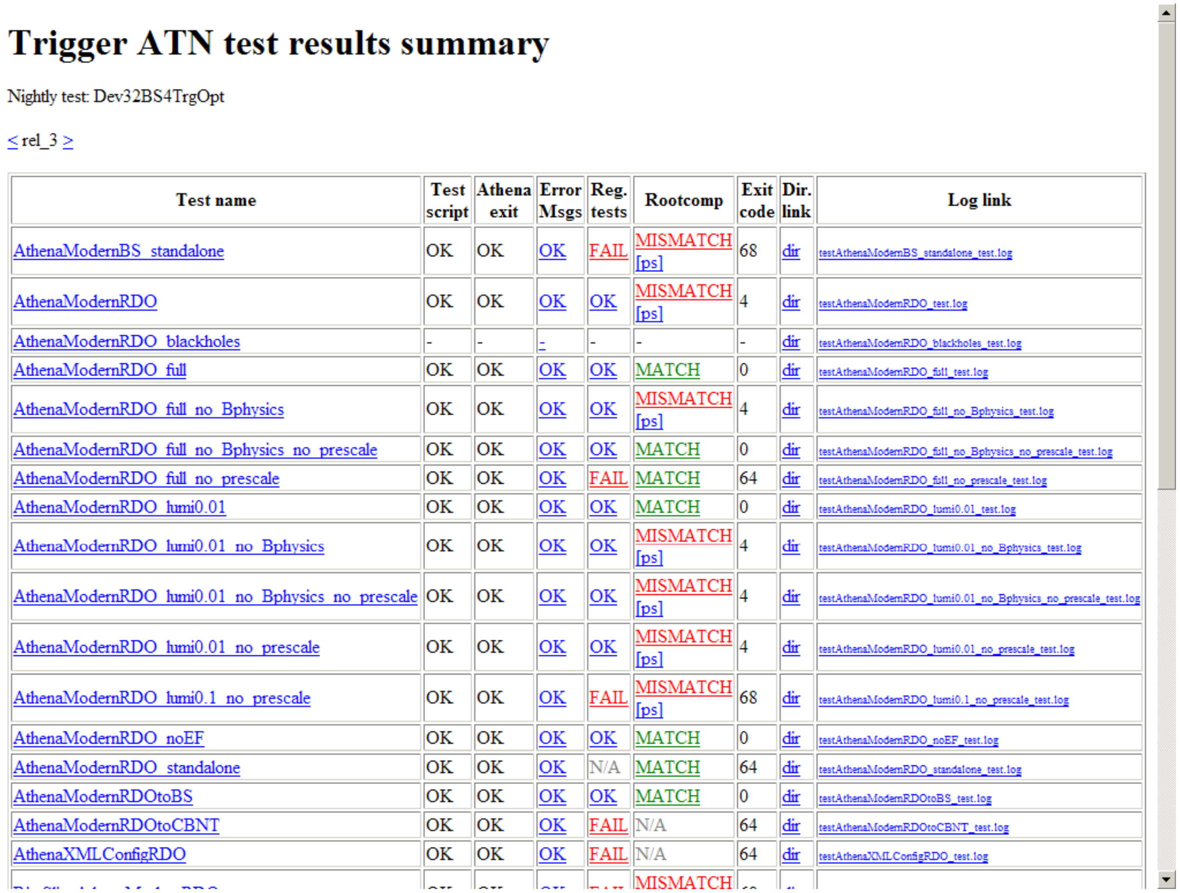
- Points in common with monitoring:
  - Monitoring histograms are used in validation tests
  - Need to be able to examine lots of partial results with little effort
  - Tries to cover whole range of possible problems (unknown unknowns…)
  - Tries to validate all steps in the processing chain
  - Hierarchical information generated for diagnostics – if overall test (rate) fails (is too high), look at which slice tests are failing (partial rates are too high)

Comparison with data quality monitoring

- Main differences:
  - Most work done before the code goes online
  - Runs different code every night on same set of data, instead of same code on new sets of data
  - Validation tries to guarantee the code will do what it is supposed to do (and in the time/memory it has to do it); monitoring tries to verify that it did
  - Not only histograms:
    - Code metrics very important: memory, CPU time, data volume
    - Other tests: chain counts, EDM quantities – also important in data quality monitoring!

Current validation infrastructure

- ATN tests:
  - Some 50 test jobs running on a few events every night
  - Regression tests on log files
  - Test different menus, each slice separately, writing/reading POOL, black-hole events, steering, EDM, TrigDecisionTool, TAG, etc



Trigger ATN test results summary

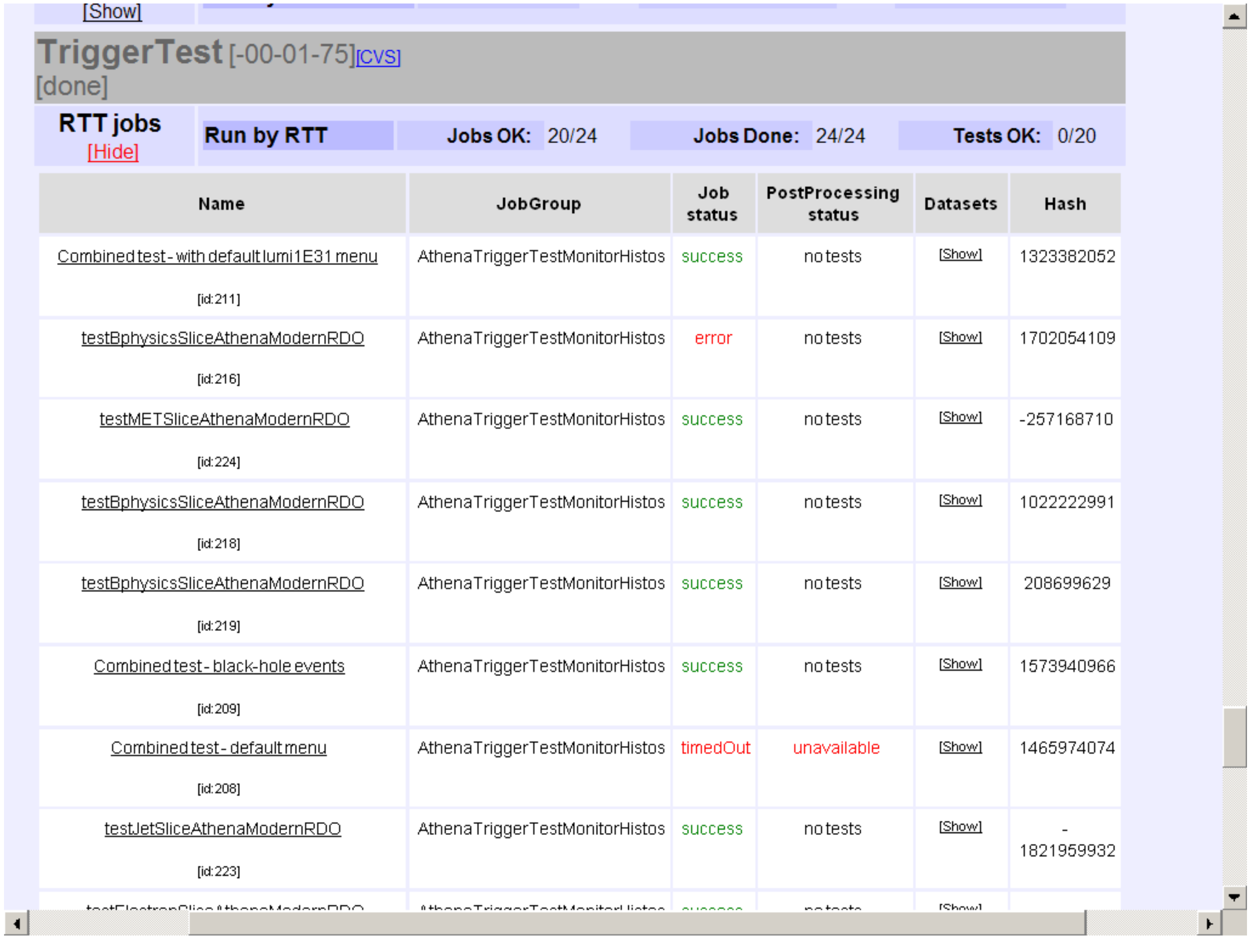
Nightly test: Dev32BS4TrgOpt

< rel_3 >

Test name	Test script	Athena exit	Error Msgs	Reg. tests	Rootcomp	Exit code	Dir. link	Log link
AthenaModernBS standalone	OK	OK	OK	FAIL	MISMATCH [ps]	68	dir	testAthenaModernBS_standalone_test.log
AthenaModernRDO	OK	OK	OK	OK	MISMATCH [ps]	4	dir	testAthenaModernRDO_test.log
AthenaModernRDO blackholes	-	-	-	-	-	-	dir	testAthenaModernRDO_blackholes_test.log
AthenaModernRDO full	OK	OK	OK	OK	MATCH	0	dir	testAthenaModernRDO_full_test.log
AthenaModernRDO full no Bphysics	OK	OK	OK	OK	MISMATCH [ps]	4	dir	testAthenaModernRDO_full_no_Bphysics_test.log
AthenaModernRDO full no Bphysics no prescale	OK	OK	OK	OK	MATCH	0	dir	testAthenaModernRDO_full_no_Bphysics_no_prescale_test.log
AthenaModernRDO full no prescale	OK	OK	OK	FAIL	MATCH	64	dir	testAthenaModernRDO_full_no_prescale_test.log
AthenaModernRDO lumi0.01	OK	OK	OK	OK	MATCH	0	dir	testAthenaModernRDO_lumi0.01_test.log
AthenaModernRDO lumi0.01 no Bphysics	OK	OK	OK	OK	MISMATCH [ps]	4	dir	testAthenaModernRDO_lumi0.01_no_Bphysics_test.log
AthenaModernRDO lumi0.01 no Bphysics no prescale	OK	OK	OK	OK	MISMATCH [ps]	4	dir	testAthenaModernRDO_lumi0.01_no_Bphysics_no_prescale_test.log
AthenaModernRDO lumi0.01 no prescale	OK	OK	OK	OK	MISMATCH [ps]	4	dir	testAthenaModernRDO_lumi0.01_no_prescale_test.log
AthenaModernRDO lumi0.1 no prescale	OK	OK	OK	FAIL	MISMATCH [ps]	68	dir	testAthenaModernRDO_lumi0.1_no_prescale_test.log
AthenaModernRDO noEF	OK	OK	OK	OK	MATCH	0	dir	testAthenaModernRDO_noEF_test.log
AthenaModernRDO standalone	OK	OK	OK	N/A	MATCH	64	dir	testAthenaModernRDO_standalone_test.log
AthenaModernRDOtoBS	OK	OK	OK	OK	MATCH	0	dir	testAthenaModernRDOtoBS_test.log
AthenaModernRDOtoCBNT	OK	OK	OK	FAIL	N/A	64	dir	testAthenaModernRDOtoCBNT_test.log
AthenaXMLConfigRDO	OK	OK	OK	FAIL	N/A	64	dir	testAthenaXMLConfigRDO_test.log
					MISMATCH			

Current validation infrastructure

- RTT tests:
  - Some 40 tests running on ~1000 events each
  - Run same job options as ATN for cross-checking
  - Also produce data for memory and CPU time monitoring



[Show]

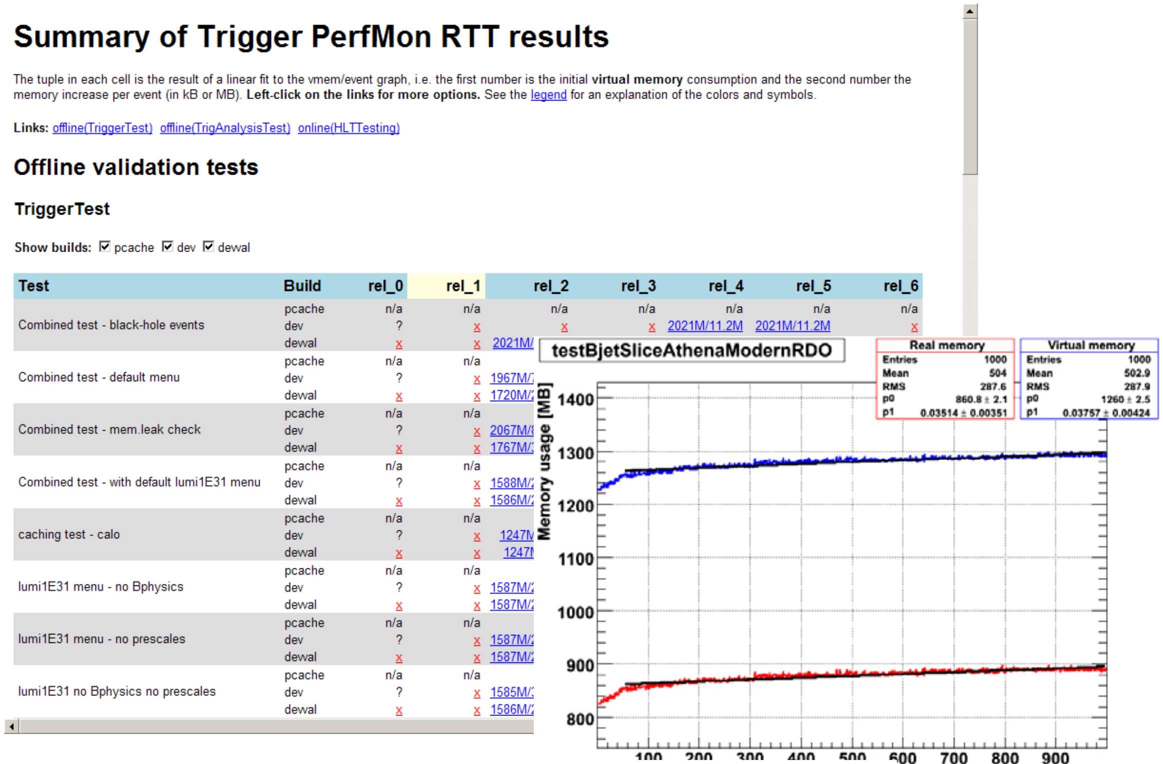
TriggerTest [-00-01-75][CVS]
[done]

RTT jobs [Hide] | **Run by RTT** | **Jobs OK:** 20/24 | **Jobs Done:** 24/24 | **Tests OK:** 0/20

Name	JobGroup	Job status	PostProcessing status	Datasets	Hash
Combined test - with default lumi1E31 menu [id:211]	AthenaTriggerTestMonitorHistos	success	no tests	[Show]	1323382052
testBphysicsSliceAthenaModernRDO [id:216]	AthenaTriggerTestMonitorHistos	error	no tests	[Show]	1702054109
testMETSliceAthenaModernRDO [id:224]	AthenaTriggerTestMonitorHistos	success	no tests	[Show]	-257168710
testBphysicsSliceAthenaModernRDO [id:218]	AthenaTriggerTestMonitorHistos	success	no tests	[Show]	1022222991
testBphysicsSliceAthenaModernRDO [id:219]	AthenaTriggerTestMonitorHistos	success	no tests	[Show]	208699629
Combined test - black-hole events [id:209]	AthenaTriggerTestMonitorHistos	success	no tests	[Show]	1573940966
Combined test - default menu [id:208]	AthenaTriggerTestMonitorHistos	timedOut	unavailable	[Show]	1465974074
testJetSliceAthenaModernRDO [id:223]	AthenaTriggerTestMonitorHistos	success	no tests	[Show]	- 1821959932
testElectronSliceAthenaModernRDO	AthenaTriggerTestMonitorHistos	success	no tests	[Show]	

Current validation infrastructure

- PerfMon metrics:
- Use data from RTT test jobs to monitor memory consumption and CPU time per algorithm



Summary of Trigger PerfMon RTT results

The tuple in each cell is the result of a linear fit to the vmem/event graph, i.e. the first number is the initial **virtual memory** consumption and the second number the memory increase per event (in kB or MB). **Left-click on the links for more options.** See the legend for an explanation of the colors and symbols.

Links: offline(TriggerTest) offline(TrigAnalysisTest) online(HLTTesting)

Offline validation tests

TriggerTest

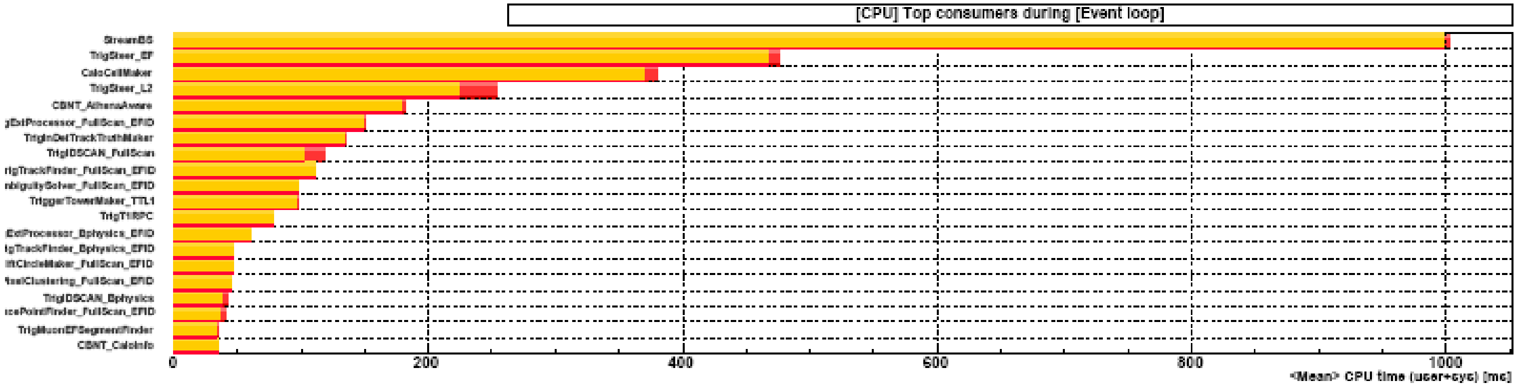
Show builds: ☑ pcache ☑ dev ☑ dewal

Test	Build	rel_0	rel_1	rel_2	rel_3	rel_4	rel_5	rel_6
Combined test - black-hole events	pcache	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	dev	?	x	x	x	2021M/11.2M	2021M/11.2M	x
	dewal	x	x	2021M/				
Combined test - default menu	pcache	n/a	n/a					
	dev	?	x	1967M/				
	dewal	x	x	1720M/				
Combined test - mem.leak check	pcache	n/a	n/a					
	dev	?	x	2067M/				
	dewal	x	x	1767M/				
Combined test - with default lumi1E31 menu	pcache	n/a	n/a					
	dev	?	x	1588M/				
	dewal	x	x	1586M/				
caching test - calo	pcache	n/a	n/a					
	dev	?	x	1247M				
	dewal	x	x	1247M				
lumi1E31 menu - no Bphysics	pcache	n/a	n/a					
	dev	?	x	1587M/				
	dewal	x	x	1587M/				
lumi1E31 menu - no prescales	pcache	n/a	n/a					
	dev	?	x	1587M/				
	dewal	x	x	1587M/				
lumi1E31 no Bphysics no prescales	pcache	n/a	n/a					
	dev	?	x	1585M/				
	dewal	x	x	1586M/				

testBjetSliceAthenaModernRDO

Real memory	
Entries	1000
Mean	504
RMS	287.6
p0	860.8 ± 2.1
p1	0.03514 ± 0.00351

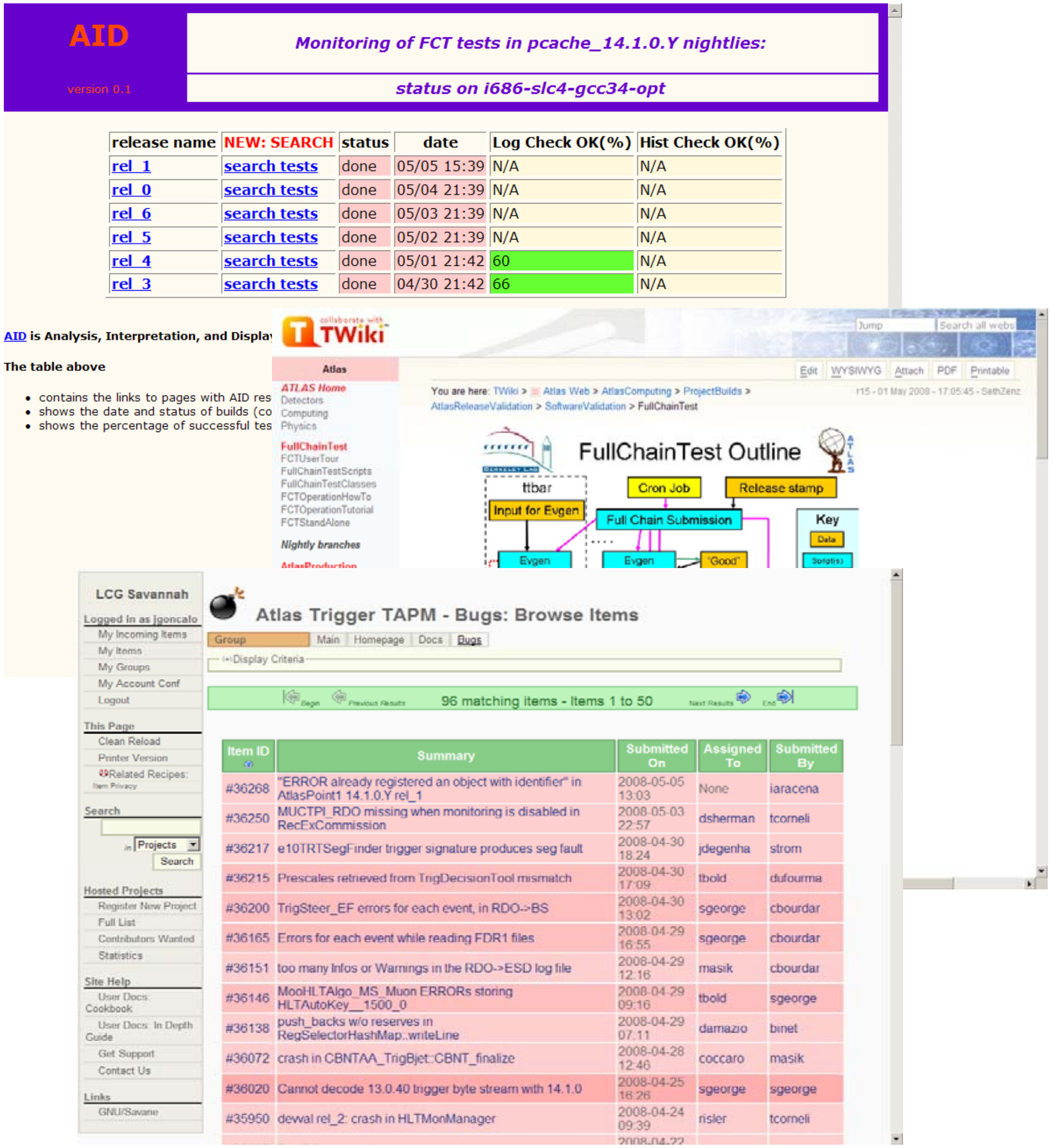
Virtual memory	
Entries	1000
Mean	502.9
RMS	287.9
p0	1260 ± 2.5
p1	0.03757 ± 0.00424



Current validation infrastructure

- Full-chain tests:
  - Run by the offline sofware validation
  - Runs on 200 events from generation to reconstruction
  - Usually last line of checks done before building a release or pcache

- Savannah (bug tracker):
  - Very useful to communicate and classify bugs



AID

version 0.1

Monitoring of FCT tests in pcache_14.1.0.Y nightlies:

status on i686-slc4-gcc34-opt

release name	NEW: SEARCH	status	date	Log Check OK(%)	Hist Check OK(%)
rel_1	search tests	done	05/05 15:39	N/A	N/A
rel_0	search tests	done	05/04 21:39	N/A	N/A
rel_6	search tests	done	05/03 21:39	N/A	N/A
rel_5	search tests	done	05/02 21:39	N/A	N/A
rel_4	search tests	done	05/01 21:42	60	N/A
rel_3	search tests	done	04/30 21:42	66	N/A

AID is Analysis, Interpretation, and Displa

The table above

- contains the links to pages with AID res
- shows the date and status of builds (co
- shows the percentage of successful tes

collaborate with TWiki

Jump | Search all webs

Atlas

ATLAS Home
Detectors
Computing
Physics

FullChainTest
FCTUserTour
FullChainTestScripts
FullChainTestClasses
FCTOperationHowTo
FCTOperationTutorial
FCTStandAlone

Nightly branches

AtlasProduction

Edit | WYSIWYG | Attach | PDF | Printable

You are here: TWiki > Atlas Web > AtlasComputing > ProjectBuilds > AtlasReleaseValidation > SoftwareValidation > FullChainTest

r15 - 01 May 2008 - 17:05:45 - SethZenz

LCG Savannah

Logged in as jgoncalo
My Incoming Items
My Items
My Groups
My Account Conf
Logout

This Page
Clean Reload
Printer Version
Related Recipes:
Item Privacy

Search
in Projects
Search

Hosted Projects
Register New Project
Full List
Contributors Wanted
Statistics

Site Help
User Docs: Cookbook
User Docs: In Depth Guide
Get Support
Contact Us

Links
GNU/Savane

Atlas Trigger TAPM - Bugs: Browse Items

Group | Main | Homepage | Docs | Bugs

(+) Display Criteria

Begin | Previous Results | 96 matching items - Items 1 to 50 | Next Results | End

Item ID	Summary	Submitted On	Assigned To	Submitted By
#36268	"ERROR already registered an object with identifier" in AtlasPoint1 14.1.0.Y rel_1	2008-05-05 13:03	None	iaracena
#36250	MUCTPI_RDO missing when monitoring is disabled in RecExCommission	2008-05-03 22:57	dsherman	tcorneli
#36217	e10TRTSegFinder trigger signature produces seg fault	2008-04-30 18:24	jdegenha	strom
#36215	Prescales retrieved from TrigDecisionTool mismatch	2008-04-30 17:09	tbold	dufourma
#36200	TrigSteer_EF errors for each event, in RDO->BS	2008-04-30 13:02	sgeorge	cbourdar
#36165	Errors for each event while reading FDR1 files	2008-04-29 16:55	sgeorge	cbourdar
#36151	too many Infos or Warnings in the RDO->ESD log file	2008-04-29 12:16	masik	cbourdar
#36146	MooHLTAlgo_MS_Muon ERRORs storing HLTAutoKey__1500_0	2008-04-29 09:16	tbold	sgeorge
#36138	push_backs w/o reserves in RegSelectorHashMap::writeLine	2008-04-29 07:11	damazio	binet
#36072	crash in CBNTAA_TrigBjet::CBNT_finalize	2008-04-28 12:46	coccaro	masik
#36020	Cannot decode 13.0.40 trigger byte stream with 14.1.0	2008-04-25 16:26	sgeorge	sgeorge
#35950	dewal rel_2: crash in HLTMonManager	2008-04-24 09:39	risler	tcorneli
		2008-04-22		

- Also:
  - Several scripts and tools used for specific tasks
  - Analysis-type jobs from each slice to validate “Sample A” – first sample produced, for validation, after each pcache is built

- Nightly tests run on dedicated farm (RTT, FCT) and build machines (ATN)

- Some 15-20 people directly involved both at CERN and remotely:
  - In weekly shifts to check test results
  - In maintaining the tests

- Again, possibly in common with DQM:
  - Lots of information to digest!
  - It takes time to dig into the available information and find cause of problems
    - More so when there are several problems…
    - Easier after a few days practice – what problems are urgent and what to ignore
  - It takes time and a few mistakes to reasonably cover vector space of possible problems
  - The biggest improvements have been on ways to display the information
  - Training/documentation effort not negligible, even among trigger experts

Validation plans

- Need “sign off” procedures for new releases?

E.g.:

1. Offline tests ok: no crashes, memory leaks, memory consumption, trigger code performance, EDM size on disk …
2. Quasi-online tests ok: AthenaMT/PT, compare trigger decision
3. Online tests ok: run on real data in preseries, memory consumption, “rate”
4. Stress tests ok: test against bad data, busy data, etc

- For commissioning:
  - Deploy – find bugs – fix bugs – deploy iteration needed
  - Later cannot be afforded often in real system

Validation plans

- What about new menus? What needs to be checked before going online?
  - Rate/overlap between slices can be best measured on L1 accepted real data

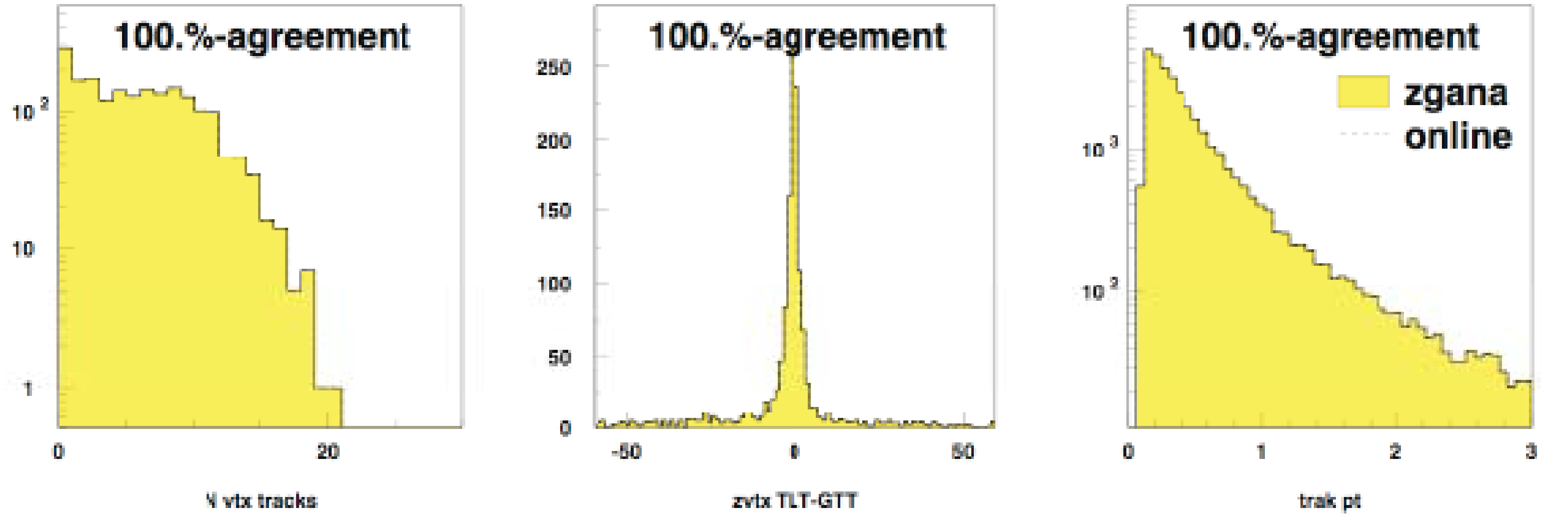
- Would it be useful to have formal release notes?
  - How to find what problems were present in the code after 2 years? Complement with Savannah numbers for bug details

Validation plans

- Build up test samples:
  - Save bad data!
    - Debug stream data would be very useful to test new code (after good events are removed)
    - Data taken under special beam conditions or enriched in beam-related backgrounds would be useful
  - Save busy data!
    - High-lumi data from beginning of fill may be useful to develop next set of menus (D0, CDF)
  - Build up samples passing L1 and passing L2:
    - Needed to test L2 and EF, estimate rates for new menus etc

Validation plans

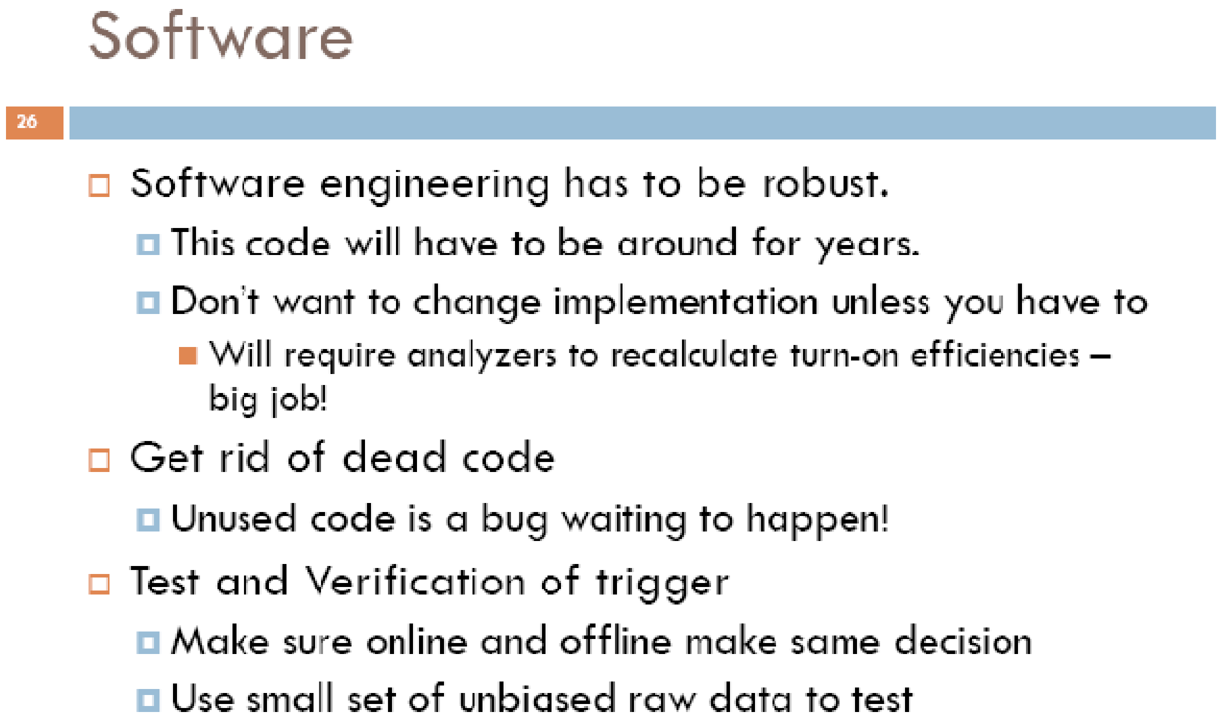
- Test L1/HLT simulation (is this monitoring or validation?):
  - Run trigger simulation in the CAF and compare with online results (event by event)
  - L1 hardware/CTP won’t change frequently, but beam/detector conditions will
  - HLT in CAF is immune to problems in DAQ buffers.
  - Conditions data? (same as online? Different?)



M.Sutton, Trigger Robustness Workshop, 4/3/08

Validation plans

- Learn from others’ experiences



Software

26

- □ Software engineering has to be robust.
  - □ This code will have to be around for years.
  - □ Don't want to change implementation unless you have to
    - ■ Will require analyzers to recalculate turn-on efficiencies – big job!
- □ Get rid of dead code
  - □ Unused code is a bug waiting to happen!
- □ Test and Verification of trigger
  - □ Make sure online and offline make same decision
  - □ Use small set of unbiased raw data to test

G.Watts, Trigger Robustness Workshop, 4/3/08

Tools…

- What tools do we have?
  - Current validation tools provide a global health check of the release
  - New checks are added as new problems show up (we will know which problems are important when Atlas comes alive)
  - Current tools still too demanding in maintenance and in time needed for checking: there are plans for improvements

- What tools do we need?
  - Extracting trigger decision and data from BS in CAF (Christiane’s talk)
  - Re-running trigger on real events and comparing outcome
  - Extracting events from debug stream

- Would be good to share some of the above: can’t afford to duplicate efforts

Conclusions

- Essential to have validated code
  - Know what to expect from code, to know when something is not right
- Validation and monitoring can be symbiotic in several ways
  - Good definition of tasks and communication needed to avoid duplication of efforts
- The need for new validation tests will become apparent with real data:
  - The whole range of possible problems should be covered
  - The most important performance metrics will be indicated by real data