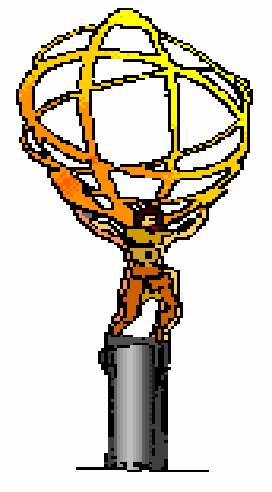
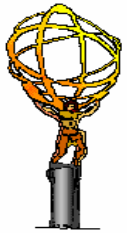
Software Validation Infrastructure for the ATLAS Trigger

Wolfgang Ehrenfeld – DESY

On behalf of the ATLAS Trigger Validation Group



CHEP 2009 – Prague – 26th March 2009

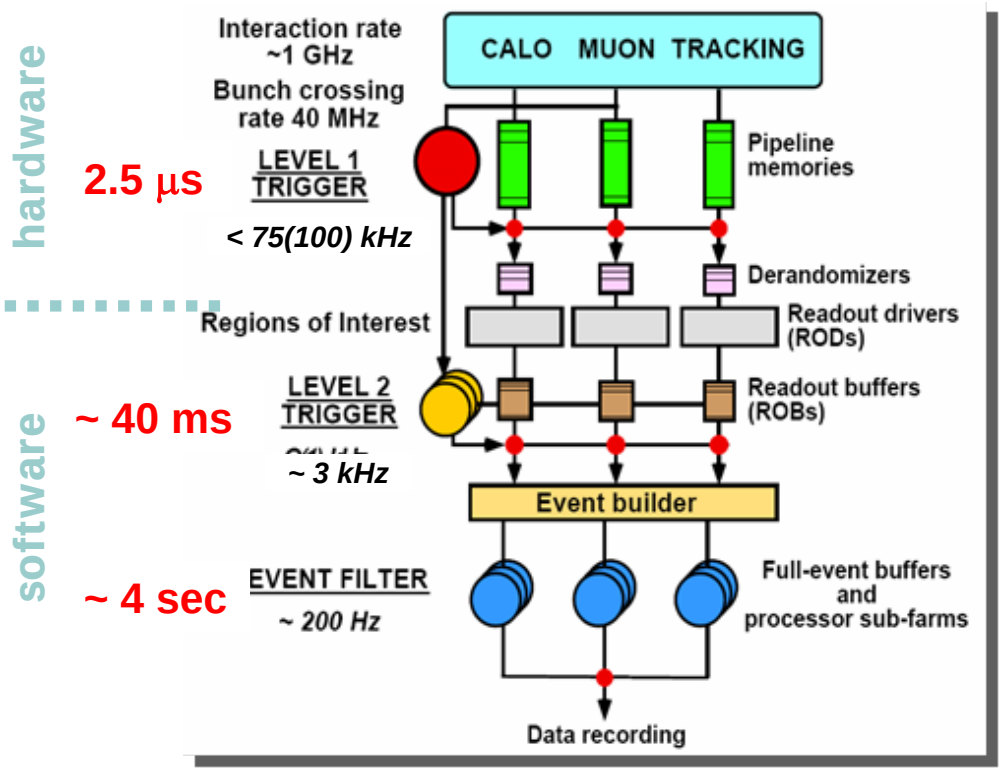


Content

- **introduction**
  - ATLAS trigger
  - ATLAS trigger software project
- **validation**
  - ATLAS nightly build system
  - automatic testing and monitoring
- **new developments**
  - trigger software validation dashboard
- **summary**

ATLAS Trigger: Overview

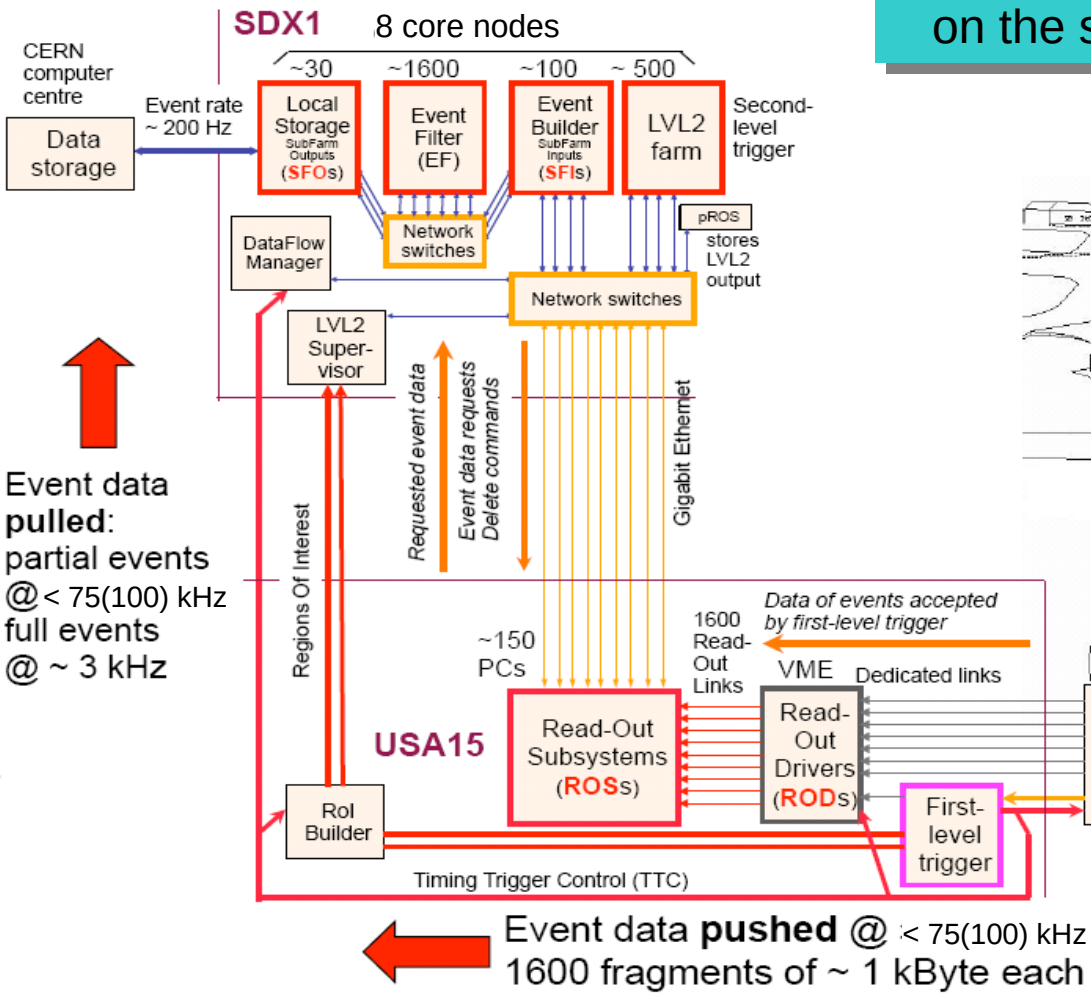
3-Level Trigger System:

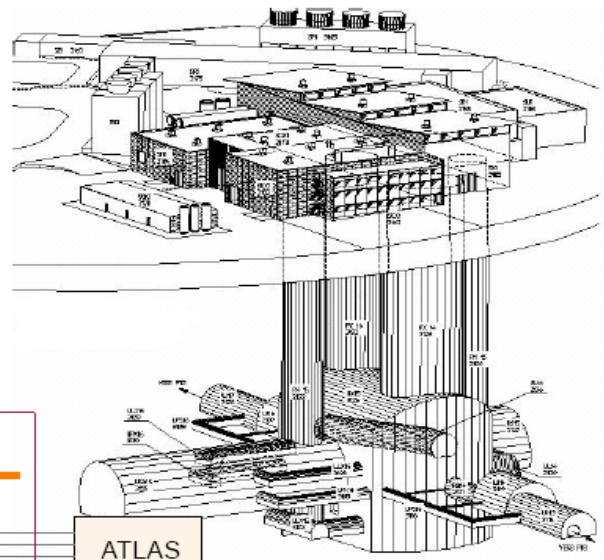


1. **LVL1** decision based on data from calorimeters and muon trigger chambers; synchronous at 40 MHz; bunch crossing identification
2. **LVL2** uses Regions of Interest (identified by LVL1) data (ca. 2%) with full granularity from all detectors
3. **Event Filter** has access to full event and can perform more refined event reconstruction

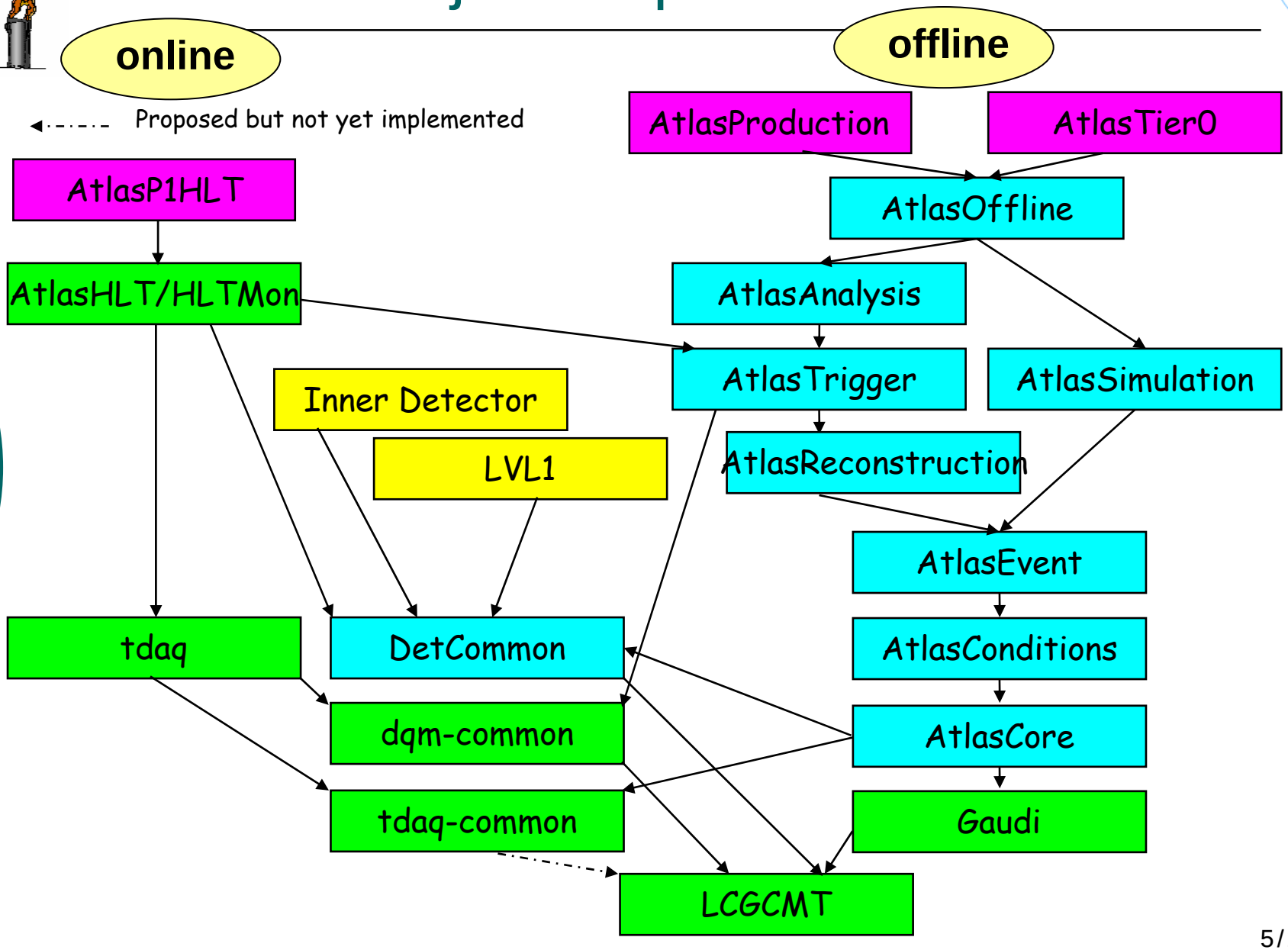
ATLAS Trigger & DAQ Architecture

- LVL2 and EF run in large PC farms on the surface





ATLAS Project Dependencies



Trigger Software Project

- **project details:**
  - developers: ~100
  - number of packages: ~250
  - distributed over many build projects: DetCommon, AtlasEvent, AtlasConditions, AtlasTrigger, AtlasAnalysis
- **used for**
  - online running at L2, EF
  - offline development and simulation for L1, L2, EF
- **requirements:**
  - stable running on the online farms
    - ~16800 cores
    - run period of ~12 hours
  - Stable running for offline studies and MC production on the GRID

AtlasTrigger project size

Language	files	blank	comment	code lines
XML	300	1594	1405	1234724
C++	1651	60225	39194	232603
Python	996	22110	20055	80010
C/C++ Header	1892	29852	38813	66338

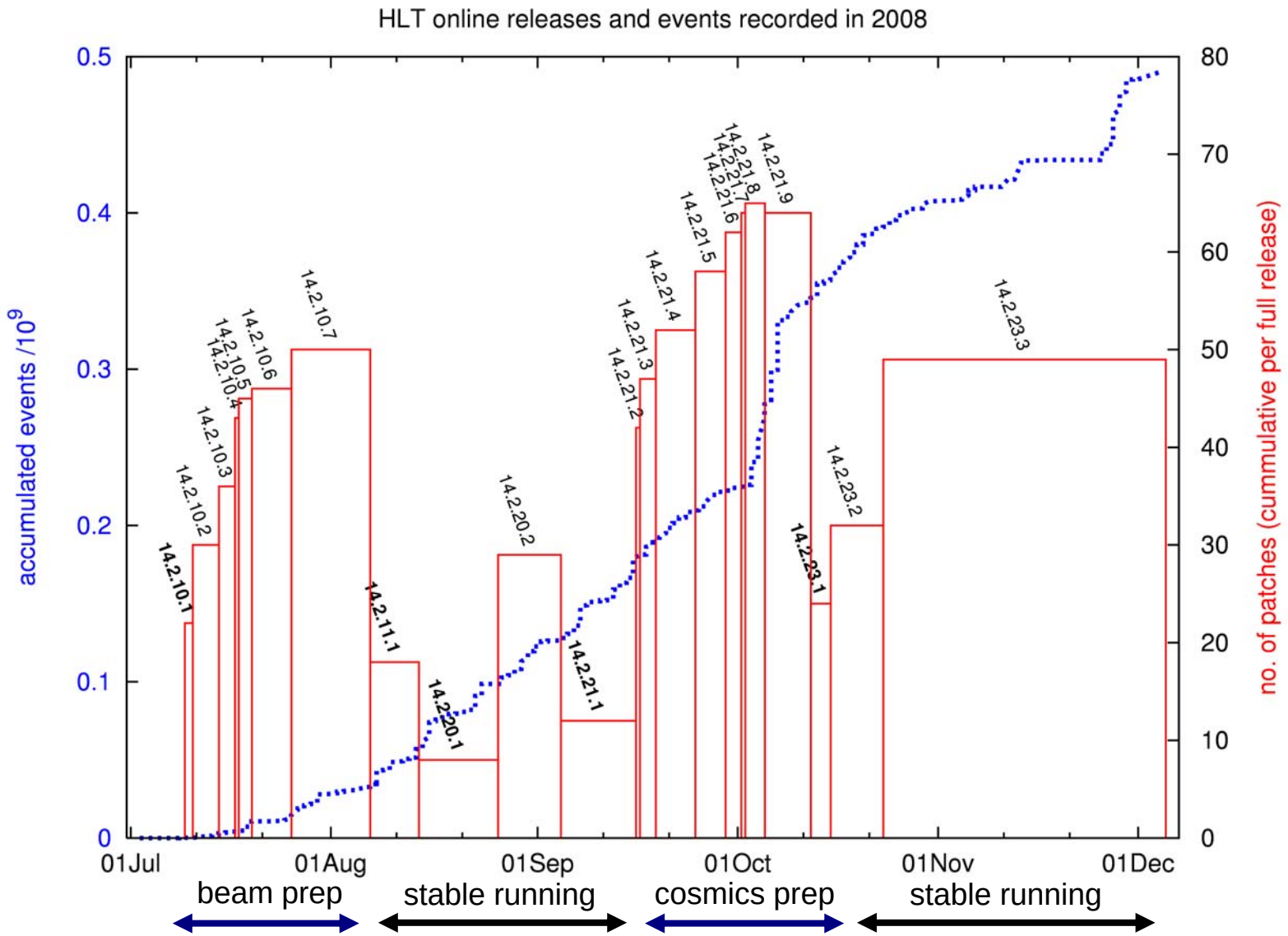
(http://cloc.sourceforge.net/)

→ ~10% of full ATLAS software

Validation goals:

deliver high quality software for stable online running, while continuous evolution of trigger code is expected to improve trigger operation and performance

Code Development during Cosmic Data Taking



Steps until Successful Data Taking

Trigger software development:
>250 packages, >100 developers
Complex software environment
Strict time and memory requirements

Software functionality
& performance validation
For online use in real time
And offline simulation & analysis

Online framework integration
Tests in realistic environment
before online deployment

Online running
Filter complex event topologies
Select 5 events for 10^6 collisions

Performance monitoring
Online in real time
Offline verification & analysis

ATLAS Trigger Validation Group

- **trigger validation group:**
  - ~40 members from different trigger areas (L1, L2, EF, different slices, steering)
  - main focus on daily validation: shifts of 4 hours/day for 1 week
    - monitor release status
    - locate problems and bugs
    - submit bug reports via Savannah
    - daily status for developers, release coordination and users
  - develop tools based on common ATLAS tools for easier validation work
- **goals:**
  - big picture: release
    - ensure, that the trigger part of a release or cache is fully functional
  - small picture: package
    - if a package fails (build, config, run time), identify problem and notify developer via Savannah bug tracker
  - unify validation (frame)work within trigger community to save man power
- **why?**
  - developers scope for testing is limited (code compiles and runs, some tests run)
  - integration with other packages (within AtlasTrigger and from other projects)
  - some problems only show up after processing a large number of events

ATLAS Release Structure

- **structure for given release line:**
  - Base release, e.g. 14.2.25
  - Production cache: mainly bug fixes needed for MC production (GRID)
  - T0 cache: mainly bug fixes for data taking (P1) / first reconstruction (T0)
- **current release lines:**
  - MC production
  - cosmic reprocessing
  - External package validation, e. g. Geant4
  - development for data taking
- **validation needs to ensure correct**
  - release or cache built
  - run time configuration
  - run time execution
  - physics correctness

NIghtly COntrol System (NICOS)

provided by ATLAS SIT

Nightly Title	# Platforms	# Projects	Latest Rel.	Build	Date	Copy	Kit	Ave. Failed Builds	Ave. Test Success(%)
MAJOR NIGHTLIES									
15.X.0	6	10	rel_1	done	03/16 07:41	done*	rel_1	3.4	54.8
15.X.0-PEN	1	10	rel_2	done	03/16 15:57	N/A	N/A	0	N/A
15.X.0-VAL	4	10	rel_1	done	03/16 10:17	done with errors	N/A	0.3	61.7
BUGFIX NIGHTLIES									
15.0.X	2	10	rel_1	done					
15.0.X-VAL	2	10	rel_1	done					
PATCH NIGHTLIES									
14.2.OLD.Y-Prod	1	1	rel_0	done					
14.2.OLD.Y-VAL-Prod	1	1	rel_0	done	03/16 00:19	done	rel_0	0	N/A
14.5.2.Y-Prod	1	1	rel_1	done	03/15 23:03	done	rel_1	0	66.0
14.5.2.Y-VAL-Prod	1	1	rel_1	done	03/16 05:33	done	rel_1	0	58.0
14.5.X.Y-Prod	1	1	rel_1	done	03/16 11:51	done	rel_1	0	83.0
14.5.X.Y-T0	1	1	rel_1	done	03/16 07:18	done	rel_1	0	85.0
					11:15	done	rel_1	0	83.0
					19:43	done	rel_1	0	87.0
						done	rel_1	0	N/A
					16:51	done	rel_1	0	14.0

Entry page to nightly build system:
- list all active builds (release, caches)
- summary of build status

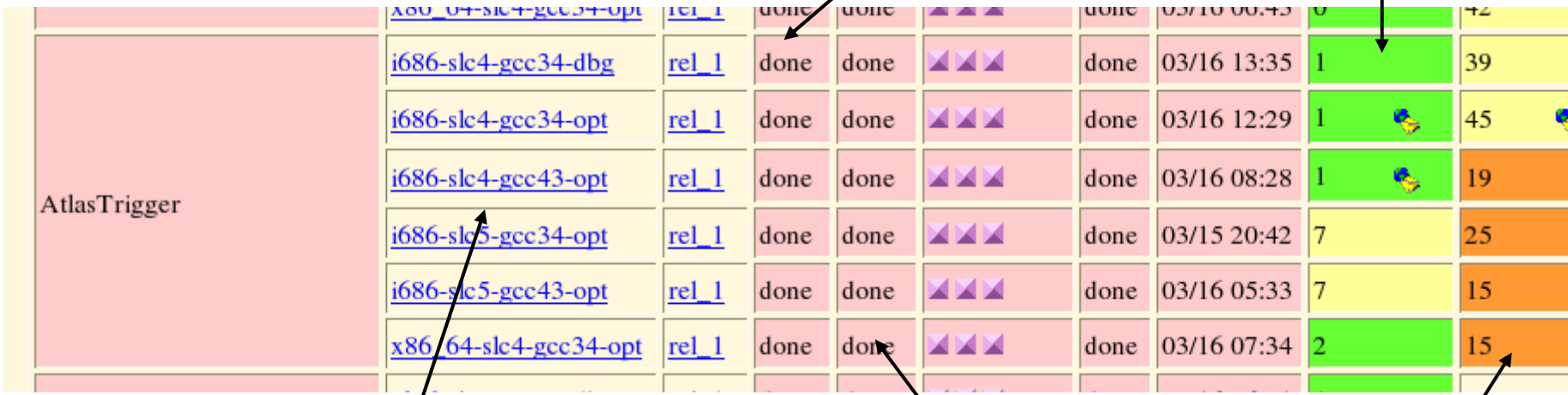
Build full ATLAS software every night:
- list build status per package and project
- run unit tests after package build
- run full tests after project build

NICOS – Project Summary

Summary of project build:
- list all build combinations
- summary of build status

build status

failed builds (packages)



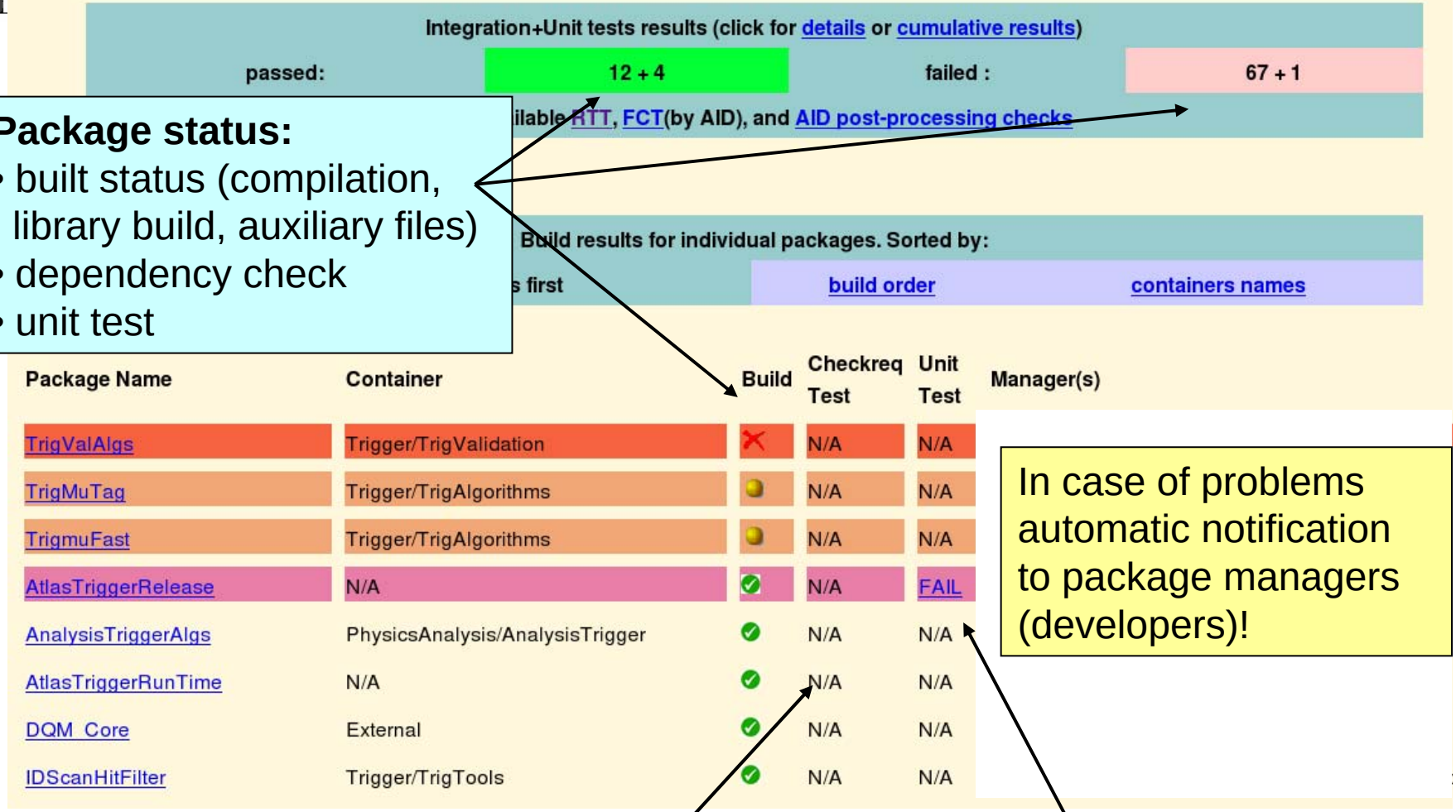
AtlasTrigger	i686-slc4-gcc34-dbg	rel_1	done	done		done	03/16 13:35	1	39
	i686-slc4-gcc34-opt	rel_1	done	done		done	03/16 12:29	1	45
	i686-slc4-gcc43-opt	rel_1	done	done		done	03/16 08:28	1	19
	i686-slc5-gcc34-opt	rel_1	done	done		done	03/15 20:42	7	25
	i686-slc5-gcc43-opt	rel_1	done	done		done	03/16 05:33	7	15
	x86_64-slc4-gcc34-opt	rel_1	done	done		done	03/16 07:34	2	15

architecture (32/64 bit),
operating system (SLC4/5),
compiler version (3.4/4.3),
compiler optimization (opt/debug)

test status

successful tests (fraction)

NICOS – Package Build for AtlasTrigger



Integration+Unit tests results (click for details or cumulative results)

passed: 12 + 4 failed : 67 + 1

ilable RTT, FCT(by AID), and AID post-processing checks

Build results for individual packages. Sorted by:

s first | build order | containers names

Package Name	Container	Build	Checkreq Test	Unit Test	Manager(s)
TrigValAlgs	Trigger/TrigValidation	✗	N/A	N/A	
TrigMuTag	Trigger/TrigAlgorithms	●	N/A	N/A	
TrigmuFast	Trigger/TrigAlgorithms	●	N/A	N/A	
AtlasTriggerRelease	N/A	✓	N/A	FAIL	
AnalysisTriggerAlgs	PhysicsAnalysis/AnalysisTrigger	✓	N/A	N/A	
AtlasTriggerRunTime	N/A	✓	N/A	N/A	
DQM_Core	External	✓	N/A	N/A	
IDScanHitFilter	Trigger/TrigTools	✓	N/A	N/A	

Package status:
- built status (compilation, library build, auxiliary files)
- dependency check
- unit test

In case of problems automatic notification to package managers (developers)!

dependency check: run only on dedicated builds

unit test: not widely used

AT Night Testing (ATN)

provided by ATLAS SIT

- **ATN allows to run custom jobs with full project framework on *O*(10) events: test run time behavior**
  - technical validation:
    - run time, e.g. accessing null pointers, infinite loops, …
    - configuration (python based): missing or broken run time configuration
  - regression test: is the code doing the same as before
  - automatically run after project build finished; standalone running possible
- **common infrastructure supplied by trigger validation group**
  - configuration and runtime environment
  - available tests: regression tests on log files and histograms
  - collection and presentation of status information
- **developer needs to supply:**
  - job configuration for test job
  - hooks for regression test; fill monitoring histograms
  - references
- **enough information available to easily find and diagnose problems**
- **sometimes too much information for all available releases/builds**
  - meta summary needed

ATN – Test Results

Test name	Test script	Athena exit	Error Msgs	Reg. tests	Rootcomp	Time(s)	Exit code	Post cmd	Dir. link	Log link
AthenaDBConfigRDO	OK	OK	FAIL	OK	N/A	195	32/32	N/A	dir	[tail] testAthenaDBConfigRDO_test.log
AthenaModernBS_standalone	OK	OK	OK	OK	MISMATCH [ps]	201	6/0	N/A	dir	[tail] testAthenaModernBS_standalone_test.log
AthenaModernRDO	OK	OK	OK	OK	MISMATCH [ps]	1034	6/0	N/A	dir	[tail] testAthenaModernRDO_test.log
AthenaModernRDO_blackholes	OK	TIMEOUT	FAIL	FAIL	MISMATCH [ps]	1539	230/32	N/A	dir	[tail] testAthenaModernRDO_blackholes_test.log
AthenaModernRDO_full	OK	OK	OK	OK	MISMATCH [ps]	272	6/0	N/A	dir	[tail] testAthenaModernRDO_full_test.log
AthenaModernRDO_full_no_Bphysi	OK	OK	OK	OK	MISMATCH [ps]	296	6/0	N/A	dir	[tail] testAthenaModernRDO_full_no_Bphysics_test.log
AthenaModernRDO_full_no_Bphysics_no_prescale	OK	OK	OK	OK	MISMATCH [ps]	272	6/0	N/A	dir	[tail] testAthenaModernRDO_full_no_Bphysics_no_prescale_test.log
AthenaModernRDO_full_no_prescale	OK	OK	OK	OK	MISMATCH [ps]	274	6/0	N/A	dir	[tail] testAthenaModernRDO_full_no_prescale_test.log
AthenaModernRDO_full_triggerLast	OK	OK	OK	OK	MISMATCH [ps]	564	4			odernRDO_full_test.log
AthenaModernRDO_lumi0.01	OK	OK	OK	OK	MISMATCH [ps]	212	6			odernRDO_lumi0.01_test.log.gz
				OK	MISMATCH [ps]	262	6/0	N/A	dir	[tail] testAthenaModernRDO_lumi0.01_no_Bphysics_test.log
				OK	MISMATCH [ps]	269	6/0	N/A	dir	[tail] testAthenaModernRDO_lumi0.01_no_Bphysics_no_prescale_test.log
				OK	MISMATCH [ps]	240	6/0	N/A	dir	[tail] testAthenaModernRDO_lumi0.01_no_prescale_test.log

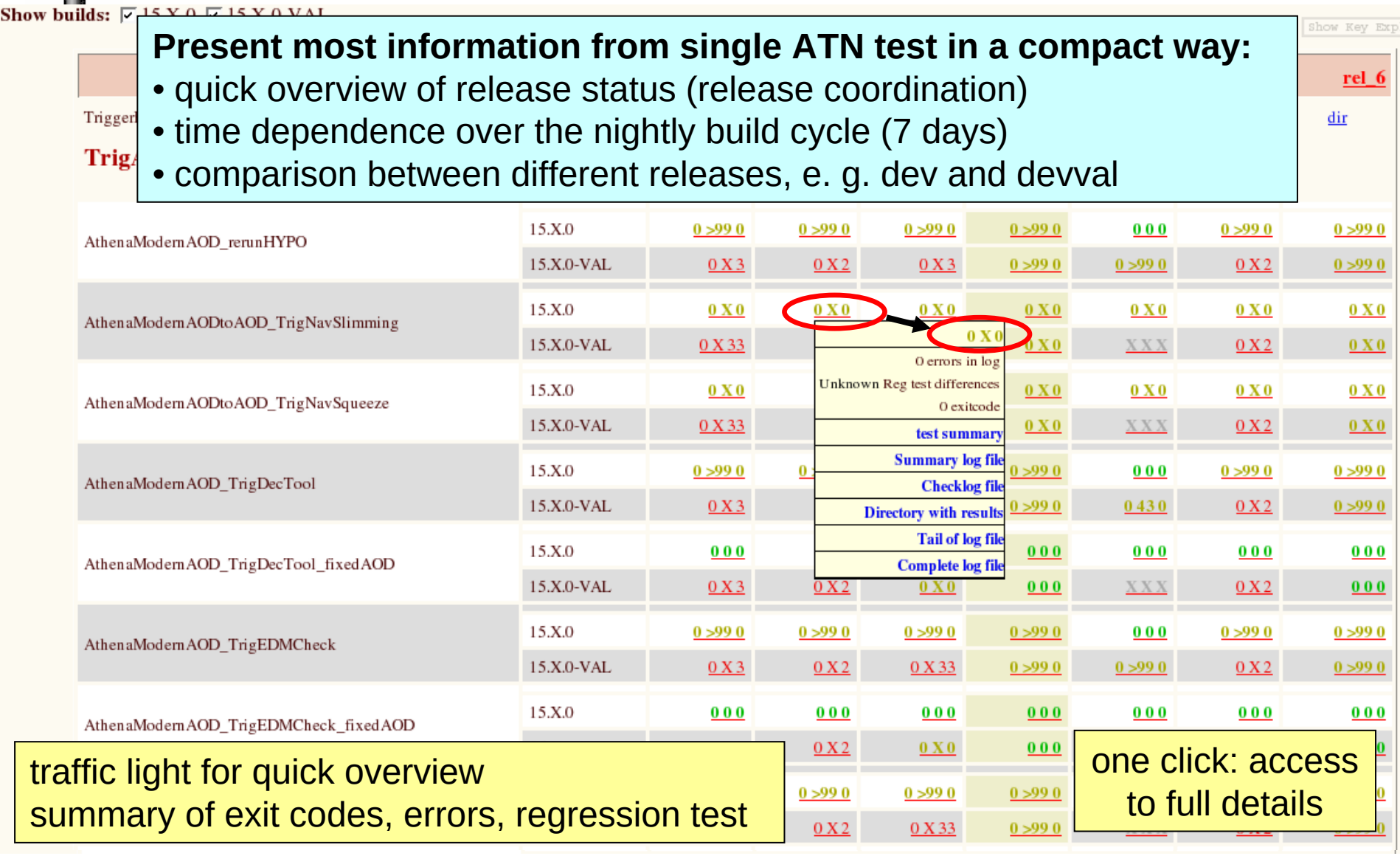
log file regression test

histogram comparison

regression and histogram tests essential to check expected behavior before release deadline

ATN – Meta Summary

Present most information from single ATN test in a compact way:
- quick overview of release status (release coordination)
- time dependence over the nightly build cycle (7 days)
- comparison between different releases, e. g. dev and devval



traffic light for quick overview
summary of exit codes, errors, regression test

one click: access to full details

Run Time Tester (RTT)

provided by ATLAS SIT

See talk by B. Simmons:
Id 140: 26.3.2009, 15:20

- **RTT allows to run over *O*(1000) events in an automated process similar to ATN for every nightly build and releases**
  - accumulate higher statistics
    - histogram based physics validation
    - find run time problems with rare conditions
  - memory monitoring
  - CPU time performance
  - floating point exceptions
  - redundancy

40k minimum bias events:
• 1 sec of data taking
1k top pair events:
• multi purpose sample
Other special samples
• e. g. B physics

- **results, log and intermediate files (configurable) are available on the web**

- **tests**
  - part of ATN tests are run in RTT
  - additional tests, which doesn't make sense on *O*(10) events

Memory Monitoring

- **any kind of memory leak has a significant effect on the total memory budged:**
  - L2 example: 500 nodes a 8 cores, 75 kHz → ~20 events/core/s
    1 GB/core installed, ~800 MB/application needed → 200 MB margin
    - 1 byte → 20 B leak/s → ~70 kB/h → ~1 MB/run(12h)
    - 10 byte → ~10 MB/run(12h)
    - 100 byte → ~100 MB/run(12h)
  - high input rate and long online time makes any memory leak a problem

→ memory monitoring is essential

- **aim:**
  - L2: memory leak below 10 B/event
  - EF: memory leak below 1 kB/event

Memory Monitoring

TriggerTest

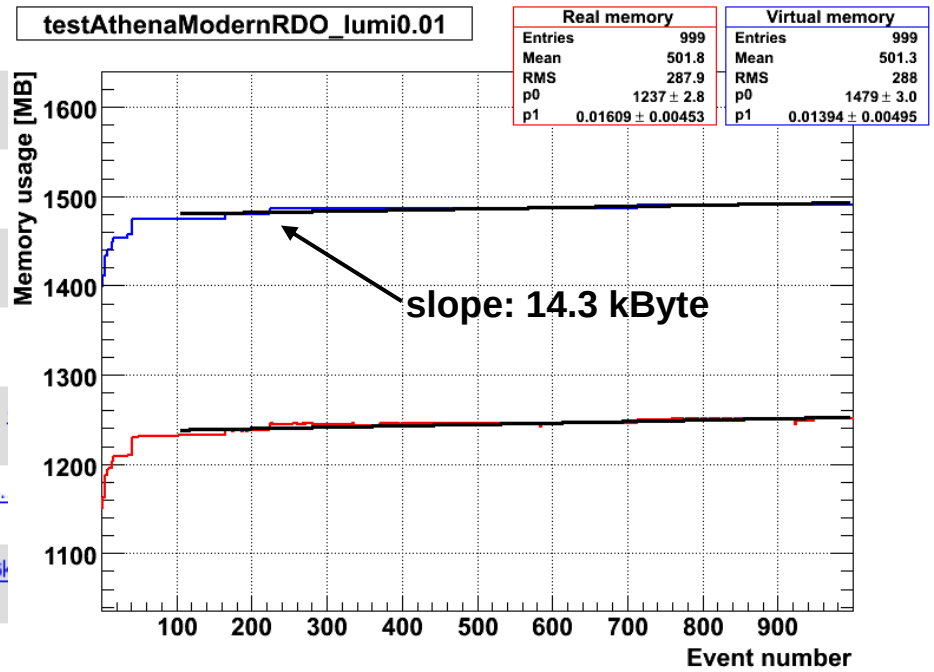
Show builds: ☐ 14.5.X.Y-VAL ☑ dev ☑ devval

Total memory usage and per event leak

Test	Build	rel_0				rel_4	rel_5	rel_6
Combined test - black-hole events	dev	2459M/7.7M	2296M/16.3M	2485M/7.6M	x	x	2489M/7.6M	2488M/7.7M
	devval	2297M/16.3M		? 2480M/7.7M	x	x	2456M/7.7M	2403M/14.3M
Combined test - default menu	dev	?						
	devval	1717M/316.0k						
Combined test - full menu	dev	1801M/56.8k						
	devval	?						
Combined test - mem.leak check	dev	1753M/9.4k						
	devval	1729M/6.3k						
Combined test - with default lumi1E31 menu	dev	x						
	devval	1479M/14.3k						
ElectronSliceAthenaModernRDO - single e	dev							
	devval							
LVL1 only test - default menu	dev							
	devval							
caching test - calo	dev							
	devval							
	devval	1550M/269.8k		? 1568M/35.3k	x	x	1557M/275.7k	1557M/275.7k
lumi1E31 no Bphysics no prescales	dev	1556M/274.4k	1553M/271.9k	1570M/35.6k	x	x	1583M/275.9k	x
	devval	1550M/262.8k	?	1568M/36.6k	x	x	x	1557M/274.1k
testBjetSliceAthenaModernRDO	dev	1183M/26.3k	1183M/29.7k	1216M/25.1k	1214M/26.5k	1212M/27.6k	1214M/26.6k	1213M/25.3k

Memory graph
PerfMon results [txt]
PerfMon results [pdf]
Job log file
RTT result page
[All Memory graphs]
status: success
done: 09/03/15 16:08
Post-processing:
DCubeRunner

Access to memory plot and all job information



Numbers close to requirements, but tests are not purely online!

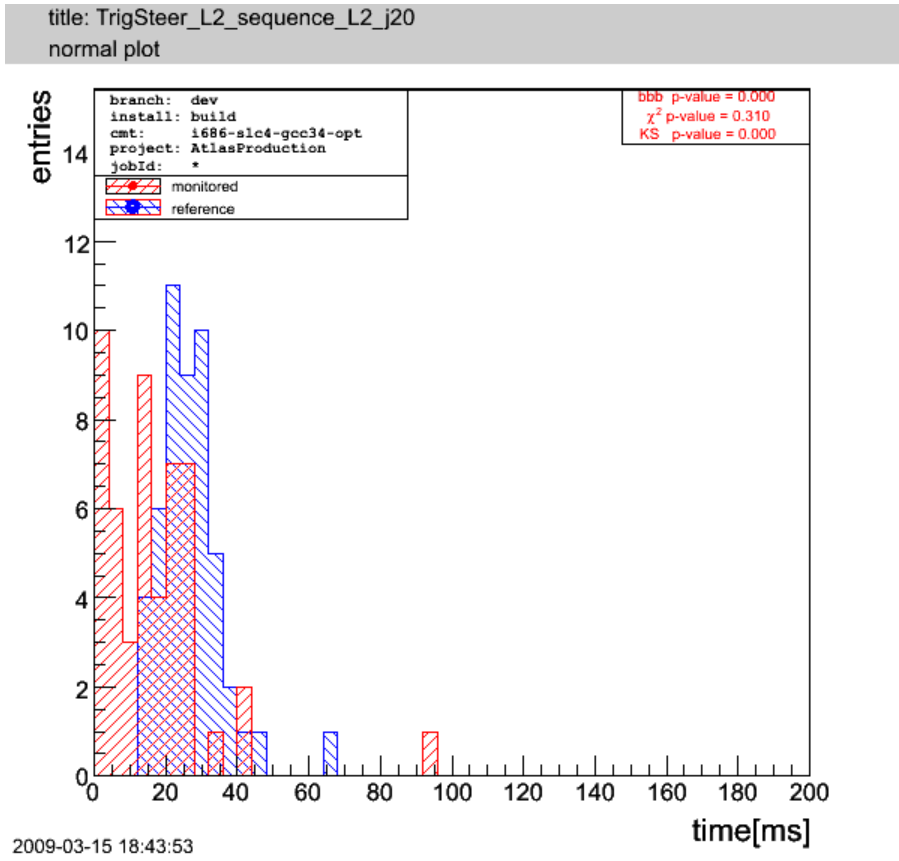
CPU Time Monitoring

- **if the average processing time per event exceeds the limit, trigger dead time will increase and potential Higgs events get rejected**

→ CPU time monitoring is crucial

- **offline CPU time monitoring is more difficult than offline memory monitoring**
  - event mixing → done in technical runs last year
  - CPU specs → online integration/cosmics
  - online/offline environment, e. g. I/O
- **but a few things are possible**
  - per algorithm monitoring (normalization from online running)
  - coarse estimate (no stable test environment with regards to CPU)

L2 jet trigger: p_T > 20 GeV



Advanced Tools in RTT – DCube

provided by ATLAS SIT

- **clearly, single distributions are essential to pin down problems, but a tool for automatic comparison is needed:**

→ **DCube (Data Quality Browser)**

- **infrastructure for automatic histogram processing and comparison, e. g. Kolmogorov-Smirnov, χ^2 or bin-by-bin test**

color code for easy readability

TrigSteer_EF_ChainsTime	0.000	1.000	0.154	!
TrigSteer_EF_LvlConverterTime	0.075	0.000	0.333	!
TrigSteer_EF_MonitoringTime	0.000	0.000	1.000	!
TrigSteer_EF_ResultBuilderTime	0.000	0.000	1.000	!
TrigSteer_EF_TotalTime	0.000	0.540	0.083	!

details / histogram for test

Name: TrigSteer_EF_TotalTime Type: TH1F

	monitored	reference
Entries	50	50
Underflows	0	0
Overflows	40	43
Bins(x)	50	
Mean(x)	1.44e+02±5.99e+00	1.71e+02±7.84e+00
RMS(x)	1.90e+01±4.24e+00	2.07e+01±5.54e+00

Advanced RTT Use Cases

- **RTT is quite a powerful tool for automatic job execution and testing**
- **ideal tool to extent validation to a broader area**
  - test trigger related tools (e.g. DB upload)
  - test more physics related quantities (e.g. trigger counts)
- **within trigger software validation there are 8 test packages:**
  - general trigger test packages: TriggerTest, TrigAnalysisTest, TrigP1Test
  - special purpose test packages: TrigEgammaValidation, TrigInDetValidation, TrigMenuValidation, TrigMinBiasPerf, TrigTauPerformAthena
- **examples:**
  - TrigMenuValidation: load trigger menu from DB, run trigger simulation and compare with results from trigger menu from XML source
  - TrigEgammaValidation: special monitoring of all eγ related trigger variables (heavy use of DCube comparison to spot small differences)

Recent Developments – Dashboard

- **interval of validity:**
  - all information from tests in nightly builds (ATN/RTT) are lost after 7 days (nightly build cycle)
  - information from tests of a full release build always accessible
- **how to do long term monitoring?**
- **trigger software validation dashboard**
  - collect test information for a long time scale using a database
  - visualize results in an easy way on the web, e. g. tables and graphs
  - advanced search and filter capabilities
- **status:**
  - small scale prototype developed and under testing for further

Trigger Software Validation Dashboard

Trigger Dashboard

query interface

Fill the filters to start a new search:

Var Name	Test Name	Release Name	Initial Date
vmem-fit	te		

Platform Name

testAthenaModernRDO
testAthenaModernRDOtoAOD
testAthenaModernRDOtoESDAOD
testCaloAthenaModernRDO
testElectronSliceAthenaModernRDO
testMuonSliceAthenaModernRDO
testTauSliceAthenaModernRDOtoESDAOD

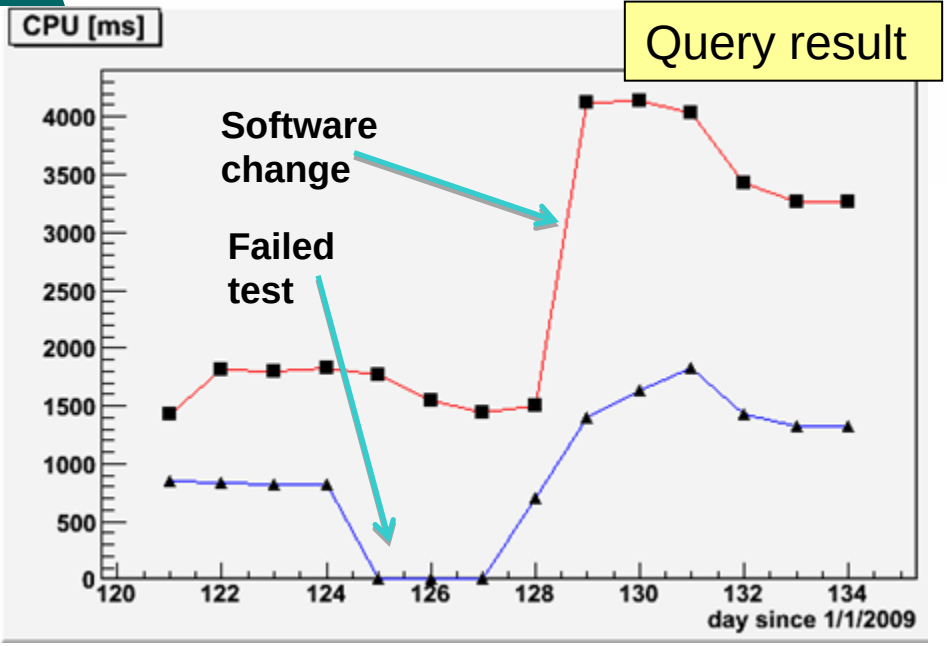
Search

March 2009

Mon	Tue	Wed	Thu	Fri	Sat	Sun
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Today is Wed, 11 Mar 2009

Query result



- **prototype functional**
- **full implementation in near future**

→ **long term validation/monitoring of physics quantities, e.g. trigger rates**

Summary

- **the trigger software project is a substantial software project within ATLAS providing online and offline trigger software**
- **thorough offline validation is an essential step towards achieving a stable online running**

- **offline trigger software validation is done on a shift basis:**
  - constant monitoring of the release status
  - prompt spotting of problems and developer notification
- **offline trigger software validation is done by a group of people:**
  - join efforts in providing common tools and infrastructure
  - relieve developers and release coordination from work load

- **future developments and plans**
  - achieve easy and maintainable validation process
  - more focus on physics validation
  - although development will continue, can not sustain high effort during data taking

Many thanks to the ATLAS Software Infrastructure Team provided by ATLAS SIT

See talk by F. Luehring:
Id 248: 26.3.2009, 16:50