

Software Infrastructure: Status and Plans

Frederick Luehring

Indiana University

September 24, 2004

Software Week Plenary Session 3

Introduction

- We had two sessions for SIT Work
 1. The Software Infrastructure Working Group (Tuesday)
 - Various Policy Issues (e.g. Release Strategy)
 - QA/QC
 - Tag Collector
 - External Packages
 - Librarian's and Release Coordinator's Reports
 - CMT
 - Building: NICOS, incremental builds, building problems
 - Organization of SIT Work
 2. The Distribution Working Group (Wednesday)
 - Software Distribution
 - Automated Testing and Validation of the Software

Release Strategy I

- Our current release strategy:
 - Every ~3 weeks there is a development release (x.y.0)
 - Even though some validation is done after the release is built, bugs found in the release are usually not fixed. Problems are simply noted on a webpage though occasionally problem packages are patched.
 - Distribution kits are usually built for each release and worked on until the kit validation test works. For failed releases, no kit is built.
 - Every ~6 months there is a production release (x.0.0) followed by a series of patches (x.0.z).
 - Intensive validation commences as soon as the release is built and bugs are either fixed by a rebuilding or a patch release. There is extensive ad-hoc validation work leading up to the release build.
 - Distribution kits are always built and worked on until they install and pass the kit validation tests.
- There is discussion on all parts of the release strategy: the number of releases, the spacing of the releases, and the amount and the timing of testing.

Release Strategy II

- I would like to ask everyone (especially the core developers) to respect the deadlines:
 - Core closes a week before release.
 - The tag collector closes two days before the release.
- Last minute core requests happen in most releases.
 - While there often are legitimate reasons for a last minute change, the potential to break the release is always present.
 - Last minute core changes result in a cascade of last minute changes by non-core developers (increasing risk).
 - IMHO some (but not all!) of the last minute core change requests are not justified and could have been avoided.
- The one week core freeze comes from the real need by the non-core developers for a stable environment to complete their work.

Documentation / Help / Bug Tracking

- The web pages are sorely in need of updating.
 - May just remove some of the out of date pages.
 - There is also some user of Wikis.
- Requests to atlas-sw-help list go unanswered.
 - Mailing list names are confusing and many people use atlas-sw-developers instead of atlas-sw-help.
- The SIT Savannah bug group was dormant until recently when Emil has started monitoring it.
 - Many bugs closed recently by Christian.
 - Please use this group it is now active again.
- Need to coordinate the use of Wikis with other help & documentation tools.
 - Lots of partly supported tools.

AMI & Tag Collector

What is done ...

-AMI

- AMI admin layer (AMI admin command) i.e. generic database operations
- AMI user management (see talk on AMI in DB session)

-Tag Collector

- Tag Collector core layer.
 - CVS and CMT proxies. (CVS was tricky!)
 - Core functionalities (lock system, user rights system...)
- Tag Collector commands
- Tag Collector deployment module (install a new Tag Collector...)
- Tag Collector importation tool from the old data structure to the new one.

What is still to do ...

-AMI

- AMI web service with user authentication

-Tag Collector

- Tag Collector Graphic User Interface.

Prototype for November?

QA/QC

- Two products used for this:

1. Rule Checker

- Checks automatically that Atlas coding rules are followed.
- Generates web pages for each package showing findings - each problem is clearly shown.
- Very CPU intensive to run.
- Not easy to get users to take findings seriously.

2. Code Metrics

- Studies “goodness” of code in a statistical way by running Together code analyzing tool night using the Run Time Tester (described later).
- Metrics range from simple like lines of code in a routine to complicated like the total number of paths through a routine.
- Web pages showing results are produced.
- Again difficult to get users to take findings seriously.

Plans for Reviews in 2005

- ❖ We have started a WIKI page to prepare for the forthcoming reviews. It will be updated as the plans become more concrete, and will provide also a framework to help the review team.
- ❖ <https://uimon.cern.ch/twiki/bin/view/Atlas/QualityAssurance>

Building Issues I

- The nightly build keeps growing:
 - It takes more and more space.
 - Since we keep multiple (currently 7) copies of the debug and optimized builds as well as branch and test nightlies and the build size keeps increasing, afs space is being consumed at a prodigious rate.
 - It takes more and more time.
 - The nightlies have been taking more than 24 hours at times.
 - This will be alleviated by incremental builds but could reappear.
 - Both the increase of the build size and the build time may be related to automatic code generation for POOL/SEAL.
- More platforms need to be supported.
 - SLC3 is nearly ready (including Kit Validation).
 - Lots of work has been done on MacOSX.
 - Actively soliciting the participation of the user community.

Building Issues II

- Local builds and retrying have improved build reliability greatly.
 - AFS continues to be unreliable and the external packages remain on AFS.
- LCGCMT “glue” package has been added to ensure complete consistency between ATLAS, Gaudi and LCG components.
 - Primary LCG client of LCGCMT is POOL.
 - Will soon be adding ROOT, GEANT4, CLHEP, and XercesC to LCGCMT.
 - Evaluating other external packages for inclusion in LCGCMT but it does not necessarily make sense to include all external packages in LCGCMT.

Building Issues III

- We urgently need more fast, big memory machines.
 - David Asbury has promised a machine for SLC3.
 - One more SLC3 machine is needed for branch releases.
 - Dual SLC3/RH 7.3 operation will continue for a while (at least to 2005). The SPMB should decide how long.
 - Large memory, fast machines for interactive debugging and valgrind are sorely needed.
 - Would consider moving builds from CERN to find a better hardware environment for the build.
- A 500 MB file size limit for the builds has been added to protect against spurious multi-GB files.
 - The largest file currently generated by the build is 265 MB!

Software Distribution

- Distribution Kit is being built for every release.
 - Kit not built for nightlies.
- There has a been considerable work to make the software distribution system more reliable.
 - Far fewer patches are now needed to successfully build the distribution kit,
- Work still remains:
 - Multiple releases in one location
 - Dev kit including source - strong request from users!
 - Waiting for new pacman version (3.0) for some features
- Working on running the kit validation in the nightlies so that it will be easier and faster to generate a working kit after building a release.
 - Kit Validation will be included in release 9.0.0.

Software Testing & Validation

- There are multiple efforts to test / validate SW
 - There is the AtlasTest tests run by QMTest which is running in the nightly using tests contributed by developers.
 - Usually more than half of these tests fail on any given nightly.
 - There is the Run Time Tester (RTT) which does numerous, lengthy tests each night.
 - Does the highest statistic tests of the automatic packages.
 - There is the Kit Validation package used to validate that the installation of the distribution kit on a grid sites is correct.
 - There are also traditional human based validation work done by the Physics Validation, Reconstruction, and other groups.

Testing Discussion

- There is actually quite a lot of info already available.
- The various testing efforts need to be better coordinated to avoid duplication.
 - There are a number of tools for comparing test results that can possibly be consolidated.
- Users would like Athena return a status other than zero to expedite detecting problems.

Kit Validation

- The Kit Validation approach has desirable features:
 - Self contained
 - Use standard physics generation and simulation
 - Simple, clear output
 - Can run one, some, or all of defined tests
- Kit Validation will be in distributed in release 9.0.0 and has been used validate the software installation for all three grids used for DC2.
- It was suggested to run the Kit Validation tests as additional nightly tests using QMTest within NICOS.
- Alessandro's slides provide a good manual on using Kit Validation:

<http://agenda.cern.ch/askArchive.php?base=agenda&categ=a036308&id=a036308s13t8/transparenties>

NICOS

- The tests NICOS makes in the nightly are very useful.
 - NICOS makes it straight forward for developers to add their own tests to the nightlies.
- The NICOS web display of the test results is powerful.
 - It was requested that the NICOS web page link to RTT.
 - RTT already links to the test results in NICOS.
- Many of these tests fail every night - we need better participation from the developers & maintainers of the tests.

Run Time Tester

- RTT is very thorough it runs a large number of multi-event jobs every night on an UCL cluster
 - It runs a minimum of 150 Atlfast jobs (~1000 events/job) each night.
 - If all defined tests and metrics are run, over 1000 jobs will run over the night.
- RTT has a nice web interface showing its results.
 - The regression mechanism for comparing test results to a standard is pretty powerful.
- It was suggested that one of the RTT tests run every night look for memory leaks.
 - Not exactly physics validation but still useful.
 - Peter Sherwood is actively implementing this.

Testing Conclusions

- It is really important to test EVERY night so that problems are spotted immediately.
 - Both NICOS and RTT do this.
- Tests requiring people to look at the output fail because of human nature even though people are interested in the results. We need to find a way to get the developers to pay attention.
- People believe that testing is really important.
 - The message that many people want good, automated testing is coming through loud and clear.
 - People expect that the tests will be used to make the go / no go decision for production releases but not development releases.
- This work must be done in close collaboration with the validation group.
 - Feedback from the physics validation community is crucial.
- We will have a phone meeting in two weeks to continue work.

WBS and Staffing Levels

- As usual I am ignoring a request by Tom LeCompte for input to his WBS about the SIT effort levels.
 - I will try to talk to the SIT group members individually to revise the numbers that Tom has (substantial revision is certainly required). I apologize to Tom for how long it has taken for me to get started on this.
 - However I do not need a WBS to tell you that we are short of people and that the people that we have are less than ideally distributed over the urgent tasks.

Personnel

- RD is stepping down as the Testing Coordinator
 - We need to identify a new testing coordinator.
 - Will try to resolve this at the Testing phone meeting the week after next.
- Jakob Nielsen who wrote and maintained the very useful checkreq tool that checks CMT requirement files for consistency with the source code is leaving and we need to identify someone to take over responsibility for checkreq.
 - Best wishes to Jakob on his new job...

Conclusions

- Coordination and lack of people remain the two biggest issues in the SIT work.
- We are considering having a one to two day SIT workshop because of the large number of open issues.