Evolution of the ATLAS Nightly Build System

A Undrus¹, on behalf of the ATLAS Collaboration
Brookhaven National Laboratory, Upton, NY 11973, USA
E-mail: undrus@bnl.gov

Abstract. The ATLAS Nightly Build System is a major component in the ATLAS collaborative software organization, validation, and code approval scheme. For over 10 years of development it has evolved into a factory for automatic release production and grid distribution. The 50 multi-platform branches of ATLAS releases provide vast opportunities for testing new packages, verification of patches to existing software, and migration to new platforms and compilers for ATLAS code that currently contains 2200 packages with 4 million C++ and 1.4 million python scripting lines written by about 1000 developers. Recent development was focused on the integration of ATLAS Nightly Build and Installation systems. The nightly releases are distributed and validated and some are transformed into stable releases used for data processing worldwide. The ATLAS Nightly System is managed by the NICOS control tool on a computing farm with 50 powerful multiprocessor nodes. NICOS provides the fully automated framework for the release builds, testing, and creation of distribution kits. The ATN testing framework of the Nightly System runs unit and integration tests in parallel suites, fully utilizing the resources of multi-core machines, and provides the first results even before compilations complete. The NICOS error detection system is based on several techniques and classifies the compilation and test errors according to their severity. It is periodically tuned to place greater emphasis on certain software defects by highlighting the problems on NICOS web pages and sending automatic e-mail notifications to responsible developers. These and other recent developments will be presented and future plans will be described.

1. Introduction

ATLAS (A Toroidal LHC Apparatus) [1] is one of the largest collaborative efforts ever attempted in the physical sciences. Long-term development and rapid changes of ATLAS software poses a challenge for its management. In 2000 the first ATLAS software releases contained about 100 packages written by several tens of authors. During over 10 years of development the software requirements, models, and tools underwent several modification cycles as dictated by various phases of the life of the ATLAS experiment (assembly, commissioning, operation, upgrades) and also by hardware improvements such as better data storage and modern multi-core CPUs. The ATLAS software community and code size grew more than 10-fold.

¹ To whom any correspondence should be addressed.
The ATLAS nightly build framework was created in 2001 to facilitate coordination between knowledgeable and experienced software developers working around the world and around the clock. From the very beginning the central part of this framework was the Nightly Control System (NICOS) [2] that manages the multi-platform nightly releases reflecting the current state of the source code. Its design is based on 4 principles: repeatability (releases can be rebuilt many times, with or without modifications), accumulation (results are stored and reflected on NICOS web pages), feedback (automatic e-mail notifications about problems), and user-friendly setup (configuration parameters are stored in the database with a web interface). The ATLAS Testing Nightly (ATN) test tool [3] is embedded within the Nightly System and performs tests of different granularity and purposes for nightly releases. Tests parameters and dependencies are specified in configuration files, which are in the universal XML format developed for all ATLAS testing frameworks. ATN automatically finds test configuration files in a nightly release tree and launches tests in parallel suites as soon as possible, concurrently with compilations. Test results are returned to NICOS for processing and publication.

The paper describes growth phases of the ATLAS Nightly Build System and how it was adapted, corrected, and improved to meet demands of ATLAS software community. It explains the role of the Nightly System in achieving accelerated development cycles leading up to the stable software releases with desired quality features. As well future development plans are described.

2. ATLAS Nightly Build System growth

The development of the ATLAS Nightly Build System can be divided in 3 stages: establishment of the System, accelerating growth, and refinement of the System. Table 1 describes the stages of the ATLAS Nightly System growth.

| Stage         | Dates        | Number of nightly branches | Build Farm Hardware | Number of packages in releases | Number of ATN tests per release | Time to build a release, hours |
|---------------|--------------|----------------------------|---------------------|-------------------------------|---------------------------------|--------------------------------|
| Establishment | 2001-2005    | 1 - 2                      | ~5 nodes, 2x2.4 GHz, 1GB RAM | up to 800                    | 50 - 80                         | 22                             |
| Accelerating Growth | 2006-2009 | 10 - 30                    | 40 nodes, 4x2.8 and 8x2.3 GHz, 8–16GB RAM | up to 2000                  | 250 – 350                       | 10                             |
| Refinement    | 2010-2012    | 45 - 55                    | 50 nodes, 8x2.3 and 8x2.5 GHz, 16–24GB RAM | 2200                        | 400 - 450                       | 9                              |

During the first stage the major components of ATLAS software organization and build system were established. The Nightly System provided two multi-platform branches of nightly releases: “cutting edge” development and more stable bug-fix versions. In 2006 – 2009 ATLAS software experienced an accelerating growth in preparations for the detector commissioning and data taking. The ATLAS Nightly System was significantly reconfigured to support multiple nightly branches. Despite doubling ATLAS code in size, the nightly release build time was shortened by utilizing new build techniques and tools. In 2010 – 2012 the ATLAS Nightly System was further improved to offer up to 55 nightly branches of different scopes and purposes including “personal” branches assigned to individuals for testing new chunks of code. The number of nightly platforms, branches, and ATN tests as a function of time is shown in figures 1 and 2. The sophisticated code validation
organization [4] allowed achieving high quality of the nightly builds with 90% tests successful as illustrated in figure 3.

![Figure 1. Number of nightly platforms and branches.](image1)

![Figure 2. Number of ATN tests in a typical ATLAS nightly release.](image2)

![Figure 3. Success rate of ATN tests in a typical ATLAS nightly release, %](image3)

3. Evolution of ATLAS nightly framework design

ATLAS nightly builds are managed by the NICOS tool [2] for over 10 years. The initial role of the Nightly System was to expose bugs in new developer’s code by building nightly releases based on the recent versions of software packages. The design and implementation of the nightly framework for this role was completed in 2004 and it is shown in figure 4. The main principles of the nightly system organization remain unchanged since then. Packages are primary management units in the ATLAS software structure [4]. The ATLAS software release [5] comprises the large number of packages with specific version tags stored in the Tag Collector [6] web interfaced database application. Developers are able to interactively select the version tags from the ATLAS code repository for the nightly releases. The release building responsibility rests with CMT [7], the configuration management tool
describing the packages properties and dependencies. CMT is in charge for building release components, such as applications and libraries.

The NICOS design strategy is based on a goal to provide a flexible and easy to use Nightly System. It is designed in a modular way so that each module is responsible for a certain step and separately described in the nightly configuration. The modular structure allows plugging external collaborative tools, such as CMT and the Tag Collector, and replacing them as needed. NICOS comes with the embedded ATN [3] test module allowing to run tests of different granularity and purpose. During the first stage the ATN tests were represented by test scripts selected from the specifically named directories of packages. Nightly releases are copied for a worldwide access on the CERN AFS distributed file system and retained for several days (usually a week). As a fast feedback to developers is one of the most important functionalities of nightly systems, NICOS automatically posts the information about the progress of nightly builds and tests, identifies problems, and creates the summary web pages reflecting the system status. Automatic e-mail notifications about problems are sent out to responsible developers.

In 2006 the fast growth of ATLAS software posed challenges for the existing Nightly System. The release build time was approaching to 24 hours. The software development was split into several streams and required multiple nightly branches. The ATLAS Nightly System was fully revamped to meet growing demands:

- Storage of configurations for different branches of nightly releases was handed over to the ATLAS metadata web interfaced database (AMI) [8] allowing easy system administration;
- ATLAS Nightly Build farm at CERN was upgraded with power multi-core nodes;
- ATLAS software release [5] was split into about 10 projects consisting of packages having similar dependencies and built as a unit;

![NICOS design in 2004.](image_url)
Build acceleration tools significantly reduced the build duration;
- file level parallelism (with distcc/gmake -j<n>);
- package level parallelism (with CMT tbroadcast tool);
- ATN tests for low level software projects were performed in parallel with the compilations of high-end packages;
- ATLAS Nightly System was synchronized with the external RTT [9] testing framework that is designed for lengthy and intensive validation tests;
- Test configurations in unified XML-based format became the standard in ATN and RTT test frameworks;
- Tag Collector was amended with administrative tools for project and release coordinators that can approve or reject new tags, merge tags from different development branches, freeze releases;
- Creation of the nightly release distribution kits [10] became a part of nightly jobs so that nightly releases could be installed at ATLAS Data Centers and personal computers.

As a result of these upgrades the ATLAS Nightly System successfully supported more than 30 multi-platform nightly branches by the end of 2009. In addition to traditional development and bug-fix branches the nightly releases for development of specific software domains were added. The patch nightly branches were dedicated for building special patch projects that sit at the top of the project hierarchy and contain override versions of packages from downstream projects.

A start of ATLAS data taking required further refinement of the ATLAS Nightly System. The current ATLAS Nightly System organization is shown in figure 5.

Figure 5. NICOS design in 2012.
ATLAS made a switch from CVS to SVN code repository which offers such advantages as better speed and easier “hook” scripts and the Nightly System was amended with the SVN interface. The creation of the ATLAS Nightly CernVM-FS [11,12] repository improved an access to nightly releases. The CernVM-FS is a fuse-based http, read-only file system which guarantees file de-duplication and on-demand file transfer with caching, scalability and performance. The ATLAS Nightly System was integrated with ATLAS GRID Installation System [13] that allowed, in particular, nightly kits installation and validation outside the CERN Nightly Farm. Further increase of parallelism in builds and testing allowed full use of the multi-core machines resources and resulted in decrease of the build and testing times.

The ATLAS Nightly System works as a factory for the preparation of stable release candidates. The successful nightly releases are relocated and transformed into the stable “numbered” releases by the team of ATLAS offline shifters. They are deployed after intensive validation in ATLAS testing frameworks [4,5]. The ATLAS Nightly System led up to 379 stable software releases in 2011.

4. ATLAS nightly branches varieties
There are many advances in the large ATLAS software system where improvements are developed in parallel efforts for software design optimization, implementation evolution, modifications caused by detector upgrades, and overall architecture revamp. The complex system of nightly branches has been developed to facilitate parallel efforts and accelerate development cycles:

- Major integration branches include tags that were already thoroughly verified in other branches, such as validation nightly releases. Several such branches are available for specific development tasks such as software preparation for data processing campaigns;
- Validation branches are based on the same tags as corresponding major branches. The release coordinators are able to test new versions in a validation branch before a final approval decision. This ensures a high stability of the integration nightly branches;
- Migration and “personal” branches are used for testing of new external packages or development of specific software domains when such development could be disruptive for the main nightly branches. ATLAS Nightly System has 12 migrations branches that can be assigned to groups or individual developers (in which case they become “personal”) for a limited time;
- ATLAS software extensively uses the LCG software created for all LHC collaborations. ATLAS LCG nightly releases [14] connect ATLAS and LCG nightly build system and provide fast feedback both to LCG and ATLAS software developers;
- Experimental nightly branches probe new operating systems and compilers;
- Nightly patch releases accumulate amendments to existing stable releases. All version tags of patch releases are eventually incorporated into the ongoing nightly and future stable releases;
- “Physics Analysis” nightly releases are dedicated to the groups of physicists for specific analysis software collections. They operate on top of patch releases.

5. Reliability of the ATLAS Nightly Build System
The ATLAS Nightly Build System maintained high reliability through all three stages of development. The number of Nightly System technical accidents that resulted in any nightly releases loss per calendar quarter is shown in figure 6. The frequent accidents causes were

- Code Repository outages;
- Tag Collector outages;
- Hardware failures;
- CERN AFS failures.
The Nightly System has the automatic failure detection and retrial mechanisms. This feature often compensates for temporary technical problems and saves nightly jobs that would otherwise fail. During critical phases of software development the release coordinators and offline release shifters check periodically the nightly system status and manually restart jobs in case of problems. Most of the time the accident rate was 10–15 per calendar quarter even during the periods of fast system growth and only less than 2% of nightly releases were actually lost (as in most accidents only few nightly branches are affected).

There were 2 short periods of instability in the ATLAS Nightly System since 2004:
- Spike in 2Q 2007 was caused by failures of aging build machines. Their replacement with modern powerful nodes helped the system recovery;
- In 4Q 2009 the CERN AFS distributed file system became overloaded as a result of increased user activity after ATLAS started the data taking. Although it was not possible to avoid AFS usage in the Nightly System, many tools such as CMT configuration tools were installed on the local disks of build machines. The nightly releases also were built and tested locally and then copied to AFS for a worldwide use. As the result the number of accidents decreased and nightly processes were accelerated.

6. Future development
The ATLAS Nightly System development plans are focused on the addition of the nightly dashboard web-based application providing the current system status and various administrative opportunities such as builds restart and rescheduling. It will make possible to introduce nightly builds “on demand”. Unlike currently running continuous builds, on-demand nightly jobs start when the release coordinator deems it necessary. The flexible on-demand organization allows reducing the build farms load because builds are performed when they are really needed. On the other hand release coordinators get an advantage of scheduling nightly jobs right after new software submissions and getting the feedback as soon as possible.
7. Conclusion
Over the last decade the ATLAS Nightly System served as a major tool in the ATLAS collaborative software organization and management schemes. It started as a simple tool to uncover defects in new developers’ code. The system managed to sustain increasing number of developers and software size after numerous amendments and improvements. The ATLAS Nightly System evolved into an important collaborative tool supporting tens of nightly releases branches and providing a stable environment for the parallel development of software functionalities.

8. Acknowledgements
The author wishes to thank members of ATLAS Software Infrastructure and Tag Collector teams for many valuable advices and useful discussions. This work was supported by the US Department of Energy and National Science Foundation.

9. References
[1] The ATLAS Collaboration 2008 JINST 3 S08003
[2] Undrus A 2003 Proc. Int. Conf. on Computing in High Energy and Nuclear Physics CHEP’03 (La Jolla, USA) eConf C0303241 pp TUIJT006 (e-Print hep-ex/0305087)
[3] Undrus A 2004 Proc. Int. Conf. on Computing in High Energy and Nuclear Physics CHEP’04 (Interlaken, Switzerland) CERN 2005-002 pp 521-3
[4] Obreshkov E 2011 J. Phys.: Conf. Ser. 331 042017
[5] Luehring F, Obreshkov E, Quarrie D, Rybkine G and Undrus A 2010 J. Phys.: Conf. Ser. 219 042045
[6] Albrand S, Collet J, Fulachier J and Lambert F 2004 The tag collector – a tool for ATLAS code release management Proc. Int. Conf. on Computing in High Energy and Nuclear Physics CHEP’04 (Interlaken, Switzerland), Obreshkov E 2008 et al. Nucl.Instrum.Meth.A 584 244-51, http://atlastagcollector.in2p3.fr
[7] Arnault C 2000 Proc. Int. Conf. on Computing in High Energy and Nuclear Physics CHEP’00 (Padova, Italy) (Computer Physics Communications Vol. 140) pp 692-5, Arnault C 2001 Proc. Int. Conf. on Computing in High Energy and Nuclear Physics CHEP’01 (Beijing) pp 8-006, http://www.cmtsite.org
[8] Albrand S, Fulachier J and Lambert F 2010 J. Phys.: Conf. Ser. 219 042030
[9] Simmons B, Sherwood P, Ciba K and Richards A 2010 J. Phys.: Conf. Ser. 219 042023
[10] Arnault C, De Salvo A, George S and Rybkine G 2004 The deployment mechanisms for the ATLAS software Proc. Int. Conf. on Computing in High Energy and Nuclear Physics CHEP’04 (Interlaken, Switzerland); Rybkine G 2012 ATLAS software packaging these Proceedings
[11] Blomer J, Aguado-Sánchez C, Buncic P and Harutyunyan A 2011 J. Phys.: Conf. Ser. 331 042003
[12] De Salvo A, De Silva A, Benjamin D, Blomer J, Buncic P, Harutyunyan A, Undrus A and Yao Y 2012 Software installation and condition data distribution via CernVM FileSystem in ATLAS these Proceedings
[13] De Salvo A, Barchiesi A, Gnanno K, Gwilliam C, Kennedy J, Krobath G, Olszewski A and Rybkine G 2008 J. Phys.: Conf. Ser. 119 052013
[14] Kruzelecki K, Roiser S and Degaudenzi H 2010 J. Phys.: Conf. Ser. 219 042042