SVN evaluation and proposal for Atlas, Atlas software week

Christian Arnault, Vincent Garonne
{arnault,garonne}@lal.in2p3.fr

Laboratoire de l'accélérateur Linéaire

April 7, 2006

Outline

SVN *in a nutshell*

What is Subversion (or SVN) ?

- ▶ Source code management system
- ▶ Open Source project (http://subversion.tigris.org/)
- ▶ *SVN* has new features that cover many limitations of *CVS*
- ▶ Subversion commands are very similar to *CVS*, most of the time it's a matter of replacing "cvs" with "svn" [1]

Questions for Atlas

As we have observed limitations/weird behaviors of *CVS*:

- ▶ Is it appropriate to do a swap ?
- ▶ Is the atlas developer is enough perturbed :) by the multi-projects structure without adding that ?

CVS *versus* SVN

- *SVN* really works faster than *CVS*:
  - Transmits less information through the network
  - Supports more operations for offline mode, e.g client sends deltas to server when committing

Source code management system	CVS	SVN
Atomic Commits	No	Yes
Efficient (constant-time) tagging	No	Yes
File renames handles cleanly (renaming, moving)	No	Yes
Advanced Merging Features	No	Yes
Manipulation of all the file types	No	Yes
Easy Development on Branches	No	Yes
Easy Server Administration	No	Yes
Remote Repository Replication	No	Yes
Scalable Performance and Admin	No	Yes
Advanced repository permissions Features	No	Yes
Web Browser Interfaces	No	Yes
HTTP proxy to connect the server	No	Yes

** For more infos, look [2]

Scenario for Atlas

Roadmap

1. Convert the Atlas CVS repository into a Subversion one
   - A tool exists, "cvs2svn" (http://cvs2svn.tigris.org)
2. Provide tools to provide synchronization operations between *CVS* and *SVN* repositories
   - For a certain period, we will have both system running at the same time
   - remark: iterative transition is less painful for the developers :)
3. Incorporate *SVN* in the complex Atlas software management chain, it supposes impact to :
   - CMT
   - Tag Collector
   - Nicos
4. If successful, we could do the permanent failover; if not ... :(

CVS repository conversion

Conversion script

For better understandings, we have developed our own cvs2svn
python script (/afs/cern.ch/atlas/scripts/cvs2svn.py):

- Keeps the history of all tags up to HEAD
- Skips the intermediate commits between 2 tags
- Can be applied iteratively

Definition of a repository organization

- ▶ SVN gives full freedom in terms of internal structure
- ▶ Tags, branches are views to the source history
  - in CVS, they are meta-data applied to individual files) and
    appear like directory structures
- ▶ Views can be freely modified and have revision number

Converting CVS Repository to Subversion

▶ We have selected the following organization for each project/package/tag:
  `<project>/.../<package>/trunk/...`
  `<project>/.../<package>/tags/<tag>/...`

▶ The complete Atlas cvs repository has been converted to a svn one. The svn url is :
  file:///afs/cern.ch/atlas/software/svnrepository/repository

▶ It takes a looong time to do that the first time
  • Try different *SVN* backends: Berkley DB, *SVN* file system
  • It takes me one week to complete the procedure, e.g. 8 hours for the AtlasCore project
  • but update will only concern recent tags

Conversion validation and management tools

Conversion validation tools

► We have implemented several tests to check the quality of the conversion:
  • Comparing the sources at each tag with diff tests
  • Comparing the list of tags for a package
  • Check global structure of the *SVN* repository
  • /afs/cern.ch/atlas/scripts/testcvs2svn.py

Management tools

► We have also implemented a script which re-organizes the SVN repository:
  ```
  <project>/.../<package>/trunk/...
  <project>/.../<package>/<tag>/...
  ```

Performance and scalability

Test stress scenario

We define the following **action** for an user:

▶ User selects randomly a package

▶ He extracts and modifies it, then registers it until the operation is successful

▶ Then he sleeps n seconds with n sort randomly $\in [1, T_{max}]$

Experiment parameters

▶ We have done experiments with different:
  • Total number of users (10, 100, 1000)
  • Total number of packages (10, 1000, 10.000)

▶ Concurrent user actions were launched by a multi-threaded program (/afs/cern.ch/user/v/vgaronne/svnTestStress)

Performance and scalability - Results

▶ Let $\mathcal{R}atio$ be the ratio $\frac{Total\ number\ of\ users}{Total\ number\ of\ packages}$

We obtained :

$\mathcal{R}atio$	T_{max}	Average action time (s)
0.01	3600.0	1.34
0.01	100.0	1.06
0.01	10.0	1.38
0.1	3600.0	5.77
0.1	100.0	4.26
0.1	10.0	6.96
1.0	3600.0	7.42
1.0	100.0	7.06
1.0	10.0	8.88

**In blue, the values relevant for Atlas

Synchronization operations

Mainly at the user level:

▶ We proposed a script "asvn" that mimics the CVS syntax for SVN, including the synchronization operations on tags:
  - Non exhaustive commands are: checkin, checkout , tag, ...
  - Provides also cmt specific commands working with SVN, e.g. recursive checkout, svntags, svnsubprojects, ...
  - see /afs/cern.ch/atlas/scripts/asvn

▶ We assume that when users switch completely to svn for a package, they will not work anymore with *CVS* for this package, otherwise we will have integrity problems

▶ The synchronization operation should be triggered at the user/developer level during tag operations
  ```
  > asvn --sync tag Package-00-00-01 Package
  ```

Impact to the Atlas software management chain and tools

Impact to the work model

- ▶ No real change is expected

Impact to the tools

- ▶ CMT: The `"asvn"` script previously presented covers these aspects
- ▶ Tag Collector: Needs an interface to SVN, but should be much simpler than the interface to CVS. If not possible immediately, we could go on temporally with the synchronized Atlas CVS repository
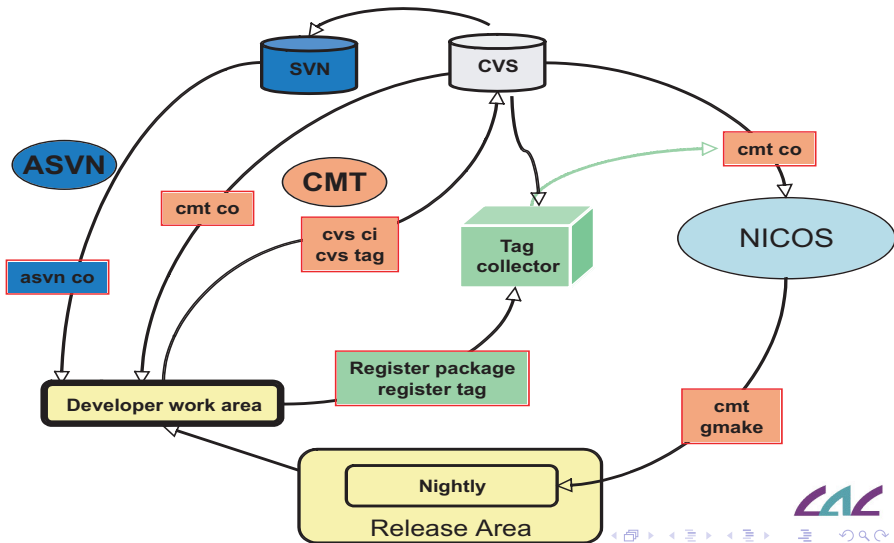- ▶ NICOS: Normaly only concerned with checkout operations ?
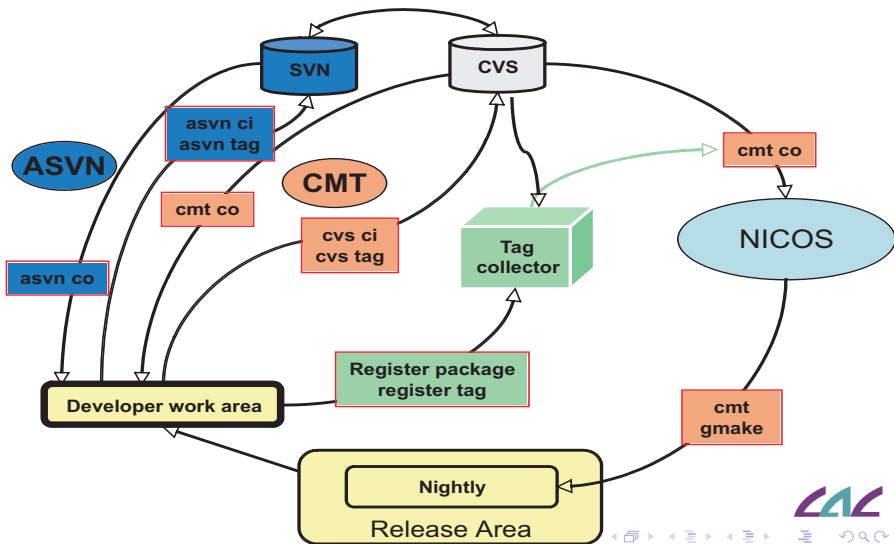
A brief summary by pictures
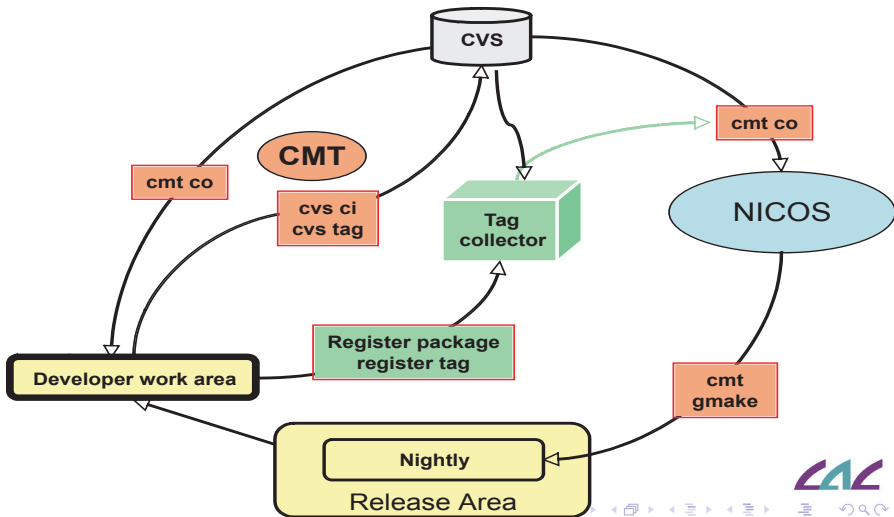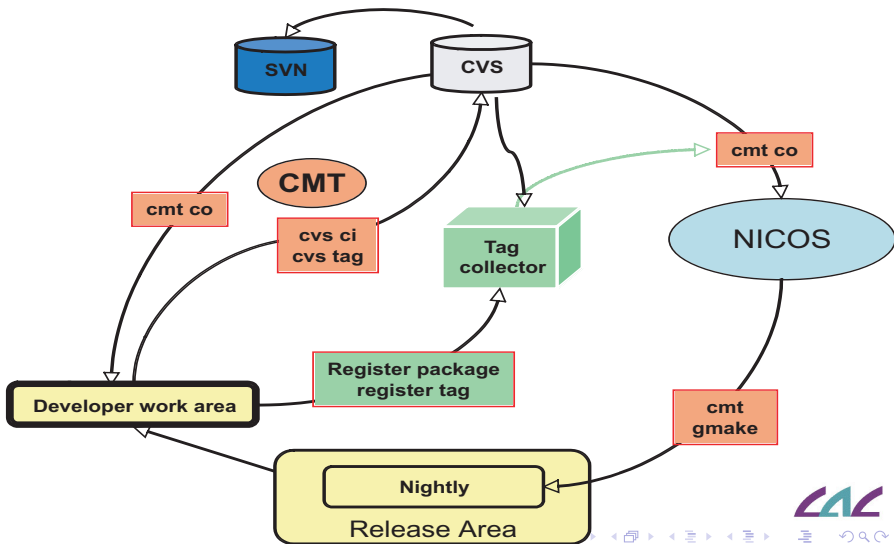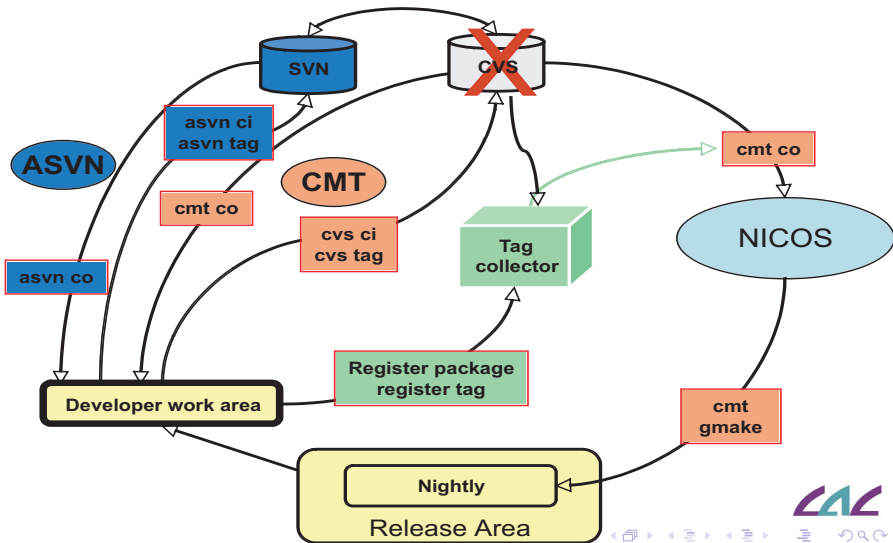
A brief summary by pictures
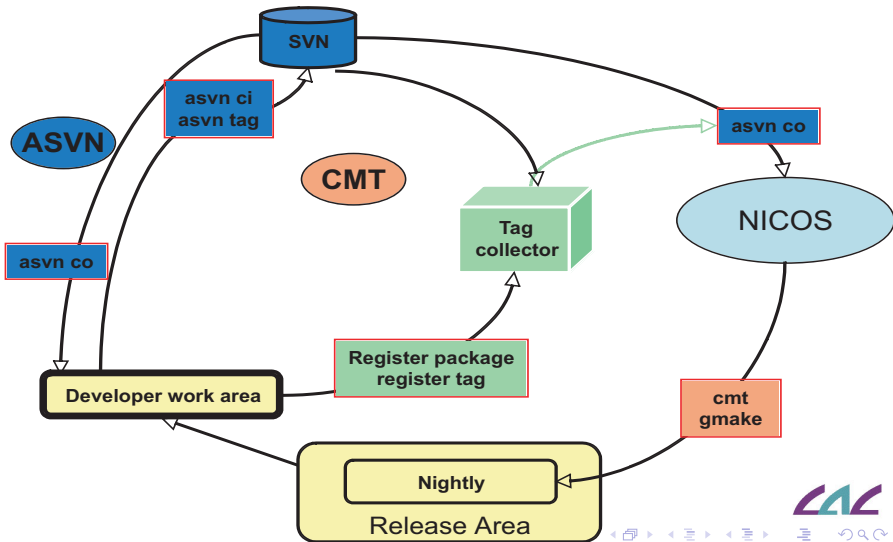
A brief summary by pictures

A brief summary by pictures

A brief summary by pictures

A brief summary by pictures

Conclusion

- ▶ All the necessary tools are in place to begin using SVN in Atlas
- ▶ Testing by real users, will help to tune and improve these tools
- ▶ Doing more tests to validate the atlas svn repository
- ▶ Implementing the interface to SVN in Tag Collector, depends on TC team availability

Future

- ▶ SVN brings lots of new interesting features, that we could use incrementally :
  - • Metadata features
  - • Real Multi-repository
  - • Directory management
  - • Transparent branch management

References I

📄 cvs and svn crossover commands.
http://uimon.cern.ch/twiki/bin/view/Atlas/
StudiesForUsingSVNInAtlas#Command_Examples_SVN_vs_
CVS.

📄 Subversion and CVS comparison.
http://wiki.gnuarch.org/SubVersionAndCvsComparison.

📄 Subversion.
http://subversion.tigris.org/.

📄 cvs2svn.
http://cvs2svn.tigris.org.