

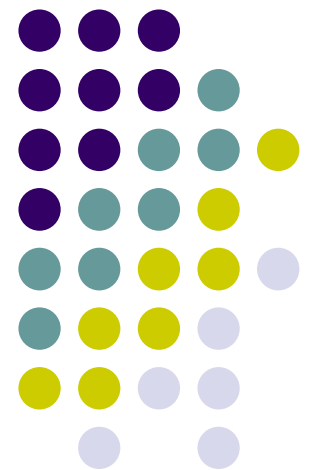
# AliRoot Build and Test System

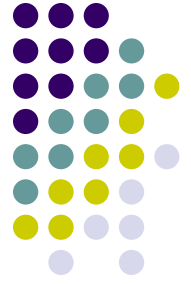
---

ALICE-FAIR Computing Meeting

28/04/2008

Presented by P.Hristov





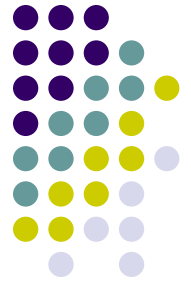
# History

- AliEn build system developed by Catalin Cirstoiu and Predrag Buncic in 2005
- First port of AliRoot: Vagner Morais in 2006
- Web interface and monitoring: Costin Grigoras in 2006
- Test examples: Peter Hristov in 2006/2007
- Final build system: Catalin Cirstoiu in 2007
- Support and future development: Alina Grigoras and Marco Meoni



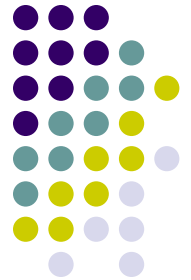
# Goals

- Automate the process of building AliRoot
- Have a current, up to date, installable binary image of AliRoot for different architectures
- Better understand the dependencies between the packages
- Provide software that works to the end-users
- Reuse the development done for the AliEn build system (AliEnBITS)



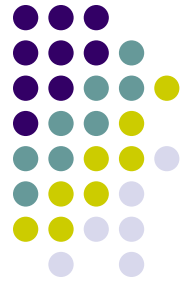
# Building System

- Based on the Konstruct framework used to build KDE and AliEnBits (originally GAR system)
- Each package has a directory in a hierarchy
- Each Makefile define a set of variables:
  - Name, version, author, web page
  - License information
  - Build and runtime dependencies
  - Sites to download the package source
  - Patches
  - Checksums (integrity of the files)
- The Makefiles are executed recursively



# Building Steps [1]

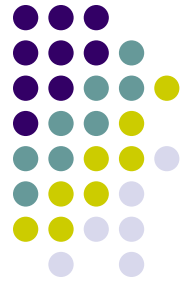
- Get the available Releases
  - Select the one to be (re)built
- Fetch the latest changes for this release
  - Do a svn update
- Determine the build order
  - Analyze the dependencies between packages (AliEn -> Root -> Geant3 -> AliRoot)
- Select the packages that have to be rebuilt
  - Based on modifications received on svn update
  - Based on last build status
  - Based on package dependencies
  - Always rebuild and test AliRoot packages



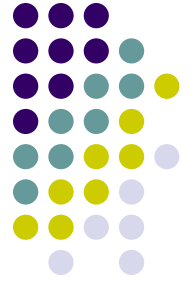
# Building Steps [2]

- For each package
  - Clean its environment
  - Fetch source archive from master sites
  - Generate SLOC statistics
  - Build, install and run tests
  - Create the binary image for binary installations
- Generate web pages
  - Build / SLOC / Graph dependencies
- Generate the pages for AliRoot tests
- Cleanup environment
- Send notification email when status changes

# Web Pages of the Build System



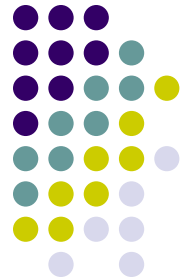
- The system generates web pages automatically (Build / SLOC / Graph dependencies)
- Package dependencies:
  - Better and faster understanding of the project
  - To see easily how many packages a certain package brings in
- Provide statistics on the size of the project:
  - For each package
  - For package's dependencies
  - For the whole group
  - Global view



# The AliRoot BITS

- AliRoot is being built nightly on all major platforms:
  - i686, x86\_64, ia64, IntelMac
  - The i686 build includes SHUTTLE
  - Building is performed only if needed
    - AliRoot or one or more dependencies were modified
      - version or tag or the tag was moved
    - Several AliRoot cvs tags can be tracked and rebuilt if changed
  - AliEn packages (AliRoot/Root/Geant3) are automatically generated
- If build is successful, tests are run
  - Tests provided in the AliRoot's repository
  - Currently: 20 pp, 1 PbPb, sim&rec
  - Monitoring information about the run is collected in ML



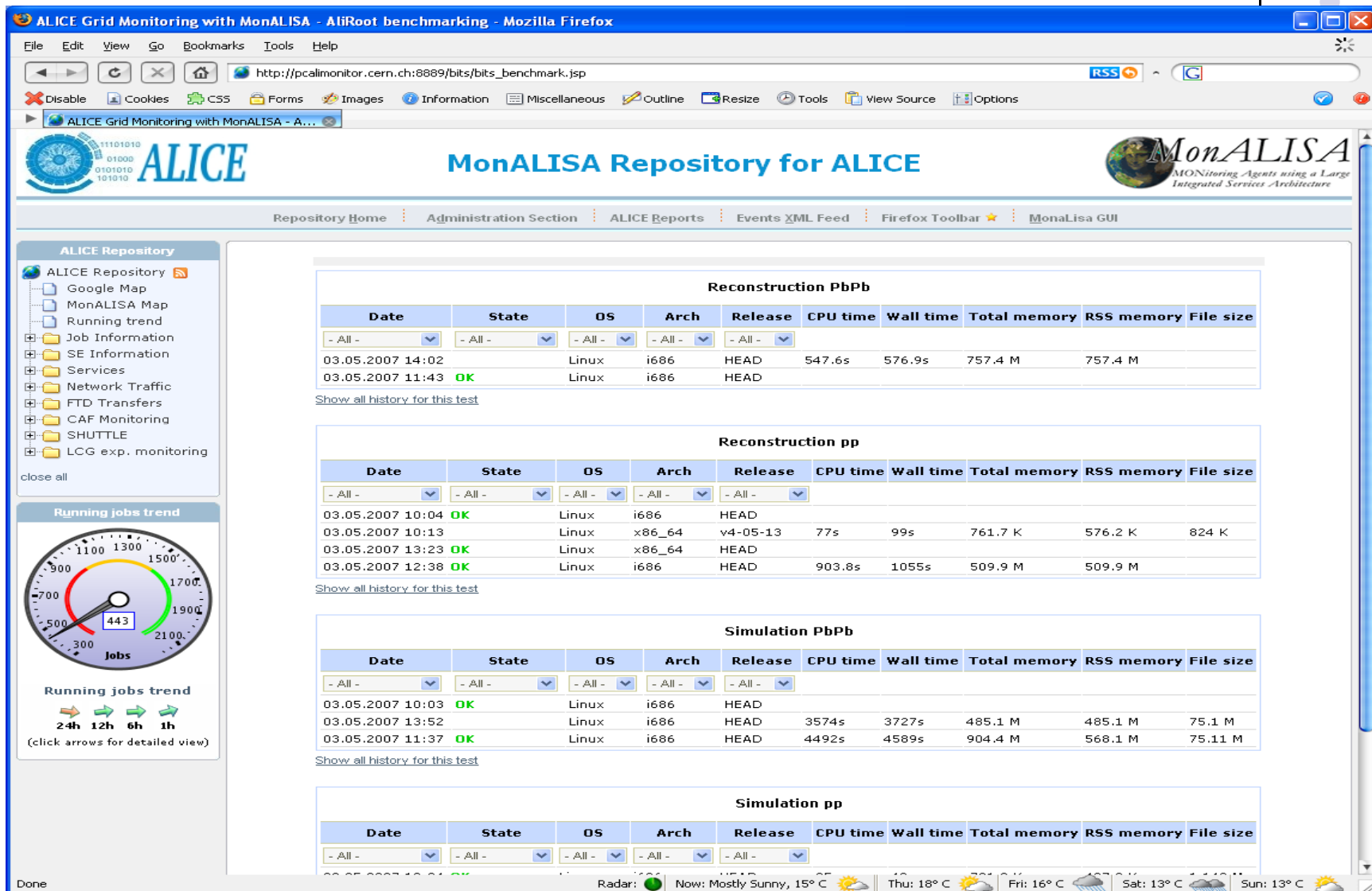


# AliRoot tests

- Both sim/rec tests, for both pp and PbPb send monitoring information to ML
  - Using the ROOT ML plugin developed by Andreas Peters
- Parameters
  - Cpu/Wall Time, Total/RSS Memory, raw.root and AliESD.root file sizes
    - Sent only if the test succeeds
  - Test success status
    - Sent only if AliRoot is successfully built

[http://pcalimonitor.cern.ch:8889/bits/bits\\_benchmark.jsp](http://pcalimonitor.cern.ch:8889/bits/bits_benchmark.jsp)

Thanks to Costin Grigoras



# Useful pointers



## AliRoot Build Integration and Testing System

**BITS: i686** - i686-pc-linux-gnu  
**Go to:** All . i686 . x86\_64 . Itanium . IntelMac

### AliRoot Releases for i686-pc-linux-gnu

Release	Date	Description	Status	Built on	Build time	Build #
HEAD	--/--/----	Current CVS head	✗	Fri Apr 25 13:12:41 2008	00:50:58	5
v4-11-Rev-04	02/04/2008	Production	✓	Tue Apr 22 20:51:29 2008	04:19:07	1
v4-11-Rev-03	14/03/2008	Production	✓	Fri Apr 25 18:55:31 2008	04:18:56	1
v4-11-Rev-02	11/03/2008	Production	✓	Thu Apr 24 18:56:45 2008	04:22:49	1
v4-10-Rev-02	22/02/2008	Production	✓	Thu Apr 24 13:19:40 2008	04:17:25	1
v4-10-Rev-01	06/02/2008	Production	✓	Thu Apr 24 01:06:52 2008	04:16:23	1
v4-09-Rev-01	19/12/2007	Production	✓	Wed Apr 23 19:03:17 2008	05:32:01	1

Generated on: Fri Apr 25 18:55:53 2008

## AliRoot BITS:

[http://pcalibuildintel.cern.ch:8889/global\\_index.html](http://pcalibuildintel.cern.ch:8889/global_index.html)

## AliRoot Benchmarks:

[http://pcalimonitor.cern.ch/bits/bits\\_benchmarks.html](http://pcalimonitor.cern.ch/bits/bits_benchmarks.html)

### Simulation pp

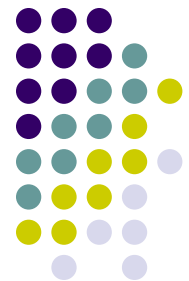
Start	End	Duration	State	OS	Arch	Release	CPU time	Wall time	Total memory	RSS memory	File size
- All -				- All -	- All -	- All -					
25.04.2008 16:23	25.04.2008 17:06	43m 31s		Linux	i686	v4-11-Rev-03	40m 39s	43m 26s	954.3 M	677.4 M	96.62 M
25.04.2008 05:47	25.04.2008 07:12	1:24	OK	Linux	ia64	v4-10-Rev-01	1:13	1:16	784.1 M	734.8 M	95.93 M
25.04.2008 02:56	25.04.2008 03:39	43m 31s	OK	Linux	x86_64	v4-10-Rev-01	32m 31s	36m 22s	770.4 M	770.4 M	96.71 M
24.04.2008 16:24	24.04.2008 17:07	43m 9s		Linux	i686	v4-11-Rev-02	40m 21s	43m 3s	955.4 M	677.4 M	96.84 M
24.04.2008 10:48	24.04.2008 11:31	43m 26s		Linux	i686	v4-10-Rev-02	40m 41s	43m 20s	972 M	689.9 M	96.16 M
24.04.2008 05:31	24.04.2008 07:09	1:38	OK	Linux	ia64	v4-09-Rev-01					
24.04.2008 02:40	24.04.2008 03:29	49m 1s	OK	Linux	x86_64	v4-09-Rev-01					
23.04.2008 22:35	23.04.2008 23:19	43m 30s		Linux	i686	v4-10-Rev-01	40m 47s	43m 25s	963.3 M	687.5 M	96.16 M
23.04.2008 05:49	23.04.2008 07:18	1:28	OK	Linux	ia64	v4-11-Rev-04	1:21	1:24	864.8 M	812.1 M	97.07 M
23.04.2008 02:58	23.04.2008 03:45	47m 30s	OK	Linux	x86_64	v4-11-Rev-04	35m 3s	39m 25s	793.2 M	793.1 M	96.6 M
04.05.2007 01:22	04.05.2007 02:26	1:04	OK	Linux	x86_64	v4-05-13	44m 27s	58m 59s	825.4 M	825.4 M	947.4 M

Show all history for this test

27/04/2008

P.Hristov

11



# Building a new release

- Used to involve manual editing of the versions, committing & tagging in BITS cvs
- Now requires only a few clicks
- New tags are discovered automatically

Add release

Date: 27/04/2008

Description: Production

Package	Version	Extra version
alien	v2-14	
alroot	v4-12-Rev-01	
geant3	1-9	-3
root	5-19-02a	

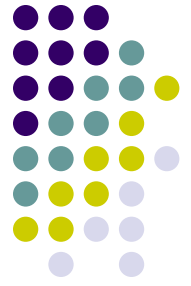
Save release Cancel

## Current AliRoot Releases

Release	Date	Description	alien	alroot	geant3	root	Active	Delete
HEAD	--/--/----	Current CVS head	HEAD	HEAD	1-9	5-19-02a	<input checked="" type="checkbox"/>	Delete
v4-11-Rev-04	02/04/2008	Production	v2-14	v4-11-Rev-04	1-9-2	5-18-00b	<input checked="" type="checkbox"/>	Delete
v4-11-Rev-03	14/03/2008	Production	v2-14	v4-11-Rev-03	1-9-2	5-18-00b	<input type="checkbox"/>	Delete
v4-11-Rev-02	11/03/2008	Production	v2-14	v4-11-Rev-02	1-9-2	5-18-00b	<input type="checkbox"/>	Delete
v4-10-Rev-02	22/02/2008	Production	v2-14	v4-10-Rev-02	1-9-1	5-18-00a	<input type="checkbox"/>	Delete
v4-10-Rev-01	06/02/2008	Production	v2-14	v4-10-Rev-01	1-9	5-18-00	<input type="checkbox"/>	Delete
v4-09-Rev-01	19/12/2007	Production	v2-14	v4-09-Rev-01	1-8	5-17-06	<input type="checkbox"/>	Delete

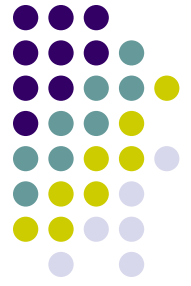
27/04/2008





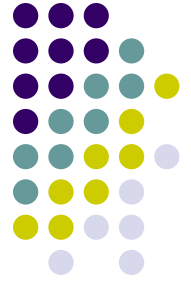
# Work in progress [1]

- Transition to SLC4
- Upgrade of the MacOSX servers
- New tests
  - Backward compatibility
    - Reconstruction of “reference” RAW samples
    - Use of “reference” OCDB data (calibration/alignment)
  - Code checker and reverse engineering tool
  - More “alarms” in case of failure



# Work in progress [2]

- “Incremental” builds from the SVN trunk
  - The trunk rarely passes all the tests (some times it even does not compile)
  - Instead of using one single revision (the latest one)
    - Prepare (manually) set of revisions (one per module) that is known to work correctly
    - Revert to the last working revision if a module has a problem
    - Replace the working revision of a module if the version from the trunk is OK



# Summary

- The tool for automatic builds has proven its usefulness in two different cases: AliEn - hundreds of packages, AliRoot - 4 packages + AliEn. It
  - Intelligently selects packages that have to be rebuilt
  - Offers different views for packages dependencies - very useful in understanding and managing large projects
  - Provides SLOC statistics
  - Saves and presents relevant logs and statistics on components testing
  - Has “one click” interface for the build management
- Building and continuous testing is important for delivering high quality software to end-users