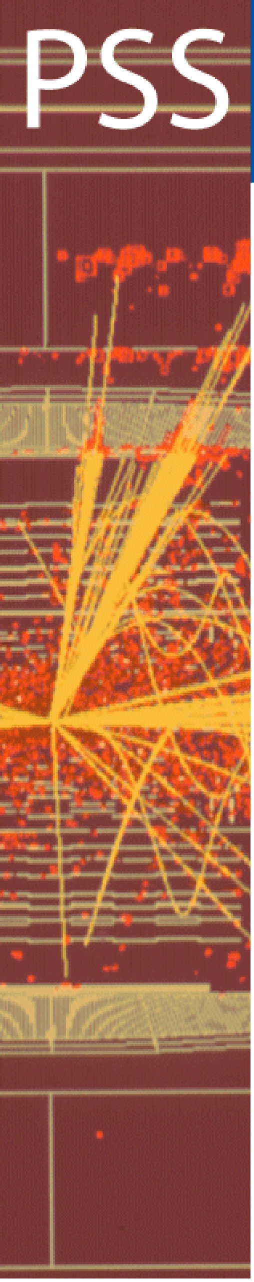
PSS Physics Services Support

CERN IT Department



COOL

Development and Deployment Status

<u>Marco Clemencic (CERN PH-LBC)</u>

On behalf of the COOL team

CHEP2007, 3rd September 2007

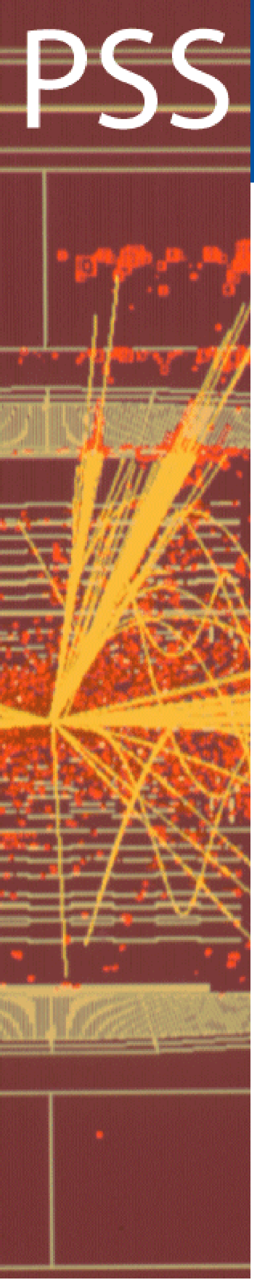
Presentation prepared by Andrea Valassi (CERN IT-PSS-DP)

PSS

Outline

CERN IT Department

- **Introduction**
- **Development overview**
  - Focus on progress since CHEP 2006
- **Deployment overview**
- **Conclusions**



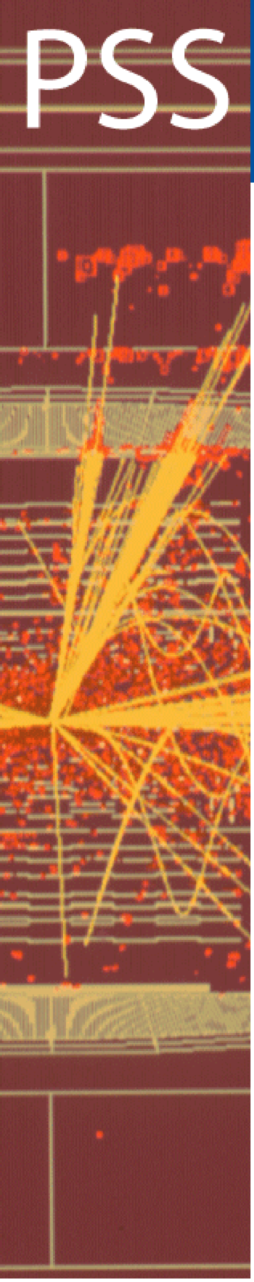
CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

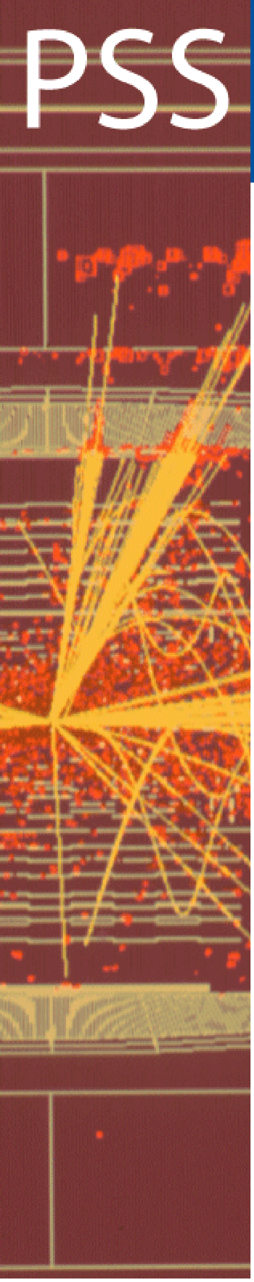
PSS

What is COOL

CERN IT Department

- **Software for LHC conditions data access**
  - Time variation (IOVs) and versioning (tags)
  - Offline (calibration, alignment) and online (DCS)
- **Common project of Atlas, LHCb, CERN IT**
  - Atlas and LHCb will use it for their conditions data
- **Collaboration with other LCG AA projects**
  - CORAL and SEAL for C++ implementation
  - ROOT/Reflex for python bindings (PyCool)
- **Support for several relational backends**
  - Oracle, MySQL, SQLite, Frontier (all via CORAL)





COOL collaborators

Core development team

- **Andrea Valassi (CERN IT/PSS)**
  - 80% FTE (core development, project coordination, release mgmt)
- **Marco Clemencic (CERN LHCb)**
  - 20% FTE (core development, release mgmt)
- **Sven A. Schmidt (Mainz ATLAS)**
  - 20% FTE (core development – previously 80% FTE in 2005-2006)
- On average, around 1.5 FTE in total since October 2004

With useful contributions from many other people!

- *Romain Basset, Gianni Pucciani* (CERN IT/PSS), *David Front* (Weizmann and CERN IT) – performance tests and optimization
- *Richard Hawkings* and other *ATLAS users, testers and DBAs*
- *The CORAL, SEAL/ROOT, SPI and 3D teams*

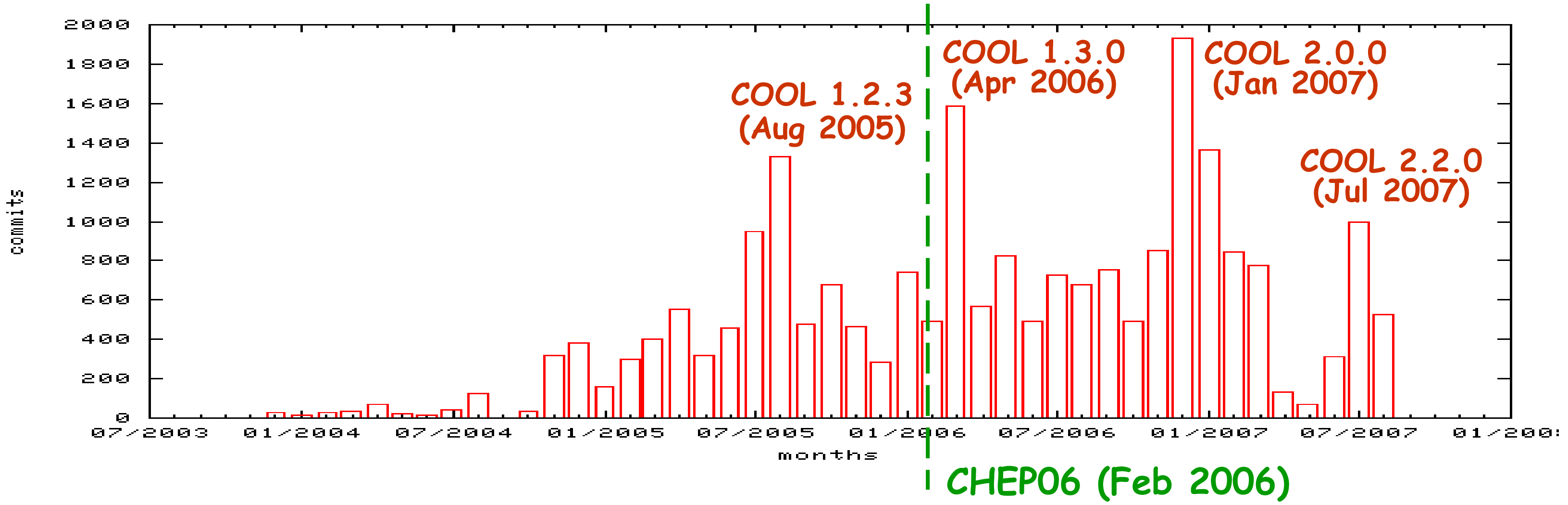
Former collaborators

- *Uli Moosbrugger* (Mainz ATLAS) – performance optimization
- *Kristoffer Dahl* (CERN IT summer student) – performance tests

PSS

Developments since CHEP'06

CERN IT Department



- **COOL 2.0 development took 9 months**
  - Parallel development on the CVS 1.3.x (bug-fix) and HEAD branches from May 2006 to Jan 2007
  - *Main focus: major API and schema changes* (group all backward-incompatible changes)
  - Extensive testing (in multi-threaded environment)
- **Further improvements in COOL 2.1 and 2.2**
  - *Main focus: performance optimizations*

CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

PSS

C++ API changes

CERN IT Department

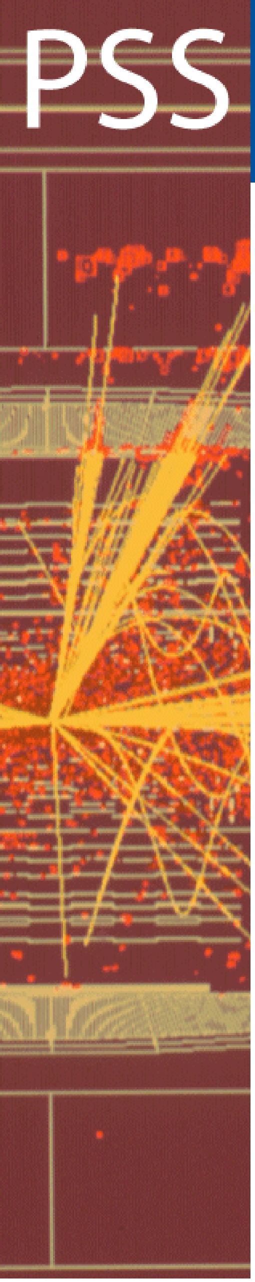
- **Motivation**
  - *Functional enhancements* (e.g. tag locking, channel metadata, BLOB payload support)
  - *Non-functional* (64 bit platform support, remove SEAL classes from API before SEAL phaseout)
  - *Performance speed-up* (C++ client profile)
- **From CORAL Attribute's to COOL Field's**
  - coral::Attribute is better suited for transient data
    - "long" integer is 32bit on i386 and 64bit on AMD64
  - New StorageType specifies *persistent* precision
    - Examples: "Int64", "String255", "Blob16M"
  - Data validation on Field construction/assignment
    - Exception thrown if String255 value is set to 256 chars

CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

New C++ API: example

```
// Create new folder and insert new data
cool::Application app;
cool::DatabaseId url = “oracle://int5r;schema=lcg_cool;dbname=TEST”;
cool::IDatabasePtr db= app.databaseService().openDatabase( url, false );
cool::RecordSpecification payloadSpec;
payloadSpec.extend( “I64”, cool::StorageType::Int64 );
cool::FolderSpecification folderSpec
  ( cool::FolderVersioning::SINGLE_VERSION, payloadSpec );
cool::IFolderPtr folder = db->createFolder( "/myfolder", folderSpec );
cool::Int64 i64 = cool::Int64Max; // user payload value is 2^63-1
cool::Record payload( payloadSpec );
payload[“I64”].setValue( i64 ); // performs data validation!
folder->storeObject( 0, 100, payload, 1 ); // t=[0,100] in channel #1

// Read back data from folder
cool::IObjectPtr obj = folder->findObject( 50, 1 ); // t=50 in channel #1
const cool::IField& field = obj->payload()[“I64”];
cool::Int64 i64 = field.data<cool::Int64>(); // const data accessor
std::cout << “I64 at t=50 in channel #1: ” << i64 << std::endl;
```

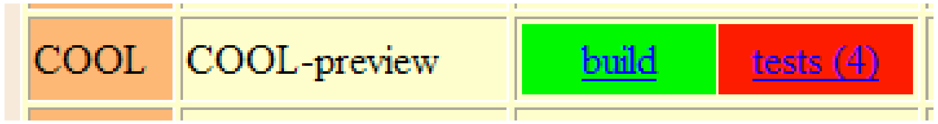


Relational schema changes

- **Motivation**
  - *Functional enhancements* (e.g. tag locking, channel metadata, dynamic replication)
  - *Non-functional* (reduce impact on the users of more granular schema changes in the future)
  - *Performance speed-up* (e.g. better SQL, indexes)
- **Schema evolution tools**
  - 1.3 to 2.0 (mandatory); 2.0 to 2.2 (recommended)
- **Schema evolution tests**
  - Create a db using COOL 1.3, evolve it to COOL 2.0, test data retrieval and insertion of new data
  - In the standard suite executed for every release

Improved building and testing

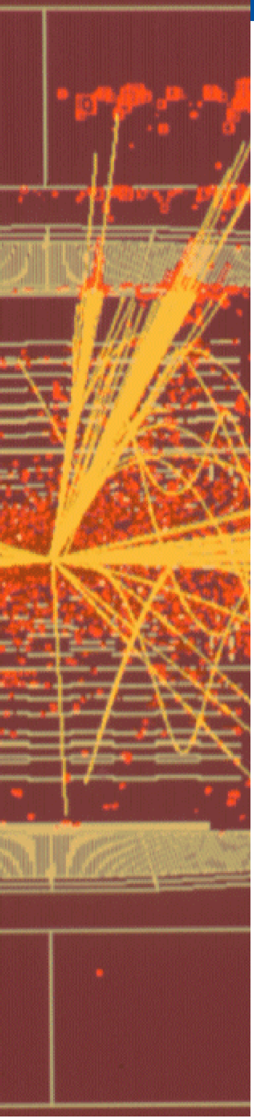
- **New build infrastructure**
  - CMT replaced SCRAM as LCG AA build system
- **Extended and enhanced test suite**
  - New "reference database" package with wide data type coverage, schema evolution tests…
  - Full integration with QMTEST
- **Automatic nightly builds and tests**
  - Several slots (bug-fix branches and HEAD)
  - Oracle, MySQL, SQLite, Frontier are all tested



COOL	COOL-preview	build	tests (4)

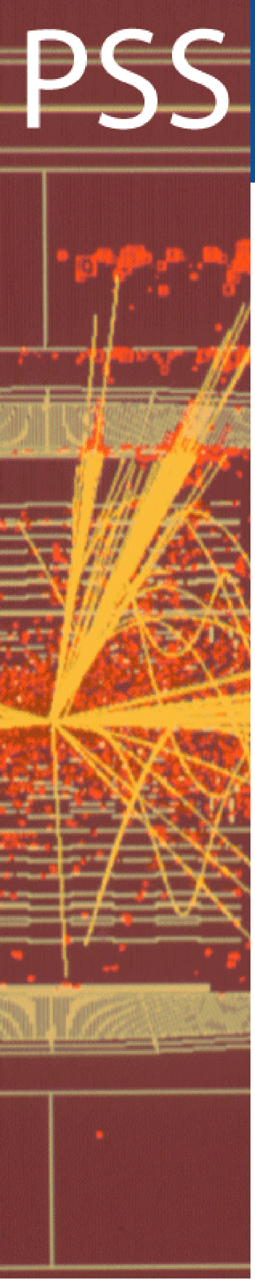
Support for new platforms

- **64bit Linux (AMD64)** – since COOL 2.0.0
  - New StorageType API removes COOL 1.3 ambiguity for ‘long’ precision on 32bit/64bit Linux
  - CERN lxplus cluster moved to AMD64 in 2007
- **MacOSX (Intel)** – soon in COOL 2.2.1
  - COOL ok (but no client released by Oracle yet)
- **MacOSX (PowerPC)** – will not be supported
  - No PowerPC support in AA (Architects Forum)
  - COOL ok (but no PyCool: problems in PyROOT)
- **gcc41 on Linux** – soon in COOL 2.2.x
  - COOL ok (but minor LCGCMT issue is pending)



Enhanced Grid-awareness

- **Improved DB authentication and lookup**
  - New URL syntax: “alias(role)/dbname”
    - Support CORAL authorization roles (e.g. reader)
    - Support Grid database lookup via CORAL alias
  - *Integration with CORAL LFCReplicaService*
    - Look-up DB and fetch authentication credentials too
- **Dynamic replication tools**
  - Create a replica of the database at time t1
  - At a later time t2, replicate all changes since t1
- **Frontier support (with Squid web cache)**



PSS

Performance optimizations

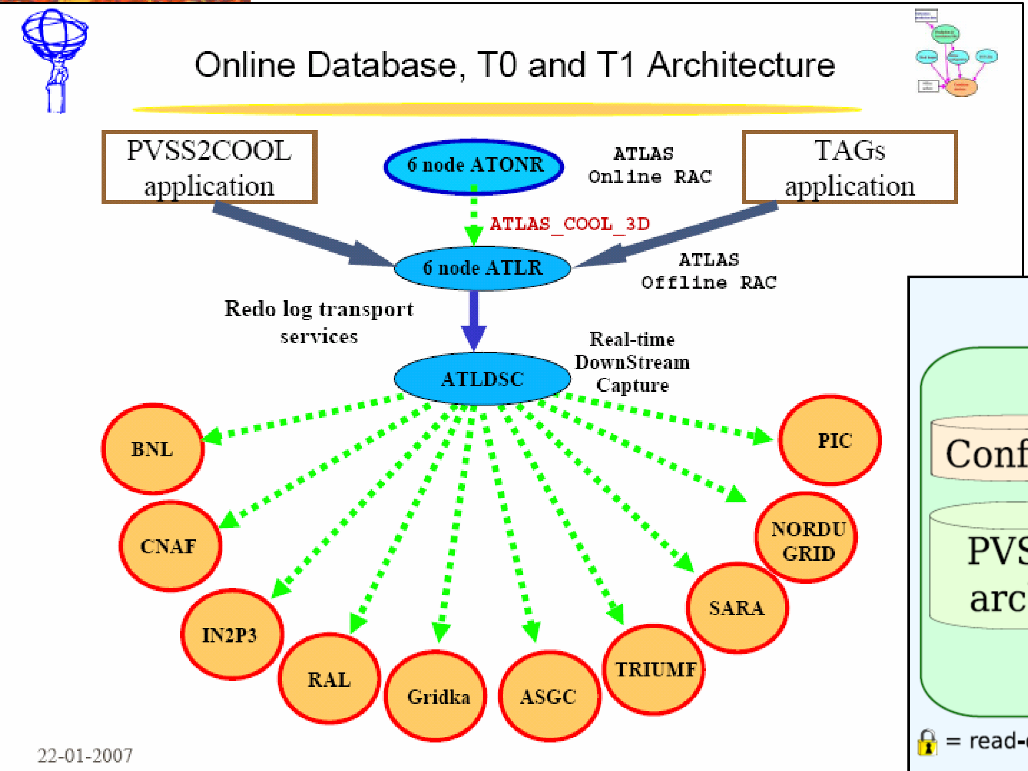
CERN IT Department

- **Server-side SQL optimizations**
  - Oracle only (may not apply to MySQL or SQLite)
  - Through several means combined
    - Reengineer the SQL query strategy
    - Add missing indexes
  - Several use cases *independently* improved
    - Single-version single-channel IOV retrieval
    - Single-version multi-channel IOV retrieval
    - Multi-version user-tag IOV retrieval
    - Multi-channel bulk insertion
  - *Still the main priority also for future development!*
- **Client-side profile optimization**
  - Reduce COOL overhead over CORAL by avoiding extra in-memory copies of the data

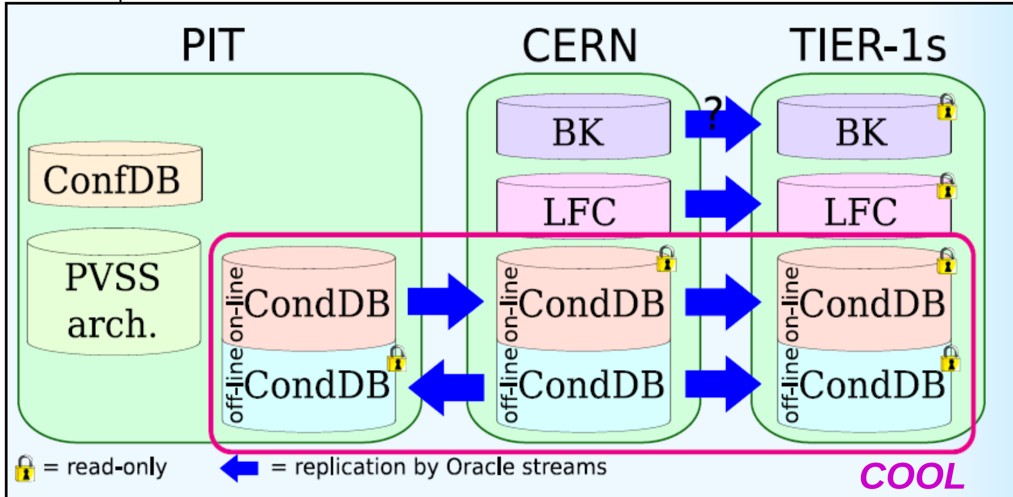
Deployment overview

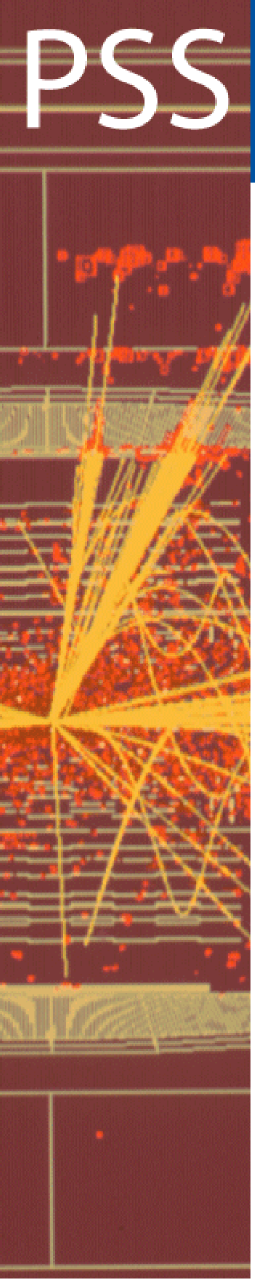
- **Similar Oracle setups in Atlas and LHCb**
  - Two separate servers at CERN (online, offline)
  - Distributed replicas at the experiment Tier1's
  - *Replication via Oracle Streams (see 3D talk)*

***Atlas** (G. Dimitrov, F. Viegas)*



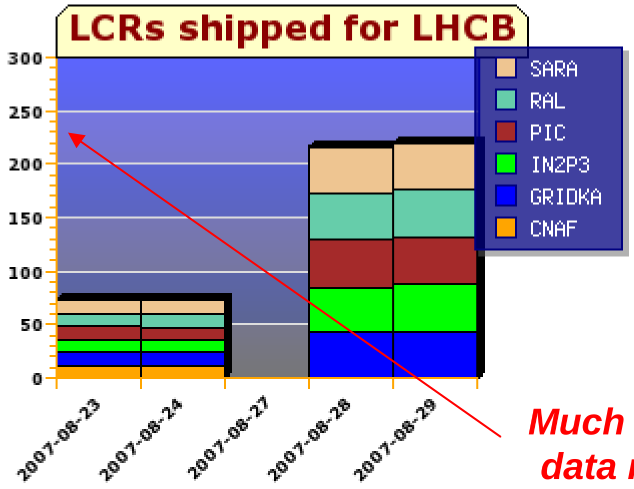
***LHCb** (M. Clemencic)*

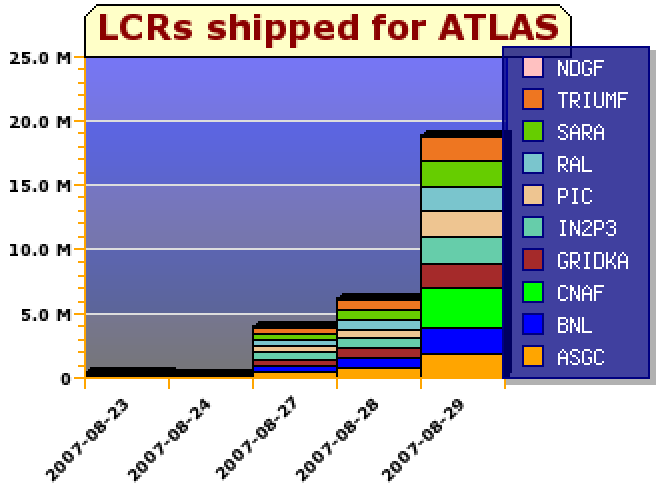




Deployment status

- **The T0 setup is (almost) complete**
  - The LHCb online server will join soon after CHEP
- **Atlas and LHCb T1 sites are all connected**
  - SARA, RAL, PIC, IN2P3, Gridka, CNAF (both)
  - Plus Nordugrid, Triumf, BNL, Taiwan (Atlas only)
  - Distributed tests underway in both experiments





Much larger data rates in ATLAS!

Conclusions

- **Many major improvements in COOL 2.0**
  - Major API and schema changes
  - Planning ahead for future enhancements
- **Development is far from finished**
  - Performance is now the highest priority
    - Many optimizations released in COOL 2.1 and 2.2
  - Work ongoing also on a few new functionalities, code and test consolidation, new platforms…
- **Distributed deployment (almost) complete**
  - *Services must be tested well before LHC startup*