The ALICE Offline Environment

F. Carminati
13 June 2001
ALICE Offline Project

ALICE is unique in the LHC context

- Major decisions already taken (DAQ, Off-line)
- Move to C++ completed
- Adoption of the ROOT framework
- Tightly knit Off-line team
- No ongoing “turf war”
- Physics performance and computing in a single team
- One single development line
- Aggressive Data Challenge program
- Impressive achievement record
- And of course
- Minimal support from CERN (apart for ADC’s)
LHC Computing Review

**Recognition of**
- Size and cost of LHC computing (240MCHF by 07 + 80/y after)
- Shortage of manpower in LHC core software teams
- Importance of Data Challenges

**Recommendations for**
- Common prototype (2002-2004) & projects (IT+LHC)
- CERN support of FLUKA / ROOT
- Establishment of SC2 to oversee LHC computing
- Drafting of a computing MoU


The battle is not ended, it just started!
ALICE Software Process highlights

- Freely inspired by the most recent Software Engineering trends
  - Extreme Programming Explained, Addison Wesley
  - Martin Fowler, The New Methodology,
  - Linux, gnu, ROOT, KDE, GSL...

- Exhaustive code documentation made nightly and on the web
  - UML diagrams, code listing, coding rule violations

- Continuous code testing & verification, nightly build and tests

- Collective ownership of the code one single repository

- Flexible release cycle: release often, release early

- Simple packaging and installation
  - Code is composed only of 2 main packages (ROOT and AliRoot)

- We could not have done it without ROOT
ALICE Software Process synoptic

- Requirements are necessary to develop the code
  - ALICE users express their requirements continuously

- Design is good for evolving the code
  - Users participate in the system design continuously

- Testing is important for quality and robustness
  - Testing is done automatically every night

- Integration is needed to ensure coherence
  - AliRoot is completely integrated every night

- Discussions are valuable for users and developers
  - Discussion is continuous in the ALICE offline list
Status of AliRoot

Version v3-05 just released

- Foreseen stable for 6 months
- Folder structure
- Major refactoring of the code AliStack introduced
- Reconstruction for all detectors from digits
- TRD tracking
- Inverse Kalman filter
- Major step toward global tracking for ITS – TPC – TRD
- Abstract interface between VMC, G3 and G4 FLUKA (in preparation)
- AliGenerator interface improved / extended (cocktail & afterburners)
Folders

- Long (stormy) discussions with René on the role of folders
  - We had to clarify what is the relative role of object design, folders structure and relation with containers
  - Now this is clear, and AliRoot is posting shared data structures to folders
  - It came at the right moment for us, as we are designing now global reconstruction
  - We should think about the relation between folders, GRID LFNs and datasets
  - Forward evolutive concept
Tasks

Less clear (at least for me) the use of tasks
- Elegant way to generic programming
- Going from data encapsulation to method encapsulation
- For the moment we are afraid by the lack of “expressivity” of the TTask->Execute()
- Also the necessary refactoring of the code is probably rather large

More prototyping is needed before a decision is taken
Update on VMC

- The G3 interface is in production and evolving
  - Possibility to translate geometry is TGeometry will be in v3-06 pre-release

- The G4 interface is technically working but
  - G4 is a monolithic and non-modular program
  - Conversion from GEANT3 geometry is heavy
  - The question of MANY is still unsolved
  - The physics of G4 seems out-of-control

- Work on the FLUKA interface is ongoing, but the tracking / stacking is different from G3 / G4 and this may be a major challenge
The Virtual MC

- **User Code**
- **VMC**
  - **G3**
    - **G3 transport**
    - **G3 geometry**
  - **G4**
    - **G4 transport**
    - **G4 geometry**
  - **FLUKA**
    - **FLUKA transport**
The framework question

1990
- PAW
- GEANT3

2000
- ROOT

Recons?

Discussion on Thu?

1990 -> 2000
PAW -> OK
GEANT3 -> LOST
ROOT

GEANT4

TParticle
VMC
TGenerator

Discussion on Thu?
The Framework question

- The problem of the MC is clearly not solvable in this context
- However a very important piece on which we can work is the Geometrical Modeller
  - G3 modeller had an incredible longevity, but it has several shortcomings
  - This has been one of the areas of IT most active in these past 20 years
- A possible way ahead
  - Evaluate few candidates (I have at least 2/3)
  - Define a ROOT abstract interface for CSG modelling
  - Interface with ROOT 3D graphics
  - Define a representation on file
The Virtual MC II

User Code → VMC

Reconstruction

Visualisation

Virtual Geometrical Modeller

G3 transport

G4 transport

FLUKA transport

Geometrical Modeller
ALICE Data Challenges

- ALICE plans to acquire data at 1.25GB/s
- To be confident we will be able to do that we run yearly data challenges of increasing complexity and size
- The goals ADC III (1H2001) were:
  - Stable bandwidth of 100 MB/s over the complete chain during a week
  - Aggregate bandwidth of 300 MB/s in the DAQ
  - Total 80 TB of data stored in MSS
  - Testing of SMP servers
  - Online monitoring tools
  - Regional centres (not done!)
ADC III hardware

- LHC testbed farm (ADC using part of the testbed)
  - Low-cost PCs only: 2 Intel CPUs running Linux
  - 35 CPU servers (22 on Gigabit Ethernet, 10 on Fast Ethernet)
  - ? disk servers
  - 12 tape servers, 12 tape drives, 1000 tape cartridges
  - Fast/Gigabit Ethernet network
    - 6 switches from 3 manufacturers on 2 media

- ALICE servers
  - 3 PC servers: 6 Intel CPUs running Linux used as disk servers (joint project ALICE/HP)

- In total: ~ 10 % of the ALICE DAQ
Alice ADC III

Event building (DATE)

Recording (ROOT+RFIO)

Mass Storage (CASTOR)
DATE + ROOT I/O

rate [MB/s] vs time

rate
Nent = 6581531
Mean = 1.021e+06
RMS = 4.391e+05

Wed Feb 7 12:02:58 2001
DATE + ROOT I/O + CASTOR

Writing to local disk
Migration to tape
DATE + ROOT I/O + CASTOR

rate [MB/s] vs time

rate
Nent = 7764561
Mean = 4.534e+05
RMS = 3.34e+05
DATE + ROOT I/O + CASTOR

Rate [MB/s] (zero suppressed)

avrate
Nent = 613073
Mean = 82.82
RMS = 24.67
Highlights

- Excellent system stability during 3 months
  - Max DATE throughput: 550 MB/s (max) 350 MB/s (ALICE-like)
  - Max DATE+ROOT throughput: 240 MB/s
  - Max DATE+ROOT+CASTOR throughput: 120 MB/s
    - average during several days: 85 MB/s (> 50 TB/week)
  - 2200 runs, 2* 10^7 events, longest DATE run (86 hours, 54 TB)
  - 500 TB in DAQ, 200 TB in DAQ+ROOT I/O
  - CASTOR > 100,000 files of 1 Gbyte => 110 Tbytes
  - Database for metadata with 10^5 entries

- HP SMP’s: cost-effective alternative to unexpensive disk servers

- Online monitoring tools developed

Concerns

- LHC Computing Prototype should receive adequate resources for software and hardware
Performances and plans

- **DAQ**
- **Mass Storage (sust)**
- **Mass Storage (peak)**

![Graphs showing performances and plans from 1999 to 2006 for DAQ, Mass Storage (sust), and Mass Storage (peak).]
ADC IV

- Planning and resources under discussion
- Test of individual components before next ADC

Goals:
- Increase performances (200MB/s to tape, 1GB/s through the switch)
- Study key issues such as computers and fabric architecture
- Include some L3 trigger functionality
- Involve 1 or 2 regional centres

- Global ADC second half of 2002 (new tape generation and 10 Gbit Eth.)
DataGRID project status

- Huge project
  - 21 partners, 10MEuro, 3 years, 12 WorkPackages
- Global design still evolving
  - An ATF has been mandated to design the architecture of the system to be delivered at PM6 (=now!)
  - To progress they need continuous feedback from the users
- Users are in three workpackages
  - HEP in WP8
  - Earth Observation in WP9
  - Biology in WP10
- ALICE long term project is distributed interactive data analysis
DataGrid & ROOT

TagDB → Selection
Parameters
Procedure

RDB

Proc.C → DB1 → CPU

CPU

PROOF

Local

Remote

TagDB → Proc.C → DB1

Proc.C → DB2

Proc.C → DB3

Proc.C → DB4

Proc.C → DB5

Proc.C → DB6
Comments to ROOT team

- Really no hard comment to the root team
- The release policy on CVS could include a branch for every version
  - This would allow to make "patches" to production versions instead of being forced to take the HEAD to have a fix
Conclusion

- The development of AliRoot continues happily
- The organisation of the Off-line project is more and more oriented toward an OS distributed development model
  - ROOT fits perfectly in this scheme
- We are confident that ROOT will evolve to meet our requirements in 2007 and beyond
  - We will try hard to have the recommendations of the LHC Computing Review implemented at CERN
- We are grateful to the ROOT team for their continued and excellent support