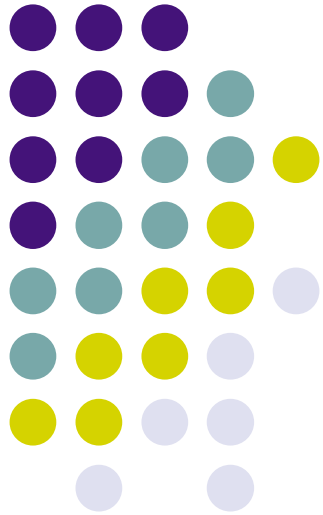
Parallel ALICE offline reconstruction with PROOF

C. Cheshkov, P. Hristov

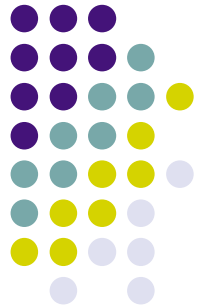
on behalf of ALICE Core Offline Team

24/03/2009

CHEP09

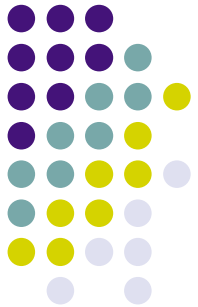


Outline

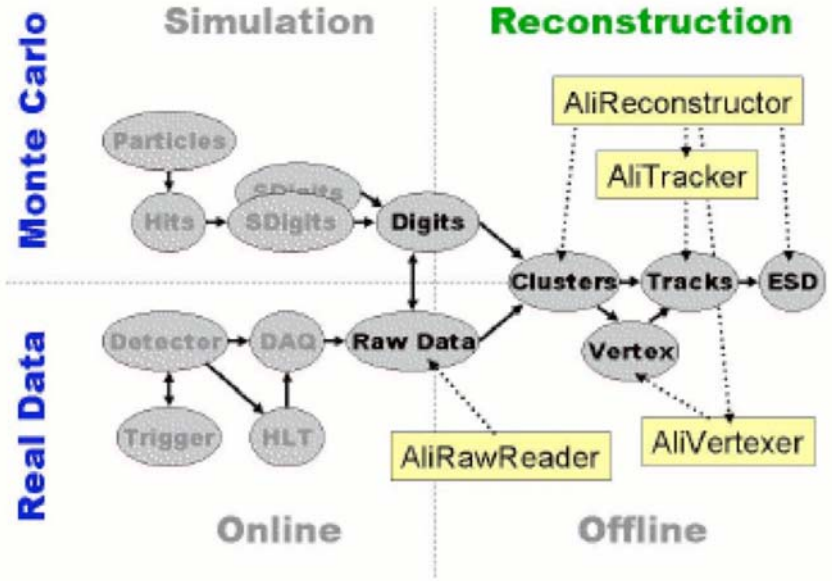


- Introduction to ALICE raw data and reconstruction
- Parallel reconstruction with PROOF
  - Motivation
  - Design and implementation
  - Performance on ALICE CAF
- Conclusions & Outlook

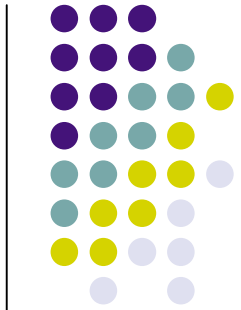
ALICE Offline Reconstruction



- Raw data from ALICE detector
- Local reconstruction (clusterization)
- Vertex finding
- Tracking
- Particle identification
- Reconstructed data -> Event Summary Data (ESD) ROOT tree

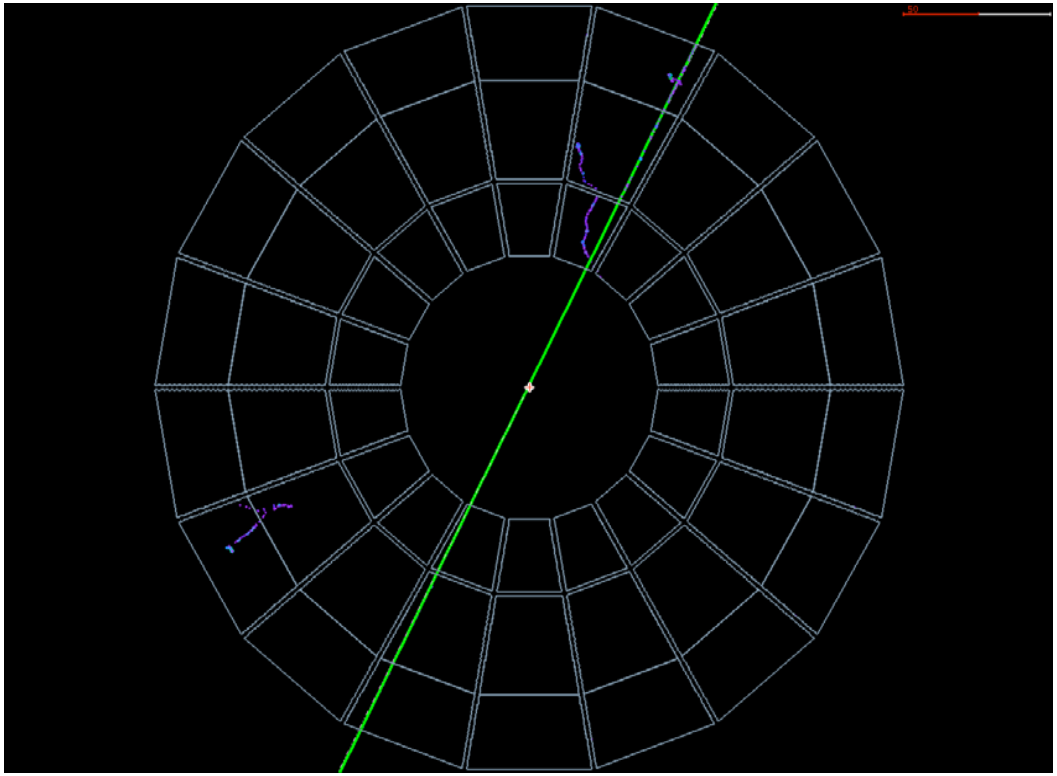


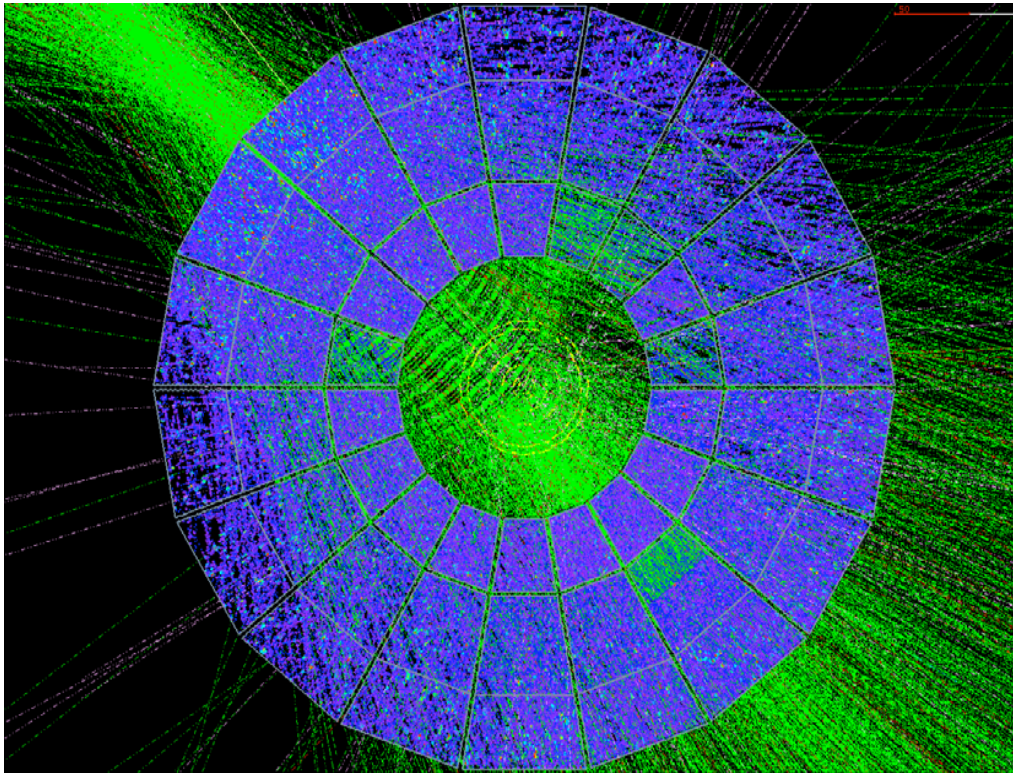
ALICE Offline Reconstruction



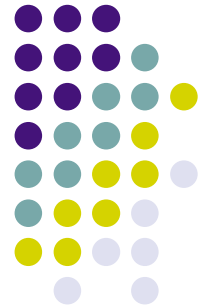
- ALICE raw data:
  - Events are stored as entries in ROOT tree
  - Loading of various detectors data on demand
  - From few to ~10^6 events in one raw-data chunk
  - Size: from few KBs up to ~100 MB/ev. depending on active detectors, collision type, luminosity, etc.
- Reconstruction:
  - Involves sophisticated algorithms
  - Recons. time varies from fraction of s to ~200s
- Event Summary Data (ESD):
  - Contains all information relevant for physics analysis
  - Size: at least one order of magnitude smaller than raw-data size

ALICE Offline Reconstruction



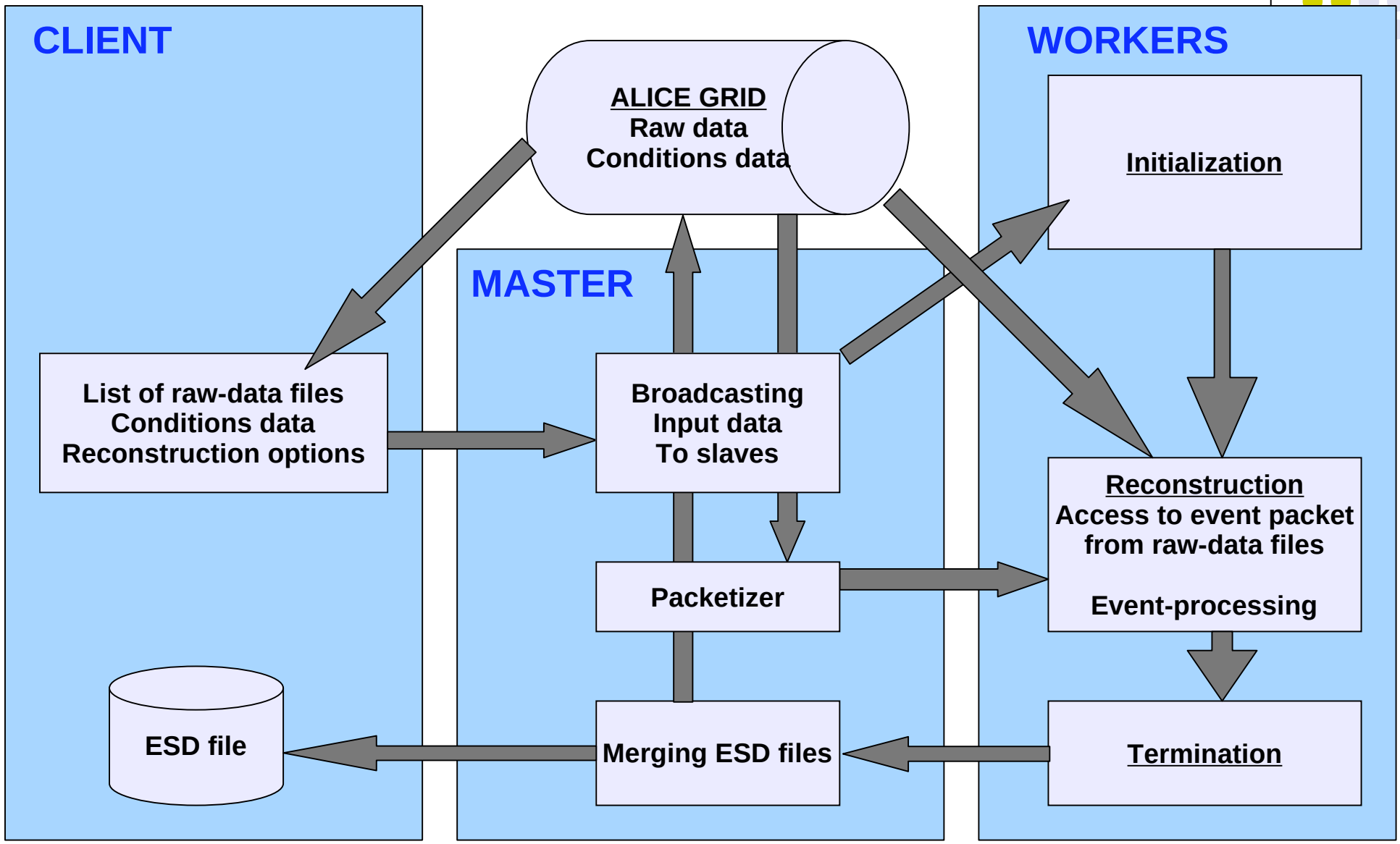


Parallel Reconstruction: Motivation

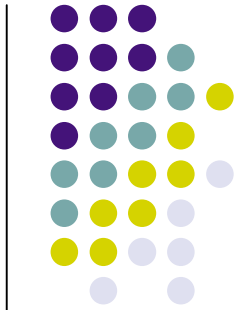


- Fast feedback from reconstruction
  - Understand ALICE detector and reconstruction software
  - Debug, tune and optimize reconstruction code
- Not a replacement for central ALICE GRID (AliEn) based reconstruction

Parallel Reconstruction: Design

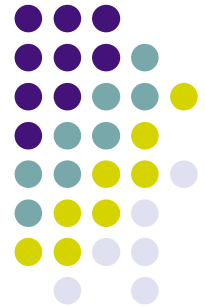


Parallel Reconstruction: implementation



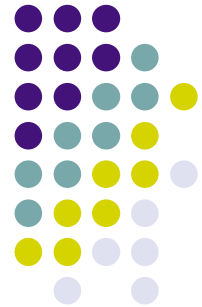
- Fully based on PROOF (TSelector)
  - All platforms supported by ROOT can be used
  - No additional code/libraries are needed
- Transparent
  - User does not notice a difference w.r.t to running locally
- Minimal data flow between components:
  - Common (conditions and options) data accessed once from the client machine
  - Workers access raw-data events directly from AliEn
- Minimal I/O on the workers
  - Diminishing number of intermediate files

Differences w.r.t. to normal Alien based reconstruction job



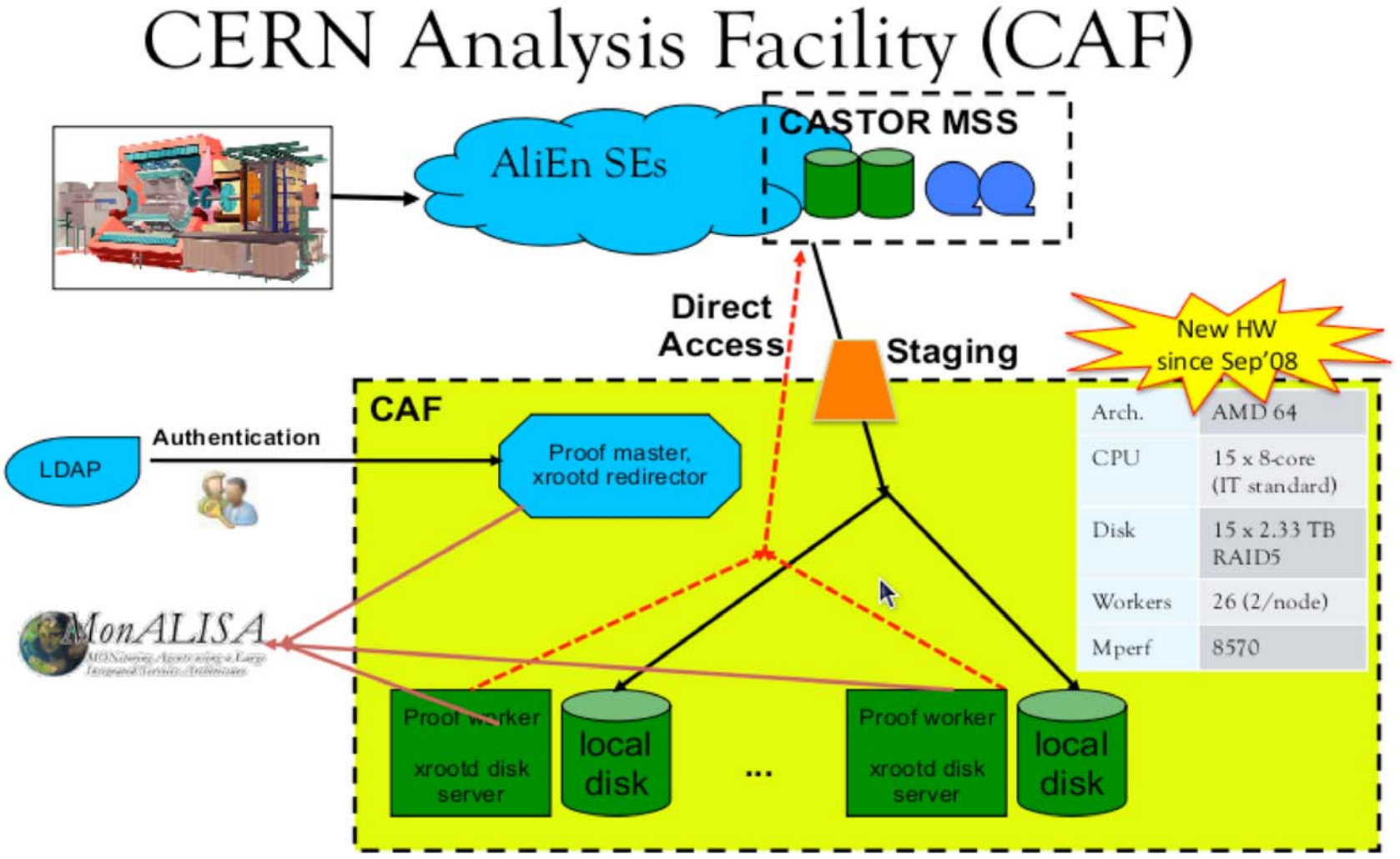
- Parallelization at event level (at file level in AliEn)
  - Allows faster execution in case of small number of big raw-data files
- Conditions data/reconstruction options sent from client -> workers (in AliEn worker nodes access directly conditions data from file catalogue)
  - Allows user to test custom conditions data and/or reconstruction options
- Output ESD file is sent back to the client machine (via xrootd)
  - Allows immediate access and check of the reconstructed data quality

Input-data handling

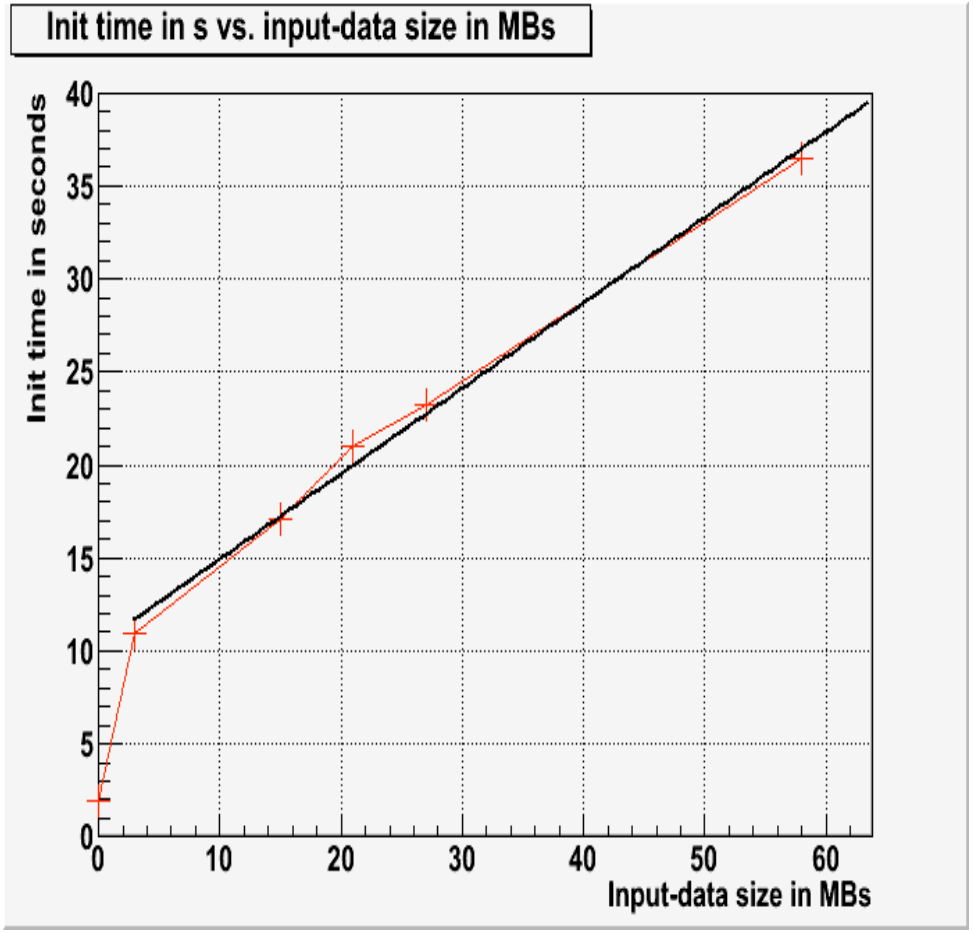


- Contrary to the other 'normal' PROOF based processing/analysis, the raw-data reconstruction needs relatively big initialization data – field map, geometry, detector offline conditions DB (OCDB) entries
  - Size: from few to ~100 MB
- Input data distributed a la ROOT “par” packages
  - Input objects assembled into an input file
  - Input file distributed on each unique file-system of the slaves
  - On the slaves input objects are extracted from the file

Performance on ALICE CERN Analysis Facility (CAF)



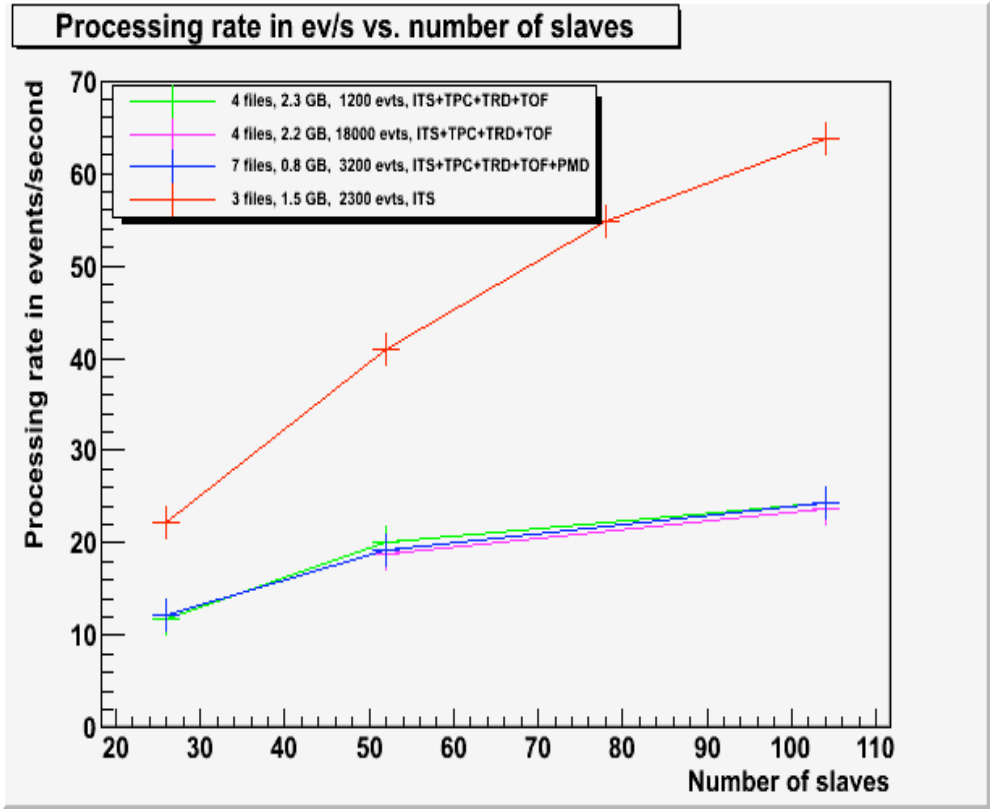
Performance – initialization time

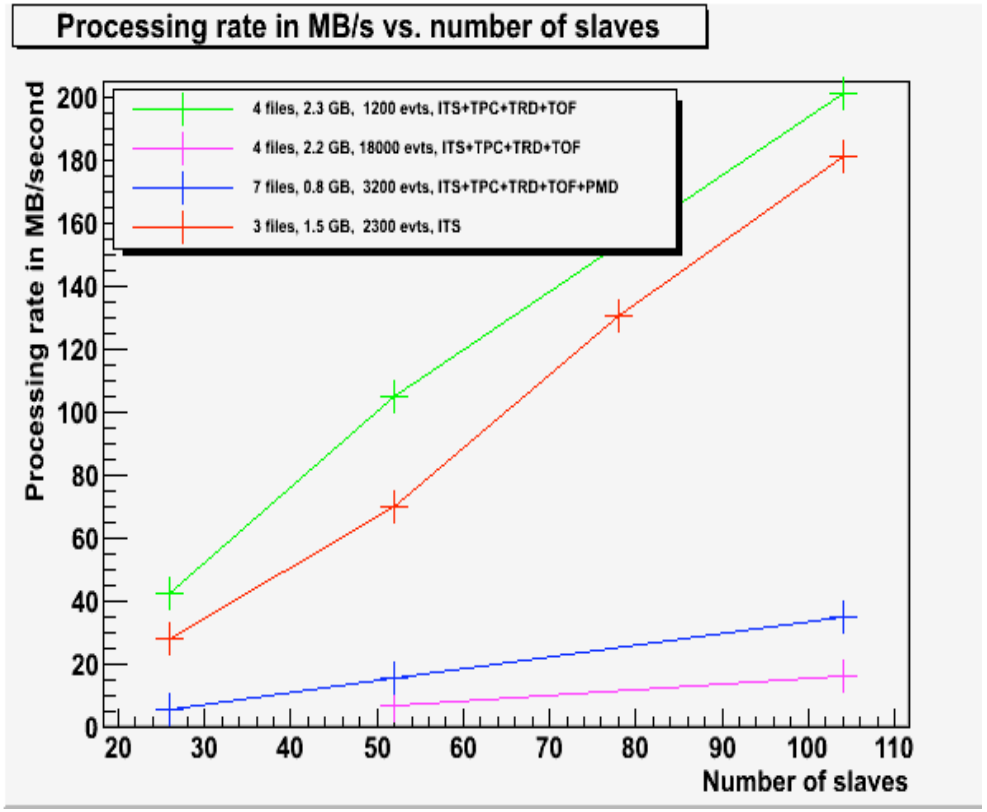


- As expected does not depend on # of slaves activated
- Typical initialization time:
  ~ 10s + 0.5 s/MB
- Jump at ~3 MB due to geometry 'unpacking'

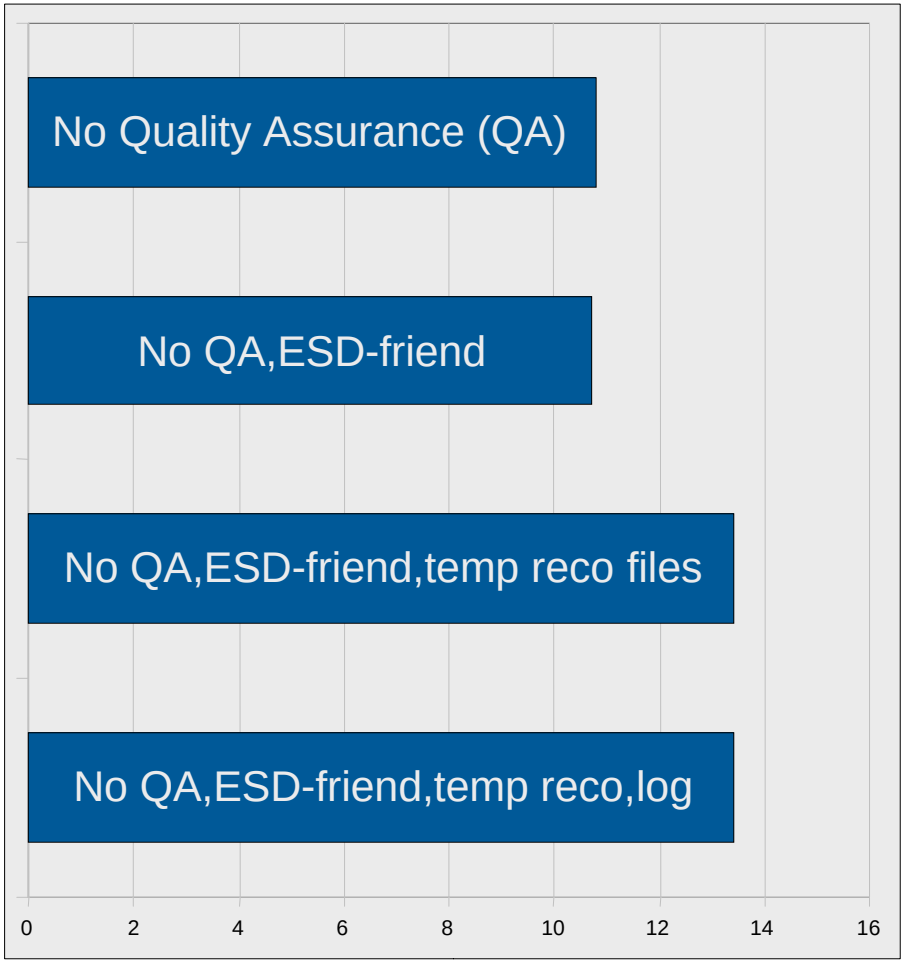
Performance – processing rates

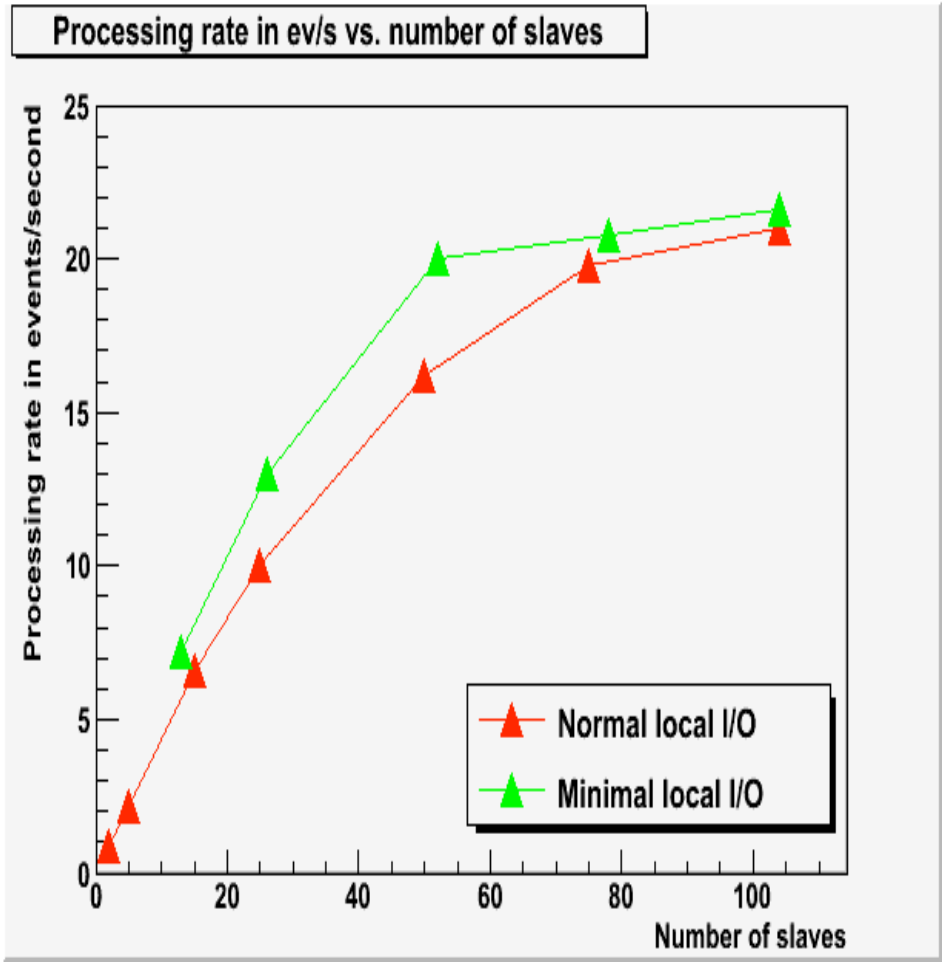
- Tested with various raw data (different participating detectors, raw-data file/event sizes)





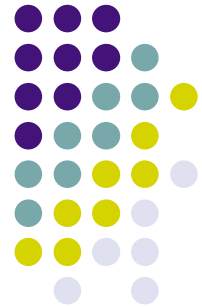
Performance – dependence on local I/O





Processing rate in events/s (4 files, 2.2 GB, 1200 ev.)

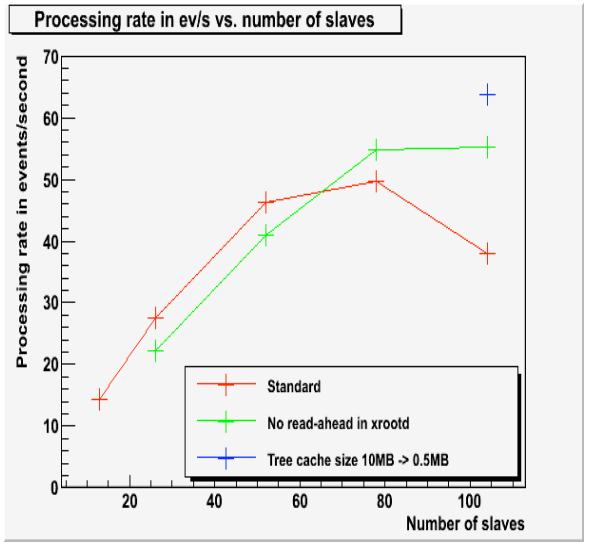
Observations



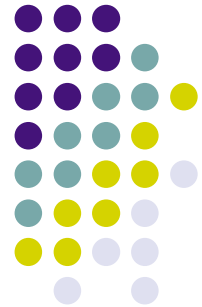
- Linear increase in the processing rate up to ~40 workers, “saturation effect” with more workers
  - The current CAF provides 26 workers for a user session, the speed-up in the processing is ~30 times, sufficient for fast feedback.
- The “saturation” is more important for events with small size
- The size of the output doesn’t affect significantly processing rate
- => Investigation of the xrootd IO and the tree cache

Test w/o read-ahead in xrootd client and smaller tree cache

- The xrootd read-ahead is essential:
  - The “saturation” effect is pronounced for all runs with small event size, if no read-ahead
- The tree cash size may play significant role
  - Can be optimized
  - Some instabilities with too small cache size

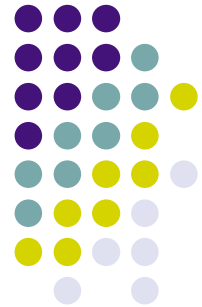


Conclusions & Outlook

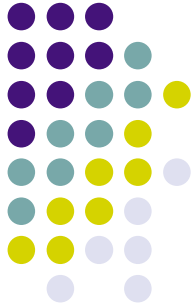


- The parallel Proof based offline reconstruction is designed, implemented, tested and ready for the data taking
- It permits fast feedback immediately after the storage of the raw data at Tier0
  - The current CAF provides ~30-fold speed-up in the processing rate for every user

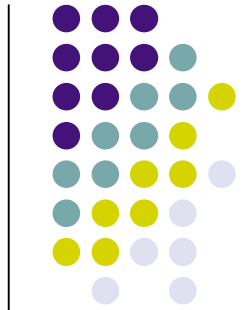
Conclusions & Outlook



- Investigate scalability
  - ROOT tree cache vs event-packet size (packetizer)
- Benchmark with PROOF Lite on multicore machines
  - The code does not have to be changed at all
- Try further optimization of input data handling and output file merging



Many thanks for ALICE CAF and ROOT PROOF teams for their great help and support!



SPARES

Exercise with local data-set