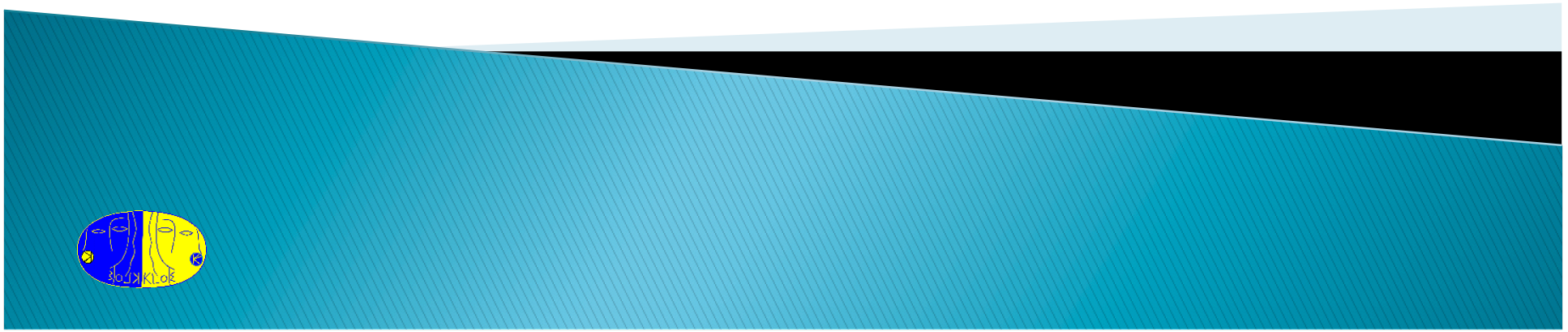
KLOE Storage Data Flow
KID – DH

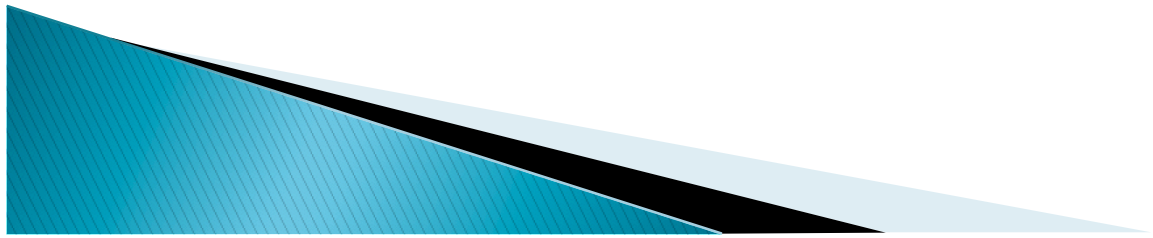
Francesco Sborzacchi



What is KID?

- KID (or KLOE Integrated Dataflow) is a software package developed to simplify the read-out of data from different sources; the source selection is achieved by mean of an URI (*Uniform Resource Identifier).*

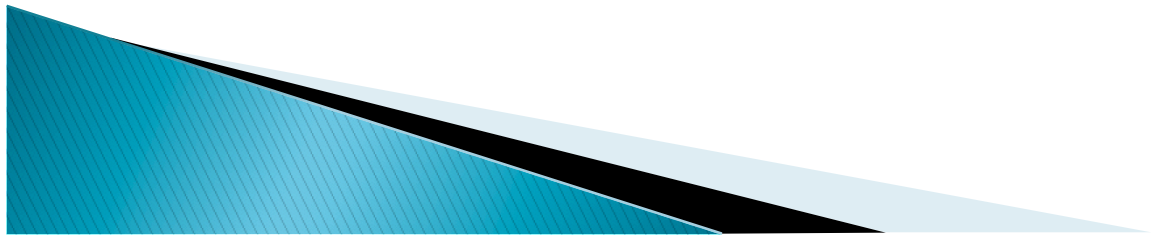
(cit. Igor Sfiligoi)



OK… What does it mean?

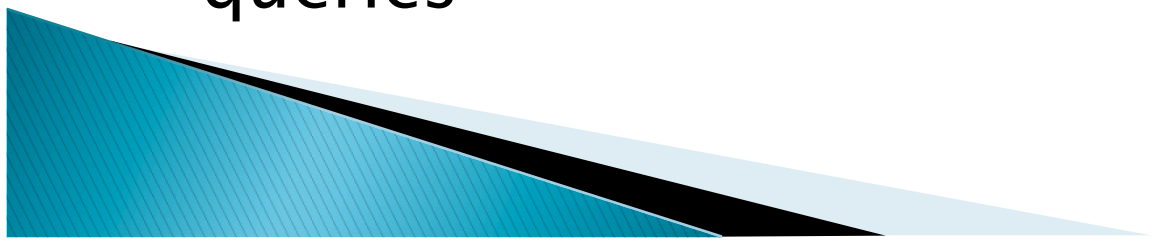
It depends, multiple answer from multiple point of view:

- SYSTEM: a set of services to handle the events produced by KLOE
- SOFTWARE: a collection of process and a C library that allows the system to handle the KLOE files independently from their current physical location
- USER: a tool for having fun easily with the data of the experiment.



In other words?

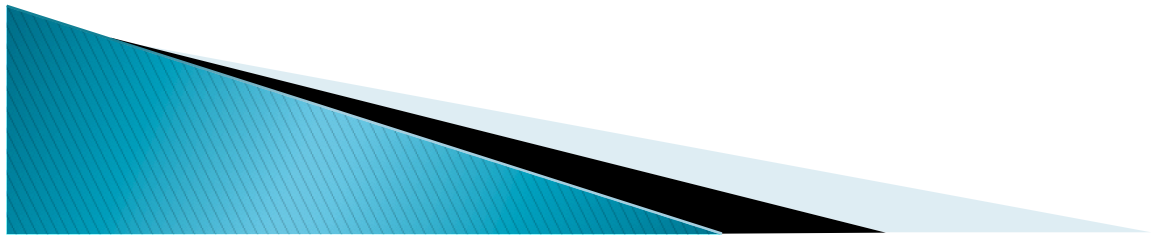
- We can define KID as a BIG FILESYSTEM
- MetaData collected and managed in the DBMS DB2
- Data (file content) stored in the underlying hardware (commercial storage solution)
- Remote file access through the middleware layer (~ equivalent file sharing like NFS/GPFS/ HSM)
- I/O commands user friendly via URI/SQL-like queries



Example:

- “Hi KID I want to read all the RAW files belonging to the run range 100 ->200 that have not yet been reconstructed”

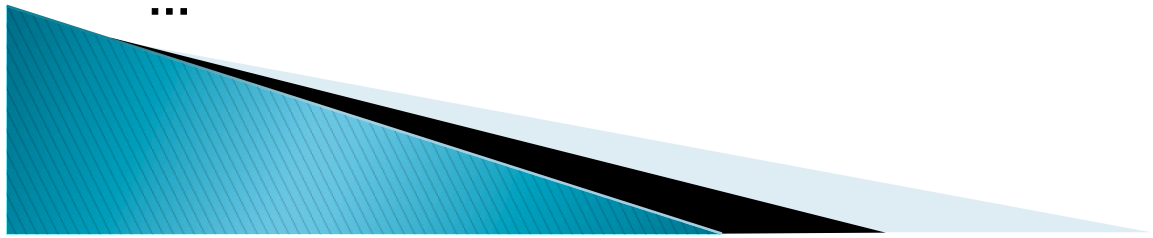
URI: *dbraw: run_nr between 100 and 200 and analyzed is null*



Example:

- fibm01:~> kls raw "run_nr=27000”

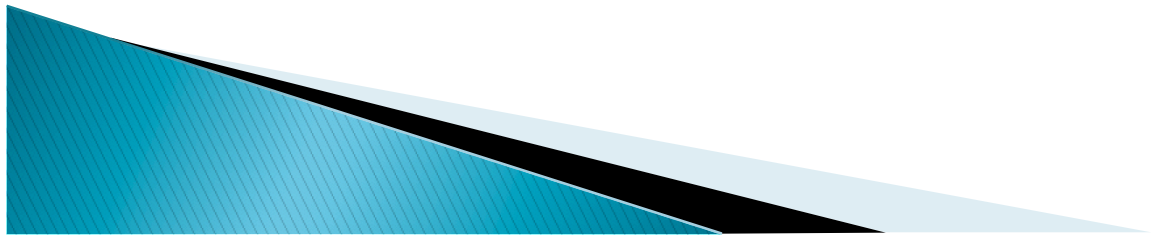
Found 27 files:
raw027000N_ALL_f05_1_1_1.000 1073.7Mb on tape
raw027000N_ALL_f05_1_1_1.001 1073.7Mb on tape
raw027000N_ALL_f05_1_1_1.002 1073.7Mb on tape
raw027000N_ALL_f05_1_1_1.003 1073.7Mb on tape
raw027000N_ALL_f05_1_1_1.004 1073.7Mb on tape
raw027000N_ALL_f05_1_1_1.005 1073.7Mb on tape
raw027000N_ALL_f05_1_1_1.006 235.4Mb on tape
raw027000N_ALL_f06_1_1_1.000 1073.7Mb on tape
raw027000N_ALL_f06_1_1_1.001 1073.7Mb on tape
raw027000N_ALL_f06_1_1_1.002 1073.7Mb on tape
raw027000N_ALL_f06_1_1_1.003 1073.7Mb on tape
raw027000N_ALL_f06_1_1_1.004 1073.7Mb on tape
raw027000N_ALL_f06_1_1_1.005 1073.7Mb on tape
raw027000N_ALL_f06_1_1_1.006 234.4Mb on tape
…



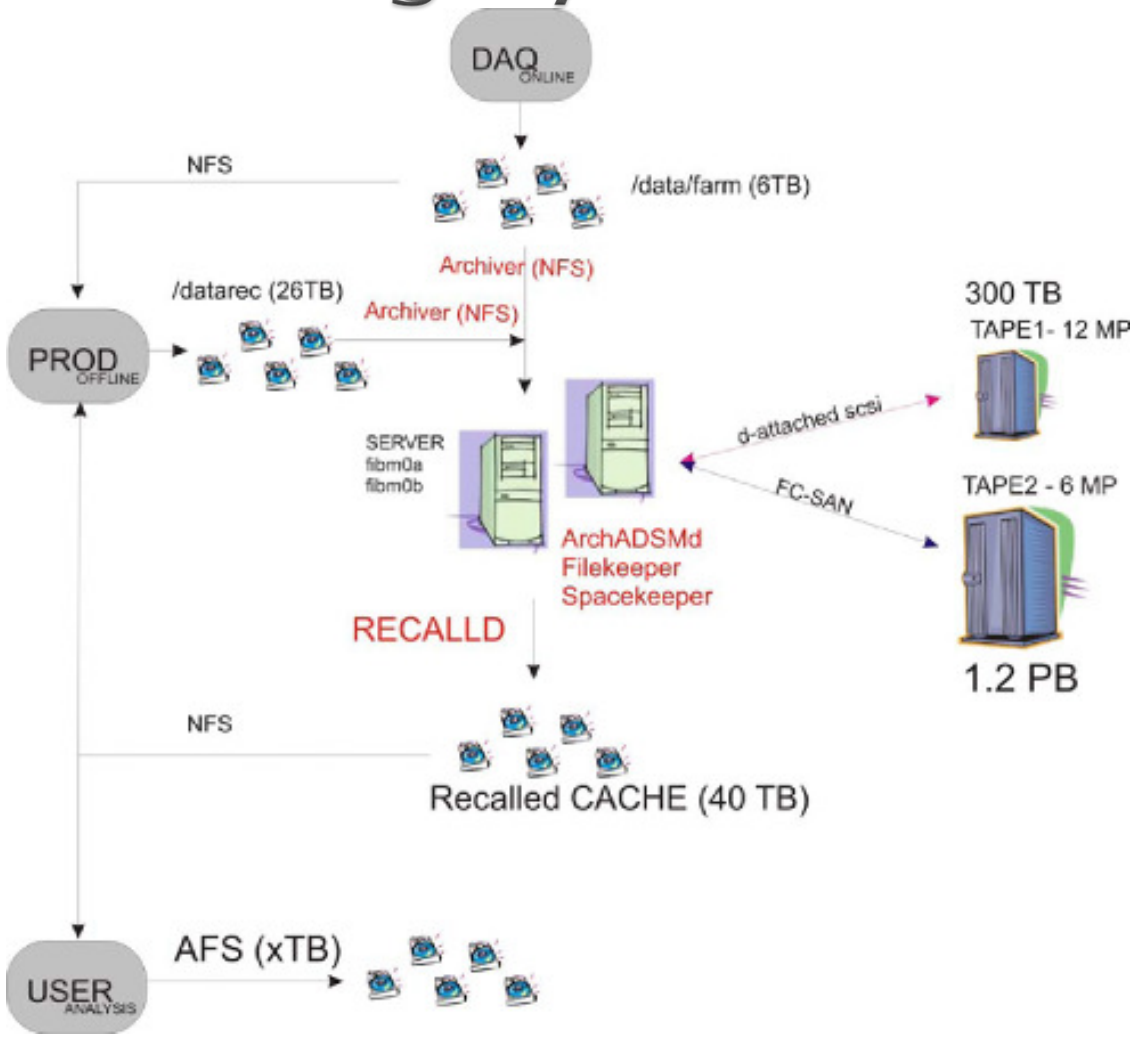
On tape? And now?

- Don’t worry the file will come to you…

KID will coordinate the distributed file-moving services



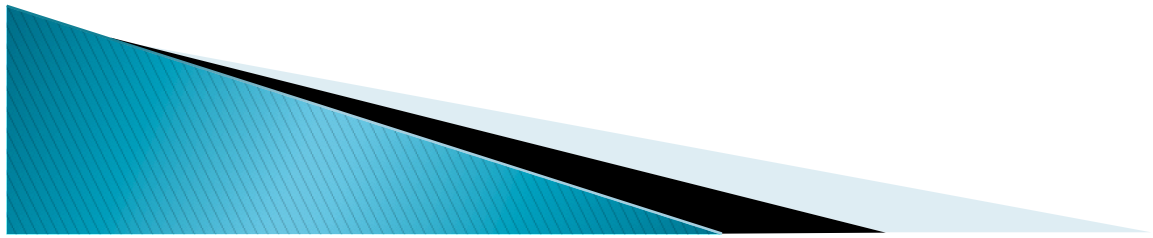
Data Handling system



Archiver

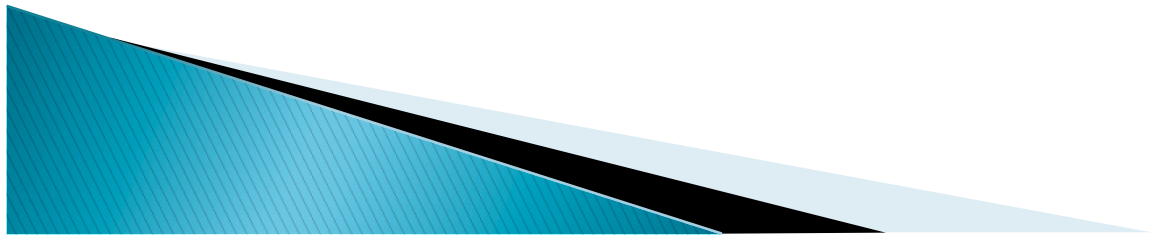
- When new data are acquired, the online servers write the raw files to the online-disk pool. These files are then asynchronously archived to the tape library over an NFS mount by the ARCHIVER daemon.
- The archiving processes are tailored to minimize the number of tape mounts while guaranteeing enough space on the disk pool.

(prediction of the future order of recall request- PREFETCHING)



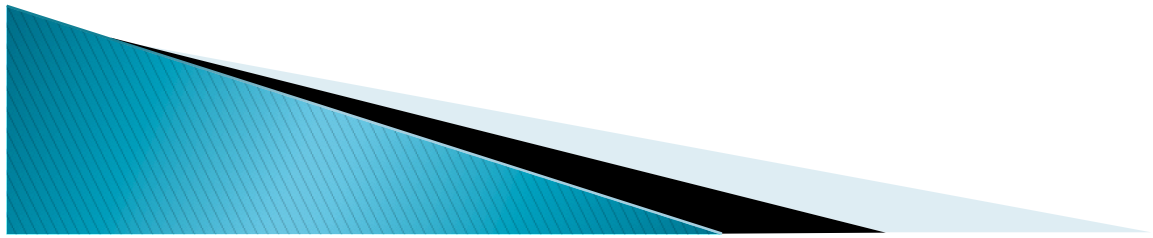
RECALL/RECALLD

- When files already archived and deleted from the online- or offline-disk pools must be processed on the offline farm, the RECALLD daemon restores the files from tape to the disk cache, from where they are served to the offline processes using the KID/NFS protocol
- PREFETCHING and MERGE of the requests are involved in this process



ArchADSMd

- It is the interface to the long-term staging storage system.
- It can give information about the files archived in the tape library and copy files to and from it
  - Local Archived files – TSM
  - CNAF Archived files – GRIDFTP

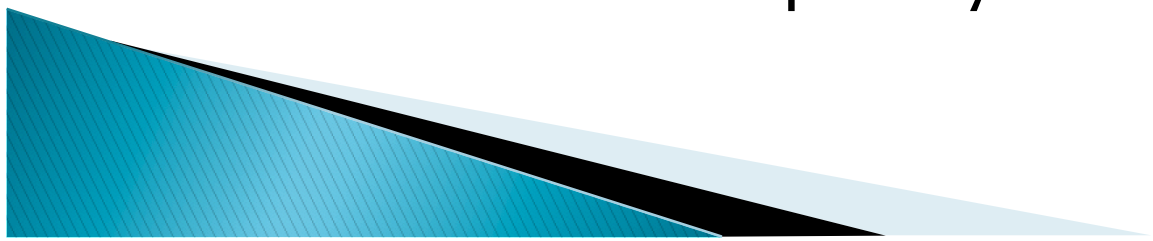


SPACEKEEPER /FILEKEEPER

- The SPACEKEEPER daemon ensures the availability of disk space in the staging areas by deleting files that have been archived
- A FILEKEEPER daemon ensures the availability of free space in the recall areas, deleting old files when necessary to make space for newly recalled data.

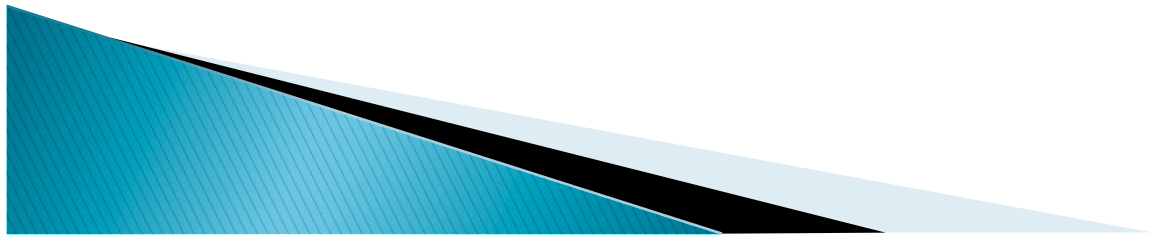
OFFLINE – Reconstruction

- Normally, reconstruction is performed while the raw files are still resident on the online disk
- For input to the reconstruction processes from the online disk, events are either read across an NFS mount or served by the data-handling system using a custom TCP/IP protocol
- Reconstruction output is written via NFS to the offline-disk pool, from which it is asynchronously archived to tape
- DSTs for each run are produced from the reconstruction output files immediately after the run has been completely reconstructed



DBMS/DB2 BOOK-KEEPING

- DB2 is used to keep track of the locations of the several million files.
- Each file is logged in the database when it is created.
- The database entry contains the reconstruction status of the file, allowing files that require processing to be easily identified.
- This database also contains run-by-run information on data-taking conditions, operational parameters and geometry of the detector (HEPDB2)

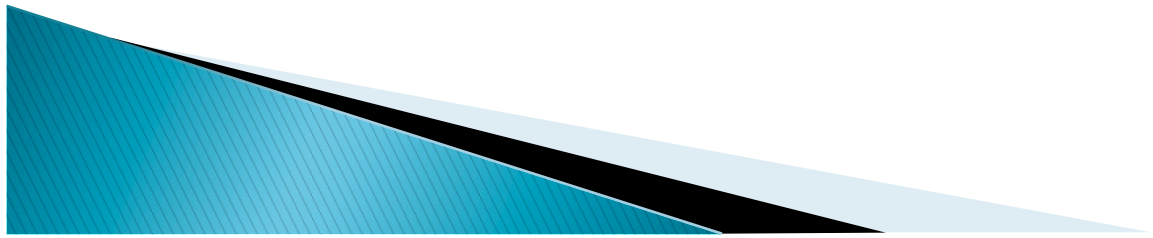


DB as MetaData Container

- communication between the clients (user applications) and the RDBMS through a set of database daemon
- The database daemon is the only link between the applications and the RDBMS
- Database consistency (no data corruption) thanks to:
  - CLUSTERED DBMS (no single point of failure)
  - TCP/IP sockets strictly pinned to the application running specific task , in case of failure a rollback is triggered in the thread leaving the transaction clean on the DB (FILE LOCKING also involved)

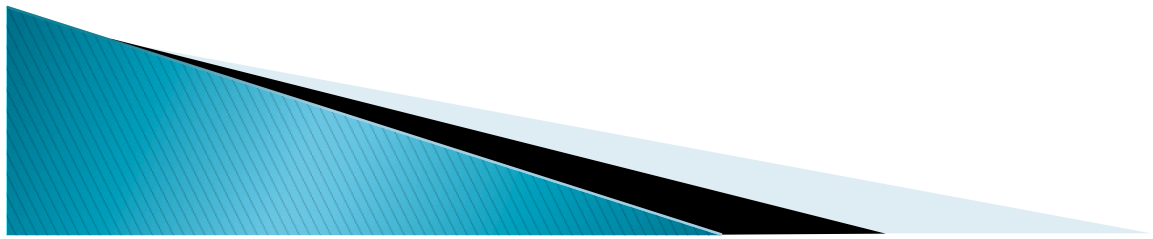
FILE FORMAT/ BATCH

- Events collected on data-tacking from the DAQ and the ones reconstructed in the offline cluster are stored in YBOS format
- Output data of the user analysis are stored directly on the KLOE AFS Cell in user specific format
- User files are not involved in the BOOK-KEEPING process
- Reconstruction and analysis are managed by the central batch scheduler LoadL



Conclusion

- DH is a mix of commercial and custom software
- the dependency on commercial software is minimized by the layers of custom software
- commercial software carries on all the vital functions
- If the commercial software Is changed, only the daemon involved has to be adapted (e.g from TSM -> GridFTP)
- The trivial point: BOOK-KEEPING relies strongly on the system model.



Data-Handling scheme