Royal Holloway
University of London

DAQ update: Data handling application

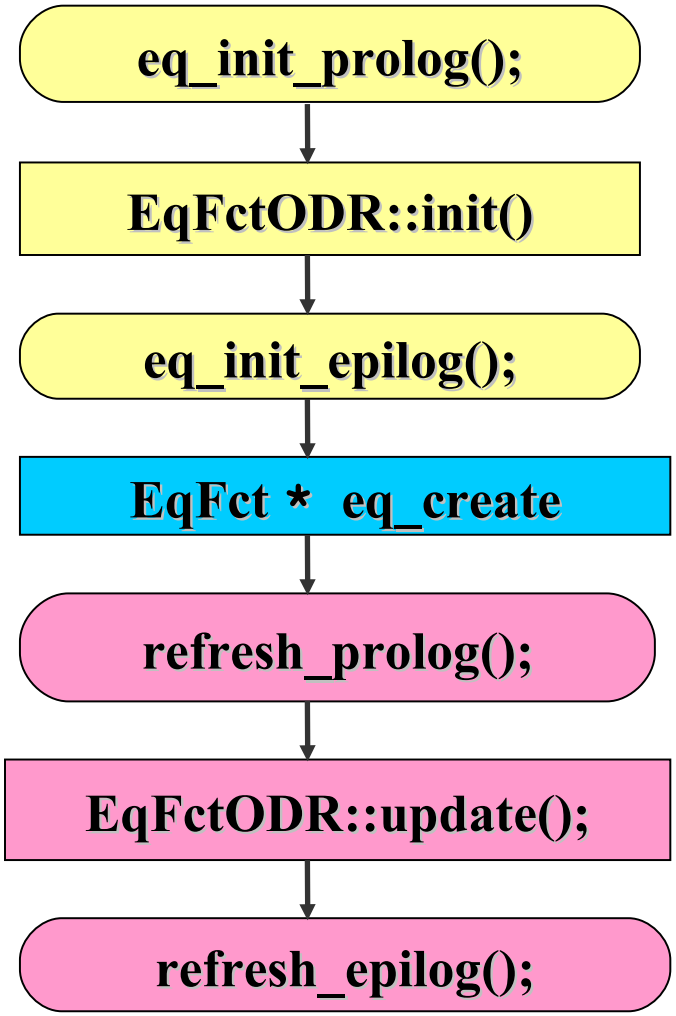
Valeria Bartsch, Tao Wu

18/Feb/2009

Tasks recall

- Being done now
  - Data handling - Configuration/Device/Data DB
  - C&C device server
  - ODR state machine
  - Error/alarm handling
- Not yet, need h/w development
  - LDA device server
  - DIF device server
  - Full state machine

Methods of device server

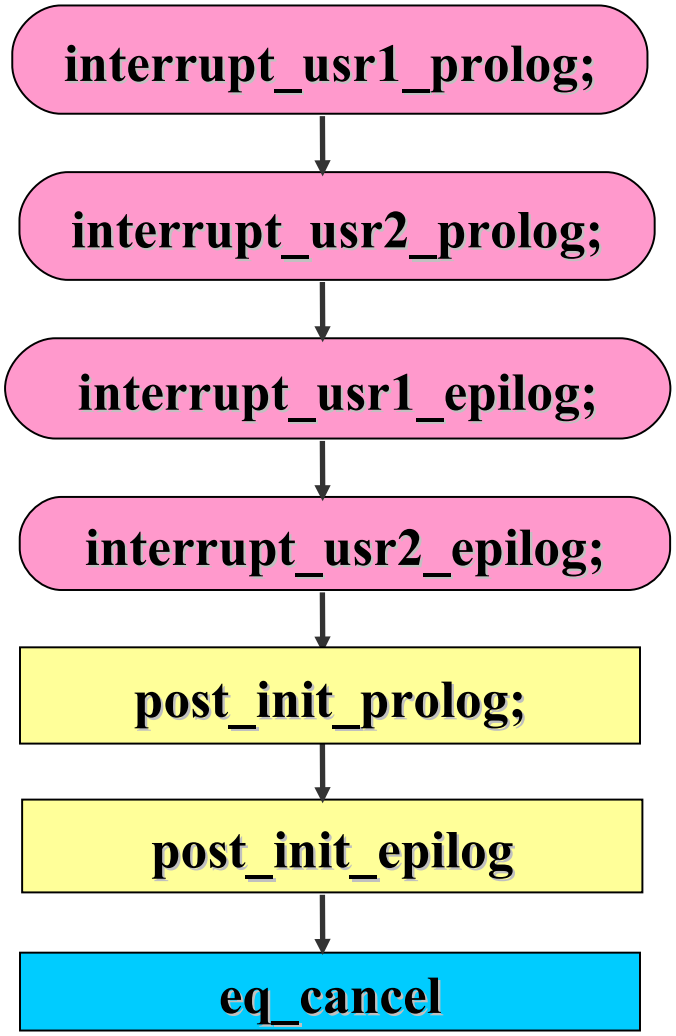


The init() method is call for every location during startup of the server. Initialization of the hardware may be done here

during startup of the server to create the locations, properties loaded.

This update method usually does the real work in a DOOCS server. It runs in a loop over all locations

Methods of device server



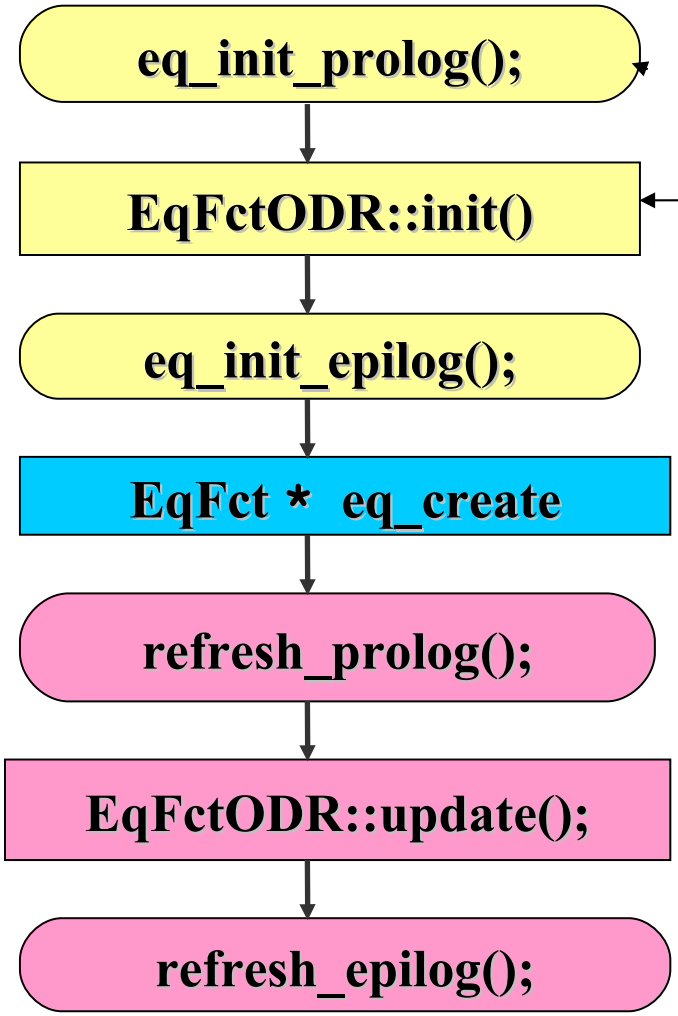
external interrupts functions
of SIGUSR1/SIGUSR2 interrupts
from timing system

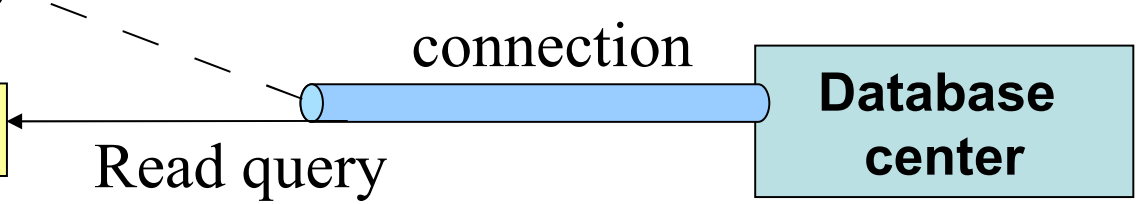
called before the first update of all EqFct

called after the first update of all EqFct

Cancel devices, kill threads

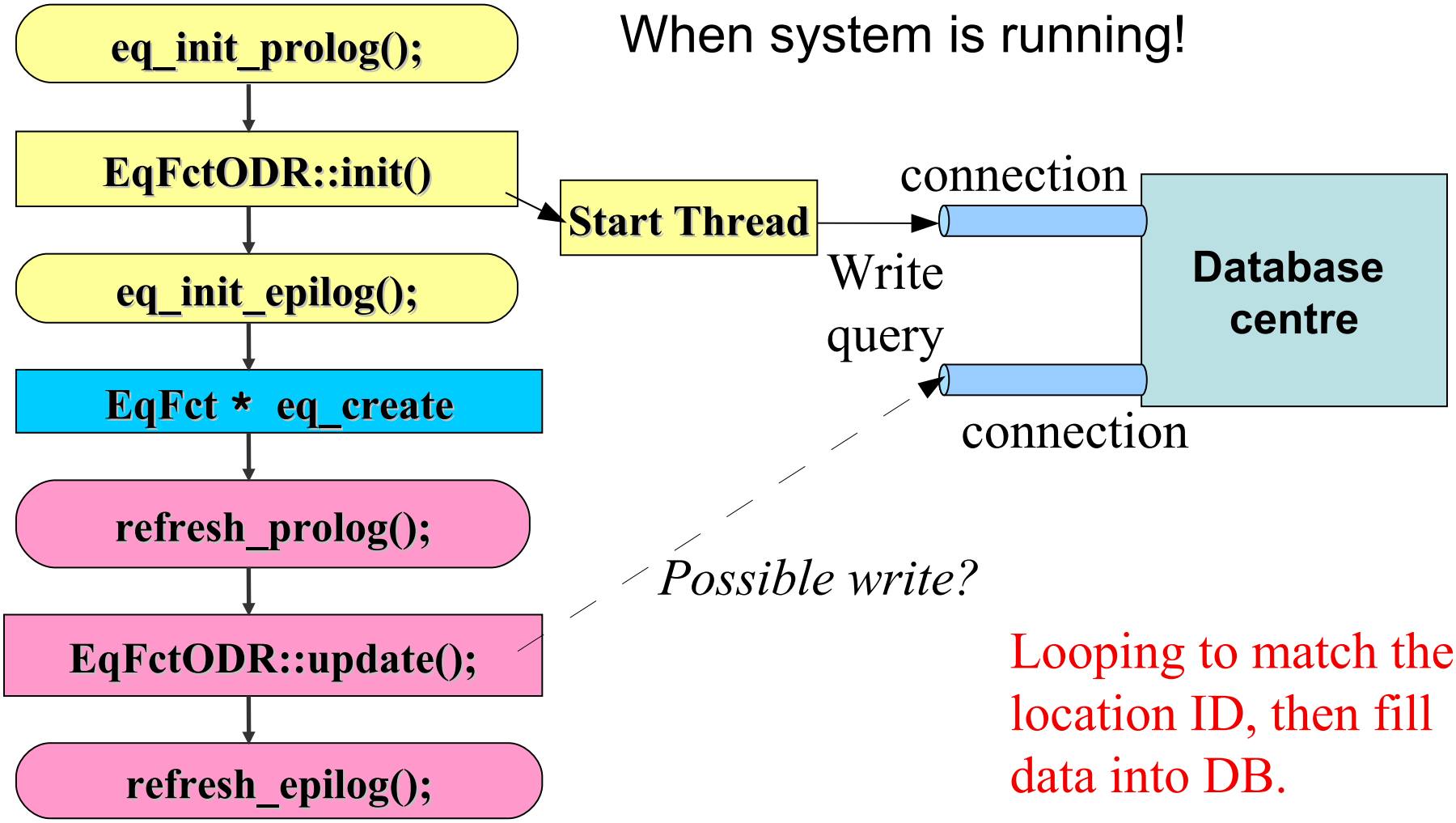
Read configuration info from database



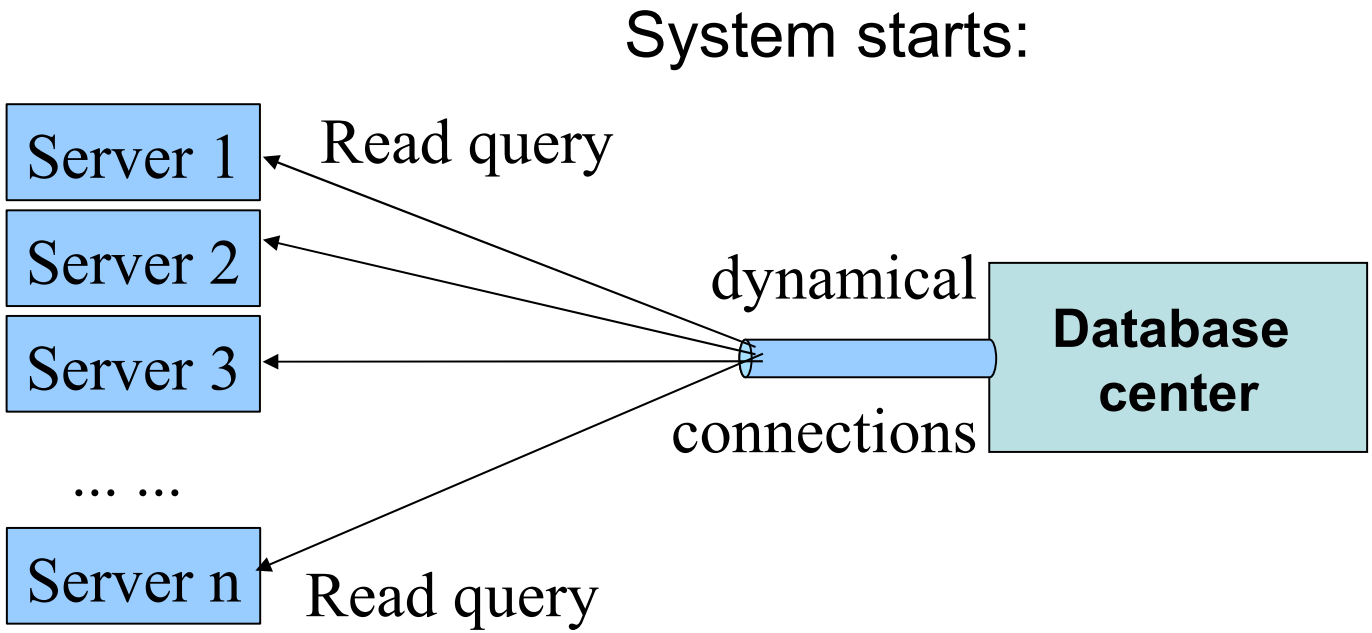


When system is started and devices are initialised! After that, connection is removed.

Write info to database

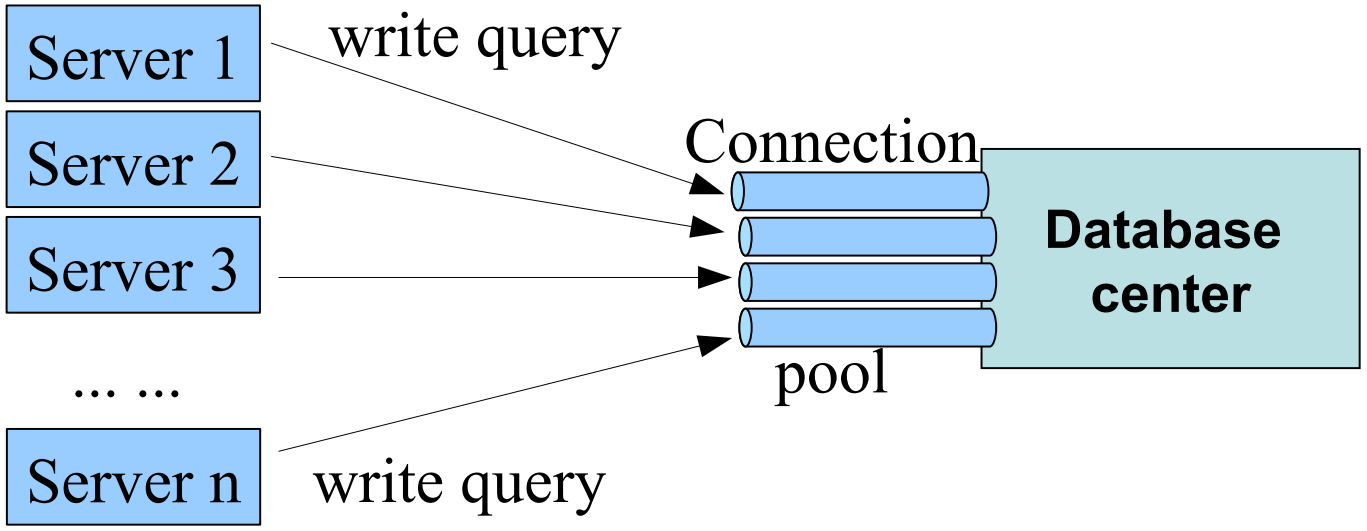


What if many servers synch read?



What if many servers synch write?

FILES (FILE_NAME, LOCATION_ID, CREATION_TIME, CREATE_USER, CREATE_DATE)



Concurrently writing when system is running!!!

Database issue & its performance

- The number of connections to DB will affect the MySQL database server performance;
- Unlikely have write conflicts, because of LOCATION_ID matching. Every ID is unique.
- Strategy: 2 thread per connection per d/s?
  - Two threads: read & write;
  - Read thread is dynamical for initialization,
  - write thread is always running;
  - Per connection per device server for ODR;
- For test system, it's OK. But for beam test DAQ, it needs to be carefully designed.

InnoDB Data storage engine

- Transactions enabled which is safer for multi-threads concurrently writing;
  - transaction log : recovery of data always possible
  - accept several command with one “commit” statement;
  - rollback without saving in case of problems
  - if write fails, all changes are reverted;
  - provides better concurrency (for synch writing)
  - can be backed up while running;
- InnoDB storage engine gives safer mechanism.

Summary

- Be aware of synch writing through connections;
- Performance of database could be decreased by large number of connections;
- Device servers use 2 threads per connection;
- Better use InnoDB storage engine.