C ontrol system based on a
H ighly
A bstracted and
O pen
S tructure

WP3 STATUS 12/02/2014

A.Michelotti

Status

- We have all the HW for ESCO use case
- Defined BTF activities for next weeks
- Spent most of this month to develop a building and test framework, that now is ready and running
- Written some basic but useful “localhost” tests for no-regression

TO DO

- WP3/WP4, given the capabilities of the sensors and controllers define and deploy a real layout for Touschek
- CHAOS DAQ for BTF with feedback on magnets
- Improve test and test strategy to cover:

1. Localhost environment
2. Distributed environmnent
3. Heterogeneous environnment

!CHAOS infrastructure is very complex

During the development we faced with multiple issues, due to the the complexity of the project and to the fewer number of people that are involved in the development that is still constant ~1 + ½.

Instability sources:

1. External library dependency
2. Heterogeneity of supported architectures (MAC,Linux (x86),Linux (armhf), Virtualized architectures.
3. Slightly different behavior of compilers and libraries among different architectures
4. Asynchronous nature of the framework that involves an high number of concurrent threads (a basic CU more than 30 threads), difficult to debug and detect “timing issues”
5. Distributed nature of the project that involve a network of connected nodes. It brings other issues due mainly to Network uncertainty (delays, node unavailability…)
6. The amount of the code and files increase day by day thanks to the great productivity of Claudio
7. Miss of an automatic and systematic build and non regression framework able to early detect issues during development

!CHAOS infrastructure is very complex

The solution to cope wit the !CHAOS complexity and the miss of developers is to invest in good top-down tests and have a good automatic non regression test framework.
The non regression exits with YES/NO and a report of resource utilization

Adopted top-down test strategy:

1. Tests of increasing complexity (from localhost, to distributed, to Heterogeneous architectures)
2. Tests that test common and public “API” (i.e. multiple init,start,stop,deinit,get dataset, get data)
3. Tests that involve a random and “high” number of configurations of CUs and UIs
4. Tests on Virtual devices that mime the class of device (i.e powersupply) supported at the moment
5. Performance Infrastructure Tests
6. ……..

thanks you