

Summer Student Project

An alternative approach to configure permanent tasks in LHCb Online farm nodes

Krzysztof Wilczyński

Supervisors: Markus Frank, Beat Jost

August 28, 2018



About me



Introduction About me

The system

Farm controller Boot script

The motivation

Upgrades API Development

The results

Results Tools

■ I am... an engineer!

- Master student of Automatic Control and Robotics (spec. Robotics)
- Faculty of Power and Aeronautical Engineering,
 Warsaw University of Technology
- I used to work at...
 - Bosch Rexroth
 - Airbus Military Defence and Space
 - Student Association for Vehicle Aerodynamics
 - Polish National Opera







Online farm computing nodes



Introduction About me

The system

Online Farm

Farm controller Boot script

The motivation

Upgrades API

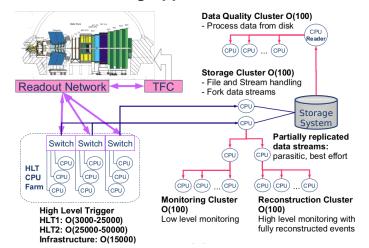
Development

The results

Results Tools



Data Processing Apps > 80.000 Instances





LHCb Online farm process controller



Introduction
About me

The system

Online Farm

Farm controller

Boot script

The motivation

Upgrades API

The results

Results

The handling of the permanent processes on the data processing nodes is based on sending commands to pcSrv process running on each of the corresponding "Controls PCs".

LHCb Online Farm Process Controler on Researchgate





Farm boot script



Introduction About me

The system

Farm controller

Boot script

The motivation Upgrades

API Development

The results

Results Tools In the current solution, all processes (scripts) started on the farm nodes are grouped in a single, huge python script that prints out ready to execute pcAdd commands for a given node name.



A command used to start a task on node(s):

pcAdd(regex, start parameters, script, script parameters)





Farm boot script



Introduction About me

The system

Farm controller

Boot script

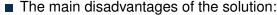
The motivation Upgrades

API Developme

Development

The results

Results Tools



- Modifications of task parameters are difficult
- It is easy to make a modification that harms dependencies in the task sets (no error prevention mechanism)
- Only a specialist who knows the boot script structure can use it (no high-level interface)
- There is no easy way of knowing which tasks are running on given node (one has to analyze the boot script line by line)

The boot script has been created as a "quick hack" about 10 years ago.

The time has come to upgrade it!





My project



Introduction About me

The system Online Farm

Farm controller Boot script

The motivation

Upgrades
API
Development

_ -----

The results

Results Tools The solution: create a system for the process controller infrastructure utilizing a database driven approach. The main goals were to:

- simplify the modifications of hierarchical structure of tasks running on the nodes
- prevent human errors breaking the system integrity
- create one source of information regarding processes running on a given node
- create a reliable and future-proof API for future developments





What do I mean by Main API?



Introduction About me

The system

Online Farm Farm controller Boot script

The motivation

Upgrades

Development

The results



API: Application Programming Interface

The created API is a Python class containing methods (add, delete, modify, get, assign, inSet) that allow safe access to the underlying database. It is a high-level connector providing an easy integration of different client applications.





Stages of development



Introduction About me

The system Online Farm

Farm controller Boot script

The motivation Upgrades

API Development

The results

Back-end

- Database schema architecture
- Main database API
- New boot script
- Unit testing script (internal error prevention)
- Frontend connectors: JSONRPC, (REST, XMLRPC)

Front-end

- Command line user interface
- Graphical user interface (web application)





Current results



Introduction About me

The system

Farm controller

The motivation

Upgrades
API
Development

The results

Results

i -			LHCb Online Far	LHCb Online Farm Process Explorer				
Navigation III Show at	Operation						O Crease O Delete	
Tasks Task Sets Node Classes Nodes			Unique name	Script name	Script parameters	Process identifier	 PcAdd parameters 	
	• 🗆		LogDefaultSrv	AagSener.sh	-no-authentication -K1	LogDefaultSrv	12 Sort Ascending 1: p /	
LogDef	0		LogSaudSrv	/LogServer.sh	-no-authentication -K1	LogGaudStv	It Sort Descending p.N.	
ROpublish	8	3	LogLHCb1Srv	Anglerversh	-no-authentication -K1	LogLHOs1Sev	☐ Columns > -p/r.	
TaskSupenisor		-4	LogLHCb2Srv	AugServer.sh	-no-authentication -K1	LogLHCb2Srv	-5 100 -11 -s hcts2 -p /r.	
witOID	0	- 5	LogLHCbASrv	/LogServer.sh	-no-authentication -K1	LogLHCsASrv	-5 100 -l 1 -s Incha -p /r.	
PropertyServer	8	6	TANSOV	/TanServ.sh	-no-authentication -K1	TANServ		
LogGaudGumSrv	8		pingSn_u	Minglevah	-no-authentication -K1	pingSrv_u		
ogGaudi	8	- 4	ROcollect	/ROMONCollect.sh	-no-authentication -K1	AGcollect		
ogLHCb15umSrv	88	- 9	SSSRecover	/home/best/ssscheck.sh	-no-authentication -K1	SSSRecover		
ogLHCb1		10	TorrereLoader	/BitTorrentLoader.sh	-no-authentication -K1	TorrentLoader		
ogLHCb2SumSrv			LogiumDelire	rLogServer.sh	-no-authentication -K1	LogiumDeliro	-13-5 100 -s fmc -p /rur	
ogLHCb2		12	LogDef	/LagCallect.sh	-no-authentication -K1	LogDef	43 s fre o handred.	
ogLHCb4SumSrv	8	13	ROpublish	/MONPublishish	-no-authentication -K1	Allpublish	-statDelay=8000 -mbm.	
ogLHCbA		14	TaskSupervisor	/TasiSupervisor.sh	-no-authentication -K1	TaskSupervisor		
ogPVSSSn.J-LT		15	witCID	/webDtD.sh	-no-authentication -K1	witOD		
ogPVSS_HLT		16	PropertyServer	/PropertyServer.sh	-no-authentication -K1	PropertyServer		
LTEQBridge	0		LogGaudSumSrv	AngServer.sh	-no-authentication -K1	LogGaud/SumSrv	-11-5100-s grudi-p /r	
s.ThrCol	٠ 🗆	18	LogSaudi	/LogCollect.sh	-no-authentication -K1	LogGaudi	-11 is gaudi -o hunffino	
	0	73	LogLHCb1SumSrv	rLogServer.sh	-no-authentication -K1	LogLHCls1SumSrv	41 -5100 -s hcb1 -p /r.	
Rigpublish: This is an incredibly important tasklit is also reportant to keep this important task as important as it is.		20	LogLHCb1	/LogCollect.sh	-no-authentication -K1	LogLHCb1	41 a hob1 a hundino	
			LogLHCb2SumSrv	/LogServer.sh	-no-authentication -K1	LogLHCs2SumSrv	41-5100-s hct2-p /r.	
	0	22	LogLHCb2	/LogCollect.sh	-no authentication -K1	LogLHCb2	-11 -s hct/2 -o /run/finc	

Open-source code repositories:

- Bitbucket Repository (K. Wilczynski)
- Will be moved to LHCb Gitlab Repository (M. Frank)





Tools used in the project



Introduction About me

The system

Farm controller Boot script

The motivation

Upgrades API

Development

The results

Tools

- Back-end programming language: Python 2.7.15rc1
- Front-end programming language: JavaScript
- Database engine: SQLite (+ python sqlite3, sqlalchemy)
- Front-end connector protocol: JSONRPC
- Front-end framework: Sencha Ext JS ver. 6.2.0
- Git version control: Bitbucket + GitKraken



Thank You,





Task grouping hierarchy



Introduction

About me

The system

Online Farm Farm controller Boot script

The motivation

Upgrades API

Development

The results Results

Tools

