



Contribution ID: 463

Type: oral presentation

Replacing the Engines without Stopping The Train; How A Production Data Handling System was Re-engineered and Replaced without anyone Noticing.

Thursday, April 16, 2015 12:15 PM (15 minutes)

As high energy physics experiments have grown, their operational needs and requirements they place on computing systems change. These changes often require new technical solutions to meet the increased demands and functionalities of the science. How do you affect sweeping change to core infrastructure, without causing major interruptions to the scientific programs?

This paper explores the operational challenges, procedures and techniques that were used to completely replace the core data handling infrastructure for the Fermilab experimental program, while continuing to store and deliver more than 1PB of data per month to the analysis and computing efforts of the experiments. It discusses designs patterns like the Command Pattern, the Façade and Delegation that were employed at different stages of the project and how they worked and didn't work to hide the underlying changes that were being made. We discuss how parallel production and parasitic integration systems were used to perform "at scale" performance testing and data validation prior to switch overs and how this allowed for more robust test environments which would not have been possible to replicate in traditional test settings. We describe how the experimenters were engaged in the transition process and how they were used to propagate subtle but necessary changes to the client interfaces over a period of 18 months.

Author: MENGEL, Marc (Fermilab)

Co-authors: Dr LYON, Adam (Fermilab); Dr NORMAN, Andrew (Fermilab); Dr DIESBURG, Michael (F); GHEITH, Michael (Fermilab); Dr ILLINGWORTH, Robert (Fermilab); WHITE, Steve (Fermilab)

Presenter: Dr NORMAN, Andrew (Fermilab)

Session Classification: Track 4 Session

Track Classification: Track4: Middleware, software development and tools, experiment frameworks, tools for distributed computing