Using Akka Platform in Unidentified Falling Object Detection on the LHC

An Example Use Case Scenario

Evangelos Motesnitsalis
Magnet Powering Interlocks & Software
Machine Protection and Electrical Integrity Group
Technology Department
European Organization for Nuclear Research (CERN)
August 2013

Contents

Section I: Theoretical Aspects

- UFOs
- The LHC Logging Database
- Beam Loss Monitors
- Akka and Actors

Section II: Use Case Scenario

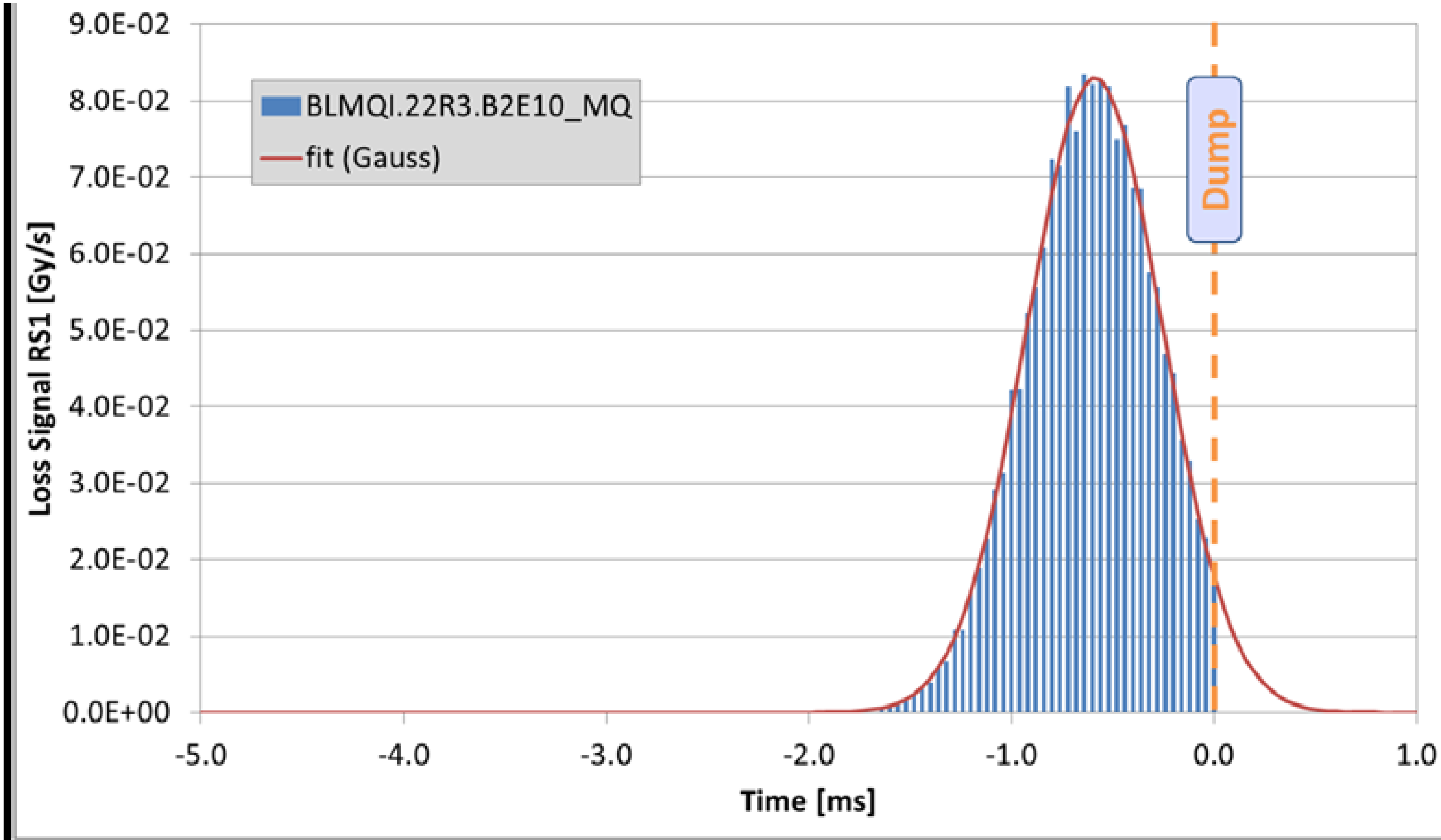
- Motivation
- Finding UFOs with Akka

Section I:
Theoretical Aspects

Unidentified Falling Objects

- UFOs are micrometer sized macroparticles which interact with the proton beam multiple times per event.
- Their creation mechanism is still uncertain.
- UFOs were first observed in July 2010.
- They have caused numerous protection beam dumps and they are one of the major known limitations of LHC operation.
- UFO events at high energies are much more dangerous than those of the lower energies.

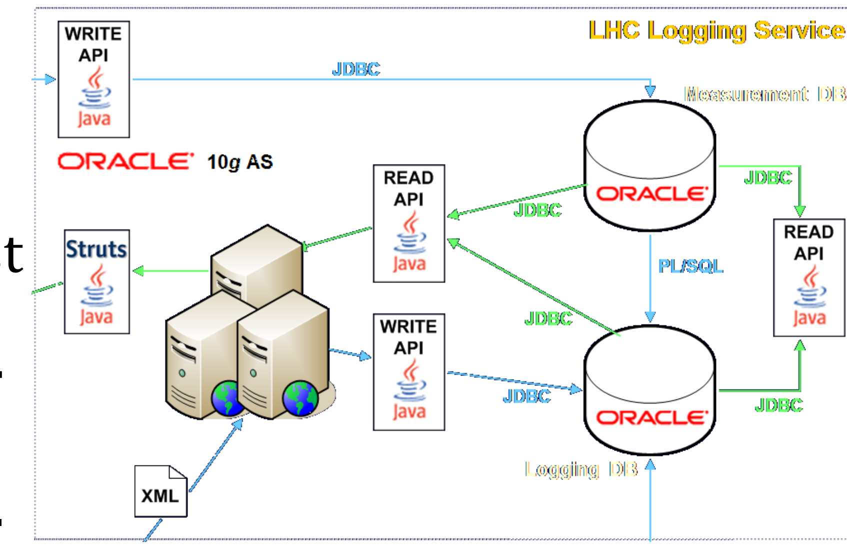
UFO-caused Beam Loss



Plot 1: Temporal loss profile of a fast beam loss event on 23.08.2010. The beam loss in running sum exceeded the corresponding threshold and caused the beam dump. The loss occurred on beam 2 in the arc of sector 34. The temporal width of the Gaussian fit is 335 μs (=approximately 3.8 turns). More than 10.000 UFO events were recorded between April and August 2011.

The LHC Logging Database

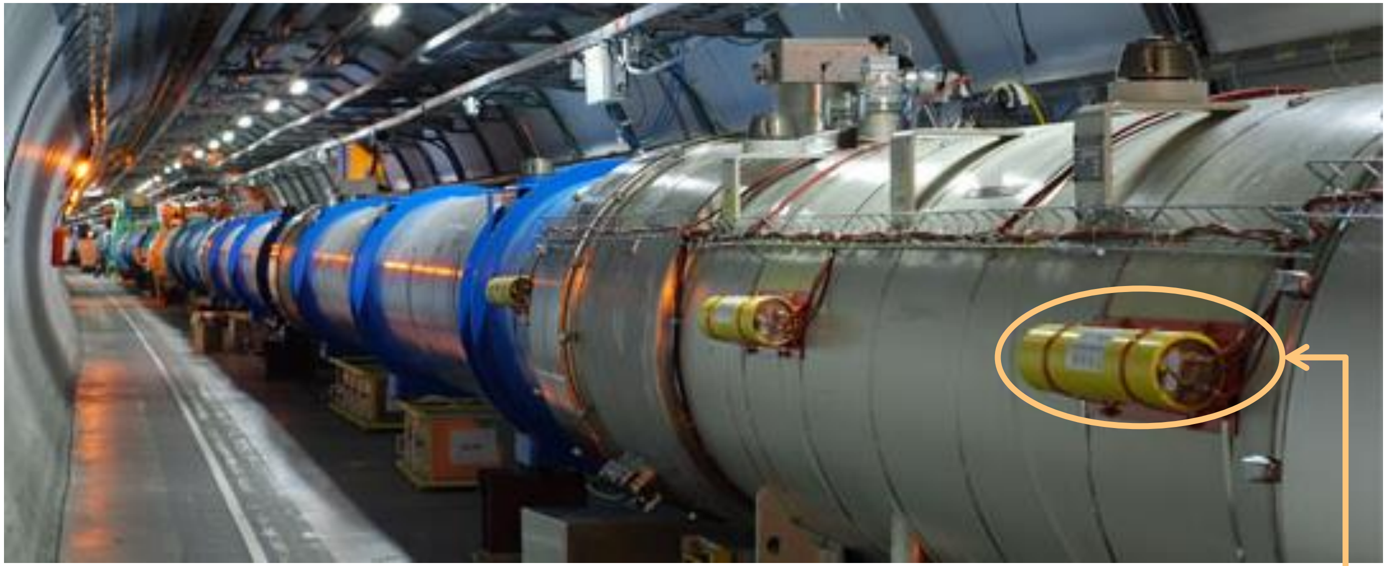
- Oracle is the standard Relational Database Management System at CERN.
- The database stores almost 5 TB of operational (and not physics) data per year and deals with more than 40 million queries per day.



- A Java API is used to extract data from the database.
- The Logging Database contains relatively slow time series data that are stored online for at least the lifetime of the LHC.
- Data are considered useful for long term use and are often filtered with respect to the raw measured data.

Beam Loss Monitors

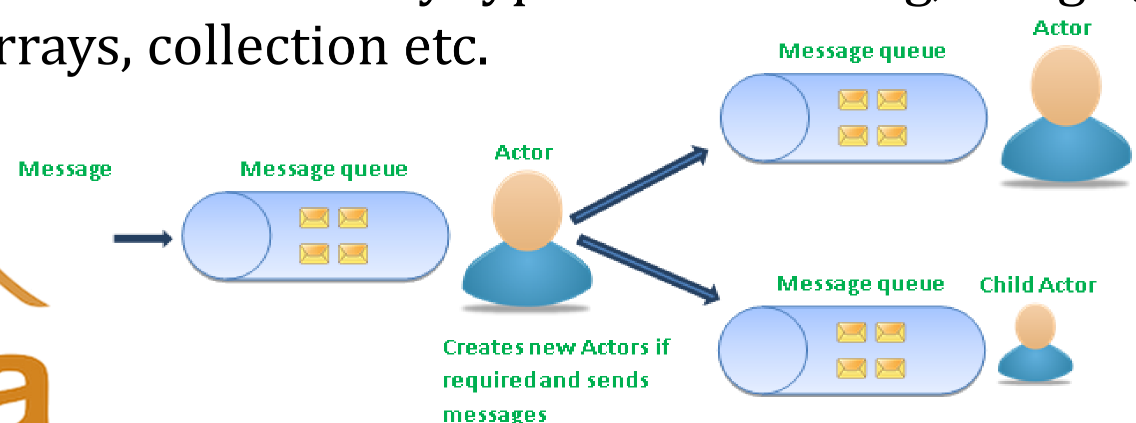
- Beam Loss Monitors (BLMs) are systems that can measure beam losses in the LHC Ring.
- Their data are constantly stored in the LHC Logging Database.
- There are more than 4000 BLMs in the LHC Ring.



An ionization chamber, a specific type of Beam Loss Monitor in the LHC Ring.

Akka and Actors

- Akka is a toolkit and runtime for building distributed, concurrent, scalable and fault tolerant applications on the Java Virtual Machine.
- Actors are very lightweight concurrent object-oriented entities which process messages asynchronously using an event-driven receive loop.
- Actors are constrained only by memory of which they consume a few hundred bytes each. Therefore, any application can easily create millions of Actors which form a hierarchical tree.
- Messages can be of arbitrary type such as String, Integer, Boolean, arrays, collection etc.



Contents

Section I: Theoretical Aspects

- UFOs
- The LHC Logging Database
- Beam Loss Monitors
- Akka and Actors

Section II: Use Case Scenario

- Motivation
- Finding UFOs with Akka

Section II:
Use Case Scenario

Motivation

Why are we interested in new technologies for the analysis of LHC operational data?

Answer: Efficiency

A distributed solution may offer:

- Increased scalability
- Increased reliability
- Increased fault-tolerance
- Increased decentralization
- Reduced execution time

Therefore, if the Akka approach offers any of these, there are some good reasons to use it!

Finding UFOs – Simple Solution

UFOSearch

1. Fetch Data from the Logging Database;
2. Import the positions of the BLMs;
3. Import the needed Running Sums and their Thresholds;
4. Define the Time Range for the Search;
5. Search for UFOS:
   For every second
   5.1. Fetch the data for this second;
   5.2. If all the Thresholds in the Running Sums are reached;
   5.3. Add the UFO to a List;
6. Print the List to the Logging Output;

! But where is the point where we can inject parallelization? !

Finding UFOs with Akka

UFOSearch

1. Fetch Data from the Logging Database;
2. Import the positions of the BLMs;
3. Import the needed Running Sums and their Thresholds;
4. Define the Time Range for the Search;
5. Search for UFOS:

   For every second

   create an actor who will:

   Actor. 5.1. Fetch the data for its second;

   Actor. 5.2. If all the Thresholds in the Running Sums are passed;

   Actor. 5.3. Add the UFO to a List;

   Print the List to the Logging Service;

! We create as many actors as the seconds that are contained in our time range but this does not affect us as an actor needs only around 300 bytes of memory to be saved !

Summary

To sum up…

- We continuously try to improve the frameworks and software running on (and for) the LHC.
- A good approach is parallelization –which is a big trend in Computer Science in any case.
- The UFO Detection Algorithm is only a try of the Akka Toolkit which will enable us to decide whether or not we will make an extensive use of it to parallelize even more parts of the LHC Operational Data Analysis in order to increase scalability and reduce execution time.

Thank you!

e-mail:
evangelos.motesnitsalis@cern.ch
mevangem@csd.auth.gr