DAQExpert

An expert system to increase CMS data-taking efficiency

ACAT 2017 Aug 21-25
18th International Workshop on Advanced Computing and Analysis Techniques in Physics Research

Track: 1. Computing Technology for Physics Research
Topic: Online computing
Subtopic: Advanced Monitoring, Diagnostics and Control
CMS - Compact Muon Solenoid

1. data-taking efficiency
2. operator crew challenges
3. automation in operations

Basic information

- general-purpose detector @ LHC, CERN
- Higgs Boson discovery (as 1 of the 2 experiments)
- 21x15x15 m, 12 500 tonnes
DAQ - Data Acquisition 1

- read out the data
- branch crossing 40 MHz rate
- event size 1-2MB
- 2-level triggering
- hardware trigger selects 100 kHz
- full events built at 200GB/s
- 26 000 cores in HLT farm select $O(1 \text{ kHz})$
DAQ - Data Acquisition 2

- human operators 24/7
- sometimes dataflow is stuck
  - recovery procedures
  - challenging during time-critical operations
- human on-call experts 24/7
- 5% downtime
- optimize this part of the operations
Motivation for new tool

objectives
- reactive recovery from data-taking upsets
- monitoring data-taking health

areas to improve
- reaction time
- suboptimal recovery actions
- external help demand
Evolution of expert tools in DAQ

DAQDoctor
- DAQ-1
- Run 1
- Perl
- Single person

Esper research project
- Long Shutdown 1
- Declarative approach
  - Group not used to this paradigm

DAQExpert
- DAQ-2
- Based purely on Java
  - Language widely known in the group
- Modular
Environment

- operator has final decision
- advise the operator
- human factor

And online software
Solution: Identify the problem → Find optimal recovery → Advise operators

Monitoring services

SnapshotService
Retrieve → Map to model → Persist

ReasoningService
Analyse → Persist

NotificationService
Dispatch → Persist → Deliver

Clients
- Control room
- Human experts
Reasoning

- Expert-knowledge representation
- Logic Module as building block
- Each LM defines some condition
- Condition in `satisfy` method, returns true if satisfied, false otherwise
- Reusing the results
public class PartitionProblem extends KnownFailure {

    public PartitionProblem() {
        this.name = "Partition problem";
        this.description = "Partition {{P}} in subsystem {{S}} is blocking triggers";
        this.action = new SimpleAction("Issue a TTCHardReset",
            "Problem not fixed: Stop current run & start a new run.",
            "Problem still not fixed: Call the DOC of {{S}}",
            "Problem fixed: Make an e-log entry.");
    }

    @Override
    public boolean satisfied(DAQ daq, Map<String, Boolean> results) {
        boolean result = false;
        if (!results.get(DataFlowStuck.class.getSimpleName()))
            return false;

        for (SubSystem subSystem : daq.getSubSystems()) {
            for (TTCPartition ttcp : subSystem.getTtcPartitions()) {
                TTSState currentState = getPartitionState(ttcp);
                if (currentState == TTSState.OUT_OF_SYNC) {
                    context.register("S", subSystem.getName());
                    context.register("P", ttcp.getName());
                    context.register("C", currentState.name());
                    result = true;
                }
            }
        }
        return result;
    }
}
Reasoning III

- Hierarchy of LM represents the knowledge
- Key parameters are monitored
- LMs cooperate
- Chain of LMs is activated to reach final conclusion
- Currently
  - 20+ LMs in the system
  - 78% coverage
- LMs ran every iteration of analysis

Diagram:
- FED rate
  - No rate
    - Dataflow stuck
      - FED stuck
  - Expected rate
    - Run ongoing
      - TCDS state
- Level Zero state
Framework

- Framework delivers LM results
- Real time notifications (1)
- Entries on the timeline (2)
Out of sequence data received

Run blocked by out-of-sync data from FED 1386 received by RU ru-c2e12-40-01.cms - now in syncloss state. Problem FED belongs to partition MUTFUP in TRG subsystem. This causes backpressure at FED 1380 in partition MUTFUP of TRG

Try to recover (try up to 2 times)

Stop the run. Red & green recycle the subsystem. Start a new Run

Problem not fixed: Call the DOC of TRG (subsystem that caused the SyncLoss)

Problem fixed: Make an e-log entry. Call the DOC TRG (subsystem that caused the SyncLoss) to inform about the problem

Main view for CR RT suggestions to shifters:

- Reduce reaction time
- Avoid wrong decisions

Suggestion format:

what's the problem + what's the best action to take
- **Goal**: post-mortem analysis
- Visualizes analysis in time
- **Analysis panel**
  - 1 Row - 1 Logic Module
- **Raw data panel**
  - Parameters from snapshots
- Freely move and zoom in time
<table>
<thead>
<tr>
<th>Date</th>
<th>Event Type</th>
<th>Sender</th>
<th>Title</th>
<th>Displayed message</th>
<th>Source LM</th>
<th>Audible</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017-08-16</td>
<td>Start</td>
<td>Expert</td>
<td>Started: FED stuck</td>
<td>TTCP TOB of TRACKER subsystem is blocking trigger, it’s in WARNING TTS state, The problem is caused by FED 434 in WARNING</td>
<td>Fed stuck</td>
<td>Default</td>
</tr>
<tr>
<td>2017-08-16</td>
<td>Start</td>
<td>Expert</td>
<td>Started: Partition problem</td>
<td>Partition EB- in ECAL subsystem is in ERROR TTS state. It’s blocking trigger.</td>
<td>Partition problem</td>
<td>Default</td>
</tr>
<tr>
<td>2017-08-15</td>
<td>Start</td>
<td>Expert</td>
<td>Started: FED stuck</td>
<td>TTCP TIBTID of TRACKER subsystem is blocking trigger, it’s in WARNING TTS state, The problem is caused by FED [163,143] in WARNING</td>
<td>Fed stuck</td>
<td>Default</td>
</tr>
<tr>
<td>2017-08-15</td>
<td>Start</td>
<td>Expert</td>
<td>Started: FED stuck</td>
<td>TTCP TOB of TRACKER subsystem is blocking trigger, it’s in WARNING TTS state, The problem is caused by FED 434 in WARNING</td>
<td>Fed stuck</td>
<td>Default</td>
</tr>
<tr>
<td>2017-08-15</td>
<td>Start</td>
<td>Expert</td>
<td>Started: FED stuck</td>
<td>TTCP TOB of TRACKER subsystem is blocking trigger, it’s in WARNING TTS state, The problem is caused by FED 79 in WARNING</td>
<td>Fed stuck</td>
<td>Default</td>
</tr>
</tbody>
</table>

- **Goal:** browse all generated notifications
- **Filter by key fields**
- **Inspect link**
Technologies used

- **Presentation layer**
  - Web application
  - Javascript, VisJS
  - Bootstrap

- **Backend**
  - Micro service architecture
  - RESTful services
  - Hibernate
  - Oracle
Results

- Efficiency of CMS
  - Many other factors interfering
- User feedback
  - Positive but not quantitative
- Recovery selection correctness
Key parameters

Problem occurs
Data-flow stuck

Operator reacts
Problem identified, recovery issued

Recovery completes
Either successful or not

Reaction time
67s

Recovery time
317s

Intervention time
6m24s

Avg on 2016
Recovery selection correctness

- Comparison based on
  - 3 months from 2016 and 2 months from 2017
  - During stable beams only
- In 2016 operators had no guidance
  - 36 data-taking failures
  - 32 optimal decisions
- In 2017 there was guidance
  - 29 data-taking failures
  - 29 optimal decisions
- Clear increase of fraction of optimal recovery 88% → 100%
Future

- Bypass the operator - automatic recovery wherever possible
- Extend coverage
- Predicting failures
Using framework elsewhere

- Define your monitoring model (1)
- Define your knowledge as LMs (2)
Summary

1. **New expert tool**
   Pure imperative language (java) to implement reasoning

2. **Successful at CMS DAQ**
   Recovery selection correctness:
   88% → 100%

3. **Generic solution**
   Can be used elsewhere

Maciej Gladki
maciej.szymon.gladki@cern.ch