Strategies for distributed data acquisition, reconstruction and analysis in ALICE

Matthias Richter

NorCC Strategy discussion Kick-off meeting
Mar 11 2022
ALICE in 2022

- Just finished LS2 upgrade and being in the final commissioning phase
- Discussion started in 2011, Upgrade LoI published in 2014
- Beside the challenging detector developments, the increased data rate also required a new computing concept
- ALICE could build upon the experience from the ALICE High Level Trigger, an online system exploiting parallel, distributed data processing and hardware acceleration on FPGA and GPU
- It was decided to build a common online-offline compute facility ALICE O² with a common concept of distributed computing for data acquisition, simulation, reconstruction, and analysis

10-years-period of design, development, construction, and commissioning ⇒ that’s the time scale.
ALICE in Run 3: 50 kHz Pb-Pb

- Record large minimum bias sample.
- All collisions stored for main detectors → no trigger.
- Continuous readout → data in drift detectors overlap
- 50x more events stored, 50x more data.
- Cannot store all raw data → online compression.
→ Use GPUs to speed up online processing.

Basic processing unit of ALICE: Time Frames
- ~10 ms of data
- Contains O(500) collisions
ALICE online-offline - ALICE O²

- ALICE had to do (and is doing) a major effort in LS2 to reduce the gap between required and affordable computing resources

- **Conceptual paradigm shift:** quasi-online processing

- **Algorithmic paradigm shift:** focus on algorithms for synchronous reconstruction

- **Triggerless** acquisition

- Massive utilization of **hardware accelerators**

⇒ Complete system designed for high data throughput
Hardware acceleration in ALICE O²

- The Time Projection Chamber is one thing making ALICE special
- Low mass detector
- Particle tracking in high occupancy environment
- Data from many collisions overlapping in the acquisition window
- > 3 TB/s full reconstruction on GPU

Installed ALICE EPN farm for Run 3:
- 250 servers with 8 AMD MI50 GPUs
- total 2000 GPUs
Workflow-oriented definition of the compute topology

On top of FairMQ as transport layer and the $O^2$ data model as data layer, a third software layer, the **Data Processing Layer (DPL)** was introduced.

- The basic building blocks of DPL workflows are DataProcessors defined as entities with **inputs**, an **algorithm**, and **outputs**
- Workflows combine/chain individual DataProcessors
- Multiple workflows can be combined into one workflow

The description is **declarative**: The user describes *what* to achieve in terms of process connectivity and algorithm, the framework takes care of *how* to realize the workflow and the connections.
Analysis Framework

The increase in data and event rate also imposed challenges to the analysis → big need for increasing and improving throughput, efficiency, and organization

- New, dedicated analysis computing model following the common distribution model
- Analysis framework built on top of ALICE O² Data Processing Layer
- Columnar in-memory representation
- Organized in workflows: modular, mergable entities
- Declarative definition of workflows
- Analysis framework applies automatic optimization based on the information from declaration of analysis

⇒ Lots of new concepts emerging and exploited,
⇒ It’s all about understanding the data model
Analysis Data Model

Interconnected tables
- Self-contained (Tables), as collections of Columns, connected by indices passed through shared memory
- Represented as ROOT TTree [5] on disk and as Apache Arrow Table [6] in memory
- Hierarchy of indices represents logical connections among data Tables (Tracks → Collisions → BCs)
- Columns and Tables are represented by C++ types for the end user resulting in negligible performance overhead

Columns

<table>
<thead>
<tr>
<th>X</th>
<th>α</th>
<th>f(X, Z, m)</th>
<th>Index</th>
<th>Z = X sin α</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Static
- Arrow::Array
- lambda function

Dynamic
- Arrow::Array

Index
- Arrow::Array

Expression
- Arrow::Array
  - created in memory
  - with Gandiva[4]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
Table Manipulation

Database-like operations

- **Join**

- **Filter/Partition**

- **Grouping**

- **Combinations**

All operations are zero-copy due to Apache Arrow backend.
Analyzer can directly request joined, grouped, partitioned of filtered table as an input to their task, combining all four operations if needed.
It is possible to inspect 2-, 3- and more rows combinations of a particular table without nested loops or memory caches, by using combinations generator.
A traditional “event loop” interface is also provided.
Summary - Distributed computing in ALICE O²

ALICE is now using one unified model for distribution of data and computing tasks within the common online-offline O² system. All components follow the same interface and strategy.

- multi-process, small, configurable entities
- data model to uniquely describe all data in the system
- declarative composition
- fully decoupled algorithms from transport and I/O
- supported plugin of hardware accelerators
- common algorithmic code base for CPU and GPU

Lots of expertise in the fields which will be required for future LHC computing
Strong participation from the Norwegian ALICE community
Summary - Norwegian ALICE computing activities in the coming years

The major LS2 upgrade has just been finished, in the field of computing. Norwegian ALICE community is contributing to:

- Core Data Processing Layer in ALICE O²
- Framework for declarative workflows
- Analysis framework
- JAiEn grid middleware
- Neic Nordic Tier 1 participation

⇒ it’s all application-motivated - “we want to do physics”

Recall: many of the challenges for future LHC computing have been tackled in ALICE already in LS2. We now have expertise, prototypes, and even full-scale production system

⇒ can be applied in the same manner to ALICE 3
Strategy - Where can we have an impact?

Fields where we can significantly contribute to computing challenges as relatively small group:

- First priority: Physics analysis → make analysis easier and more efficient
- Simulation and modeling
- Verification of algorithms, data quality, and performance
- Automated optimization
- GPU expertise → extend to analysis and ML surrogate models

We have the unique chance of connecting simulation and modeling to a vast amount of real data, covering physics, algorithms, operation. Need to continue exploiting this.