An Oracle-based Event Index for ATLAS

Elizabeth Gallas, Gancho Dimitrov, Petya Petrova, Zbigniew Baranowski, Luca Canali, Andrei Dumitru, Andrea Formica

CHEP 2016 Conference
San Francisco, CA, USA
Oct 8-14, 2016
• Introduction(s):
  • The ATLAS experiment at the LHC/CERN
  • ATLAS data processing stages
  • What is the ATLAS EventIndex (EI)?
    • And what is EIO (Event Index Oracle)?

• Requirements & Challenges; Features & Solutions
  • Use cases
  • Schema design
  • Data sources and Data selection
  • Storage and upload optimizations
  • User Interfaces

• Summary and Conclusions
ATLAS studies particle collisions (events) in the heart of its multi-component particle detector. 

- Collision rate: 600M/sec → after Trigger filtering → Recording rate: 1000/sec → Events per year: Billions!
- Events are recorded during ‘Runs’ (1-24 hours) sectioned by Luminosity Blocks ‘LBN’ (minute periods of stable beam luminosity)
- Events are assigned a unique Event Number in the Run

Event data is written to files for offline processing
Files: unit of data management & job-wise processing
A dataset is the set of files with events from one Run
Datasets: unit of task-wise processing & physics analysis

ATLAS data file formats produced in stages of processing:
- RAW: raw output from detector components
- ESD (Event Summary Data): initial reconstruction of raw data processed with clustering, tracking, … algorithms
- AOD (Analysis Object Data): summary of reconstructed event suitable for most analysis (all events)
- dAOD (‘derived’ AOD): datasets formed from AOD customized for the needs of specific physics groups (subset of events)
What is ATLAS EventIndex ? and EIO ?

ATLAS EventIndex (EI) is a catalog of ATLAS events:
- Purpose: To provide various event-wise services
  - Described in CHEP 2015: The ATLAS EventIndex (and next slide)

- Master EI storage repository is Hadoop-based
  CHEP 2016 References (in this conference):
  1. “ATLAS EventIndex General Dataflow and Monitoring Infrastructure”: describes the data collection for EI
  2. “A study of data representations in Hadoop to optimize data storage and search performance of the ATLAS EventIndex”: exploring other data formats

- The EventIndex Oracle-based system (EIO) arose out of exploring various technologies for providing event-wise services
  - EIO Services include all the most common use cases and some other services proved to be easy to implement
Main use case: “Event Lookup”

For any Event: **return references to file(s)** along the processing chain containing the event (called the GUID == the Global Unique Identifier)

- For making event displays of special events
- For selecting a subset of events for special studies or processing

→ How does an index help?

Datasets can have millions of events stored in thousands of files … It saves computing resources knowing which files contain events of interest

Goal: “Event Lookup” in a fraction of a second!

Other considerations and use cases:

**Integrity checks:**
- Identify event duplication w/in datasets and other issues in processing
  - e.g. Duplication can occur at any stage: online or offline
  - → EIO data **integrity is important** for “Lookup” and any services as well as to relay any problems found to ATLAS processing experts

Study “event overlap”: events in common between the Datasets of a Run

- By trigger stream, processing, or filtering
  - → much **easier to do with a database** than with a file system

General goals in developing EI Oracle:

Serve the main use cases with excellent performance
See what else we can do with it!
Quantities of interest to satisfy main use cases:

**Dataset Properties**
- **Project**
  - Year of data taking
  - LHC beam type
- **RUN_NUMBER**
  - Online data taking run identifier
- **Stream**
  - Trigger system determines the output data stream of an event
- **Production Step**
  - Internal ATLAS processing step
- **Data Format**
  - e.g. RAW, ESD, AOD, dAOD
- **AMITag**
  - Version of processing of this set of events

**Event Properties**
- **Event Number**
  - Unique in the dataset
- **LBN : Luminosity Block Number**
  - A segmentation of the Run
- **BCID : LHC bunch crossing ID**
- **GUID type and GUID reference 0**
  - The dataset file being indexed
- **GUID type and GUID reference 1**
  - The file of its upstream parent dataset
- **GUID type and GUID reference 2**
  - The file of its further upstream parent dataset

**Challenges:**
- Sheer number of EVENTS:
  - Billions of events (currently $25.6 \times 10^9$ events)
  - Loading and deletion workflow (1 to $O(10^6)$ events in such transactions)
  - Duplicate event detection and handling
- Auxiliary data for cross checks, browsing, ranking, and reports
Event Metadata (for the main use cases) is structured and the structure is very simple (2 tables)

Datasets table
- P – DATASET_ID
- Dataset properties:
  - Project (year, beam type)
  - RUN_NUMBER
  - Stream
  - Production Step
  - Data Format
  - AMI Tag (version)
  - Other dataset-wise info

Events table
- F,U – DATASET_ID
  - U – EVENT_NUMBER
- Event properties:
  - LBN: Luminosity Block #
  - BCID: LHC bunch crossing ID
  - GUID type 0 and Reference 0
  - GUID type 1 and Reference 1
  - GUID type 2 and Reference 2

Duplicate_Events
- DATASET_ID
- EVENT_NUMBER
- LBN
- BUNCHID
- GUID_0
- GUID_1
- GUID_2

Add a third table for duplicate events to help understand the nature and origin of duplication.

Other immediate relational optimizations:
- Store the three GUID types at the dataset level
  - They vary by dataset, but are common to all events in a dataset
- Store only up to 3 levels of GUIDs per event
  - In principle, can be as many as 4 references per event, but in practice, 3 is sufficient for use cases and sometimes only 2 are needed.
Efficient EVENT storage is achieved using:

Normalization: basic relational model techniques and some extreme database tuning!

1. New data:
   - Import into a staging table (no constraints, no indexes)
   - Oracle scheduler jobs: verify, optimize, then move the events to destination tables EVENTS (and possibly the DUPLICATE_EVENTS table)

2. Further reduce Event table volume/row:
   Use non-standard Oracle datatypes for GUID references:
   - example: “21EC2020-3AEA-4069-A2DD-08002B30309D”
     - 3 GUIDs per event: 108 bytes in CHAR datatype
     - Solution: use Oracle “RAW” data type
   - Oracle functions convert to CHAR as needed
   - Lookup always by Event #, not GUID itself

3. OLTP Compression:
   - Reduces data volume and improves transaction speed
   - Moderates the side effects of transactions (1 to O(10⁶) events)
     - Minimizes undo and redo footprint on the database

4. Partition the Events table to handle removal of obsolete datasets
   - “LIST” partitioned (by Dataset ID):
     delete events → by partition removal!
In pre- and post-upload processing, we

- exploit information in other ATLAS metadata catalogs
  - AMI: Dataset-level metadata (CHEP 2016)
  - COMA: Run-level metadata (CHEP 2012)

- aggregate information from the Events table

This data is used during processing (to make decisions) and is sometimes added to the Datasets table to provide more robust and coherent interfaces and reports.

- Specifically:
  
  | Filter input datasets |
  |------------------------|
  | Some Runs and Streams are eliminated |
  | Check dataset status on insert and with time |
  | Checks of event counts: |
  | Dataset completeness |
  | Show counts of duplicated events |
  | Number of duplicate events |
  | Number of unique duplicate events |
  | Show additional counts |
  | Number of data files |
  | Number of unique GUID counts |
  | Rank the datasets to find the ‘latest’ processing |
The EI Oracle Dataset Browser:

Users easily find indexed datasets and their properties.

Robust set of services are available from the browser:

- **Event Lookup**
  - Return GUIDs for events of interest

- **Dataset Report**
  - Show details about the collected dataset
  - Provide links to related reports in other systems

- **Dataset Overlaps Report** → example on next slide …
  - Show events in common between datasets of the same Run

- **Duplicate Event Report**
  - Show datasets with duplicate events and
  - Investigate the source of duplication

- **Count by BCID and Count by LB Report**
  - Show event counts broken down by these properties

- **Missing Event Report**
  - Can show which events are missing in some cases
## EventIndexO Dataset Overlaps Report

**Purpose:** Show event counts (and %) of events in common between selected datasets of a Run

Simple steps from the EI Oracle Browser:

- User specifies the Run and datasets of interest
- Then clicks on the “Dataset Overlaps” Button

### Interface Features:

- **A configurable threshold:** output matrix shows only overlaps above the threshold
- **Offers 2 overlap algorithms:**
  - Intersection / Union \times 100 \% (default)
  - Intersection / (Count in Dataset 1) \times 100\%

### Dataset Overlaps Report

| Dataset      | DAOD_EXOT0 | DAOD_EXOT4 | DAOD_EXOT10 | DAOD_HIGH1 | DAOD_HIGH2 | DAOD_JET1 | DAOD_JET8 | DAOD_JETM | DAOD_JETM8 |
|--------------|------------|------------|-------------|------------|------------|-----------|-----------|-----------|------------|
| DAOD_EXOT0   | 502831     | 10647772   |             |            |            |           |           |           |            |
|              | (100\%)    | (100\%)    |             |            |            |           |           |           |            |
| DAOD_EXOT4   | 427243     | 10647772   |             |            |            |           |           |           |            |
|              | (3.95\%)   | (100\%)    |             |            |            |           |           |           |            |
| DAOD_EXOT10  | 128908     | 107197     | 570221      |            |            |           |           |           |            |
|              | (13.63\%)  | (1.54\%)   | (100\%)     |            |            |           |           |           |            |
| DAOD_HIGH1   | 169929     | 226624     | 570774      | 804424     |            |           |           |           |            |
|              | (14.94\%)  | (2.02\%)   | (7.95\%)    | (100\%)    |            |           |           |           |            |
| DAOD_HIGH2   | 16459      | 1371402    | 6100        | 6982       | 8740180    |           |           |           |            |
|              | (0.2\%)    | (7.61\%)   | (0.07\%)    | (0.07\%)   | (100\%)    |           |           |           |            |
| DAOD_JET1    | 2538       | 225187     | 4599        | 4624       | 159209     | 2145354   |           |           |            |
|              | (0.11\%)   | (1.79\%)   | (0.17\%)    | (0.16\%)   | (1.48\%)   | (100\%)   |           |           |            |
| DAOD_JET2    | 305241     | 357195     | 94981       | 121017     | 13162      | 1013      | 411007    |           |            |
|              | (73.2\%)   | (2.34\%)   | (10.71\%)   | (11.06\%)  | (1.14\%)   | (0.04\%)  | (100\%)   |           |            |
| DAOD_JET3    | 2920       | 212144     | 4588        | 4604       | 158577     | 1600993   | 1007      | 411007    |            |
|              | (0.11\%)   | (1.75\%)   | (0.21\%)    | (0.15\%)   | (1.52\%)   | (77.38\%) | (0.05\%)  | (100\%)   |            |
| DAOD_JETM    | 2939       | 255107     | 4599        | 4624       | 159209     | 2145354   | 1013      | 411007    |            |
|              | (0.11\%)   | (1.79\%)   | (0.17\%)    | (0.16\%)   | (1.48\%)   | (100\%)   | (100\%)   |           |            |
| DAOD_JETM8   | 250965     | 637217     | 756094      | 244036     | 750522     | 141477    | 147833    | 223839    |            |
|              | (2.98\%)   | (57.67\%)  | (8.12\%)    | (9.31\%)   | (7.65\%)   | (0.99\%)  | (9.99\%)  | (100\%)   |            |
| DAOD_STD4    | 491629     | 1043712    | 195011      | 263457     | 1384221    | 231396    | 411007    | 217481    | 231396     |
|              | (2.04\%)   | (48.49\%)  | (1.48\%)    | (1.97\%)   | (6.87\%)   | (1.57\%)  | (1.52\%)  | (1.57\%)  | (1.57\%)   |
|              |             |             |             |            |           |          |          |          |            |
| DAOD_SUS5    | 502891     | 854843     | 282829      | 390570     | 1515835    | 353323    | 411007    | 328423    | 353323     |
|              | (2.92\%)   | (47.46\%)  | (2.12\%)    | (2.22\%)   | (2.22\%)   | (1.66\%)  | (1.52\%)  | (1.76\%)  | (1.66\%)   |
| DAOD_SUS6    | 205660     | 402422     | 116448      | 147191     | 972440     | 160664    | 369635    | 159925    | 160664     |
|              | (2.12\%)   | (21.62\%)  | (1.54\%)    | (1.17\%)   | (72.62\%)  | (1.18\%)  | (3.08\%)  | (19.51\%) | (1.18\%)   |
| DAOD_SUS8    | 262590     | 438310     | 6605        | 6982       | 872402     | 160670    | 265598    | 165262    | 166070     |
|              | (2.04\%)   | (23.13\%)  | (0.03\%)    | (0.02\%)   | (65.66\%)  | (1.13\%)  | (2.07\%)  | (1.17\%)  | (1.13\%)   |

### Color Legend for Percentage Overlap:
- **100\% > 90\% > 80\% > 70\% > 60\% > 50\% > 40\% > 30\% > 20\% > 10\% > 0.01\% > 0\% = 0\%**

Oct 2016

E.Gallas - CHEP: EIOracle
Summary

ATLAS Oracle-based Event Index EIO arose out of exploring various technologies for providing event-wise services using ATLAS Event Index data.

Impressive minimization of resources while achieving performance goals.

Raw data volume: significantly reduced using a relational model and a number of carefully chosen techniques available in Oracle DBMS.

- Non-optimized volume: ~210 bytes/event
  - Volume to store 25E9 events: 5 TB!
- EIO optimizations: volume ~20 bytes/event!
  - 26E9 rows currently in the system: 510 GB!
  - (plus index overhead: 465 GB)

Performance is excellent for “Lookup”: Returns references to single events in a fraction of a second.

- Simulation of 100 users: AVG < 2s for requests of 2000 run-event pairs

A number of integrity issues were found in the initial EI data.

- Many checks are now integrated into the EI Collector – now caught further upstream.
Event-level metadata services for ATLAS based on an Oracle DBMS is deployed
- Users and Experts: Like the performance and the interfaces
- Services cover the most common use cases
  - and additional services were easy to implement
- Success of the project rests on hyper-efficient modelling of the storage underlying the services
  - The system is simple, robust and reliable
- EIO reached a production state very quickly!
  - Design discussions started:
    - December 2015
  - Production DB and interfaces
    - Early March 2016
  - Smooth operations in Run 2
  - Designed to handle future rates
    - 10’s of billions of events/year

Conclusions