Dataset properties

David Adams
Nov 7, 2002

Introduction

A dataset is a collection of associated data intended for use as input to a production or analysis job. Most interesting to HEP is the event dataset where each piece of data is associated with a single event (beam crossing). However we allow for the possibility of other types of data to allow application to other realms and to allow for a dataset that holds both event data and associated conditions data.

Use cases

1. A collection of raw data is used to define a dataset (raw dataset).
2. This dataset and data from a conditions database are used as input to a reconstruction job. The output is a new dataset (reco dataset).
3. The reco dataset is used as input to a job that extracts summary data for each event. The output is another dataset (summary dataset) and might be in the form of an ntuple or database table.
4. The summary dataset is used as input to an analysis job that selects events of interest. The resulting event list is used to make event selections on the preceding datasets to produce new datasets (selected raw dataset, selected reco dataset and selected summary dataset).
5. These selected raw and selected reco datasets and the relevant conditions data are used to define a new dataset (combined dataset).
6. The combined dataset is used as input to a new partial reconstruction program. The output is a new dataset (new reco dataset) in which some of the original reco data is replaced and new pieces are added.
7. The new reco dataset is used as input to another summary job and the output is another dataset (new summary dataset).

Properties of generic datasets

From the above and other use cases, we derive the following properties of datasets.

Data specification

A dataset specifies exactly what data is included. This content does not change with time.

Data access

There must be at least one means to access the data associated with a dataset. The means may be different for different datasets or parts of the same database.

Data organization

The data within a dataset is organized into objects. These may be some persistent representation of C++ objects or may be a row in a database or ntuple. Datasets may share objects and collections of objects.

Global object identification

Each object within a dataset has a global identifier. This should be unique within the scope in which datasets are defined so that objects with the same global identifier in two different datasets are guaranteed to have the same content.

Local object identification

Objects within a dataset may have a local identifier. In an ATLAS event dataset, this might be and event (beam crossing) ID, a type and a string key.

Concatenation

Two (or more) datasets can be combined to form a new dataset.

Extraction

A subset of the data in one dataset can be used to define a new dataset. The subset can be specified using either global or local identifiers.

History

There is history associated with the production of a dataset. There is a means to discover the history associated with a dataset.

Collective properties

In addition to the individual data objects and the history, a dataset may have other collective properties. For an event dataset, this includes the events (beam crossings) that are represented and the type of data for each. There are means to access this data. This data may or may not be part of the data that makes up the dataset.

Equivalence

Two datasets are equivalent if the reference the same data, i.e. the same data objects or their replicas. Object replicas need not be in the same storage format.

Local dataset

It is possible to construct an equivalent dataset for which the dataset and its data may be accessed without reference to external sites (e.g. databases).

Locating objects

In general, a dataset does not physically contain its objects but provides enough information so that the objects can be located. We are most interested in the case where this location is a file (most likely logical) but there are other possibilities. The location might be specified on an object-by-object basis but it will often be useful to collect this

data an hold a list of files where the data may be located. If object replication without change in identity is supported, there might be more than one such list of files.

Properties of event datasets

An event dataset provides access to data associated with specific beam crossings or simulation thereof. Events (i.e. the data associated with each beam crossing) are processed independently and usually sequentially so the event dataset provides access in this manner.

Events

An event dataset provides access to a collection of events. At the minimum, sequential access is required, e.g. an iterator over all events. There is a distinct collection of event data objects associated with each event and the dataset must provide a means to identify and fetch the data associated with each such object.

We assume that each beam crossing has a unique identifier called the event ID. Event datasets must provide a means to access the list of event ID's for which they hold data. Normally (always?) we require that each event ID appear no more than once in this list.

Content

Objects within an event are labeled by a content identifier. For ATLAS, we might follow the transient model and label the objects by type (transient class ID) and a string key. For a given event, the event dataset is able to return the included type-keys and for each type-key, it provides access to the associated data. The event ID and content ID for an object constitute the local object identifier discussed above.

Normally we will require that the content be consistent across events, i.e. that each event was processed in a similar manner and so has the same content (list of content ID's). We allow for the possibility that some content ID's are absent for particular events. It is now sensible to speak of the content of an event dataset.

Merging

It will be a common operation to merge two or more datasets to form a new dataset. It is sensible to merge datasets with the same content and different events (event merge) or to merge datasets with the same events and different content. We expect there will be a special type of dataset to represent each of these operations.

Content restriction

It will often be known that a job only requires a subset of the content in a dataset, e.g. only the summary data or only the tracking data. In this case it is useful to define a new dataset that only presents this view. If the data is organized appropriately, then the list of files required to access the data would be smaller in the new dataset.

Splitting

Datasets can be very large and the files holding their data may be distributed over different physical locations. In these cases, it is natural to divide the processing of the

dataset over multiple jobs. This is accomplished by splitting a dataset along event boundaries and separately processing each sub-dataset.