How to compute the Processing Pass for new data

In another section we have explained how the Processing Pass (PP) has been computed for the already existing data, on the basis of the metadata stored in the Bookkeeping (BK) database. Here we will deal with the calculation of the PP for the new data (simulated and real).

Meeting on 2nd July 2008

Present: Philippe, Joel, Zoltan and Elisa

It was discussed how to compute the Processing Pass on the fly when new data are registered in the BK.

During the migration of the data to the new schema, the processing pass has been computed on the basis of the information contained in the BK. But in the future, for simulated and for real data, the processing pass can be computed at the same moment when a new production is started.

What is the information needed to compute the processing pass?

The information necessary to compute the processing pass is:

- The application name and version of the programs used in a given production, for all the steps (ex. simulation, digitization, reconstruction etc.) and the condition DB tag.
- The Processing Pass Group of the input productions.

Who can provide this info?

The production manager when he sets the workflow for a new production.

How can this info be sent to the BK manager?

The BK will provide an interface (Zoltan proposes a method of the BK client) to be used by the production manager to send this information to the BK. When starting a new production, the production manager knows the versions of the applications to be used in the production and he knows the Processing Pass Group of the input data. (Actually, the Processing Pass Group should be used by the production manager to select the input data. See below for more details on this point). Before starting the production, he will send the information to the BK manager. The BK Manager will do the following:

- Read the application name and version and condition DB tag and compare with the content of the *Pass_Index* table. If the row already exists, it will pick up the corresponding *PassId* and associate it to the production. If the row doesn't exist, it will add a new row and generate a new *PassId*. Then it will insert a new row in the *Prod2Pass* table, containing the production number and the *PassId*.
- If the production has some input data, it will go to the *Processing_Pass* table and look for the Processing Pass of the input data.
- Finally, it will add a new row in the *Processing_Pass* table with: the production number (this is the unique key of the table), the *PassId* (which links to the *Pass_Index* table to expand the information about program name and versions), the *GroupId* (this is the group the *PassId* belongs to), the Total Processing Pass (this is the concatanation of the processing pass of the input production and the processing pass of the current production), the event type and the *DAQId*.

In addition to the information necessary to compute the Processing Pass, we decided to add also the Event Type and the *DAQId* (i.e. the identifier of the Simulation Conditions (for MC) or of the Data Taking conditions (real rata)). This is not necessary for the computation of the processing pass, nevertheless this information is available at this point of the workflow, and it was decided to store also this information in the *Processing_Pass* table for sake of completeness. In this way, each row of the *Processing_Pass* table provides

a full characterization of a production: the processing pass index (*PassId*), the Processing Pass Group, the simulation conditions (or data taking conditions) , the event type.

How can the production manager select the input datasets for a new production

It has been proposed to use the BK to select the input datasets for a new production. Currently, the selection of the input data is based on a regular expression. This method is not safe, because it relies on the convention used to name files, which could change in some moment.

The new proposal is that the production manager queries the BK to obtain the input datasets. The BK should provide an interface to make the query. This could be a method of the BK client.
The parameters to be passed in this query are:

- the Processing Pass Group
- the DAQId
- the Event Type
- the File Type

How to compute the Processing Pass Online (it was started as minutes of the meeting of 22nd September 2008).

Participants: Philippe, Joel, Zoltan and Elisa

How the processing pass is computed online and then inserted into the BK database

First of all, a description of the tables used to implement the processing pass: the diagram of the schema is displayed in the attached picture ppass_tables.png, at the bottom of this page.

the *pass_index* table : each row consists of a unique pass ID and a set of application names and versions (see picture pass_index.html attached). Each of the passes listed in the *pass_index* table belongs to a pass group (the association is defined by the groupid key, as shown in the diagram). Passes of the same pass group are by definition compatible, even if the application version is different.

In the *pass_group* table each row consists of a group ID and a description, which is a user friendly name reminding the type of data processing (ex. stripping v31. or sim_reco_v31 etc...). See picture PASS_GROUP.html attached to this page.

Finally, the *processing_pass* table has a row for each production, where we store the corresponding passid (which connects to the *pass_index* table) and the total processing pass. The total processing pass is a string composed by the processing pass of the input files + the processing pass of the current production (see the attached file below: processin_pass.html). This is the relevant information for users to make queries.

In the meeting it was decided to add a column to the *processing_pass* table, with the simulation conditions ID. It was also decided to modify the definition of 'total processing pass' , adding all the pass descriptions of the input files, from the simulation. Currently only the previous step is included, so for example the total processing pass of a stripping production is 'DC06-Reco_v30 + DC06-Stripping_v31' while it should be: 'DC06-Sim + DC06-Reco_v30 + DC06-Stripping_v31'.

Moreover, the total processing pass should not be given as a string (ex. DC06-Reco_v30 + DC06-Stripping_v31) but as a sequence of the group IDs separated by some separator: ex. '3<5'.

How it works in practice:

when a new production is set the BK interface does the following operations:

- Looks into the *pass_index* table to see if that pass is already present. If yes: it picks the ID of the pass and also the ID of the pass group. If not, it inserts a new row generating a new ID.
- The group ID associated to this new pass should be known when the new pass index is inserted. This is logic, because in principle the PM should know if the version of the application he is setting is compatible or not with some previous versions. So, he will insert the group_id in the pass index, to connect to the *PASS_GROUP* table.
- Inserts a new entry in the *PROCESSING_PASS* table. Here we need: the production number, the simulation conditions (this is known, as it is in the settings of the production), the pass index ID (this is known from the previous operation) and the total processing pass. Here comes the tricky part..
  The total processing pass is by definition the processing pass of the input files + the processing pass of the current production. The processing pass of the input file is known (it is also in the settings of the production), whereas the processing pass of the current production has to be computed. How? We have 2 possibilities:
  1- easy case: the pass index was already in the *PASS_INDEX* table. So we already have the group ID and we can immediately build the total processing pass as the group ID of the input productions + the group ID of the current production.
  2- the not so easy case: the pass index was NOT in the *PASS_INDEX* table. In this case, we do NOT know the group ID of the current production, as it was set as unknown. So, the total processing pass will be, for ex., '3 < X', if 3 is the group ID of the input. Later, the group ID of this new pass index will be added manually in the *PASS_INDEX* table, and a trigger will take care of updating the column 'total processing pass' of the *PROCESSING_PASS* table. This is feasible because the *PROCESSING_PASS* table is connected to the *PASS_INDEX* table with the pass ID key, so any modification of an entry relative to a given pass ID in the *PASS_INDEX* table can trigger an operation in the *PROCESSING_PASS* table.

Some remarks

- What happens if we realize that the groupID associated to a pass index in the *PASS_INDEX* table is not correct? It means that the pass index is NOT compatible with the pass group. So, we have to associate to this pass index a different group, or (if it doesn't exist) create a new group in the *PASS_GROUP* table. In any case, we have to modify the foreign key 'passgroup' in the *PASS_INDEX* table. By consequent, we have to modify the 'totalProcessingPass' entry in the *PROCESSING_PASS* table, for all the productions referring to this pass index. No problem, this can be implemented with a trigger.

- When a new production is started, the BK interface immediately inserts the new entry relative to this production into the *PROCESSING_PASS* table. Then, when the jobs will be registered to the BK, in the xml files we will only provide the information about the production, whereas all the information about processing pass and simulation condition will be redundant.

- The attributes 'simulation conditions ID' will be removed from the jobs table. In this way we can register a job or file in the BK even if the corresponding simulation conditions ID is still not in the *Simulation_conditions* table. This will avoid errors at registration time.

The simulation conditions ID, as the processing pass, will only be stored in the *PROCESSING_PASS* table.

Instructions for the Production Manager (PM)

Here we summarize the actions the PM should follow each time he starts a new production.

- Define the work flow. This is the set of applications to be used in the new production.
- Define the *pass_index* : this is the work flow + the versions of the applications + the pass group ID and insert this new row into the *pass_index* table of the BK, if it is not there.
- Define the production. This requires to have a work flow defined and to make a query to the BK in

order to get a set of input files. The set of input files depends on the following parameters: 1-Configuration 2-event type, 3-simulation conditions, 4- processing pass, 5- file type.
- Enter the production in the BK. This is automatically done by the BK interface which is used to define the production. So it is part of the previous step.

Stripping

To start a new stripping production, a set of input files is needed. Currently these files are retrieved in this way: the PM select a configuration and an event type and on the basis of these parameters he sets a regular expression which matches the names of these files in LFC. The disadvantage of this method is that capturing files with a regular expression is error prone (based on a naming convention, that could change..) and moreover it can happen that a file is in the LFC but it is not stored in the BK. In this case, when the job is sent to be registered in the BK, an error occurs because its input file is not present in the BK! this is an inconsistency which will be avoided if we ask for the list of input files directly in the BK.

According to what proposed in the meeting, the list of input files will be obtained with a query to the BK. The query requires the following parameters (see also above in the item relative to the instructions to the PM): 1-Configuration 2-event type, 3-simulation conditions, 4- processing pass, 5- file type.

The result of the query, together with the work flow previously defined by the PM, will set the new production.

Instructions for the Online people

As soon as they start a new run they should send the corresponding XML file with the following information:

1- Program name and version. The BK will read the XML, check if an entry already exists in the *pass_index* table with this program name and version. If yes, ok, just picks the pass index ID (it will need it to compute the processing pass, see point 3 below). If not, it adds a row in the *pass_index* table with this program name and version and with an undetermined group ID. It also will send an alarm to notify that a new row has been added to the *pass_index* table and that the group ID has to be filled (and possibly the pass description field). This operation can be done a posteriori and has to be done manually. Here there are 2 possibilities: 1-the new pass index belongs to a pass group which already exists in the *pass_group* table. Then the corresponding group ID has to be inserted into the *pass_index* table. 2- this pass index is not compatible with any of the groups in the *pass_group* table. In this case a new row has to be added in the *pass_group* table and a user friendly name has to be associated. An oracle sequence will generate a unique ID for the pass group. Then, this pass group ID will be inserted in the *pass_index* table.

2- DAQConditionId a set of strings defining the daq conditions. Also in this case there are 2 possibilities: 1-an entry already exists in the Data Taking Conditions table. 2- it doesn't exist: in this case a new entry has to be added and an alarm will be sent to notify that a user friendly description has to be associated to this new data taking period entry.

3- Processing Pass they should not provide any number for the production ID. This will be computed automatically by the BK. They should provide the program name and version (they are needed to get the corresponding pass index ID, see point 1) and the data taking conditions (see point 2). At this point the BK can check if there is any entry in the Processing Pass table. If yes, then the production number associated to that entry is taken as the production ID for that run and it is entered into the *jobs* table of the BK database. If not, it adds a new row of the Processing Pass table, generating a new production ID with an oracle sequence. This number will be used as production ID for the run. In the case of raw data, the total processing pass is just the processing pass of the current production (no input here!). This will be computed by the BK interface in the same way used for simulated data (see above).

-- ElisaLanciotti - 02 Jul 2008

- prodDiagram.pdf: Production workflow diagram

This topic: LHCb > ProcessinPass
Topic revision: r12 - 2008-10-14 - ElisaLanciotti