The BRAHMS data production System

F.Videbæk, H.Ito

March 14, 2005

Abstract

The data production system in BRAHMS has been overhauled to reflect better the ideas move forward during the run-3 D-Au analysis, and in particular to have a system that maintains a filecatalog.

1 Introduction

The organization of the production datafiles in Run-2, and albeit somewhat improved in the run-3 analysis, always lacked in clarity an organization. The status of production as well as where to find the newest official version were murky. During the analysis of run-3 D-Au data, the concept of standardized file location and names identified by run period, specie, and energy proved quite succesfull. During the spring of 2004 there were several discussion between Hiro, Bjoern,Kris and Flemming on how a file-catalog should be organized. The idea is basically that the filecatalog is filled on a per run basis every time datafiles are produced in the ltr (local tracking), gtr (global tracking) or dst (summary tree) step, with the production information, and the location of the generated files. This enables one to ask for any further analysis where the files are located, how they were produced and so on.

Utility cgi scripts have been written so one can query about the prouction version and location of most recent files for any reconstruction step. Methos have been added to the brat-classes in order to add the production files for a given run to a bratmain script.

This document describes the information available in the filecatalog, and how it can be used through usage of standard scripts for reconstruction. As mentioned later it is absolutely essential that these and only these scripts are use for prouction, otherwise we loose the avantage of the filecatalog, and will return to general confusion of placement of disks.

Since RCF is promoting the usage of the Condor batch system over LSF will be phased out due to costs both the crs access, as well as the production of gtr and dst has been implemeted using Condor.

The file catalog usage and the utility scripts are still under development. At this time though we feel it is ready to be deployed, and can be used for production reconstruction of the run-4 datasets. Standard scripts for ltr, gtr and dst production are working. Feedback is of course important to ensure a good working system. Please send the message to the brahms-dev-l@lists.bnl.gov on this. As of now (December 11) the system has been in use for a couple of month and seems very stable.

2 Filecatalog Overview

One of the basic concepts is that the location on files on any disk system is well defined. The official files can all be found in directories named `<base><runperiod><system><energy><datatype>`,
where
`<base>:= diskarea | hpssarea`,
`<runperiod>:=run03|run04|run05`,
`<system>:=auau|pp|dau|fefe>`,
`<energy>:=200|130|62|19`.

Likewise the naming of the files is uniform, containing the runnumber, sequence number, a version number and a production version number. The runnumber and sequence number of course matches the original raw data information. The version and production numbers are used to keep track of the generated files, and make it easy for a given analysis to request the newest and most up todate ones. Thus each reconstruction task corresponds will have a specific version number and production no. The production number is designated by the users, while the version number is ensured to be updated for each run. It is possible to assign new production number e.g. if a major revision of software has occured.

Information stored. utilities.

3 Guide for Usage of Reconstruction Scripts

There are specific scripts for ltr, gtr and dst production. For a given runperiod they are to be found in `DataProduction/<run><specie><energy>[ltr|gtr|dst]` directories unbder the bramreco account. These are the only official scripts to be used if the integrety and advantage of the filecatalog is to be maintained. Please do not use the older shell scripts, there are also now scripts available that runs the new CRS system w/o creating official output file. This is useful for testing and debugging. File generated that way should be placed in scratch areas.

3.1 Scripts for *ltr*-production

Since all ltr-files are generated by reading raw data stored in HPSS the CRS software developed by RCF must be used. The production scripts uses the most recent CRS software that is deploye on all the farm and uses CONDOR for the underlying batch system.

The official scripts for running ltr production is in the relevant rundirectory e.g. `/DataProduction//run04/auau/200/ltr`.

This text describe how to use masterReductionCondor.rb to submit a job. This script is (hopefully) improvement from the above script in one thing: it automatically update FILECATALOG database. This database includes many information about how and where the files (ltr, gtr, dst, etc...) are created. For production running in additions the call to the masterReductionCondor.rb is done via a perl script to enable logging of the outputs from running the scripts into an ascii file ltrsubmit.log.

This only works at present from rcrsuser4 since it uses Condor, and the only crs machine where the software is maintained.

1 At first, you need to set up enviromental things for Condor by sourcing `setupCrsCondor.sh`. This is done by login time for the bramreco account.

2 Then, you can run this script with options. To see the list of options, use `--help` option. A complete listing is in the appendix.

For example, to submit a job for run 11880 - 11882

```
./masterReductionCondor.rb --userName hito --email hito@bnl.gov \
    --beginRunNo 11880 --endRunNo 11882 --productionNo -1 \
    --write-jsf --submit-jsf
```

In fact with the perl script you should simply execute

```
./submitLtr.perl 11880 11882
```

The perl script has the advantage it also maintains a log of submission i.e. one can see when and where jobs are submitted.

3.2 scripts for *gtr*-production

These production script submits job to the Condor system. The global trak reconstruction needs offsets and cuts parameters for matching. The idea of first generating offset files followed by a reconstruction pass w/o having specific files was first implemented by Djamel. Subsequently this was implemented using ruby scripts by Hiro. For a given run the sequences are divided into subsets that are analysis once to generate the offset files, and then to run reconstruction using those files. Thus it takes about twice along, but one is ensure to use a proper set of matchings.

(Note we need some script/tools to check the quality of the fit i.e. summary root based scripts). As always one really have to check the quality of both the histogram offsets and the

These scripts are submitted from the cas machines, and will by default run on the CRS nodes. By default the jobs will be submitted to the queues that encompasses all machines starting with the fastest. In addition the crs job has priority over other other condor jobs submitted, that will be preempted if running

3.3 Scripts for *dst*-production

example:

For example, to submit a dst generation job for run 11331.

```
./masterDst.rb --userName=videbaek --email=videbaek@bnl.gov \
    --beginRunNo=11331 --inputProductionNo=1 --outputProductionNo= -1
```

The output files i.e. the standard output and standard input can be found in the data directory files associated with the dst output files i.e.

4 Access to filecatalog from brat

The great advantage of the filecatalog becomes clear when using files for analysis. The end-user do not need to know on what disk the actual files are located, but can just refer to them by runnumber, type, and production version. This will pick up the latest version.

The easiest way is use the overloaded `AddFileSet` method to BrIOModule. It uses our FILECATALOG database to find locations of files. You don't need to find and remember the file locations anymore. The format is shown below.

```
AddFileSet(Int_t runno, Char_t* type, Int_t prono, Int_t verno=-1111)
```

The parameters are

runno run number

type "ltr", "gtr", "dst"

prono — production number. The production numbers can be found on `http://pii3.brahms.bnl.gov/daq-cgi-bin/EnterNewProduction.pl`

verno — optional Only if you really want to use the specific version number. Default behavior is to pick the most recent version.

As a second option you want to find the dst files and add to a chain for further analysis. This can be done in a regular script, assuming you are executing

```
BrFileCatalogManager *fileCatalog = BrFileCatalogManager::Instance();
fileCatalog->Update(runno,"dst",prono);
BrFileCatalogInfo *info = fileCatalog->GetCurrentRun();
const Char_t *filename = info->GetNextFile();
```

And of course before that the db infrastructure has to be started in the regular way at the beginning except that the file catalog has to be connected. ie

```
BrMainDb* mainDb = BrMainDb::Instance();
mainDb->SetUserName("query");
mainDb->SetHostName("brahms-db0.rcf.bnl.gov");
mainDb->SetDbName("BrahmsMain");

// Un comment if you need access to database
if (!mainDb->Connect()) return;
if (!mainDb->ConnectToRun("query")) return;
if (!mainDb->ConnectToGeom("query")) return;
if (!mainDb->ConnectToFileCatalog("query")) return;
BrGeometryDbManager *geometryManager = BrGeometryDbManager::Instance();
geometryManager->SetDbModeMySQL();

BrRunInfoManager *runManager = BrRunInfoManager::Instance();
runManager->Init();

BrFileCatalogManager *fileCatalog = BrFileCatalogManager::Instance();
fileCatalog->Init();
```

The technique described above can also be used in regular bratmain scripts where most of the statement from the latter section are already there.

4.0.1 Example.

For example, you want to analyze run 11724 ltr file with production
version number of 1, and will add the following line to the bratmain script

```
inputmodule->AddFileSet(11724, "ltr", 1);
```

5 Managing the Filecatalog

The end-user in general only need to know how to fin the status of specific runs, possible how to submit runs when responsible for production. There are though another set of tools needed to maintain the system. These are briefly described in the following subsections.

5.1 File handling

Since the central storage is limited, the capability for deleting, moving, and storing on HPSS will become crucial. A tool for moving to HPSS has been written. Please consult with Hiro before using. The web based query will tell you if files are saved in HPSS.

5.1.1 File removal

This is a management tool and can only be executed by bramreco. Please exercise with care. It is possible to delete lots of files with a few key strokes.

Since the files are in general lost, unless previous store on the HPSS system this should be done with great care. The actual status of files should be chekcked with the DAQ webpage query before actually using the command. The command remove ALL files associated with a given run, datatype, production and version i.e. datafiles, histogram files as well as log and error files.

For example, to delete the ltr files on disk from run 10202

```
./rmata.rb --beginRunNo 10202 ---inputProductionNo=1 \
                --dataType ltr --versionNumber 0
```

This removed the files, error, output and histograms files generated , and deletes the entry in the filecatalog. Note that no warning is given if it is the most recent version.

5.2 Script maintanence

For each production all the scripts are really identical apart from the default values. Therefore all scripts (RunInfo.rb,...) are in /run04/scripts/ltr as RunInfo.rb.in etc. These files are commeitted and maintaned in CVS. For each production there is a simple shell script `updateScripts.sh` that takes the scripts from this master directory replaces defaults and puts scripts in `<run><specie><energy>/ltr` ready to use. Thus if changes are to be made in the run by scripts there is ONLY one place now, followed by exceucting the updatescripts for each prouction directory.

5.3 Important things still to be done

TODO: There are several things not in place yet.

1 repeat for gtr scripts (DONE)

1 repeat for dst scripts (DONE)

2 place the common scripts e.g rmdata.rb same place(DONE)

3 scripts to check consisntency of files on disk - vs filecatalog. This will involve the following tasks. Delete files on official directories if NOT in filecatalog. If files are not on disk

4 scripts to move data to/from HPSS, and to move between different disk areas, including local disks on rcas machines. official disk area. (IN PROGRESS)

5 Setup so the large local disk on farm machines can be used for dst, gtr files via the `rootd` mechanism.

5 Note we need some script/tools to check the quality of the reconstruction i.e. summary root based scripts from all the steps. This should included merging histograms, and producing summary output plots, possible as for the supermonitor that puts the hist in specific places where they can be looke at `bramrecopubli_htmlrun04auau200.. ??)`

6 CRS software

The documentation for the CRS software is pretty slim. The following section is taken ad verbatim from the RCF webpage. Also this is not quite stable, and may be changed for optimization.

See `http://www.rhic.bnl.gov/RCF/UserInfo/Software/CRS-New/`

The New Reconstruction Control Software (New-CRS) handles the scheduling of reconstruction code and defines the environment in which reconstruction code will run. The definition of the environment is outlined in the Reconstruction Control Software Interface document which is available in HTML. The job description in new CRS is identical to the job description in the old system.

6.1 How to use the software.

1. Logon to rcrsuser4 using the reconstruction account.
2. Set the DISPLAY environment variable. Set the xhost permissions on your machine.
3. Execute setup_crs command. This command makes sure that you have the most up to date version of software. You should execute it each time you start working with CRS, at least once per day.
4. Start crs panel: crs_panel [options]. The options define the font type and display width of the panel, you should adjust them to your terminal. Use h option to get help.
5. Prepare a job description file (jdf), exactly like you did it with the old CRS system.
6. Create job using crs_job command, then submit it using submit job command from crs_panel OR create and submit job using crs_job command
7. You can follow the status of jobs using crs panel, or using crs_job -stat command. (The panel is recommended). I will add a web page with job status later.

Description of commands
crs_job [-command] [jdf_file] [-options]
The available commands are:

- -stat : get information about known jobs
- `-create job_description_file`

  $-options$

  Create a job from the job description. Job will be registered in CRS system in CREATED state under a name job_description_file-timestamp. The valid options are:

`-qn` send job to queue n;

`-pn` send job with priority n;

`-drop` allow the job to drop to lower queue if the requested one is full.

- -submit jdf_file : submit a job described in jdf_file. It must be in CREATED state.
- -submit_all : submit all jobs in CREATED status
- -create_and_submit jdf_file: create and start a job.
- -kill jobname : kill job with this name.
- -kill_status status : kill all jobs with given status. For example:
- crs_job kill_status ERROR : kill all jobs in status ERROR
- -reset jobname : reset job, bring it to CREATED state

Examples:

- `crs_job` stat : show list of known CRS jobs
- `crs_job` create jdf_file p10 q2 : create a job from jdf_file, send it to queue 2 with priority 10.
- crs_job create_and_submit jdf_file p10 q2 : create a job from jdf_file, send it to queue 2 with priority 10 and submit the job to condor.
- crs_panel [-options]

  Start crs panel. The options are: -wn set the panel widt to n; -fn use the font n. recommended stettings are -w80 -f10 to get onto most display systems.

7 Appendices

7.1 Options for ltr-scripts

Some specific details on the scripts are given here. For any given point in time it may be best to to check the actual master scripts.

```
"./masterReductionCondor.rb --help"   will produce
You asked to see available options.  See list below
\begin{verbatim}
--userName
--outputDir
--paramFile
--dataDir
--priority
```

```
--ffsAngle
--productionNo
--write-jsf
--maxSeq
--mrsAngle
--help
--email
--verbose
--beginRunNo
--configFile
--submit-jsf
--comments
--endRunNo
```

Some options have a default value.

```
configFile = /brahms/u/bramreco/DataProduction/run04/pp/200/ltr/ppLocalTracking.C
paramFile =
outputDir = /brahms/data18/
dataDir   =
mrsAngle  = nil
ffsAngle  = nil
priority  = 10
maxSeq    = 10000
```

The default values are gicen for the pp localtracking. If the defaults are to be changed i.e. outputDir and configFile edit the updateScripts.sh script, execute it and commit to CVS. Do **NOT** edit the ruby script files themselves.

```
Some options are required.
userName
email
beginRunNo
endRunNo
productionNo
```