Job and Data Management in DC’04

Ricardo Graciani Díaz
Universidad de Barcelona

12 January 2004

Abstract

In order to improve the system performance for DC’04 a proposal for job splitting and merging has been presented. The aim of the proposal being that the size of the final files stored in the system is significantly increased for a more efficient access to them and to the databases that answer queries on them. The implementation of this proposal has several implications in the job submission mechanism and in the handling of the produced data. Different implementations are discussed and their consequences analysed.

1. Introduction

On LHCb-Note 2003-158 a proposal for a new job splitting and merging scheme for DC'04 was presented. The aim was to increase the size of the data-files stored during the production by merging them once they are produced. An optimal file size is estimated to be in the range 1-2 GB.

Imposing the condition that jobs can not be arbitrarily long, this results in a scheme like the one shown in Figure 1. The fact that the merging of the intermediate files has to be done between "Steps" is due to the way crossed references are implemented. They include explicit references to the LFN containing the referenced objects. If the merging is done after these references are created they will be lost afterwards.

This scheme introduces yet another problem in what respect the managing of production jobs. Namely, the input data for some of the steps (Boole+Brunel[1]) will not be available until Jobs from the previous Steps are ready and their output is merged.

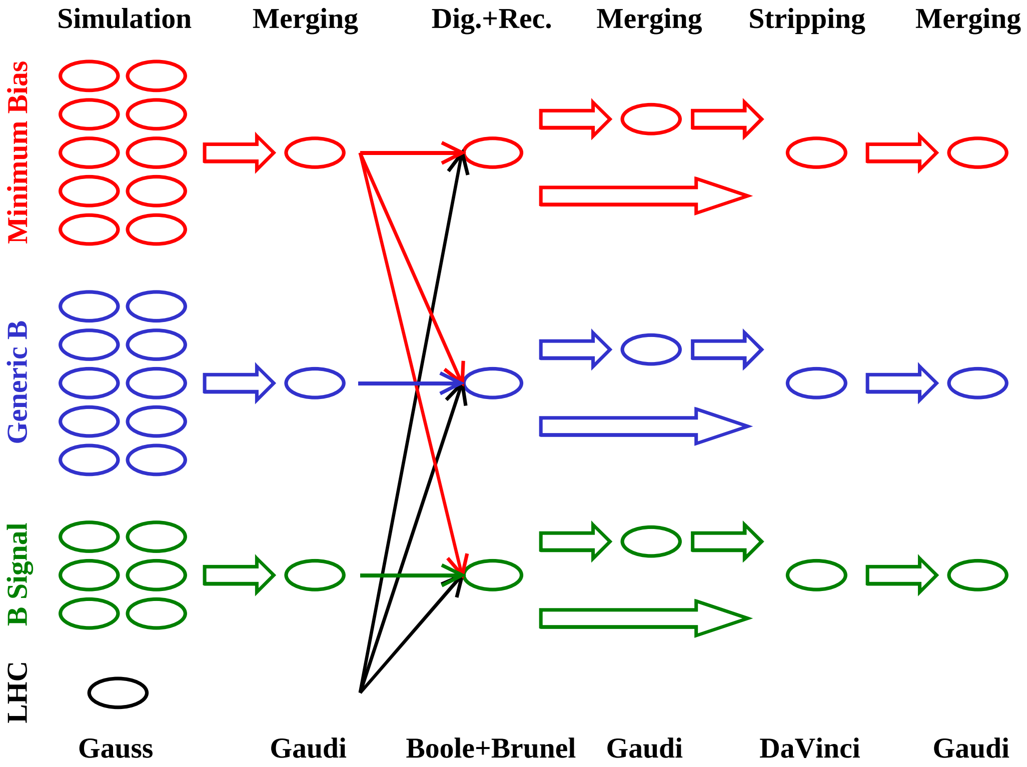

Figure 1 Pictorial view of the proposed production scheme for DC'04. Simulation data are produced in standalone jobs that are input to merging jobs. The outputs of these merging jobs are later used by digitisation and reconstruction jobs. If these jobs include some event selecting algorithm, their output is to be merged. Finally striping algorithms will be used and the resulting outputs are to be merged again.

The aim of this note is to examine different alternatives to realize this splitting and merging during the DC'04. Possible alternatives are presented in the following sections, their advantages and disadvantages are discussed.

[1] Or DaVinci if the stripping is included as a last Step in the production.

Table 1 presents the job parameters corresponding to a possible scenario of the splitting of the production. It is taken from the proposal on LHCb-Note 2003-158.

There are two different aspects of the problem to be considered, one is the managing of production jobs and the second is the managing of the data. This note will mainly concern with the managing of the jobs.

Job	N of Events	CPU (h)	Size (MB)
Gauss Minimum Bias	2000	28,89	228,00
Gauss Generic B	500	18,61	148,00
Gauss b Signal	500	29,17	148,00
Gauss LHC	4000	11,11	104,00
Merging Minimum Bias	8000	0,04	912,00
Merging Generic B	4000	0,05	1184,00
Merging b Signal	4000	0,05	1184,00
B+B P-S Minimum Bias	24000	42,67	3,89
B+B Minimum Bias	24000	42,67	3888,00
B+B Generic B	8000	39,78	1296,00
B+B b Signal	8000	39,78	1296,00
Merging P-S Min. Bias	10000	0,07	1620,00

Table 1 Summary of main job parameters for DC’04 production. The columns show the number processed of Events, the CPU time and the size of the output files for each of the Steps to be used.

2. Predefined Input Files

The simplest approach from the point of view of the job submission is to predefined, from the very beginning, all the input files for a complete production including the simulation, digitisation + reconstruction and the merging jobs. In this scenario the DC’04 Production could be split in self contained mini-productions, with fully defined input and output files. Examples of these minimal productions are given in Table 2, Table 3 and Table 4 for the b Signal, Generic B and Special Minimum Bias cases.

This alternatives requires the following extra functionalities (with respect to last year’s production):

1. Workflow editor tool able to handle these more complex workflows.

2. Workflow submission tool able to split these long workflows into independent jobs (including, in some cases, several steps to optimise the access to input data).

3. Workload Manager tool able to submit all jobs of a given mini-production to the same production centre and able to discover the availability of the input data before subsequent jobs of the mini-production are submitted to the centre.

4. Failed Job recovery mechanism, that “resubmits” them since their outputs will needed for subsequent jobs to be submitted.

The main drawbacks of this approach are that it puts many new requirements on the Workload Managing System, including a Job Recovery mechanism, and that it does not completely solve the problem since it can not include the merging of pre-selected Minimum Bias Events (the workflows will be huge in this case) and it can not be applied to handle the stripping.

	Events	Jobs	CPU (h)	Size (MB)
M-B Simulation	16.000	8	231,11	1.824
M-B Merging	16.000	2	0,08	1.824
b Simulation	8.000	16	466,67	2.368
b Merging	8.000	2	0,11	2.368
LHC Simulation	24.000	6	66,67	624
B+B	8.000	1	39,78	1.296
Production	**8.000**	**35**	**804,41**	**6.112**

Table 2 b Signal minimal predefined production parameters.

	Events	Jobs	CPU (h)	Size (MB)
M-B Simulation	24.000	12	346,67	2.736
M-B Merging	24.000	3	0,12	2.736
LHC Simulation	24.000	6	66,67	624
B+B	24.000	1	42,67	3.888
Production	**24.000**	**22**	**456,12**	**7.248**

Table 3 Generic B minimal predefined production parameters.

	Events	Jobs	CPU (h)	Size (MB)
M-B Simulation	48.000	24	693,33	5.472
M-B Merging	48.000	6	0,24	5.472
B Simulation	24.000	48	893,33	7.104
B Merging	24.000	6	0,32	7.104
LHC Simulation	72.000	18	200,00	1.872
B+B M-B	48.000	2	85,33	8
B+B B	24.000	3	119,33	3.888
Production	**24.000**	**104**	**1872,56**	**18.344**

Table 4 Special Minimum Bias predefined production parameters.

3. Automatic Input File Definition

A completely opposite alternative is to introduce a new type of Workflow that allows to define is input files as a set of N different output files corresponding to one or more already defined productions. These new Workflows are instantiated into jobs in two asynchronous steps. First they are submitted to a “Production Server” (it might be included in DIRAC or be something different). Later on, the Server is in charge of “collecting” the information about new files becoming available in the system, instantiating jobs as soon as enough data files are there. The Server then submits them as normal DIRAC jobs.

The required new functionalities are:

1. A modification on the Workflow definition to handle this new type of Workflows.

2. The “Production Server” functionality must be implemented either as part of the existing DIRAC System or as an independent module. It must include the “collecting” of new data files, the job instantiation, the job submission plus an accounting mechanism to avoid duplicated used of the input files.

This scheme is very flexible and can easily be used for reprocessing of data, stripping and merging of pre-selected or striped data. The main drawback is that is not optimised from the point of view of data handling in what respects the merging. Data to be merged is copied once to storage and then read back and copied again to the storage by the merging job, that eventually will also remove its input.

4. Manual Input File Definition

An intermediate solution is that the Production Manager takes care of querying the Bookkeeping System about the data that is ready at the different production sites, instantiating the new jobs for the different productions (merging, B+B, …). Some tools are needed to help him in the task:

1. The Workflow editor must easy the job submission by accepting the input of several input file lists (i.e. LHC background, Minimum Bias, and Generic B). It should provide a Instantiating and Submitting Tool able to prepare (and submit) as many instantiated jobs as possible with the given files, knowing how to include the appropriated input files at the different places of the job description.

2. An accounting tool is needed, between the query to the Bookkeeping system and the input to the Workflow editor, to avoid duplicated use of files.

There are major objections to this approach, since it puts all the responsibility on the hands of the Production Manager, that instead of taking care will spend most of his time preparing and launching small productions. A possible improvement to this situation implies waiting until all the jobs of a “Step” in the production are all ready before launching the next “Step”, this eases the task of the Production Manager at the cost of introducing important latencies.

5. Comparison of the Alternatives

Three different alternatives have been presented that put most of the changes at different places of the production chain. The predefined alternative puts all the effort in the Workflow Editor Tool, the automatic one puts all the stress on the new “Production Server”, and the manual one requires extra effort from the Production Manager.

As mentioned above, the predefined option is not able to complete solve all the aspects of the problem, therefore is considered the less favourable one. Furthermore a complete Failed Job recovery mechanism is need in place to avoid that a single job failure causes the lost of large amounts of data.

The automatic alternative is probably the optimal one from the point of view of the managing of the production since the introduction of the “Production Server” solves most of the issues.

The manual alternative, assuming that long latencies are acceptable so that the requirements to the Production Manager are reduced as much as possible, is probably the most conservative approach and can be considered the fallback solution.

If the “Production Server” implementation is done outside DIRAC, the manual and the automatic alternatives are highly compatible in terms of requirements both to the Workflow editor and the DIRAC systems.

6. Conclusions

For the implementation of the Job Splitting and Merging proposal presented on LHCb-Note 2003-158, three different alternatives are contemplated. They all deal with the problem of merging several intermediate output files into larger files that are to be further processed.

From the three alternatives presented, the automatic one is found the most interesting, since it would solved in a general case the problem of launching a production (merging, reprocessing, filtering,…) in the case where the input files are *a priori* non-existing.

The manual alternative, allowing significant latency to simplify the task of the Production Manager is considered the fallback alternative.

I Hope this brief document could serve to guide the discussion on the different alternatives towards a more efficient use of our computing resources during the coming DC'04.

7. Acknowledgements