DØ Collaboration (Note number 3992)
May 3, 2002

Level-3 Muon Dynamic Unpacker

C. Clément[a], S. Snyder[b]

[a]**Stockholm University**
[b]**Brookhaven National Laboratory**

Abstract

This note presents the dynamic unpacker for muon data at level-3 contained in the p11.04.00 release and forward. A description of the dynamic unpacker algorithm is given. Various data samples including runs with a missing muon module and a missing muon crate are unpacked using the dynamic unpacker as well as the old configuration-file based unpacking algorithm. The resulting hits in the muon system are compared. Timing, memory consumption and stability tests for level-3 are also presented.

1 Introduction

Until p11.04.00 the muon data at level-3 were unpacked using a configuration-file based unpacker. This “old” unpacker has the major drawback of requiring a so called “configuration file” or `cfg.dat` file, reflecting the exact muon crate configuration of each run. We refer to this unpacking scheme as “file-based” unpacker.

The configuration file required by the file-based unpacker must contain the *exact* list of crates in the run. For the file-based unpacking to work properly, if a crate is removed from the run a new configuration file reflecting this change must be used and a new run must be started. In this scheme the configuration file must also hold the list of modules and their type (PDT, MDT or scintillator) for each crate. Furthermore the list of modules per crates must reflect the *real* order of the modules in the crates. If a module is removed from a run or if two modules are swapped the configuration file must be edited correspondingly and a new run must be started.

Changing the configuration file on the level-3 farm requires making a new tarfile of a whole release and download it on to the farm nodes. Since changes in the muon system crate configuration can happen often, the file-based unpacker is very unpractical for the level-3.

For this reason a dynamic unpacker is prefered to the file-based unpacker. The dynamic unpacker only needs to know the list of all crates *potentially* in the run, but uses the data itself to determine which crates were really read out. It also needs to know about all possible modules per crate and their type. But it does not need to know their order in the crate, and determines from the data itself which modules were read out.

The list of all crates *potentially* in the run and the list of all modules *potentially* in each crate with their type is called the “master configuration”. While the exact crate configuration can often change from run to run, the master configuration is in principle very stable. In practice it means that a single configuration file (`cfg.dat`) can be used throughout all runs. This configuration file can be the file used for processing Monte Carlo events. The dynamic level-3 muon unpacker presented here is similar to the unpacker used for offline analysis [1].

The dynamic muon unpacker for level-3 has been integrated into the DØ software from the p11.04.00 release and forward [2]. This note describes the algorithm of the dynamic unpacker and compares its results with those of the configuration file-based algorithm on several data and Monte Carlo samples including runs with missing crates and missing modules.

2 The muon dynamic unpacking algorithm

The muon data formats to level-2 and 3 are described in [3]. The dynamic unpacker proceeds through the following steps:

1. The unpacker gets a list of muon crates from the run configuration manager. It loops over these crates and tries to unpack all muon crates listed by the run configuration manager that are also found in the raw data chunk.

2. The dynamic unpacker determines the number of modules in each crate by looking at the data. It starts reading the muon crate header and its size and then moves to the first

module header. The length of the first module data is read from the module header. It then jumps by the length of the module to reach the second module header. There it reads the length of the second module data and jumps to the third module header and so on until the end of the crate data. The buffer address of the beginning of each module data as well as the number of modules in each crate and the size of the data associated to each module is determined in this process. The buffer address of the beginning of each module data is stored in a **vector of module addresses**. The number of modules found in data is called the **counted number of modules** and the size of the data associated to each module is called the **actual module data size**.

3. The unpacker checks the size of the muon crate header, that the crate number in the muon crate header agrees with the crate number from the run configuration manager and that the number of modules given in the muon crate header agrees with the number of counted modules. If a disagreement if detected a message is sent to the logfile but the crate is still unpacked.

4. In each crate the dynamic unpacker loops over all muon modules by looking at the vector of module addresses created when reading through the crate data. The size of the module header is checked for each module. The unpacker obtains the module ID from the data and the corresponding module type is obtained from the run configuration manager.

5. For each module several checks are done. In the following cases the module will be skipped:
    (a) if the size of the module data indicated in the module header is not equal to the actual module data size,
    (b) if the overflow bit (bit 15 in event status register II of module header) is at 1,
    (c) if there is a bunch crossing number mismatch at L1 (bit 3 of event status register I),
    (d) if there is a bunch crossing number mismatch at L2 (bit 5 of event status register I),

6. In addition the unpacker will issue an error message but will unpack the module if the crossing or the turn number found in the module header disagree with the crossing and turn number given by the event collision ID.

The muon unpacking tool is defined in the `l3fmuo_unpack` framework package. The classes and methods for the unpacking itself are implemented in the `simpleunp_muo_unpacker` package.

3 Comparison between file-based and dynamic unpacking

In this section the two unpackers are compared on several data samples. The PDT, MDT and scintillator hits are computed with both unpackers and all variables associated to every hit are compared between the two unpackers. The variables associated to the PDT and MDT hits are the barrel number, the drift distance and drift time, the eta and phi angles, the global coordinates of the hit, the layer, octant and plane numbers. For the scintillator hits there is no drift distance nor drift time, but the time given by the scintillator. The following samples are used for this study:

1. a central muon sample (102,033 events passing the central muon level-1 trigger),
2. a forward muon sample (75,674 events passing the forward muon level-1 trigger)
3. 5761 data events taken while a PDT module was not read out (run 146338),
4. 6418 data events taken while the muon crate x35 was not read out (run 148939),
5. 4157 single muon Monte Carlo events with $5 < p_T < 100$ GeV.

3.1 Central and forward muon samples

For all events both unpackers find exactly the same number of MDT, PDT and scintillator hits. All variables associated to the hits are identical in both unpackers.

3.2 Missing module run

The missing module in this run is a PDT module. Of the 5761 events of the run with a missing PDT module that were processed, the two unpackers give exactly the same PDT, MDT and scintillator hits except for event 4074251 of run 146338. For this event the dynamic unpacker finds 12 scintillator hits instead of 11 in the file-based unpacker. A closer look at packed data shows that the actual number of hits is indeed 12. It appears that the hit that is not found by the file-based unpacker is actually coming from a SFE channel used for calibration purpose. For that particular event this calibration channel was fired by electronic noise. The file-based unpacker checks that the unpacked channels are not calibration channels and therefore rejects this hit while the dynamic unpacker does not.

The impact of this extra hit on the physics is negligible for the following reasons:

- such hits are very rare (indeed absent of the ~177,000 events of the forward and central muon runs used in the previous section),
- although there is an extra scintillator hit in this event, this is not enough to affect the number of LOOSE and TIGHT muons in the event.
- the threshold for the calibration channels to fire will be raised to decrease the occurence of such hits [4].

3.3 Missing crate run

In this run crate x35 (PDT crate) was not read out. 6418 events of this run were processed with the file-based and dynamic unpacker. For all events the unpackers give the exact same number of scintillator, PDT and MDT hits, and the hit properties are identical.

3.4 Monte Carlo sample

4157 single muon Monte Carlo events were processed with the file-based and dynamic unpacker. For all events the unpackers give the exact same number of scintillator, PDT and MDT hits, and the hit properties are identical.

4 Configuration file tests

For level-3 the master configuration is read by the `L3RunConfigMgr` tool from the `cfg.dat` configuration file. Since the dynamic unpacker uses only the master configuration and not the detailed crate and module configuration it should be insensitive to swaps of modules inside the crate descriptions in the configuration file. This feature has been tested by processing events through the dynamic unpacker with three different `cfg.dat` files:

1. the configuration file containing the exact crate and module configuration as required by the file-based algorithm,
2. the configuration file as in 1 where all lines within a crate have been mixed up,
3. the configuration file used for Monte Carlo runs.

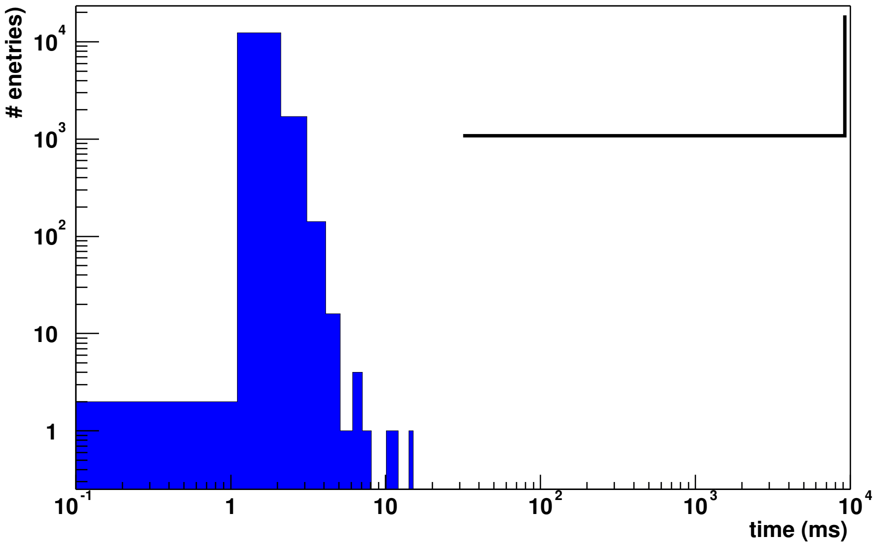
The same PDT, MDT and scintillator hits are found using the three different configuration files.

5 Level-3 certification

Before being run on the level-3 farm all level-3 tools must fullfill strict memory consumption, timing and stability criteria [5].

5.1 Stability and online test

The certification of level-3 tools and filters requires a stability check on at least 100,000 events. To date the level-3 muon dynamic unpacker has been run online in test runs using 2 farm nodes (including also a level-3 muon filter) for about 300,000 events without crash.


enetries)
10^4
10^3
10^2
10
1
10^{-1}
1
10
10^2
10^3
10^4
time (ms)