CERN, 22.06.1997 CHORUS Internal Note 97009

REPORT on EFICASS Improvements

Roumen Tzenov

This is a short Internal Note about EFICASS status and improvements, achieved during my stay at CERN (10.05 - 24.06.1997)

The work has been done in close collaboration with Gabriella Catanesi, Jan Visschers and Helge Meinhard.

1. STATUS of the code before the intervention.

A. Memory needed for execution.

The latest versions of executables for RS_AIX4.1, HP_UX10.20 and ALPHA_OSF3.2c in /chorus/dev/bin have sizes and demands for memory given below:

```
RS_AIX4.1
/afs/cern.ch/chorus/offl/rs_aix41/dev/bin
-rwxr-xr-x   1 jvissche v5       6205497 Mar 25 10:08 eficass*
```

memory: 69028853 bytes (65.83 MB)

```
HP_UX10.20
/afs/cern.ch/chorus/offl/hp_ux102/dev/bin
-rwxr-xr-x   1 jvissche  v5      5148672 Feb 21 13:08 eficass*
```

memory: 65405710 (62.38 MB)

```
ALPHA_OSF3.2c
/afs/cern.ch/chorus/offl/alpha_osf32c/dev/bin
-rwxr-xr-x   1 jvissche v5       6586368 Mar 25 10:04 eficass*
```

memory: 66046992 (62.99 MB)

In particular, the large needed memory on HP_UX platform prevent us of running the code on CSF facility (max allowed memory demand there is 50 MB) and using allocated for CHORUS CPU time there.

B. CPU time consumption.

Tests have been performed on execution time per generated event and where this time is spent.

Remark: All tests have been done with a beam file
        /afs/cern.ch/user/j/jvissche/public/beam/jetta.rfz,
        which contains tau -> 0 mu events.

The time chart, when running with standard run cards (cuts and step sizes) is as follows (based on first 10 events from the beam file run on CERNSP):

Initialisation of geometry etc. : 8 s (once per job);
Mean CPU time per event: 274 s;
from this for:

- tracking: 175 s (64%);

- digitisation: 96 s (35%) This time is spent entirely (>99%) for fiber target and diamond trackers digitisation;
- output : 2.2s (1%).

Tracking the particles through the set up consists of:

~ 2x10**6 steps/event (~11200 steps/sec)

~ 1.75x10**6 calls to subr. GTNEXT (to find next point to go to) and

~ 0.68x10**6 calls to subr. GTMEDIA (to find in which volume the particle is).

Sharing the number of steps among sub-detectors (time spent is proportional to this number):

Target region:	1.9%
Hexa magn:	.6%
Calorimeter	96.0%
Spectrometer	.2%

and the rest elsewhere.

It's not surprising from physical point of view, that almost all tracking is in CALO. This is the first (and most) massive part of the detector which secondary particles enter. When CALO is "hidden" than 81% of steps are in SPEC (but already out of 0.41x10**6 per event).

Other tests:

1) set AUTO flag = 1 (automatic step length calculation). In standard cards it is 0, which means that user defines step lengths in different tracking media.
This gives 17% less time per event.

2) use of standard GEANT cuts on energies below which particles are not tracked more. (Some of our cuts are 2 orders of magnitude lower...).
This reduces the tracking time 4.5 times (with AUTO=1):
from 175 s/event to 39 s/event.

I talked to Pierro Zucchelli about steps and cuts that we currently use. His opinion was:

> NO increase of step sizes and cuts is possible without spoiling the fine tuned consistency between MC simulation of calorimeter response an data. If we'd like to use MC events for tuning (and understanding) of kinematical selection procedures on real data, we could not loose steps and cuts.

2. Optimisation of MEMORY usage.

Some efforts have been made to optimise and reduce the memory needed by the code for execution.

All arrays of size more than 64 kB have been inspected.Some of them have been EQUIVALENCE'd to ZEBRA working space or to predefined arrays in standard EFICASS common blocks The use of others has been avoided at all.

Conditional compilation of those parts of code, which deal with graphical representation of generated events, has been implemented. Now producing of two executables is possible:
eficass_graph - with full graphical functionality implemented so far, and
eficass - without graphics.

The last has an advantage that needs less memory (-1.25 MB) and, more important, has more space in main GEANT dynamical storage for the event. (Information needed for drawing of tracks is not stored there.) This will make the code more stable against ZEBRA FATAL abnormal ends.

Now the sizes and memory demands look like:

	RS_AIX4.1	HP_UX10.20
eficass:		
size	4825783	4277888
memory	44633414 (42.6MB)	41476974 (39.6MB)
eficass_graph:		
size	6074984	5251624
memory	46021163 (43.9MB)	42585737 (40.6MB)

This allows execution on CSF and on PC's (under LINUX) equipped with 48 MB of RAM (like most of Chorus PC).

Warning:

Obtaining executable, running on HP_UX has been possible only by applying -K compiler's option (all local variables static and predefined to 0). This degrades the run time performance but without this option one gets memory fault like this one :

```
( 0)  0xc0c708d4   traceback + 0x14  [/usr/lib/pa1.1/libcl.1]
( 1)  0xc012f2e0   _sigreturn  [/usr/lib/libc.1]
( 2)  0x00020c00   genccd_ + 0x460  [./bin/eficass]
( 3)  0x0002772c   ugdrpd_ + 0xc1c  [./bin/eficass]
( 4)  0x00023ba4   gudigi_ + 0x1e0c  [./bin/eficass]
( 5)  0x002307d0   jumpt0_ + 0x18  [./bin/eficass]
( 6)  0x000bfe9c   gtrig_ + 0x7c  [./bin/eficass]
( 7)  0x000798d8   grun_ + 0x80  [./bin/eficass]
( 8)  0x0005b83c   _start + 0x2d4  [./bin/eficass]
eficass_run[50]: 150 Memory fault
```

"genccd" is one of routines for digitisation of fiber trackers. I would strongly advise to debug it and, as a result, to avoid -K option in compilation.

3. The size of OUTPUT file

The size of output file (per event) is quite large, due to many hits for many particles stored in emulsion sheets and written to this file. Tested again on that tau0mu beam file the output was of about 281 kB per event.

An option, controlled by run card, has been implemented to store hits only in that emulsion stack, where the primary vertex (neutrino interaction) is positioned.

Now the situation is like this:

= we have option (run card flag THRE(1)) for storing hits in emulsion (bulk+SS+CC) and, hence, "digitised" output only when in the stack, where primary vertex (neutrino interaction) is. If it is NOT in emulsion (not in so called EMPA volume, can be when BEAM = 7), hits are not stored at all.

= we have momentum threshold, set by run card THRE(2) and THRE(3), controlling whether we'll store secondary particles in permanent structure JKINE for further tracking (and producing hits). Default for whole detector is 0.75 GeV/c. For emulsion now it is set to 0.003 GeV/c.

= we have another momentum cut (run card TARG(3)), applied when filling emulsion output bank EMUL (MCEV daughter). Only hits of particles with momentum greater than its value are transferred to output. This cut now is set to 0. This cut allows us, for example, to track particles down to very low momentum, but give their "traces" to the output only when they are energetic enough.

Utilisation of the first option gives ~47% less output file: 150 kB per tau0mu event.

4. TEST of the code

I ran very small (50 tau0mu events) pilot production on CSF.

Production was done with BEAM 7 option (realistic vertex positioning) and THRE(1) = 1. (store hits only in vertex stack).

Some statistics about the vertex position.
Out of 50 events :

```
 - 12 (24%) outside emulsion volume EMPA (no emulsion hits are stored);
 - 11 (22%) in EMPA (hits are stored) but not in emulsion sheet;
            (There are honeycomb supports and fiber trackers there.);
 - 27 (52%) in emulsion sheets. (The only useful events.)
```

Everything looks normal. Events are processable by CHORAL (I used chorintr to look at them). Execution time was 234 s per event. Standard GEANT histograms with run statistics have been filled in this run. They are attached as ps file (eficass.hist.ps) to this report. You can see there histograms with:

= execution time per event (mean value 234 s)
= used space in event division IXDIV (mean value 1.37x10**6 words)
= number of tracks per event(mean value 5.8x10**4)
= number of long life tracks per event (mean value 144)
= max stack size per event (mean value 746) .

5. SOURCES and EXECUTABLES

All changes in Fortran source files, makefile, run script and input card files are committed to official Eficass repository:

/afs/cern.ch/chorus/offl/rs_aix41/dev/src/cvsroot/eficass

Executables can be found under my public directory for RS_AIX4.1 in:

/afs/cern.ch/user/r/rtsenov/public/eficass/bin/rs_aix41/

for HP_UX10.20 in:

/afs/cern.ch/user/r/rtsenov/public/eficass/bin/hp_ux102/

6. CONCLUSION

Some optimisation of the EFICASS code has been achieved, notably in memory demand and utilisation.

The most urgent thing now is reduction of CPU time consumption. This requires revision of calorimeter simulation and fiber trackers digitisation.

Other lines of improvements have been identified also:

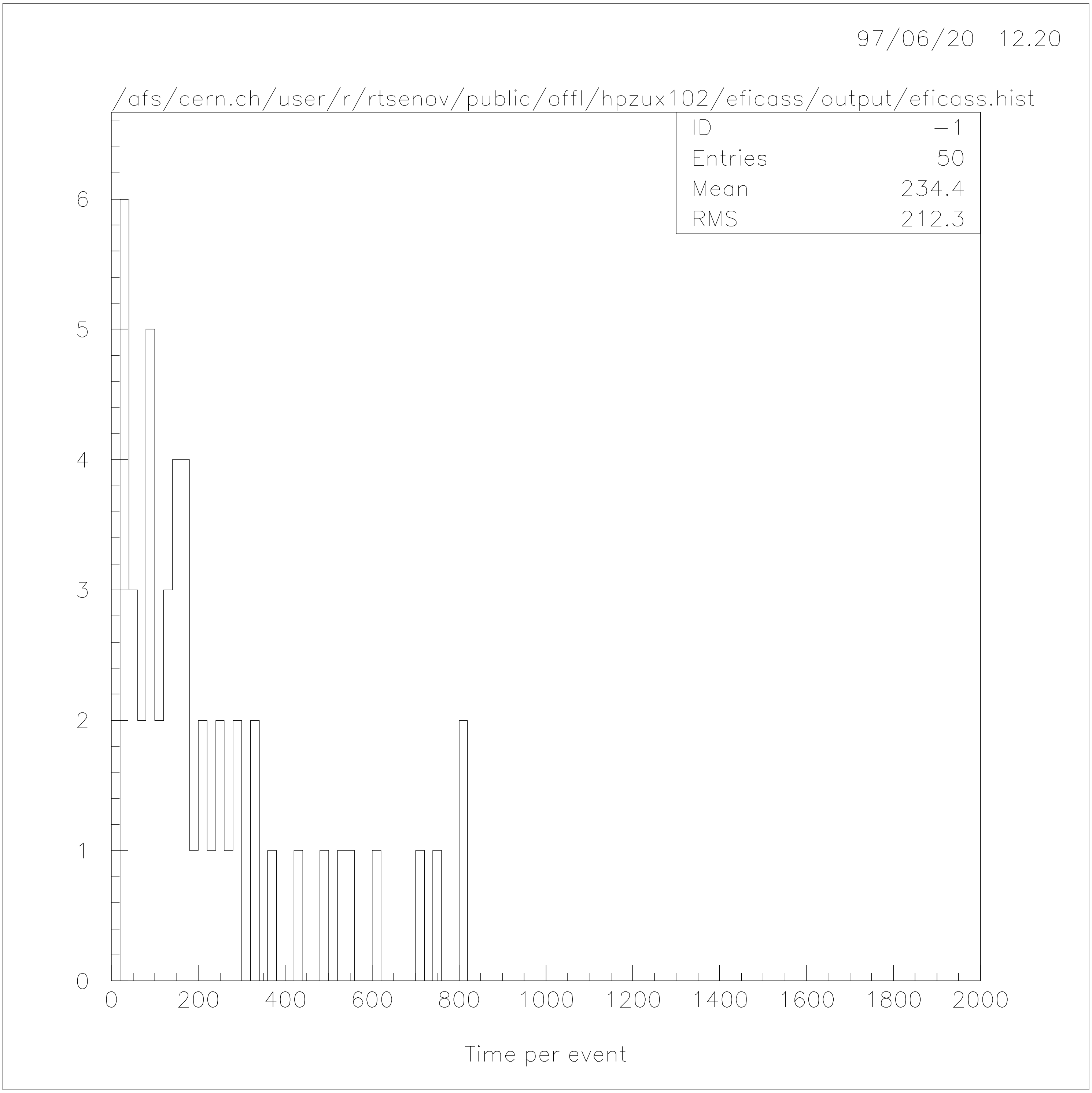
= rewrite of GUSTEP routine aiming at speed optimisation (It is called 2 million times per event...);

= rewrite GUKINE routine and implement "geantino" approach to realistic vertex positioning

= put in OPEN and CLOSE statements explicit file names, thus making the code independent of the (UNIX) platform on which it is run.

New release of the code, based on CVS, should appear very soon.

```
+------------------------------+---------------------------------------+
|  E-MAIL:                     |  Prof. Roumen TZENOV                  |
|  R.Tzenov@cern.ch            |  CERN/PPE (phone: +41227674589)       |
|  tsenov@phys.uni-sofia.bg    |           (FAX:   +41227673100        |
|                              |  University of Sofia, Fac.of Physics  |
|                              |            through s/b: +35926256410  |
|                              |       TEL: direct:      +3592622546   |
|                              |            FAX:         +35929625276  |
+------------------------------+---------------------------------------+
```


97/06/20 12.20
/afs/cern.ch/user/r/rtsenov/public/offl/hpzux102/eficass/output/eficass.hist
ID −1
Entries 50
Mean 234.4
RMS 212.3
0
1
2
3
4
5
6
0
200
400
600
800
1000
1200
1400
1600
1800
2000
Time per event