ICRC'07 Mérida, México

ICRC 2007 Proceedings - Pre-Conference Edition

A New Parallel Processing Scheme Enabling Full Monte Carlo EAS Simulation in the GZK Energy Region

K.KASAHARA
Research Inst. for Science and Engineering, Waseda Univ., Tokyo, Japan
kasahara@icrr.u-tokyo.ac.jp

Abstract: We developed a new distributed-parallel processing method enabling full M.C EAS simulation (say, with minimum energy of 500 keV) without using thin sampling even at 10^{19} eV. At higher energies, say, 10^{20} eV, this method is further extended so that we can get essentially full M.C result

Introduction

The thin sampling method[1] is widely used for EAS simulations at very high energies such as at 10^{17}eV or more. As far as the integrated total number of particles at a given depth is concerned, rather large thinning parameter can be safely used. However, the method put a weight on a particle so that if the weight becomes large, we observe a huge number of particles with the same properties (say, energy, position, direction, arrival time etc). This is not a desirable feature of the thin sampling method and we have to be careful about the thinning parameter.

Another method for enabling M.C simulation of high energy EAS is to use distributed-parallel processing. Normally, distributed-parallel processing needs a specific software and programs must be organized to match with such system. During the computation such a scheme also requires complex communications among many computer hosts.

In contrast to these methods, our distributed-parallel scheme dose not require any communication during computation; communication is needed only at the beginning and end of the simulation of an event. The method is compatible with the thin sampling, so the user can use both the methods, if needed. The method has been implemented in the Cosmos code[2].

Method

"skeleton-flesh"

The Cosmos code has the "skeleton-flesh" method. It has been implemented long time[4].

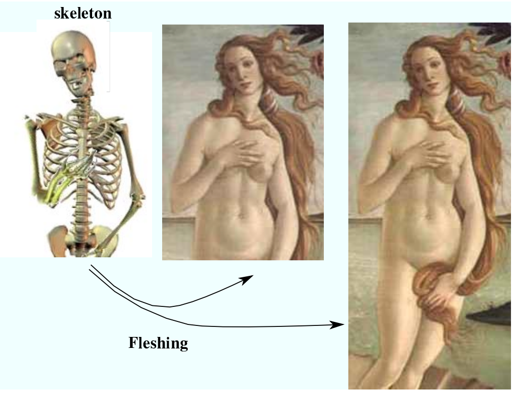

Figure 1: Conceptual "Skeleton-flesh" method: the right one is a fleshed image which even has a part non-existent in the skeleton.

The original "skeleton-flesh" method was born to support the following case. The user first generate air showers with very high minimum energy (say, 1TeV). These are called skeletons. If a skeleton satisfy a "trigger condition" (it is supposed to be small fraction of the skeletons), the user can flesh it, i.e, regenerate the air shower with very low minimum energy, say, 100 keV, while keeping the skeleton part unchanged. The method was applied by the Tibet ASγ experiment (For example, [3]). The skeleton could be fleshed to a deeper depths where there is no skeleton at all (Fig.1).

For the method to work, one may think that we may remember the initial random number seed for each event and with that seed we may start regeneration of the event, and if the concerned energy is lower than the skeleton making time, we may use a different random number generator. This actually works if the event generation depends only on the random number seed but not on the history (i.e, whether the event is first event or 10th event to be generated). Unfortunately, this is not the case for many interaction codes.

Therefore, the current method memorizes all the particles below E_{min} generated at various depths. If a particle has energy larger than E_{min} at generation, we follow its cascading until every energy becomes lower than E_{min}. We also have to memorize particles of energy larger than E_{min}, if they cross an observation level. In this method, we may follow particles with energy $< E_{min}$ at fleshing. A skeleton consists of such low energy particles and observed high energy particles.

“skeleton-smash-flesh-assemble”

The present method extends this “skeleton-flesh” method; we first creates a skeleton of a shower, and smashes it into n sub-skeletons and distributes each sub-skeleton to n cpu to flesh them. After all sub-skeletons are completely fleshed, they are assembled to make a complete picture of the shower. Thus, during the computation time we need no complex communication.

The sub-skeletons can be adjusted so that the total energy and the number of particles in each sub-skeleton are almost the same (say, energy sum is the same up to 7 digits or more and the number of particles are within difference of 1.

For example, a skeleton is made from a proton primary of 10^{20} eV with $Emin = 2 \times 10^{15}$eV; then the number of particles in the skeleton was 1534303, that of “observed” ones was 35542. The skeleton was smashed into 999 sub-skeletons; The sub-skeleton number, the sum of energy and the number of particles in it are listed below for the first 5 and last 5 sub-skeletons.

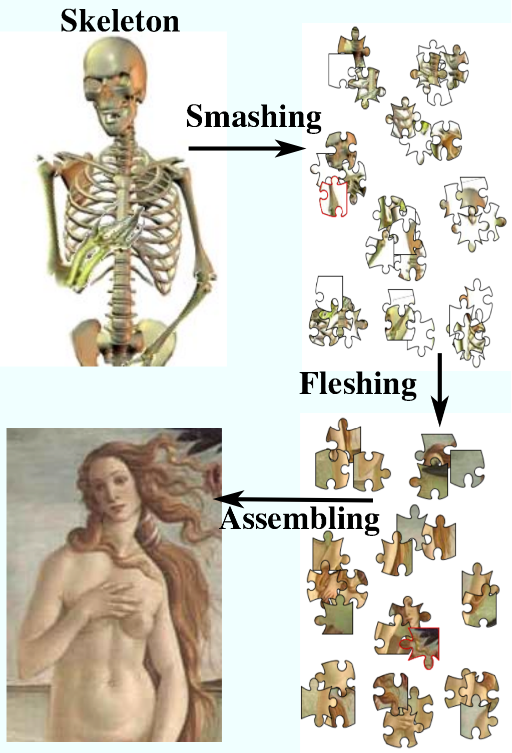

Figure 2: “Skeleton-smash-flesh-assemble” method

The “cpuPW” in the list shows the relative cpu power of each host where sub-skeleton is to be fleshed. If some of the cpu is much faster than others, we can assign a larger number than 1, then the host is alloted a larger sub-skeleton.

```
cpu#  cpuPW  Sum E       # of ptcls
   1  1.0   0.9827795E+08  1535
   2  1.0   0.9827795E+08  1536
   3  1.0   0.9827795E+08  1536
   4  1.0   0.9827795E+08  1536
   5  1.0   0.9827795E+08  1535
 ...
 995  1.0   0.9827795E+08  1536
 996  1.0   0.9827795E+08  1536
 997  1.0   0.9827795E+08  1536
 998  1.0   0.9827795E+08  1536
 999  1.0   0.9827795E+08  1535
```

Smashing algorithm

To be able to get almost the same sub-skeletons (otherwise, fleshing at each host cannot ends almost the same time and we have to wait until overburdened host finishes the task), we need a some algorithm for how to smash a skeleton into n peaces.

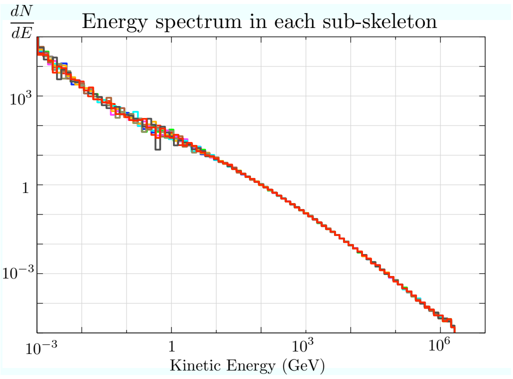

Figure 3: Energy spectrum of 10 random sub-skeletons among 999. They are almost identical

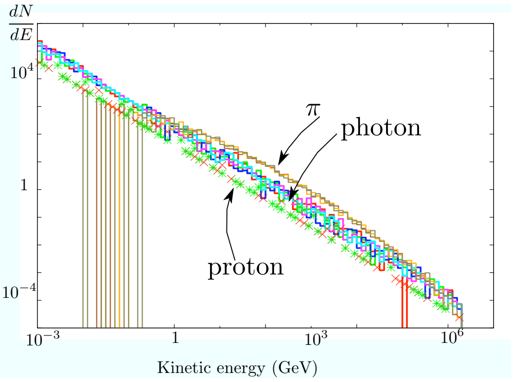

Figure 4: Energy spectra of photons, pions and protons of 2 to 5 random sub-skeletons among 999.

We first sort low energy particles in the skeleton by energy, and put first n particles into n sub-skeleton. Then, find the lowest energy sum skeleton (at the first step, this is the $n - th$ skeleton, although others have almost the same energy), and add to it one particle with the highest energy among the remaining ones. Then, we find the lowest sum energy sub-skeleton and repeat the above process.

This simple algorithm works fine and we can get almost the identical sub-skeletons as shown above. This is also seen in the energy spectrum of skeleton particles as in Fig.3. The similarity holds even if we see the spectra for each component. As an example, we show them for photons, pions and protons in Fig.4

Figure 5 also tells the similarity of sub-skeletons.

With $n = 50$, a 10^{19} eV shower can be simulated in $\sim$ 10 days.

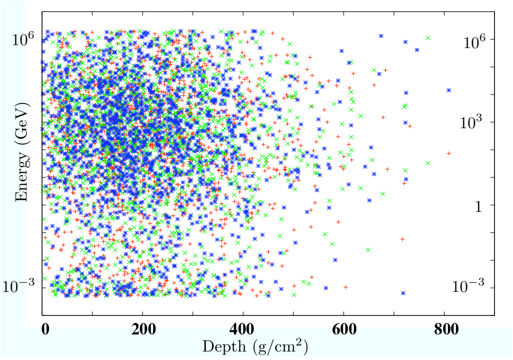

Figure 5: Depth vs energy of skeleton particles: 3 random sub-skeletons are super-imposed.

How to use?

The full Monte-Carlo for 10^{19} eV takes still long time and one may suspect that it is useless because EAS development fluctuates so much event by event so that we must generate at least order of few hundred events. Another point is that at such high energies, the number of particles at an observation level exceeds even 10^{11} and cannot record all of them.

First, the large number means that randomly chosen subset of particles (if the number is moderately large) well reflects the mother set (it is similar to the mother) so that one may record moderate number of particles and use them for, say, detector response simulations. It is important to note that any distribution (such as energy, angle, arrival time etc) in the random subset is quite smooth and has no danger of "spike" like structure which could result from the thin sampling method.

As to the fluctuation, the number of particles at a given depth actually fluctuates much, but if we look into the particle distributions (such as energy, angle .. etc or their correlation) at the same "age", they are remarkably similar. Suppose we have one or few Full M.C showers, and a number of fully fluctuated showers generated by more conventional way (this is actually safely possible with thin sampling as long as the total number of particles is concerned). The latter may contain only the total number at various depths (no each particle information) (called LDD: Longitudinal Development Data).

Then, we can get any particle distributions at a given depth in a shower selected from LDD by looking the same age point in the Full M.C shower.

For a 10^{20} eV shower, we need order of 1000 cpu's to finish its full M.C simulation. This is practically impossible. However, we already know that every sub-skeleton is almost identical so that we may randomly sample a fraction of $n-$peaces (say, $m = 100$ for $n = 1000$), and safely reconstruct whole picture of the shower. In this case, we don't put weight, $n/m = 10$, on individual particles, but we increase the random selection probability by n/m times.

Summary

The present scheme is implemented in the Cosmos code and applied to simulate EAS in the GZK region for the TA project. The details are given in an accompanying paper[5].

References

[1] A.M,Hillas, Proc. IX-th ISVHECI, Nucl. Phys.B(Proc. Suppl.) 52B(1997)139.

[2] K.Kasahara, see `http://cosmos.n.kanagawa-u.ac.jp/`.

[3] Amenomori et al., Phys. Rev. D 62, 112002 (2000) pp.13.

[4] K. Kasahara, Proc. 24th ICRC(Rome) 1 (1995) 399.

[5] K.Kasahara and F.Cohen et al, this conference ID200.