Proceedings of the 30th International Cosmic Ray Conference
Rogelio Caballero, Juan Carlos D'Olivo, Gustavo Medina-Tanco,
Lukas Nellen, Federico A. Sánchez, José F. Valdés-Galicia (eds.)
Universidad Nacional Autónoma de México,
Mexico City, Mexico, 2008

Vol. 4 (HE part 1), pages 581–584

30TH INTERNATIONAL COSMIC RAY CONFERENCE

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

K.KASAHARA AND F.COHEN
Research Inst. for Science and Engineering, Waseda Univ., Tokyo, Japan
Institute for Cosmic Ray Research, Univ. of Tokyo, Chiba, 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

A full Monte Carlo simulation of an air shower by a 10^{17}eV proton needs $\sim$ 1 week/cpu. The number becomes $\sim$2 years for 10^{19} eV and $\sim$20 years for 10^{20} eV. The GZK full M.C is hopeless. To overcome this situation, the thin sampling method[1] is widely used for EAS simulations at very high energies. 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 and we have to be careful about the thinning parameter.

Another method for enabling M.C simulation of high energy EAS would be 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, resulting in saturation of speed-up.

In contrast to these methods, our distributed-parallel scheme does not require any communication during computation; it 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[3].

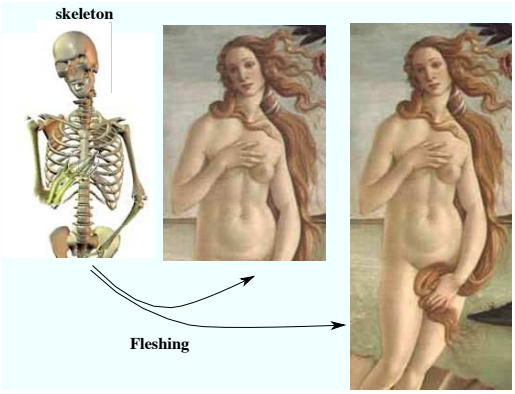

Figure 1: Conceptual "Skeleton-flesh" method: the right one is a fleshed image which even has a part non-existent in the skeleton.

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 skele-

ton 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, [4]). 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.

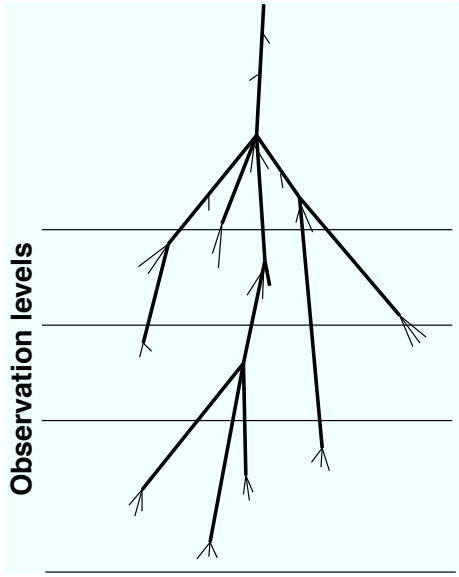

Figure 2: More realistic skeleton ingredients: Thick lines are high energy particle tracks made at skeleton making time. Short thin lines represent low energy particles generated at interactions (say, knock-on, multiple production, bremsstrahlung, pair-creation, dE/dx loss etc), and not tracked at the time of skeleton making. Their information is memorized for fleshing

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 (Fig.2).

Extended "skeleton-flesh" for distributed-parallel processing

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 extended skeleton-flesh method consists of "skeleton-smash-flesh-assemble".

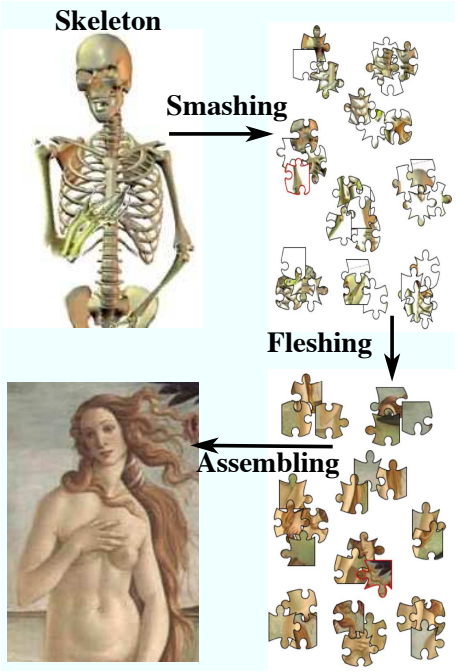

Figure 3: "Skeleton-smash-flesh-assemble"

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 $E_{min} = 2\times10^{15}$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.

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 more sub-skeletons.

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.

We first sort particles in the skeleton by energy, and put first n particles into n sub-skeletons. 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.

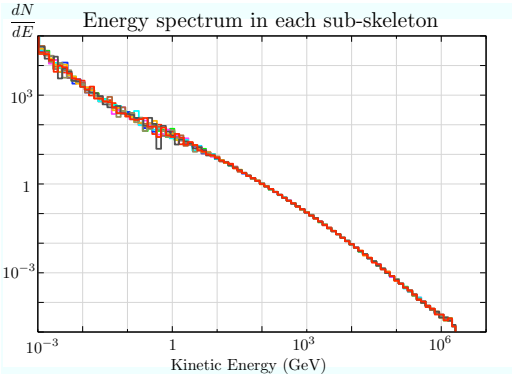

Figure 4: Energy spectrum of 10 random sub-skeletons among 999. They are almost identical

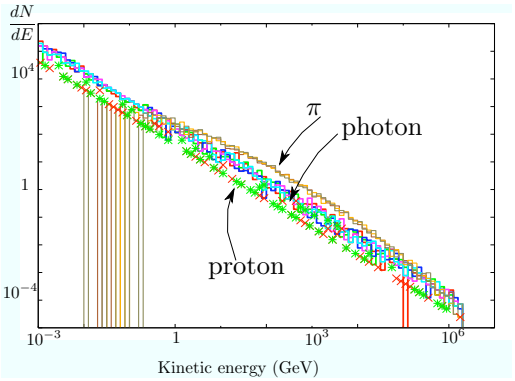

Figure 5: Energy spectra of photons, pions and protons of 2 to 5 random sub-skeletons among 999.

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.4. The similarity holds even if we see the spectra of ingredient component separately . As an example, we show them for photons, pions and protons in Fig.5

Figure 6 also tells the similarity of sub-skeletons.

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

Rescue method

If we use a number of pc's, some of them may go down when all other pc's are finishing the job. Then, do we need another 10 days for recovering ? In such a case, we smash the skeletons of those

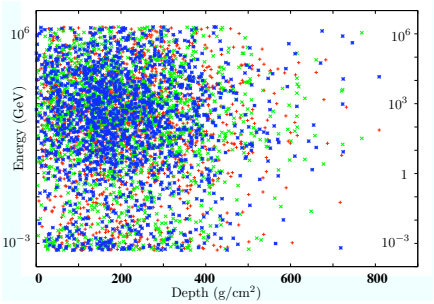

Figure 6: Depth vs energy of skeleton particles: 3 random sub-skeletons are super-imposed.

crashed hosts and flesh them; we need only 1 additional day for that.

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 data (FDD). Actually, there is no depth with exactly the same "age": however, many properties of particles changes very slowly with depth, we can use the closest FDD depth safely. Some, say, arrival time, must be corrected by seeing the difference of detector height. For muons and hadrons, "age" correspondence between LDD and FDD is not as good as for electrons and photons. This feature is overcome by using the correspondence by cog depth (depth measured by the center of gravity of the transition curve).

For a 10^{20} eV shower, we need order of $n = 1000$ 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, take $m = 100$ 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 (though, we use this weight for counting the total number of 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].

This work is partly supported by Grant-in-Aid for Scientific Research on Priority Areas (Highest Cosmic Rays:15077205)

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] K. Kasahara, Proc. 24th ICRC(Rome) 1 (1995) 399.
[4] Amenomori et al., Phys. Rev. D 62, 112002 (2000) pp.13.
[5] F.Cohen and K.Kasahara et al, this conference ID200.