KEK-CP-050
KEK Preprint 96-134
October 1996
H/D

su 97/4

PVM-GRACE *

Fukuko Yuasa [†]

CERN, 1211 Geneva 23, Switzerland

and

Denis Perret-Gallix

*LAPP, Laboratoire d'Annecy-le-Vieux de Physique des Particules
74941 Annecy-le-Vieux CEDEX, France*

and

Setsuya Kawabata, Tadashi Ishikawa

*KEK, National Laboratory for High Energy Physics, 1-1 OHO
Tsukuba City, Ibaraki 305, Japan*

Abstract

We have implemented PVM in GRACE: an automatic Feynman diagram computation package, to benefit from the available CPU power distributed over a cluster of workstations. It is clearly shown that PVM-GRACE gives excellent performance in term of reduction of elapsed time when the network overhead is kept small as compared to the processing time.

*To be published in the *Proc. of the Fifth International Workshop on Software Engineering, Artificial Intelligence and Expert Systems for High Energy and Nuclear Physics, Lausanne, Switzerland, Sep. 2-6, 1996.*

[†]On leave of absence from KEK, National Laboratory for High Energy Physics, Japan

National Laboratory for High Energy Physics, 1996

KEK Reports are available from:

Technical Information & Library
National Laboratory for High Energy Physics
1-1 Oho, Tsukuba-shi
Ibaraki-ken, 305
JAPAN

Phone: 0298-64-5136
Telex: 3652-534 (Domestic)
(0)3652-534 (International)
Fax: 0298-64-4604
Cable: KEK OHO
E-mail: Library@kekvox.kek.jp (Internet Address)
Internet: <http://www.kek.jp>

1. Introduction

We present a PVM¹ based automatic Feynman diagrams calculation system, PVM-GRACE. GRACE² is a powerful software package for calculating the cross section of high energy physics process and for generating the simulating events.

Using the Feynman diagrams technique, all observables can be computed for any process at any order of the perturbative theory. However, the calculation becomes rapidly tedious and involved as the number of final state particle increases. Automatic computation systems have been developed to cover the increasing need for such calculations. The MINAMI-TATEYA group has developed the GRACE package which generates automatically the FORTRAN code corresponding to the matrix element of the process. Integration over the multi-dimensional phase space leads to cross-section and event generation. The larger the center of mass energy and the number of external state particles, the larger the CPU time involved.

The basic idea behind the PVM-GRACE project is to distribute the amplitude calculation over several PVM slaves spawned by the PVM master. This simultaneous calculation of the total amplitude reduces the total elapsed time of GRACE drastically as well as the required memory allocation on each workstation.

2. PVM-GRACE

GRACE consists of four subsystems, the graph generation, the source generation, the numerical integration, and the event generation subsystems. When the initial and final states of the process, the physics model, and the kinematics are set by input data-cards, GRACE generates the complete set of Feynman diagrams and the matrix elements FORTRAN code. During the numerical integration, a multi-dimensional grid mapping the integration phase space is produced where the hyper-cell dimensions are adapted to the behavior of the differential cross section. It contains the event generation probabilities. Events are then generated according to this grid. GRACE uses the program package BASES/SPRING³ for multi-dimensional numerical integration and event generation. Of all steps, the numerical integration is the most

CPU time consuming part.

Generated FORTRAN subroutines are grouped into three categories, the master part, the utility part, and the amplitude calculation part. The amplitude calculation contains as many subroutines as the number of Feynman diagrams plus utility subroutines. In the master part, the amplitude calculation is spawned over the available CPU's. After the calculation is finished, the results of each CPU are sent back to the master, where the numerical value of the cross section is estimated.

The sample points are represented by `real*8` variables and the results are stored in an array of `complex*16`. Since the amount of data to be transferred between tasks affects deeply the performance of PVM-GRACE, we have tested the following data transfer approaches:

- Each sample point selected by the master task is broadcast to the slave processes in charge of the amplitude computation, then each result is returned to the master:
- A full set of phase space points are sent to the various slave processes and an array of results is returned.

The merit of the first approach is that the basic version of BASES can be used. But the problem is the large number of transfer operations needed.

In the second case the number of the data transfer is much reduced. But the problem, here, is that BASES must be modified so as to generate sets of appropriate phase space points. For a network such as Ethernet or FDDI, the second approach shows better performance than the first one. Therefore, in this paper we will concentrate on the second approach.

3. Performance Measurement

We have tested PVM-GRACE on two types of CERN systems. One is the workstation cluster, HPPLUS with nine HP 9000/735s connected by FDDI. The version of PVM is 3.3.10. The other is the Meiko CS-2 machine, CERNCS2, a massive parallel computer where modules are linked by a proprietary network. The version of PVM is CSPVM based on PVM 3.2.

3.1. Measurement Conditions

The performance measurements of PVM-GRACE have been made for four conditions:

- *Case 1* : $e^+e^- \rightarrow WW\gamma$, 18 Feynman diagrams in unitary gauge with 768 sample points,
- *Case 2* : $e^+e^- \rightarrow e^-\bar{\nu}_e u \bar{d}$, 88 Feynman diagrams in covariant gauge with 2048 sample points,
- *Case 3* : $e^+e^- \rightarrow e^-\bar{\nu}_e u \bar{d}$, 28 Feynman diagrams in unitary gauge with 39366 sample points,
- *Case 4* : $e^+e^- \rightarrow e^-\bar{\nu}_e u \bar{d}$, 88 Feynman diagrams in covariant gauge with 32768 sample points,

On the HP cluster, due to the limited available CPU time, only *Case 1* and *2* were measured. The detailed measurement conditions are shown in Table. 1. Here, N_{slave} is the number of the PVM slave processes, each running on a different CPU. $N_{slave} = 0$ is the non-PVM program running on 1 CPU.

The number of the Feynman diagrams is fixed by the physics process and its gauge. The size of the matrix elements is directly related to the number of Feynman diagrams, so is the processing time. The number of the sample points needed to reach the required integration accuracy depends on how singular the integrand function is. For a fixed data size per transfer (`nsiz`), the number of sample points will set the number of data transfer.

The amount of the byte per transfer is roughly 1.5MB (*Case 1*) and 3MB (*Case 2 ~ 4*) when we choose `nsiz` = 1024 in *Case 1* and 2048 in *Case 2 ~ 4* respectively. The number of the data transfer is also presented in Table 1.

3.2. The Elapsed Time Measurement

The performance curves and the measured elapsed time on the HP cluster are shown in Fig. 1 and Table 2. In Fig. 2, 3, and 4, the results on the Meiko CS-2 are shown. Also in Table 3, 4, and 5, the elapsed times are presented. The performance curves are represented

Fig. 1. Reduction ratio vs. # of slaves (N_{slave}) for Case 1 ~ 2 on HP.

Fig. 2. Reduction ratio vs. # of slaves (N_{slave}) for Case 1 ~ 2 on Meiko.

Fig. 3. Reduction ratio vs. # of slaves (N_{slave}) for Case 3 on Meiko.

Fig. 4. Reduction ratio vs. # of slaves (N_{slave}) for Case 4 on Meiko.

by the ratio = (elapsed time)/(elapsed time of $N_{slave} = 1$) versus the number of slaves.

In Fig. 1, the elapsed time is the smallest for $N_{slave} = 4, 5$ for Case 1 being reduced to 42%. For Case 2, the elapsed time is the shortest for N_{slave} over 9, less than 20%. In Fig. 2 ~ 4, the elapsed time is the shortest for all the cases with N_{slave} over 9. The ratios become 26%, 19%, and 14% respectively.

In all the figures, the performance curves saturate when the overhead due to the data transfer compensates the time reduction obtained by the addition of a new CPU. In Fig. 1, for Case 1, the saturation appears faster than Case 2 as the processing time in each CPU is shorter. On the Meiko CS-2, (Fig. 2 ~ 4,) the saturation point shifts gradually to larger slave number.

Table 1. PVM-GRACE conditions.

	<i>Case 1</i>	<i>Case 2</i>	<i>Case 3</i>	<i>Case 4</i>	comments
nsiz	1024	2048	2048	2048	
ncall	768	39366	2048	32768	# of sample points
N_{group}	1	20	1	16	# of the data transfer
N_{slave}	1 ~ 9	1 ~ 9, 10	1 ~ 10, 11	1 ~ 13	# of slave processes
igraph	18	28	88	88	# of Feynman diagrams

Table 2. The elapsed time and the ratio for *Case 1* and *2* on HP cluster (sec)

<i>Case</i>	0	1	2	3	4	5	6	7	8	9
1	138.61 0.92	150.29 1.0	91.51 0.61	72.72 0.48	62.94 0.42	63.68 0.42	64.38 0.43	68.40 0.46	69.31 0.46	73.52 0.49
2	654.98 0.99	663.91 1.0	360.99 0.54	261.23 0.39	200.79 0.30	172.83 0.26	151.55 0.23	137.09 0.21	129.94 0.20	124.41 0.19

Table 3. The elapsed time and the ratio for *Case 1* and *2* on Meiko (sec)

<i>Case</i>	0	1	2	3	4	5	6	7	8	9	10	11
1	163.42 0.96	171 1.0	93 0.54	67 0.39	58 0.34	50 0.29	48 0.28	55 0.32	50 0.29	45 0.26		
2	452 1.17	386 1.0	205 0.53	149 0.39	116 0.30	105 0.27	93 0.24	83 0.22	79 0.20	81 0.21	74 0.19	77 0.20

Table 4. The elapsed time and the ratio for *Case 3* on Meiko (sec)

0	1	2	3	4	5	6	7	8	9
27863.26 1.03	27109 1.0	14123 0.52	10040 0.37	7836 0.29	6643 0.25	6491 0.24	5505 0.20	5245 0.19	5215 0.19

Table 5. The elapsed time and the ratio for *Case 4* on Meiko (sec)

0	1	2	3	4	5	6	7	8	9	10	11	12	13
6846 1.13	6033 1.0	3106 0.51	2220 0.37	1667 0.28	1437 0.24	1252 0.21	1101 0.18	1084 0.18	1047 0.17	940 0.16	894 0.15	847 0.14	890 0.15

4. Discussions

This behavior can be understood as follows:

$$\frac{T_{process}}{N_{slave}} + N_{slave} \cdot T_{network} + \alpha = T_{PVM-GRACE} \quad (1)$$

where $T_{process}$ is the total processing time and $T_{network}$, the transfer time for returning results to the master process. α is the overhead constant. Equation (1) shows $T_{PVM-GRACE}$ has a minimum at $N_{slave} = \sqrt{\frac{T_{process}}{T_{network}}}$. Since $T_{process}$ is fixed for each physics process, N_{slave}^{opt} is depending on the characteristics of the physics process. Therefore, we have to choose the adequate number of N_{slave} for each physics process so as to make PVM-GRACE most effective.

The data was fitted by using Equation (1). We obtained $T_{process}=134$ sec, $T_{network}=5.2$ sec, and $\alpha=11.5$ sec. Using these values, $N_{slave}^{opt} = 5$.

The increase of the elapsed time for $N_{slave} = 1$ in Fig. 1 and *Case 1* of Fig. 2 is due to the overhead of PVM (packing/unpacking data, sending/receiving data between tasks and network waiting time). In Fig. 2 ~ 4, for the *Case 2 ~ 4*, the elapsed times for $N_{slave} = 0$ are the largest. This is thought to be due to an increase of the memory cache misses for large program size.

5. Conclusion

We have implemented PVM in GRACE successfully by modifying the integration program BASES. The performance of PVM-GRACE for four cases have been measured on two systems and we have got the following results:

- PVM-GRACE provides an efficient way to reduce the elapsed time by using idle CPU cycle on a workstation cluster. For very long program and fast turnover, massive parallel systems like the Meiko CS-2 are preferred.
- The time reduction is strongly depending on the ratio, $R = \frac{T_{process}}{T_{network}}$, and therefore on the physics process.
- The best reduction time corresponds to an optimum $N_{slave}^{opt} = \sqrt{R}$.
- However, to avoid network saturation, one should select a configuration with $N_{slave} \leq N_{slave}^{opt}$.

- PVM-GRACE becomes more powerful when the physics process becomes more complex and its processing time longer.

Since the results obtained with PVM-GRACE are encouraging, we have added a special option in the code generation of GRACE to generate the FORTRAN code for PVM automatically.

Acknowledgments

One of us (F. Y.) would like to acknowledge to F. Gagliardi, E. McIntosh and B. Panzer-Steindel for encouragements and valuable discussions. We are also grateful to express our sincere gratitude to Professors Y. Watase, Y. Shimizu and T. Kaneko for the encouragement.

References

1. Al Geist et al., *PVM3 user's guide and reference manual*. ORNL/TM-12187.
2. MINAMI-TATEYA group, *GRACE manual*. **KEK Report 92-19**. T.Kaneko, A Feynman-graph generator for any order of coupling constants. *Computer Physics Communication* **92** (1995) 127-152.
3. S.Kawabata, A new Monte Carlo event generator for High Energy Physics. *Computer Physics Communication* **41** (1986) 127-153.
S.Kawabata, A New Version of the Multi-dimensional Integration and Event Generation Package BASES/SPRING. *Computer Physics Communication* **88** (1995) 309-326.