Recent Developments in Parallelization of the Multidimensional Integration Package DICE

F.Yuasa/KEK
K.Tobimatsu/Kogakuin Univ.
S.Kawabata/KEK

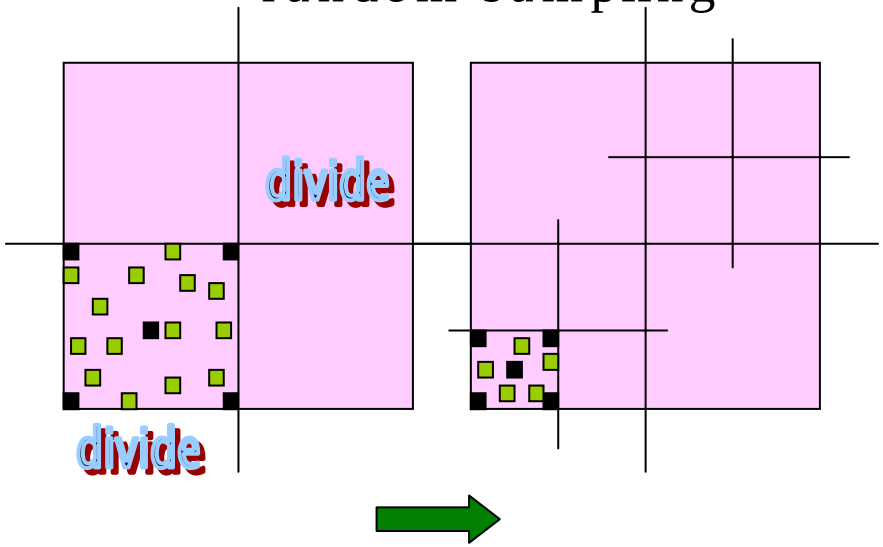
ACAT 2005
May 22 - 27, 2005 at DESY, Zeuthen

DICE

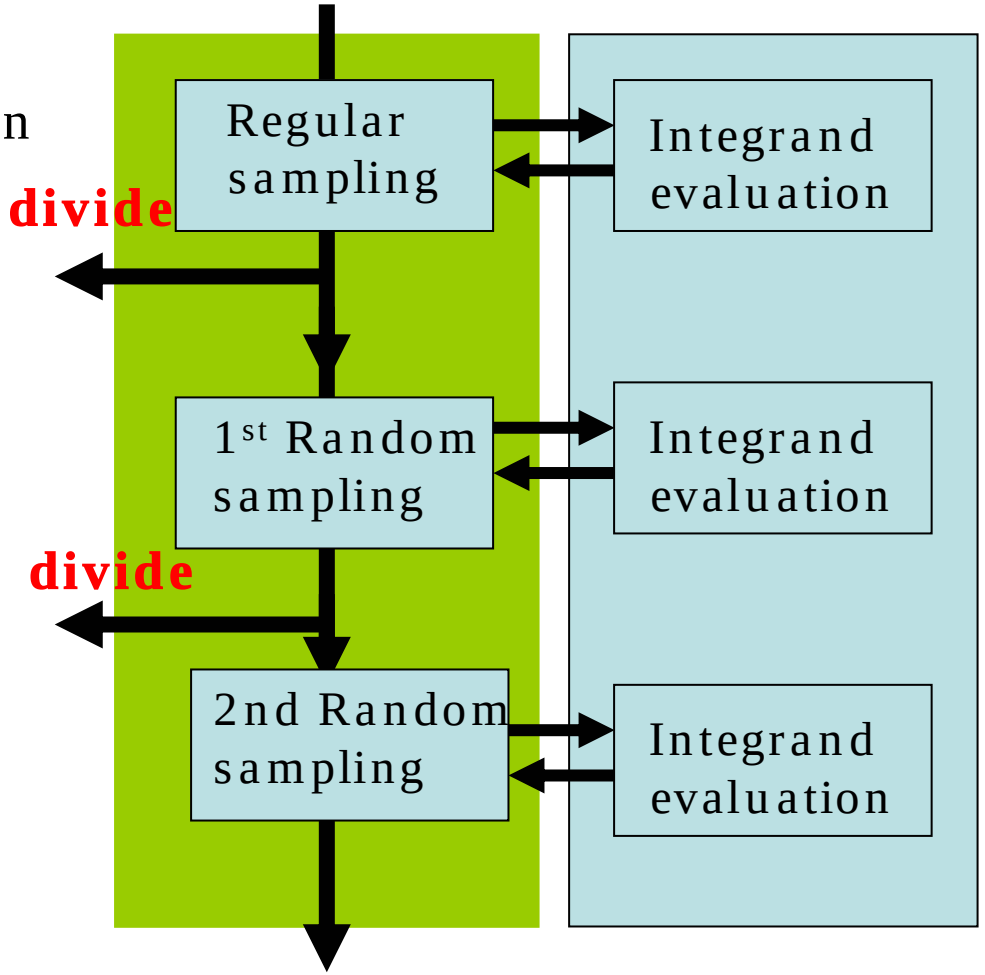
- Developed by K.Tobimatsu and S.Kawabata
- First Version of DICE in 1992
  - Research Reports of Kogakuin Univ. No.72 ('92)
- Vectorized version of DICE (DICE 1.3Vh) in 1998
  - Research Reports of Kogakuin Univ. No.85 ('98)

Algorithm of DICE

- Simple algorithm
  - Divide the integral region into 2^{Ndim} hypercubes
  - Two kinds of sampling method
  - Regular sampling and random sampling



Level = 2 **Level = 3**



Characteristics of DICE

- DICE is strong in handling steep singularities
- DICE (1.3Vh) was vectorized to reduce the wall-clock time
- DICE 1.3Vh has the concept as "virtual worker" to handle each hypercube

Why Parallelization ?

- We want to reduce the wall-clock time further.
  - Vector Processor Architecture disappeared.
  - Parallel Processor Architecture is common.
    - Distributed memory
    - Shared memory
- We want to calculate larger physics problems.
- We want to save money.
  - PC cluster with Parallel Library such as MPI or with OpenMP

Profile of DICE

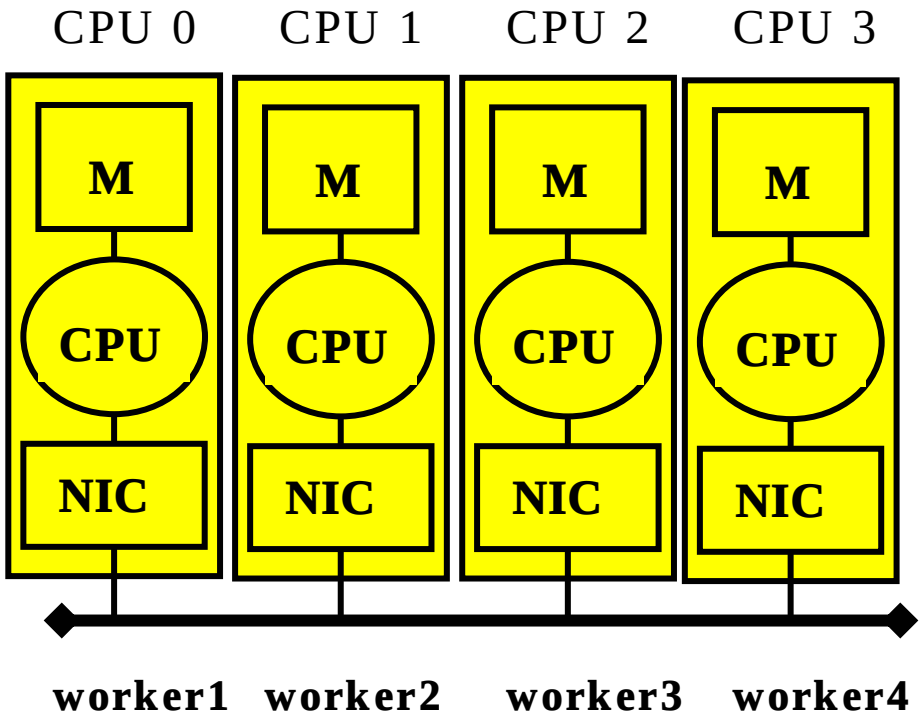
"gprof" output : Flat profile of non-parallelized DICE

Time [%]	Cumulative time [sec]	Self time [sec]	calls	Self [ms/call]	total	Name of subroutine
82.95	7.60	7.60	26214	0.29	0.29	elwks_
12.41	8.73	1.14	26214	0.04	0.33	func_
2.52	8.96	0.23	3072	0.08	0.08	vbrndm_
0.93	9.05	0.08	1536	0.06	2.80	randm2_
0.92	9.13	0.08	1536	0.05	2.79	randm1_
0.13	9.14	0.01	1638	0.01	0.34	regular_

Total CPU time required was 9.16 sec
with expected error = 10%

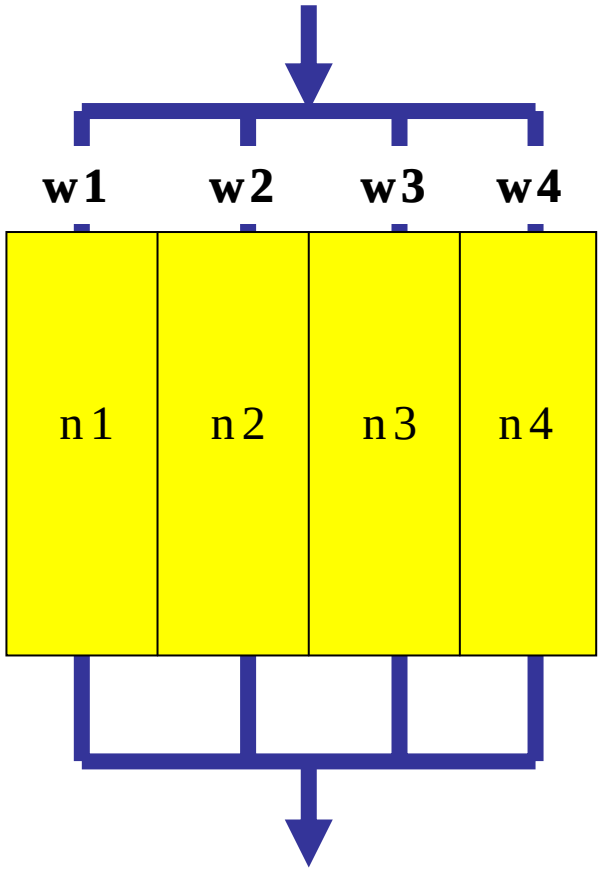
Parallelization of DICE

- MPI
  - Message Passing Interface
- SPMD
  - Single Program Multiple Data
  - Independent processes
  - Distributed Memory Space
  - Data transferred among processors



Parallelization of DICE in 2002

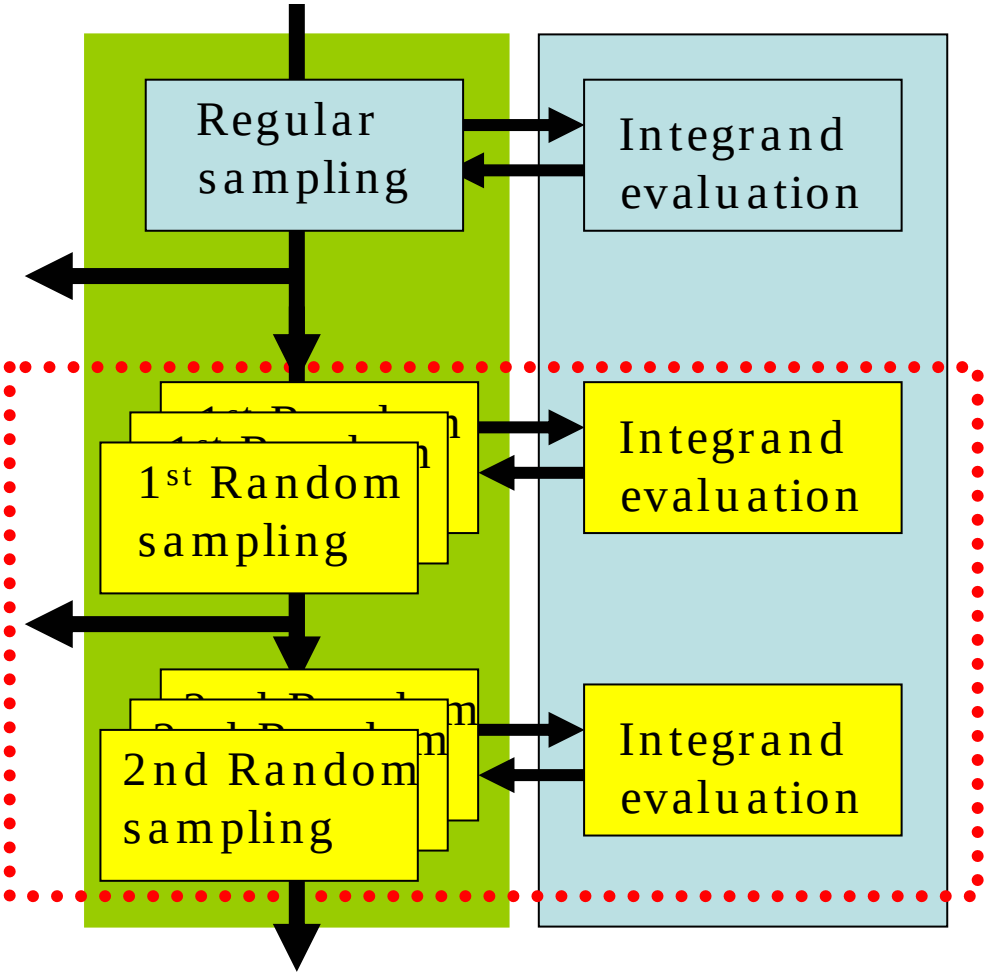
- Random sampling points are distributed into workers
- Efficiency of parallelization is very good
- Data Parallelism method
- It was presented at ACAT 2002 at MOSCOW



Nsample = n1 + n2 + n3+ n4

Parallelization of DICE in 2005

- Hypercubes are distributed into processors (workers).
- The results are gathered to the root process (worker 1).
- The root process scattered the results to all processors.



Efficiency Measurement

- Environment : PC cluster
  - CPU : Xeon dual 3.06GHz, 8 systems
  - Memory : 2GB
  - Switch : cheap 10/100 /1000 switch
  - Compiler : /usr/local/mpich-intel81/bin/mpif77
  - MPI Bandwidth : 56.79 MB/s in average (by 10 times measurement) by using MPI send-receive. Data size = 1MB

Efficiency Measurement using Physics Problem

$$e^+e^- \to \mu\mu\gamma$$

$$W = 70GeV, K_c = 100MeV$$

- Naïve kinematics
  - The Integrand has strong singularities.
- # of dimensions is 4
- # of random sampling points handled by each worker is 100
- Maximum # of workers is 8

Efficiency of the parallelization with expected error = 2%

Xeon 3.06GHz, non-parallelized DICE,
Required CPU time = 8704.84 sec

processors (workers)	CPU time [sec]	elapsed time [sec]	speedup: CPU time	speedup : elapsed time
1	8882.03	8894	1.00	1.00
2	5179.62	6178	0.58	0.69
4	3308.92	5011	0.37	0.56
8	2394.32	4863	0.27	0.55

Efficiency of the parallelization with expected error = 1%

Xeon 3.06GHz, non-parallelized DICE,
Required CPU time = 2 /3 : 4 : 44.34 (183884.34 sec)

processors (Workers)	CPU time [sec]	elapsed time [sec]	speedup: CPU time	speedup: elapsed time
2	109401.92	134377	0.59	0.73
4	69676.01	108234	0.38	0.59
8	51056.55	103126	0.28	0.56

Ref. VPP500 1PE (1.6 GFLOPS), vectorized DICE,
Required CPU time = 1 /20 : 56 : 26.21 (161786.21 sec)
(measured in 1998)

Summary

- We parallelized DICE by MPI
- The hypercubes are distributed to workers
- Efficiency measurement has been done with naïve kinematics for $e^+e^- \to \mu\mu\gamma$
- The wall-clock time was reduced
- We believe that further more reduction of the wall-clock time will be possible with applying the load balancing algorithm