ATLAS Internal Note
SOFT-NO-018
29 March 1995

GEANT / DICE Profile

R.Yaari and S.Jarp

Purpose

Give a flavour of

the execution profile of

CERN's physics programs

through a recent case study (N)

(also relevant for
High Energy Physics in general)

By:
Sverre Jarp
+ Rafi Yaari

Agenda

- What kinds of programs
- Aleph's move to RISC
- Some background information
- Claimed performance
- Actual performance
- What were the problems ?
- Atlas's Dice profiling
- Conclusion

Overview

Three kinds of jobs:

Simulation (Galeph)

Simulate a physics detector

and the data collection

(based on physics processes)

Reconstruction (Julia)

Recreate meaningful data
(energy, position, track, velocity, etc.)

from the binary information

coming from real/simulated detector

Analysis (Alpha)

Analyse meaningful data

from a selected set of events

looking for correlations

Execution profiles

High Energy Jobs

Performance follows SPECint

not SPECfp,

although there are many FP calculations

Lots of if/then/else logic

Often referred to as 'Mega-IF' code

If (such a particle)

If (that angle)

If (inside such a volume).

Aleph

One of the four LEP groups:

Used to be on mainframes: IBM, VAX, Cray

Very heavy demand for computing resources,

so moved to RISC systems:

DEC, HP, IBM, SGI

Thousands of SPECints

(always wanting more and more)

Issue:

Aren't CERN's internal benchmarks
(and the SPEC suite) 'overselling' RISC ?

Aleph claimed:

' We only see half the performance you promise ! '

Background information

Programming language

FORTRAN (and C)

Desire to move to C++

Heavy use of pre-compiled libraries

CERN-wide (CERNLIB)

Mathlib, Kernlib, Graphlib, etc.

Private to Aleph

In all cases:

Correctness outweighs speed

use: OPT - LEVEL = 2

Claimed performance

Based on internal benchmark that roughly follows SPECint, but stresses the cache less than Aleph's jobs.

	IBM/9000/ 900	DEC/3000 /400	HP/735 /99	IBM/RS6K/ 350	SGI/Challenge
Frequency	-	133	99	41.6	150
SPECint	-	74.7	109	35.4	90
CERN units	20	18	27	9.8	20
Ratio	-	4.1	4.0	3.6	4.5

Peak performance

Latest machines from vendors:

	IBM/9000/ 900	DEC/3000/ 900	HP/735/ 125	IBM/RS6K/ 590	SGI/Challenge 200
Frequency	-	275	125	66.6	200
SPECint	-	189	136	121.4	119
CERN units	20	52	34.1	27.0	29
Ratio	-	3.9	4.0	4.5	4.1

Even with such top end machines, does Aleph get 'full' yield ?

Started with Galeph

461 events	CERNVM	DEC/3K/400	HP/99	IBM/350	SGI/150
Run_1 (s)	6622	10263	9977	15852	12057
Perf_.ratio	1.0	0.65	0.66	0.42	0.57
Norm.ratio	1.0	0.71	0.49	0.86	0.57

Big surprise:

Surprisingly low on HP

Faster machines are often worse !

(CERN unit numbers improve more than real-life jobs)

461 events	CERNVM	DEC/3K/900	HP/125	IBM/590	SGI/200
Run_1 (s)	6622	4190	7936	6398	8612
Perf_.ratio	1.0	1.58	0.66	1.04	0.77
Norm.ratio	1.0	0.63	0.49	0.76	0.53

Results with profiler

	DEC	HP	SGI
RNDM	16.6	31	15.6
IUHUNT	20.3	14.6	20.0

Percentage time in routine

On HP, two routines consumed almost 50%

In RNDM (random number generation)

$$divI and $$mulI consumed high portions

Also: traced **2.8 million calls** per event !

In IUHUNT,

time was spent (inefficiently)
looking for a variable in an array

26K calls per event !

RNDM analysis

Algorithm is:

```
Do 100  I = 1,LEN
      K = ISEED1/53668
      ISEED1 = 40014 * (ISEED1 - K*53668) - K * 12211
      IF (ISEED1 .LT. 0) ISEED1 = ISEED1 + 2147483563
      K = ISEED2/52774
      ISEED1 = 40692 * (ISEED1 - K*52774) - K * 3791
      IF (ISEED1 .LT. 0) ISEED1 = ISEED1 + 2147483399
      IZ = ISEED1 - ISEED2
      IF (IZ .LT. 0) IZ = IZ + 2147483562
      RVEC(I) = IZ * 4.6566128E-10
 100 Continue
```

	CERNVM	DEC	HP	IBM	SGI
RNDM/int	3.12	6.78	7.24	2.13	4.60
RNDM/dp	3.82	4.90	3.84	5.77	4.70
Ranlux	1.80	2.30	1.26	1.54	1.09

Ranlux

Somewhat similar to RNDM (but FP)

Initialisation:

```
Do 25   I = 1,24
    TwoM24 = TwoM24 * 0.5    [TwoM24 = 1.0 initially]
    K = JSEED/53668
    JSEED = 40014 * (JSEED - K*53668) - K * 12211
    IF (JSEED .LT. 0) JSEED = JSEED + 2147483563
    ISEEDS(I) = MOD(JSEED,ITWO24)
25 Continue
```

Real routine:

```
Do 100 IVEC = 1,LEN
    UNI = SEEDS(J24) - SEEDS(I24) - CARRY
    IF (UNI .LT. 0.) Then
        UNI = UNI +1.0
        CARRY = TwoM24
    Else
        CARRY = 0
    Endif
    Seeds(I24) = Uni
    I24 = Next(I24)
    J24 = Next(J24)
    Rvec(Ivec) = Uni
100 Continue
```

IUHUNT

Certainly a 'stupid' program, but unrolling may help:

	CERNVM	DEC/3400	HP/99	IBM/350	SGI/150
Not unrolled	28.7	16.0	30.9	37.4	36.4
Unrolled	10.7	15.6	15.3	29.8	16.7
Ratio	2.68	1.02	2.02	1.25	2.18

Clear difference between various systems !

Back to GALEPH

Improvements w/new Ranlux and new IUHUNT:

461 events	IBM/9000	DEC/3400	HP/99	IBM/350	SGI/150
Run_1 (s)	6622	10263	9977	15852	12057
Run_2 (s)	5832	9770	7024	13782	10751
Ratio (1)/(2)	1.14	1.12	1.42	1.15	1.12
Run_3 (s)	5659	8873	5321	13681	8883
Ratio (2)/(3)	1.03	1.10	1.32	1.01	1.21
Perf_.ratio (3)	1.0	0.64	1.06	0.41	0.64
Norm_ratio:	1.0	0.71	0.79	0.84	0.64

HP/99 now about 80 % of 'promised' performance

Absolute timing: DEC/3900 3620 secs (is fastest)

Julia

Short glimpse of second program

Reconstruction:

461 events	CERNVM	DEC/3400	HP/99	IBM/350	SGI/150
Run_1 (s)	1167	2227	1042	1972	1260
Perf_.ratio	1.0	0.52	1.12	0.59	0.93
Norm.ratio	1.0	0.57	0.83	1.20	0.93

DEC is surprisingly low !