Time as a Microcosm: Was the Year 2000 Crisis Inevitable?

IEEE Computer Society
Los Alamos and Northern New Mexico Chapter
Thursday, February 23, 2006

Robert Gezelter Software Consultant
35 – 20 167th Street, Suite 215
Flushing, New York 11358 – 1731
United States of America

+1 (718) 463 1079
gezelter@rlgsc.com
http://www.rlgsc.com

The Year 2000 Crisis

- The costliest programming error in history
- A threat to the computerized world on many levels
- Fantastically expensive – likely trillions and trillions of dollars
- It was certainly real
- But, was it inevitable?

Was YYMMDD ever the correct choice?

- Commonly proffered reason: Space
- YYMMDD is 6 bytes; YYYYMMDD is 8 bytes
- Space – in memory and on disk is critical

Really?

- Dates invariably represented as character strings
- If space was the concern, why were dates almost never represented as packed decimal
- Packed decimal would have reduced space by approximately 50% for little computation
- YYYYMMDD can be represented as an integer for even greater savings

“The Rule of 48'

- “All Scientists are Blind”
  – Crichton, “The Andromeda Strain”
- Just because a fact is commonly believed does not make it correct
- It was never efficient – in time OR space

Where was YYMMDD needed?

- Electro-mechanical Accounting Machines
- On unformatted direct printouts of files
- in short: binaphobia

The costs of YYMMDDHHmmss... (and YYYYMMDDHHmmss...)

```
hh = hh + 1;
if  (hh > 100){hh = hh - 100; ss = ss + 1;}
if  (ss > 60){ss = ss - 60; mm = mm + 1;}
if  (mm > 60){mm = mm - 60; HH = HH + 1;}
if  (HH > 24){HH = HH - 24; DD = DD + 1;}
if  (DD > EndMonth(MM, YY))
    {DD = DD - EndMonth(MM, YY); MM = MM + 1;}
if  (MM > 12){MM = MM - 12; YY = YY + 1;}
```

A Note on Efficiency

```
while !EndFile(input){
     readline(input, buffer)
     ... SOME COMPUTATION ...
     writeline(output, buffer)
     }
```

- How many instructions are really executed?
- Is the computation significant?

The Costs (cont’d)

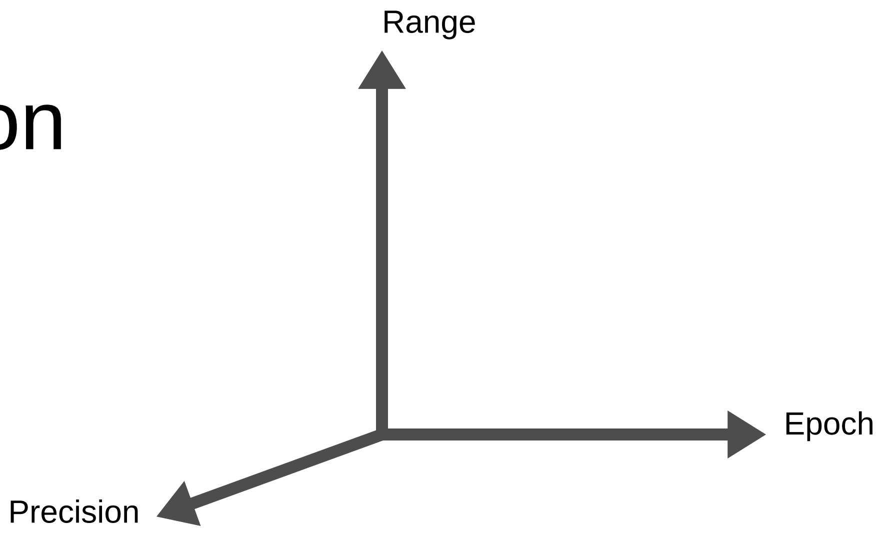
- polymodulus arithmetic
- computationally expensive
- break even is very low

Was this ever noticed?

- Yes and No
- Some OS Architects did notice
- Others did not
- The applications community appears to have not noticed

An Objective Evaluation Criteria

- ok, my nomenclature (Gezelter, 2004)
- each dimension is independent
- every time representation has othogonal choices
- freedom of substitution



Real Examples:

- *NIX
- MS-DOS
- Excel/LOTUS 1-2-3/?
- VAX/VMS (Now OpenVMS)
- z/OS
- Windows32

*NIX

- Epochal Date: January 1, 1970
- Precision: 1 second
- Range: originally 32 bits

MS-DOS

- Epochal Date: January 1, 1980
- Precision: 1 second

Excel/LOTUS 1-2-3/?

- Epochal Date:
  January 2, 1904/January 1, 1900
- Precision: fractions of a day
- Range: up to December 9999

VAX-VMS (Now OpenVMS)

- Epochal Date: November 17, 1858 (Smithsonian base time)
- Precision: 100 microsecond
- Range: 63 bits; well past 9999

IBM z/OS

- Epochal Date: January 1, 1900
- Precision: 1 microsecond
- Range: 64/128 bits (originally September 17, 2042)

Windows32

- Epochal Date: January 1, 1601
- Precision: 100 ns
- Range: 64 bits

A note on Range Limits

- the O/S supplied TOY values are extendible
- while they differ, they are all 2-complement
- they can be extended easily to higher precision and range
- compatible with their existing formats and support

Applications Usage

- most applications “rolled their own”
- even on O/S where the system had support
- this was the true Year 2000 crisis
- it was not fixed
- remediation consisted of minimal fixes

Year 2000 was IT architecture in a microcosm

- the long term damage caused by myopic choices
- efficiency claim a rationalization
- format choices are VERY long lived
- once systems are built to an interface, changes are expensive in effort, and schedule

The "Snowball" Effect

- Good begets better!
- Bad just gets worse –
  "The gift that keeps on giving. PAIN!"

Conclusions

- Year 2000 was not inevitable
- Could (Should?) have been a non-issue
- Beware Format/Representation rationalizations

Questions?

Robert Gezelter Software Consultant
35 – 20 167th Street, Suite 215
Flushing, New York 11358 – 1731
United States of America

+1 (718) 463 1079
gezelter@rlgsc.com
http://www.rlgsc.com

Session Notes & Materials:
http://www.rlgsc.com/ieee/LosAlamos/2006-02/index.html