

Re: Occurs Depending Memory Use

Source: <http://coding.derkeiler.com/Archive/Cobol/comp.lang.cobol/2005-06/msg00065.html>

- *From:* Scott Peterson <scottp4.removethistoreply@xxxxxxxxxxxxxxxx>
 - *Date:* Thu, 02 Jun 2005 14:22:26 -0700
-

"Chuck Stevens" <charles.stevens@xxxxxxxx> wrote:

>If a processor is running full-bore on a given construct, I would expect the
>amount of processor time consumed by that construct to be very close to
>equal to the *elapsed* time for the construct. There's nothing except
>influences *external* to the construct to interrupt it.

The way that IBM handles job accounting, CPU will almost always be
fare less than elapsed time. As I remember, UNIX reports it
differently as you don't really have job steps in the sense that IBM
big iron does so it's an apple/oranges comparison.

>
>> By definition SEARCH ALL is a keyed search as opposed to a sequential
>> search it should be faster in many cases.
>
>No, not exactly. Both SEARCH and SEARCH ALL operate on tables that have an
>OCCURS ... INDEXED BY. See below for your comment on "keyed" vs.
>"sequential".

But SEARCH ALL requires the INDEXED option, SEARCH does not.

>> The larger the table, the
>> more difference there should be.
>
>Presuming that SEARCH ALL *is* a nonserial (binary or otherwise) search. It
>is *often* true, but not *necessarily* true.

So what's the problem? I said from the first exchange that this
assumes fairly random distribution and even provided an example where
you it would be better not to use SEARCH ALL. But if you can use it,
the differences should be substantial on large tables.

>
>> I don't know you. How familiar are you with IBM COBOL
>> implementations?
>
>It's been a long, long time since I've been directly involved in IBM COBOL;
>I have been involved with maintenance and development of Unisys MCP-based

Re: Occurs Depending Memory Use

>COBOL implementations conforming to the '68, '74 and '85 instantiations of
>the corresponding international standards. Part of that process has
>required that I familiarize myself with what the *language standards*
>actually say.

>
{snip}

I used to have a lot of that junk memorized too. Pity that most of it
has nothing to do with how IBM implemented this feature.

>
>Goody for them. Their compiler conforms to the standard. So does one that
>does a serial search for both.

My answer dealt very specifically with IBM COBOL? Comments about what
other compilers do is just so much irrelevant noise. Why do you keep
bringing returning to this point when it just confuses things? .

>As to what COBOL *in general* is supposed, or required, to do, as distinct
>from what IBM thinks it ought to do for their customers, I stand on the
>distinction in the wording in ANSI X3.23–1985 between page VI–123,

OK....and I paid \$2.27 for gas this morning which is equally relevant
to the discussion

All you're saying is that the IBM compiler is compliant and does
use a non-linear (binary) search in the SEARCH ALL which we already
knew.

>
>As an active participant in both INCITS/J4 and ISO/IEC JTC1/SC22/WG4, in
>terms of what I expect COBOL to do, what individual implementations do is of
>far less importance to me than what users should expect of standard COBOL
>*in general*. That goes for Unisys MCP COBOL implementations, with which
>I'm fairly intimately familiar, just as much as it does for IBM COBOL
>implementations, to which my last significant exposure was some thirty-five
>years ago. .

>

Since the original comment was on IBM mainframe COBOL what difference
does how another vendors UNIX implementation have to do with
anything?

>The *COBOL* verb SEARCH ALL is *not* required by any COBOL *standard* to do
>a binary search (or for that matter a combination of a binary and a serial
>search). The authors of the standard have carefully worded the requirements
>so that this is the case; in fact, the standard doesn't say anything at all
>about what sort of search algorithm should be used; it addresses *results*,
>not *methods*.

Re: Occurs Depending Memory Use

OK...and having said that, IBM has defined and published the method they chose to implement this....and have going back at least as far as COBOL F in the 1960's.

>I can also observe from experience that the authors of the
>COBOL standard are also more concerned about *accuracy* than *performance*,
>{snip}. Whether the improved performance of a
>strictly-binary SEARCH ALL is worth the extra performance is a matter for
>discussion between the *implementor* and the *implementor's customers*.
>
,

....OK, and the question was settled to IBM and their customers viewpoint about 3 generations of compilers back, and about 3 generations of programmers too.

>> I think you're getting far more generalized than I am. My responses
>> were strictly aimed at my experiences with mainframe COBOL. With that
>> limit your comments are meaningless.
>

>Yes, I am. From several standpoints that's appropriate. I've been
>participating on the US and international COBOL standards committees, and as
>such I believe it is appropriate for me to take a broader view than the
>particular opinions of one vendor (including my employer!).

If you're speaking about compiler theory you're right. But this discussion was never about that. It was dealing with specific issues of IBM mainframe COBOL and, as such, the "wider" view just confuses things.

>And I've been maintaining "mainframe COBOL" compilers since September 1984.

If you want to push dates around, I first started as a systems programmer responsible for COBOL in 1978 after several years as a programmer.

>There are architectures that qualify as "mainframes" other than those with
>which you are familiar.

Really? That's a rather presumptuous statement. It would almost be amusing to have you list them since you know nothing about me.

>
>> As a developer or coder I should know my data well enough to make a
>> practical decision about which version of SEARCH or PERFORM is the
>> most appropriate. It has nothing to do with "wanting" or "expecting"
>> anything. Know your tools.
>

>I agree wholeheartedly. But that also implies that it's inappropriate to
>generalize your experiences with a particular implementation of COBOL to

Re: Occurs Depending Memory Use

>come to the conclusion that those experiences are applicable to COBOL in
>general. *Your* tools may not be *the only* tools, or even *the only
>appropriate* tools, and it's important also to know the difference.
>

Excuse me. I never tried to generalize this issue to other environments. You're the only one who keeps on doing this. I have repeatedly pointed that out to you if you bother to go back and look.

Scott Peterson

--

Rome did not create a great empire
by having meetings...they did it by
killing all those who opposed them.

417/612

.

- *Follow-Ups:*

- ◆ *Re: Occurs Depending Memory Use*
 ◇ *From: Richard*

- *References:*

- ◆ *Re: Occurs Depending Memory Use*
 ◇ *From: Scott Peterson*
- ◆ *Re: Occurs Depending Memory Use*
 ◇ *From: Chuck Stevens*

- Prev by Date: *Re: interesting use of NEXT SENTENCE vs. CONTINUE*
- Next by Date: *Re: Occurs Depending Memory Use*
- Previous by thread: *Re: Occurs Depending Memory Use*
- Next by thread: *Re: Occurs Depending Memory Use*
- Index(es):
 - ◆ *Date*
 - ◆ *Thread*