
Subject: Memory management by 5.4 on Sunblade
Posted by [weitkamp](#) on Mon, 17 Dec 2001 10:57:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

I'm mostly using IDL 5.4 on my laboratory's Linux86 cluster, and running memory-critical simulation code on it which uses lots of large temporary arrays.

Now they've bought a Sunblade as the first test machine of a future cluster, and so I tried running my code on that one, using 5.4 as before. That is,

```
IDL> print,!version  
{ sparc sunos unix 5.4 Sep 25 2000    64    64}
```

However, I noticed that memory management by IDL 5.4 on the Sunblade is extremely poor in that variable space "freed" by TEMPORARY, DELVAR, or simply by dynamic resizing of a variable is not actually freed but kept allocated (so tells me "top").

I wonder if this bug will persist with 5.5, which hasn't yet been installed on any machine here. Has anybody else made any experience with 5.4 or 5.5 on Sunblade in this context?

Timm

Subject: Re: Memory management by 5.4 on Sunblade
Posted by [John-David T. Smith](#) on Wed, 19 Dec 2001 22:41:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

Timm Weitkamp wrote:

```
>  
> Hi,  
>  
> I'm mostly using IDL 5.4 on my laboratory's Linux86 cluster, and  
> running memory-critical simulation code on it which uses lots of large  
> temporary arrays.  
>  
> Now they've bought a Sunblade as the first test machine of a future  
> cluster, and so I tried running my code on that one, using 5.4 as  
> before. That is,  
>  
> IDL> print,!version  
> { sparc sunos unix 5.4 Sep 25 2000    64    64}  
>
```

> However, I noticed that memory management by IDL 5.4 on the Sunblade
> is extremely poor in that variable space "freed" by TEMPORARY, DELVAR,
> or simply by dynamic resizing of a variable is not actually freed but
> kept allocated (so tells me "top").
>
> I wonder if this bug will persist with 5.5, which hasn't yet been
> installed on any machine here. Has anybody else made any experience
> with 5.4 or 5.5 on Sunblade in this context?

It's not a *bug*, it's a *feature*. IDL allocates memory as necessary from the OS, and then, even if it doesn't need it any more, hangs onto it just in case. This is true I think on all platforms, and all recent versions of IDL. You still have the memory available, just not to the system as a whole.

Example:

```
IDL> help,/memory
heap memory used:  370549, max:  372383, gets:  297,
frees:  92
IDL> a=fltarr(256,256,256)
IDL> help,/memory
heap memory used:  67479528, max:  67479547, gets:  301,
frees:  93
IDL> a=0
IDL> help,/memory
heap memory used:  370617, max:  67479545, gets:  304,
frees:  95
```

So you see, the giant array used up 64MB or so. IDL allocated the memory for it, at which point 67479528 bytes of heap memory are used. When you free that variable, only 370617 bytes are used, but 67479545 bytes are still allocated. So the memory is available, just not to any other program.

JD

Subject: Re: Memory management by 5.4 on Sunblade
Posted by [Craig Markwardt](#) on Thu, 20 Dec 2001 05:50:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

JD Smith <jdsmith@astro.cornell.edu> writes:

> Timm Weitkamp wrote:
>>
>> Hi,
>>

>> I'm mostly using IDL 5.4 on my laboratory's Linux86 cluster, and
>> running memory-critical simulation code on it which uses lots of large
>> temporary arrays.
>>
...
>> However, I noticed that memory management by IDL 5.4 on the Sunblade
>> is extremely poor in that variable space "freed" by TEMPORARY, DELVAR,
>> or simply by dynamic resizing of a variable is not actually freed but
>> kept allocated (so tells me "top").
...
>
> It's not a *bug*, it's a *feature*. IDL allocates memory as necessary
> from the OS, and then, even if it doesn't need it any more, hangs onto
> it just in case. This is true I think on all platforms, and all recent
> versions of IDL. You still have the memory available, just not to the
> system as a whole.

Hi JD--

I do not think this is always true. I find that I regularly create
300 MB arrays in memory, and then free them. While the procedure is
running, the memory usage is indeed around 300 MB, but afterwards the
memory use, as reported by the external program "top", drops down
again to the quiescent level.

If I recall correctly, there was a bally-hoo in one of the What's News
from ages ago about IDL using an allocator that is able to release
memory back to the system. *However*, this is surely (a) extremely
system dependent; and (b) not always possible depending on the
fragmentation of the memory at the time. Perhaps this is what Timm is
running in to.

Good luck Timm!
Craig

--

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response

Subject: Re: Memory management by 5.4 on Sunblade
Posted by [Nigel Wade](#) on Thu, 20 Dec 2001 09:45:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

Timm Weitkamp wrote:

> Hi,
>
> I'm mostly using IDL 5.4 on my laboratory's Linux86 cluster, and
> running memory-critical simulation code on it which uses lots of large
> temporary arrays.
>
> Now they've bought a Sunblade as the first test machine of a future
> cluster, and so I tried running my code on that one, using 5.4 as
> before. That is,
>
> IDL> print,!version
> { sparc sunos unix 5.4 Sep 25 2000 64 64}
>
> However, I noticed that memory management by IDL 5.4 on the Sunblade
> is extremely poor in that variable space "freed" by TEMPORARY, DELVAR,
> or simply by dynamic resizing of a variable is not actually freed but
> kept allocated (so tells me "top").
>
> I wonder if this bug will persist with 5.5, which hasn't yet been
> installed on any machine here. Has anybody else made any experience
> with 5.4 or 5.5 on Sunblade in this context?
>
> Timm

I don't think it's IDL, but the underlying OS.

AFAIK IDL uses the system memory management provided by malloc/free. On Linux free() returns memory back to the OS when it can. Under Solaris this is not the case, so the application retains the virtual memory pages.

With good virtual memory management this is rarely important. Physical memory is allocated on a page-by-page basis as needed by applications. Provided you have sufficient swap space to hold the virtual memory in use by all applications, and each application has sufficient limits to allow the necessary virtual memory there shouldn't be a problem. If an application holds on to a page of virtual memory which it no longer requires it only takes up space on the swap disk not in physical memory. There will be some overhead as this page is swapped out to disk, but that only occurs if some other application requires the physical memory.

--

Nigel Wade, System Administrator, Space Plasma Physics Group,
University of Leicester, Leicester, LE1 7RH, UK
E-mail : nmw@ion.le.ac.uk
Phone : +44 (0)116 2523568, Fax : +44 (0)116 2523555

Subject: Re: Memory management by 5.4 on Sunblade
Posted by [weitkamp](#) on Thu, 20 Dec 2001 10:36:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

Craig Markwardt <craigmnet@cow.physics.wisc.edu> wrote in message
news:<onn10etp7t.fsf@cow.physics.wisc.edu>...
> JD Smith <jdsmith@astro.cornell.edu> writes:
>> Timm Weitkamp wrote:
>>>
>>> Hi,
>>>
>>> I'm mostly using IDL 5.4 on my laboratory's Linux86 cluster, and
>>> running memory-critical simulation code on it which uses lots of large
>>> temporary arrays.
>>>
> ...
>>> However, I noticed that memory management by IDL 5.4 on the Sunblade
>>> is extremely poor in that variable space "freed" by TEMPORARY, DELVAR,
>>> or simply by dynamic resizing of a variable is not actually freed but
>>> kept allocated (so tells me "top").
> ...
>>
>> It's not a *bug*, it's a *feature*. IDL allocates memory as necessary
>> from the OS, and then, even if it doesn't need it any more, hangs onto
>> it just in case. This is true I think on all platforms, and all recent
>> versions of IDL. You still have the memory available, just not to the
>> system as a whole.
>
> Hi JD--
>
> I do not think this is always true. I find that I regularly create
> 300 MB arrays in memory, and then free them. While the procedure is
> running, the memory usage is indeed around 300 MB, but afterwards the
> memory use, as reported by the external program "top", drops down
> again to the quiescent level.
>
> [...]

This is pretty much exactly the kind of thing my programs do too. On
x86 Linux all unused memory is given back to the system, on
Sunblade/Solaris it isn't. You may call it a feature, but I think it
is one that may cause big problems. JD, what kind of platform are you
using that one shouldn't buy? :-)

Timm

Subject: Re: Memory management by 5.4 on Sunblade

Craig Markwardt wrote:

```
>
> JD Smith <jdsmith@astro.cornell.edu> writes:
>> Timm Weitkamp wrote:
>>>
>>> Hi,
>>>
>>> I'm mostly using IDL 5.4 on my laboratory's Linux86 cluster, and
>>> running memory-critical simulation code on it which uses lots of large
>>> temporary arrays.
>>>
> ...
>>> However, I noticed that memory management by IDL 5.4 on the Sunblade
>>> is extremely poor in that variable space "freed" by TEMPORARY, DELVAR,
>>> or simply by dynamic resizing of a variable is not actually freed but
>>> kept allocated (so tells me "top").
> ...
>>
>> It's not a *bug*, it's a *feature*. IDL allocates memory as necessary
>> from the OS, and then, even if it doesn't need it any more, hangs onto
>> it just in case. This is true I think on all platforms, and all recent
>> versions of IDL. You still have the memory available, just not to the
>> system as a whole.
>
> Hi JD--
>
> I do not think this is always true. I find that I regularly create
> 300 MB arrays in memory, and then free them. While the procedure is
> running, the memory usage is indeed around 300 MB, but afterwards the
> memory use, as reported by the external program "top", drops down
> again to the quiescent level.
>
```

Two strikes this week. Maybe I should stick to histograms. Indeed, so much has this mis-feature been touted, that I presumed it applied everywhere. Testing on my linux system reveals Craig to be entirely correct: even as far back as v5.2.1, memory is released to the OS after it's not needed. Yet another reason to avoid Solaris, I suppose.

Here's a fun command to use to track memory usage:

```
% watch -n 1 'free;echo;ps -C idl -o rss,vsize,cmd'
```

JD

Subject: Re: Memory management by 5.4 on Sunblade
Posted by [David Fanning](#) on Thu, 20 Dec 2001 16:57:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

JD Smith (jdsmith@astro.cornell.edu) writes:

> Two strikes this week. Maybe I should stick to histograms.

It happens. Geez, I thought seriously about retiring
a month or so ago when I had three strikes in the same
day. :-(

But the best hitters always swing for the fences. :-)

Cheers,

David

--

David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Memory management by 5.4 on Sunblade
Posted by [John-David T. Smith](#) on Thu, 20 Dec 2001 18:50:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

JD Smith wrote:

>

> Timm Weitkamp wrote:

>>

>> Hi,

>>

>> I'm mostly using IDL 5.4 on my laboratory's Linux86 cluster, and
>> running memory-critical simulation code on it which uses lots of large
>> temporary arrays.

>>

>> Now they've bought a Sunblade as the first test machine of a future
>> cluster, and so I tried running my code on that one, using 5.4 as
>> before. That is,

>>

>> IDL> print,!version

>> { sparc sunos unix 5.4 Sep 25 2000 64 64}

>>

>> However, I noticed that memory management by IDL 5.4 on the Sunblade

>> is extremely poor in that variable space "freed" by TEMPORARY, DELVAR,
>> or simply by dynamic resizing of a variable is not actually freed but
>> kept allocated (so tells me "top").
>>
>> I wonder if this bug will persist with 5.5, which hasn't yet been
>> installed on any machine here. Has anybody else made any experience
>> with 5.4 or 5.5 on Sunblade in this context?
>
> It's not a *bug*, it's a *feature*. IDL allocates memory as necessary
> from the OS, and then, even if it doesn't need it any more, hangs onto
> it just in case. This is true I think on all platforms, and all recent
> versions of IDL. You still have the memory available, just not to the
> system as a whole.

I should clarify my statement: it's a *bug* or *feature* (depending on your point of view) of the memory management sub-system of your OS, not of IDL. IDL wisely relies on the host OS for memory management, and concentrates on what it does best. One of the RSI developers made a good point regarding the differing behaviors of, e.g., Solaris vs. Linux memory management: it's a tradeoff between returning memory to the system (which can be costly), and speed. As is usual in these cases, you cannot have your cake and eat it too.

Good luck,

JD
