Subject: Re: array operations and memory consumption
Posted by don.woodraska on Fri, 04 Jan 2008 23:16:06 GMT
View Forum Message <> Reply to Message

On Jan 4, 3:19 pm, dktr....@gmail.com wrote:
> Hi all,
>
> Apologies in advance if this is old hat ... I've got a question
> regarding IDL's memory usage that can be boiled down in the following
> example:
>
> IDL> a = BINDGEN(100,100,100)
> IDL> baseMem = (MEMORY())[0]
> IDL> a = a + 1
> IDL> PRINT, (MEMORY())[3] - baseMem
>   2000049

Consider using the increment operator like this:

IDL> a++

This does an increment in-place with little additional memory (a few bytes). The assignment c-style operators like this one below, use double the memory.

IDL> a += 1

I haven't seen MEMORY before. Is this new in 7.0? I use help,/memory which doesn't behave like I expect.

Good luck,
Don

Subject: Re: array operations and memory consumption
Posted by Michael Galloy on Fri, 04 Jan 2008 23:16:59 GMT
View Forum Message <> Reply to Message

On Jan 4, 3:19 pm, dktr....@gmail.com wrote:
> Hi all,
>
> Apologies in advance if this is old hat ... I've got a question
> regarding IDL's memory usage that can be boiled down in the following
> example:
>
> IDL> a = BINDGEN(100,100,100)
> IDL> baseMem = (MEMORY())[0]
> IDL> a = a + 1

> IDL> PRINT, (MEMORY())[3] - baseMem
> 2000049
>
> I've tried modifying "a = a + 1" with various combinations of pointers
> and the TEMPORARY function, but can't reduce this temporary elevated
> memory consumption. Compare the above with a call to CONGRID:
>
> IDL> a = BINDGEN(100,100,100)
> IDL> baseMem = (MEMORY())[0]
> IDL> a = CONGRID(a, 100, 100, 100)
> IDL> PRINT, (MEMORY())[3] - baseMem
> 1003941
>
> I'm working with some very large image arrays and trying to do some
> "in place" manipulations. While I can slink by with the temporary
> memory usage of the latter CONGRID-type operations, addition (or
> multiplication) of an array by a scalar breaks the bank.
>
> Any tips or directions to a helpful tutorial on IDL's memory policies
> would be greatly appreciated. Thanks!

Be careful with type conversion here! In particular, 1 is a short integer (or long integer depending on if compile_opt defint32 is used), so a = a + 1 here converts a to type int. Consider the following:

```
IDL> a = bindgen(100, 100, 100)
IDL> orig_mem = memory(/current)
IDL> a = a + 1
IDL> print, memory(/highwater) - orig_mem
    2000082

IDL> a = bindgen(100, 100, 100)
IDL> orig_mem = memory(/current)
IDL> a = a + 1B
IDL> print, memory(/highwater) - orig_mem
    1000096

IDL> a = bindgen(100, 100, 100)
IDL> orig_mem = memory(/current)
IDL> a = temporary(a) + 1B
IDL> print, memory(/highwater) - orig_mem
        91
IDL>
IDL> a = bindgen(100, 100, 100)
IDL> orig_mem = memory(/current)
IDL> a += 1B
IDL> print, memory(/highwater) - orig_mem
```

```
        91

IDL>
IDL> a = bindgen(100, 100, 100)
IDL> orig_mem = memory(/current)
IDL> a++
IDL> print, memory(/highwater) - orig_mem
        91
```

Mike
--
www.michaelgalloy.com
Tech-X Corporation
Software Developer II

Subject: Re: array operations and memory consumption
Posted by dktr.ted on Fri, 04 Jan 2008 23:55:33 GMT
View Forum Message <> Reply to Message

Type conversion! That was what was missing from my understanding. This makes much more sense now. Thanks for the pointers. I also had forgotten the increment operator, although I was looking for something more general.

-Ted

On Jan 4, 3:16 pm, "mgal...@gmail.com" <mgal...@gmail.com> wrote:
> On Jan 4, 3:19 pm, dktr....@gmail.com wrote:
>
>
>
>> Hi all,
>
>> Apologies in advance if this is old hat ... I've got a question
>> regarding IDL's memory usage that can be boiled down in the following
>> example:
>
>> IDL> a = BINDGEN(100,100,100)
>> IDL> baseMem = (MEMORY())[0]
>> IDL> a = a + 1
>> IDL> PRINT, (MEMORY())[3] - baseMem
>>   2000049

>
>> I've tried modifying "a = a + 1" with various combinations of pointers
>> and the TEMPORARY function, but can't reduce this temporary elevated
>> memory consumption. Compare the above with a call to CONGRID:
>
>> IDL> a = BINDGEN(100,100,100)
>> IDL> baseMem = (MEMORY())[0]
>> IDL> a = CONGRID(a, 100, 100, 100)
>> IDL> PRINT, (MEMORY())[3] - baseMem
>> 1003941
>
>> I'm working with some very large image arrays and trying to do some
>> "in place" manipulations. While I can slink by with the temporary
>> memory usage of the latter CONGRID-type operations, addition (or
>> multiplication) of an array by a scalar breaks the bank.
>
>> Any tips or directions to a helpful tutorial on IDL's memory policies
>> would be greatly appreciated. Thanks!
>
> Be careful with type conversion here! In particular, 1 is a short
> integer (or long integer depending on if compile_opt defint32 is
> used), so a = a + 1 here converts a to type int. Consider the
> following:
>
> IDL> a = bindgen(100, 100, 100)
> IDL> orig_mem = memory(/current)
> IDL> a = a + 1
> IDL> print, memory(/highwater) - orig_mem
> 2000082
>
> IDL> a = bindgen(100, 100, 100)
> IDL> orig_mem = memory(/current)
> IDL> a = a + 1B
> IDL> print, memory(/highwater) - orig_mem
> 1000096
>
> IDL> a = bindgen(100, 100, 100)
> IDL> orig_mem = memory(/current)
> IDL> a = temporary(a) + 1B
> IDL> print, memory(/highwater) - orig_mem
> 91
> IDL>
> IDL> a = bindgen(100, 100, 100)
> IDL> orig_mem = memory(/current)
> IDL> a += 1B
> IDL> print, memory(/highwater) - orig_mem
> 91
>

> IDL>
> IDL> a = bindgen(100, 100, 100)
> IDL> orig_mem = memory(/current)
> IDL> a++
> IDL> print, memory(/highwater) - orig_mem
>   91
>
> Mike
> --www.michaelgalloy.com
> Tech-X Corporation
> Software Developer II
