Subject: Re: arithmetic operation on array
Posted by Phillip Miller on Mon, 12 Aug 2013 23:08:38 GMT
View Forum Message <> Reply to Message

On 2013-08-12 22:22:14 +0000, Phillip Bitzer said:

> OK, I'll bite. There are three ways I can think to do this off the top
> of my head:
> 1) Do a loop, like Phillip said (what a fantastic name :-))
> 2) Rebin, like David said
> 3) Use the mysterious "add an extra dimension" method
> (https://groups.google.com/d/msg/comp.lang.idl-pvwave/Vu9rzqc kBNQ/HvkK_QnJrsgJ
> and more recently
> https://groups.google.com/d/msg/comp.lang.idl-pvwave/dM8XXas Eio0/d3__pvX7svMJ)
>
>
> Here are the sample code I used:
>
> data = RANDOMU(1L, 360, 180, 456)
> avg = MEAN(data, DIM=3)
> mData1 = FLTARR(360, 180, 456)
>
> tic & for i=0, 455 do mData1[*,*,i] = data[*,*,i] - avg & toc
>
> mData2 = FLTARR(360, 180, 456)
>
> tic & mData2 = data - Rebin(avg, 360, 180, 456) & toc
> tic & data = TEMPORARY(data) - Rebin(avg, 360, 180, 456) & toc ;just
> for completeness
>
> data = RANDOMU(1L, 360, 180, 456) ;redefine data - we changed it above
> mData3 = FLTARR(360, 180, 456)
>
> tic & mData3 = data - avg[*, *, 0] & toc
>
> I used mData (modified data) arrays so I can check I get the same
> answer, regardless of the method.
>
> The four times I get are, in order relative to the above:
> % Time elapsed: 0.12749481 seconds.
> % Time elapsed: 0.33231401 seconds.
> % Time elapsed: 0.33304191 seconds.
> % Time elapsed: 0.019619942 seconds.
>
> So, it seems the mysterious extra dimension method is the fastest, by
> an order of magnitude. Whoa.
>
> For prosperity,

```
>  IDL> print, !VERSION
>  { x86_64 darwin unix Mac OS X 8.2.2 Jan 23 2013      64      64}
```

Excellent, thanks for the tip.
I tried it on my 360x180x1000 double precision array and got the similar results as you, though not quite an order of magnitude difference between it and the "improved" loop method, the mysterious extra dimension method was indeed the fastest