Subject: Re: majority voting
Posted by JD Smith on Thu, 12 Feb 2009 22:45:18 GMT
View Forum Message <> Reply to Message

On Feb 12, 3:11 pm, David Fanning <n...@dfanning.com> wrote:
> Paolo writes:
>> And in case any other person a bit more slow on the intake
>> (like me) is wondering what the heck ++ and -- do in IDL,
>> they can be found in the documentation under "Mathematical Operators"
>> (searching for "++" did not return anything useful for me).
>
> Thank you. I have been searching for it off and on all
> morning. :-)
>
> There is nothing in the documentation, though, to imply this
> is a vectorized operation. In fact, just the opposite for the
> discussion on postfix operations. And insights, JD, on
> how that can be so?

The surprise isn't that it's vectorized: all of IDL's mathematical, relational, and bit operators are vectorized. The surprise is in how it treats repeated indices. I can only guess that ITT regards the old "no repeats" behavior as undesirable but inalterable due to legacy code, whereas when ++ and -- were introduced recently, no such legacy baggage existed, so they were free to improve the behavior (at the cost of consistency). That said, I know IDL has accumulated lots of cruft over the years, but I can imagine getting different answers for a [i]++ and a[i]+=1 might turn some people off.

I also noticed that a[i]+=1, in addition to being plagued by the "no repeats" issue, is at least 4x slower than a[i]++. As for ++a vs. a++ in expressions, when you don't care about the order of evaluation, using the former is somewhat faster, since it saves making a temporary copy of a.

JD