Subject: Large structures in IDL turn CPU to sweet, sticky molasses
Posted by MarioIncandenza on Sat, 13 May 2006 01:02:31 GMT
View Forum Message <> Reply to Message

I found this, most recently, when using a longish loop (250k) to assign a structure tag. I thought I was doing something that was crashing IDL, but indeed no, just making it very, very slow, like "Fly to Bogota for a cup of coffee" slow.

Now, when I set the structure up as a single structure, so that each tag is a vector N elements long, I have no performance problem. It's when I use a large array of structures, I find that

IDL> for i=0l,n do struct.tag[ri[ri[i]:(ri[i+1]-1)]]=x

takes FOREVER (and, notably, actually seems to decelerate, as if there's a memory issue?), while

IDL> temp=struct.tag * 0.
IDL> for i=0l,n do temp[ri[ri[i]:(ri[i+1]-1)]]=x
IDL> struct.tag = temp

is very fast.

I found a newsgroup thread discussing this issue in IDL, but it's from back in 2000, and I thought maybe there might be more information about ways around this problem.

...and yeah, loops are bad, but in this case I'm already using HIST_ND on the problem, and while I am aware of JD's ninja tricks using the histogram of the histogram, I worry about the their effects on code maintainability and sanity maintainability.