Subject: Re: IDL with multiple processors
Posted by steinhh on Fri, 04 Dec 1998 08:00:00 GMT
View Forum Message <> Reply to Message

In article <7479pd$j5i$1@agate.berkeley.edu>
korpela@islay.ssl.berkeley.edu (Eric J. Korpela) writes:

> IDL is pretty much single threaded. If you've got two processors, it's up
> to you to use 'em. You'd be suprised what you can do if you try.....
> This works under sunos... (you need to write your own kill proceedure,
> though. That's not too hard.)
>
> pid=call_external("/usr/lib/libc.so.1.9","_fork") ; your libc name may vary
> if pid then begin
> do some processing
> kill,pid
> endif else begin
> do some other processing
> dummy=call_external("/usr/lib/libc.so.1.9","_wait")
> endelse
>

Hmmm... dreaming of a "PIDL" (Parallel IDL).... How about
something like a few extra "directives":

```
 parallel begin

   task begin
     <some processing>
   endtask

   task begin
     <some other processing>
   endtask

   task begin
     <some third kind of processing>
   endtask

 endpara
```

... having approximately the same meaning as your example,
but with an unspecified mechanism. If no parallel processing
is available on the platform, IDL could just evaluate the tasks
in sequence. I.e. the programmer should never *rely* on these
tasks being executed in parallel, and (for now) interprocess
communication should thus not be used.. but making the
mechanism more or less like threads (threads share the

same address space/global variables) would eliminate most
of the need - synchronization would be done by syntax instead
(at each "endpara" directive).

In fact, RSI could implement something like this pretty soon
(after a tiny period of thinking it through, but without
putting any meat on the bone until later... Just
allowing the directives and ignoring them would be ok,
but one would at least be able to write programs that
will be speeded up in some future version...

I don't see it happening, though in my view it could be a nice
selling point over competitors to have "mechanisms to take
advantage of multiple processors, anticpating future h/w & s/w
developments".

[..snip..]

> I don't suppose RSI would be willing to add fork, kill, and wait to the
> language. :) It's also probably illegal to use fork on a machine with
> a floating license.

Actually, I don't think RSI would mind from the licensing point of
view: AFAIK, they have a "one user, one screen, any number of processes"
policy (you can even run two IDL processes on separate machines,
but on the same display, using just one license).

Regards,

Stein Vidar
