Data Analysis Applications Group (DAAG) Update: Compiling for faster Fortran

Justin Burruss
2004.08.06

Speed up your Fortran with the right compiler flags

- IDL is already tuned to take advantage of our new multiprocessor systems
- But Fortran compilers will, by default, not fully exploit this power
- You can make your Fortran codes faster just by adding the right compiler flags
  - No code changes required
- We will illustrate this using an example from a Friday Science Meeting in 2002
  - http://web/comp/daag/MeetingsArchive/SciMtg_020919-IDL.pdf

Optimizing reduced this from 6 seconds to 1 second

- We start with a small Fortran code based on an IDL example
  - We will time just this code fragment:

```
do i=1, 4000
  do j=1, 4000
    b(i,j) = tan(sin(cos(a(i,j))))
  end do
end do
```

- On Zeus, this executes in 6.18 seconds with debug flags enabled
  - Debug flag for lf95 is ***-g***
- Removing the debug flag takes us to 5.79 seconds execution time
- Adding ***–parallel*** takes us to 3.30 seconds execution time
- Adding ***–threads 4*** and setting the **`PARALLEL`** env var to 4 reduces execution time to a mere 1.06 seconds!

For reference, here is the exact usage

- Debug enabled (6.18s avg execution time)
  `lf95 par.f -o par -g`
- No debug (5.79 avg execution time)
  `lf95 par.f -o par`
- Parallel execution enabled (3.30 avg execution time)
  `lf95 par.f -o par --parallel`
- Parallel execution with 4 threads (1.06 avg execution time)
  `setenv PARALLEL 4`
  `lf95 par.f -o par --parallel --threads 4`

- Note that you can have the **`setenv PARALLEL 4`** without the **`--threads 4`**, but you **can't** have **`--threads 4`** without **`setenv PARALLEL 4`**
- Also, to see what optimizations are being used, add –info, e.g.
  `lf95 par.f -o par --parallel --info`