Fortran 90 and HPF Prototyping at ECS STL - Lessons Learned

1. Goals

- Test F90 compilers on F77 code
- Study F77 conversion to F90
- Study portability of F90 code
- Study performance gain of F90 over F77
- Study Fortran 90, HPF and parallel processing

2. Science Algorithm

- Special Sensor Microwave/Imager (SSM/I) Precipitation Rate Algorithm from MSFC

3. Compilers Tested

- NAG F90
- Portland Group High Performance Fortran (*pghpf*)
- Cray F90
- Applied Parallel Research (APR) *kapr*

4. Conversion of F77 to F90

- Expansion of tab character by F90 compiler is different from F77
  - Extensive code modification to replace tab character. A utility program can do the tab character replacement.
- Initialization of DOUBLE PRECISION constants with SINGLE PRECISION is not allowed by F90 standard
- Some F77 extensions are incompatible with F90 standard/libraries
  - External BYTE data
- F90 has stronger Type checking, so weakly typed F77 code can create problems
- Use of non-standard data size specifications (Integer*1) is necessary to access binary data

5. Portability of F90 code

- BYTE arrays may not be portable among F90 compilers
- Opening data files requires different parameters on each F90 compiler
- Cray F90 currently does not support DOUBLE PRECISION both in declarations and mathematical functions used in F77
- Many compilers (pghpf, kapr, etc.) translate F90 code to F77. DIGITAL F90 compiler converts F90 to some intermediate language

6. Performance

SGI Indigo with R4000 chip and 100 MHz clockspeed

Compiler/Language	User time (sec)	System time (sec)	Wall clock time (min:sec)	Speed up with respect to F77 compiled on native F77 compiler
F77 on F77 code	306.8	10.2	8:14	1.00
NAG F90 on F77 code	306.0	10.9	7:23	1.11
NAG F90 on F90 code	164.0	12.5	4:59	1.65
pghpf on F90 code	245.0	10.4	5:59	1.37
APR's kapr on F77/F90 code	200.7	9.9	5:18	1.55

SGI Power Challenge with R8000 chip and two 95 MHz processors

Compiler/Language	User time (sec)	System time (sec)	Wall clock time (min:sec)	Speed up with respect to F77 compiled on native F77 compiler
F77 on F77 code	222.0	3.97	3:42	1.00
NAG F90 on F77 code	182.0	3.5	3:09	1.17
NAG F90 on F90 code	160.0	3.8	2:48	1.38
pghpf on F90 code	221.0	4.24	3:41	1.00

7. F90, HPF and Parallel Processing

- Forge 90/xhpf uses HPF directives to distribute a F77 program across an array of processors using Single Program, Multiple Data (SPMD) Model in a distributed memory paradigm
- pghpf uses HPF directives on F90 program and distributes program across an array of processors using Single Program, Multiple Data (SPMD) Model in a distributed memory paradigm
- Hierarchical Data Format (HDF) library calls need "wrappers" to prevent parallel I/O operations on single-threaded HDF libraries. Wrappers can offset any improvement in performance gained through parallelism

8. Conclusions

- Fortran 90, because of many vendor specific implementations can introduce complexities during Science Software Integration & Test
- F77 code may not be all compilable on a F90 compiler due to misinterpretation of F77 extensions
- HDF libraries in its present form may not be suitable for parallel software implementation, especially in a Distributed Memory environment. Most parallel software development tools are based on the SPMD model. Wrappers are required around I/O calls to the HDF library to prevent parallel calls to the HDF libraries. Such an approach can offset any performance gains due to parallelism.
- Currently, only subset HPF (only a subset of the HPF constructs are implemented) compilers are available. Parallel software development using these tools have been encouraging.