Planning for Polyglot Programming

Gary Kumfert, Tamara Dahlgren, and Thomas Epperly

Lawrence Livermore National Laboratory

This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

UCRL-PRES-204176
Prediction

By 2015, HPC Programming will be essentially the same...

Multi-Physics Simulation (SPMD)

- F90
- C++
- C
- F77
- Python
- Java
- Physics
- AMR
- Linear Algebra
- Solvers
- Steering
- Portlet
Prediction

By 2015, HPC Programming will be essentially the same... only moreso!

Multi-Scale, Multi-Physics Simulation (MPMD)

Macro-Scale Simulation
- Python
- F90
- C++
- C
- F77

Micro-Scale Simulation
- F200x
- C99
- Ruby

Traditional Libraries
- C
- C
- F

Coupler
- Fwk Coupler
- Phys Coupler
Conclusions

- New languages for HPC will not replace the current patchwork

- Success of new languages in HPC may depend on how effectively they contribute to the patchwork
Interesting Story

- Current patchwork evolved in spite of significant barriers to interoperability
  - Type System
  - Programming Model
  - Support Tools
  - Libraries
Interoperability Problems with Type Systems

- **32 bit integer**
  - **ANSI C:** `short`, `int`, `long`, or `long long` (depends on platform) (use `int32_t`)
  - **Fortran 77:** `INTEGER*4` (as long as not compiled with fancy flag)
  - **Fortran 90/95:**
    ```
    integer( selected_int_kind(9) )
    ``
    “not guaranteed by standard, just empirically so” –T.E.

- **F90 Arrays**
Interoperability Problems With the Programming Model

- Fortran common blocks vs. OOP and strong encapsulation
- Scripting vs. immature support for dynamic loading in compiled codes
  - e.g.
    - Matlab has cmex/fmex compilation scripts to deal with platform specifics of building DLLs...
    - MPI often has mpi_cc scripts to handle flags and libraries for MPI
    - How to build a MEX file that launches a parallel run on the backside?
Interoperability Problems With the Support Tools

- **Java and Make don’t mix**
  - Difficult/Impossible to determine minimum set of files to recompile

- **Fortran 90 allows linker symbols with ‘.’ ‘%’ or ‘$’ in them** (Lahey does ‘.’)
  - One F90 coupling tool uses emacs to globally find and replace strings in .o files (binary files overflow sed’s buffers)

- **Fortran 90 Module files and Make have issues too**
Interoperability Problems and Library Design

- MPI-1.1 doesn’t let communicators be passed from one language to another
- MPI-2 fixes some of this:
  - for important language pairs (e.g. `MPI_Comm_f2c`)
- IMPI (in progress?) to handle interoperable MPI implementations
Babel is an IDL-based language interoperability tool

SIDL = Scientific Interface Definition Language
My Questions to the Workshop (1/2)

- Is Babel useful for language designers’ plans for polyglot programming?
  - We hope so.
    - Babel is extendable
    - Each language binding is an extension module
    - 3rd party extensions in progress at Utah, IU, & SUNY Binghamton
  - Unusual/exciting role.
    - We typically add bindings for languages we’re stuck with
    - Never had an opportunity to influence a language design
  - May be more useful in Polyglot Programming than a single (say C) API
My Questions to the Workshop (2/2)

- Is it useful to design Grammars of new languages so SIDL can be extracted easily from source code?
  - Much harder question
  - May require closer collaboration
  - Too creatively constraining?
    - Already established SIDL is richer than F77
    - SIDL’s object model relates only to public APIs and is not quite same as a programming language’s type system
Closing Remarks:

- Interoperability is a tricky thing to get right
- If people really want to program in your language, they will find some way to interoperate with it
- If interoperability with your language is really easy, more people will want to program in it
- Plan for Polyglot Programming
Contact Info

- Project: [http://www.llnl.gov/CASC/components](http://www.llnl.gov/CASC/components)
- Project Team Email: components@llnl.gov
- My Email: kumfert@llnl.gov
- Mailing Lists: majordomo@lists.llnl.gov
  
  subscribe babel-users [email address]
  subscribe babel-announce [email address]