Desarrollo de Aplicaciones Paralelas en Python

Lisandro Dalcín, Mario Storti, Rodrigo Paz
Centro Internacional de Métodos
Computacionales en Ingeniería (CIMEC)
(3000) Santa Fe, Argentina
e-mail: dalcinl@intec.unl.edu.ar

Las aplicaciones para cálculo científico orientadas a la simulación de problemas multifísica deben presentar a los usuarios del código una interfaz capaz de proveer un núcleo general de funcionalidades básicas, pero con la flexiblilidad suficiente para incorporar las necesidades particulares de cada modelo.

Con el paso del tiempo, el uso de aplicaciones de este tipo evoluciona en la generación de archivos de configuración y entrada de datos más o menos complicados, con sentencias que se adicionan a medida que surgen nuevos modelos a simular. En definitiva, la aplicación debe interpretar una especie de *"script ad-hoc"*, lo que origina dificultades para los usuarios como para el encargado del mantenimiento del código.

Una solución alternativa es la de utilizar algún lenguaje de extensión bien establecido, portable y con abundante soporte, que idealmente provea soporte para la programación funcional y orientada a objetos. De esta manera se provee a los usuarios una interfaz totalmente programable y fácil de extender. Es posible, por ejemplo, definir trozos de código (*"code snippets"*) para expresar la funcionalidad de una propiedad física con determinados parámetros, o modificar incluso en tiempo de ejecución el tipo y parámetros del *solver* a utilizar, e incluso desarrollar con rapidez versiones modificadas de dicho *solver*.

Python [1] es un lenguaje de programación moderno, orientado a objetos, interpretado e interactivo. Su sintaxis es muy clara y de rápido aprendizaje. Su repertorio incluye programación funcional, clases, excepciones, tipos de datos de alto nivel y soporte para *threads*.

Python es especialmente atractivo para el desarrollo de aplicaciones científicas.Su implementación es portable, corre en varias variantes de UNIX, Mac y Windows. El código fuente es totalmente abierto, y puede distribuirse incluso en aplicaciones comerciales. Existen interfaces a servicios del sistema y varias librerías populares, incluyendo las utilizadas en desarrollo de GUI's y visualización (X11, Motif, Tk, Mac, MFC, GTK, VTK, Qt). Puede utilizarse como lenguaje de extensión en aplicaciones que requieran una interface programable. Finalmente, es fácilmente extensible mediante nuevos módulos escritos en C/C++ que se incorporan inmediatamente al lenguage mediante importación dinámica. Esta última es quizás una de las características que más atractiva.

En este trabajo describimos nuestras experiencias en CIMEC utilizando Python en la implementación de aplicaciones paralelas sobre clusters de PC's; mostramos como generar interfaces a algunas librerías populares, tales como MPI [2] y PETSc [3], y su integración a nuestro código de elementos finitos PETSc-FEM [4]. También comentamos algunos detalles a tener en cuenta al instalar Python, MPI y PETSc para funcionamiento conjunto en un cluster de PC's.

Referencias

[1] Guido van Rossum. Python home page. http://www.python.org, 2003.

[2] Message Passing Interface Forum. Home page. http://www.mpi-forum.org, 2003.

[3] Satish Balay et al. PETSc home page. http://www.mcs.anl.gov/petsc, 2001.

[4] Mario Storti and Norberto Nigro. PETSc-FEM: A General Purpose, Parallel, Multi-Physics FEM Program. http://www.cimec.org.ar/petscfem, 2003.