User guidance to Non-Intrusive and Incremental Software Component Enabling NetSolve with Direct Communications V1.0

(http://cssa.ucd.ie/xin/phd/)
Xin Zuo

Version 1.0 Edition
Non-intrusive and incremental evolution of Grid programming systems Project,
Heterogeneous Computing Laboratory (HCL), School of computer Science and Informatics,
University College Dublin

========== PROGRAMMING ENVIRONMENT ==========

The software component for NetSolve to enabling direct communications is installed and run on following linux workstations:

Red Hat Linux 3.3.3-7, gcc version 3.3.3 20040412;
Fedora Core 2.6.11-1.1369_FC4, gcc version 4.0.0 20050519;

========== REQUIREMENTS ==========

The version of NetSolve which our software component is added and tested on is GirdSolve/NetSolve 2.0.

Important:

1. To use our software component, GPG option must be disabled.
2. Direct communications uses port 6234.

========== DOWNLOAD ==========

The Software Component can be downloaded from http://hcl.ucd.ie. A recent version can also be downloaded from following links:

http://cssa.ucd.ie/xin/phd/sc.htm

========== ORGANIZATION OF SOFTWARE COMPONENT ==========

Source codes:
Client wrapper API: mynetsl.c ; mynetsl.h ;
Server Connector: serverConnector.c; serverConnector.h; serverSetup.c;

========== INSTALLATION ==========

[On the server side]
To install our software component to a netsolve server, the steps of installation are as follows:
1) Create a directory named “Dc” at the root of NetSolve root directory.
2) Copy files serverConnector.c, serverConnector.h, serverSetup.c to the Dc

directory.
3) Build Obj file for Server Connector:

$gcc -Wall -g -c serverSetup.c -o serverSetup.o -I$NETSOLVE_ROOT/include -I$NETSOLVE_ROOT/Dc

4) Link it to the NetSolve Library:

$gcc -g -o serverSetup serverSetup.o -L. -lserverConnector -L$NETSOLVE_ROOT/Dc -lnetsolve -L$NETSOLVE_ROOT/lib/i686_pc_linux_gnu

5) Run the server Connector:

$./serverSetup

[On the client side]
To enabling direct communications, the steps of installation of our software component are as follows:
1) Create a directory named "Dc" at the root of NetSolve root directory.
2) Copy files mynetsl.c ; mynetsl.h to the Dc directory.
3) Build lib files for Wrapper API:

$gcc -Wall -g -c -o libmynetsl.o mynetsl.c -I$NETSOLVE_ROOT/include
$ar rcs libmynetsl.a libmynetsl.o

========== APPLICATION TUTORIAL ==========
This tutorial shows how to enabling direct communications by using the software component. The example is performing matrix operations. Files can be downloaded from:

http://cssa.ucd.ie/xin/phd/dc_tutorial.tar.gz

This includes:
- o libmatmul.c - contains the function which should perform the calculation
- o matmul.idl - a file which describes how the problem should be included into the NetSolve repository
- o myprog.c - an example program which invokes the NetSolve function

Instructions:
1. Building the Library
NetSolve provides all functions through statically linked libraries. Execute the following steps on the server to build libmatmul.a

```
$ gcc -c libmatmul.c
$ ar rc libmatmul.a libmatmul.o
```

2. Installing the Problem to the NetSolve Repository
To invoke a new function from the NetSolve client you have to add the function

to the problems list in a NetSolve server and recompile the server. Perform the following steps to include 'matmul'

o create a problem description file with
  $ $NETSOLVE_ROOT/bin/$NETSOLVE_ARCH/idltopdf matmul.idl
  which generates the file matmul.pdf

o copy the file matmul.pdf to the problems subdirectory in your NetSolve directory

o edit the file "server_config" in your NetSolve directory and add the following line in the "@PROBLEMS:" section:
  ./problems/matmul.pdf

o to rebuild the server, you have to set an environment variable which points to the directory of the previously created library (required by the problems definition file); depending on you shell execute
  $ export NSMATMUL_LIB=/path/to/libmatmul
  or
  $ setenv NSMATMUL_LIB /path/to/libmatmul

o afterwards rebuild the server with the command
  $ make server
  in your NetSolve directory

3. Invoking the Function via NetSolve
The file myprog.c demonstrates how to invoke the function calc via NetSolve in C. the original NetSolve calling is like:

Info = netsl("matmul()", metA, metB, metC);
Info = netsl("matmul()", metC, metD, metE);

To enable direct communication, our wrapper API *mynetsl() and handlers are used*:

Info = mynetsl("matmul()", matA, matB, hdlC);
Info = mynetsl("matmul()", hdlC, matD, matE);

To build an application program, the command of using our library is as follows:

$gcc -Wall -g -c myprog.c myprog.o -I$NETSOLVE_ROOT/include -I$NETSOLVE_ROOT/Dc
$gcc -g -o myprog myprog.o -L. -lmynetsl -L$NETSOLVE_ROOT/Dc -lnetsolve -L$NETSOLVE_ROOT/lib/i686_pc_linux_gnu

Invoking myprog uses 2 arguments:
o matrix size: integer with the dimension of the used matrix
o mode: 1 - blocking call
        2 - non-blocking call

========== MORE INFORMATION ==========
Please visit http://hcl.ucd.ie or contact xin.zuo@ucd.ie