Global Arrays Tutorial
Hands-on Guide

1. TUTORIAL Programs: (C and Fortran options available)

- Download tutorial source code “**ga-tutorial.tgz**” from here
  - http://www.emsl.pnl.gov/docs/global/tutorials.html
- unzip and untar the tarball. For example:
  - gunzip ga-tutorial.tgz
  - tar xvf ga-tutorial.tar
- The tutorial source codes are as follows:
  - transp1D.tutorial.F, transp1D-c.tutorial.c
  - matrix.tutorial.F, matrix-c.tutorial.c
  - The complete working codes are available for reference:
    - transp1D.F, transp1D-c.c matrix.F, matrix-c
- Pick any problem as explained below

3. Exercises:

- Chose Fortran or C version of tutorial problems (whichever language you prefer)
  - xxxx.c or xxxx.F
- Tutorial programs are incomplete. All you have to do is search file for comments marked with ###, and using the text as hints, replace the comments with subroutines or functions from the GA library to create a working code
- Compile and run as described below
- Compile your tutorial test program (in **ga-tutorial** directory**)** as follows:
  - Example: If your test program is transp1D-c.tutorial.c, compile the program as follows:
    - **gmake transp1D-c.tutorial.x**
- To execute the test, submit using the jobscript in the tutorial directory . This jobscript in the tutorial directory is for IBM loadleveller – you might need to write your own job script based on your job scheduler). Please make sure to change the name of the test program in jobscript.poe
    - **Example: llsubmit jobscript.poe**
- Compiling and running Instructions are also in the README file in tutorial directory

4. Problem 1: 1D Transpose (transp1D.tutorial.F, transp1D-c.tutorial.c)

- Transpose a distributed 1D vector containing N elements in the order 1,2,...,N into a distributed vector containing N elements in the order N,N-1,...,2,1
- Fortran version of this problem is in the file transp1D.F.tutorial
- C version is in transp1D.c.tutorial.
- Working versions of these codes are in transp1D.F and transp1D.c

Problem 2: Matrix Multiplication (matrix.tutorial.F, matrix-c.tutorial.c)

- A simple matrix multiply algorithm that initializes two large matrices as GAs. It then multiplies a block of columns by a block or rows from the GAs locally on each processor and copies the result into a third global array.
- Fortran version of this problem is in the file matrix.F.tutorial
- C version is in matrix.c.tutorial.
- Working versions are in matrix.F and matrix.c

Problem 3:

- Both the codes in problems 1 & 2 initialize the data by initializing a local array on processor 0 with all the data and then copying it to a distributed global array. For real problems it is usually undesirable to have all the data located on one processor at any point in the calculation. Can you modify these codes (problem 1 and 2) so that each processor only initializes the data owned by that processor?
- **1D transpose (Problem 1)**
  - Modify code so that each processor only initializes the local array a() with the data owned by that processor and then copy that data to the global array g_a
  - Hint: Use nga_distribution and nga_put
  - You will also need to modify the result checking part of the code as well so that it also only uses smaller portions of the total GA
  - Hint: copy locally held part of result GA into local array b and corresponding part of original vector into local array a and compare (use arrays lo, hi, lo2, hi2 to get this data).
- **Matrix Multiply (Problem 2)**
  - Modify code so that each processor only initializes the local arrays a and b with the data held locally by that processor. Then copy that data to the global arrays g_a and g_b.
  - Hint: Use nga_distribution and nga_put

GA Support

For GA Help/Support/Bug Report, please send an email to hpctools@pnl.gov

References:

GA Webpage: http://www.emsl.pnl.gov/docs/global/

GA API: http://www.emsl.pnl.gov/docs/global/userinterface.html
(or, go to GA webpage and click User Interface)

GA User Manual: http://www.emsl.pnl.gov/docs/global/documentation.html
(or. go to GA webpage and click documentation)