User's Guide for UPC-CHECK 1.0

High Performance Computing Group
Iowa State University

Background

UPC is an extension of C programming language for parallel execution on shared or distributed memory parallel machines (see http://upc.gwu.edu/) that uses the Partitioned Global Address Space (PGAS) programming model. UPC-CHECK 1.0 is a tool for the automatic detection of deadlocks and argument errors in UPC functions. UPC-CHECK was developed by Professor Glenn Luecke's High Performance Computing (HPC) Group at Iowa State University (ISU). Error messages issued by UPC-CHECK have been designed to help users quickly fix the problems detected. UPC-CHECK 1.0 was tested using the UPC Run-Time Error Detection test suite developed by ISU'S HPC Group, http://rted.public.iastate.edu/UPC.

How to use UPC-CHECK

UPC-CHECK uses the ROSE Toolkit from Lawrence Livermore National Laboratory to instrument UPC source code. When the instrumented source code is run, run-time errors are detected and error messages are issued to help fix the errors. Error messages provide the line number, executing thread and file name where the error occurred as well as additional information that will help fix the detected error.

To instrument sourcefile.upc and compile the instrumented UPC program, issue

```
upc-check [compiler options] [--upccheck:flag [--upccheck:flag] ...] -c sourcefile.upc
```

where "compiler options" are options which will be passed to the UPC compiler after instrumentation. This will produce a file named sourcefile.instrumented.o.

To instrument sourcefile.upc, compile and link the instrumented UPC program issue

```
upc-check [compiler and upccheck: options] –o a.out sourcefile.upc
```

This will produce an executable named a.out with the UPC-CHECK support routines automatically included. The executable created can then be run and the errors detected by UPC-CHECK will be reported. UPC-CHECK allows multiple source files, e.g.

```
upc-check [compiler and upccheck: options] –o a.out sourcefile1.upc sourcefile2.upc
```

Notice that UPC-CHECK is used by merely replacing the compiler name with upc-check.

UPC-CHECK automatically performs both deadlock and argument error checking, but each of these checks can be turned off to reduce running time for the instrumented executable if desired. The flags for the UPC-CHECK option, --upccheck, are listed in the following table. Notice that there is a flag to enable UPC-CHECK to trace function call stacks.

-a|-d_argument_check	disables argument checking (enabled by default)
-d|-d_deadlock_check	disables deadlock checking (enabled by default)
-s|-e_track_func_call_stack	enables tracing of function call stack (disabled by default)
-h|--h|--help	prints help for UPC-CHECK

For example, to disable deadlock checking in UPC-CHECK, one would issue

```
upc-check –d_deadlock_check –o source.out sourcefile.upc
```

Environmental Variables:

UPCCHECK_STOP_ON_ERROR : By default, UPC-CHECK will stop on an error and continue on a warning. Setting the environmental variable UPCCHECK_STOP_ON_ERROR to FALSE will allow execution to continue when UPC-CHECK finds an error. This could be used to find more than one error in a single run.

UPCCHECK_LIVELOCK_TIMEOUT : In some cases, a process may loop on upc_lock_attempt forever, because another thread holds the lock. This is not a deadlock because at least one thread may continue executing. To contrast this from deadlock, this is called a livelock. To detect this case, we use a timeout value 300 seconds from the time a upc_lock_attempt is issued until it is satisfied. If the upc_lock_attempt is not satisfied within that time, an error message will be printed to STDERR reporting that a livelock condition may be present. This message will include the upc_lock_attempt location and the lock involved. The environmental variable UPCCHECK_LIVELOCK_TIMEOUT allows the timeout value to be changed from the default value of 300 to some other number of seconds.

Installation Guide for UPC-CHECK 1.0

To use UPC-CHECK, one must already have a UPC compiler installed. The install_UPC-CHECK script below assumes that a UPC compiler is installed and is in the path of the user who is performing the install. The procedure checks first for upcc, which is LLNL Berkley UPC, then for upc which is either GNU-UPC or HP UPC, and then defaults to cc which is used for the Cray UPC compiler.

UPC-CHECK uses the ROSE Toolkit (http://www.rosecompiler.org/) and the ROSE Toolkit uses BOOST (www.boost.org). If BOOST and/or ROSE are not already installed, the afore-mentioned websites contain download and installation information. UPC-CHECK is installed by issuing the following:
STEP 1: wget http://hpcgroup.public.iastate.edu/UPC-CHECK/UPC-CHECK.tar.gz
STEP 2: tar -zxf UPC-CHECK.tar.gz
STEP 3: cd UPC-CHECK
STEP 4: ./install_UPC-CHECK -p INSTALL_DIR -b BOOST_DIR -r ROSE_DIR

If -p INSTALL_DIR is omitted, this is the same as -p /usr/local , similarly for -b and -r.

Once installed, the UPC-CHECK executable can be copied form INSTALL_DIR/bin to a system bin directory, like /usr/local/bin if desired.