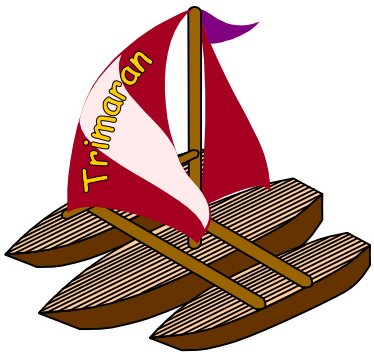

Installing and Running the Trimaran System

1. System Requirements

Trimaran requires your system to have certain packages installed. Some of these requirements may be relaxed in future releases, but for now they must be followed exactly. The install script will test for all of these, so if you like you can skip this section and come back to it if you have trouble.

These packages must be in the PATH both during installation and during usage.

- Trimaran currently runs on HP systems running HP-UX 10.20 or higher with the ability to run jobs as large as 256 Meg

Required Packages:

- **TCL/Tk** 8.0p2 (binaries named "wish" and "tclsh"), 8.1a2+ should work.
  Available at: http://salam.cae.wisc.edu/hppd/hpux/Tcl/tcl-8.0p2/

- **Perl** 5.0p3+ (binary named "perl"), installed in /usr/local/bin
  Available at: http://hpux.cae.wisc.edu/hppd/hpux/Languages/perl-5.004_04/

- **gnumake** version 3.76.1 to be installed as "gnumake" ***and*** "make".
  Available at: http://hpux.cae.wisc.edu/hppd/hpux/Gnu/make-3.76.1/

- **imake** - current version of 'imake' (usually the version included with HP-UX will not work), including the 'makedepend' portion to be installed and placed in your path. The O'Reilly & Associates book "Software Portability with imake (2nd edition)" is a good reference for imake. Their web site has the all the software available for free and has all the installation instructions. See their web site at: http://www.primate.wisc.edu/software/imake-book/index.html
  We strongly recommend installing itools-R6.3.tar.gz (from the above web site), since it is known to be compatible with IMPACT's Imakefiles.

- **GNU gzip** and **GNU gunzip** to be installed and placed in your path.
  Available at: http://hpux.cae.wisc.edu/hppd/hpux/Gnu/gzip-1.2.4/

- **cc** version G.10.32.05 (HP ANSI C Compiler)
  - **This is NOT the bundled version**
  - This is a code generation issue with Trimaran. All the modules in Trimaran itself can either be compiled with gcc or with the cc/aCC combination described below.

  - Trimaran's front end and Pcode profiler currently explicitly uses cc and is not compatible with gcc. Trimaran is currently investigating how to be become compatible with gcc's header files and libraries, especially with gcc's varargs implementation.

  - an older cc version is acceptable, but we recommend patching cc to a current version.

- **gcc** version 2.7.2.* or **aCC** version A.01.12 (HP ANSI C++ Compiler) for C++ compilation
  gcc and packages required by gcc are available at:
  http://hpux.cae.wisc.edu/hppd/hpux/Gnu/gcc-2.7.2.3/
  http://hpux.cae.wisc.edu/hppd/hpux/Gnu/binutils-2.7/
  http://hpux.cae.wisc.edu/hppd/hpux/Gnu/sed-2.05/

  - if aCC is used for c++, then cc must be used, **NOT** gcc
  - Trimaran defaults to using gcc for both C and C++.
    To use aCC and/or HP cc, set the variables CC and CXX in your environment before using Trimaran
    Please read the Installation instructions for proper settings.

- **GNU ar** version 2.7, in /usr/local/bin (may need a symbolic link)
  Available as part of the GNU binutils package:
  http://hpux.cae.wisc.edu/hppd/hpux/Gnu/binutils-2.7/

- **GNU ranlib** version 2.7, in /usr/local/bin (may need a symbolic link)
  Available as part of the GNU binutils package:
  http://hpux.cae.wisc.edu/hppd/hpux/Gnu/binutils-2.7/

- **Xterm**

- **GNU emacs**, invoked by the Trimaran GUI to edit machine descriptions.
  Available at: http://salam.cae.wisc.edu/hppd/hpux/Editors/emacs-20.2/
  Note that earlier versions are also acceptable.

- **tcsh** in /usr/local/bin/tcsh

NOTE: the following are required for IR Graphic Visualization

- **Dot** - Graph visualizing tool
  http://www.research.att.com/sw/tools/graphviz/
- **vcg** (binary named "xvcg") - Graph visualizing tool
  http://www.cs.uni-sb.de/RW/users/sander/html/gsvcg1.html
- **daVinci** - Graph visualizing tool
  http://www.informatik.uni-bremen.de/~agbkb/forschung/daVinci/daVinci.html

2. Installation

Before running the "install" script, you may wish to set the variables CC and CXX in your environment. The CC variable is used for ANSI C compilation and may be set to either "**gcc**" or "**cc -Aa +DA1.1**". If you wish to use aCC for C++ compilation then CC ***must*** be set to "cc -Aa +DA1.1" (and CXX to aCC).

The install script takes care to set any other environment variables.

To begin the installation process, type "**./install >& install.log**" at your shell prompt. Installation time is approximately one hour.

Executing '**grep Error install.out**' and '**grep Exit install.out**' and getting no output indicates a successful installation. If you encounter compilation errors or other problems please see the Trouble-shooting section.

After the install script is complete you will have a functional Trimaran distribution. Trimaran is designed to be installed in a multi-user setting.
The first time a user runs the Trimaran GUI (**trimaran/gui/trimaran-gui**) a local workspace is created for the user to hold their files. The default location for this is ~/trimaran-workspace. If you want to override this default, then set **TRIMARAN_WORKSPACE** to the desired path in your shell initialization files. In general, there is no need to do this.

Users of the system may also want to add **trimaran/bin** to their **PATH**s or a system administrator could add it to the central initialization file for users shells.

Users also need to set **SPEC95_PATH** appropriately (see 11 in **IMPACT's readme**).

3. Directory Structure

For the interested, the directory structure of the Trimaran release is as follows:

impact/	front end and opts on intermediate representation from UIUC.
elcor/	backend with optimizations on HPL-PD intermediate representation from HP-CAR.
simu/	Trimaran simulator and performance evaulator from NYU
gui/	gui for controlling all of the above from NYU.

In each user's workspace, pointed to by the environment variable TRIMARAN_WORKSPACE or defaulting to ~/trimaran-workspace :

benchmarks/	Links to the distributed set of benchmarks for use with Trimaran. Users can create new benchmarks here.
machines/	HMDES2 format machine descriptions
parms/	Parameter sets for compilation
projects/	The place where all intermediate and result file go after a compilation.
temp/	A temporary dumping-ground for the Trimaran GUI.

4. Troubleshooting

Problem: When I run the GUI I get an error saying it can't find package "cde".

Solution: Either make sure the environment variable TRIMARAN_REL_PATH is not set or make sure you have set TRIMARAN_REL_PATH to the place where Trimaran is installed. In general it shouldn't be necessary to set this variable.

Problem: When I run the GUI I get a message saying it is my first time running the GUI but I've already used the GUI before.

Solution: If you set the environment variable TRIMARAN_WORKSPACE when you first ran the GUI, make sure you have it set to the same value every time you run the GUI. An easy way to do this is to set TRIMARAN_WORKSPACE in your shell's initialization file.

Problem : I want to use HP cc and/or aCC for compilation and I don't know how.

Solution : Trimaran defaults to using gcc for both C and C++. To use aCC and/or HP cc, set the variables CC and CXX in your environment before using Trimaran as shown:

setenv CC '**cc -Aa +DA1.1**'
setenv CXX '**aCC**'

Problem : I'm trying to use aCC for C++ and gcc for C compilation but its not working.

Solution : It won't work. Only HP cc works with aCC.

Problem : I hate trying to remember the path to trimaran!

Solution : Add "yourpathtotrimaran"/bin to your PATH in your shell initialization file.

Problem : The install script says that make isn't really gnumake, but it is!

Solution : Maybe you have a make earlier in your path that is getting picked up before the real "make" (gnumake). Check "which make" and adjust your PATH if necessary.

If your problem does not appear above, please send us a detailed bug report stating exactly what you did up to the point where you had the problem and your system configuration. You can find our contact information in Section 8 : Contacting the Trimaran Team.

5. Using Trimaran

Trimaran provides a GUI based approach to carry out ILP (instruction-level parallelism) experiments with the Trimaran compiler system. The command **trimran-gui** launches this GUI. The main panel of the GUI holds buttons that spawn various other windows to help configure the system. A brief description of each of these windows is given below. Please browse through the online help in the GUI for a more detailed usage.

The Compile window is used to select a Project, compile the source code and execute its object code. A Project is a collection of source files and parameter settings associated with a compilation.

The Machine window allows one to create a customized version of the HPL-PD processor or to select one from a list of pre-configured processors. The selected processor will be used by the compiler as the target architecture for compilation.

The Parameters window allows one to set the parameters in the compiler's front-end (IMPACT), the back-end (Elcor), and the HPL-PD simulation environment (Simu).

The Statistics window is used to view and compare statistical information collected from the different runs of the executable. The information is rendered in various forms such as pie-charts, bar-diagrams, etc.

View IR allows one to view the IR (Internal representation) of the control flow graph used by the back-end (Elcor). It provides navigation mechanisms to move through the graph and view the various components in it.

The Projects window allows one to browse through the items in a Project. A Project is a collection of source files and parameter settings associated with a compilation.

The Configuration window allows one to customize this GUI.

6. Using tcc (Trimaran C compiler)

usage:

> tcc [**options**] *sources*	; options and source are described below
-d *bench*	; define/set benchmark name to ***'bench'***
-bb	; generate basic blocks only (no hyper- or superblocks)
-bh	; generate hyperblocks (no superblocks) -- **requires -p**
-bs	; generate superblocks -- **requires -p**
-O*n*	; set IMPACT optimization level to ***n*** (0 = no optimization)
-p	; profile the code using profiling information for the benchmark
	; -- **requires -d**
-inline	; turn on inlining -- **requires -d**
-Si	; compile up to the IMPACT stage (generate *.mc* files)
-Se	; compile up to the ELCOR stage (generate *.el* files)
-Ss	; compile up to the SIMULATOR stage (generate *executable* file)
-Sr	; compile up to the SIMULATOR stage and run generated binary file
	; -- **requires -d**
-o *file*	; set output file name to ***'file'***
-G*opt*	; pass ***'-opt'*** to the underlying C compiler
-g	; debugging information (passed to the underlying C compiler)
-I*dir*	; add ***'dir'*** to the list of include directories
-L*dir*	; add ***'dir'*** to the list of library directories
-l*lib*	; add ***'lib'*** to the list of libraries linked
-n	; generate native binary
-ref	; generate and native binary for benchmark. The output is stored in
	; ***$BENCHMARK_REF_DIR/benchmark_name.OUT***
	; (if $BENCHMARK_REF_DIR is not set, then by default this is set to
	; **trimaran/benchmarks/OUTPUTS**) -- **requires -d**
	; NOTE: **-n** and **-ref** options override any **-S** options
**.c*	; C source files
**.mc*	; mc source files
**.el*	; el source files
**.o*	; additional external object files to link

7. Contacting the Trimaran team

The Trimaran team can be contacted at support@trimaran.org
Bug reports should be sent to bugreport@trimaran.org