DEFAULTS User Manual

TONGPING LIU

1 Compiling the llvm compiler

1. mkdir llvm-clang-src
2. cd llvm-clang-src
3. wget http://llvm.org/releases/3.2/llvm-3.2.src.tar.gz
4. wget http://llvm.org/releases/3.2/clang-3.2.src.tar.gz
5. tar zxvf llvm-3.2.src.tar.gz
6. tar zxvf clang-3.2.src.tar.gz
7. mv clang-3.2.src llvm-3.2.src/tools/clang
8. cd llvm-3.2.src
9. patch -Np1 < ../../falsesharing/project/compiler/instrumenter.llvm-3.2.patch
10. mkdir -p llvm-3.2-build
11. cd llvm-3.2-build
12. ../llvm-clang-src/llvm-3.2.src/configure --prefix=/home/tianc/git/llvm-3.2-build/ --sysconfdir=/etc --libdir=/home/tianc/git/llvm-3.2-build/lib/llvm --enable-optimized --enable-shared --enable-targets=all --disable-assertions --disable-debug-runtime --disable-expensive-checks
13. make -j8 & make install

2 Compiling the runtime system (library)

1. cd runtime
2. make

3 Compiling a program

Here, we are using the patched clang to compile a program. The following step is using memtest for an example.

1. cd memtest
2. $(CLANG_DIR)/bin/clang -Wl,$(RUNTIME_DIR)/runtime/libdefault64.so -finstrumenter -g thread_memtest.c -lpthread -o thread_memtest.

   In this step, we should make sure the following things. $(CLANG_DIR) and $(RUNTIME_DIR) should be replaced by real directory name.

   - First, we must use the patched clang to compile the program (specified in $(CLANG_DIR));
   - Second, we should specify explicitely the runtime system by "-Wl,$(RUNTIME_DIR)/runtime/libdefault64.so".
   - Third, we must specify the flag "-finstrumenter".

4 Compiling and Running PARSEC benchmarks

We assume to use new making system, provided under "evaluation" package. The original way to run PARSEC can be seen in http://parsec.cs.princeton.edu/.

In this package, there are 3 different directories, a script and some basic rules of makefile(Defult.mk).

- datasets: it is used to hold different datasets for different benchmarks.
- tests: source code of different benchmarks.
- include: some header files used by some Phoenix benchmarks.
- run_benchmarks.py: script to run a benchmark or all benchmarks.
- Default.mk: basic rules of makefile. Normally, we should make sure the directory of runtime system. CXX_DEFAULT = clang -Wl,$(RUNTIME_DIR)/libdefault64.so -finstrumenter CC_DEFAULT = clang -Wl,$(RUNTIME_DIR)/libdefault64.so -finstrumenter -std=gnu89

In order to run PARSEC benchmarks, we first have to set corresponding datasets in the beginning.
The parsec sourcecode and its datasets can be got from http://parsec.cs.princeton.edu/download/2.1/parsec-2.1.tar.gz.
Here is an example to copy datasets. We are using the "dedup" as an example here.

1. tar zxvf parsec-2.1.tar.gz
2. Go to $(EVALUATION)/datasets.
3. mkdir dedup
4. cp parsec-2.1/pkgs/kernels/dedup/inputs/input_native.tar ./ & tar xvf input_native.tar
5. Go to $(EVALUATION)/tests/dedup. After we do this, we make sure that the file or directory after decompressed has the same name as that showed in TEST_ARG of dedup Makefile.

   For example, Makefile of dedup showed liked this: "TEST_ARGS = -c -p -f -t $(THREADS)-i(DATASET_HOME)/dedup/media -o output.dat.ddp".
6. To run the false sharing detection tools on dedup benchmark, we can run "make eval-defaults". To run normal pthreads program, we can run "make eval-pthread".