

Running CDT Simulations on a Linux Machine

Rajesh Kommu
`rkkommu@ucdavis.edu`

Physics Department
University of California, Davis

June 1st, 2011

Outline

Prerequisites

Installing and Running CDT

Software Required for CDT

- 1 Steel Bank Common Lisp Compiler
- 2 QuickLisp
- 3 Emacs and SLIME

Installing SBCL I

- ① Download SBCL version 1.0.49 from <http://www.sbcl.org/platform-table.html>.
- ② `tar xvjf sbcl-1.0.49-x86-64-linux-binary.tar.bz2`
- ③ `cd sbcl-1.0.49-x86-64-linux`
- ④ `export SBCL_HOME=`
- ⑤ `INSTALL_ROOT=${HOME}/sbcl-1.0.49 sh install.sh`
- ⑥ Add the following lines to your `.bashrc` file
 - `export SBCL=${HOME}/sbcl-1.0.49`
 - `export SBCL_HOME=${SBCL}/lib/sbcl`
 - `export SBCLBIN=${SBCL}/bin`
- ⑦ Modify `$PATH` in your `.bashrc` file by adding `$SBCLBIN` to the list of directories in the `$PATH` variable.
- ⑧ If you do not find `$PATH` in your `.bashrc`, add the following line to your `.bashrc`

Installing SBCL II

- `export PATH=.:$SBCLBIN:$PATH`

- 9 Step 4 is needed when you are upgrading SBCL, when any previous definitions of `SBCL_HOME` need to be undefined.
- 10 Close all terminals, open a new terminal, and type `sbcl`. You should see the following:

This is SBCL 1.0.49, an implementation of ANSI Common Lisp.
More information about SBCL is available at <http://www.sbcl.org/>.

SBCL is free software, provided as is, with absolutely no warranty.
It is mostly in the public domain; some portions are provided under
BSD-style licenses. See the CREDITS and COPYING files in the
distribution for more information.

*

- 11 Type (quit) at the SBCL prompt and hit enter.

Installing Quicklisp I

- 1 <http://www.quicklisp.org/beta> has some useful information about Quicklisp.
- 2 Download `quicklisp.lisp` from <http://beta.quicklisp.org/quicklisp.lisp>
- 3 Make sure that `quicklisp.lisp` is in your `$HOME` directory.
- 4 Type `sbcl --load quicklisp.lisp` in your `$HOME` directory and follow the instructions.
- 5 Evaluating `(foo)` means typing `(foo)` (including the parentheses) at the SBCL prompt and hitting enter.
- 6 A flurry of messages should end with the following:

Installing Quicklisp II

==== quicklisp installed ====

To load a system, use: `(ql:quickload "system-name")`

To find systems, use: `(ql:system-apropos "term")`

To load Quicklisp every time you start Lisp, use: `(ql:add-to-init-file)`

For more information, see <http://www.quicklisp.org/beta/>

NIL

*

- ⑦ Type `(ql:add-to-init-file)` at the SBCL prompt and hit enter.
- ⑧ Then type `(ql:quickload "quicklisp-slime-helper")` at the SBCL prompt and hit enter.
- ⑨ Once again, a flurry of messages follow, ending with

Installing Quicklisp III

To use, add this to your ~/.emacs:

```
(load (expand-file-name "~/quicklisp/slime-helper.el"))  
;; Replace "sbcl" with the path to your implementation  
(setq inferior-lisp-program "sbcl")
```

```
("quicklisp-slime-helper")  
*
```

- 10 Quit SBCL by evaluating (quit)
- 11 Modify your \$HOME/.emacs file according to the above instructions by adding the following two lines:

```
(load (expand-file-name "~/quicklisp/slime-helper.el"))  
(setq inferior-lisp-program "sbcl")
```

- 12 Start Emacs, and then the following sequence ESC x slime. You should see the following inside Emacs:

```
; SLIME 2011-05-22  
CL-USER>
```

- 13 To exit SLIME, type the sequence ESC x slime-quit-lisp
- 14 To exit Emacs type the sequence CTRL-x CTRL-c

Installing CDT

- 1 Download `/home/rajesh/cdt-latest.tar.gz` from `dewitt.physics.ucdavis.edu` into your `$HOME` directory.
- 2 Soon, this code will be in the github repository.
- 3 `tar xvjf cdt-latest.tar.gz`

Running CDT I

① `cd cdt-release-1.0/periodic/2p1`

② Start Emacs and SLIME.

③ Evaluate `(load ‘‘cdt2p1.lisp’’)`

④ Evaluate the following code:

```
(initialize-t-slices-with-v-volume :num-time-slices 32
                                  :target-volume (* 32 1024)
                                  :spatial-topology ‘‘S2’’
                                  :boundary-conditions ‘‘periodic’’)
```

⑤ Then evaluate the following, in sequence:

① `(set-k0-k3-alpha 1.0 0.78 -1.0)`

② `(setf NUM-SWEEPS 1000)`

③ `(generate-data-console)`

⑥ `(generate-data-console)` useful only during initial exploratory stages.

Running CDT II

- ⑦ Use `(generate-data)`,
`(generate-spacetime-and-movie-data)`,
`(generate-data-v2)`, `(generate-data-v3)`
- ⑧ `(generate-data)` overwrites the spacetime data file
(`.3sx2p1` or `.4sx3p1`) at each save point, thus saving only
the latest state of the spacetime. Useful during thermalization.
- ⑨ `(generate-data-v2)` creates a new spacetime data file at
each save point, thus generating members of ensemble.
- ⑩ `(generate-data-v3)` similar to `(generate-data-v2)`,
except also saves the spatial slice information at each save
point, in a `.s2sx2p1` or `.s3sx3p1` file.
- ⑪ *File names automatically generated!*
- ⑫ Save points are controlled by the `SAVE-EVERY-N-SWEEPS`
variable, which defaults to 10.

Running CDT III

- 13 To generate a fresh simulation, as a background job, create a directory for the simulation, under 2p1. For example `mkdir -pv S2-PERIODIC-T064-V081921-1.0-0.78-0.02`.
- 14 In this directory, create a file named `S2-PERIODIC-T064-V081921-1.0-0.78-0.02.lisp`, which looks like:

```
(proclaim '(optimize (speed 3)
                    (compilation-speed 0)
                    (safety 0)
                    (debug 0)))

(load "../utilities.lisp")
(load "../globals.lisp")
(load "../simplex.lisp")
(load "../moves.lisp")
(load "../initialization.lisp")
(load "../montecarlo.lisp")

;; fresh simulation section
(initialize-t-slices-with-v-volume :num-time-slices 64
                                  :target-volume (* 80 1024)
                                  :spatial-topology "S2"
                                  :boundary-conditions "PERIODIC")
```

Running CDT IV

```
(set-k0-k3-alpha 1.0 0.78 -1.0)
(setf SAVE-EVERY-N-SWEEPS 100)
(setf NUM-SWEEPS 50000)
(generate-data)
;; end of fresh simulation section

;; resume simulation section
;;(setf SAVE-EVERY-N-SWEEPS 100)
;;(setf NUM-SWEEPS 50000)
;;(with-open-file (infile "???.3sx2p1")
;;(load-spacetime-from-file infile))
;;(format t "finished loading data at ~A~%" (cdt-now-str))
;;(generate-data-v2 250001)
```

- 15 `nohup sbcl --dynamic-space-size 1024 --script S2-PERIODIC-T064-V081921-1.0-0.78-0.02.lisp >> S2-PERIODIC-T064-V081921-1.0-0.78-0.02.log &`
- 16 To resume a simulation:
 - 1 comment out the fresh simulation section and uncomment the resume simulation section.
 - 2 specify the correct file name in place of `???.3sx2p1`
 - 3 Start `(generate-data-v2)` at the correct sweep number