Porting RISC-V to GNU Guix

Efraim Flashner
Porting RISC-V to Guix

- Some stuff about guix
- Transactional diagram
- Build container
- DAG
- Built natively
Porting RISC-V to Guix
Porting RISC-V to Guix

What other architecture(s) is it similar to?
Porting RISC-V to Guix

- Just another niche architecture
- Faster adoption than aarch64
Porting RISC-V to Guix

- No JIT in most situations like powerpc64le
- Too new for old config.sub config.guess scripts like aarch64
Porting RISC-V to Guix

config.guess timestamp = 2014-11-04

uname -m = riscv64
uname -r = 5.19.0-1012-generic
uname -s = Linux
uname -v = #13-22.04.1-Ubuntu SMP Thu Jan 12 15:34:31 UTC 2023

/usr/bin/uname -p =
/bin/uname -X =

hostinfo =
/bin/universe =
/usr/bin/arch -k =
/bin/arch =
/usr/bin/oslevel =
/usr/convex/getsysinfo =

UNAME_MACHINE = riscv64
UNAME_RELEASE = 5.19.0-1012-generic
UNAME_SYSTEM = Linux
UNAME_VERSION = #13-22.04.1-Ubuntu SMP Thu Jan 12 15:34:31 UTC 2023

configure: error: cannot guess build type; you must specify one
Porting RISC-V to Guix

gcc/config/riscv/linux.h

#define STARTFILE_PREFIX_SPEC                   
   "/lib" XLEN_SPEC "/&" ABI_SPEC "/"    
   "/usr/lib" XLEN_SPEC "/&" ABI_SPEC "/"    
   "/lib/ 
   "/usr/lib/ "
Porting RISC-V to Guix

**Millak** commented on Jul 10, 2022

This was tested on a HiFive Unmatched board with rust-1.54.

- Add support for riscv64

**thepowersgang** commented on Jul 11, 2022

Did that "Just Work" doing a full bootstrap on a riscv64 machine? (i.e. `./build-1.54.sh` and `make -C run_rustc RUSTC_VERSION=1.54.0`)

**Millak** commented on Jul 11, 2022

I also passed RUSTC_TARGET=riscv64gc-unknown-linux-gnu and used the distro packaged llvm-13.

I broke out the steps in build-1.54.sh. Looks like I ran 'make test' after make run_rustc and skipped 'make local_tests'. I also used the bundled libgit2. So yeah, I think you can say it basically just worked :)

[https://git.savannah.gnu.org/cgit/guix.git/tree/gnu/packages/rust.scm?h=staging#n286](https://git.savannah.gnu.org/cgit/guix.git/tree/gnu/packages/rust.scm?h=staging#n286)

Took about 56 hours to build rust-1.54. I used the tagged 0.10 mrustc commit with my patch. About 6 hours for each of rust-1.55, 1.56 and 1.57. I then used rust-1.57 to build librsvg.

**thepowersgang** merged commit **495f85b** into **thepowersgang:master** on Jul 17, 2022

6 checks passed
(synopsis "Compiler for the Rust programming language")
(description "Rust is a systems programming language that provides memory
safety and thread safety guarantees.")
(home-page "https://github.com/thepowersgang/mrustc")

;; The intermediate generated code is known to be inefficient and
;; therefore the build process needs 8GB of RAM while building.
;; It may support i686 soon:
;; <https://github.com/thepowersgang/mrustc/issues/78>.
(supported-systems ("x86_64-linux" "aarch64-linux" "riscv64-linux"))
Porting RISC-V to Guix

@@ -338,9 +343,11 @@
  safety and thread safety guarantees.")
  (home-page "https://github.com/thepowersgang/mrustc")

-  ;; So far mrustc is (x86_64|aarch64)-only. It may support i686 soon:
+  ;; The intermediate generated code is known to be inefficient and
+  ;; therefore the build process needs 8GB of RAM while building.
+  ;; It may support i686 soon:
  ;; <https://github.com/thepowersgang/mrustc/issues/78>.
-  (supported-systems '("x86_64-linux" "aarch64-linux"))
+  (supported-systems '("x86_64-linux" "aarch64-linux" "riscv64-linux"))
Porting RISC-V to Guix

- go: gccgo@10 => go@1.16 => newer go
  go@1.16 fails the test suite
- node: backport support to node@10
  later versions have a circular dependency between llhttp and node
- java: backport support to before java7?
- haskell: grab a binary from Debian?
guix build hello --tune
Porting RISC-V to Guix
Thank you