CheriCloud Architecture

Version 1.0

This interim document is not released for public consumption

Brooks Davis, Robert N. M. Watson
SRI International and the University of Cambridge*

March 16, 2013

Abstract

CheriCloud is a facility permitting remote access to CHERI-based computer systems for the purposes of kernel and userspace software development. This document proposes a hardware, software, and network architecture for a prototype CheriCloud to be implemented at the University of Cambridge Computer Laboratory. Its goal is to support not just local SRI and Cambridge users in developing on CHERI systems, but also selected "guest users" to be drawn from DARPA's CRASH and MRC2 programmes. Although CheriCloud must be able to support development and refinement of the CHERI CPU architecture, scalable remote hardware development is not a current design goal.

*Sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL), under contract FA8750-10-C-0237. The views, opinions, and/or findings contained in this report are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

1 Purpose

In order to allow broader access to the CHERI platform we building *CheriCloud*, a self-contained network of DE4 CHERI systems and a FreeBSD development host. This platform will support an enhanced internal software development cycle at SRI International and the University of Cambridge, and also be a foundation on which to provide remote access to CHERI hosts for other DARPA CRASH and MRC2 programme performers.

While the primary goal is to support software development, CHERI's status as a highly experimental hardware platform means that CheriCloud must provide facilities for hardware-level debugging and upgrade of test hosts. Critically, the architecture must support debugging of CPU bugs and ongoing refinement of the CHERI design – in particular, *in situ* CPU upgrades.

2 Hardware

The core of the system is a 4U rack-mount chassis holding eight Terasic DE4s and an Intel NUC compact PC. The NUC will use a suitable version of Ubuntu Linux, consistent with other CHERI development hosts, able to run Altera's development and debugging tools (although it is not intended to be used for bitfile synthesis). The NUC is connected to the DE4s through a USB hub and provides access to the JTAG units to provide console access and support debugging. The DE4 boards also have RJ45 serial ports which may be used to provide console access. Connectivity to the serial ports will be via a USB serial console or a hardware console server with the final configuration TBD; however, the intent is that end users of the CheriCloud system will not require direct access to the NUC.

The first Ethernet port of each DE4 is connected to a top of rack gigabit Ethernet switch. This port is used for SSH access and NFS traffic. We intend to run the network with jumbo frames enabled to improve per-packet throughput. In the future we are considering splitting SSH and NFS traffic so that each one uses a separate interface.

A 1U FreeBSD server is the CheriCloud frontend and NFS server for the nodes. One ports is connected to a public network and the other to the top of rack switch shared with the DE4s. As the FreeBSD server will host non-Computer Laboratory guest users, it will be an independently managed host that does not make use of the lab's directory or file services.

A diagram of the system connections can be found in Figure 1.

3 Operational Model

Boards will be reserved using a table in a wiki as is done on the FreeBSD netperf cluster. An example of a reservation page can be found at `h`ttps://wiki.freebsd.org/TestClusterOneReservations.

Ordinary users will have directly access to the FreeBSD server. They will have an account and access to a ZFS file system that is exported to the private network so it can be mounted by the DE4s. This will allow rapid compilation and fast access For users requiring complete control of the operating system it will be possible to install a full operating system to a ZFS file system and configure a kernel to boot that file system.

In addition to the FreeBSD head node limited access will be provided to some functions such as resetting hung boards, restoring boards to known configurations, and viewing the jtag console via the Linux controller. Wrapper scripts will provide these functions that use dedicated SSH keys to run specific commands on the remote end. Console access may be directly or via conserver.

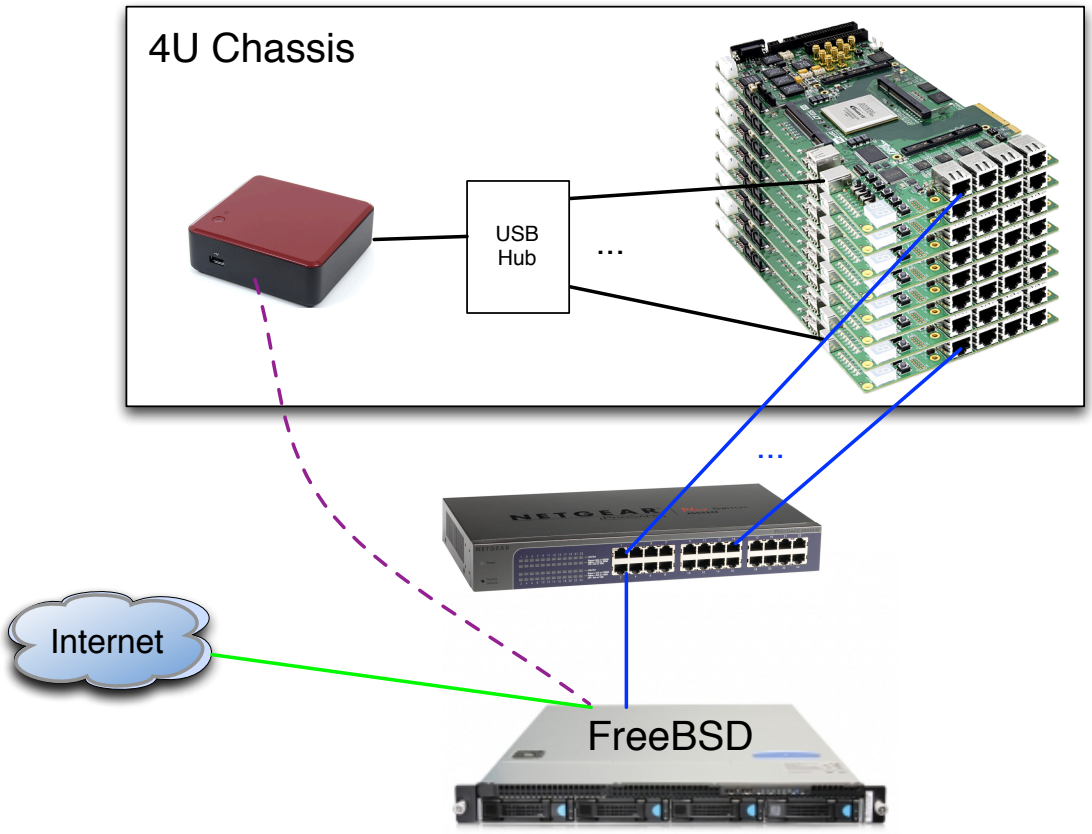

Figure 1: CheriCloud Connections

Trusted users will have full access to the Linux controller so that they can directly load new bitfiles and kernel or run diagnostic tools such as `cherictl`. Most likely these users will be existing people with CL accounts. The reason for this restriction is the potential for errors resulting in loss of work due to actions performed on the wrong board. Future versions may include stronger wrapper or NUC units for each board to allow broader access, for example access by students doing hardware development.

4 Required Development

We will need to develop or complete a number of features to make CheriCloud a reality. These include:

- DE4 hard reset functionality
- DE4 software reboot functionality (ideally MIPS soft reset)
- Scripts to image a DE4 from power on
- Jtag console access scripts
- First stage boot loader (boot2)
- Second stage boot loader (loader) with netboot
- Programmable interrupt controller (PIC)
- Scripts to automate account setup; particularly setup of delegated zfs file systems.