

HOWTO Set up icecream

As some of you may have noticed, Pat V has added icecream (Tue Aug 14 00:10:54 UTC 2018) to -current. What others may not know is what purpose for which icecream was created. In a nutshell, icecream is a centralized model for distributed compilation of packages; similar to distcc, but uses a central “server” to coordinate compiling.

Why would someone want to use this?

- A lot of users nowadays have lots of computers lying around their house with no purpose; just taking up space, but their owners are unwilling to throw away what still works.
- There are some packages which users may want to compile themselves, but one machine won't cut it (takes too long). Sharing the load among multiple machines cuts down on the load that one machine had to do in the past.
- We are a community-based distro; true to its roots, Slackware's strength lies in the willingness of its users to improve on deficiencies, such as compiling their own packages (Perhaps taking the load off of core members who need a break)?

How to set it up

Software

I would recommend using -current, and **only** selecting sets A,AP,D,K,L & N. Others may be added as desired, but this is the setup I use, and it has worked well for me on the client machines.

Networking

Post-install, set your machine hostname to whatever your fancy is; icecream ignores this and uses the domain name (next prompt), so make sure that all of the machines you intend to use for icecream are in the same domain! Next, select static IP and assign it an IP of the form 192.168.x.x, where the first x is usually 0, and the second x can be any number between 1 and 254. (0 and 255 are used for special purposes).

NOTE: If you are intending to connect your client machines to the Internet, please be sure to use a firewall. I highly recommend IPFire. It is simple to install and once you do the post-install configuration, all future interaction can be done using the Web interface.

The way I have my client machines configured to use the firewall is as follows: I edit /etc/resolv.conf to point to my ISP's DNS servers, and /etc/rc.d/rc.inet1.conf to assign an IP address and netmask to eth0. Then I configure the gateway address to point to the IP address I assigned to the green0 interface (Refer to the IPFire documentation for explanation). This works well, and effectively hides the client machines from direct access from the Internet.

Icecream Configuration

On first boot, the client machines should have the *rc.iceccd* startup script made executable. Select the machine you intend to use as a scheduler machine and make the *rc.icecc-scheduler* script executable. Only one machine can have this script executable! This is usually the main machine. I

usually don't need to set up an additional user account, as these machines are on a private network and totally inaccessible from the Internet;and, per the icecream wiki, root access is necessary for it to run. No additional setup for icecream is necessary.

NOTE: The following section is via Pat V.:

...[I]f `/etc/profile.d/icecream.sh` sees either `iceccd` or `icecc-scheduler` running on the machine, it will add `/usr/libexec/icecc/bin` to the beginning of the `$PATH`, and this directory contains symlinks from all the usual compiler names to `/usr/bin/icecc`.

Really, all there is to do is make `/etc/rc.d/rc.icecc-scheduler` executable on one machine, and `/etc/rc.d/rc.iceccd` executable on all the machines that will be part of the compile cluster. And then set `-j` high enough.

Hardware

Icecream's effective power to parallelize compilation depends on having several client machines for it to utilize.

For example, I have 10 Mac Pros running Slackware, and the kernel compilation uses all of them (I launch `htop` to monitor the CPU usage pattern on select machines.)

I have all my machines hooked up to a keyboard and monitor (4 to 2 4-port KVM VGA switches; I use a DVI to VGA adapter, then connect the VGA cable to that) for local control, but if you like to ssh into your machines "remotely", feel free to do so.

Obviously, what kind of setup you use is immaterial to icecream, but I highly recommend using a Gigabit Ethernet switch to interconnect them. They are inexpensive and will handle the bandwidth necessary easily.

Conclusion

You can use any number of packages to demonstrate icecream's effectiveness, but I usually use a kernel compile to demonstrate icecc's power. The number of jobs used will obviously be dependent on the total number of cores in your cluster.

Icecream is an easy way to parallelize computing, and is well worth checking out.

Happy Slacking!

Originally written by Matthew Miller (1337_powerslacker on LQ).

From:
<https://docs.slackware.com/> - **SlackDocs**

Permanent link:
https://docs.slackware.com/howtos:misc:howto_set_up_icecream

Last update: **2020/02/11 13:13 (UTC)**



