

## Tips & Tricks

 [Click To Print](#)

### Tips & Tricks

October 2008 • Vol.8 Issue 10

Page(s) 96-97 in print issue

# Warm Up To Penguins

## Idle Speculation



For one reason or another, many people choose to leave their computers on all the time. Rather than let those computational cycles go to waste, there are a number of initiatives that will make your idle computer part of a giant, distributed supercomputer. Using the vast power unleashed by such an arrangement, these massively parallel computers are working to solve some of the most important scientific and mathematical problems of the day, as well as a few frivolous ones. And unlike many software categories where Linux lags behind the commercial operating systems, these projects have gone out of their way to make Linux part of the equation.




***When you initially start BOINC, it will prompt you to choose from a variety of distributed computing projects.***

Before we look at the specific projects, we'll discuss in general terms how they work and issue some caveats. These programs implement what we computer jocks call massively parallel, coarse-grained computation. Coarse-grained is a way of saying that each processor in the overall system works on a relatively large piece of data for a prolonged period of time, rather than doing a few simple operations on a small piece of data. There are only certain types of problems that work well on a coarse-grained system, usually those that require trying a lot of permutations of a possible solution or performing the same detection algorithm over a large set of data. These projects work by taking the overall problem set and dividing it up over a large number of computers. For example, if the (admittedly simple) problem was to determine out what the square of every number between one and 100 was, the system might assign the numbers one through 10 to machine A, 11 through 20 to machine B, and so on. Each machine would chew over its respective piece of the puzzle and return its findings, which would be gathered together into a final result.

Because distributed computing projects need to send new jobs, retrieve work units, and return results, all of them require network connectivity. Typically, the bandwidth required is not high, because it isn't the volume of data that's the issue. It's the complex computations each chunk requires that consumes the bulk of your resources. Each work unit is usually given to more than one computer to double-check results. This also prevents a malicious user from corrupting the data by running a bogus version of the client that returns bad results.

The client programs try to only consume idle process time, staying out of the way of running programs. Most can be configured to only start running after the system has been idle for a certain amount of time. They shut down when you resume normal activity, although some are less adept at reducing their resource consumption, which can lead to sluggish performance immediately after resuming from an idle state.

The other caution is that when this idea was first developed, chip makers weren't as conscientious about power savings as they are today. Modern processors can drastically cut down their power usage when idle, but running one of these programs can have the side effect of raising your power consumption by keeping your CPU actively working.



Originally, all of the projects had their own client programs, but many of them have now



***The World Community Grid is a collection of projects that are all available at your fingertips.***

consolidated under one common platform called BOINC (Berkeley Open Infrastructure for Network Computing), which can be downloaded at [boinc.berkeley.edu](http://boinc.berkeley.edu). The nice thing about BOINC is that distributed computing junkies can easily switch between projects without loading different software. If you suddenly decide to try curing cancer rather than searching for little green men, it's a snap.

If you're running some distributions, you may find BOINC already available in your distro's package manager. For example, under Ubuntu you can find the boinc-client and boinc-manager packages. If you select them for install, you'll get a new item in your System Tools menu called "BOINC Manager." You use this tool to configure BOINC and control it when it runs.

When you first start the manager, you'll have nothing running, so the first step is to click the Add Project button. This opens a window that presents a list of projects that are currently available.

Once you've selected a project, you'll need to register. Each project maintains its own user registry, which is mainly used to keep track of how much each user has contributed (primarily for bragging rights). As a final step, most projects will launch a browser window that gives you the opportunity to provide some additional user information and join a team. Teams pool the points they've earned by completing work, but just like individual scores, it's merely a fun way to compete to see who can do the most jobs.

Once you've added your first project, BOINC will start grabbing spare CPU resources and using it to run computations for the project. If you add more than one project, BOINC will queue up work for each project you're involved with, starting the next job in the queue after your system completes its current work unit.

There are a few significant differences between BOINC under Linux and Windows. BOINC runs as a screen saver in Windows; it only starts up after the system has been idle. Under Linux, it starts running if there hasn't been keyboard activity for a certain amount of time, which you can set with the manager. BOINC won't run as a screen saver in Linux, so the only way to see the nice visualizations that some of the projects display as they chug along is to click the Advanced View button, then select the running project, and click the Show Graphics button. This will launch a separate window with the visualization you'd normally see as a screen saver under Linux. This is probably the biggest advantage that the Windows version has over the Linux one. We'd like to see the BOINC folks step up and get a screen-saver mode working under Linux.

One project, the World Community Grid ([www.worldcommunitygrid.org](http://www.worldcommunitygrid.org)), actually encompasses a lot of smaller projects. The World Community Grid project has about a half dozen sub-projects underneath it, including the Fight-AIDS@Home project. To participate in WGC projects, first create a WGC account, then select on the Web site the WGC projects in which you want to participate. Add WGC as a project under BOINC, and when WGC runs, it downloads work from one of the WGC projects you designated.

There are even some BOINC clients designed to run on the graphical GPUs of game consoles, such as the PS3. But whether you run BOINC on one system or a dozen, you'll be doing your part to solve some of today's most important scientific and mathematical problems. ■

*by James Turner*

## **Infinite Loop: Warp Drives Nanometers Closer To Reality**

Two Baylor University physicists believe it's possible to create a warp engine that could make a vessel "travel" faster than the speed of light without violating the theory of relativity. The engine would create a space-time "bubble" around the ship, which would remain stationary as space-time shrinks in front of it. But before you place a want ad for a Scottish Chief Engineer, know this: The physicists' calculations depend on "some arbitrary advance in technology or some alien technology."

*dsc.discovery.com/news/2008/07/28/warp-speed-engine-02.html*

---

Copyright © 2009 Sandhills Publishing Company U.S.A. All rights reserved.