

# Advanced Unix System Administration

## Spring 2007

### Project 1

This project is due by 11:59 PM on **Friday, March 16**. Writeups should be sent via email to <sluo+decal@ocf.berkeley.edu>; project setups need to be complete by this time.

Once set up (which should be in a couple of days), your virtual servers for this project will be accessible from the login server (`plague.ocf.berkeley.edu`) at the IP addresses `10.20.1.xx[56]`, where `xx` is the last two digits of your UID on the login server. You will be able to log in as yourself, with your password on the login server, and you will have full root access. Each one is configured with two processors and 512 MB of RAM (though `free` reports otherwise, and if you all try to use that much RAM at once, you'll fail); the servers share disk space on two 250 GB hard disks concatenated together.

## Scenario

You are the lead system administrator for Foobar Software Solutions, Inc., a company distributing and providing support for a Linux distribution which is an enhanced version of the product of a prominent North American enterprise Linux vendor (the one with the funny hat logo). Your company is small, but the quality of its support and its cut-rate prices mean that it is growing in popularity rapidly. With growth, of course, comes growing pains; your computing infrastructure is straining to handle the load placed on it, and your low-cost business model means no money for getting new hardware anytime soon. (Rumors are floating around that management is negotiating a big contract with a Tier 1 server vendor, but even if that pans out, it'll be a few quarters before the money really starts rolling in.)

## Tasks

1. As your distribution has moved up the list on the DistroWatch site, traffic to your website has increased considerably, to the point where your webserver is now somewhat slow in responding. You are asked to take a look and develop a more responsive setup.
  - a. Your current website uses static HTML pages for most of its content. Start by setting up Apache 2.2 on your server and configuring a handful of static pages (nothing elaborate; there will be a standard set of test pages once I get around to creating it). It's suggested that you work from source for maximum flexibility.

- b. Benchmark the configuration under load (tools such as `ab` (Apache Benchmark), `http_load`, or WebStone may be helpful). Identify the biggest bottlenecks on performance, and propose ways of resolving them. Fix at least one of them, and show that your changes increase the performance of your setup.
  - c. The developers use Bugzilla to keep track of bugs in your software (and there are quite a few, since no one's entirely happy with the QA done by that prominent North American enterprise Linux vendor). Set up a Bugzilla install, creating a few users and sample bugs. (You will need either MySQL or PostgreSQL; you may choose which one you want to install. For extra credit, install both and compare the two, though keep in mind that Bugzilla's MySQL backend is more mature than its PostgreSQL one.) Benchmark it, identify the bottlenecks on performance, and propose ways of resolving them. Fix at least one of them.
  - d. The typical traffic load on your webservers has 90% of the traffic going to the static pages, with only 10% going to the bugs. With this mixed load, examine the performance of your server and the responsiveness of the Bugzilla interface for your developers (who need to modify bugs, remember) under load. Suggest any changes that you think necessary; if you can carry them through, make them.
  - e. Just in case that deal goes through with the Tier 1 server vendor, make a wish-list of hardware purchases that would improve performance the most, based on your benchmarking experience.
2. You are approached by one of your company's developers, who is looking for ways to speed up the compiling of software (he frequently recompiles packages with tweaks and bugfixes several times a day). Analyze a Linux kernel compile, identify the bottlenecks, and suggest ways to speed the compile. (You don't need to implement the changes; presumably your developers are smart enough to do this work for themselves.)