Optimizing RAID and other performance techniques
FAST TRACK

This month we show you some tools for optimizing disks, speeding up web servers, and building custom parallel applications. **BY JOE CASAD**

The more powerful our computers become, the more we ask of them – and the less we are willing to pay for top performance. After all the startups, shutdowns, reboots, and technological breakthroughs, the game is still the same: Stop the toe tapping and clock watching – *bring me better performance.* This month we examine some strategies for speeding up your Linux systems.

Our first article shows how to build better performance for RAID storage systems. You'll learn about the benefits of aligning the filesystem with the underlying RAID infrastructure, and we'll show you the results of some practical tests that demonstrate where and when you can expect better performance.

The next article studies the memcached distributed memory object caching system, which is often used for building faster and more efficient websites. Memcached, which is used on high-profile websites such as LiveJournal.com, improves page load performance by reducing the load on database servers by as much as 90%.

The last article in this month's cover story looks at high-performance computing with the LAM/MPI parallel programming environment. The open-source LAM implementation of the Message Passing Interface (MPI) specification provides a stable and efficient framework for writing parallel applications. Although LAM/MPI has recently taken a back seat to the newer and more advanced Open MPI system, a large install base around the world, and support from Red Hat Enterprise and other Linux systems, means LAM is still a leading choice for many parallel-programming scenarios.

In a year when companies, non-profits, and even home users are trying to save money on IT expenses, many Linux users hope to patch the holes in their budgets with a few good ideas. This month's high-performance collection offers some practical solutions for stretching out your systems. ■