Putting Linux reliability to the test

The Linux Technology Center evaluates the long-term reliability of Linux

Li Ge
Linda Scott
Mark VanderWiele

This article documents the test results and analysis of the Linux kernel and other core OS components, including everything from libraries and device drivers to file systems and networking, all under some fairly adverse conditions, and over lengthy durations. The IBM Linux Technology Center has just finished this comprehensive testing over a period of more than three months and shares the results of their LTP (Linux Test Project) testing with developerWorks® readers.

The IBM Linux Technology Center (LTC) was founded in August 1999 to work directly with the Linux development community with a shared vision of making Linux succeed. Its 200-odd employees make it one of the larger corporate groups of open source developers. They contribute code ranging from patches to structural kernel changes; from file systems and internationalization work to GPL'd drivers. They also work to track Linux-related developments within IBM.

Test results at a glance

The following summary is based on the results of the tests and observations on the duration of the runs:

- The Linux kernel and other core OS components -- including libraries, device drivers, file systems, networking, IPC, and memory management -- operated consistently and completed all the expected durations of runs with zero critical system failures.
- Every run generated a high success rate (over 95%), with a very small number of expected intermittent failures that were the result of the concurrent executions of tests that are designed to overload resources.
- Linux system performance was not degraded during the long duration of the run.
- The Linux kernel properly scaled to use hardware resources (CPU, memory, disk) on SMP systems.
- The Linux system handled continuous full CPU load (over 99%) and high memory stress well.
- The Linux system handled overloaded circumstances correctly.
The tests demonstrate that the Linux kernel and other core OS components are reliable and stable over 30, 60, and 90 days, and can provide a robust, enterprise-level environment for customers over long periods of time.

Particular areas of interest for the LTC are Linux scalability, serviceability, reliability, and systems management -- all with a view to making Linux ever more enterprise-ready. Enabling Linux to work on the S/390 mainframe and porting the JFS journaling file system to Linux are among their many contributions to the community.

Another of the LTC's core missions is to professionally test Linux in lab settings the way any commercial project is tested. The LTC contributes to the LTP Linux Test Project (LTP), as do SGI, OSDL, Bull, and Wipro Technologies. What follows are the results obtained from a comprehensive set of tests from the LTP suite on the Linux kernel for an extended period of time. As you may have guessed, Linux held up admirably under the continued stress.

**Linux reliability measurement**

**Objectives**

The objective of the Linux reliability effort at the IBM Linux Technology Center is to measure the Linux operating system's stability and reliability over long periods of time with an emphasis on workloads relevant to Linux customer environments using the LTP test suite (see Related topics for more on the LTP). Identification of defects was not the primary focus.

**Test environment overview**

This article describes the test results and analysis of 30- and 60-day Linux reliability measure tests using the LTP test suite. The tests used SuSE Linux Enterprise Server v8 (SLES 8) as the testing kernel and IBM pSeries servers as testing hardware. A specially designed stress-test scenario of LTP was used to exercise a wide range of kernel components in parallel with networking and memory management, and to create a high stress workload on the testing system. The Linux kernel, TCP, NFS, and I/O test components were targeted with a heavy-stress workload.

**The tests**

**At 30 days**

30-day LTP stress execution results for pSeries

- Machine: p650 LPAR
- CPU: (2) Power4- 1.2 GHz
- Kernel: Linux 2.4.19-ull-ppc64-SMP (SLES 8 SP 1)
- LTP version: 20030514
- 99.00 percent Average CPU utilization (User: 48.65 percent, System: 50.35 percent)
- 80.09 percent Average memory utilization (8GB)

**Observations:**
Putting Linux reliability to the test

- SLES 8 PPC64 30-day stress run successfully completed on p650 LPAR
- LTPstress was the test tool. Test cases were executed both in parallel and in sequence
- Kernel, TCP, NFS, and I/O test components were targeted with heavy stress workloads
- Success rate: 97.88 percent
- Zero critical system failures

**Figure 1. 30-day LTP stress execution results**

At 60 days

60-day LTP stress execution results: pSeries

- Machine: B80
- CPU: (2) Power3- 375 MHz
- Kernel: Linux 2.4.19-ull-ppc64-SMP (SLES 8 SP 1)
- LTP version: 20030514
- 99.96 percent average CPU utilization (User: 75.02 percent, System: 24.94 percent)
- 61.69 percent average memory utilization (8GB)
- 3.86 percent average swap utilization (1GB)

**Observations:**

- SLES 8 PPC64 60-day stress run successfully completed on pSeries B80
- LTPstress was the test tool. Test cases were executed both in parallel and in sequence
- Kernel, TCP, NFS, and I/O tests components were targeted with heavy stress workloads
- Success rate: 95.12 percent
- Zero critical system failures

**Figure 2. 60-day LTP stress execution results**
Test infrastructure

Hardware and software environment

Table 1 shows the hardware environment.

### Table 1. Hardware environment

<table>
<thead>
<tr>
<th>System</th>
<th>Processors</th>
<th>Memory</th>
<th>Disk</th>
<th>Swap partition</th>
<th>Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>pSeries 650 (LPAR) Model 7038-6M2</td>
<td>2 - POWER4+(TM) 1.2GHz</td>
<td>8GB</td>
<td>36GB IBM Ultrastar (other disks present, but unused)</td>
<td>1GB</td>
<td>Ethernet controller: AMD PCnet32</td>
</tr>
<tr>
<td>pSeries 630 Model 7026-B80</td>
<td>2 - POWER3(TM) 375 MHz</td>
<td>8GB</td>
<td>16GB</td>
<td>1GB</td>
<td>Ethernet controller: AMD PCnet32</td>
</tr>
</tbody>
</table>

The software environment was the same for both the pSeries 630 Model 7026-B80 and the pSeries 650 (LPAR) Model 7038-6M2. Table 2 shows the software environment.

### Table 2. Software environment

<table>
<thead>
<tr>
<th>Component</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>SuSE SLES 8 with Service Pack 1</td>
</tr>
<tr>
<td>Kernel</td>
<td>2.4.19-ul1-ppc64-SMP</td>
</tr>
<tr>
<td>LTP</td>
<td>20030514</td>
</tr>
</tbody>
</table>

Methodology

System stability and reliability are generally measured as continuous hours of operation and reliable uptime of a system.

The runs started with a set of 30-day baseline runs and progressed to 60- and 90-day Linux test runs on xSeries and pSeries servers. Initial emphasis was placed on kernel, networking, and I/O testing.

Test tool

The Linux Test Project (LTP; see Related topics for links and more information) is a joint project with SGI, IBM, OSDL, Bull, and Wipro Technologies with a goal to deliver test suites to the open source community that test the reliability, robustness, and stability of Linux. The Linux Test Project is a collection of tools for testing the Linux kernel and related features. The goal is to help improve the Linux kernel by bringing test automation to the kernel testing effort.

Currently, there are over 2000 test cases within the LTP suite, covering the majority of kernel interfaces such as syscalls, memory, IPC, I/O, filesystems, and networking. The test suite is
updated and released monthly and runs on multiple architectures. There are 11 known LTP test suite tested architectures including i386, ia64, PowerPC, PowerPC 64, S/390, S/390x (64bit), MIPS, mipsel, cris, AMD Opteron, and embedded architectures. We used LTP version 20030514 -- the latest available at the time -- in our reliability testing.

Test strategy

There were two unique phases in the baseline run: a 24-hour "initial test," followed by the stress reliability run phase, or "stress test."

Passing the initial test was an entry requirement. The initial test consisted of a successful 24-hour run of the LTP test suite on the hardware and operating system that would be used for reliability runs. The driver script runalltests.sh, which comes with the LTP test suite package, was used to validate the kernel. This script runs a group of packaged tests in sequential order and reports the overall result. It also has the option to launch several instances running in parallel simultaneously. By default, this script executes:

- Filesystem stress tests
- Disk I/O tests
- Memory management stress tests
- IPC stress tests
- Scheduler tests
- Commands functional verification tests
- System call functional verification tests

The stress test verified the robustness of the product during high system usage. In addition to runalltests.sh, a test scenario called ltpstress.sh was specially designed to exercise a wide range of kernel components in parallel with networking and memory management and to create a high-stress workload on the testing system. ltpstress.sh is also part of the LTP test suite. The script runs similar test cases in parallel and different test cases in sequence in order to avoid intermittent failures caused by running into the same resources or interfering with one another. By default, this script executes:

- NFS stress tests
- Memory management stress tests
- Filesystem stress tests
- Math (floating point) tests
- pthread stress tests
- Disk I/O tests
- IPC (pipeio, semaphore) tests
- System call functional verification tests
- Networking stress tests

System monitoring

The modified top utility that comes with the LTP test suite was used as a system monitoring tool. top provides an ongoing look at processor activity in real time. The enhanced top utility has
additional functions that can save snapshots of `top` results to a file and give the average summary of the resulting file, including information such as CPU, memory, and swap space utilizations.

In our tests, snapshots of system utilization (or `top` output files) were taken every 10 seconds and saved to result files. In addition, snapshots of system utilization and LTP test output files were taken daily or weekly to have data points to determine whether systems were degrading during long-period runs. This function was controlled by `cron` jobs and scripts.

**Before testing**
All selected testing systems had hardware configured as similarly to each other as possible. Extra hardware was removed to reduce potential hardware failure. Minimum-security options were selected during image installation. At least 2 GB of disk space was reserved for storing the `top` data files and LTP log files.

Note that this is a testing scenario; in real life, users would be well advised to keep security settings at much higher than minimum.

**During testing**
The system was left undisturbed for the duration of the tests. Occasional access of the system to verify that the test was still executing was acceptable. Verification included using the `ps` command, checking `top` data, and checking LTP log data.

**After testing**
When the test completed, the system monitoring tool `top` was stopped immediately. All `top` data files, including daily or weekly snapshots and LTP log files, were saved and processed in order to provide data for analysis.

**Conclusions**
The findings discussed in this article are based on a solution that was created and tested under laboratory conditions. These findings may not be realized in all environments, and implementation in such environments may require additional steps, configurations, and performance analysis.

However, as most Linux kernel testing efforts have only been conducted over short periods of time, this series of tests provides us first-hand data and results of longer runs. The series of tests also provides data for heavy-stress workloads on Linux kernel components, as well as TCP, NFS, and other test components. The tests demonstrate that the Linux system is reliable and stable over long durations and can provide a robust, enterprise-level environment.
Related topics

- Find useful information and links at the LTP Linux Test Project home page and at the LTP project page -- both at at SourceForge.
- Documentation for the project includes the LTP man pages (among others).
- The LTP pages also offer summaries and links to other Linux Test Tools.
- Visit the IBM Linux Technology Center home page to read their latest news and announcements.
- The IBM Linux Technology Center projects page lists the group's current working projects.
- The article "Inside IBM: Dan Frye and the Linux Technology Center" (Consulting Times) gives more background on the IBM LTC.

© Copyright IBM Corporation 2003
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)