性能测试工具 Lmbench 的安装使用与参数说明

1 工具简介

Linux 性能测试工具 Lmbench 是一套简易可移植的，符合 ANSI/C 标准为 UNIX/POSIX 而制定的微型测评工具。一般来说，它衡量两个关键 特征：反应时间和带宽。Lmbench 旨在使系统开发者深入了解关键操作的基础成本。其官方网站是：http://www.bitmover.com /lmbench/。

2 安装过程及一般错误解决办法

安装使用 Linux 性能测试工具 Lmbench 的安装相对比较简单，到其官方网站下载压缩包 Lmbench3.tar.gz

下面以 lmbench3.tar.gz 在 /opt 目录下为列，说明安装方法

解压 tar -xzvf lmbench3.tar.gz

cd lmbench3

make results

如果在 make 的时候出错，提示类似

$make results

make[1]: Entering directory `/home/kyuan/lmbench3/src'

gmake[2]: Entering directory `/home/kyuan/lmbench3/src'

gmake[2]: *** No rule to make target `../SCCS/s.ChangeSet', needed by bk.ver'..

gmake[2]: Leaving directory `/home/kyuan/lmbench3/src'

make[1]: *** [lmbench] Error 2

make[1]: Leaving directory `/home/kyuan/lmbench3/src'

make: *** [results] Error 2

这是需要修改 src/Makefile，将这么一行(在 231 行的样子)，将$O/lmbench : ../scripts/lmbench bk.ver 中的 bk.ver 去掉，就可以了。

如果编译没有错误，就会出现一些选择提示以对测试进行一个配置并生成配置脚本，首先出现的如下（：

If you are running on an MP machine and you want to try running
multiple copies of lmbench in parallel, you can specify how many here.
Using this option will make the benchmark run 100x slower (sorry).
NOTE: WARNING! This feature is experimental and many results are
known to be incorrect or random!
MULTIPLE COPIES [default 1]

此处是提示你同时运行多少个并行的测试，对应为结果中的 scal load 项
接下来是选择作业调度控制方法，选 1 允许调度作业即可。如下：

```
Options to control job placement
1) Allow scheduler to place jobs
2) Assign each benchmark process with any attendent child processes
     to its own processor
3) Assign each benchmark process with any attendent child processes
     to its own processor, except that it will be as far as possible
     from other processes
4) Assign each benchmark and attendent processes to their own
     processors
5) Assign each benchmark and attendent processes to their own
     processors, except that they will be as far as possible from
     each other and other processes
6) Custom placement: you assign each benchmark process with attendent
     child processes to processors
7) Custom placement: you assign each benchmark and attendent
     processes to processors
Note: some benchmarks, such as bw_pipe, create attendent child
processes for each benchmark process.    For example, bw_pipe
needs a second process to send data down the pipe to be read
by the benchmark process.    If you have three copies of the
benchmark process running, then you actually have six processes;
three attendent child processes sending data down the pipes and
three benchmark processes reading data and doing the measurements.
Job placement selection: 1
```

再接下来就是指定内存，本次指定为 512M，如下

```
==========================================================================
Several benchmarks operate on a range of memory.    This memory should be
sized such that it is at least 4 times as big as the external cache[s]
on your system.      It should be no more than 80% of your physical memory.
The bigger the range, the more accurate the results, but larger sizes
take somewhat longer to run the benchmark.
MB [default 2814]
512
```

下一步为所要运行的子集，默认为全部，本次我们用默认值即可，如下

```
lmbench measures a wide variety of system performance, and the full suite
of benchmarks can take a long time on some platforms.    Consequently, we
offer the capability to run only predefined subsets of benchmarks, one
for operating system specific benchmarks and one for hardware specific
benchmarks.    We also offer the option of running only selected benchmarks
which is useful during operating system development.

Please remember that if you intend to publish the results you either need
to do a full run or one of the predefined OS or hardware subsets.

SUBSET (ALL|HARWARE|OS|DEVELOPMENT) [default all]
```

最后出现如下画面即为测试开始了，下面只需耐心的等就可以了。
Confguration done, thanks.
There is a mailing list for discussing lmbench hosted at BitMover.
Send mail to majordomo@bitmover.com to join the list.
Using config in CONFIG.user3-VT3456-8614CMB
2013 年 11 月 11 日 星期一 10:06:07 CST
Latency measurements
2013 年 11 月 11 日 星期一 10:06:45 CST

Calculating file system latency
2013 年 11 月 11 日 星期一 10:06:49 CST

Local networking
2013 年 11 月 11 日 星期一 10:08:29 CST

Bandwidth measurements

3 测试结果查看

测试完毕执行 make see 可查看到测试结果报告 Lmbench 的结果及其说明、本次测试结果如下
cd results && make summary percent 2>/dev/null | more
make[1]: 正在进入目录 `/opt/performance/lmbench3/results'

L M B E N C H 3 . 0 S U M M A R Y

(Alpha software, do not distribute)

Basic system parameters

Host OS Description Mhz tlb cache mem scal
pages line par load
bytes

--- --- --- --- --- --- --- ---

user2-VT3 Linux 3.8.0-c i686-pc-linux-gnu 1598 88 128 3.0400 1
Processor, Processes - times in microseconds - smaller is better

---

Host OS Mhz null null open slct sig sig fork exec sh

```
                                        call    I/O stat clos TCP    inst hndl proc proc proc
--------- ------------- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----

user2-VT3 Linux 3.8.0-c 1598 0.20 0.39 1.57 3.53 8.23 0.57 2.52 647. 1873 4330

Basic integer operations - times in nanoseconds - smaller is better
-------------------------------------------------------------------

Host                 OS   intgr intgr   intgr   intgr   intgr
                            bit    add      mul      div      mod
--------- ------------- ------ ------ ------ ------ ------
ser2-VT3 Linux 3.8.0-c 0.6300 0.0900 1.3100     14.1     13.5
Basic float operations - times in nanoseconds - smaller is better

-------------------------------------------------------------------

Host                 OS   float   float   float   float

                          add      mul      div      bogo
--------- ------------- ------ ------ ------ ------
user2-VT3 Linux 3.8.0-c 1.2500 2.2000     14.5     14.4
Basic double operations - times in nanoseconds - smaller is better
-------------------------------------------------------------------
Host                 OS   double double double double

                          add      mul      div      bogo
--------- ------------- ------   ------ ------ ------

user2-VT3 Linux 3.8.0-c 1.2500 2.5100     14.5     14.5
Context switching - times in microseconds - smaller is better
-------------------------------------------------------------------------

Host                 OS   2p/0K 2p/16K 2p/64K 8p/16K 8p/64K 16p/16K 16p/64K

                          ctxsw   ctxsw   ctxsw ctxsw   ctxsw     ctxsw     ctxsw
--------- ------------- ------ ------ ------ ------ ------ ------- -------
user2-VT3 Linux 3.8.0-c 3.6600 2.8800     73.0 5.8600     35.7      20.8      42.6
*Local* Communication latencies in microseconds - smaller is better
-------------------------------------------------------------------------
Host                 OS 2p/0K   Pipe AF       UDP   RPC/     TCP   RPC/ TCP

                        ctxsw          UNIX             UDP             TCP conn
--------- ------------- ----- ----- ---- ----- ----- ----- ----- ----
user2-VT3 Linux 3.8.0-c 3.660   17.4 13.7   17.0             22.9            106.

File & VM system latencies in microseconds - smaller is better
-------------------------------------------------------------------------------

Host                 OS    0K File        10K File       Mmap      Prot     Page     100fd
```

Create Delete Create Delete Latency Fault Fault selct
--- --- --- --- --- --- --- --- --- ---
user2-VT3 Linux 3.8.0-c 16.6 13.0 54.7 20.5 18.1K 0.577 4.03690 3.841
Local Communication bandwidths in MB/s - bigger is better

Host OS Pipe AF TCP File Mmap Bcopy Bcopy Mem Mem

UNIX reread reread (libc) (hand) read write
--- --- --- --- --- --- --- --- --- --- ---
user2-VT3 Linux 3.8.0-c 759. 1214 833. 1166.3 2846.5 1125.7 1119.8 2874 1517.

Memory latencies in nanoseconds - smaller is better
(WARNING - may not be correct, check graphs)

Host OS Mhz L1 $ L2 $ Main mem Rand mem Guesses
--- --- --- --- --- --- --- ---
user2-VT3 Linux 3.8.0-c 1598 2.5060 12.5 56.5 143.7
make[1]:正在离开目录 `/opt/performance/lmbench3/results'

4 相关测试结果参数说明如下

ID	测试分类	技术参数	中文名称	测试结果	测试项描述
1	Basic system parameters	Tlb pages	转换缓存页数	88	转换后备缓存的页面数
2		Cache line bytes	缓存行字节数	128	高速缓存行字节数
3		Mem par	存储器分层并行化	3.0400	存储器分层并行化
4		Scal load	并行负载	1	并行执行的 lmbench 数目
5	Processor, Processes	Null call	简单系统调用（取进程号）	0.20（单位：μs）	简单系统调用所花时间（单位微秒）
6		Null I/O	简单 IO 操作（空读写的平均）	0.39（单位：μs）	简单 IO 操作（空读写的平均时间）
7		stat	取文档状态的操作	1.57（单位：μs）	取文档状态的操作所花时间
8		Open clos	打开然后立即关闭文档操作	3.53（单位：μs）	打开文档，然后再关闭文档操作所花的时间
9		Slct tcp	Select 设置	8.23（单位：μs）	Select 设置所花时间

10		Sig hndl	捕获处理信号	2.52（单位：μs）	捕获处理信号所花的时间
11		Fork proc	Fork 进程后直接退出	647（单位：μs）	Fork 进程后址接退出所花的时间
12		Exec proc	Fork 后执行 execve 调用再退出	1873（单位：μs）	Fork 后执行 execve 调用再退出所花的时间
13		Sh proc	Fork 后执行 shell 再退出	4330（单位：μs）	Fork 后执行 shell 再退出所花的时间
14	Basic float operations	intgr bit / add/mul/div/mod	整数位操作／加／乘／除／求模操作	0.6300／0.0900／1.3100／14.1／13.5（单位：ns）	整数位操作，加，乘，除，等的运算所花时间
15	Basic float operations	Float/add/mul/div/bogo	浮点型操作／加／乘／	1.2500／2.2000／14.5／14.4（单位：ns）	浮点数操作，加，乘，除，等的运算所花时间
16	Basic double operations	Double add/mul/div/bogo	双精度数操作／加／乘／除	1.2500／2.5100／14.5／14.5（单位：ns）	双精度数操作，加，乘，除，等的运算所花时间
17	Context switching	2p/0k ctxsw	2 个并行处理 0K 大小的数据	3.660（单位：μs）	2 个并行处理 0K 大小的数据所花时间
18		2p/16k ctxsw	2 个并行处理 16K 大小的数据	2.8800（单位：μs）	2 个并行处理 16K 大小的数据所花的时间
19		2p/64k ctxsw	2 个并行处理 64K 大小的数据	73.00（单位：μs）	2 个并行处理 64K 大小的数据所花时间
20		8p/16k ctxsw	8 个并行处理 16K 大小的数据	5.8600（单位：μs）	8 个并行处理 16K 大小的数据所花时间
21		8p/64k ctxsw	8 个并行处理 64K 大小的数据	35.7（单位：μs）	8 个并行处理 64K 大小的数据所花时间

22		16p/16k ctxsw	16 个并行处理 16K 大小的数据	20.8（单位：μs）	16 个并行处理 16K 大小的数据所花时间
23		16p/64k ctxsw	16 个并行处理 64K 大小的数据	42.6（单位：μs）	16 个并行处理 64K 大小的数据所花时间
24	Local* Communication latencies	Pipe	本地管道通信延时	17.4（单位：μs）	本地管道通信延时时间
25		AF UNIX		13.7	
26		UDP		17.0	本地 UDP 通信延时时间
27		TCP		22.9	本地 TCP 通信延时时间
28		Tcp conn	TCP 建立 connect 并关闭描述字	106	TCP 建立连接并关闭所花时间
29	File & VM system latencies	0k file create/delete	0K 文件创建与删除	16.6 / 13.0（单位：μs）	0K 文件创建 / 删除所花的时间
30		10k file create/delete	10K 文件创建与删除	54.7 / 20.5（单位：μs）	10K 文件创建 / 删除所花的时间
31		Prot fault	保护页	0.577（单位：μs）	保护页延时时间
32		Page fault	缺页	4.03690（单位：μs）	缺页延时时间
33		100fd selct	对 100 个文档描述符配置 select	3.841（单位：μs）	对 100 个文档描述符配置 select 的时间
34	Local* Communication bandwidths	Pipe	本地通信带宽方面管道操作	759MB / s	本地通信带宽方面管道操作速度
35		File reread	文档重复读	1166.3MB/s	文档重复读取的速度
36		Mmap reread	内存映射重复读取	2846.5MB/s	内存映射重复读取速度
37		bcopy(libc)	内存拷贝	1125.7MB/s	内存拷贝使用 libc
38		bcopy(hand)	内存拷贝	1119.8MB/s	内存拷贝手工拷贝速度

39		Mem read	内存读	2874MB/s	内存读取速度
40		Mem write	内存写	1517MB/s	内存写入速度
41	Memory latencies	L1	L1 缓存	2.5060（单位：ns）	L1 缓存操作延时
42		L2	L2 缓存	12.5（单位：ns）	L2 缓存操作延时
43		Main mem	连续内存	56.5（单位：ns）	系统内存连续操作延时
44		Rand mem	内存随机访问延时	143.7（单位：ns）	系统内存随机访问操作延时