

pycod

很多人在说我们讲故事
吹NB 但是我们吹过的
NB , 正在——变为现实
!!!!

博客园 首页 博问 闪存 新随笔 - 订阅 RSS 管理 posts - 489, comments - 3, trackbacks - 0

Spec2006使用说明

Spec2006使用说明

五 10 十月 2014

By penglee

工具介绍

SPEC CPU 2006 benchmark是SPEC新一代的行业标准化的CPU测试基准套件。重点测试系统的处理器，内存子系统和编译器。这个基准测试套件包括的SPECint基准和SPECfp基准。其中SPECint2006基准包含12个不同的基准测试和SPECfp2006年基准包含19个不同的基准测试。SPEC设计了这个套件提供了一个比较标准的计算密集型，高性能的跨硬件的CPU测试工具。在SPEC CPU 2006基准有几种不同的方法来衡量计算机性能。一种方式是测量计算机完成单一任务的速度；另一种方式吞吐量，容量或速率的测量。说明：由于spec2006支持多种类型操作系统。以下安装、测试、移植等介绍均基于Unix 和其他的 Unix-like system如linux。Windows系统不在此范围内。

spec2006与spec2000的对比简要说明

2006年8月24日，标准性能评估公司（SPEC）宣布CPU2006将取代CPU2000。Spec cpu基准测试被广泛的应用于工业界和学术界。新套件比cpu2000要大很多，而且将涵盖到CPU、存储器系统、编译器特别是C++编译器。Cpu2000只有1个关于C++的基准测试，CPU2006则含有7个相关套件，这其中包括了50W行c++代码，Fortran和C语言和之前一样有很好的覆盖。

Spec2000和spec20006具体的套件信息如下：

Spec int2000 12个整点测试项目

- 164.gzip C语言编写，很流行的数据压缩程序。使用Lempel-Ziv coding(LZ77)作为压缩算法。SPEC中的gzip除了读入集外没有其他文件I/O操作，所有压缩和解压都在主存中进行，方便测试CPU和存储子系统。
- 175.vpr C语言编写，FPGA中的电路定位和路由程序,用于计算机辅助电路设计领域。
- 176.gcc: C语言编写，C语言优化编译器。该GCC基于GCC 2.7.2.2版本，为摩托罗拉88100处理器生成机器码。
- 181.mcf: C语言编写，组合优化/单点轮换调度(Single-depot vehicle scheduling),派生自使用单站轮换调度的公共物流应用程序。

昵称: pycod
园龄: 1年3个月
粉丝: 11
关注: 0
[+加关注](#)

2020年4月						
日	一	二	三	四	五	六
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

搜索

找找看

谷歌搜索

常用链接

- [我的随笔](#)
- [我的评论](#)
- [我的参与](#)
- [最新评论](#)
- [我的标签](#)

我的标签

- [root\(2\)](#)
- [桌面\(2\)](#)
- [自动\(1\)](#)
- [自动登录\(1\)](#)
- [登录\(1\)](#)
- [小米\(1\)](#)
- [用户\(1\)](#)
- [Z\(1\)](#)
- [CentOS\(1\)](#)
- [fedora21\(1\)](#)
- [更多](#)

随笔档案

[2020年4月\(43\)](#)

186.crafty: C语言编写, 国际象棋游戏程序。这是一个在64位机上设计的高性能计算机国际象棋程序, 在32位机上可以使用long long来表示该数据类型。主要是一个整点程序, 大量的逻辑运算。通过配置, 能生成不同的搜索集, 可以对处理器的整点运算, 转移预测和流水线等做测试。

197.parser: C语言编写, 字处理程序。连接语法分析器(Link Grammar Parser)是一个基于连接语法的英语句法分析器。连接语法时英语语法的较早理论。给定一个句子, 系统能给它赋予一个词法结构, 这个结构是两个词之间被标记的连接。该分析器有大约60000个词表的字典。

252.eon: C++语言编写, 计算机可视化。eon是一个基于概率的射线跟踪程序, 基于Kajiya 1986 ACM SIGGRAPH会议的论文。它将一定数量的3D射线发射到一个3D多面体中, 计算射线和多边形的交点, 并通过计算入射光和交点生成新射线。最终的计算结果是一个能在照相机内看到的图片。程序中的计算需求和基本计算机图形学中介绍的传统确定性射线跟踪程序很相似, 但是它有较少的存储一致性需求, 因为很多在同一个位置生成的随机射线横切3D空间中完全不同的区域。输入: eon渲染一个150×150像素的一把椅子在房间前脚的图片。

253.perlbnk: C语言编写, Perl语言解释器。这是一个Perl v5.005_03版本的删减版, 删除了很多操作系统相关的特性, 并增加了一些第三方的模块, 包括MD5 v1.7, MHonArc v2.3.3, IO-stringy v1.205, MailTools v1.11, TimeDate v1.08。输入: 由四个perl脚本组成, 一个email-to-HTML 转换器, 占用时间最长; 一个spec CPU2000用来对比较输出的diff脚本, 一个通过迭代算法寻找合格数字的脚本, 一个用于测试伪随机数按照预定顺序生成的脚本。输出分别对应各个输入。

254.gap: C语言编写, 群论, 解释器中常用。它实现一个语言和库用于做群论计算。GAP是(Groups, Algorithms和Programming的缩写)。

255.vortex: C语言编写, 单用户面向对象数据库中事务处理的benchmark。SPEC中的vortex来自完整面向对象数据库程序VORTEX的一个子集。输入和输入数据库的事务都会被转换成一个方案。

256.bzip2: C语言编写, 压缩程序。基于Julian Seward的bzip2 v0.1, 唯一的区别是SPEC中的bzip2除了输入外, 无文件I/O。所有的压缩和解压都在内存中进行。输入, 输出都和gzip的相似。

300.twolf: C语言实现, 定位和路由模拟。twolf来自TimberWolfSC定位和全局路由, 它被用于创建微处理器产品印刷电路布局的流程中, 确定晶体管的位置和组内的全局连接。定位问题是一个排列, 穷举法耗时太长, TimberWolfSC程序则使用模拟退火作为启发来找到较优解。

Spec int2006 12个整点测试项目

400.perlbench: C语言编写, 同spec2000的253.perlbnk类似, 是Perl V5.8.7版本的删减版, 同样删除了一下操作系统特性有关的东西, 增加了第三方的模块, 如: SpamAssassin V2.6.1, Digest-MD5 v2.33, HTML-Parser v3.35, MHonArc V2.6.8, IO-stringy V1.205, MailTools V1.60, TimeDate V1.16。

400.perlbench采用3个脚本进行增加负载或工作。1.主要的负载在于打开 垃圾邮件过滤软件SpamAssassin。采取的过滤方式是采用记分制, 也就是说会根据我们所设定的标准来给予分数超过标准值的时候即判定为 SPAM (垃圾邮件), 其余为非垃圾邮件。但对SpamAssassin软件做了很大程度的修改, 避免一些I/O操作, 并且没有使用贝叶斯过滤。2.是比较流行的email到HTML的转换器MHonArc 3.使用了spec2006的工具套件specdiff。

401.bzip2: C语言实现, 同spec2000的256.bzip2类似, 基于Julian Seward的bzip2 v1.03。唯一的区别是SPEC中的bzip2除了输入外, 无文件I/O。所有的压缩和解压都在内存中进行。输入, 输出都和gzip的相似。负载包括六个部分: 两个小的JPEG图片、一个程序、一个tar包起的几个源程序文件、一个HTML文件、混合文

[2020年3月\(72\)](#)
[2020年2月\(15\)](#)
[2020年1月\(24\)](#)
[2019年12月\(31\)](#)
[2019年11月\(68\)](#)
[2019年10月\(59\)](#)
[2019年9月\(54\)](#)
[2019年8月\(8\)](#)
[2019年7月\(23\)](#)
[2019年6月\(37\)](#)
[2019年5月\(27\)](#)
[2019年4月\(6\)](#)
[2019年3月\(5\)](#)
[2019年2月\(7\)](#)
[2019年1月\(6\)](#)

最新评论

[1. Re:Pycharm2019.2.1永久激活](#)

大神求公众号

--曲径浅缘

[2. Re:Pycharm2019.2.1永久激活](#)

公众号?

--yangmiemie0352

[3. Re:Pycharm2019.2.1永久激活](#)

请问大佬公众号是多少哇

--kuck

阅读排行榜

[1. 【转载】在MySQL登录时出现Access denied for user 'root'@'localhost' \(using password: YES\) 拒绝访问, 并可修改MySQL密码\(48694\)](#)
[2. Pycharm2019.2.1永久激活\(12664\)](#)
[3. 优启通-PE启动盘制作工具 原版Win7系统安装超详细教程!!!! \(9854\)](#)
[4. Linux下查看某一进程所占内存的方法\(6357\)](#)
[5. SPEC CPU2006 Spec2006 使用说明\(5237\)](#)

评论排行榜

[1. Pycharm2019.2.1永久激活\(3\)](#)

推荐排行榜

件, 包括压缩起来的高可压缩文件及不怎么可压缩的文件。测试分别使用了三个不同的压缩等级进行压缩和解压缩

403.gcc:C语言实现, 同spec2000的176.gcc类似。该测试是基于gcc V3.2,为AMDCPU生成机器码。相比spec2000的176.gcc, 该测试有更多的输入文件, 因此测试压力会更大, 负载来源于对9组C代码进行编译。

429.mcf:C语言实现, 同时需要libm库支持, 同spec2000的181.mcf类似, MCF是一个用于大型公共交通中的单站车辆调度的程序。但对输入文件做了一定的修改, 由32位变成64位, 用以兼容64位系统。并且增加了cache命中和程序的性能。相比spec2000的181.mcf来说, 占用的内存由之前的100M-190M变为860M-1700M。

445.gobmk: C语言实现, 同spec2000的186.crafty类似。不同的是这里是实现的围棋游戏。相比spec2000程序更复杂。

456.hmmmer: C语言实现。HMMER是基于隐马尔可夫模型(profile HMMs), 用于生物序列分析工作。同Timed HMMer Search 类似。

458.sjeng: C语言实现。基于一种象棋游戏Sjeng11.2, 属于人工智能的范畴。

462.libquantum:C语言实现 (C99)。libquantum是模拟量子计算机的库文件, 用来进行量子计算机应用的研究。

464.h264ref:C语言实现。一种视频压缩程序, 基于H264AVC 9.3版, 去除了I/O和平台相关的东西。

471.omnetpp:C++语言实现。OMNeT++, 离散事件仿真。包括约8000台计算机和900个交换机/集线器, 以及混合了各种从10Mb到1000Mb速率的大型CSMA/CD协议以太网网络模拟。

473 astar: C++语言实现, 实现了2D寻路算法A*的三种不同版本。

1. xalancbmk: C++语言实现, 包含Xalan-C++ v1.8 mod、Xerces-C++ v2.5.0, 实现了XML文档/XSL表到HTML文档的转换。

Spec fp2000 13个浮点测试项目

168 wupwise: Fortran 77语言实现, 格点规范理论与威尔逊数值模拟对费米子格是夸克传播子的计算。输入文件wupwise.in包含几个创建该问题的参数, SEED:提供一个随机数 NITER: 迭代次数KAPPA: 参数表征矩阵 $M = I - KAPPA * D$ 。输出文件wupwise.out,包含每次NITER迭代的校验值 (EPS) 和包含最终的EPSILON特征的结果。

171 swim: Fortran 77语言实现, 气象学中的浅水模型。

172.mgrid : Fortran 77语言实现, 多重网格求解器: 3D势场

173.applu: Fortran 77语言实现, 抛物线/椭圆型偏微分方程

177.mesa: C语言实现, 3D图形库。

178.galgel :Fortran 90语言实现, 计算流体动力学

179.art :C, 图像识别/神经网络

183.quake:C语言实现, 地震波传播模拟

187.facerec: Fortran 90语言实现, 图像处理: 人脸识别

189.lucas :Fortran 90语言实现, 数论/素性测试

191.fma3d :Fortran 90语言实现, 有限元模拟碰撞

200.sixtrack :Fortran 77语言实现, 高能核物理加速器设计

1. [【转载】在MySQL登录时出现Access denied for user 'root'@'localhost' \(using password: YES\) 拒绝访问, 并可修改MySQL密码\(1\)](#)
2. [fedora21 桌面用户自动登录lighdm.conf -20190520 方法\(1\)](#)
3. [【转载】Linux查看PCIe版本及速率【方法】PCIe的X4X8X16 查看 数量 怎么看\(1\)](#)
4. [SPEC CPU2006 Spec2006 使用说明\(1\)](#)
5. [Pycharm2019.2.1永久激活\(1\)](#)

301.apsi: Fortran 77语言实现, 气象学中污染物分布

Spec fp2006 17个浮点测试项目

410.bwaves: Fortran 77语言实现, 流体力学范畴。对三维瞬跨音速粘性流中冲击波的模拟计算

416.gamess: Fortran语言实现, 量子化学范畴。采用GMAESS组件, 包含 胞嘧啶分子、水和Cu²⁺离子、三唑离子等3种SCF自洽场计算。

433.milc: C语言实现, 量子力学范畴, 采用MILC组件, 包含用来研究QCD量子色动力学、夸克及胶子的四维SU(3)格点规范理论的模。

434.zeusmp : Fortran 77/REAL*8语言实现, 属于计算流体力学范畴, ZEUS-MP程序, 用来计算理想、非相对论条件下的流体力学和磁流体力学, 434.zeusmp模拟计算了一个统一磁场中的3D冲击波。

435.gromacs : C & Fortran语言实现, 属于生物化学/分子力学范畴。GROMACS是一个分子力学计算套件, 然而也可以用于非生物系统, 435.gromacs模拟了在一个水和离子溶液中的蛋白质溶菌酶结构在各种实验手段如核磁共振的X光照射下的变化。

436.cactusADM: Fortran 90, ANSI C语言实现, 物理中的广义相对论。来源于Cactus BenchADM。436.cactusADM对时空曲率由内部物质决定的爱因斯坦演化方程进行求解, 爱因斯坦演化方程由10个标准ADM 3+1分解的二阶非线性偏微分方程组成。

437.leslie3d : Fortran 90语言实现, 属于流体力学范畴。LESLIE3d是用来计算湍流的计算流体力学程序, 437.leslie3d计算了一个如燃油注入燃烧室的时间分层混合流体。

444.namd: C++语言实现。属于生物/分子学范畴。NAMD是一个大型生物分子系统并行计算程序, 444.namd模拟了了92224个原子组成的A-I载脂蛋白

447.dealII: C++语言实现。进行有限元分析, w/Boost lib、deal.II lib、deal.II是定位于自适应有限元及误差估计的C++库, 447.dealII对非常系数的亥姆霍兹方程进行求解, 它使用了基于二元加权误差估计生成最佳网格的自适应方法, 该方程在3维得解。

450.soplex:C++语言实现。采用Plex v1.2.1程序, Plex使用单纯形算法解线性方程

453.povray: ISO C++ 语言实现。POV-Ray POV-Ray是一个光线追踪渲染软件, 453.povray渲染一幅1280x1024的反锯齿国际象棋棋盘图像

454.calculix: Fortran 90 & C语言实现, 结构力学范畴。CalculiX是一个用于线性及非线性三位结构力学的有限元分析软件, 454.calculix计算了一个高速旋转的压缩盘片在离心力的作用下的应力和变形情况

459.GemsFDTD: Fortran 90语言实现。属于电磁学范畴。 459.GemsFDTD使用FDTD (有限差分域) 方法求解三维域中的麦克斯韦方程, 计算了一个理想导体的雷达散射截面

465.tonto: Fortran 95实现。量子化学范畴Tonto是一个面向对象的量子化学程序包, 465.tonto计算面向量子晶体学, 它基于一个符合X光衍射实验数据的、约束的分子Hartree-Fock波函数

470.lbm : ANSI C语言实现。 470.lbm使用LBM (格子波尔兹曼方法) 模拟非压缩流体, 它模拟了两种情况: 类似活塞推动的剪切驱动流体和管道流体, 测试包含了3000个步骤

481.wrf : Fortran 90 & C语言实现。WRF v2.0.2 481.wrf基于WRF(Weather Research and Forecastin)模型, 对NCAR的数据进行了计算, 数据包括了UTC 2001.06.11到UTC 2001.06.12以三小时为间隔的数据

482.sphinx3: C语言实现。Sphinx-3一种语音识别软件。

工具安装

标准安装方式

```
$mount -t iso9660 -o ro,exec /dev/cdrom /mnt
$cd /mnt
$./install.sh
```

可以指定安装目录如:

```
$./install.sh -d /home/cpu2006
```

安装过程中可能遇到以下提示:

```
Installing FROM /Volumes/SPEC_CPU2006
Installing TO /home/cpu2006
Is this correct? (Please enter 'yes' or 'no')
yes
```

通过安装完成的软件进行测试

先准备相关测试文件, 如: cpu2006-20141030.tar。解压文件:

```
$tar xf cpu2006-20141030.tar
```

进入安装目录:

```
$cd cpu2006v99/ (根据实际路径)
```

导入测试工具:

```
$ shrc
```

注意: 输入时, 圆点与shrc之间至少一个空格, 该命令开始导入SPEC CPU2006程序。修改测试文件绝对路径:

```
$bin/relocate
显示 "Top of SPEC benchmark tree is '/cpu2006v99' "
```

运行测试

根据需求, 进行参数选择, 开始运行测试脚本: 例如: runspec -c linux-mipsel-gcc -i ref -n 3 all 命令参数说明:

-c: 读取测试配置文件, linux-mipsel-gcc位于SPEC CPU2000安装目录的config目录下, 该文件可以根据实际信息进行修改。其编译选项不建议修改。
-i: 输入测试数据集的大小, ref代表最大测试数据集, test代表最小测试数据集;
-n: 每个测试项目运行的次数, 如果需要SPEC CPU2006自动计算测试分值, 需要指定运行次数等于或大于3, 即n>=3;
all: SPEC CPU2006将运行基准测试程序中的所有测试项目。

出现以下信息, 表明测试已经启动"

```
runspec v4283 - Copyright 1999-2006 Standard Performance Evaluation
Corporation
Using 'mipsel-linux' tools
Reading MANIFEST... 17899 files
Loading runspec modules.....
```



```

Locating benchmarks...found 31 benchmarks in 13 benchsets.
Locating output formats: ASCII, config, CSV, flags, html (DISABLED), mail,
PDF, PostScript, raw, Screen, Submission Check
Reading config file '/cpu2006v99/config/linux-mipsel-gcc.cfg'
Benchmarks selected: 400.perlbench, 401.bzip2, 403.gcc, 429.mcf,
445.gobmk, 456.hammer, 458.sjeng, 462.libquantum, 464.h264ref, 471.omnetpp,
473.astar, 483.xalancbmk, 999.specrand, 410.bwaves, 416.gamess, 433.milc,
434.zeusmp, 435.gromacs, 436.cactusADM, 437.leslie3d, 444.namd,
447.dealIII, 450.soplex, 453.povray, 454.calculix, 459.GemsFDTD, 465.tonto,
470.lbm, 481.wrf, 482.sphinx3, 998.specrand
Compiling Binaries
  up to date 400.perlbench ref base linux-mipsel-gcc default
  Up to date 401.bzip2 ref base linux-mipsel-gcc default
  Up to date 403.gcc ref base linux-mipsel-gcc default
  Up to date 429.mcf ref base linux-mipsel-gcc default
.....

```

这组测试配置基本和spec2000类似，不同的是，spec2006在测试前，会把每个测试组件重新编译建立一次。所以在测试前，应确保此时的系统中包含以下编译器：
g++、gcc、gfortran编译器。如果进行连续测试，则不必进行再次编译，可以加参数“-nobuild”。附config文件大体说明：斜体部分为文本内容 //为解释说明

```

#####
ignore_errors = yes
tune           = base
ext            = loongson.3a2000
output_format = asc, Screen, html
reportable     = 0
teeout         = yes
teerunout      = yes
hw_avail       = Feb-2015
sw_avail       = Feb-2015
license_num    =
company_name   = Loongson
prepared_by    = Loongson
tester_name    = isoft
test_date      = July-2016

verbose        = 9

default=default=default=default:
#####
#
# Compiler selection
#
#####

makeflags = -j 4

CC          = mips64el-loongson-linux-gcc
CXX         = mips64el-loongson-linux-g++
FC          = mips64el-loongson-linux-gfortran

## HW config
hw_cpu_name = Loongson

```

```

hw_model      = 3A2000
hw_cpu_mhz    = 1000
hw_fpu        = Integrated
hw_nchips     = 1
hw_ncores    = 4
hw_ncoresperchip= 4
hw_nthreads   = 4
hw_ncpuorder  = 1
hw_parallel   = yes
hw_pcache     = 64KB I + 64KB D
hw_scache     = 4096KB(I + D)
hw_memory     = 2 x 4GB DDR1066
hw_vendor     = Loongson

## SW config
sw_os         = iSoft
sw_file       = ext4
sw_state      = runlevel 3
sw_compiler   = gcc , g++ & gfortran 4.8

#####
# Optimization
#####
## Base is low opt
default=default=default=default:
# For this data model, -m32 needs to be here, as one of the lines below
OPTIMIZE      = -static -march=loongson3a
PORTABILITY    =

COPTIMIZE     = -flto -Ofast
CXXOPTIMIZE   = -flto -Ofast
FOPTIMIZE     = -flto -Ofast

notes0100= C/C++ base flags:  -flto -Ofast
notes0110= Fortran base flags: -flto -Ofast

#####
# 32/64 bit Portability Flags - all
#####

default=base=default=default:
notes25=

default=peak=default=default:
notes25=

#####
# Portability Flags - INT
#####

400.perlbench=default=default=default:
notes35      = 400.perlbench: -DSPEC_CPU_LINUX_IA32
CPORTABILITY= -DSPEC_CPU_LINUX_IA32 -fno-strict-aliasing
#notes35     = 400.perlbench: -DSPEC_CPU_LINUX

```

```
#CPORTABILITY= -DSPEC_CPU_LINUX -DSPEC_CPU_LITTLEENDIAN
#COPTIMIZE      = -O2
.....
```

其他具体组件的编译信息不再一一说明，基本可以保持默认即可。

更多测试类型与方法参见下文。

测试结果收集与分析

测试结束后在result目录下，查看测试结果，测试结果一般以测试类型和测试次序进行区别，并且会按照整点和浮点进行划分。如果是按照正式reportable的模式进行测试的话，会出一份网页版的正式测试结果，参见下图：非正式结果可以去asc或txt文档查看，格式例如下文：

Benchmarks	Base Ref.	Base Run Time	Base Ratio	Peak Ref.	Peak Run Time	Peak Ratio
-----	-----	-----	-----	-----	-----	-----
400.perlbench	9770	430	22.7 S	9770	339	
28.8 S						
400.perlbench	9770	429	22.8 *	9770	338	
28.9 S						
400.perlbench	9770	429	22.8 S	9770	339	
28.8 *						
401.bzip2	9650	559	17.3 *	9650	552	
17.5 *						
401.bzip2	9650	559	17.3 S	9650	553	
17.5 S						
401.bzip2	9650	560	17.2 S	9650	552	
17.5 S						
403.gcc	8050	374	21.5 S	8050	337	
23.9 S						
403.gcc	8050	367	22.0 *	8050	337	
23.9 S						
403.gcc	8050	366	22.0 S	8050	337	
23.9 *						
429.mcf	9120	187	48.9 S	9120	187	
48.9 S						
429.mcf	9120	188	48.6 *	9120	188	
48.6 *						
429.mcf	9120	211	43.2 S	9120	211	
43.2 S						
445.gobmk	10490	559	18.8 S	10490	542	
19.4 S						
445.gobmk	10490	559	18.8 S	10490	542	
19.4 *						
445.gobmk	10490	559	18.8 *	10490	541	
19.4 S						
456.hmm	9330	213	43.8 S	9330	213	
43.7 *						
456.hmm	9330	247	37.8 S	9330	214	
43.6 S						
456.hmm	9330	214	43.5 *	9330	213	
43.7 S						
458.sjeng	12100	616	19.6 *	12100	574	
21.1 *						

458. sjeng	12100	617	19. 6	S	12100	575	
21. 0	S						
458. sjeng	12100	585	20. 7	S	12100	574	
21. 1	S						
462. libquantum	20720	14. 2	1460	*	20720	14. 2	1460
*							
462. libquantum	20720	14. 4	1440	S	20720	14. 4	1440
S							
462. libquantum	20720	13. 5	1540	S	20720	13. 5	1540
S							
464. h264ref	22130	578	38. 3	*	22130	527	
42. 0	S						
464. h264ref	22130	579	38. 2	S	22130	526	
42. 0	*						
464. h264ref	22130	578	38. 3	S	22130	526	
42. 0	S						
471. omnetpp	6250	349	17. 9	S	6250	288	
21. 7	*						
471. omnetpp	6250	348	17. 9	*	6250	288	
21. 7	S						
471. omnetpp	6250	348	18. 0	S	6250	289	
21. 7	S						
473. astar	7020	313	22. 4	S	7020	313	
22. 4	S						
473. astar	7020	311	22. 6	S	7020	311	
22. 6	S						
473. astar	7020	311	22. 6	*	7020	311	
22. 6	*						
483. xalancbmk	6900	174	39. 7	*	6900	170	
40. 7	S						
483. xalancbmk	6900	174	39. 6	S	6900	170	
40. 5	S						
483. xalancbmk	6900	174	39. 7	S	6900	170	
40. 6	*						
=====							
=====							
400. perlbench	9770	429	22. 8	*	9770	339	
28. 8	*						
401. bzip2	9650	559	17. 3	*	9650	552	
17. 5	*						
403. gcc	8050	367	22. 0	*	8050	337	
23. 9	*						
429. mcf	9120	188	48. 6	*	9120	188	
48. 6	*						
445. gobmk	10490	559	18. 8	*	10490	542	
19. 4	*						
456. hammer	9330	214	43. 5	*	9330	213	
43. 7	*						
458. sjeng	12100	616	19. 6	*	12100	574	
21. 1	*						
462. libquantum	20720	14. 2	1460	*	20720	14. 2	1460
*							
464. h264ref	22130	578	38. 3	*	22130	526	
42. 0	*						
471. omnetpp	6250	348	17. 9	*	6250	288	

21.7 *					
473. astar	7020	311	22.6 *	7020	311
22.6 *					
483. xalancbmk	6900	174	39.7 *	6900	170
40.6 *					
SPECint (R)_base2006			36.7		
SPECint2006			39.1		

Base Ratio为测试分值，该测试一般运行3遍取中值。整体结果为所有结果的几何平均值。

举例说明SPEC CPU2006常用测试命令组合

1. `runspec -c test.cfg -i test -I all` 基于最小测试数据集快速执行所有的测试，测试过程中如果某个用例发生错误，则跳过错误用例，继续执行其他用例。 2. `runspec -c test.cfg -i ref -n 3 -I all` 基于最大测试数据集全面执行所有的测试，用于测试单核CPU，测试过程中如果某个用例发生错误，则跳过错误用例，继续执行其他用例。 3. `runspec -c test.cfg -r ref -n 3 fp` 基于最大测试数据集，只运行fp测试 4. `runspec -c test.cfg -r ref -n 3 int` 基于最大测试数据集，只运行int测试 5. `runspec -c test.cfg -i ref 473.astar` 基于最大测试数据集只执行473.astar单个测试。 6 `runspec -c test.cfg -i ref --rate 4 int` 基于最大数据集进行rate测试，运行4线程测试的分值

其他命令补充

`--action build` 创建测试项目
`--action onlyrun` 只是运行，不进行确认测试结果是否正确
`--action report` 标准测试
`--action clean` 清理已经建立的测试项目

模式很多，不再一一介绍。

`--check_version` 检查版本更新
`--copies -c number` 默认1 用来进行SPECrate run
`--help` 显示帮助文档
`--ignore_errors` 忽略错误
`ations number -n` 测试次数，一次reportable测试，必须是3次
`--iter--loose` 未设置reportable测试
`--output_format` 输出文件格式 默认为html text 可以选择 all cfg test csv html mail pdf ps printer raw screen text 等多种方式。
`--rate -r` 进行SPECspeed测试可以指定copies数量运行。
`--rebuild` 重新编译测试文件
`--tune tuning` 默认是base 可以选择base, peak, or all , Report 首先是base, 其次是peak.

以上是比较常用的命令，一些不常用的命令不再介绍，可参考下文了解。

`-a` Same as `--action`
`--action action` Do: build|buildsetup|clean|clobber|configpp|onlyrun|realclean|report|run|runsetup|scrub| setup|trash|validate
`--basepeak` Copy base results to peak (use with `--rawformat`)
`--nobuild` Do not attempt to build binaries
`-c` Same as `--config`
`-C` Same as `--copies`
`--check_version` Check whether an updated version of CPU2006 is available
`--comment "text"` Add a comment to the log and the stored configfile.
`--config file` Set config file for runspec to use
`--copies` Set the number of copies for a SPECrate run
`-D` Same as `--rebuild`

```

-d Same as --deletework
--debug Same as --verbose
--define SYMBOL[=VALUE] Define a config preprocessor macro
--delay secs Add delay before and after benchmark invocation
--deletework Force work directories to be rebuilt
--dryrun Same as --fake
--dry-run Same as --fake
-e Same as --extension
--ext Same as --extension
--extension ext[,ext...] Set the extensions
-F Same as --flagsurl
--fake Show what commands would be executed.
--fakereport Generate a report without compiling codes or doing a run.
--fakereportable Generate a fake report as if "--reportable" were set.
--[no]feedback Control whether builds use feedback directed optimization
--flagupdate Same as --update
--flagsupdate Same as --update
--flagsurl url Location (url or filespec) where to find your flags file
--getflags Same as --update
--graph_auto Let the tools pick minimum and maximum for the graph
--graph_min N Set the minimum for the graph
--graph_max N Set the maximum for the graph
-h Same as --help
--help Print usage message
--http_proxy Specify the proxy for internet access
--http_timeout Timeout when attempting http access
-I Same as --ignore_errors
-i Same as --size
--ignore_errors Continue with benchmark runs even if some fail
--ignoreerror Same as --ignore_errors
.....

```

手动编译及跨平台移植方法（仅供参考）

Spec2006包含的测试套件基本上与平台无关，因此理论上是可以直接进行跨平台进行测试。Spec2006为了实现测试结果的准确及降低对平台的依赖，制作了自己的工具包。其中包括：specdiff (examines results to see if the correct answer was obtained)、specinvoke (invokes benchmarks for CPU2006)、specmake (GNU make, Calling it "specmake" avoids possible conflicts with versions of make that may already be on your system. SPEC requires that published results use the versions of the tools that it supplies, so that if SPEC applies patches or extensions from time to time, all users run with a consistent tool set. Similar considerations apply to other tools in this list.)、specxz (The xz compression program)、specmd5sum (md5sum from GNU coreutils, with enhancements)、spectar (GNU tar)、specperl (Perl)。Various Perl modules such as Algorithm::Diff, GD, HTML::Parser, PDF::API2, URI, XML::SAX, etc.。编译工具：C99 编译器 C++编译器 如果是初始安装包，则需要解压源码包

```
# xz -dc install_archives/cpu2006.tar.xz | tar -xf - tools/src
```

建立工具包的命令如下：

```

AIX (32-bit):
CC=xlc CFLAGS='-O -qarch=ppc -D_LARGE_FILES -D_ILS_MACROS' PERLFLAGS="-
Doptimize='-O' -Accflags='-D_LARGE_FILES'" OBJECT_MODE=32 ./buildtools

```

AIX (64-bit):

```
CC=xlc CFLAGS='-O3 -qarch=ppc64 -q64 -D_ILS_MACROS' PERLFLAGS="--
Doptimize='-O' -Aoptimize='-qarch=ppc64 -q64' " OBJECT_MODE=64 ./buildtools
```

HPUX:

```
CC="/opt/ansic/bin/cc" CFLAGS="-Ae +O2 +z -D_FILE_OFFSET_BITS=64 -AC99"
XZCFLAGS=-AC99 PERLFLAGS="--Doptimize=+O1 -Dlibpth=/usr/lib/hpux32 -
Dldflags=-lm -Dcppstdin=cc -Acppstdin=-E -Acppstdin=-Aa -Acppstdin=-
D__STDC_EXT__ -Acppstdin=-AC89" ./buildtools
```

IRIX:

```
ulimit -s 128000; CFLAGS="-c99 -D_FILE_OFFSET_BITS=64" MD5CFLAGS="--O2"
PERLFLAGS="--Doptimize=-O3 -Uloclib -Dlibpth=/usr/lib32" XZFLAGS="--
disable-assembler --disable-shared" XZCFLAGS="-O1" ./buildtools
```

建立完成之后，确认是否成功的方法：

```
cd $SPEC (Unix)
```

```
. shrc
```

```
或 runspec -V
```

```
或 specmake -v
```

```
specxz -h
```

```
specperl -v
```

```
specdiff -h
```

```
runspec -h
```

```
specinvoke -h
```

工具编译完成之后，需要对工具进行打包

```
packagetools <archname>
```

如果自动化编译出现问题，可以手动编译每个测试工具。编译方法为进入对应的工具包，手动编译。

[Top^](#)

Comments

好文要顶

关注我

收藏该文



pycod

[关注 - 0](#)

[粉丝 - 11](#)

0

0

[+加关注](#)

« 上一篇: [vipvp常用linux命令](#)

» 下一篇: [倒车 车头是盲区 车头撞墙](#)

posted on 2019-09-21 23:35 [pycod](#) 阅读(640) 评论(0) [编辑](#) [收藏](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) [网站首页](#)。

[【推荐】超50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库](#)

[【推荐】腾讯云产品限时秒杀，爆款1核2G云服务器99元/年！](#)

相关博文：

- [Ubuntu16.04安装QQ \(图文说明\)](#)
- [《数据结构-C语言版》\(严蔚敏,吴伟民版\) 课本源码+习题集解析使用说明](#)
- [Valgrind使用说明](#)
- [修复MBR的工具MBRFix下载](#)
- [MinGW安装和使用](#)
- » [更多推荐...](#)

最新 IT 新闻：

- [虎扑直男“炮轰”B站UP主：1千万播放量到底该捐多少](#)
- [用着我一样的设备 NASA科学家们在家操控火星探测器](#)
- [腾讯安全与北京市方正公证处战略合作](#)
- [AI做“军师”？先赢过Reddit用户再说吧](#)
- [星星海SA2云服务器助力腾讯广告“千人千面”，广告计算提速25%](#)
- » [更多新闻...](#)

Copyright © 2020 pycod

Powered by .NET Core on Kubernetes Powered By[博客园](#)