# 附加实验 perf性能剖析工具

## 实验目的和内容

Perf是linux内核中集成的一款性能剖析工具，目前使用十分广泛，本实验让学生初步掌握perf的基本使用方法以及相关指令。

主要掌握perf的几个基础命令，如perf top , perf stat, perf record/report。后面我们会详细介绍这几个命令。

## 实验环境：

Linux系统，内核在2.6+。

必须在实体机上的linux使用，虚拟机不支持。

perf工具使用

学习使用perf工具，了解CPU硬件事件计数器（CPU周期、stall次数，指令数，CPI等），

附加实验主要完成三个任务：

1. 性能剖析工具perf的安装及认识
2. perf性能指标学习
3. perf相关指令学习以及性能分析

具体实验方案细节见所附的参考文档《perf性能剖析.docx》

## 实验步骤以及说明

* Perf安装，安装perf非常简单，只需要内核版本高于2.6.31.perf的安装分为源码安装和yum源安装。

1. yum源安装只需要在终端输入命令（或 sudo apt-get install perf）
2. 源代码安装，由于perf已经加入到内核功能中，所以在内核的源代码目录下的tool/perf/目录下就是perf的源代码，在该目录下输入make make install即可（若make出错，可根据相关提示yum安装依赖包）

* perf list查看用户当前硬件平台上支持的所有性能事件，如下图，每行后面的括好中的信息表示该事件是硬件事件、软件事件还是Tracepoint events

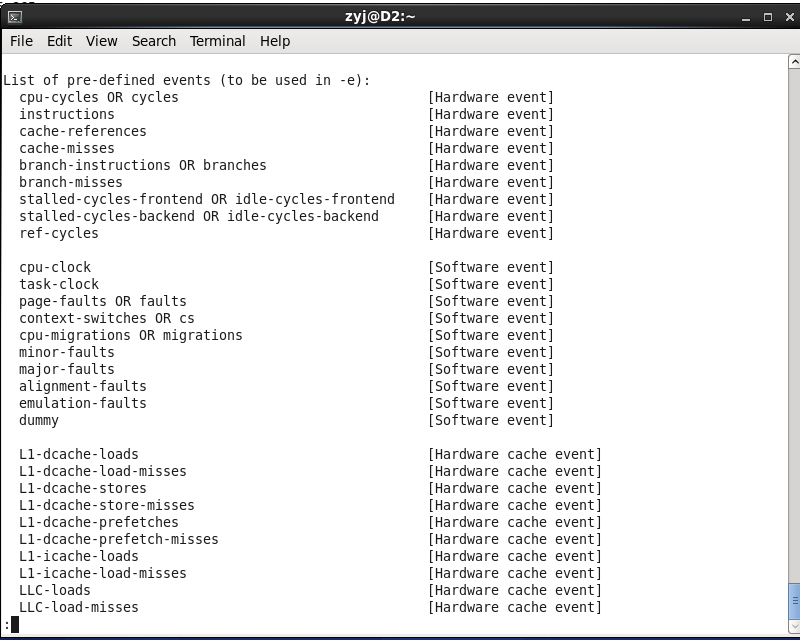


图1.perf list 输出结果

表一、perf基本指令

|  |  |
| --- | --- |
| Perf list | 用途：主要用来查看当前软硬件环境支持的性能事件，性能事件与CPU及内核版本有关  使用方法： perf list |
| Perf stat | 用途：主要用来分析程序的整体性能  使用方法： perf stat ./youprog |
| Perf top | 用途：实时显示系统/进程的性能统计信息  使用方法：perf top |
| Perf record | 用途：记录一段时间内系统/进程的性能事件  使用方法：perf record [options][<command>]  Perf record [options]--<command>[options] |
| Perf report | 用途：读取perf record生成的perf.data文件，并显示分析数据  使用方法：perf report [-I <file> | --input=file] |

接下来我们将通过一个简单的例子让同学们对上述命令有个直观的认识

* **实例一**

首先创建一个测试程序test1.c，其代码如下，其中函数longa()是个很长的循环，比较耗时。函数fun1和fun2分别调用函数longa十次和一百次，

下面就此程序进行剖析

首先编辑好该程序并编译运行，在编译时加入编译选项-g来保留调试信息



代码清单：

//test1.c

void waste\_cpu()

{

int i,j=1;

for(i = 0; i < 1000000; i++)

j=i\*j;

}

void fun1()

{

int i;

for(i=0 ; i < 10; i++)

waste\_cpu();

}

void fun2()

{

int i;

for(i = 0; i< 100; i++)

waste\_cpu();

}

int main(void)

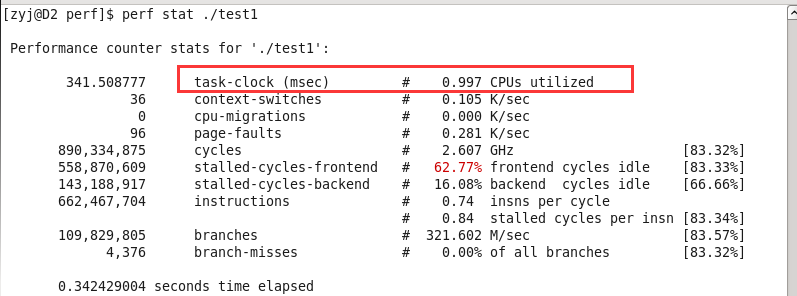
{

fun1();

fun2();

}

编译好后，下面还是简单的使用perf来剖析该段代码



图二、使用perf stat简单的剖析程序性能信息

* 如图，**使用perf stat简单的剖析程序性能**，图中task-clock接近1，表明该程序是一个计算密集型程序。

表二、图二中各个时间详细说明如

task-clock：目标程序使用CPU的时间，CPUs utilized 表示程序的CPU占用率，该值越高，表明是CPU密集型程序

context-switches:进程切换次数，前面数字是切换次数，后面是平均每秒发生次数，频繁的切换降低程序性能

cpu-migrations:处理器迁移，即为了维持CPU负载均衡，将进程从一个CPU迁移到另一个CPU

page-fault:缺页异常

cycles:消耗处理器周期数

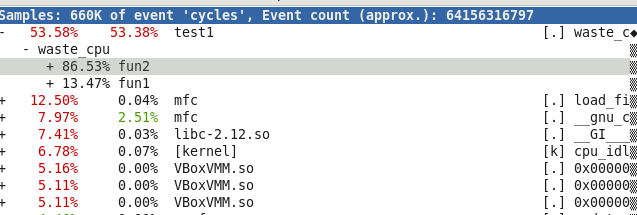
instructions:程序执行期间产生的处理器指令数，值越大越好，表明程序充分的利用处理器特征

branches:执行过程中遇到的分支指令数

branch-misses:分支预测错误次数

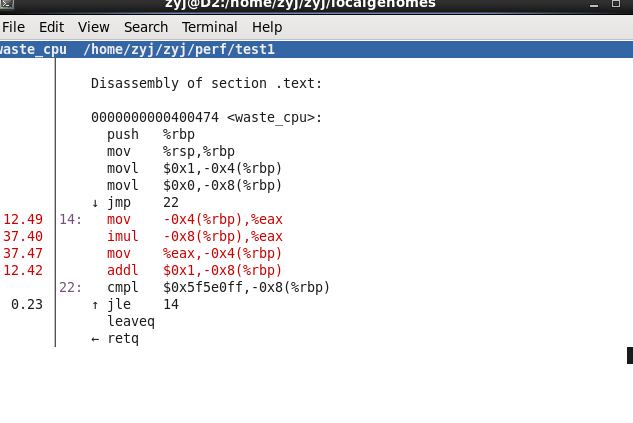
* **利用perf辨识程序热点**

首先可以通过perf top（其功能类似于top指令）查看实时的热点代码信息，如下图



左侧第一列为该符号的热度，中间是该程序名，右侧最后一列为函数名，

通过a命令可以看到选中函数的汇编代码信息。当然，这只是初略的判断，并且对于时间较短的程序很难捕捉到。



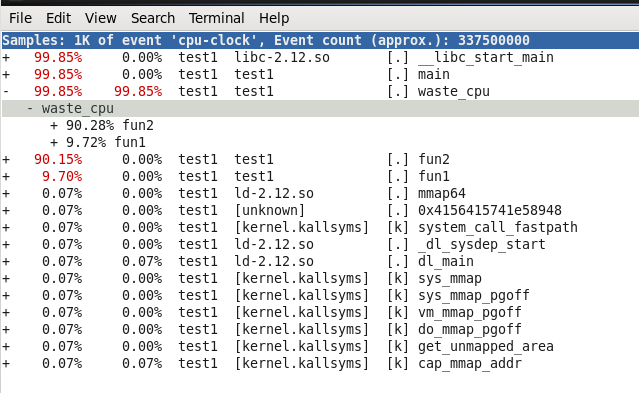
* **perf record 可以记录一段时间内进程的性能信息**

通过指令record 如下图



其中-e 指定所需记录的事件，-g 指显示调用图信息，默认生成的报告文件为perf.data。

现在通过perf report读取上述指令记录的性能信息(ls查看文件为perf.data)，命令如下



通过上图可以看到主要的热点代码为waste\_cpu，其中被fun2调用的次数远多于fun1调用的次数。所以代码中的waste\_cpu和fun2是主要优化的重点。

希望大家能详细的学习下 perf record 和 perf report这对指令

**结束语**

当然以上介绍只是一个初步的简介性质的，pref的功能远不止这些，感兴趣的同学可以按参考文献中的链接进行深入的学习，研究perf一些高级功能。同时可以结合自己之前写的代码，使用perf剖析其性能并进行改进。

参考文献：

IBM Developerworks 网站上的文章，Perf -- Linux下的系统性能调优工具，<http://www.ibm.com/developerworks/cn/linux/l-cn-perf1/>

<https://www.ibm.com/developerworks/cn/linux/l-cn-perf2/>

淘宝内核组上的文档，perf在linux性能评估中的应用，<http://kernel.taobao.org/index.php?title=Documents/Kernel_Perf>