



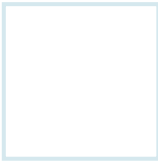
Introduction to Linux Systems

Basics of Performance Analysis Part II

Chia-Heng Tu

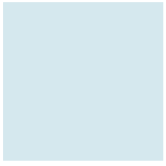
Dept. of Computer Science and Information
Engineering

National Cheng Kung University
Fall 2022



Lab

- Install Perf
- Perf Configuration
- Userspace Perf Commands
- Apply Perf



Install Perf

- `sudo apt-get install linux-tools-common`
- `sudo apt-get install linux-tools-5.15.0-48-generic`
`linux-cloud-tools-5.15.0-48-generic`



Perf Configuration

- `sudo sh -c " echo 0 > /proc/sys/kernel/perf_event_paranoid"`

`/proc/sys/kernel/perf_event_paranoid`

The `perf_event_paranoid` file can be set to restrict access to the performance counters.

- 2 allow only user-space measurements (default since Linux 4.6).
- 1 allow both kernel and user measurements (default before Linux 4.6).
- 0 allow access to CPU-specific data but not raw tracepoint samples.
- 1 no restrictions.

The existence of the `perf_event_paranoid` file is the official method for determining if a kernel supports `perf_event_open()`.



Userspace Perf Commands

- perf list: display the symbolic event types
- perf stat: obtain event counts
- perf record: record events for later reporting
- perf report: break down events by process, function, etc.
- perf annotate: annotate assembly or source code with event counts
- perf top: see live event count
- perf bench: run different kernel microbenchmarks



Apply Perf

lab11.c Code

lab10.c
ZONG5 DEC 6TH, 2021 (EDITED) 1 NEVER

```
C++ 0.16 KB
1. static char array[10000][10000];
2. int main (void){
3.     int i, j;
4.     for (i = 0; i < 10000; i++)
5.         for (j = 0; j < 10000; j++)
6.             array[j][i]++;
7.     return 0;
8. }
```

- gcc -o lab11.out lab11.c
- perf stat ./lab11.out

Virtual Machine may not support some Hardware events.

```
zong@zong-VirtualBox:~$ perf stat ./lab11.out
```

```
Performance counter stats for './lab11.out':
```

673.04 msec	task-clock	#	0.999 CPUs utilized
19	context-switches	#	28.230 /sec
0	cpu-migrations	#	0.000 /sec
4,8872	page-faults	#	72.614 K/sec



Apply Perf (Cont'd)



lab10.c

 ZONG5  DEC 6TH, 2021 (EDITED)  1  NEVER

C++ 0.16 KB

```
1. static char array[10000][10000];
2. int main (void){
3.     int i, j;
4.     for (i = 0; i < 10000; i++)
5.         for (j = 0; j < 10000; j++)
6.             array[j][i]++;
7.     return 0;
8. }
```

⇒ array[i][j]++;

Please modify the 6th line in lab11.c.

Before : array[j][i]++;

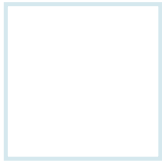
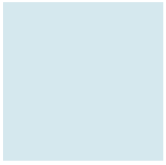
After : array[i][j]++;

Then apply gcc and perf again.

```
zong@zong-VirtualBox:~$ perf stat ./lab11.out
```

Performance counter stats for './lab11.out':

301.70 msec	task-clock	#	0.999 CPUs utilized
7	context-switches	#	23.202 /sec
0	cpu-migrations	#	0.000 /sec
4,8872	page-faults	#	161.989 K/sec



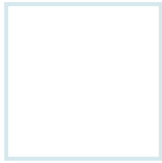
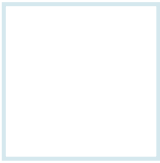
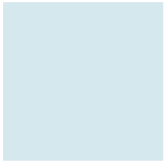
Demo

- Please upload 1 screenshot to moodle.
- Modify the original lab11.c.
- Apply perf on its execution.

```
zong@zong-VirtualBox:~$ perf stat ./lab11.out
```

```
Performance counter stats for './lab11.out':
```

301.70 msec	task-clock	#	0.999 CPUs utilized
7	context-switches	#	23.202 /sec
0	cpu-migrations	#	0.000 /sec
4,8872	page-faults	#	161.989 K/sec



QUESTIONS