Performance Counters, Affinity, File Formats, System Tools

MO801

Profile

- gprof
- Oprofile
- perf
- papi

gprof – static profiler

- The compiled program creates a file called gmon.out
- gprof reads this file
- You are not supposed to execute

  `gprof myprogram`

  – To get the profile, this command is the third step

gprof example

- Consider the telecom/gsm benchmark from MiBench
- To compile, edit Makefile and add –pg to CFLAGS and LDFLAGS
- Run accordingly (runme_large.sh)
- Get the profile information

  gprof bin/toast

Profile information

```
Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  ms/call  ms/call  name
 46.15      0.06     0.06    28976     0.00     0.00  Gsm_Long_Term_Predictor
 15.38      0.08     0.02    28976     0.00     0.00  Gsm_RPE_Encoding
 15.38      0.10     0.02     7244     0.00     0.00  Gsm_LPC_Analysis
 15.38      0.12     0.02     7244     0.00     0.00  Gsm_Short_Term_Analysis_Filter
  7.69      0.13     0.01     7244     0.00     0.00  Gsm_Preprocess
  0.00      0.13     0.00   376688     0.00     0.00  gsm_asr
  0.00      0.13     0.00    72059     0.00     0.00  gsm_norm
```

Caption

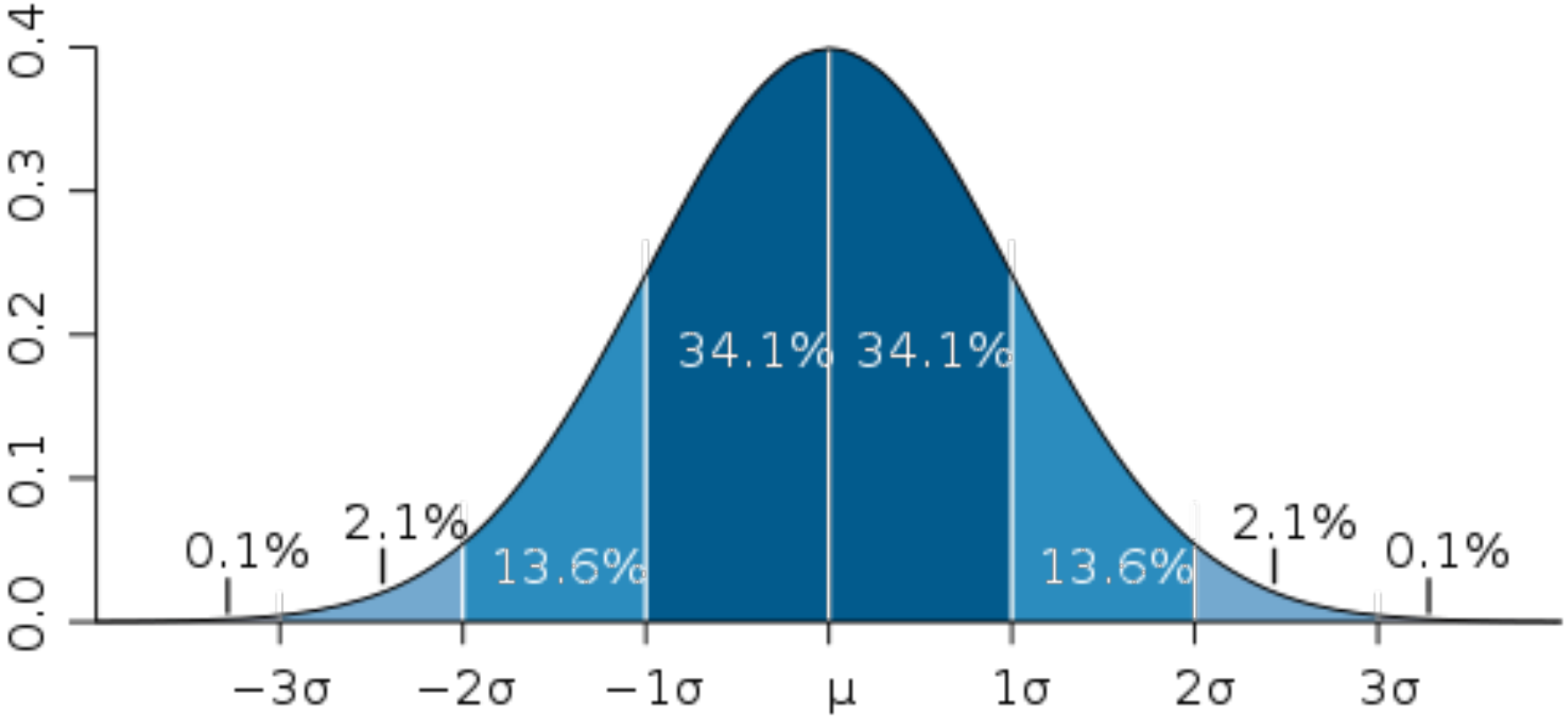
Column	Description
% time	The percentage of the total running time of the program used by this function.
Cumulative seconds	A running sum of the number of seconds accounted for by this function and those listed above it.
Self seconds	The number of seconds accounted for by this function alone. This is the major sort for this listing.
Calls	The number of times this function was invoked, if this function is profiled, else blank.
Self ms/call	The average number of milliseconds spent in this function per call, if this function is profiled, else blank.
Total ms/call	The average number of milliseconds spent in this function and its descendents per call, if this function is profiled, else blank.
Name	The name of the function.

Per Function

```
index % time    self  children    called     name
                0.00    0.13    7244/7244        gsm_encode [2]
[1]    100.0    0.00    0.13    7244         Gsm_Coder [1]
                0.06    0.00   28976/28976       Gsm_Long_Term_Predictor [5]
                0.02    0.00   28976/28976       Gsm_RPE_Encoding [6]
                0.02    0.00    7244/7244        Gsm_LPC_Analysis [7]
                0.02    0.00    7244/7244        Gsm_Short_Term_Analysis_Filter [8]
                0.01    0.00    7244/7244        Gsm_Preprocess [9]
```

oprofile – dynamic profiler

- Statistical sampling



Output

- operf bin/toast …
- opreport -l → show the profile
- Annotate mixed source/assembly

```
opannotate –source –assembly bin/toast
```

- Image summary for a single application

```
opreport bin/toast
```

Performance Counters

- Counters embedded inside the processor
- Counts events in several different granularity
- Events vs Counters
  - Not always the same
  - Processors can generate a lot of different events
  - Processors have a few counters that can be assigned to selected events

perf

- Linux tool for performance counter
- Not all events available for all processor and OS
- List of all events supported

```
perf list
```

- Simplest execution line

```
perf stat /bin/ls
```

- Record and view

```
perf record …
```

Perf events (-e option)

cpu-cycles OR cycles	cpu-clock	L1-dcache-load-misses	branch-instructions OR cpu/branch-instructions/
instructions	task-clock	L1-dcache-store-misses	branch-misses OR cpu/branch-misses/
cache-references	page-faults OR faults	L1-dcache-prefetch-misses	bus-cycles OR cpu/bus-cycles/
cache-misses	context-switches OR cs	L1-icache-load-misses	cache-misses OR cpu/cache-misses/
branch-instructions OR branches	cpu-migrations OR migrations	LLC-loads	cache-references OR cpu/cache-references/
branch-misses	minor-faults	LLC-stores	cpu-cycles OR cpu/cpu-cycles/
bus-cycles	major-faults	LLC-prefetches	instructions OR cpu/instructions/
stalled-cycles-frontend OR idle-cycles-frontend	alignment-faults	dTLB-load-misses	mem-loads OR cpu/mem-loads/
ref-cycles	emulation-faults	dTLB-store-misses	mem-stores OR cpu/mem-stores/
	dummy	iTLB-loads	stalled-cycles-frontend OR cpu/stalled-cycles-frontend/
		iTLB-load-misses	uncore_cbox_0/clockticks/
		branch-loads	uncore_cbox_1/clockticks/
		branch-load-misses	uncore_cbox_2/clockticks/
			uncore_cbox_3/clockticks/

Example

- Run PARSEC blackscholes collecting statistics

```
parsecmgmt -a run -p blackscholes -i
native -s "perf stat”
```

- Run PARSEC blackscholes collecting cache misses

```
parsecmgmt -a run -p blackscholes -i
native -s "perf stat -e cache-misses”
```

PAPI

- Performance Application Programming Interface
- Allow standardized access to Performance Counters
- papi_avail → List of available events
- papi_mem_info → Memory hierarchy information

CPU Affinity

- You can control the cores used by your programs
- Easy way to place each program thread/ process in each core
- Taskset command in Linux

```
taskset 03 /bin/ls
taskset –c 0,1 /bin/ls
taskset –c 0 –p 1000
```

File Formats

- To store data tables
  - csv
- To store configurations or structured data
  - yaml or yml
- Do not forget about naming conventions

System Tools

- Process files
  - awk
  - sed
- Control execution
  - shell script
- Create graphics
  - gnuplot