Pontificia Universidad
JAVERIANA
Cali

Facultad de Ingeniería y Ciencias
Departamento de Electrónica y Ciencias de la Computación

Carrera de Ingeniería de Sistemas y Computación

Programación Paralela

Profiling C programs

Mayo 2022

Tek

Profiling C programs – `gprof`

1. Add the –`pg` flag to the compiler flags in the corresponding `Makefile`
2. Rebuild the application by typing `make`
3. Run the application. You should get an output file called `gmon.out`
4. Run the post-processing step by typing

   `gprof MyApplication` (full output)

   `gprof -p MyApplication` (flat profile)

   `gprof -q MyApplication` (call graph)

   `gprof MyApp | gprof2dot | dot -Tpng -o MyApp_gprof.png` (graphical hotspots call graph)

Profiling C programs line-by-line – `gprof`

1. Add the `–pg  –g` flags to the compiler flags in the corresponding `Makefile`
2. Rebuild the application by typing `make`
3. Run the application. You should get an output file called `gmon.out`
4. Run the post-processing step by typing

   `gprof –l MyApplication` (full output)

   `gprof -pl MyApplication` (flat profile)

Profiling C programs – `valgrind`

1. Add the `–pg` flag to the compiler flags in the corresponding `Makefile`
2. Rebuild the application by typing `make`
3. Run the application

   ```
   valgrind --tool=callgrind ./MyApp
   ```

   You should get an output file called `callgrind.out.<pid>`
4. Run the post-processing step by typing

   ```
   kcachegrind &
   ```

Profiling C programs – GPerfTools

1. Run the application

```
LD_PRELOAD=/usr/lib/x86_64-linux-
gnu/libprofiler.so
CPUPROFILE=sort_gperf.prof ./sort_gperf
```

1. Analyse profiling output

```
google-pprof --web ./sort_gperf
./sort_gperf.prof &
```

Pre-requisites

- `pip install graphviz`
- `pip install gprof2dot`
- `sudo apt-get install graphviz`
- `sudo apt-get install valgrind kcachegrind`
- `sudo apt-get install google-perftools`
- `sudo apt-get install libgoogle-perftools-dev`

Not required, but nice to have it installed:

- `sudo apt-get install vorbis-tools`