Getting your performance up

Where did my cycles go?

Marcus Müller

Ettus Research

September 16th 2016

Vaguely based on experience

Preface

- The debugging: I'm stupid, but it's been a fun experience debugging this, kudos to noc0lour
- The Shell: This is `zsh`, enhanced by `oh-my-zsh`, with the Hokietux-maintained `Powerlevel9k` theme, running in `gnome-terminal`, with a `solarized-dark` color theme (which is the same used for this presentation)
- Shout if you spot mistakes. I'll repeat your heckle, so it's on record.
- funky animation: `https://github.com/marcusmueller/curses_animation`

Outline

https://github.com/marcusmueller/perfpres

Outline

1. adhoc stuff

https://github.com/marcusmueller/perfpres

Outline

1. adhoc stuff
2. barely prepared stuff

https://github.com/marcusmueller/perfpres

Stuff is too slow

Typical reflex: blame the software / SDR Vendor / CPU
Yeah. This is FOSS. Go and fix things yourself, right?

In GNU Radio, things are especially hard

- GNU Radio is inherently multithreaded
- there's the whole Python/C++/SWIG fun thing
- oh, and by the way, we do things at a couple $\frac{\text{MS}}{\text{s}}$

perf is your friend

- Install deps first

  Fedora, CentOS `sudo dnf install perf`

  Debian `sudo apt-get install linux-tools`

- `sudo sysctl -w kernel/perf_event_paranoid=-1`

perf commands

record	samples performance and writes a perf.data file
report	reads perf.data and displays stats
top	super handy instant inspection

perf typical usage

```
perf record -ag executable --program_arguments
```

- `record` write data
- `-a` all CPUs (super important if you're running stuff with multiple threads)
- `-g` record call graphs – who called whom?

Demo