Tunables

The Future of Platform Optimization Selection in glibc
Outline

- What problem are we trying to solve?
- Goals
  - Tunables for users
  - Tunables for distribution maintainers
- Roadmap (a.k.a The future of the future)
The Problem

- **Constants in glibc**
  - Decided years ago based on the state of the art
  - May not be relevant anymore
  - May be heuristic (64k ought to be enough for anybody!)

- **Naming consistency**
  - MALLOC_MMAP_THRESHOLD_: what's with the trailing underscore?!
  - Distribution-specific tunables
Goals

● For Users
  - Per-process tunable values
  - Per-user tunable values
  - System-wide tunable values
  - Tweak tunables at runtime

● For Distribution maintainers
  - Add distribution-specific tunables
  - Distribution defaults for tunables
Tunables for users

- One environment variable to rule them all!
  - GLIBC_TUNABLES
  - Colon-separated name=value pairs
  - e.g. GLIBC_TUNABLES="glibc.malloc.check=3:glibc.malloc.arena_max=1"
Tunables for users – the other plan

- One environment variable per tunable
  - Standard environment variable names
  - e.g. GLIBC_MALLOC_CHECK=2
Tunables for users – the future

Coming up, stay tuned!
Tunables for distributions

- **Add distribution-specific tunables**
  - Tunables relevant to customer base
  - More importantly, NOT relevant upstream
  - Users know which tunables are supported across distributions
  - e.g. `redhat.malloc.allocator=tcmalloc`

- **Distribution defaults**
  - e.g. disable per-thread arenas for cloud images with less than 1GB RAM
Roadmap (cue hand-waving…)

- System-wide and per-user configuration files
  - ~/.sysctl.user.conf
  - /etc/sysctl.user.conf
  - Figure out performance cost
  - Used to set distribution defaults
- Tweaking tunables at runtime
  - echo foo > /proc/PID/top_namespace/tunable_namespace/tunable
Thank you!

siddhesh@linaro.org