# cgroup v2 Support for BenchExec

BSc Thesis

Robin Gloster

2022-03-09

# cgroup

- Linux Kernel feature
- Groups processes
- Monitors and limits resoures
- Interaction through files in virtual file system
- v2 added in Linux 4.5 due to inconsistencies, complexity and security issues

# BenchExec

- Precise benchmarking for arbitrary commands
- Resource limits
- Isolation
- `benchexec` & `runexec`

# cgroup Subsystems

- **blkio/io**
- **cpuset**
- **cpu & cpuacct**
- **memory**
- **pid**

# cgroup Features

- **freezer**
- **kill** (Linux 5.14, August 2021)
- Pressure Stall Information (PSI)

# cgroup v1

*old multi-tree hierarchy*

```
/sys/fs/cgroup/cpuacct/system.slice/postgresql.service
├── tasks
├── cpuacct.usage
├── ...
```

```
/sys/fs/cgroup/memory/system.slice/postgresql.service
├── tasks
├── memory.max_usage_in_bytes
├── ...
```

```
$ mkdir /sys/fs/cgroup/cpuacct/example-cgroup

$ echo $PID > /sys/fs/cgroup/cpuacct/example-cgroup/tasks

$ cat /sys/fs/cgroup/cpuacct/example-cgroup/cpuacct.usage
324099881755
```

# cgroup v2

*unified hierarchy*

```
/sys/fs/cgroup/system.slice/postgresql.service
├── cgroup.controllers
├── cgroup.freeze
├── cgroup.procs
├── cgroup.subtree_control
├── cpu.max
├── cpu.stat
├── cpuset.cpus
├── io.max
├── io.stat
├── memory.max
├── memory.stat
├── ...
```

# No Processes in Inner Nodes

# Permission Boundaries

- Not possible to move process through subtree that is non-writable

# Delegation

- Create subfolder in cgroup
- Move with `cgroup.procs`
- Allow controllers in parent `cgroup.subtree_control`
- Change write permissions to lesser-privileged user

# cgroup and systemd

- **slice**
- **scope**
- **service**
- `Delegate=true`

# cgroup v2 Adoption

- systemd support in May 2016, default changed in September 2019
- Fedora 31 October 2019
- Debian stable August 2021
- Ubuntu 21.10 October 2021
- Docker support in December 2020

# Requirements

- Parallel support for v1 and v2
- Respect new restrictions
- systemd delegation recommendations
- Possibility for root-less usage of BenchExec

# cgroup v1 Usage in BenchExec

- CPU pinning (**cpuset**)
- Memory limit and OOM handling (**memory**)
- Memory peak measurement (**memory**)
- CPU time measurement (**cpuacct**)
- I/O measurement (**blkio**)
- reliably kill processes (**freezer**)

# cgroup v2 Usage in BenchExec

- CPU pinning (**cpuset**)
- Memory limit and OOM handling (**memory**)
- CPU time measurement (**cpu**)
- I/O measurement (**io**)

# cgroup v2 Usage in BenchExec

- Memory peak measurement
  (not available in cgroup v2)
- Reliably kill processes (**freezer** or **kill**)

# cgroup v1 Setup in BenchExec

- If cgroups writable in which `runexec` is started, these are used
- Otherwise use fallback cgroups
  - Service created by .deb package
  - Creates `system.slice/benchexec-cgroup.service`
  - Writable by benchexec group

# cgroup v2 Setup in BenchExec

- `runexec` has to be run in correct cgroup
- `systemd-run --user --scope -p Delegate=yes benchexec ...`
- Main `benchexec` process in its own cgroup
- All `benchexec` and `runexec` processes in one cgroup subtree
- No root permissions needed

# pystemd

- Python library to communicate with systemd via D-Bus
- Chosen after evaluation of active maintenance, availability, ease of use
- `benchexec` moves itself to systemd scope if available

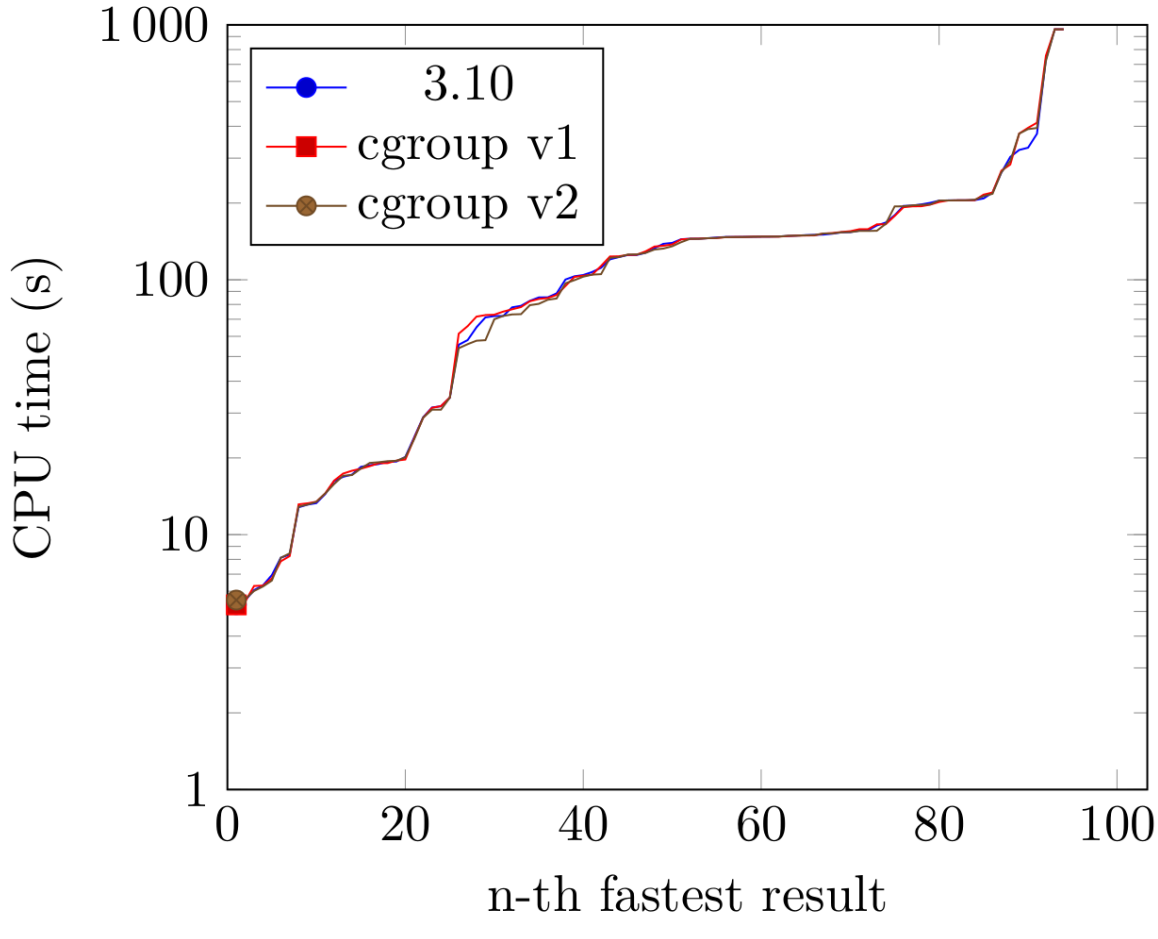
# Evaluation

- Ubuntu 21.10 with Linux 5.13
- `Intel(R) Xeon(R) CPU E3-1246 v3 @ 3.60GHz` and 32 GiB RAM
- BenchExec 3.10 and `cgroupsv2` branch.

# Regular Benchmark

- SV-COMP 2022 taskset
- CPAChecker 2.1
- ReachSafety-ControlFlow.set

# Regular Benchmark



# Feature Verification

- `doc/benchmark-example-calculatepi.xml`
- Memory limit
- CPU time limit

# Conclusion

- Feature parity
  - to previous implementation
  - between cgroup v1 and v2 (except peak memory and per CPU metrics)
- Possibility to run on newer Linux distributions
- No administrative setup required with cgroup v2

# Future

- Add peak memory usage support in Linux kernel
- cgroup namespaces -> usage of cgroup in benchmarked process
- cgroup v2 provides better soft limit interface
  - e.g. react to soft memory limit
- Evaluate usefulness of PSI metrics for real-world benchmarks

# References

- https://github.com/sosy-lab/benchexec
- Reliable benchmarking: requirements and solutions
- https://www.kernel.org/doc/Documentation/cgroup-v1/cgroups.txt
- https://www.kernel.org/doc/Documentation/cgroup-v2.txt
- https://systemd.io/CGROUP_DELEGATION/
- https://man7.org/linux/man-pages/man7/cgroups.7.html