Sandmark 2.0

A benchmark suite for OCaml

Shakthi Kannan

Sandmark Benchmarking Suite

README.md

Sandmark

A benchmark suite for OCaml.

Quick Start

On Ubuntu 18.04.4 LTS you can try the following commands:

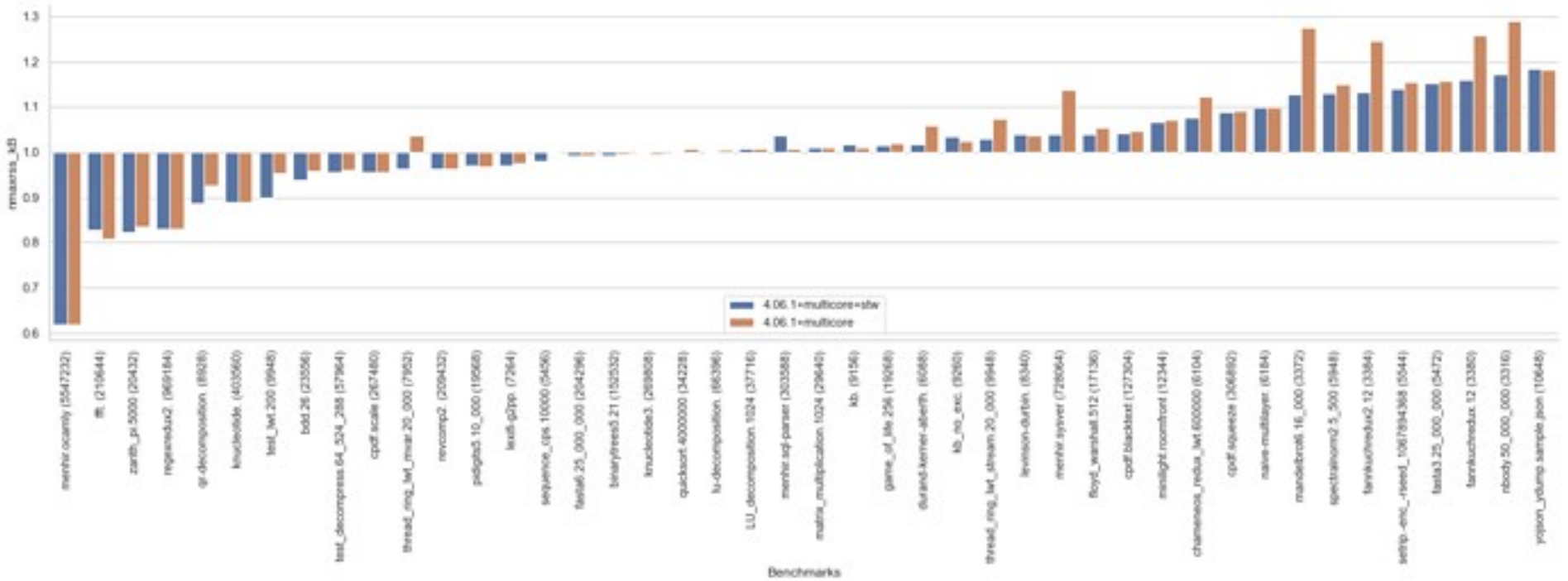
```
sandmark $ (master) ls
benchmarks      dependencies          dune                 dune-project   LICENSE
Makefile        notebooks             ocaml-versions       README.md      orun
rungen          run_all_serial.sh     pausetimes           run_config.json
run_all_effects.sh  run_all_parallel.sh      micro_multicore.json
multicore_effects_run_config.json  multicore_parallel_run_config.json
```

Source: https://github.com/ocaml-bench/sandmark

Sandmark Benchmarking Suite (...)

```
$ cat _results/4.10.0+multicore_1.orun.summary.bench

{"name":"test_decompress.64_524_288", "command":"taskset --cpu-list 5 ./test_decompress.exe 64 524_288",
"time_secs":3.564422845840454, "user_time_secs":3.538128, "sys_time_secs":0.023987,"maxrss_kB":13548,
"ocaml": "version":"4.10.0+multicore", "c_compiler":"gcc", "architecture":"amd64", "word_size":"64",
"system":"linux", "stats":"false", "function_sections":"true", "supports_shared_libraries":"true"}, "gc":
{"allocated_words":408512497, "minor_words":302819520, "promoted_words":79095, "major_words":105772072,
"minor_collections":1592, "major_collections":455, "heap_words":681083, "top_heap_words":943449,
"mean_space_overhead":83.177215}, "codesize":340742.0,"ocaml_url":"
https://github.com/ocaml-multicore/ocaml-multicore/archive/parallel_minor_gc.tar.gz"}

{"name":"imrin_mem_rw.10_000_50_000_80_100_000_000","command":"taskset --cpu-list 5 ./irmin_mem_rw.exe 10_000
50_000 80 100_000_000", "time_secs":5.536180019378662, "user_time_secs":5.534524, "sys_time_secs":0.0,
"maxrss_kB":14444,"ocaml":{"version":"4.10.0+multicore", "c_compiler":"gcc", "architecture":"amd64",
"word_size":"64", "system":"linux", "stats":"false", "function_sections":"true",
"supports_shared_libraries":"true"}, "gc":{"allocated_words":6021851723, "minor_words":6021850632,
"promoted_words":799549, "major_words":800640, "minor_collections":23003, "major_collections":35,
"heap_words":143140, "top_heap_words":638756, "mean_space_overhead":274.472053}, "codesize":1401138.0,
"ocaml_url":"https://github.com/ocaml-multicore/ocaml-multicore/archive/parallel_minor_gc.tar.gz"}

...
```

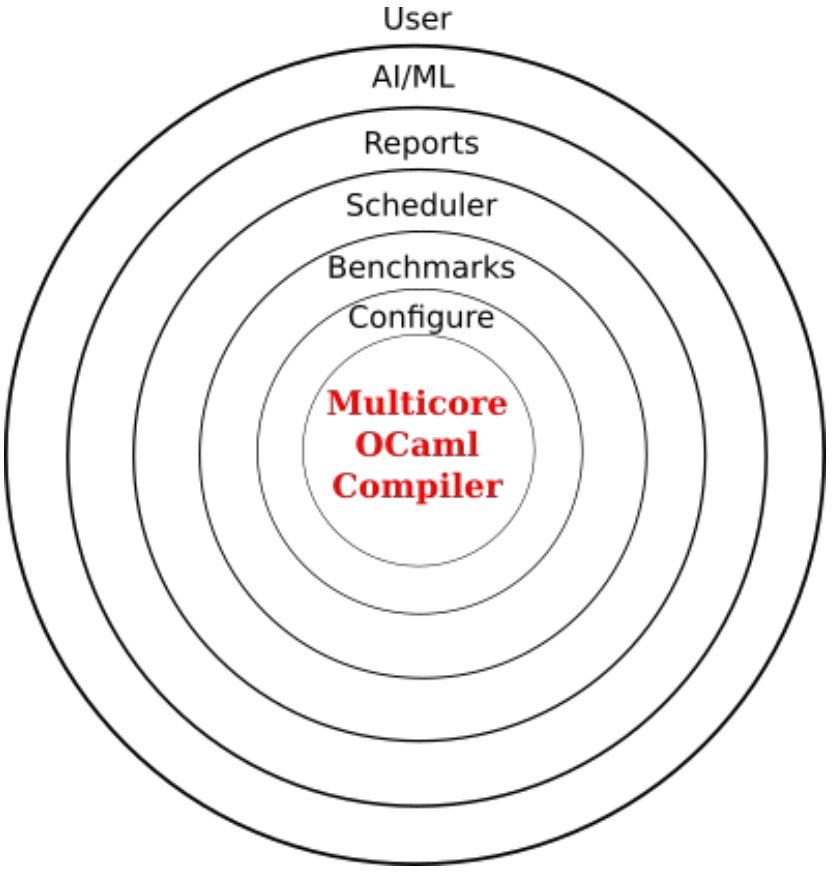
Sandmark Benchmarking Suite (...)



Motivation

Sandmark 1.0

* Dune sys hack
* Dependencies
* ITER
* Codespeed



Sandmark 2.0

* Dune native
* Minimal dependencies
* ITER
* Benchmark classification
* User configuration
* ocaml-ci
* Deployment
* UI Dashboard
* Analytics

Dune (native)

Makefile

```
# HACK: we are using the system installed dune to avoid breakages with
# multicore and 4.09/trunk
# This is a workaround for r14/4.09/trunk until better solutions arrive
SYS_DUNE_BASE_DIR ?= $(subst /bin/dune,,$(shell which dune))

setup_sys_dune:
ifeq (,$(SYS_DUNE_BASE_DIR))
        $(error Could not find a system installation of dune (try `opam install dune`?))
else
        @echo "Linking to system dune files found at: "$(SYS_DUNE_BASE_DIR)
        @echo $(SYS_DUNE_BASE_DIR)"/bin/dune --version = "$(shell $(SYS_DUNE_BASE_DIR)/bin/dune –
                version)
        @rm -rf $(CURDIR)/_opam/sys_dune
        @mkdir -p $(CURDIR)/_opam/sys_dune/bin
        @mkdir -p $(CURDIR)/_opam/sys_dune/lib
        ln -s $(SYS_DUNE_BASE_DIR)/bin/dune $(CURDIR)/_opam/sys_dune/bin/dune
        ln -s $(SYS_DUNE_BASE_DIR)/bin/jbuilder $(CURDIR)/_opam/sys_dune/bin/jbuilder
        ln -s $(SYS_DUNE_BASE_DIR)/lib/dune $(CURDIR)/_opam/sys_dune/lib/dune
endif
```

PR#407: Systhreads on Multicore
https://github.com/ocaml-multicore/ocaml-multicore/pull/407

Makefile

```
opam install –switch=$(CONFIG_SWITCH_NAME) –yes “dune.2.6.0”
```

orun and rungen

rungen

Generates dune files to run benchmarks from a centralized configuration

OCaml MIT 1 0 0 0 Updated yesterday

sandmark

A benchmark suite for the OCaml compiler

Jupyter Notebook CC0-1.0 15 62 16 7 Updated 3 days ago

orun

orun is a benchmarking tool that gives you runtime and OCaml garbage collector statistics.

OCaml MIT 1 1 0 0 Updated 5 days ago

Source: https://github.com/ocaml-bench

Dependencies

```
$ ls dependencies/packages/
```

```
alt-ergo          base-unix         conf-gmp          decompress        fraplib           lwt
ocamlbuild        react             uri               alt-ergo-lib      bigarray-compat   conf-m4
digestif          integers          menhir            ocaml-config      result            uuidm
alt-ergo-parsers  bigstringaf       conf-perl         domainslib        irmin             menhirLib
ocamlfind         rresult           uutf              astring           biniou            conf-pkg-config
dune              irmin-mem         menhirSdk         ocamlgraph        seq               yojson
base              camlimages        conf-zlib         dune-configurator jbuilder          minilight
ocplib-endian     sexplib0          zarith            base64            camlpdf           coq
dune-private-libs js_of_ocaml-compiler                mmap              ocplib-simplex    stdio
base-bigarray     camlzip           cpdf              easy-format       jsonm             nbcodec
optint            stdlib-shims      base-bytes        checkseum         cppo              eqaf
kcas              num               ppx_derivers      stringext         base-num          cmdliner
ctypes            fmt               lockfree          ocaml             psmt2-frontend    topkg
base-threads      conf-findutils    cubicle           frama-c           logs              ocaml-base-compiler
re                uchar
```

Makefile

```
opam repo add upstream "https://opam.ocaml.org" --on-switch=$(CONFIG_SWITCH_NAME) --rank 2
```

```
$ ls dependencies/packages/
```

```
base-bigarray         base-threads      base-unix         coq         fraplib         ocaml
ocaml-base-compiler   ocaml-config      frama-c
```

Package Overrides

```
$ cat ocaml-versions/4.10.0+stock.json

{
      "url" : "https://github.com/ocaml/ocaml/archive/4.10.0.tar.gz",
      "package_overrides" : ["dune.2.8.0"],
      "configure" : "-q",
      "runparams" : "v=0x400"
}


$ RUN_CONFIG_JSON=run_config_filtered.json make ocaml-versions/4.10.0+stock.bench

...
The following actions will be performed:
  * install seq                 base
  * install conf-gmp            3
  * install conf-m4             1
  * install conf-perl           1
  * install conf-pkg-config     1.3
  * install ocamlbuild          0.14.0
  ↗ upgrade dune                2.6.0 to 2.8.0
...
```

Meta Header

```
$ cat _results/4.10.0+multicore_1.orun.summary.bench

{"version":"5.4.0-48-generic","hostname":"navajo","kernel":"Linux","arch":"x86_64"}

{"name":"test_decompress.64_524_288", "command":"taskset --cpu-list 5 ./test_decompress.exe 64 524_288",
"time_secs":3.564422845840454, "user_time_secs":3.538128, "sys_time_secs":0.023987,"maxrss_kB":13548,
"ocaml": "version":"4.10.0+multicore", "c_compiler":"gcc", "architecture":"amd64", "word_size":"64",
"system":"linux", "stats":"false", "function_sections":"true", "supports_shared_libraries":"true"}, "gc":
{"allocated_words":408512497, "minor_words":302819520, "promoted_words":79095, "major_words":105772072,
"minor_collections":1592, "major_collections":455, "heap_words":681083, "top_heap_words":943449,
"mean_space_overhead":83.177215}, "codesize":340742.0,"ocaml_url":"
https://github.com/ocaml-multicore/ocaml-multicore/archive/parallel_minor_gc.tar.gz"}

{"name":"imrin_mem_rw.10_000_50_000_80_100_000_000","command":"taskset --cpu-list 5 ./irmin_mem_rw.exe 10_000
50_000 80 100_000_000", "time_secs":5.536180019378662, "user_time_secs":5.534524, "sys_time_secs":0.0,
"maxrss_kB":14444,"ocaml":{"version":"4.10.0+multicore", "c_compiler":"gcc", "architecture":"amd64",
"word_size":"64", "system":"linux", "stats":"false", "function_sections":"true",
"supports_shared_libraries":"true"}, "gc":{"allocated_words":6021851723, "minor_words":6021850632,
"promoted_words":799549, "major_words":800640, "minor_collections":23003, "major_collections":35,
"heap_words":143140, "top_heap_words":638756, "mean_space_overhead":274.472053}, "codesize":1401138.0,
"ocaml_url":"https://github.com/ocaml-multicore/ocaml-multicore/archive/parallel_minor_gc.tar.gz"}
```

RFC: Header entry attributes for the summary benchmark result
https://github.com/ocaml-bench/sandmark/issues/193

List Tags

```
$ ls *.json

micro_multicore.json                  multicore_effects_run_config.json
multicore_parallel_run_config.json    run_config.json
```

Makefile

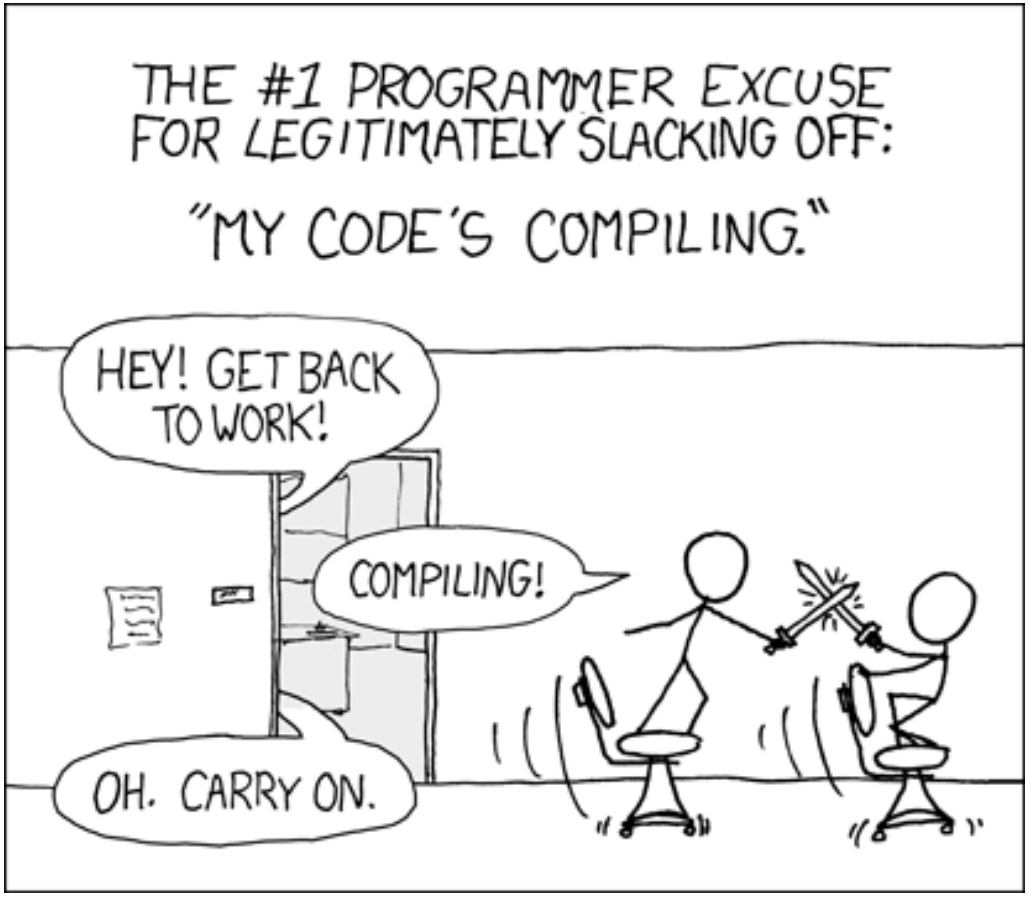
```
list_tags:
        @echo "List of Tags"
        @jq '[.benchmarks[].tags] | add | flatten | .[]' *.json | sort -u
```

```
$ make list_tags

List of Tags
"10s_100s"
"1s_10s"
"gt_100s"
"lt_1s"
"macro_bench"
"run_in_ci"
```

ITER

```
$ TAG='"run_in_ci"' make run_config_filtered.json

$ ITER=2 RUN_CONFIG_JSON=run_config_filtered.json make ocaml-versions/4.10.0+stock.bench

...
Done.
opam exec --switch 4.10.0+stock -- rungen _build/4.10.0+stock_1 run_config_filtered.json > runs_dune.inc
opam exec --switch 4.10.0+stock -- dune build --profile=release \
      --workspace=ocaml-versions/.workspace.4.10.0+stock @buildbench;
Executing benchmarks with:
  RUN_CONFIG_JSON=run_config_filtered.json
  RUN_BENCH_TARGET=run_orun  (WRAPPER=orun)
  PRE_BENCH_EXEC=


$ ls _build/
4.10.0+stock_1  4.10.0+stock_2  log


$ ls _results/
4.10.0+stock_1.orun.summary.bench  4.10.0+stock_2.orun.summary.bench
```

Demo



Source: https://xkcd.com/371/

Classify Benchmarks

```
$ ls benchmarks/
```

```
almabench              chameneos  decompress      kb          multicore-effects     multicore-minilight
numerical-analysis     stdlib     zarith          alt-ergo    coq                   frama-c
lexifi-g2pp            multicore-gcroots          multicore-numerical               sauvola
thread-lwt             bdd        cpdf            graph500seq                       menhir
multicore-grammatrix   multicore-structures       sequence    valet                 benchmarksgame
cubicle                irmin      minilight       multicore-lockfree                nbcodec
simple-tests           yojson
```

RFC: Categorize and group by benchmarks
https://github.com/ocaml-bench/sandmark/issues/205

```
$ ls benchmarks/
```

```
library/   formal/    numerical/      graph /     multicore/
```

User Configuration

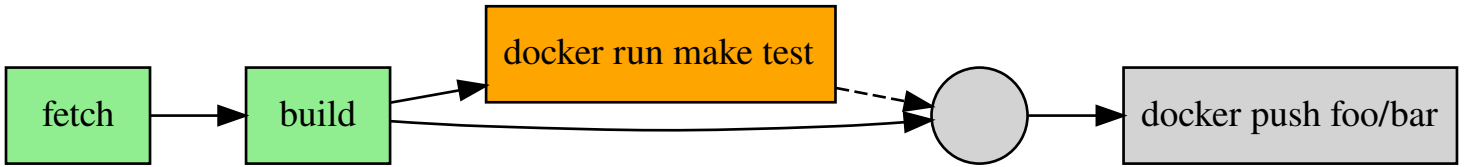
```
$ cat ocaml-versions/4.10.0+stock.json

{
     "url" : "https://github.com/ocaml/ocaml/archive/4.10.0.tar.gz",
     "package_overrides" : ["dune.2.8.0"],
     "configure" : "-q",
     "runparams" : "v=0x400"
}
```

RFC: How should a user configure a Sandmark run?
https://github.com/ocaml-bench/sandmark/issues/142

* Choice of benchmarks (graph, numerical, formal)
* List of configure switches
* Set of binary programs for statistics (perf, orun, size)
* Environment configuration (taskset, ASLR, OCAMLRUNPARAM)
* Profile definition to describe experiments
* OS specific settings for evaluation

...

OCaml CI




```
let pipeline ~repo () =
  let src = Git.Local.head_commit repo in
  let base = Docker.pull ~schedule:weekly "ocaml/opam2" in
  let build ocaml_version =
    let dockerfile =
      let+ base = base in
      dockerfile ~base ~ocaml_version
    in
    Docker.build ~label:ocaml_version ~pull:false ~dockerfile (`Git src) |>
    Docker.tag ~tag:(Fmt.strf "example-%s" ocaml_version)
  in
  Current.all [
    build "4.07";
    build "4.08"
  ]
```

Source: https://ocurrent.org/

UI Dashboard

Source: https://play.grafana.org

AI/ML

* Optimal configuration for high performance and low memory footprint
* Auto-parallelization for use with Multicore OCaml
* Impact of code change on different CPU architectures
* Health score for OCaml packages and feedback to developers
* Efficient scheduling for the run-time based on benchmark metrics
* Code refactoring recommendations to developers
* Real-time compiler warnings and error indicators in the IDE
* Automated patch fixes and suggestions for code reviews
* OCaml Scientific Computing algorithms

References

- Sandmark
  https://github.com/ocaml-bench/sandmark

  Sandmark Issues
  https://github.com/ocaml-bench/sandmark/issues

- Sandmark 2.0.0-alpha branch
  ```
  $ git clone https://github.com/shakthimaan/sandmark.git
  $ git checkout 2.0.0-alpha
  ```

- orun
  https://github.com/ocaml-bench/orun

- rungen
  https://github.com/ocaml-bench/rungen

- OCaml discuss
  https://discuss.ocaml.org/

Thank You