

logging
benchmarking
monitoring

2019-04-09

overview

- package `contra-tracer`
- Denis: contravariant *Tracer*
- logging
 - architecture
 - message routing
- micro-benchmarking
 - aggregation
- monitoring
- outlook

`contra-tracer` package



logging 1-pager

- minimal implementation of contravariant *Tracer*
 - captures observables
 - *Tracer* transformers like filters
- simple dependency to start with “tracing”
add `contra-tracer` package to cabal file
and `iohk-monitoring-framework` repo to cabal.project
- there is a “bridge” to more elaborate logging, benchmarking, and monitoring

[`toLogObject`](#) :: *Tracer* m (LogObject a) -> *Tracer* m a

transformer which “consumes” a *Tracer* of *LogObject*

contravariant Tracer

=> Denis

logging

- setup procedure: configuration -> *Switchboard* -> backends

```
trace <- setupTrace (Left "config.yaml") "top"
```

- central *Switchboard* dispatches messages according to configuration
- configuration can be changed at runtime; re-exported to file
- structured logging: logged items require instance of ToObject (default: ToJSON)

```
class ToJSON a => ToObject a where
  toObject :: a -> Object
  default toObject :: a -> Object
  toObject v = case toJSON v of
    Object o -> o
    _         -> mempty
```

message routing

- logged messages in named context (*LoggerName*)
- configuration maps *LoggerName* to definition:
 - select backend
 - select scribe (katip backend)
 - local severity filter
 - subtrace behaviour
- backends:
 - LogBuffer (map *LoggerName* -> *LogObject*)
 - EKG view
 - Aggregation
 - Configuration editor
 - Log (katip)

Micro-benchmarking

- record operating system counters (before, after)
- relate to “bracketed” function (by `LoggerName`)
- observe monadic and STM actions; bracket returns action’s result

```
bracketObserveIO :: Configuration -> Trace IO a -> Severity -> LoggerName -> IO t -> IO t  
bracketObserveIO :: Configuration -> Trace IO a -> Severity -> LoggerName -> STM t -> IO t
```

- local name: lookup set of counters to observe in configuration
- traced counters can be routed; for example to aggregation
- no code change needed in function - just “bracket” it
- performance considerations: turn on/off the capturing and tracing of counters for this *LoggerName* in the configuration

Aggregation

- stateful backend
- aggregates multiple measurements by updating map of *LoggerName* -> statistics
 - compute simple statistics (count,min,max,mean,stdev) over
 - traced values
 - rate of change
 - time between messages
 - ewma (exponentially weighted moving average)
- aggregated values enter *Switchboard*

prefixed name: #aggregation

type: AggregatedMessage

```
{
  "loName": "#aggregation.complex.random",
  "loContent": {
    "tag": "AggregatedMessage",
    "contents": [
      [ "rr",
        {
          "tag": "AggregatedStats",
          "contents": {
            "fbasic": {
              "fcount": 1083,
              "fmax": { "tag": "PureD",
                "contents": 83.99436778214672
              },
              "fmin": { "tag": "PureD",
                "contents": 0.2654550099230195
              },
              "fsum_A": 42.48318120639908,
              "fsum_B": 637617.419056741
            },
            "ftimed": {
              "fcount": 1081,
              "fmax": { "tag": "Nanoseconds",
                "contents": 78603588000
              }
            }
          }
        }
      ]
    ]
  },
  "loMeta": {
    "tstamp": "2019-04-08T18:35:00.437498Z",
    "severity": "Debug",
    "privacy": "Public",
    "tid": "\\ThreadId 38\\\"
  }
}
```

Monitoring

- DSL for comparison to threshold
 - included in configuration
- Actions
 - changes to runtime configuration (e.g. severity filters)
 - changes of threshold, severity annotation, ..

```
'#aggregation.critproc.observable' :  
  monitor: (mean >= (42))  
  actions:  
    - 'CreateMessage "exceeded" "the observable has been too long too high!"'  
    - 'AlterGlobalMinSeverity Info'
```

Outlook

- integration into `cardano-shell`
- integration into `ouroboros-network`, starting micro-benchmarks
- integration into `cardano-ledger`, block validation tool
- docs docs docs:

<https://input-output-hk.github.io/iohk-monitoring-framework>

- repo is here:

<https://github.com/input-output-hk/iohk-monitoring-framework>

Thank you and

Duncan and Neil for their inputs and feedback

Alexander Vieth for his initial implementation of the contravariant *Trace*

Andreas Triantafyllos

Denis Shevchenko

Appendix Configuration Editor



<http://localhost:13789>

in the root of the repository launch: `cabal new-run example-complex`

and access EKG view on <http://localhost:12789>,
and the configuration editor on <http://localhost:13789>

Global severity filter

 INPUT | OUTPUT

Monitoring/Logging Configuration Editor

Set minimum severity to:

✓ Debug

Info

Notice

Warning

Error

Critical

Alert

Emergency

Backends

Scribes

Severities

SubTrace

Aggregation

Buffer

ExportConfiguration

:

New

Save

Cancel

LoggerName

Backends value

Output selection

 INPUT | OUTPUT

Monitoring/Logging Configuration Editor

Set minimum severity to: Debug ▼

Backends

Scribes

Severities

SubTrace

Aggregation

Buffer

ExportConfiguration

: New Save Cancel

LoggerName	Backends value		
complex.observeSTM.4	[AggregationBK]	Edit	Delete
#aggregation.complex.message	[EKGViewBK]	Edit	Delete
#aggregation.complex.observeSTM.6	[KatipBK]	Edit	Delete



Set minimum severity to: Debug



Backends

Scribes

Severities

SubTrace

Aggregation

Buffer

ExportConfiguration

:

New

Save

Cancel

Setting 'complex.random' to ['StdoutSK::stdout','FileTextSK::logs/out.txt'] in Scribes

LoggerName

Scribes value

#aggregation.complex.observeSTM.6

["FileJsonSK::logs/out.even.json"]

Edit

Delete

complex.random

["StdoutSK::stdout","FileTextSK::logs/out.txt"]

Edit

Delete

#aggregation.complex.observeSTM.1

["FileJsonSK::logs/out.odd.json"]

Edit

Delete



Set minimum severity to:

Debug



Backends

Scribes

Severities

SubTrace

Aggregation

Buffer

ExportConfiguration

complex.random

:

TeeTrace "ewma"

New

Save

Cancel

LoggerName

SubTrace value

complex.observeSTM.4

ObservableTrace [GhcRtsStats,MemoryStats]

Edit

Delete

complex.observeSTM.2

ObservableTrace [GhcRtsStats,MemoryStats]

Edit

Delete

complex.random

TeeTrace "ewma"

Edit

Delete

BackendsScribesSeveritiesSubTraceAggregationBufferExportConfiguration

:

NewSaveCancel

LoggerName	Buffer value
#buffered.#aggregation.complex.random	<pre>{ "loName": "#aggregation.complex.random", "loContent": { "tag": "AggregatedMessage", "contents": [{ "tag": "AggregatedStats", "contents": { "fbasic": { "fmax": { "tag": "PureD", "contents": 83.96842183763376, "fcount": 2397, "fsum_B": 1439381.8965625193, "fmin": { "tag": "PureD", "contents": 5.3387652944445474e-2, "fsum_A": 42.56526383431052, "ftimed": { "fmax": { "tag": "Nanoseconds", "contents": 507100000, "fcount": 2395, "fsum_B": 6044258073007926, "fmin": { "tag": "Nanoseconds", "contents": 500105000, "fsum_A": 5.03648680167015e8, "fold": { "tag": "Nanoseconds", "contents": 32472287271000, "flast": { "tag": "PureD", "contents": 63.68147738841184, "fdelta": { "fmax": { "tag": "PureD", "contents": 80.48583909357063, "fcount": 2396, "fsum_B": 2796239.795811094, "fmin": { "tag": "PureD", "contents": -82.92603697091808, "fsum_A": -7.5790756027934655e-3, "loMeta": { "tstamp": "2019-04-09T08:30:24.845319Z", "severity": "Debug", "privacy": "Public", "tid": "\Threadld 45\\"" } } } } } } } } } } } } } }] } }</pre>
#buffered.#aggregation.complex.random.ewma	<pre>{ "loName": "#aggregation.complex.random.ewma", "loContent": { "tag": "AggregatedMessage", "contents": [{ "rr": { "tag": "AggregatedEWMA", "contents": { "tag": "EWMA", "avg": { "tag": "PureD", "contents": 45.46896677038596, "alpha": 0.42, "loMeta": { "tstamp": "2019-04-09T08:30:24.845319Z", "severity": "Debug", "privacy": "Public", "tid": "\Threadld 45\\"" } } } } }] } }</pre>



INPUT | OUTPUT

Monitoring/Logging Configuration Editor

Set minimum severity to: Debug



Backends

Scribes

Severities

SubTrace

Aggregation

Buffer

ExportConfiguration

:

New

Save

Cancel

Configuration was exported to the file: /Volumes/Coding/src/hs/IOHK/iohk-monitoring-framework.git/iohk-monitoring/static/conf/

[Link to configuration file](#)