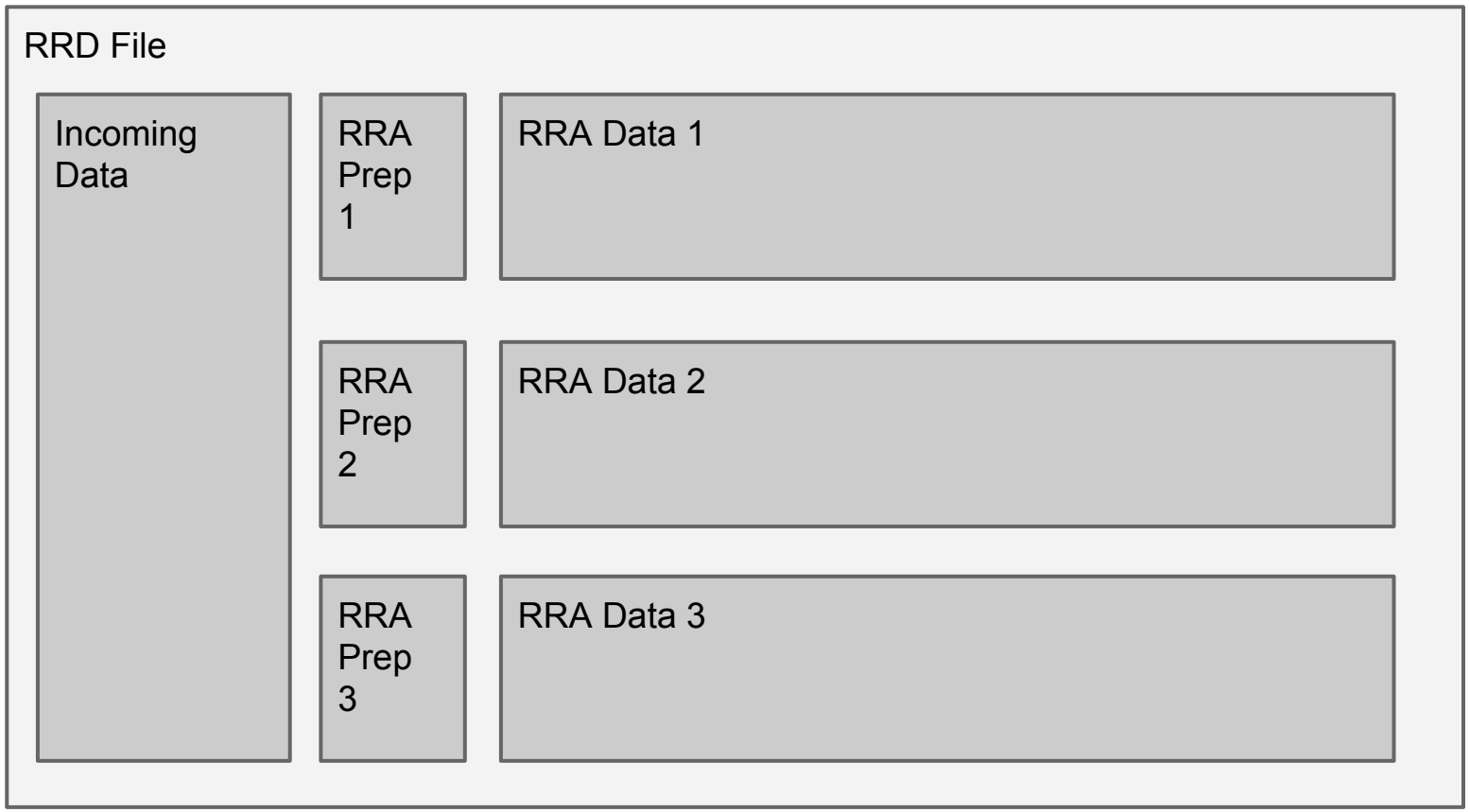
RRDtool für Fortgeschrittene

Tobias Oetiker
OETIKER+PARTNER AG
www.oetiker.ch

Wie geht RRDtool

Eine no-SQL Datenbank aus der Zeit als das Wort noch nicht erfunden war.
Wie funktioniert die RRDtool Datenbank?


RRD File
Incoming Data
RRA Prep 1
RRA Data 1
RRA Prep 2
RRA Data 2
RRA Prep 3
RRA Data 3

Wie RRDtool Daten schreibt und was wirklich passiert. Und warum mehr Speicher mehr Performance bringt.

RRDcached - was macht der ?

Was bringt RRDtool 1.5

gültigen JSON output von `rrdtool xport`

```
{ "hello": "yes this is json" }
```

simulierte "step 1 RRA"s
für `MIN`, `MAX` und `LAST`

Coole neue RPN Funktionen

ein `MEDIAN` operator für `CDEF` Ausdrücke

```
CDEF:x=a,b,c,d,4,MEDIAN
```

neue Stack Operatoren: DEPT, INDEX, COPY, ROLL

```
a,b,DEPTH          -> a,b,2
a,b,c,1,INDEX      -> a,b,c,c
a,b,c,d,2,COPY     -> a,b,c,b,c,d
a,b,c,d,3,1,ROLL   -> a,d,b,c
a,b,c,d,3,-1,ROLL -> a,c,d,b
```

`LINE` und `AREA` Befehle
können fürs Scaling ignoriert werden.

```
LINE:ds1#ff0000:Test:skipscale
```

alte Updates könne ignoriert werden

```
rrdtool --skip-past-updates x.rrd 1410794816:42
```

erkennt das
2038 32-bit time overflow problem

Leistungssteigerung für grosse RPN Ausdrücke mit 1000+ DEF Anweisungen durch Einsatz von HASH maps.

continuous integration dank
travis-ci auf GitHub

endlich hat RRDtool eine test suite

restore aus einer Pipe

```
rrdtool dump x.rrd | rrdtool restore - y.rrd
```

RRA Zeilenzahl und step können in absoluten Zeiträumen definiert werden

```
rrdtool create power.rrd \
   --start now-2h --step 1s \
   DS:watts:GAUGE:5m:0:24000 \
   RRA:AVERAGE:0.5:1s:10d \
   RRA:AVERAGE:0.5:1m:90d \
   RRA:AVERAGE:0.5:1h:18M
```

create on steroids: kann beim erzeugen neuer RRDs sowohl die Struktur wie auch den Inhalt von einem bestehenden RRD beziehen. *Finanziert von check_MK*

```
rrdtool create --template s.rrd y.rrd
rrdtool create --source a.rrd \
   DS:a=b:DERIVE:300:0:1 RRA:AVERAGE:0.5:1:100
```

danke fetch-Callback Support kann rrdtool graph beliebige Datenquellen nutzen.

```
RRDs::fetch_register_callback sub {
   my $request = shift;
   ...
   return { start => $unix_timestamp,
            step => $step_width,
            data => { dsName1 => [ value1, …]},
   }};
RRDs::graph( …, 'DEF:a=cb//string:ds1:AVERAGE');
```

key value parser für rrdtool graph FCNT:p1:[p2][:k1[=v1][:k2[=v2]]...

```
DEF:ds1=a.rrd:a:AVERAGE:step=7200:reduce=MAX
LINE1:ds1#FF0000:test:dashes=15,5:dash-offset=10
```

clone rrdtool master von github:
https://github.com/oetiker/rrdtool-1.x

```
> git clone git@github.com:oetiker/rrdtool-1.x.git
> cd rrdtool-1.x
> ./bootstrap
> ./configure && make install
```

and one more thing …

interaktive rrdgraphen im Web

http://tobi.oetiker.ch/rrdgraph.js