RRDTool

RRDtool — это небольшая утилита, которая замечательно делает три вещи:

1. Создаёт циклические базы данных (Round-Robin Databases, RRDs),
2. записывает в них данные,
3. и создаёт графики на основе того, что туда записала.

RRD (Round-Robin Databases) это такая специальная база данных, которая, начиная с какого-то момента, записывает новые значения поверх старых. Например, если базе полагается хранить 7 дней данных, то восьмой день запишется поверх первого, девятый — поверх второго, и так далее.

```
rrdtool create cpu.rrd \
      --step 10 \
      DS:cpu:GAUGE:20:0:100 \
      RRA:AVERAGE:0.5:6:120
```

Первая строка — создаём БД по имени *cpu.rrd*. В нашем случае это циклическая база данных для хранения метрик процессора. Следующий параметр — *–step10* — задаёт, как часто мы будем записывать новую порцию данных. В данном случае — раз в 10 секунд. Если данные придут раньше или позже, rrdtool интерполирует значения и выровняет их по десятисекундным границам. Последние две строки описывают *источник данных* и *архив*, в котором они будут храниться.

Источник данных (Data source, DS)

DS параметр описывает источник данных. То, что он есть, еще совсем не значит, что данные будут сохранены — для этого нужны архивы. И в одной базе может быть несколько источников данных.

Наш параметр — DS:cpu:GAUGE:20:0:100 — буквально значит следующее:

1. Создай источник данных (*DS)*
2. под названием *cpu*
3. и с типом *GAUGE*.
4. Если в течение *20 секунд* (heartbeat interval) новых данных не поступает — сохраняем UNDEFINED.
5. Значения варьируются между *0*
6. и *100*.

Тип *GAUGE* говорит rrdtool интерпретировать входящие значения «как есть». Есть еще типы *COUNTER*, *DERIVE* и *ABSOLUTE*, для которых rrdtool сохраняла бы не значение, а скорость, с которой они меняются (берем текущее значение, вычитаем предыдущее, делим на –step). С шагом *–step* , *heartbeat*-параметром и встроенной интерполяцией раз в 10 секунд в базе будет появляться новое значение, нравится нам это, или нет. Это значение называются *первичной точкой данных* (Primary Data Point, PDP)

Циклический архив (Round-robin archive, RRA)

RRA — это агрегированное временное окно с данными. Как и в случае с DS, в базе их может быть несколько. Наш единственный архив RRA:AVERAGE:0.5:6:120 расшифровывается так:

1. Создай циклический архив (*RRA*)
2. длиной в 120 элементов,
3. каждый из которых это среднее (*AVERAGE*)
4. от *6-ти* первичных точек данных.
5. Если больше половины (*>0.5*) значений в шестерке — UNDEFINED, в архив так же идёт UNDEFINED.

В итоге получится архив, который будет хранить поминутные (10 секунд * 6) средние значения загрузки CPU за последние два часа (10 секунд * 6 * 120). Один элемент архива называется консолидированной точкой данных (Consolidated Data Point, CDP), а AVERAGE — консолидирующей функцией (Consolidation Function, CF). Первичные и консолидированные точки связаны между собой вот так:

- $CDP_1 = CF(PDP_1, PDP_2, \ldots, PDP_n)$
- $CDP_2 = CF(PDP_{n+1}, \ldots, PDP_{2n})$
- и т. д.

Кроме AVERAGE есть и другие консолидирующие функции: MIN, MAX, и LAST.

Добавляем данные в RRD

Добавление новых данных в циклическую базу тоже идёт через rrdtool. Так как время влияет на то, как и куда будут сохраняться значения, его нужно передавать вместе с ними. Например, можно добавить значение с меткой «сейчас»:

```
rrdtool update cpu.rrd N:51 # Now:51%
```

или с юниксовым временем (количеством секунд, прошедших с 1970-01-01):

```
rrdtool update cpu.rrd 1482814719:52 # Tue, 27 Dec 2016 04:58:39 GMT:52%
```

или с количеством секунд «назад»:

```
rrdtool update cpu.rrd -- -15:3 # 15 seconds ago:3%
```

Кроме цифровых значений в базу можно сохранить *‘U’* — UNDEFINED.

Хотя данные и можно добавлять руками через командную строку, будет разумнее, если это будет делать скрипт или другая программа.

Создание графиков из RRD

rrdtool умеет не только хранить данные, но и рисовать. Команда для создания графиков может принимать на вход миллион параметров, но для демки хватит всего нескольких основных.

```
rrdtool graph cpu.png \ # Create graph cpu.png
     -s 'end-30m' \ # for data range starting 30mins ago
     -e 'now' \ # and until now
```

```
    -w '700' -h '350' \ # width/height: 700/350
    -u 40 \ # y-axe upper bound: 40
    -t 'cpu-0' \ # title 'cpu-0'
    -v 'Jiffies' # vertical title: 'Jiffies'
    'DEF:user_avg=cpu-user.rrd:value:AVERAGE' \ # import from
cpu-user.rrd
     'CDEF:user_clean=user_avg,UN,0,user_avg,IF' \ # replace
UNDEFINED with 0
    'DEF:system_avg=cpu-system.rrd:value:AVERAGE' \ # import
from cpu-system.rrd
     'CDEF:system_clean=system_avg,UN,0,system_avg,IF' \ #
replace UNDEFINED with 0
    'CDEF:user_stack=system_clean,user_clean,+' \ # calculate
new 'user' series built on top of 'system'
     'AREA:user_clean#FFF000:user' \ # Draw yellow area for
'user_clean' definition with 'user' legend
    'AREA:system_clean#FF0000:system' \ # Draw red area for
'system_clean' definition with 'system legend
    'LINE1:user_clean#FF0000' # Draw thin red line on top of
'user_clean' area
```

Первая половина аргументов достаточно понятна: задать заголовок, размеры изображения, и т. п. Вторая половина, которая, собственно, и отвечает за отрисовку, выглядит более загадочно.

DEF (Definition), по сути, это объявление переменной, а строка целиком – ‘DEF:user_avg=cpu-user.rrd:value:AVERAGE’ – означает: «начиная с этого момента под *user_avg* имеется ввиду циклический архив со средними для DS с именем *value,*что в cpu-user.rrd’. Так как в cpu-user.rrd может быть больше чем один архив, rrdtool выберет тот, который по временному диапазону больше подходит к данному графику.

CDEF (Calculated Definition), с другой стороны, это тоже объявление, но вычисляемое. Его значение – математическое или логическое выражение в [обратной польской записи](), через которое обычно пропускают ряд с данными. В нашем примере таких три. При помощи первых двух, вроде user_clean=user_avg,UN,0,user_avg,IF , мы объявляем

user_clean и system_clean, которым присваиваем значения из user_avg и system_avg, но нулями вместо UNDEFINED. Третье выражение –user_stack=system_clean,user_clean,+ – вводит новый ряд, который суммирует user_clean и system_clean, чтобы получить общую загрузку

Наконец, AREA и LINE1 отвечают за саму отрисовку. На вход им подаётся переменная с данными, цвет, и, опционально, имя для легенды. Есть еще LINE2 и LINE3, которые отличаются только толщиной линии.

Краткое введение в rrdtool

https://ru.wikibooks.org/wiki/RRDtool

https://www.ylsoftware.com/news/644