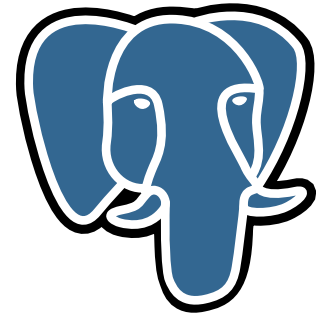
PostRR – PostgreSQL Round Robin Extension

Time-series Datensätze in PostgreSQL

Sebastian 'tokkee' Harl
<sh@tokkee.org>

Hacker

PGConf.DE 2013
08. November 2013
Oberhausen



Disclaimer

HINWEIS:
PostRR ist bestenfalls als pre-alpha Software zu betrachten.

Flaming, bashing oder andere Formen von konstruktivem Feedback sind sehr willkommen! :-)

Überblick: RRDtool

RRDtool – Überblick

- RRDtool ist der de-facto Standard zum Sammeln und Darstellen von Zeit-basierten Werten
- vielfältige Möglichkeiten zur Darstellung
- einfaches Setup
- feste Datenbank-Größe

- Grundidee:
  - feste Anzahl an gespeicherten Werten ($\rightarrow$ Optimierung für Darstellung)
  - Konsolidierung von Werten, um verschiedene Zeitspannen abzudecken

RRDtool – Probleme

- sehr unflexible Speicherlösung (Anpassungen an bestehenden RRD Dateien sind ein PITA)
- I/O Probleme in großen Setups ($\rightarrow$ RRDCacheD; aber ohne Clustering, Replikation, etc.)
- fehlende (sehr eingeschränkte) Netzwerk-Unterstützung
- eingeschränkte Unterstützung zur Datenabfrage
- schwierige Korrelation mit anderen Daten

- NB: das ist auch nicht das Ziel von RRDtool

Idee

- ein RDBMS löst einige dieser Probleme (bringt dafür aber neue)
- Schwerpunkt: Nutzung der Vielfältigen Möglichkeiten zur Datenabfrage, -korrelation und -verarbeitung
- Wichtig: nicht als drop-in Ersatz für RRDtool gedacht
- RRDCacheD bleibt nach wie vor eine gute / bessere Lösung für große RRD-Setups

Performance-Daten in PostgreSQL speichern

Probleme

Performance-Daten in PostgreSQL speichern

Folgende Themen müssen beachtet werden ...

- Datenbank-Layout (Identifier, Zeitstempel, Werte)
- Identifier können hierarchisch sein
- Partitionierung von Daten
- graphische Auswertung / Darstellung
- Behandlung von alten Daten

PostgreSQL Round-Robin Extension (PostRR)

PostgreSQL Round-Robin Extension

- `https://github.com/tokkee/postrr`
- BSD Lizenz
- neue Datentypen und Funktionen für PostgreSQL
- angelehnt an einige Ideen aus RRDtool
- flexible Nutzungsmöglichkeiten
- sehr frühes pre-Alpha Stadium ;-)

Warum?

- Interesse an PostgreSQL Backend-Programmierung ;-)
- flexible Abfrage von Daten
- Verknüpfung mit anderen (Meta-)Daten
  (z.B. geographisch Informationen)

- Ziele: flexible und generische Implementierung
  $\Rightarrow$ einfaches einfach und schwieriges möglich

Warum?

- Interesse an PostgreSQL Backend-Programmierung ;-)
- flexible Abfrage von Daten
- Verknüpfung mit anderen (Meta-)Daten
  (z.B. geographisch Informationen)

- Ziele: flexible und generische Implementierung
  $\Rightarrow$ einfaches einfach und schwieriges möglich
  (soweit ist PostRR noch nicht)

`RRTimeslice` Datentyp

- Round-Robin Timeslice
- basiert auf `timestamptz`
- Aufteilung eines Zeitstrangs in Abschnitte fester Länge
- zusätzliche Attribute:
  - `length`: Länge einer Zeitscheibe
  - `count`: Anzahl an Zeitscheiben (Round-Robin)
- $\rightarrow$ Definition wird in interner Tabelle gespeichert
- Sequenznummer identifiziert eine Zeitscheibe
- btree Index auf Sequenznummer (TODO: Zeitstempel)
- Casts von und zu `timestamp`
- Usage: `RRTimeslice(<len>, <count>)`

RRTimeslice Beispiel

```
-- rrtimeslice(60, 10)
db=# select ts from table order by ts;
                                ts
-----------------------------------------------------
 ("2013-11-07 18:07:00", "2013-11-07 18:08:00"] #8/10
 ("2013-11-07 18:08:00", "2013-11-07 18:09:00"] #9/10
 ("2013-11-07 18:09:00", "2013-11-07 18:10:00"] #0/10
 ("2013-11-07 18:10:00", "2013-11-07 18:11:00"] #1/10
 ("2013-11-07 18:11:00", "2013-11-07 18:12:00"] #2/10
 ("2013-11-07 18:12:00", "2013-11-07 18:13:00"] #3/10
 ("2013-11-07 18:13:00", "2013-11-07 18:14:00"] #4/10
 ("2013-11-07 18:14:00", "2013-11-07 18:15:00"] #5/10
 ("2013-11-07 18:15:00", "2013-11-07 18:16:00"] #6/10
 ("2013-11-07 18:16:00", "2013-11-07 18:17:00"] #7/10
```

`CData` Datentyp

- konsolidierte Daten
- basiert auf `double precision`
- unterstützt eingebaute Konsolidierungsfunktionen (AVG, MIN, MAX)
- speichert mehrere konsolidierte Werte in einem Datenpunkt (plus Meta-Informationen wie Anzahl Datenpunkt und undefinierte Werte)
- Casts von und zu numerischen Datentypen
- Usage: `CData(<AVG>|<MIN>|<MAX>)`
- `CData_update(cdata, cdata)` – Konsolidierung zweier Werte

CData Beispiel

```
-- cdata('MIN'), cdata('AVG'), cdata('MAX')
db=# select min, avg, max from table;
       min          |        avg          |        max
--------------------+---------------------+--------------------
 240 (MIN U:0/4)    | 1345 (AVG U:0/4)    | 2386 (MAX U:0/4)
 1561 (MIN U:0/4)   | 2246 (AVG U:0/4)    | 3005 (MAX U:0/4)
 40 (MIN U:0/4)     | 511 (AVG U:0/4)     | 1084 (MAX U:0/4)
 374 (MIN U:0/4)    | 1424 (AVG U:0/4)    | 2519 (MAX U:0/4)
 132 (MIN U:0/24)   | 1801 (AVG U:0/24)   | 3201 (MAX U:0/24)
 14 (MIN U:0/4)     | 1115 (AVG U:0/4)    | 3206 (MAX U:0/4)
 134 (MIN U:0/14)   | 1618 (AVG U:0/14)   | 3259 (MAX U:0/14)
 547 (MIN U:0/4)    | 1795 (AVG U:0/4)    | 2976 (MAX U:0/4)
 381 (MIN U:0/4)    | 1477 (AVG U:0/4)    | 2798 (MAX U:0/4)
 3 (MIN U:0/5)      | 887 (AVG U:0/5)     | 2456 (MAX U:0/5)
```

Anwendung

	avg_data	max_data
2013-11-03	42 (0/5)	4223 (0/5)
2013-11-04	23 (0/5)	2342 (0/5)
2013-11-05	64 (0/5)	4711 (0/5)

Anwendung

	avg_data	max_data
2013-11-03	42 (0/5)	4223 (0/5)
2013-11-04	23 (0/5)	2342 (0/5)
2013-11-05	64 (0/5)	4711 (0/5)
2013-11-06	83 (0/1)	83 (0/1)

Anwendung

	avg_data	max_data
2013-11-03	42 (0/5)	4223 (0/5)
2013-11-04	23 (0/5)	2342 (0/5)
2013-11-05	64 (0/5)	4711 (0/5)
2013-11-06	83 (0/1)	83 (0/1)
2013-11-07	5 (0/1)	5 (0/1)

Anwendung

	avg_data	max_data
2013-11-08	128 (0/1)	128 (0/1)
2013-11-04	23 (0/5)	2342 (0/5)
2013-11-05	64 (0/5)	4711 (0/5)
2013-11-06	83 (0/1)	83 (0/1)
2013-11-07	5 (0/1)	5 (0/1)

Round Robin Archives / Stores

- Name / Identifier für mehrere Archive (vgl. RRA in RRDtool)
- "Multiplexer" um einen Wert in mehrere Tabellen oder Spalten einzufügen
- `PostRR_update(<rra_name>, <timestamp>, <value>)` fügt einem Archiv einen neuen Wert hinzu
- Überschreiben alter Werte an Hand ihrer Sequenznummer
- Definition wird in `postrr.rrarchives` gespeichert

Round Robin Archives: Definition

```
db=# \d postrr.rrarchives
 Table "postrr.rrarchives"
 Column  | Type | Modifiers
---------+------+-----------
 rraname | text | not null
 tbl     | name | not null
 tscol   | name | not null
 vcol    | name | not null
```

Inserting Data

```
db=# select postrr_update('name', 'now', 10);
 postrr_update
---------------
 10 (AVG U:0/1)
 10 (MIN U:0/1)
 10 (MAX U:0/1)

db=# select postrr_update('name', 'now', 100);
...
db=# select postrr_update('name', 'now', 1);
  postrr_update
-----------------
 37 (AVG U:0/3)
 1 (MIN U:0/3)
 100 (MAX U:0/3)
```

Post-Processing

```
db=# \d raw
              Table "public.raw"
 Column |           Type           | Modifiers
--------+--------------------------+-----------
 ts     | timestamp with time zone |
 value  | double precision         |

db=# select ts::rrtimeslice(14400,1000)::timestamptz as s,
db-#     count(*), avg(value) from raw group by s;
           s             | count |       avg
-------------------------+-------+-----------------
 2013-11-06 09:00:00+01 |    46 | 15772.7608695652
 2013-11-06 05:00:00+01 |    43 | 16683.1627906977
 2013-11-06 13:00:00+01 |    11 | 16965.8181818182
```

Ausblick

- Umbau von Archiven
  - Identifier und Speicherung mehrer Archive in einer Tabelle
  - mehr Kapselung (optional): automatische Verwaltung (mit Helfer-Funktionen) im “postrr” Schema
- Erweiterungen für `CData`
  - ähnlich zu eigenen Datentypen in PostgreSQL
  - Benutzer definiert Callback-Funktionen und registriert diese als Konsolidierungsfunktion
  - $\Rightarrow$ andere Datentypen und Konsolidierung

Performance

Random ideas:

- Background Process / Cron-Job der Roh-Daten periodisch in Archive kopiert
- Archiv-Umbau
- weitere Indizes / kein Index (?)
- $\Rightarrow$ zunächst sollte es erst einmal genutzt werden und Performance Analyse durchgeführt werden
- ...

Ausblick: Graphing

Ausblick: Graphing

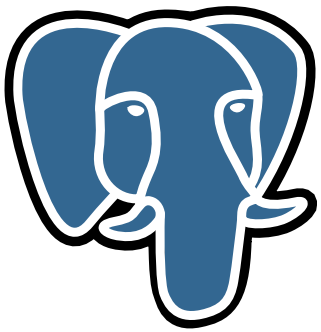
PostRR bietet flexbile Möglichkeiten zur Verarbeitung von Werten, aber keine Graphing-Möglichkeiten.

- `rrdgraph_libdbi` könnte erweitert werden, um PostRR zu unterstützen (flexiblere Queries müssten unterstützt werden)
- PostgreSQL CSV Ausgabe mit Gnuplot verarbeiten
  - Helferskripte gewünscht :-)
  - Vorteil: sehr mächtig und viele Backends verfügbar
- Graphite DB Backend?
- ...

PostRR – PostgreSQL Round Robin Ext.

Danke für die Aufmerksamkeit!

Fragen, Kommentare, Anmerkungen, Rants?



PostRR – PostgreSQL Round Robin Ext.

Kontakt:
Sebastian “tokkee” Harl
<sh@tokkee.org>

https://github.com/tokkee/postrr

Send patches! :-)