collectd

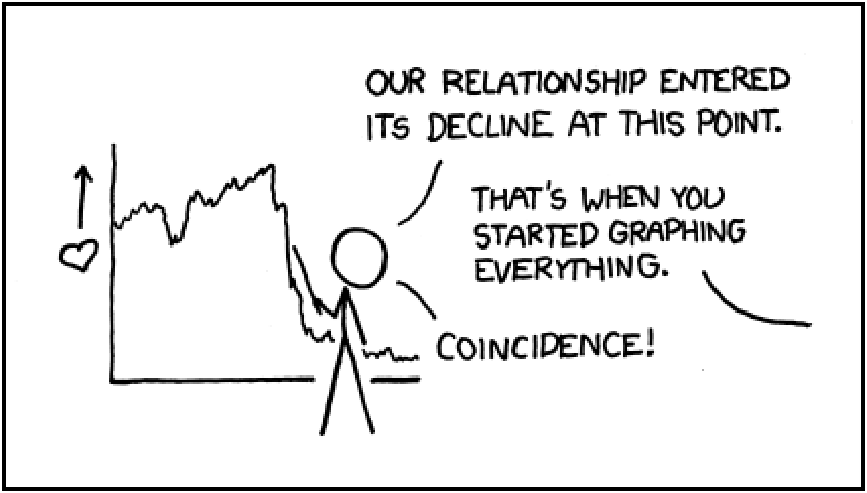
Performancedaten und Monitoring

Sebastian ‘tokkee’ Harl
<tokkee@collectd.org>

OPEN
RHEIN
RUHR
Ein Pott voll Software

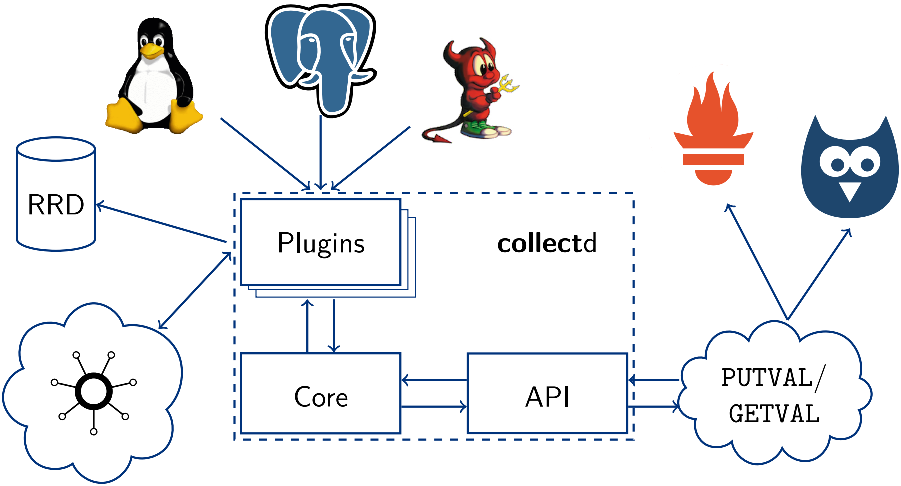
Überblick

- `https://collectd.org/`
- Daemon zum Sammeln, Verarbeiten und Speichern von Performance-Daten
- Einfach erweiterbar, viele Integrationsmöglichkeiten
- Open Source: MIT und GPL
- Platformunabhängig: div. Unixe, Windows*


OUR RELATIONSHIP ENTERED ITS DECLINE AT THIS POINT.
THAT'S WHEN YOU STARTED GRAPHING EVERYTHING.
COINCIDENCE!
↑ ♡

https://xkcd.com/523

Architektur



Modulare Architektur

- 95+ Lese-Plugins
  - System-Metriken, z.B. CPU, Speicher
  - Anwendungsmetriken, z.B. PostgreSQL
  - Vieles Weiteres, z.B. SNMP, OneWire
- 15+ Ausgabe-Plugins
  - z.B. Graphite, RRDTool
  - HTTP
  - MongoDB, PostgreSQL

Input

Konfiguration

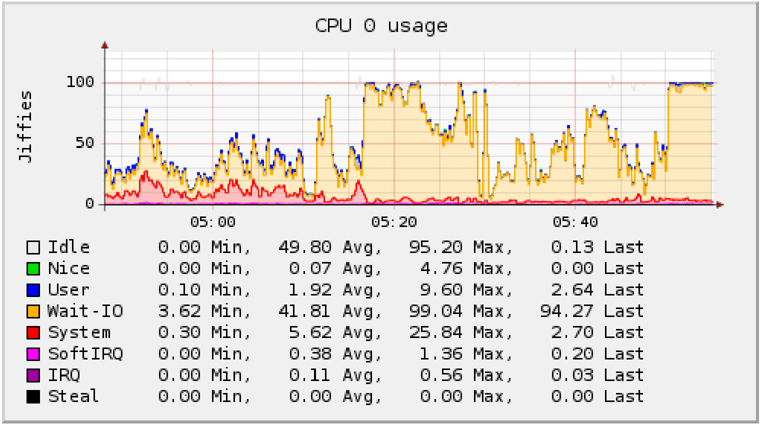
```
LoadPlugin "cpu"
LoadPlugin "memory"
LoadPlugin "interface"
```

Input

Konfiguration

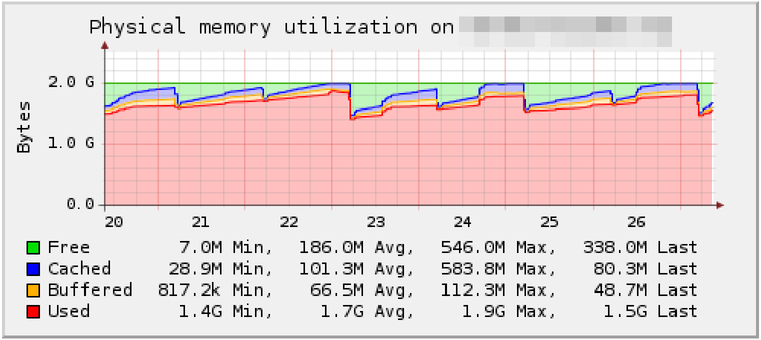
```
LoadPlugin "cpu"
LoadPlugin "memory"
LoadPlugin "interface"

<Plugin interface>
  Interface lo
  Interface sit0
  IgnoreSelected true
</Plugin>
```

CPU, memory, network I/O



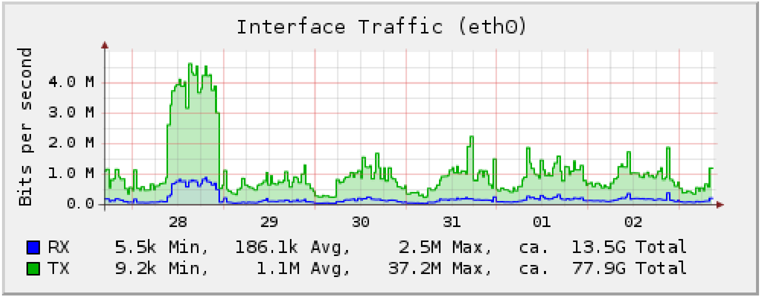
```
□ Idle        0.00 Min,  49.80 Avg,  95.20 Max,   0.13 Last
🟩 Nice        0.00 Min,   0.07 Avg,   4.76 Max,   0.00 Last
🟦 User        0.10 Min,   1.92 Avg,   9.60 Max,   2.64 Last
🟧 Wait-IO     3.62 Min,  41.81 Avg,  99.04 Max,  94.27 Last
🟥 System      0.30 Min,   5.62 Avg,  25.84 Max,   2.70 Last
🟪 SoftIRQ     0.00 Min,   0.38 Avg,   1.36 Max,   0.20 Last
🟪 IRQ         0.00 Min,   0.11 Avg,   0.56 Max,   0.03 Last
■ Steal       0.00 Min,   0.00 Avg,   0.00 Max,   0.00 Last
```

CPU, memory, network I/O



🟩	Free	7.0M Min,	186.0M Avg,	546.0M Max,	338.0M Last
🟦	Cached	28.9M Min,	101.3M Avg,	583.8M Max,	80.3M Last
🟧	Buffered	817.2k Min,	66.5M Avg,	112.3M Max,	48.7M Last
🟥	Used	1.4G Min,	1.7G Avg,	1.9G Max,	1.5G Last

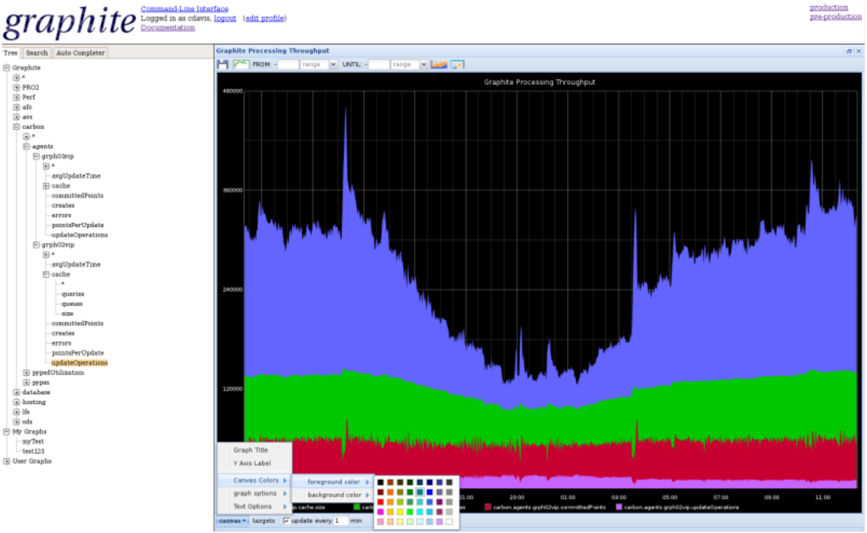
CPU, memory, network I/O



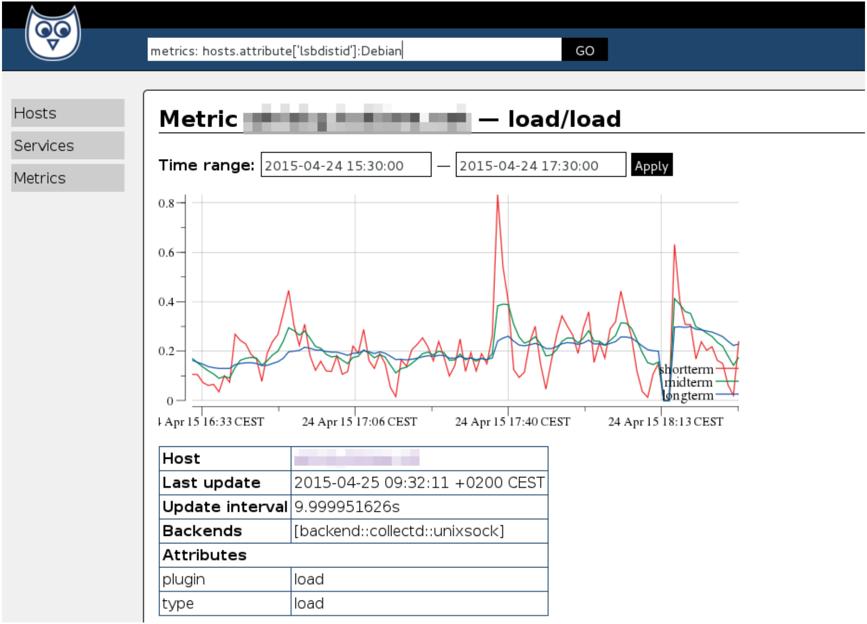
Output

Konfiguration

```
LoadPlugin write_graphite
<Plugin write_graphite>
  <Node "default">
    Host "graphite.example.com"
  </Node>
</Plugin>
```



https://graphiteapp.org/



metrics: hosts.attribute['lsbdistid']:Debian

Metric [redacted] — load/load

Time range: 2015-04-24 15:30:00 — 2015-04-24 17:30:00

Host	[redacted]
Last update	2015-04-25 09:32:11 +0200 CEST
Update interval	9.999951626s
Backends	[backend::collectd::unixsock]
Attributes	
plugin	load
type	load

https://sysdb.io/

Putput

Weiterverarbeitung von Metriken

- Z.B. Aggregierung von Metriken
  - Vermeidung von unnötigem I/O
  - Überblicksgraphen
- Auswahl der betroffenen Metriken
- Zusammenführen durch Grouping
- Erzeugt neue Metriken

Aggregierung

Konfiguration

```
LoadPlugin Aggregation
<Plugin aggregation>
  <Aggregation>




  </Aggregation>
</Plugin>
```

```
example.com/cpu-1/cpu-idle
example.com/cpu-1/cpu-user
example.com/cpu-1/cpu-wait
example.com/cpu-2/cpu-idle
example.com/cpu-2/cpu-user
example.com/cpu-2/cpu-wait
example.com/user/user
example.com/load/load
example.com/swap/swap-cached
example.com/swap/swap-free
example.com/swap/swap-used
...
```

Aggregierung

Konfiguration

```
LoadPlugin Aggregation
<Plugin aggregation>
  <Aggregation>
    Plugin cpu
    Type cpu



  </Aggregation>
</Plugin>
```

```
example.com/cpu-1/cpu-idle
example.com/cpu-1/cpu-user
example.com/cpu-1/cpu-wait
example.com/cpu-2/cpu-idle
example.com/cpu-2/cpu-user
example.com/cpu-2/cpu-wait
```

Aggregierung

Konfiguration

```
LoadPlugin Aggregation
<Plugin aggregation>
  <Aggregation>
    Plugin cpu
    Type cpu
    GroupBy Host



  </Aggregation>
</Plugin>
```

```
example.com/cpu-???/cpu-???
```

Aggregierung

Konfiguration

```
LoadPlugin Aggregation
<Plugin aggregation>
  <Aggregation>
    Plugin cpu
    Type cpu
    GroupBy Host
    GroupBy TypeInstance

  </Aggregation>
</Plugin>
```

```
example.com/cpu-???/cpu-idle
example.com/cpu-???/cpu-user
example.com/cpu-???/cpu-wait
```

Aggregierung

Konfiguration

```
LoadPlugin Aggregation
<Plugin aggregation>
  <Aggregation>
    Plugin cpu
    Type cpu
    GroupBy Host
    GroupBy TypeInstance
    CalculateSum true
  </Aggregation>
</Plugin>
```

```
example.com/cpu-sum/cpu-idle
example.com/cpu-sum/cpu-user
example.com/cpu-sum/cpu-wait
```

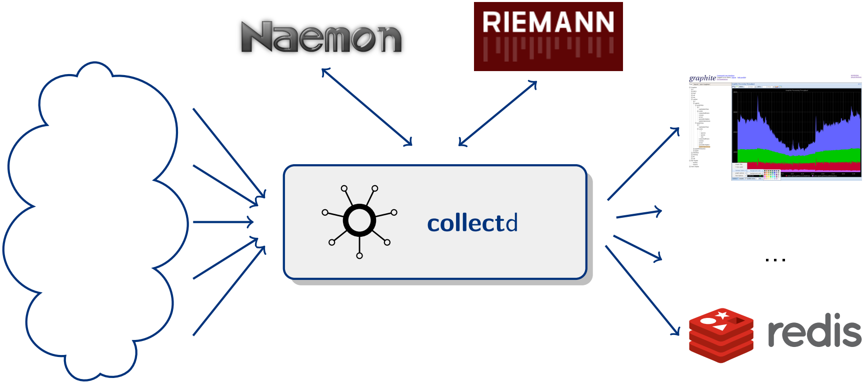
Integration

- UNIXSOCK, gRPC Plugins
- RPC Schnittstellen
- Abfragen und Speichern
  - `PUTVAL`, `FLUSH`, `LISTVAL`, `GETVAL`
- Client Werkzeuge
  - `collectd-nagios`
  - `collectdctl`

UNIXSOCK Plugin: Beispiel

```
-> | GETVAL "FQDN/load/load"
<- | 3 Values found
<- | shortterm=4.000000e-02
<- | midterm=6.000000e-02
<- | longterm=7.000000e-02

-> | PUTVAL FQDN/users/users 1341851406:42
<- | 0 Success: 1 value has been dispatched.
```

Monitoring



collectd

…

Monitoring: Riemann

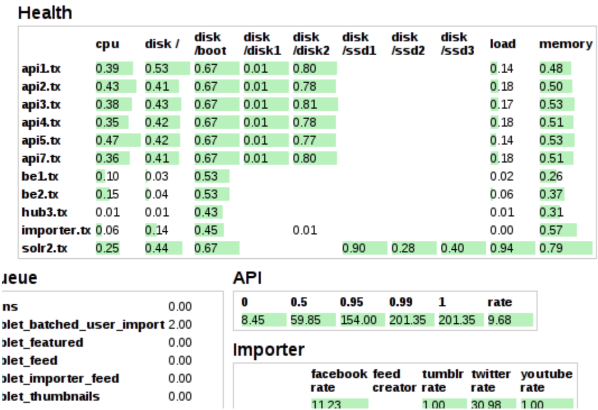
- In Clojure geschrieben und konfiguriert
- Filtern, Verarbeitung und Kombination von Event-Streams
- Einfache Erweiterung durch Clients (Kommunikation mittels Protobufs)
- Läuft auf JVMs und damit in **collect**d :-)
  → collmann (proof-of-concept)
  → Besser: `write_riemann`
- Kann Daten auch zu Graphite schreiben

Riemann: Konfiguration

- http://riemann.io/howto.html
- http://riemann.io/api.html

```
(where (or (service #"^api")
           (service #"^app"))
  (where (tagged "exception")
    (rollup 5 3600
      (email "dev@foo.com"))
    (else
      (changed-state
        (email "ops@foo.com")))))
```

Riemann: Dashboard



Health

	cpu	disk /	disk /boot	disk /disk1	disk /disk2	disk /ssd1	disk /ssd2	disk /ssd3	load	memory
api1.tx	0.39	0.53	0.67	0.01	0.80				0.14	0.48
api2.tx	0.43	0.41	0.67	0.01	0.78				0.18	0.50
api3.tx	0.38	0.43	0.67	0.01	0.81				0.17	0.53
api4.tx	0.35	0.42	0.67	0.01	0.78				0.18	0.51
api5.tx	0.47	0.42	0.67	0.01	0.77				0.14	0.53
api7.tx	0.36	0.41	0.67	0.01	0.80				0.18	0.51
be1.tx	0.10	0.03	0.53						0.02	0.26
be2.tx	0.15	0.04	0.53						0.06	0.37
hub3.tx	0.01	0.01	0.43						0.01	0.31
importer.tx	0.06	0.14	0.45		0.01				0.00	0.57
solr2.tx	0.25	0.44	0.67			0.90	0.28	0.40	0.94	0.79

ıeue

ns	0.00
ɔlet_batched_user_import	2.00
ɔlet_featured	0.00
ɔlet_feed	0.00
ɔlet_importer_feed	0.00
ɔlet_thumbnails	0.00

API

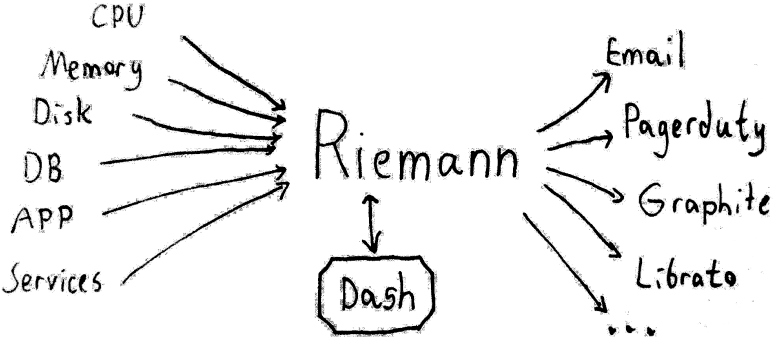
0	0.5	0.95	0.99	1	rate
8.45	59.85	154.00	201.35	201.35	9.68

Importer

	facebook rate	feed creator	tumblr rate	twitter rate	youtube rate
	11.23		1.00	30.98	1.00

`http://riemann.io/`

Riemann: Architektur



`http://riemann.io/`

Danke für die Aufmerksamkeit

Fragen, Kommentare?

Sebastian ‘tokkee’ Harl
<tokkee@collectd.org>