Next generation infrastructure and monitoring : PCP meets Redis and Grafana

PCP-Conf2019 Mark Goodwin

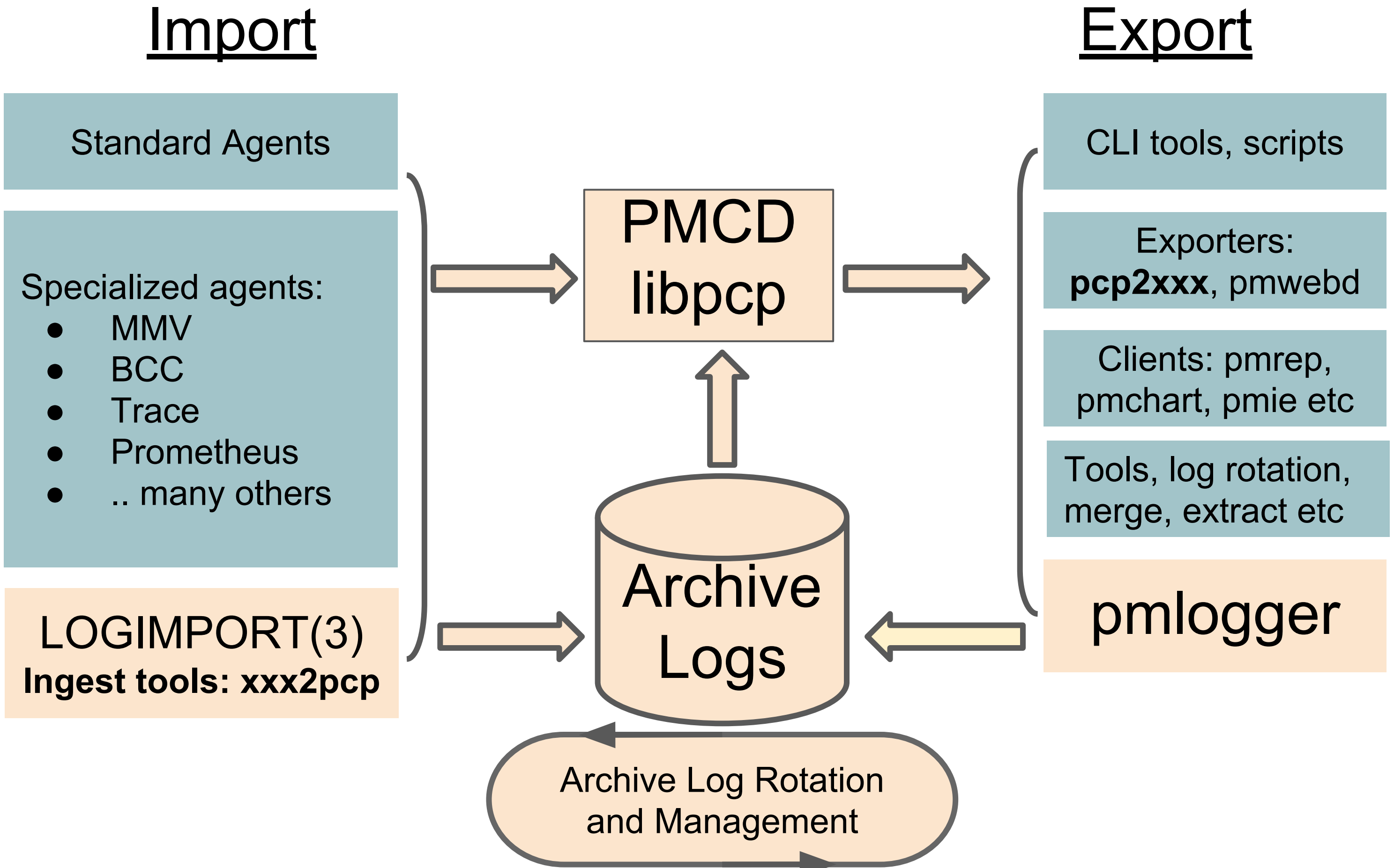
mgoodwin@redhat.com

@goodwinos

Agenda

- PCP Logging infrastructure overview
- Scaling Issues
- PCP extensions, Redis and Grafana
- Native PCP Grafana Data-source
- Grafana + Demos
- Work in progress, Q&A

Core PCP Logging Infrastructure

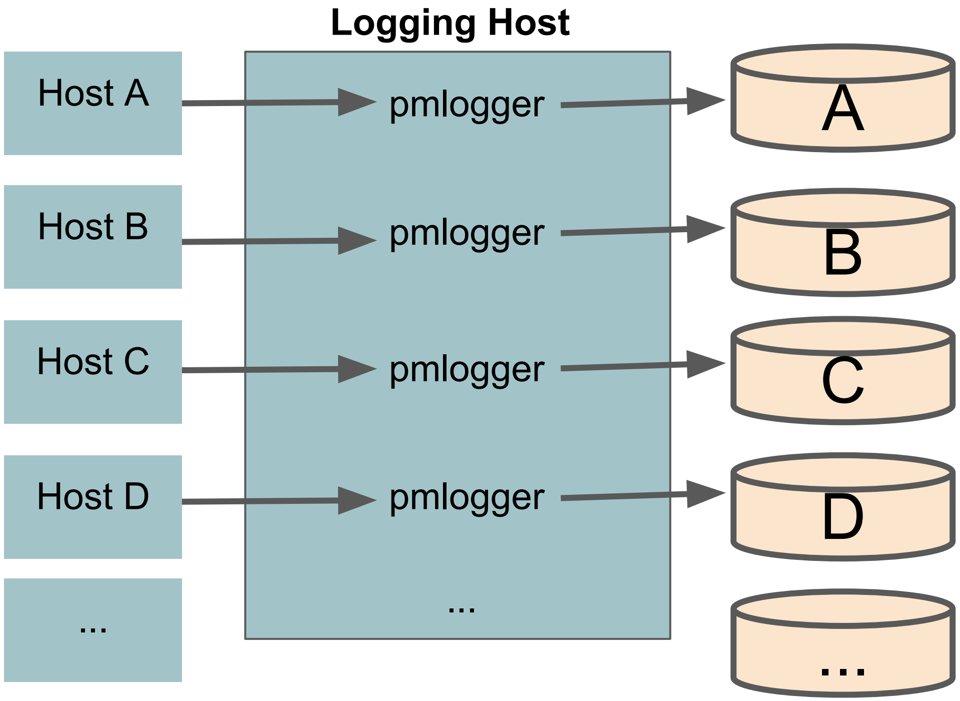


PCP Standard PMDAs (Agents)

- ~ 75 plugins / agents (PMDAs)
  - .. more being added every release
  - Managed by the PCP pmcd service.
  - DSOs and daemons. Lots of IPC options
- Ingest data into PCP metrics
  - Canonical, uniform name space
  - strongly typed metadata and values
  - Low overheads: “Pull” model: service to completion: client request -> pmcd -> agent -> pmcd -> client
- Extensible API
  - libpcp_pmda has C/C++, Python and Perl bindings
- Separately Packaged: pcp-pmda-*foo*
  - Isolate exotic dependencies
  - Not all installed by default.

- **linux** - kernel metrics. CPU, Disk, Network, Memory, Filesystem, etc. everything exported by /proc, /sys and most other kernel interfaces
- **proc** - per-process metrics
- **XFS** - XFS filesystem specific metrics
- **nfsclient** - NFS client stats
- **mmv** - memory mapped instrumentation
- **dm** - device mapper and LVM
- **jbd2** - journal block device
- **lio** - Linux I/O - iSCSI, FCP, FCoE
- **pmcd** - PCP statistics
- **root** - container, privileged PMDAs, etc
- **apache** - web server stats
- **BCC** - Extended Berkley Packet Filter metrics
- **docker** - container management stats
- **KVM** - libvirt
- **mysql** and **postgresql** - database stats
- **prometheus** - end-points
- **redis** - system stats for redis daemons
- **samba** - filesystem
- **smart** - disk health
- **vmware** - platform stats
- … many more.

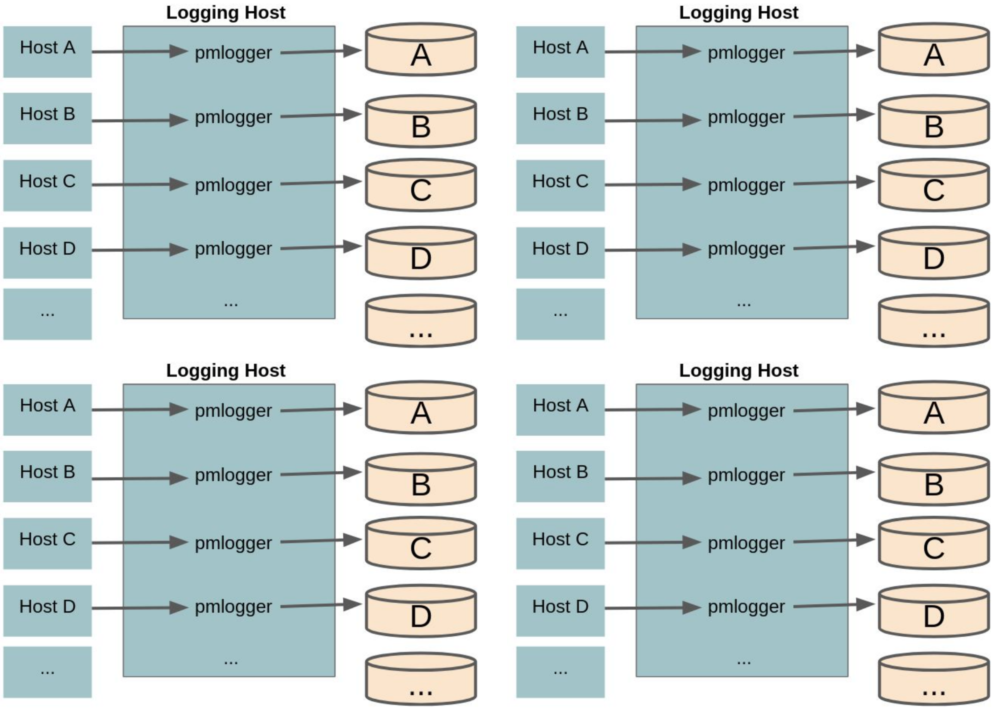
PCP Logger "farm" : 1 Logger host, O(100) hosts



- One directory and archive set per host
  - /var/log/pcp/pmlogger/*HOSTNAME*
- Daily rotation and compression
- Default 14 day retention
- Easy to set up and manage
  - /etc/pcp/pmlogger/control.d/*HOSTNAME*
- Common metrics logging config
  - /var/lib/pcp/config/pmlogger/config.default

- O(50) GB storage per host
- ~ 5 TB total storage
- ~ 1400 archives in 100 directories

PCP Data Centre : ~10 Logger “farms”, O(1000) hosts



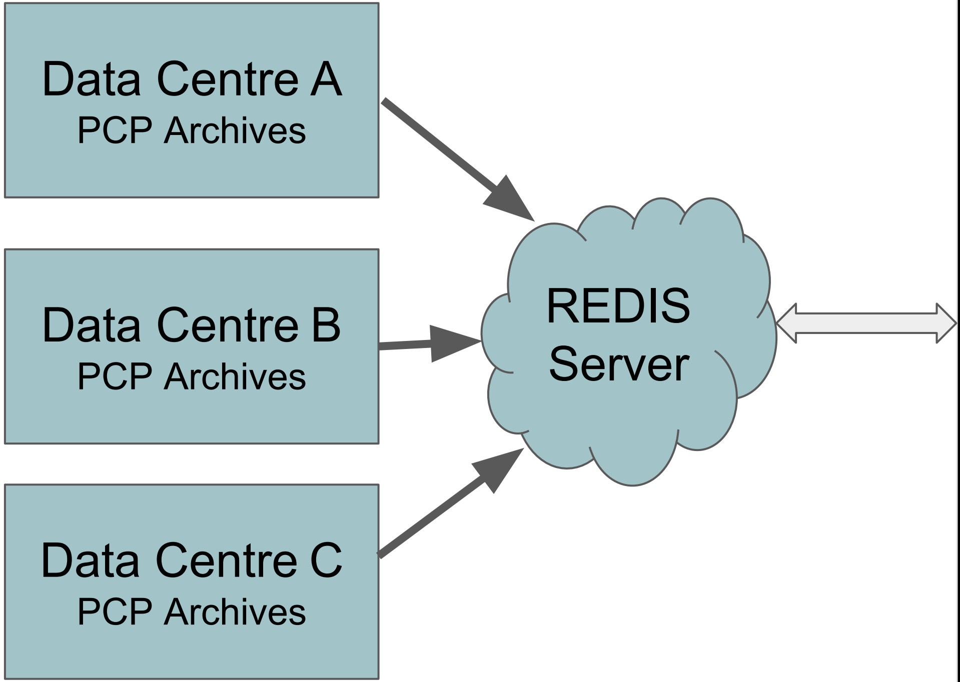
- Multiple logger “farms”
  - Split across application domains
- ~ 5 TB per farm
- ~ 50 TB total storage - just for perf data!
- ~ 14000 archives
- ~ 1000 archive directories
  - spread across ~10 logger hosts
- Difficult to manage so many archives
- Difficult to monitor ~

- How can we scale globally with multiple datacentres?

Scaling

- Original PCP PMAPI was not designed to efficiently query/search across a large number of archives/hosts - one PMAPI context per host or archive
- This has served well for many years, helping to solve countless performance analysis cases involving classic client-server production scenarios
- We recently added “multi-archive” contexts, so monitoring tools could e.g. name a directory (or multiple archives) and the archives would be “stitched together” on the fly. We also added transparent archive decompression.
- This all works, but it can be slow, especially when the archives are compressed and/or have large dynamic instance domains (like per-process data).
- … and it doesn’t scale to thousands of hosts/archives on a global scale

PCP with REDIS & Grafana

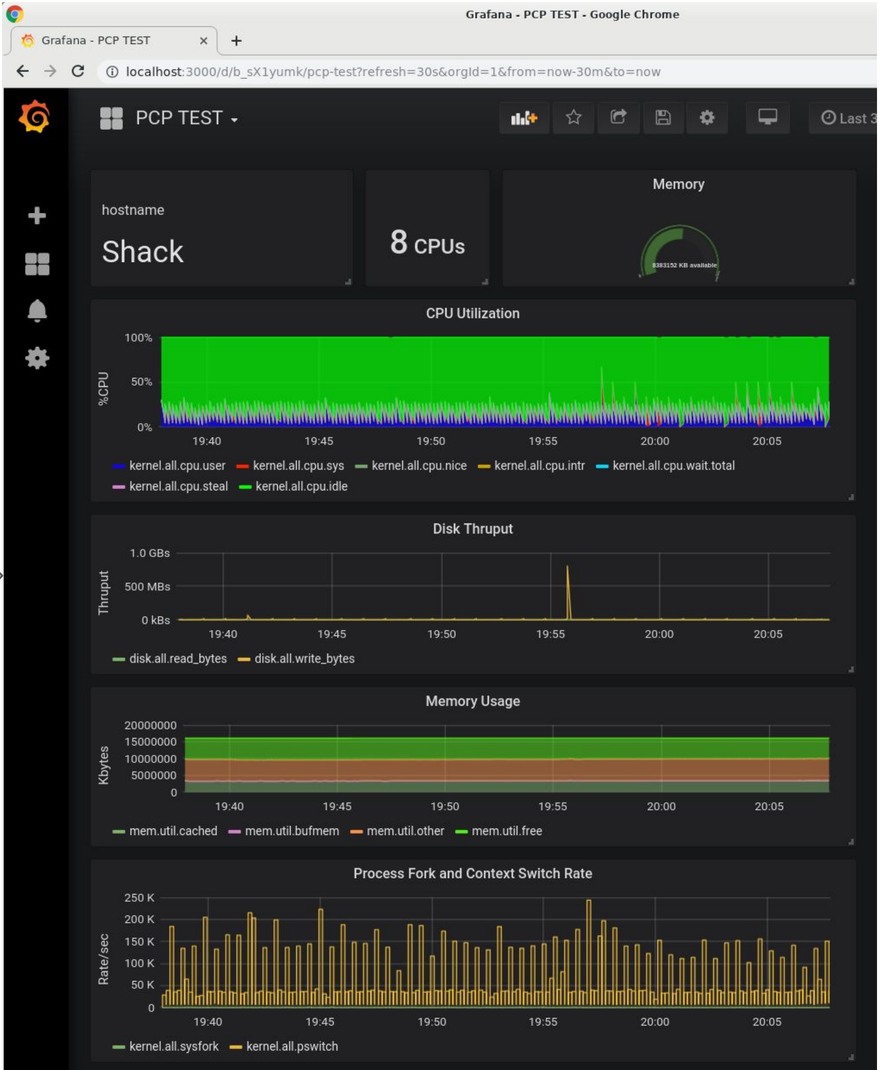


Data Centre A
PCP Archives

Data Centre B
PCP Archives

Data Centre C
PCP Archives

REDIS
Server



Grafana - PCP TEST - Google Chrome

localhost:3000/d/b_sX1yumk/pcp-test?refresh=30s&orgId=1&from=now-30m&to=now

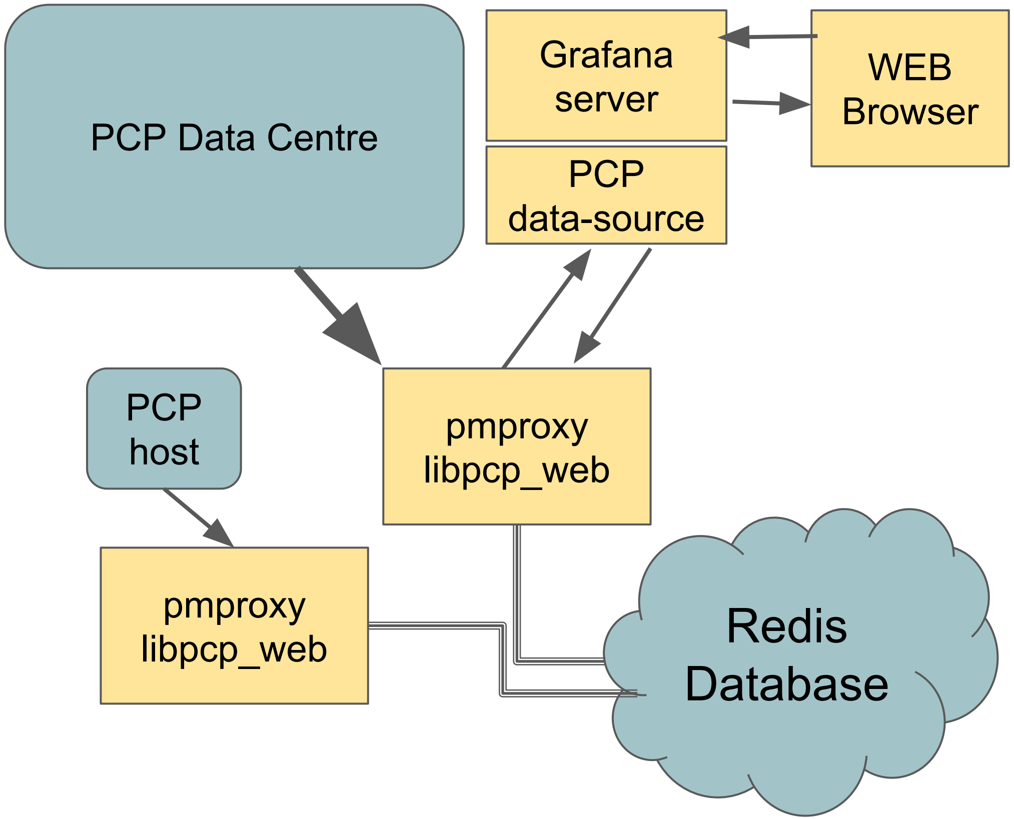
hostname
Shack

8 CPUs

Redis - scalable key-value data store

- http://redis.io
- V5 and later supports native time-series
- Extremely fast/scalable, with extensive indexing for time-series data
  - Runs at ram speed, disk backed cache
  - Configurable retention, automatic FIFO data discard
  - Replication and clustering options
- Secure - authentication, SSL, etc
- Commercial services available, e.g. google-cloud
- Cloud and Hybrid-Cloud (private/public) friendly
  - run your own server, run on
- open-source (and written in C :)
- Used by pmproxy to store, index and query PCP archive data

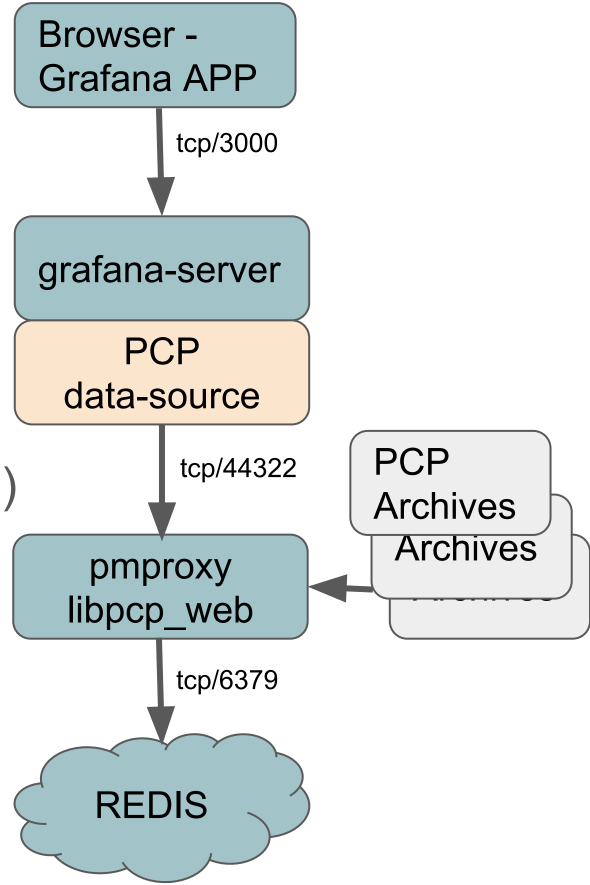
Global: PCP + Redis Database + Grafana Monitor



- **No changes to core PCP** - we extend and enhance pmproxy / libpcp_web
- pmproxy -t discovers and scrapes (aka "logtails") PCP data from archive logs as soon as it is written by pmlogger
  - Fully async using libuv (no polling)
  - -t option Ingests into REDIS
- Near-live data *and* historical data available for queries (no host contexts)
- No fetched live data - has to be logged
- Data for *all* hosts in one DB "unified contexts"
- Extremely fast time-series queries using pmseries query language
- http REST API for webapps - GRAFANA
- Secure - Hybrid/Cloud - global scalability
- Handy for support - no need to upload huge perfdata archives for analysis!

PCP Native Grafana Data-source

- Provides the “glue layer” between Grafana panels and the data back-end (pmproxy).
- Different panels in same dashboard can use different data-sources
- Uses html protocols, can be local or remote
- Supports no auth, basic auth, CA and oauth2 (e.g. GH)
- Implemented in Typescript / javascript
  - ○ https://github.com/goodwinos/pcp-json-datasource
- Same datasource can be configured multiple times, each to a different back-end (host:port)
- Under concurrent development with pmproxy and libpcp_web.
- Not yet packaged.



PCP Grafana - demos

- Install
  - Install grafana builds for Fedora / RHEL https://copr.fedorainfracloud.org/coprs/mgoodwin/grafana/
  - Redis v5 or later: dnf install redis; enable and start redis service
  - PCP 4.3.1 + pmproxy/libpcp_web patch - contact PCP team
    - Enable and start pmproxy service
  - pcp-json-datasource https://github.com/goodwinos/pcp-grafana-datasource
- Configure, test and save pcp-datasource - demo
- Create and rename a Dashboard, add Panels
  - Singlestat, graph, table, text, heatmap, etc. Many others on-line
  - Query syntax - **[metric.name] '{metadata qualifiers}' '[time-window specification]'**
    - Omit the time-window specification - grafana supplies it. See pmseries(1) for details.
    - Metadata qualifiers not yet implemented - instances, hosts, labels etc
- Time controls - demo
  - Absolute intervals, or relative to ‘now’, optional periodic refresh
  - Supplies time-window (&start, &finish and &step) parameters for queries issued by datasource.
- Share Dashboard, add drill-down links - demo
  - Export JSON panels - proposed as a PCP GSOC project

WIP - PCP pmproxy and grafana datasource

- Fix the dropped response write bug (may be a grafana bug, or pmproxy bug?)
- Support metadata qualifiers in queries, see pmseries(1)
    - In data-source, query URL params will need to be encodeURIComponent()'d to escape special chars such as **{ } " ' ?** etc
- Data-source - handle responses with multiple time-series, e.g. when the query matches multiple instances or hosts. **Currently expects and mandates exactly one time-series in the response.**
- Data-source and back-end - support responses in table format - needed by some Grafana panels
- Data-source - use PCP metadata. E.g. metric type (counter, instant), units, scale, help text
- Data-source - allow legend string override - needs a text box next to Query, template variables, etc
- Data-source - use /grafana/search URL to provide query helper/hints, e.g. metric name completion
- libpcp_web - implement functions, e.g. server-side rate conversion, statistical functions
- Authentication, https/ssl in back-end
- Data-source - packaging - grafana-pcp-datasource (in grafana)? pcp-grafana-datasource (in pcp)?
- Packaging Grafana itself in Fedora, see BZ#1670656
- pmproxy import live data into redis
- pmproxy import existing archives into redis (not just log-tail active archives)
- QA tests (for grafana too)