# Scalable Time Series in PCP

Lukas Berk

# Summary

- Problem Statement
- Proposed Solution
- Redis
    - Basic Types Summary
    - Current Work
    - Future Work Items

# Problem Statement

Scaling PCP’s metrics querying to hundreds/thousands of hosts and archives among thousands of metrics

https://groups.io/g/pcp/topic/6059074#17181

# Current Approach (pre pcp4)

Utilize pmlog* tools to create a specific archive from multiple machines

- Requires knowledge of the metrics needed
- Still difficult to pin-point specific instances within a metric
    - Eg: “cpu X on host Y is pinned at 100% for 50 minutes”
- Added labels in pcp-4.0.0

We need the ability to query metrics as needed, without limitation of scope or knowing details of archive/host specifics

# Redis - REmote DIctionary Server

Open Source, portable, in memory data store suitable for cache, or database.

- 8 Built-in Datatypes (including timeseries)
- Highly Portable (ansi C)
- Lua Script Modules
- Extensive Testsuite
- Large Variety of bindings
    - C/C++, Python, Java, Go, (yes Ryan -- even ruby)

# Redis - Cont

Basic Datatypes:

- Strings
- Hashes
- Lists
- Sets (and Sorted Sets)
- Streams (timeseries)

# Redis Types - Strings

SET/GET KEY [VALUE]

```
> SET FOO “BAR”
(integer) 1

> GET FOO
“BAR”

> GET pcp:version:schema
“2”
```

# Redis Types - Hashes

HSET/HGET KEY FIELD [VALUE]

```
> HSET myhash field1 “Hello.”
(integer) 1

> HGET myhash field1
“Hello.”

> HSET pcp:map:inst.name “1 minute” 14
(integer) 1
```

# Redis Types - Lists

RPUSH/LPUSH KEY VALUE | RPOP/LPOP KEY

```
> RPUSH foo “bar”
(integer) 1

> RPUSH foo “baz”
(integer) 2

> LPOP foo
“bar”
```

# Redis Types - Sets

SADD KEY MEMBER [MEMBER…]

```
> SADD books “hobbit” “1984” “titanic” “don quixote”
(integer) 4
```

```
> SADD movies “solo” “avengers” “titanic” “hobbit”
(integer) 4
```

```
> SINTER books movies
1) “hobbit”
2) “titanic”
```

# Redis Types - streams

## XADD, XLEN, XRANGE

```
XADD pcp-chatroom * name lberk message “hello everybody!”
XADD pcp-chatroom * name jdoe message “hello lberk”
XADD pcp-chatroom * name lberk message “Anybody here going to the pcp conference in Tokyo?”
XADD pcp-chatroom * name jdoe message “yes”
```

```
XLEN pcp-chatroom
(integer) 4
```

```
XRANGE pcp-chatroom - +
(output)
```

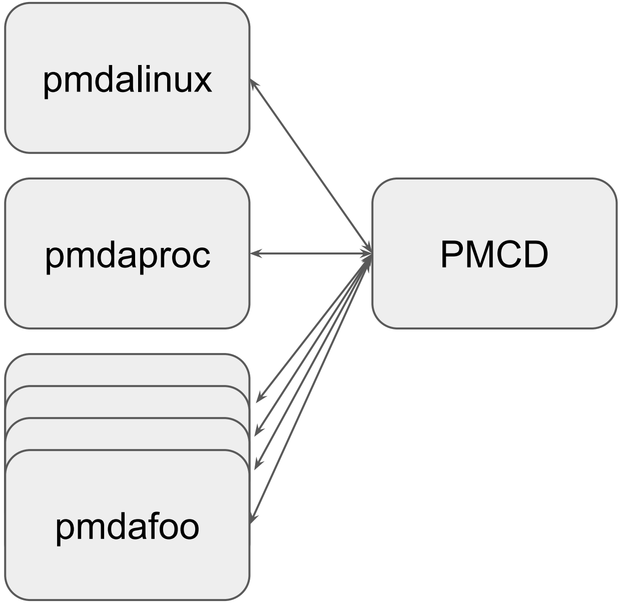
```
XREAD BLOCK 0 STREAMS pcp-chatroom $
```

# Redis Conclusions

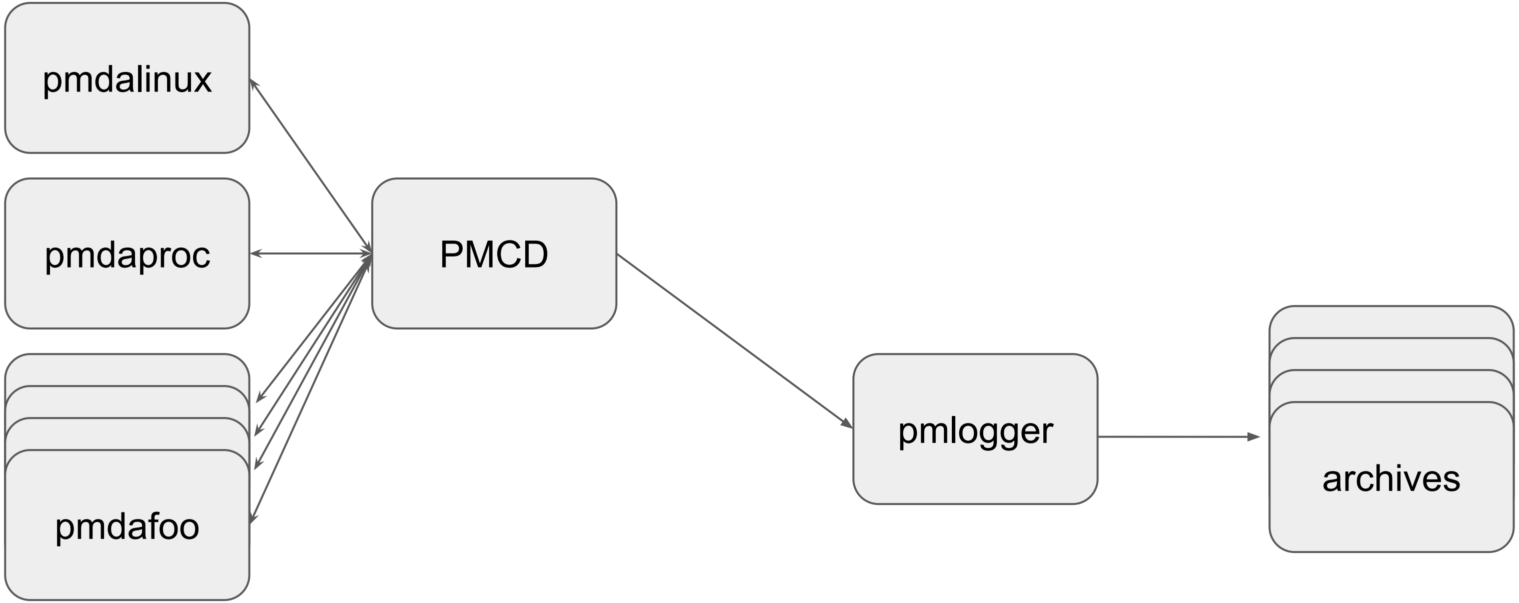
Redis Seems to Support Our Needs

- Working Schema
    - https://github.com/performancecopilot/pcp/blob/master/src/pmseries/schema.svg
- Established community around the database, avoiding NIH
- Very performant (will show in demo)
- Streams (suitable for timeseries) shipping in Redis-5
    - Redis currently shipping in majority of linux distros
    - Developing & testing on Redis 5
    - Will standardize on Redis 5 once release

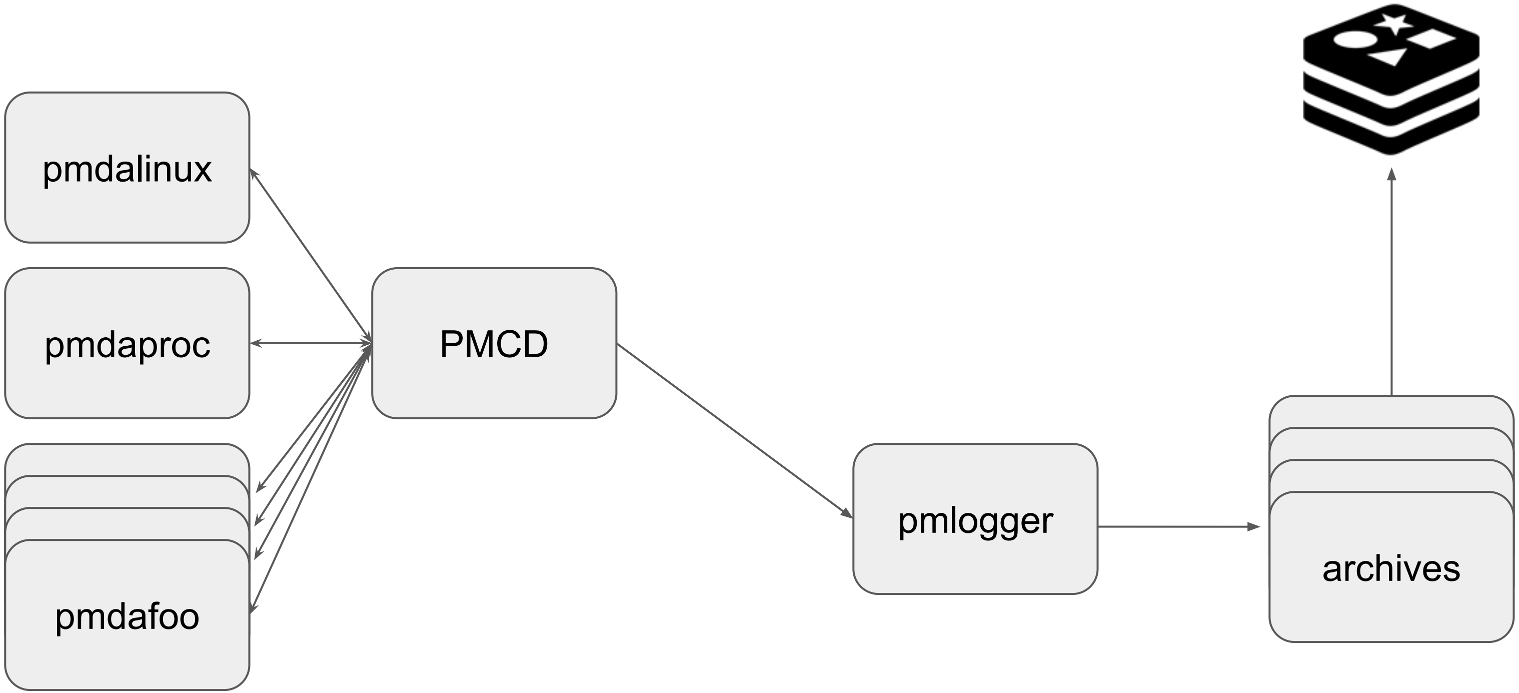
# Architecture



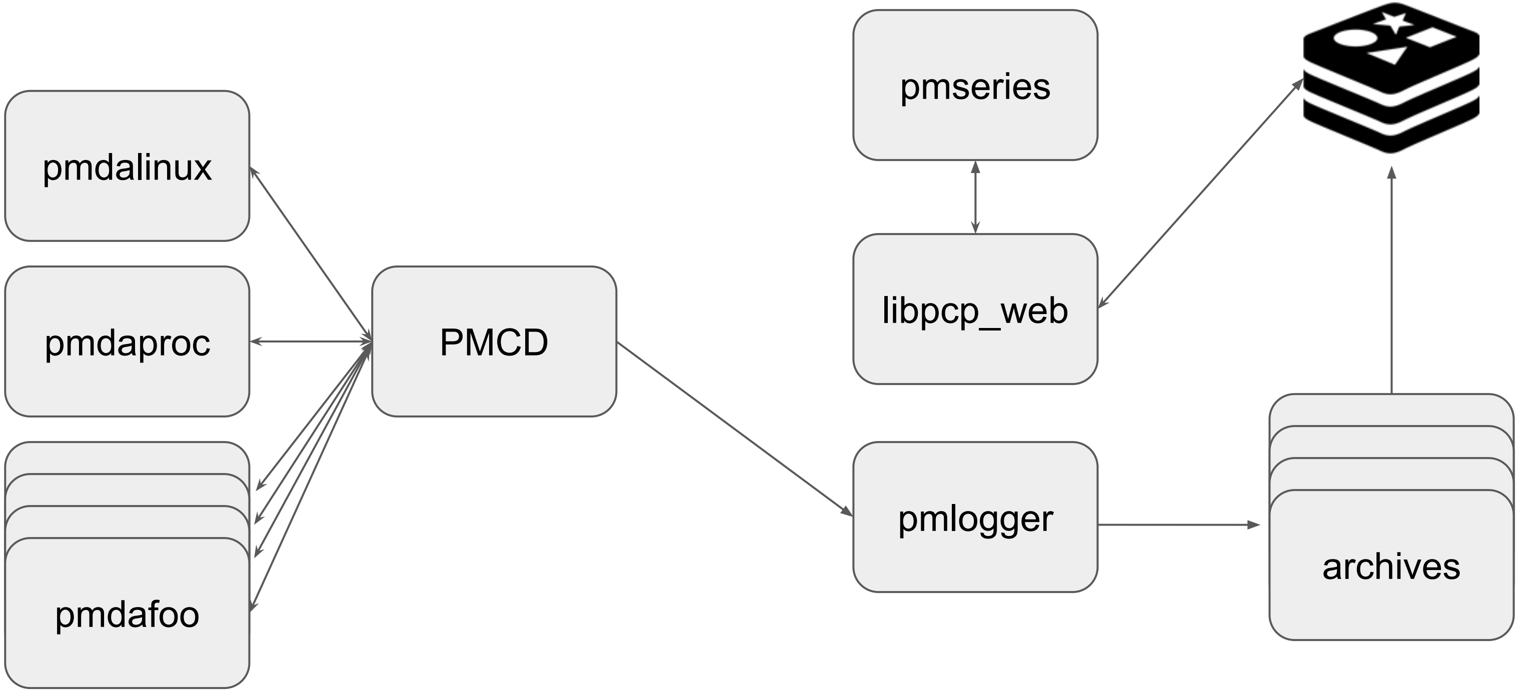
# Architecture



# Architecture



# Architecture



# pmseries

Query tool that will load metrics from archives to redis nodes and query

- Redis functionality primarily built into libpcp_web.so
- Currently a synchronous model for PoC, currently moving to async
- Able to query metrics values, descriptions, contexts, instances, labels
    - Automatically add hostname to each metric loaded

# pmseries syntax

```
$ pmseries [options] METRIC {label:qualifiers} [range]
```

Labels:

| | |
|---|---|
| - source.path | { source.path: “/path/to/archives” } |
| - instance.name | { instance.name: “1 minute” } |
| | { instance.name: “1*” } |
| - agent | { agent: “proc” \|\| agent: “xfs” } |
| - hostname | { hostname == “toium” && hostname != foobar } |

# pmseries syntax

```
$ pmseries [options] METRIC {label:qualifiers} [range]
```

Range:

- count [ count: 30 ]
- intervals [ interval: 10 ]
- time [1h]<br>
  [30m]<br>
  [1m]

# Demo

# Improvements

- Speed up loading archives (lots of optimizations to be made)
- Continue switch to async
- Complete clustering support
    - Add config loading
    - Fix globbing support
- Tail archives directly into redis (See Mark’s talk later today)
- Documentation Documentation Documentation
- Open the door to things like grafana plugins
- Explore redis plugins (perhaps redis-ml?)
- Load more labels by default
- Fill out the language features to our spec

# Questions?