Programming the Performance Co-Pilot toolkit

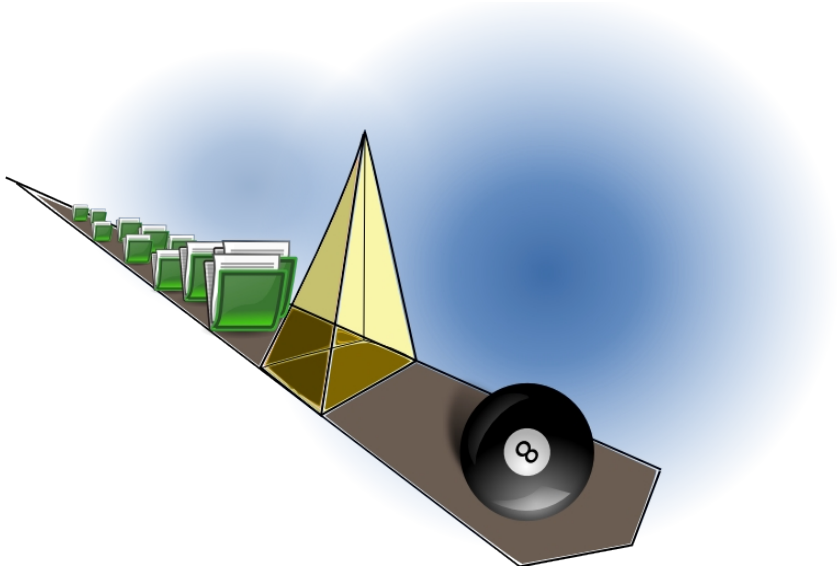
Nathan Scott
Performance Tools, Red Hat
August 2014

Outline

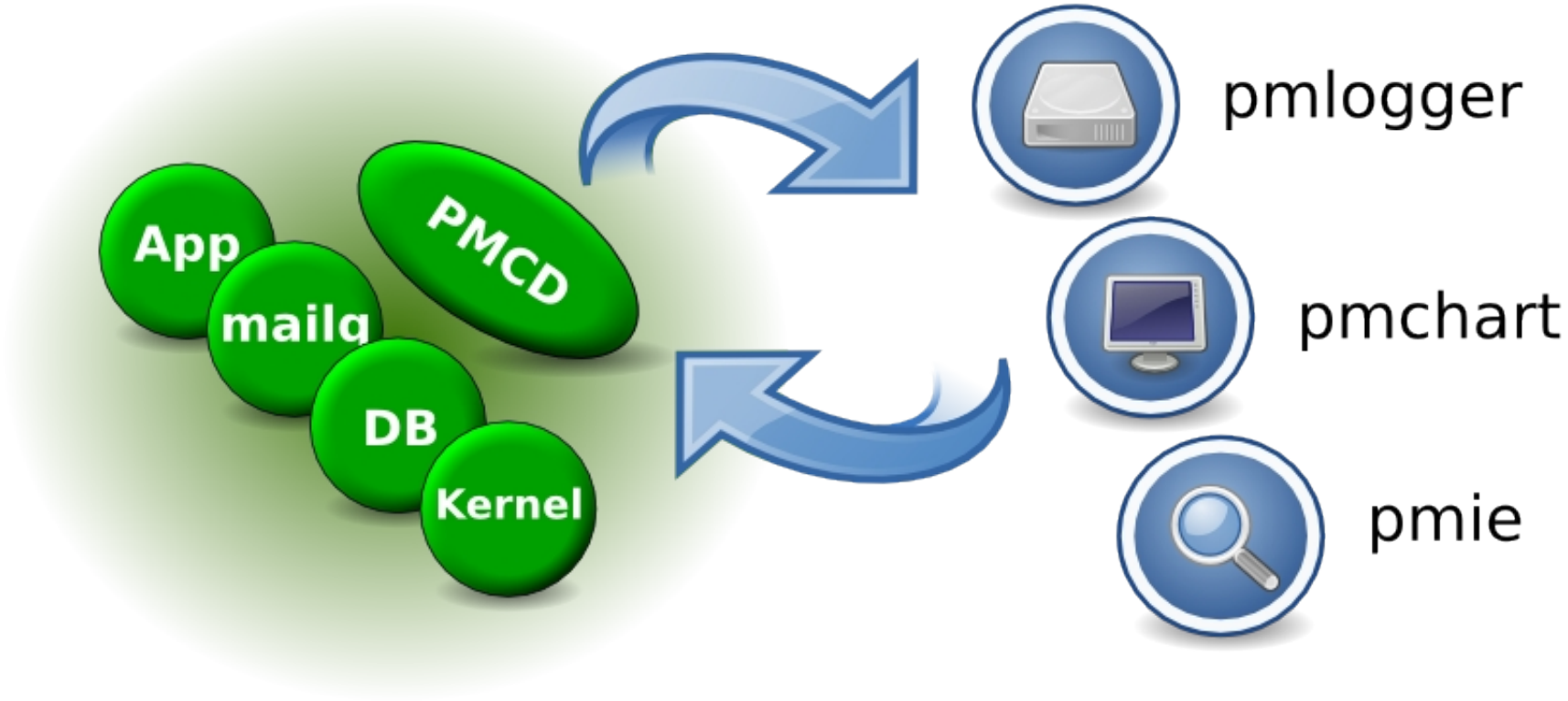
- Performance Co-Pilot (PCP)
  - Overview
  - Exploring PCP
- Extending with Python
  - Collectors
  - Monitors

Overview

- What is PCP?
  - Open source **toolkit**
  - System-level analysis
  - Live and historical
  - Extensible (monitors, collectors)
  - Distributed



Architecture



Metrics

- **pminfo** --desc -tT --fetch ***disk.dev.read***

disk.dev.read [*per-disk read operations*]

Data Type: *32-bit unsigned int* InDom: *60.1*

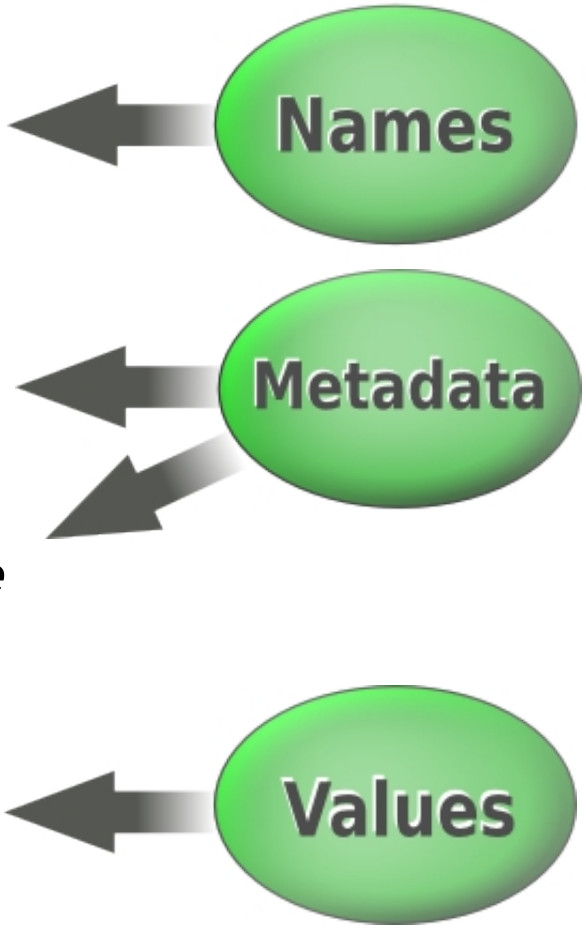
Semantics: *counter* Units: *count*

Help: *Cumulative count of disk reads since boot time*

Values:

inst [0 or "***sda***"] value ***3382299***

inst [1 or "***sdb***"] value ***178421***

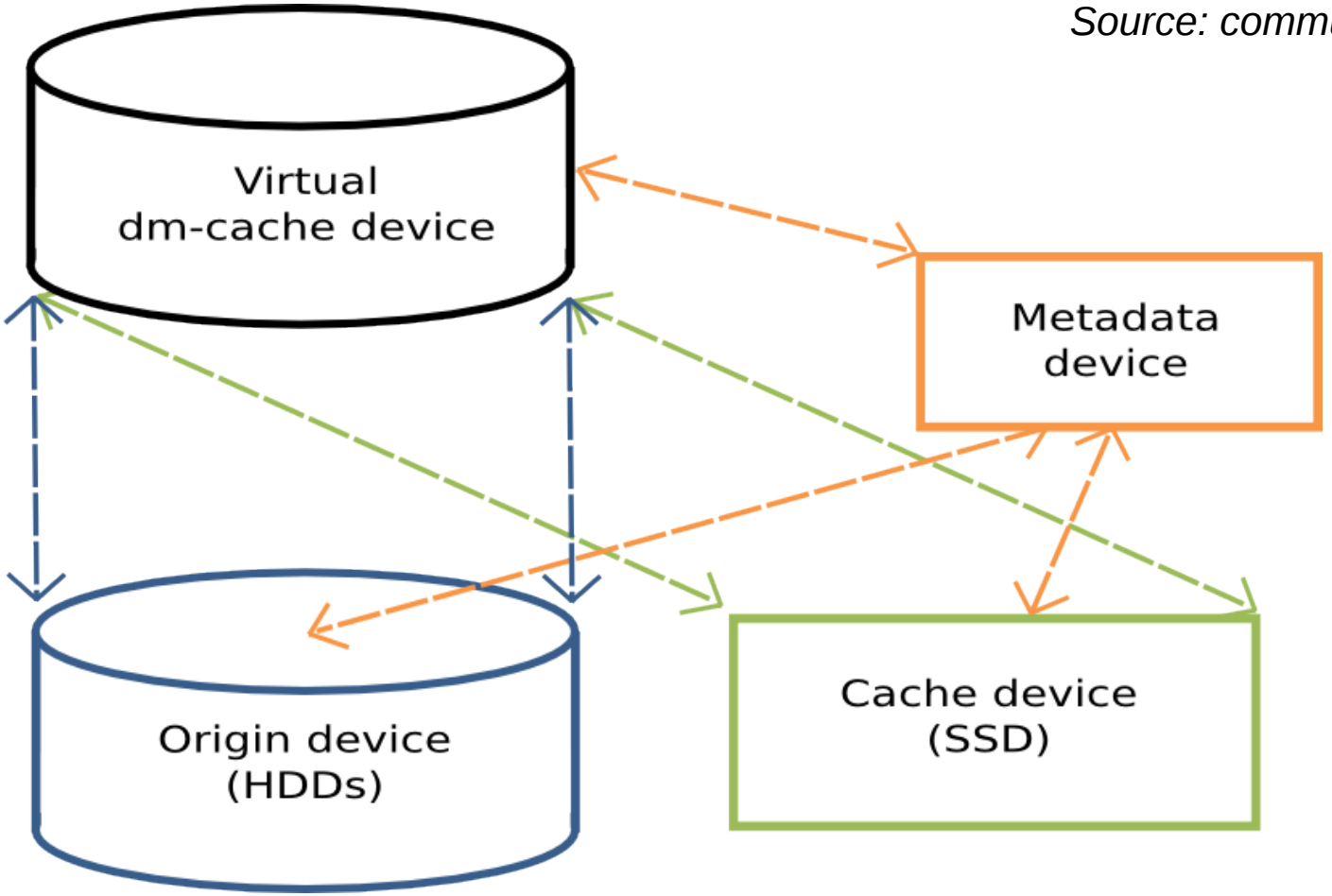



Extending the toolkit,
by example

Device Mapper Cache target

“Linux Block Caching Choices [...]” (pdf)

Source: community.dell.com



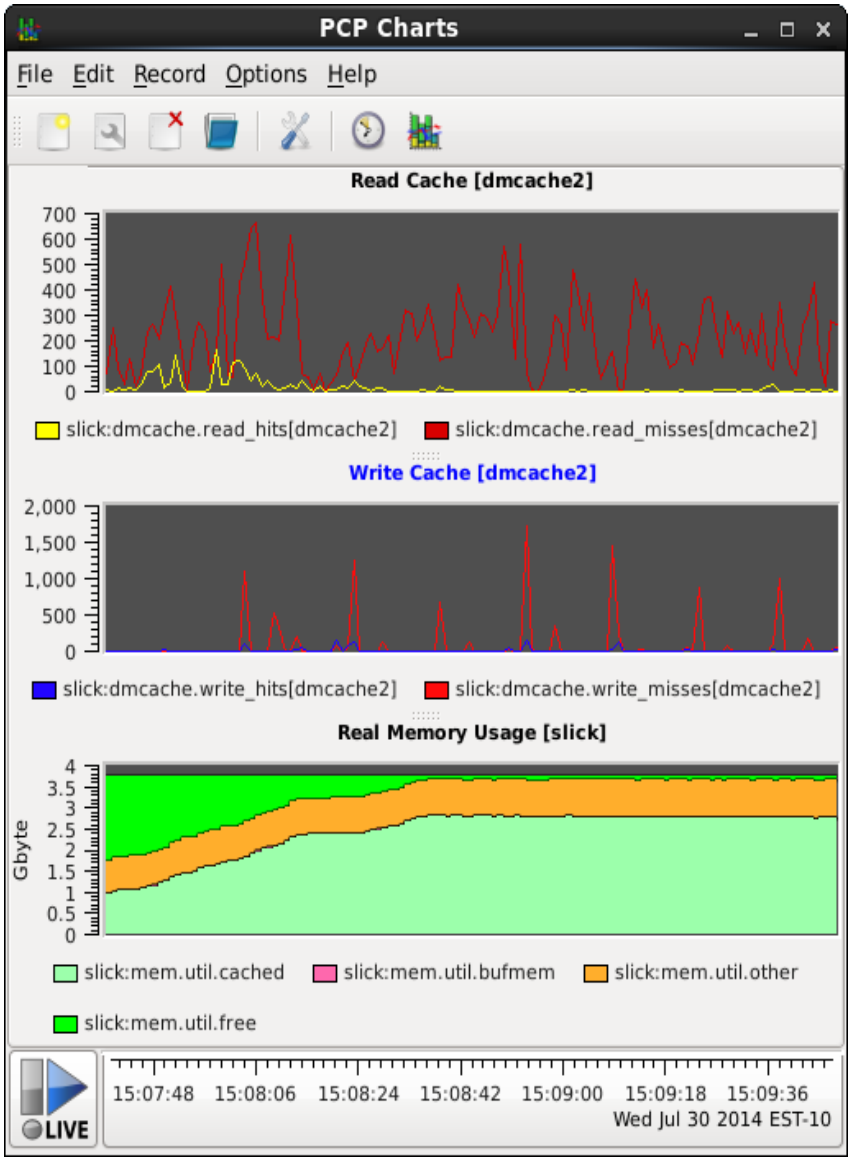
Collector extensions in Python

- Goal: new dmcache metrics
  - For recording, charting, console tools

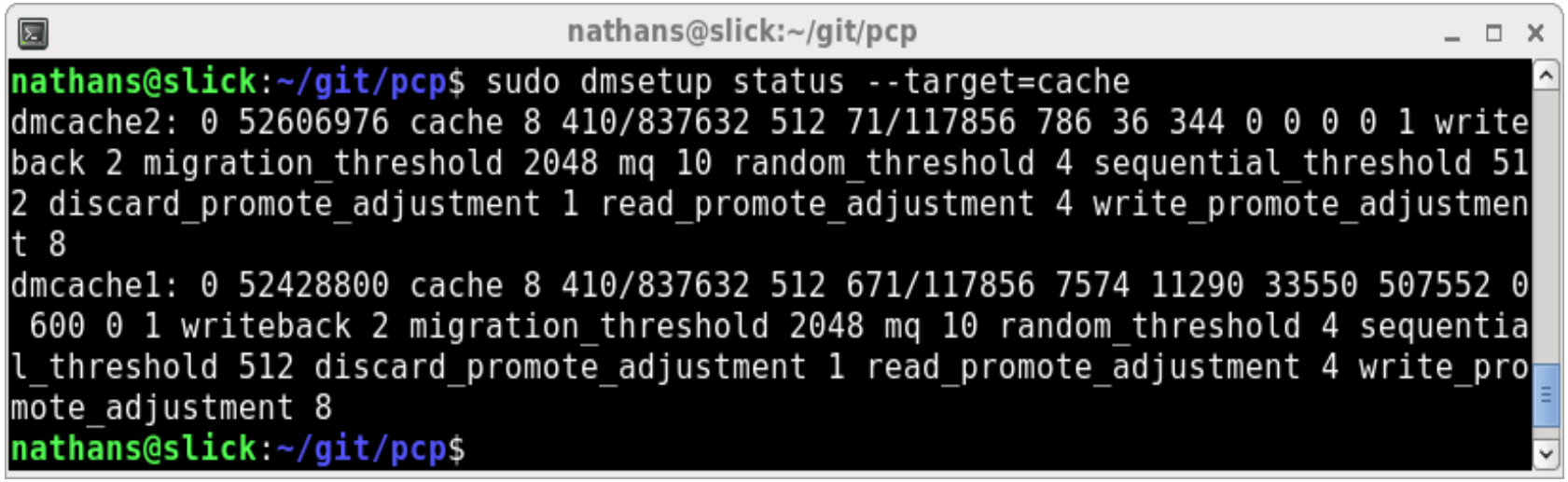
```
nathans@smash:~/git/pcp$ pmval -s3 --host=slick dmcache.dirty

metric:    dmcache.dirty
host:      slick
semantics: instantaneous value
units:     Kbyte
samples:   3
interval:  1.00 sec

            dmcache2             dmcache1
                 512                    0
                   0                  768
                1792                  256
nathans@smash:~/git/pcp$
```



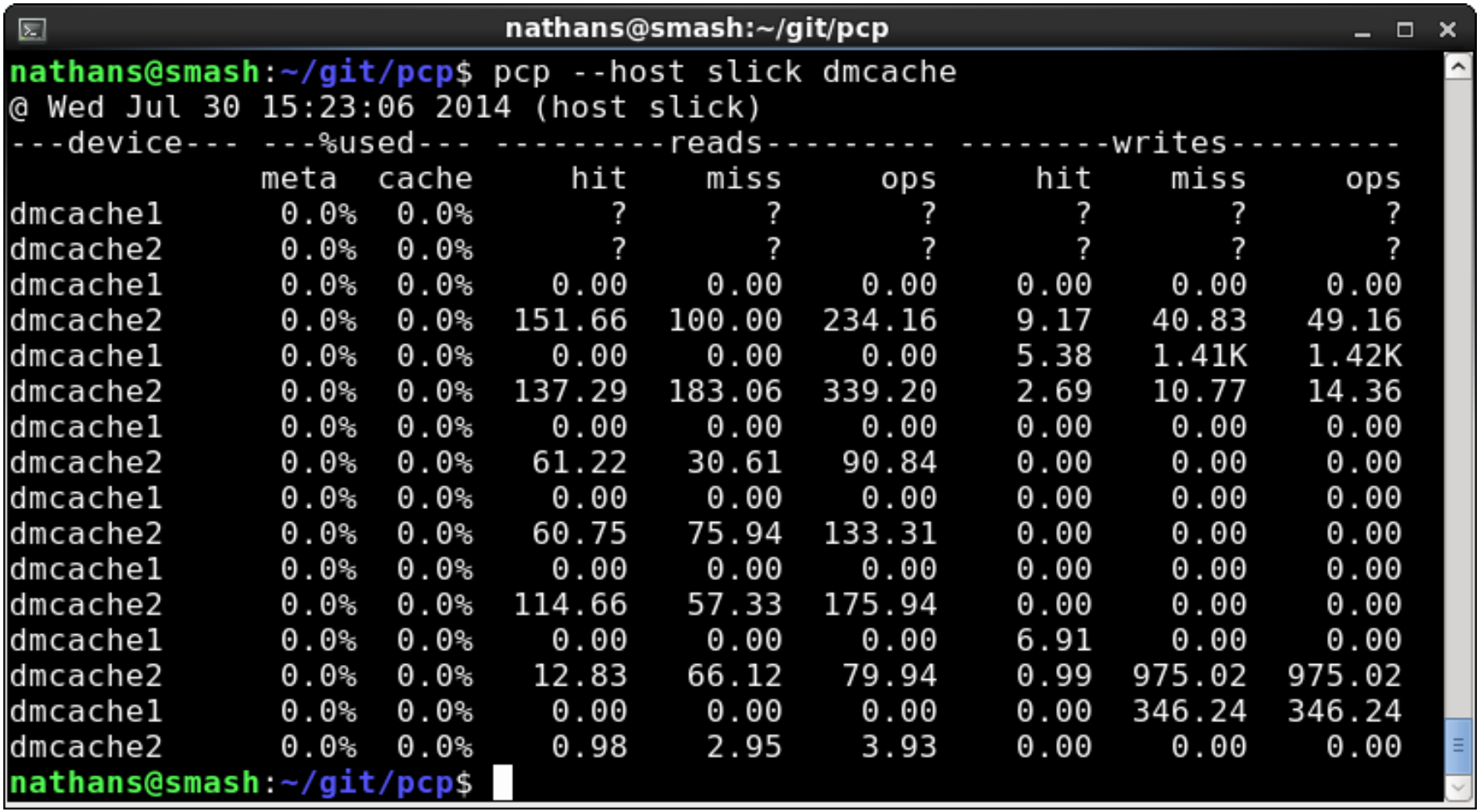
Collector extensions in Python



- Domains, Metrics, Instances
- *import pcp.pmda*
- Create a PMDA subclass
- *add_metric* (...names, IDs, units, etc...)
- Fetch callback – turns metric IDs into values

Monitor extensions in Python

- Goal: a simple, custom dmcache stats tool...



Monitor extensions in Python

- Report metrics (dmcache.* and disk.dm.*)
- *import pcp.pmcc*
- Create a *MetricGroupManager* object
- Feed it metric names + a *MetricGroupPrinter* subclass

- *MetricGroupManager* does all the heavy lifting:
  - Sampling (fetch interval, count)
  - Source (archives, local/remote hosts)
  - Time window, timezones, options, etc.

Resources

- **git clone git.pcp.io/pcp dev**
  - *src/pmdas/dmcache (collecting code)*
  - *src/pcp/dmcache (monitoring code)*
- “PCP Programmer's Guide” book
- **dbpmda**(1)
- http://www.pcp.io
  - Source, downloads, books, FAQ, mailing lists
- http://developerblog.redhat.com
  - PCP articles, demo videos


Demo &
Questions