Applications and container monitoring with Performance Co-Pilot

Zabbix Summit 2018, Riga, Latvia

Andrew Nelson

Senior Consultant

Friday, October 5, 2018

Introduction

Senior Consultant with Red Hat in North America

8 Years with Red Hat

Zabbix user for over 15 years

Author of zbxapi, an API library for Ruby

Occasional juggler and weekend woodworker

redhat.

What is Performance Co-Pilot

“ A System Performance and Analysis Framework “

A framework for system-level performance analysis

For the collection, monitoring, and analysis of system metrics

Uses a distributed architecture

Provides a full API (C, Python, Perl)

Easily extensible and flexible

Often just referred to as PCP

It is recommended to use “Performance Co Pilot” when searching for information online

What is Performance Co Pilot

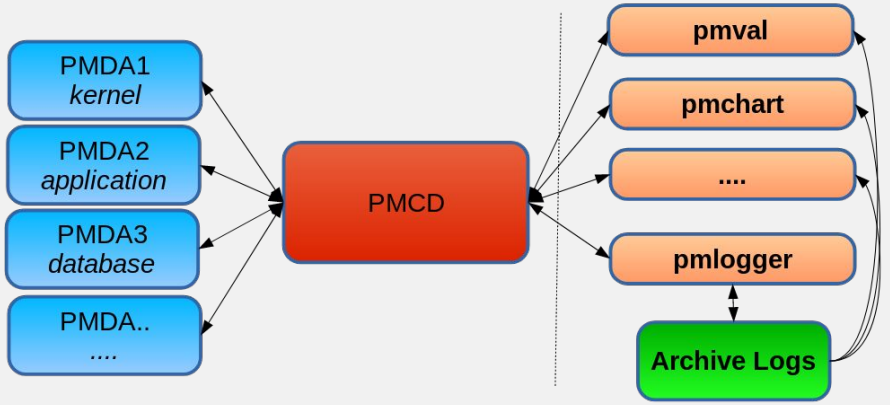
The core components

pmcd - Performance Metrics Collector Daemon

pmdas - Performance Metrics Domain Agents

pmns - Performance Metrics Name Space

Clients



What is Performance Co Pilot

A small sample of the PMDAs available

Activemq	KVM	nfs clients
Apache	Linux	Redis
Docker	Mysql	Samba
Ds389	Postgresql	SNMP
Elasticsearch	memory-mapped-values	Solaris
Freebsd	prometheus endpoints	VMware
Nginx	Perfevent	Windows

Red Hat products:
Satellite 6.4 (NEW!)
Gluster
Red Hat Directory Server (core to Red Hat IdM)

Using PCP

pmprobe

```
[root@elliot pmcd]# pmprobe libvirt.domstats.net
libvirt.domstats.net.name 3
libvirt.domstats.net.all.tx.drop 3
libvirt.domstats.net.all.tx.errs 3
libvirt.domstats.net.all.tx.pkts 3
libvirt.domstats.net.all.tx.bytes 3
libvirt.domstats.net.all.rx.drop 3
libvirt.domstats.net.all.rx.errs 3
libvirt.domstats.net.all.rx.pkts 3
libvirt.domstats.net.all.rx.bytes 3
libvirt.domstats.net.all.name 3
libvirt.domstats.net.count 3
...
```

Using PCP

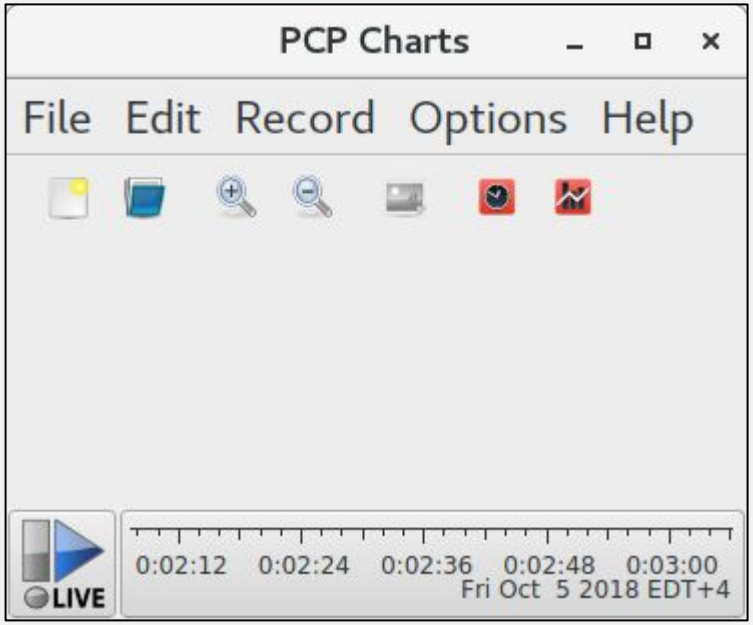
pminfo

```
[root@elliot pmcd]# pminfo -dtfT libvirt.domstats.net.all.tx.pkts

libvirt.domstats.net.all.tx.pkts [VM NICs, total tx pkts]
    Data Type: 64-bit unsigned int  InDom: 140.0 0x23000000
    Semantics: counter  Units: count
Help:
VM NICs, total tx pkts
    inst [0 or "923e087e-f0bb-49cd-b91f-8f42f9c07712"] value 12541241
    inst [1 or "d771652b-bf82-40bc-ae61-4d06398ed9dc"] value 899528
    inst [2 or "7e3b727b-1a82-43df-84d6-2bfbad174496"] value 3567731
```

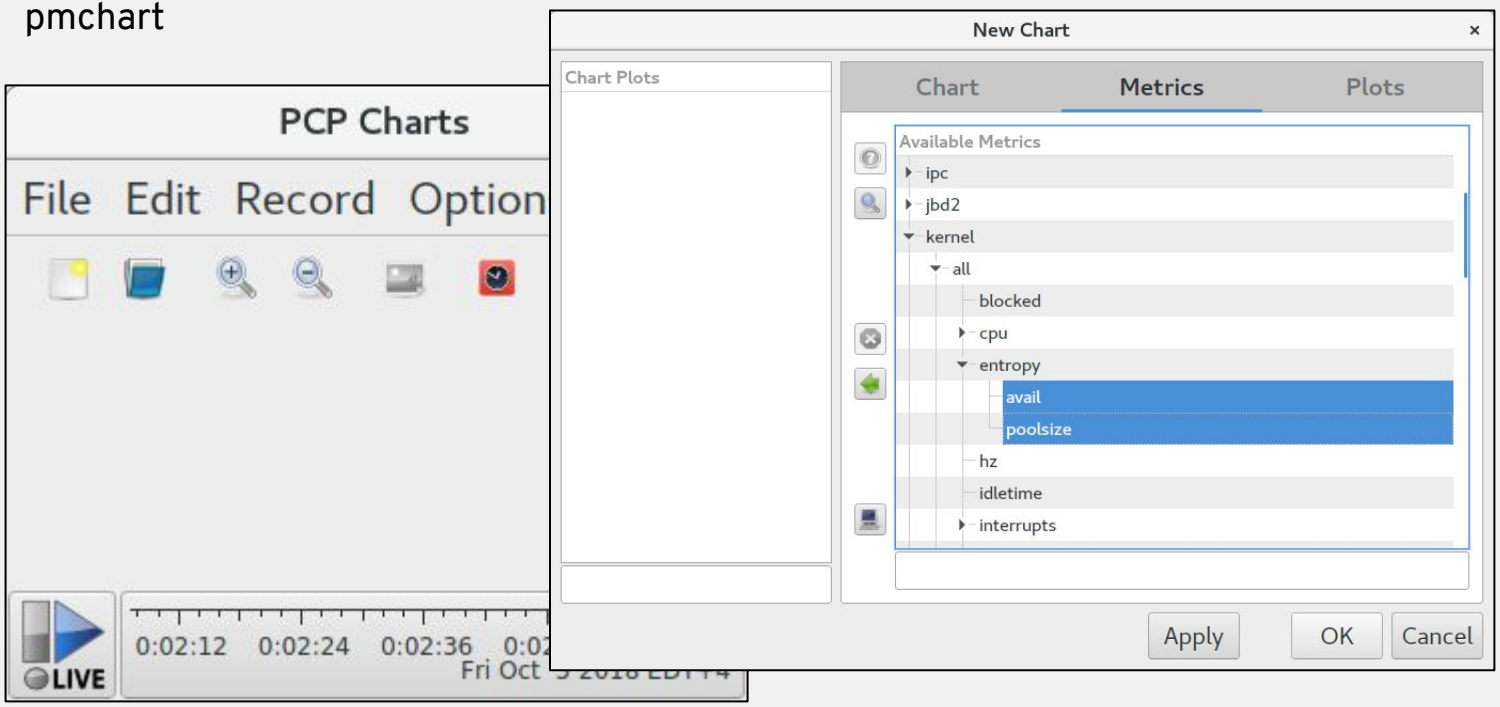
Using PCP

pmchart



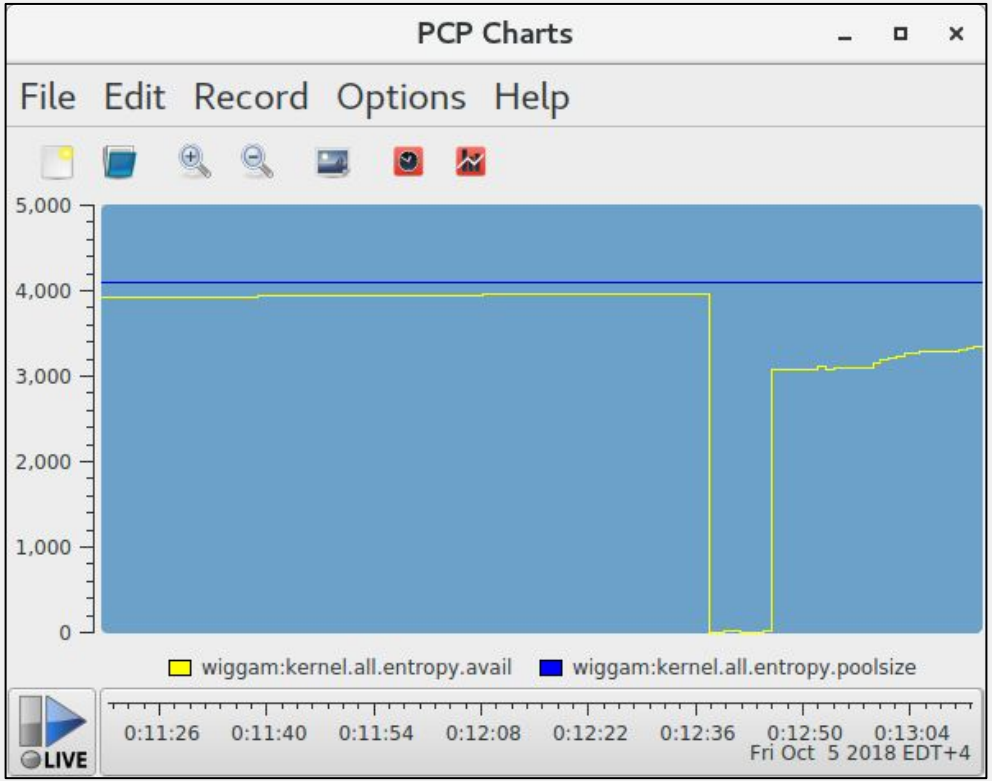
Using PCP

pmchart



Using PCP

pmchart



Using PCP

pmchart

NFS4 Reads/Writes

Network bytes In(yellow)/Out(Blue)

TCP Sockets:
Inuse (Yellow)
Time Wait (Red)
Allocated (Green)
Orphan (Purple)

TCP Socket memory

Various NFS Operations

Link Zabbix with Performance Co Pilot

- Install PCP (On RHEL7 enable the optional repository)

```
# yum install pcp pcp-export-zabbix-agent
```

- Enable PCP

```
# systemctl enable pmcd
# systemctl start pmcd
```

- Add PCP export module to your zabbix agent config file

```
LoadModule=zbxpcp.so
```

- TEST!

```
# zabbix_agentd -t pcp.kernel.all.sysfork
pcp.kernel.all.sysfork                          [u|17068591]
```

Let’s monitor something with PCP

Monitor the network for a guest from the hypervisor

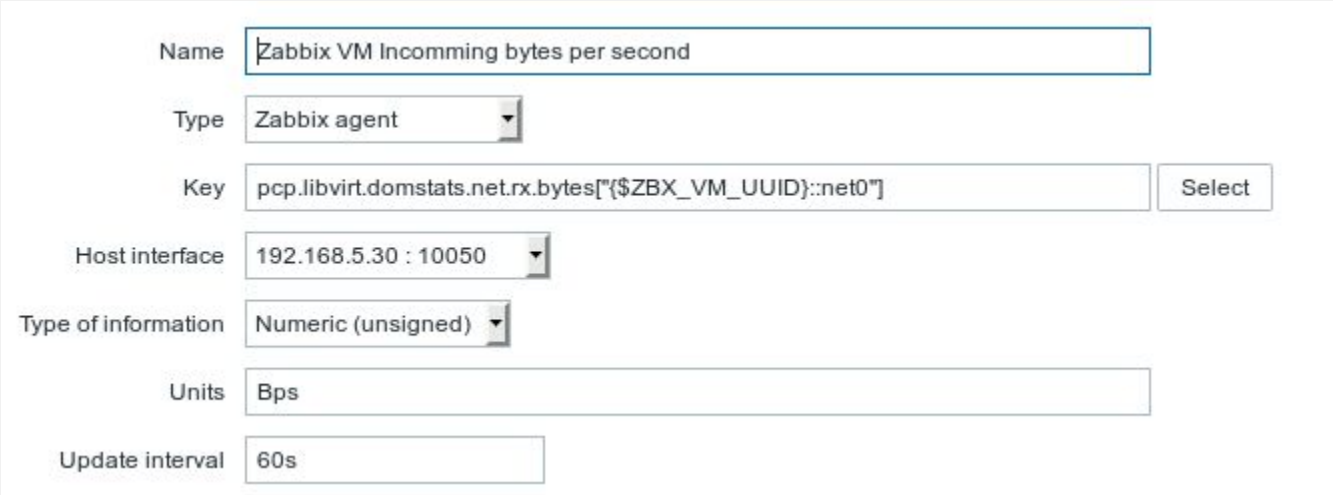
```
[root@elliot pmcd]# pminfo -F libvirt.domstats.net.rx.pkts

libvirt.domstats.net.rx.pkts
    inst [0 or "d771652b-bf82-40bc-ae61-4d06398ed9dc::net0"] value 1513252
    inst [1 or "923e087e-f0bb-49cd-b91f-8f42f9c07712::net0"] value 14469116
    inst [2 or "7e3b727b-1a82-43df-84d6-2bfbad174496::net0"] value 4542744
```

First we need to find figure out the UUIDs for the guests

Let’s monitor something with PCP

Monitor the network for a guest from the hypervisor



Don’t forget to add a pre-processor to convert the value to bytes to bytes per second.

Let’s monitor something with PCP

Monitor the network for a guest from the hypervisor

```
[root@elliot libvirt]# virsh list --uuid --name
d771652b-bf82-40bc-ae61-4d06398ed9dc Tower
923e087e-f0bb-49cd-b91f-8f42f9c07712 Zabbix
7e3b727b-1a82-43df-84d6-2bfbad174496 gitlab
```

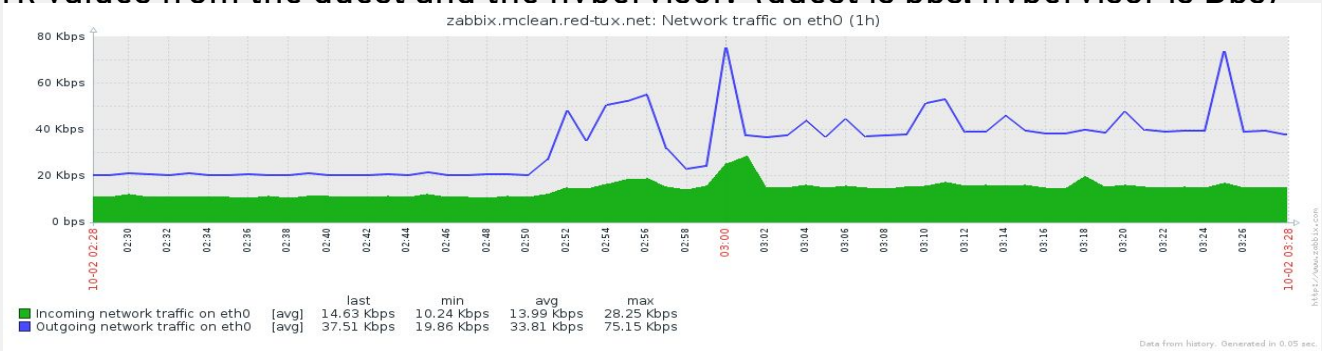
Alternatively…

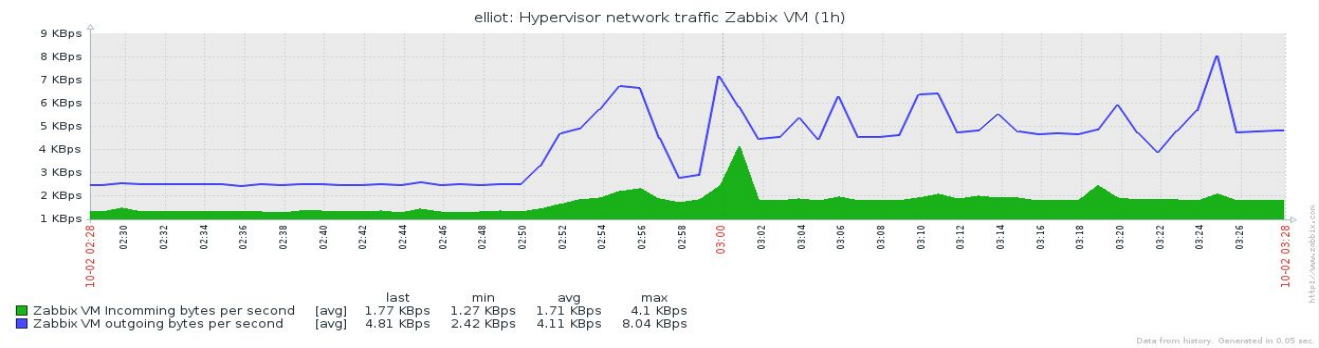
```
[root@elliot pmcd]# pminfo -F libvirt.dominfo.name

libvirt.dominfo.name
    inst [0 or "923e087e-f0bb-49cd-b91f-8f42f9c07712"] value "Zabbix"
    inst [1 or "d771652b-bf82-40bc-ae61-4d06398ed9dc"] value "Tower"
    inst [2 or "7e3b727b-1a82-43df-84d6-2bfbad174496"] value "gitlab"
```

Let’s monitor something with PCP

Network values from the guest and the hypervisor. (guest is bps, hypervisor is Bps)





redhat.

Creating derived checks with PCP

Using pcp for calculated items

“Derived Metrics” akin to “Calculated items” in zabbix

Configured in: /var/lib/pcp/config/derived

Zabbix specific ones in: /etc/zabbix/zbxpcp-derived-metrics.conf

Derived items have some requirements:

- All data must be in the same instance domain
- All data must have the same metadata type

Creating derived checks with PCP

Using pcp for calculated items

Some items are already defined as derived metrics:

```
disk.[dev|dm|md].await        disk.[dev|dm|md].avg_rqsz
disk.[dev|dm|md].r_await      disk.[dev|dm|md].r_avg_rqsz
disk.[dev|dm|md].w_await      disk.[dev|dm|md].w_avg_rqsz
disk.[dev|dm|md].avg_qlen
```

```
disk.md.avg_qlen = rate(disk.md.read_rawactive) + rate(disk.md.write_rawactive)
```

The rate function is akin to Zabbix's "change over time"

eBPF, BCC and PCP oh my!

Tracing the kernel.

Extended Berkley Packet Filter allows for Linux Kernel tracing.

BCC is the BPF Compiler Collection

- Provides a more straight forward interface to compiling eBPF traces

```
root@lenny-mclean-red-tux-net bcc]# pminfo -f bcc.runq.latency

[bcc.runq.latency
    inst [0 or "0-1"] value 622
    inst [1 or "2-3"] value 5696
    inst [2 or "4-7"] value 9961
    inst [3 or "8-15"] value 14202
    inst [4 or "16-31"] value 2547
...
```

eBPF, BCC and PCP oh my!

```
[root@lenny-mclean-red-tux-net derived]# !549
pminfo -f bcc.runq.latency

bcc.runq.latency
    inst [0 or "0-1"] value 26682
    inst [1 or "2-3"] value 182745
    inst [2 or "4-7"] value 224928
    inst [3 or "8-15"] value 445640
    inst [4 or "16-31"] value 87867
    inst [5 or "32-63"] value 205554
  ...
```

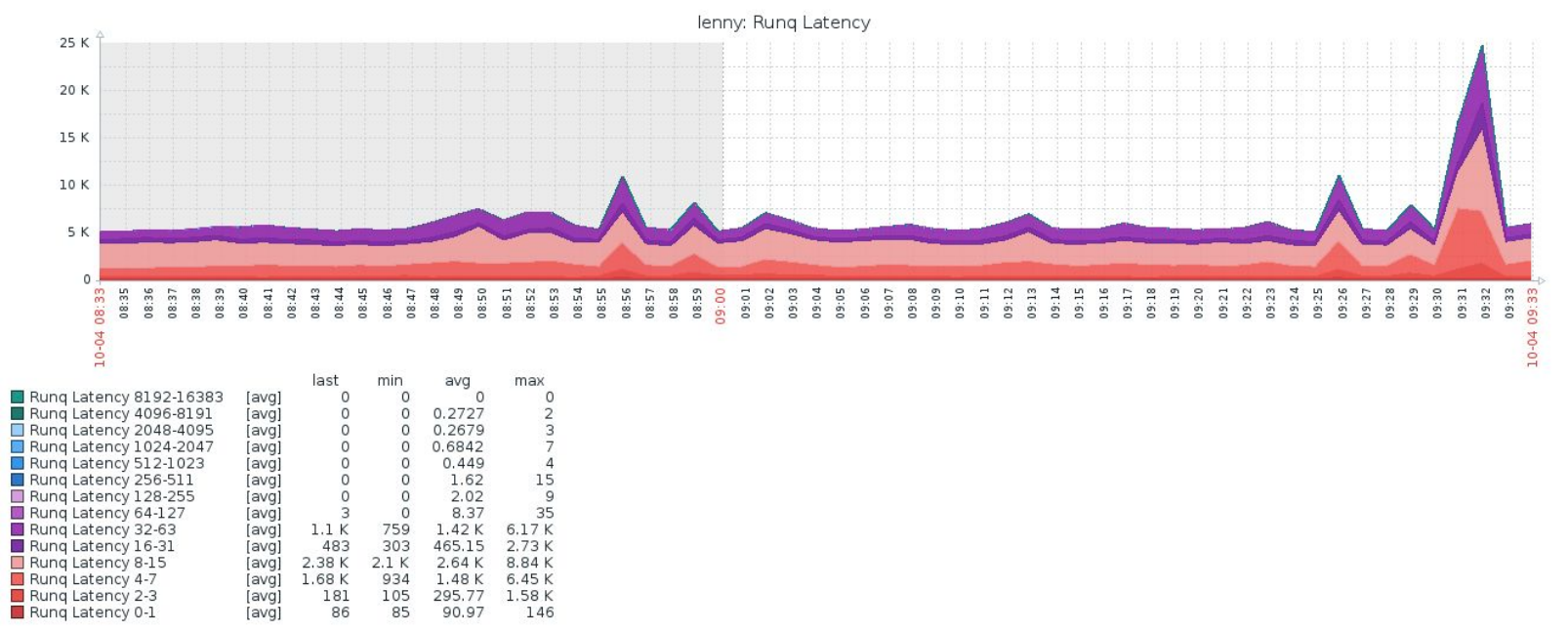
eBPF, BCC and PCP oh my!

Let's link it to Zabbix

Name ▲	Triggers	Key	Interval	History	Trends	Type	Applications
Runq Latency 2-3		pcp.bcc.runq.latency[2-3]	60	90d	365d	Zabbix agent	pcp
Runq Latency 0-1		pcp.bcc.runq.latency[0-1]	60	90d	365d	Zabbix agent	pcp
Runq Latency 4-7		pcp.bcc.runq.latency[4-7]	60	90d	365d	Zabbix agent	pcp
Runq Latency 8-15		pcp.bcc.runq.latency[8-15]	60	90d	365d	Zabbix agent	pcp
Runq Latency 512-1023		pcp.bcc.runq.latency[512-1023]	60	90d	365d	Zabbix agent	pcp
Runq Latency 4096-8191		pcp.bcc.runq.latency[4096-8191]	60	90d	365d	Zabbix agent	pcp
Runq Latency 2048-4095		pcp.bcc.runq.latency[2048-4095]	60	90d	365d	Zabbix agent	pcp
Runq Latency 1024-2047		pcp.bcc.runq.latency[1024-2047]	60	90d	365d	Zabbix agent	pcp
Runq Latency 32-63		pcp.bcc.runq.latency[32-63]	60	90d	365d	Zabbix agent	pcp
Runq Latency 16-31		pcp.bcc.runq.latency[16-31]	60	90d	365d	Zabbix agent	pcp
Runq Latency 64-127		pcp.bcc.runq.latency[64-127]	60	90d	365d	Zabbix agent	pcp
Runq Latency 128-255		pcp.bcc.runq.latency[128-255]	60	90d	365d	Zabbix agent	pcp
Runq Latency 256-511		pcp.bcc.runq.latency[256-511]	60	90d	365d	Zabbix agent	pcp
Runq Latency 8192-16383		pcp.bcc.runq.latency[8192-16383]	60	90d	365d	Zabbix agent	pcp

eBPF, BCC and PCP oh my!

Let’s link it to Zabbix



		last	min	avg	max
Runq Latency 8192-16383	[avg]	0	0	0	0
Runq Latency 4096-8191	[avg]	0	0	0.2727	2
Runq Latency 2048-4095	[avg]	0	0	0.2679	3
Runq Latency 1024-2047	[avg]	0	0	0.6842	7
Runq Latency 512-1023	[avg]	0	0	0.449	4
Runq Latency 256-511	[avg]	0	0	1.62	15
Runq Latency 128-255	[avg]	0	0	2.02	9
Runq Latency 64-127	[avg]	3	0	8.37	35
Runq Latency 32-63	[avg]	1.1 K	759	1.42 K	6.17 K
Runq Latency 16-31	[avg]	483	303	465.15	2.73 K
Runq Latency 8-15	[avg]	2.38 K	2.1 K	2.64 K	8.84 K
Runq Latency 4-7	[avg]	1.68 K	934	1.48 K	6.45 K
Runq Latency 2-3	[avg]	181	105	295.77	1.58 K
Runq Latency 0-1	[avg]	86	85	90.97	146

Resources

Main website

https://pcp.io/

Index of Performance Co-Pilot (PCP) articles, solutions, tutorials and white papers:

https://access.redhat.com/articles/1145953

THANK YOU


plus.google.com/+RedHat


facebook.com/redhatinc


linkedin.com/company/red-hat


twitter.com/RedHat


youtube.com/user/RedHatVideos