



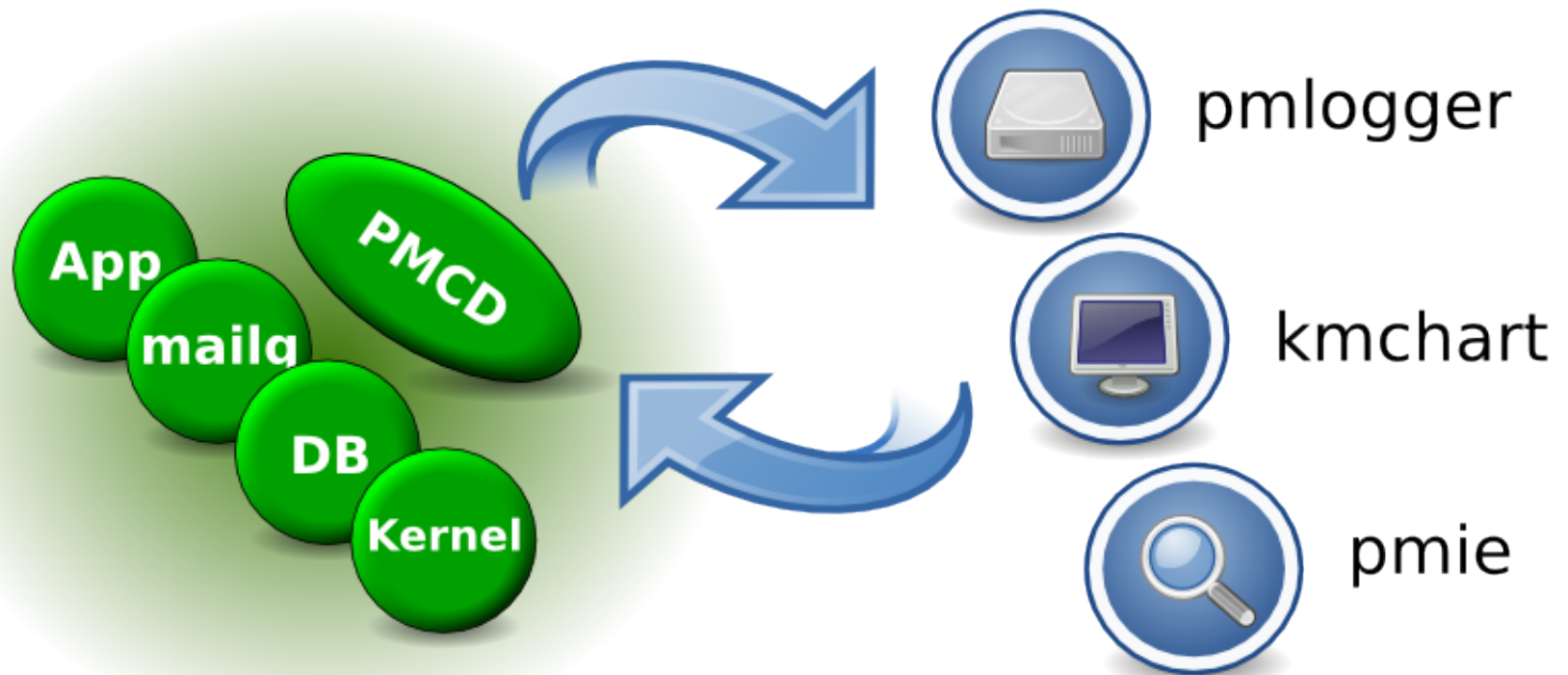
# System Level Performance Management with PCP

*Nathan Scott*  
*<nathans@debian.org>*

# Overview

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

# Architecture



# Data Model

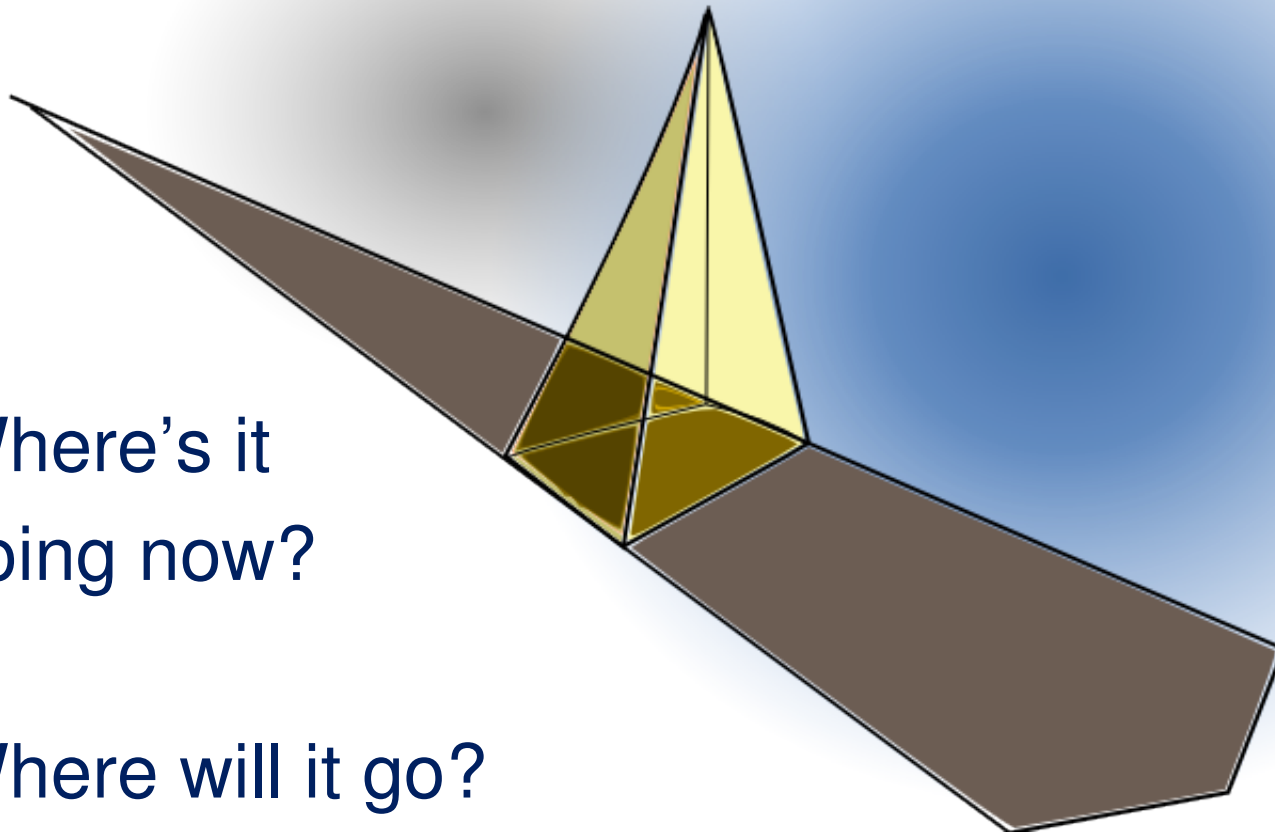
- Metrics come from one **source** (host / archive)
- Source can be queried at any interval by any monitor tool
- Hierarchical metric names  
e.g. disk.dev.read and aconex.response\_time.avg
- Metrics are singular or set-valued (“instance domain”)
- Metadata associated with every metric
  - Data type (int32, uint64, double, ...)
  - Data semantics (units, scale, ...)
  - Instance domain

# Performance Timeline

- Where *does* the time go?

- Where's it going now?

- Where will it go?

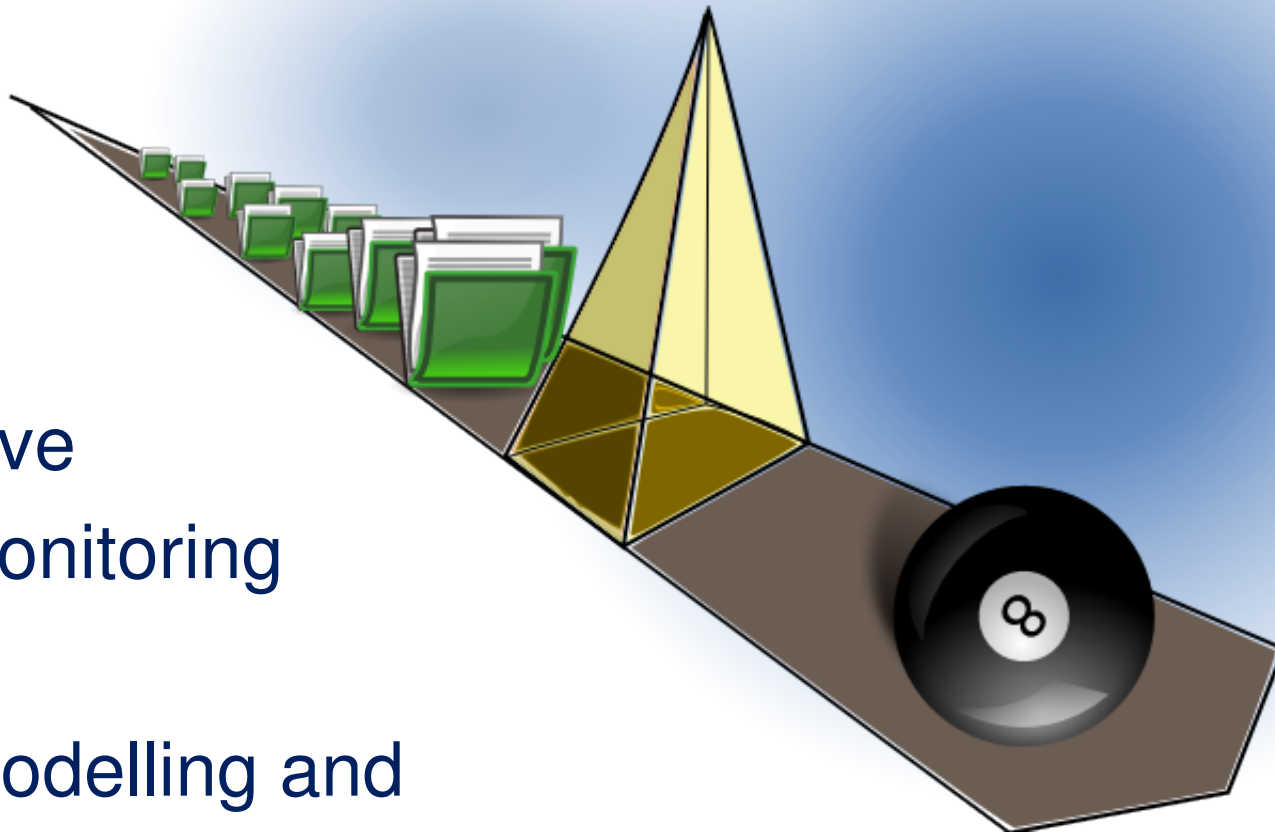


# Performance Timeline - PCP Toolkit

- Archives

- Live monitoring

- Modelling and statistical prediction



# Performance Timeline - PCP Toolkit

- Yesterday, last week, last month, ...
- All starts with pmlogger
  - Arbitrary metrics, intervals
  - One instance produces one PCP archive for one host
  - An archive consists of 3 files
    - Metadata, temporal index, data volume(s)
- pmlogger\_daily, pmlogger\_check
  - Ensure the data keeps flowing
- pmlogsummary, pmwtf, pmdumptext
- pmlogextract, pmlogreduce

# Performance Timeline - PCP Toolkit

- Graphical tools – kmchart, kmtime
  - Strip charts – align data from different subsystems on a single time axis
  - Time controls
    - VCR paradigm
    - Multiple tools can share

• [ Demo ]

Tempdb growth (1)



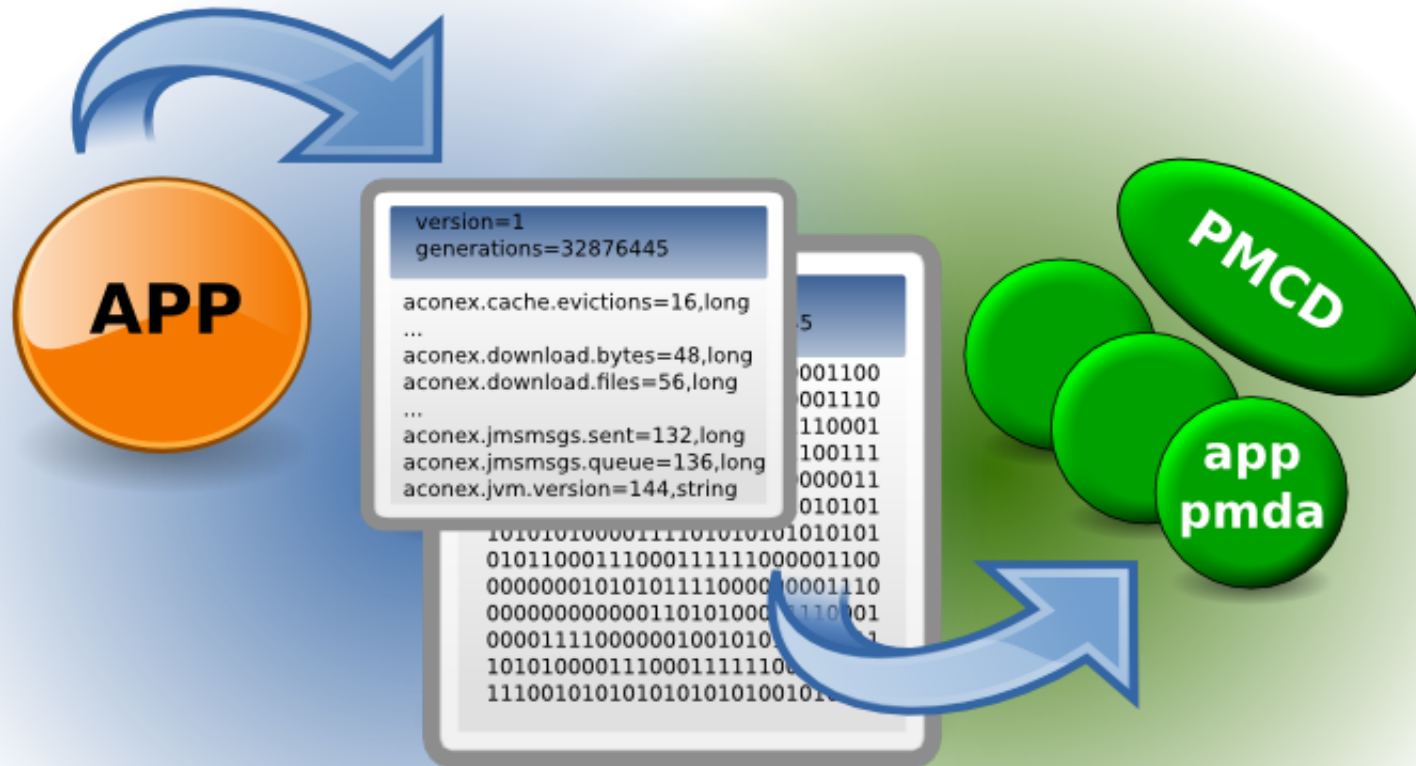
# Performance Timeline - PCP Toolkit

- Inference Engine - pmie
    - Evaluates arithmetic, logical and rule expressions at arbitrary frequencies
    - Scan historical data looking for given conditions
  - Archive mode uses:
    - Data reduction
    - Alarm verification
- [ Demo ]
- Tempdb growth (2)

# Performance Timeline - PCP Toolkit

- What's happening right now?
- Hardware, kernel, services, databases, ... application PMDAs.
  - PCP toolkit provides many PMDAs and APIs for customisation
- Important to be able to match user-perceived response time back to system activity
  - [ Demo ]  
Kernel, pmcd, shping PMDAs  
pmchart, pmval - monitor tools

# Custom Instrumentation (Applications)



# Applied PCP

- Establish performance baselines
  - Setup constant logging
  - Automate detection of known issues (pmie)
- Monitor end-user perceived response time
  - Custom collectors
  - Generic collectors – shping, dbping
- Understand where that time is spent
  - Distributed systems, distributed queues
  - Monitor for transient / unexpected events

# Tricks and Tips

- Have a model of performance in your head, and evaluate new information against it

## Subsystem    Measured Time    Scaled Time

CPU cycle	0.31 nanosec	0.31 sec
L1 cache	0.31 nanosec	0.31 sec
L2 cache	1.25 nanosec	1.25 sec
Memory bus	2 nanosec	2 sec
DRAM chip	60 nanosec	1 min
Disk seek	3.5 millisec	1.35 months
NFS3 read	32 millisec	1.01 years
RDBMS update	0.5 sec	15.85 years
Tape access	5 sec	1.59 century

*Source:*

*Analysing Computer System Performance with Perl PDQ, NJ Gunther (2005)*

- `units(1)`

# Tricks and Tips

- Use the “Scientific Method”
  - Postulate, test hypotheses
  - Record results, iterate
- Find good (user) response time metrics
  - Drive analysis based on issues they detect
  - CPU and disk utilisation (time based metrics)

# Tricks and Tips

- Regularly apply “Little’s Law” to all data... generic (queueing theory) form:

$$Q = \lambda R$$

- Length = Arrival Rate x Response Time
- e.g. 10 MB = 2 MB/sec x 5 sec
- Utilisation = Arrival Rate x Service Time
- e.g. 20% = 0.2 = 100 msec/sec x 2 sec

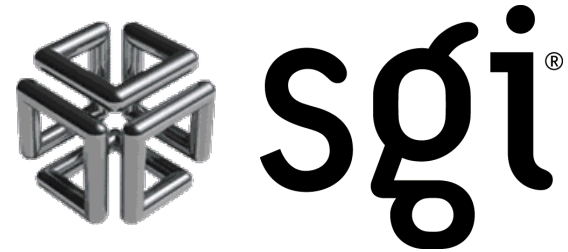
# Present and future

- Recent past
  - Development moved to git
  - kmchart developed independently to PCP
  - Included in Debian and Ubuntu (SuSE for ages)
- Moving toward PCP 3.0
  - Native Windows version, Perl APIs, new PMDAs
  - New archive temporal reduction tool
- Longer term
  - SGI releasing 3D visualisation code
  - Many more kmchart features planned
  - Capacity planning ... PCP meets R? PDQ?



# Finally...

- Major corporate sponsors of PCP development



- Thanks!
- <http://oss.sgi.com/projects/pcp/>
  - Docs, tutorials, git repos, mailing list, IRC channel
  - Binaries - Mac, Windows, RPMs

# Questions...?

