Designing a Robust Monitoring System

Scott McCarty
@fatherlinux
scott.mccarty@gmail.com

My Background

- 13 years with Linux, 11 years in **operations**
- Cut teeth on Statistics in Computer Science and Anthropology
- Web Operations
- Petit: Open source log analysis. In Fedora, Ubuntu, EPEL
- Written many Nagios Checks (Mostly Perl/Bash)
- Maintainer for Cacti/Nagios BGP Checks
- Maintainer for Cacti MySQL Stats

Background Knowledge

- Four Types of Data
  - Nominal: Names of servers
  - Ordinal: Active Users
  - Interval: Load Average, CPU (No Natural Zero)
  - Ratio: Open Socket, Pipes, Files
- FCAPS
  - Fault: up/down alerting
  - Configuration: CFEngine, Perl, Bash
  - Accounting: auditd, syslog, cloud
  - Performance: Data Acquisition, Graphs
  - Security: Confidentiality, Integrity, Availability

Background Knowledge

- Fault Monitoring
  - Fire and Police
  - Minority Report
- Data Acquisition
  - Criminal Record
  - Cameras
- What is Knowledge: For Philosophers
- Loose framework, not too constrained
- Petit – Log Analysis Program used for examples
  - Originally written in perl (logtool, lt)

Basics

- Differences between Data Acquisition and Fault Monitoring
  - Logging
  - Graphing
  - Alerting
- Cost of more robust systems
  - 8 to 5 Systems Administrators
  - 24 x 7 support center

Event Categories

- **Recorded**
  - Events which are worth recording
  - Operations people do not need to know about these events unless there is a problem somewhere else in the system
  - Letters and Numbers: Nominal to Ratio Data
  - Graphs and Logs
- **Action**
  - Events for which action must be taken, but can be during business hours
  - Prefer operations dashboard: Red Light/Green Light
- **Critical Action**
  - Alert Paging

Recorded

- Data Import/Exports
- Granular Job Tracking
- Load Average, CPU, Memory
- BGP Route Views Checks
- Trace Routes
- Configuration File Generation
- Backup Processes

Action

- Software Vulnerabilities
- SLA in Danger
- Tape Cleaning
- Captured Command Output: Catch All
- Fail Flags

Critical Action

- Network Down
- Server Down
- Service Down
- SLA Not Met
- Good Thresholds
  - Kernel Structures
  - Open Sockets
  - Open Pipes
  - Open Files
- Bad Thresholds
  - Not Load Average
  - Not CPU
  - Not Memory

System Design

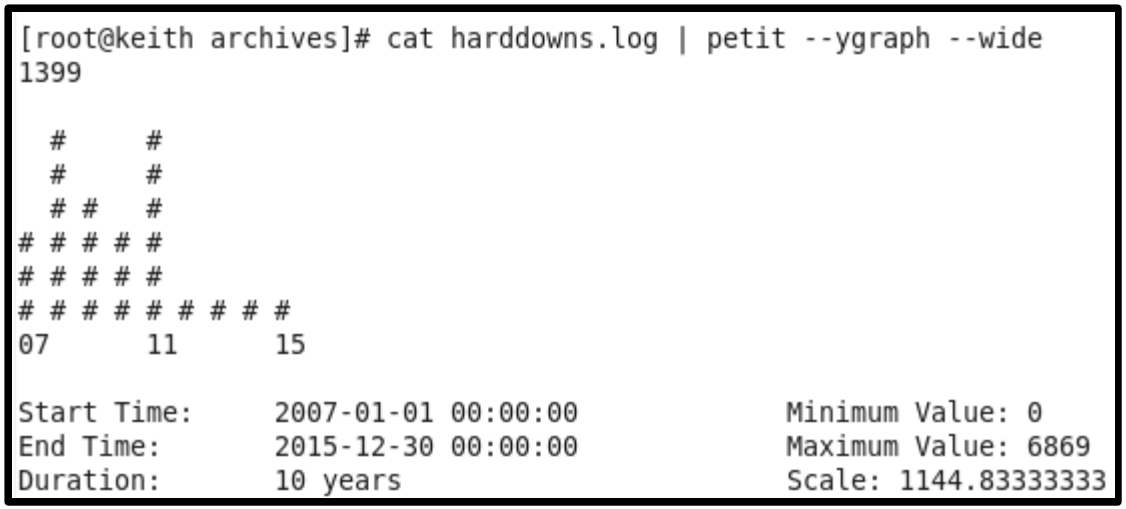
- Remember data acquisition and fault monitoring
- Always cross monitor
- Be very careful determining what is production
- Be realistic SLA
  - At hosting company, we moved from 1-2 mintute alerts to 7-8 and service did not change noticeably, but moral did.
- Be creative fault detection
  - Record traceroutes
  - Open sockets, pipes, files
  - BGP Route Views
  - Granular job tracking

Quantitative Data

- Developed theory: *increase to alert time would not impact return to service*
- Academic Honesty: Did not have quantitative data to suport or deny theory
- Amount of paging was too sporadic
  - Did not attempt calculation of standard deviation
  - Will be added to petit eventually :-)

Over Years

- End of 2007 changed alert SLA
- Do not have data for all of 2007



2007

- Notice scale is much larger
- There were more pages in 2007

```
[root@keith archives]# cat harddowns.log | grep " 2007 " | petit --mograph --wide
347
144
45

      #
      #
  #   #
  #   #
# # # #
# # # # # # # # # # # #
09          03        08

Start Time:     2007-09-01 00:00:00            Minimum Value: 0
End Time:       2008-08-01 00:00:00            Maximum Value: 1726
Duration:       12 months                      Scale: 287.666666667
```

2008

- Notice scale is much less than half
- There were less pages in 2008

```
[root@keith archives]# cat harddowns.log | grep " 2008 " | petit --mograph --wide

#           #
#         # #
# # # #   # #
# # # # # # # # #
# # # # # # # # # #   #
# # # # # # # # # # # #
01          07        12

Start Time:     2008-01-01 00:00:00             Minimum Value: 352
End Time:       2008-12-01 00:00:00             Maximum Value: 980
Duration:       12 months                       Scale: 104.666666667
```

2009

- Similar to 2008 (Normal???)
- Quiet in December

```
[root@keith archives]# cat harddowns.log | grep " 2009 " | petit --mograph --wide

# # # #
# # # # #
# # # # #   #
# # # # #   #   #
# # # # # # # # # #   #
# # # # # # # # # # # #
01          07        12

Start Time:     2009-01-01 00:00:00            Minimum Value: 124
End Time:       2009-12-02 00:00:00            Maximum Value: 646
Duration:       12 months                      Scale: 87.0
```

2010

- Completely Thrown Off
- Tried to normalize, didn't work

```
[root@keith archives]# cat harddowns.log | grep " 2010 " | petit --mograph --wide
195
57

      #
      #
      #
      #     #
# # # # # # # #   # # #
# # # # # # # # # # # #
01          07        12

Start Time:     2010-01-01 00:00:00             Minimum Value: 0
End Time:       2010-12-02 00:00:00             Maximum Value: 1125
Duration:       12 months                       Scale: 187.5
```

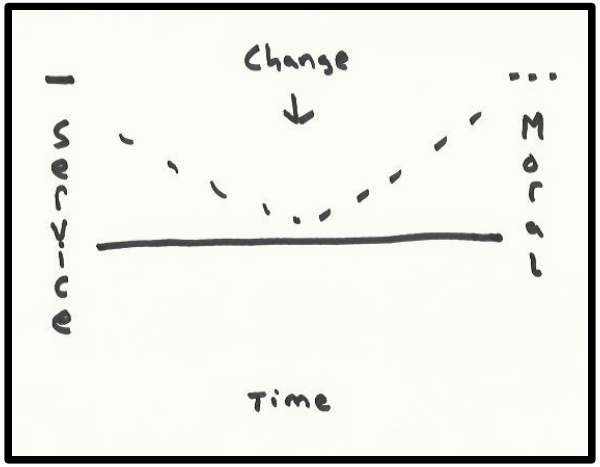
```
[root@keith archives]# cat harddowns.log | grep " 2010 " | grep -v Apr | petit --mograph --wide
57

           #
           #
   #   #   # #     # #
#   #   # # # #   # # #
# # #   # # # #   # # #
# # # # # # # # # # # #
01          07        12

Start Time:     2010-01-01 00:00:00             Minimum Value: 0
End Time:       2010-12-02 00:00:00             Maximum Value: 390
Duration:       12 months                       Scale: 65.0
```

Qualitative Data

- Service level did not change
  - Top customers where called each year and polled
  - No new complaints from customers
- Moral improved immensely
  - Instead of waiting when a page was received, operations responded immediately
  - Better communcation during outages



Conclusions

- Don't create artificial constraints
- Don't determine what is acceptable by some gut feeling
- Let business make decisions
- Don't ever say “should”, form theories
- Measure, measure, measure
- Support or Disprove
- Develop predictions

Familiarity: Monthly, Weekly, Daily Checklists

Bibliography

- Data Types: http://www.usablestats.com/lessons/noir
- FCAPS: http://en.wikipedia.org/wiki/FCAPS
- http://en.wikipedia.org/wiki/Scientific_method
- http://crunchtools.com/designing-a-robust-monitoring-system/
- http://crunchtools.com/