Centralized Logging

Logging Into A Centralized SQL Database

by Adam Tauno Williams (awilliam@whitemice.org)

Copyright

The home page for this presentation is:
http://www.whitemiceconsulting.com/node/103

UNIX/LINUX Logging

- Traditional UNIX/LINUX logging records messages in text files.
  - Logs live in some directory like "/var/log"
  - Log records have a fairly consistent format
    - Feb 15 18:36:33 aleph pppd[9495]: remote IP address 192.168.1.39
  - You search logs with tools like "grep" or text parsing scripts (perl, awk, python, etc...)
  - Most services send messages to a "syslog" daemon.
    - The syslog daemon actually writes the message, or not, to a file based upon some simple rules.

syslog

- The syslog service opens the /dev/log socket and listens for log messages.
  - $ fuser -u /dev/log
    /dev/log: 4391(root)
    $ ps ax | grep 4391
     4391 ? Ss 0:00 /sbin/syslog-ng
  - Most syslog services can also receive log messages over the network.
    - Log messages are sent in a UDP fire-and-forget manner.
      - syslog is no replacement for an ESB or MOM.
    - Logging over the network is not secure unless security is provided at some other layer.

Facilities & Levels

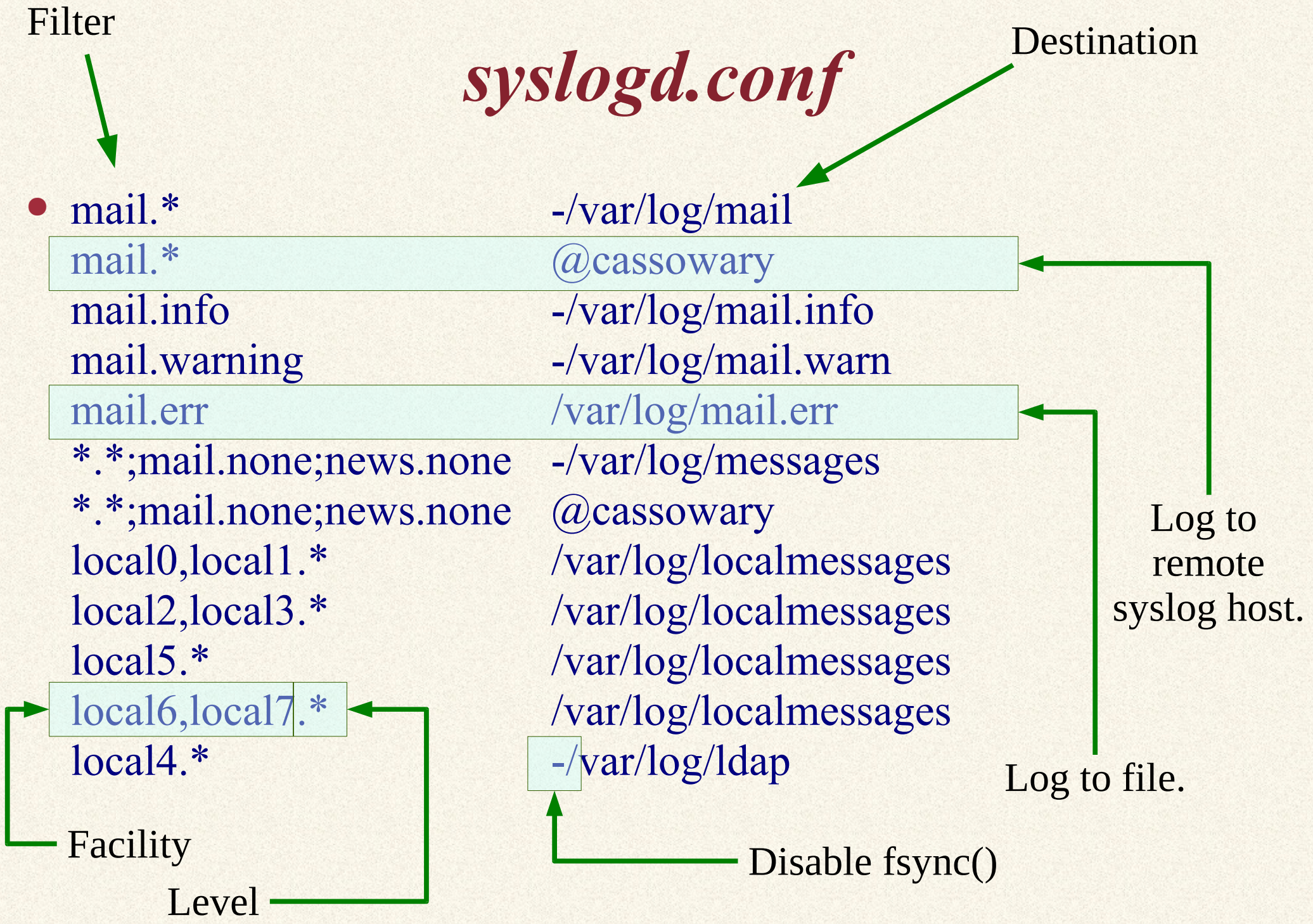
- Each log message has a *{facility}* and a *{level}*
  - *{facility}*
    - The source of the message.
      - authpriv, cron, daemon, kern, lpr, mail, mark, news, security, syslog, user, uucp, local0, local1, local2, local3, local4, local5, local6, local7
        - Use of "security" is deprecated, and equivalent to authpriv
        - local4 is used by OpenLDAP
        - local7 is typically used by Cisco routers
  - *{level}*
    - The severity or urgensy of the message
      - Emerg, alert, crit, err, warning, notice, info, debug, none

Facilities

- Facilities
  - Authpriv - Security and authorization messages.
  - Cron - Messages from the task scheduling system.
  - Daemon - Generic category for a message from a daemon
  - Kern - Messages from the kernel (klogd).
  - lpr - Print subsystem messages.
  - Mail - Messages from any MUA or MTA.
  - News - Messages from any NNTP server.
  - Syslog - Messages from syslog itself or another host's syslog.
  - User - Message sent from a user program.
  - Uucp - UUCP subsystem messages.
  - local0...local7 - Local/Administrator defined message categories

Levels

- Levels
  - Emerg - Panic, immediate attention required
    Alert - Important, action should be taken
    Crit - Some module or subsystem is inoperable.
    Err - General error.
    Warning - A recoverable error.
    Notice - A normal event has occurred.
    Info - Any informational message.
    Debug - Debugging messages.
  - The exact definition of the levels is frustratingly vague.
  - Specifying a level implies all higher level.s


Filter
syslogd.conf
Destination
mail.* -/var/log/mail
mail.* @cassowary
mail.info -/var/log/mail.info
mail.warning -/var/log/mail.warn
mail.err /var/log/mail.err
.;mail.none;news.none -/var/log/messages
.;mail.none;news.none @cassowary
local0,local1.* /var/log/localmessages
local2,local3.* /var/log/localmessages
local5.* /var/log/localmessages
local6,local7.* /var/log/localmessages
local4.* -/var/log/ldap
Log to remote syslog host.
Log to file.
Facility
Level
Disable fsync()

syslog listening

- Traditional syslogd can listen to domain sockets and UDP/514.
  - Listening to the network is enabled via the “-r” switch.
    - The “syslog” service must be listed in /etc/services.
  - Can listen on additional filesystem sockets via the “-a” option.
    - Useful for getting logging messages from chroot'd services.
      - /sbin/syslogd -a /var/lib/ntp/dev/log
    - Up to 19 additional sockets

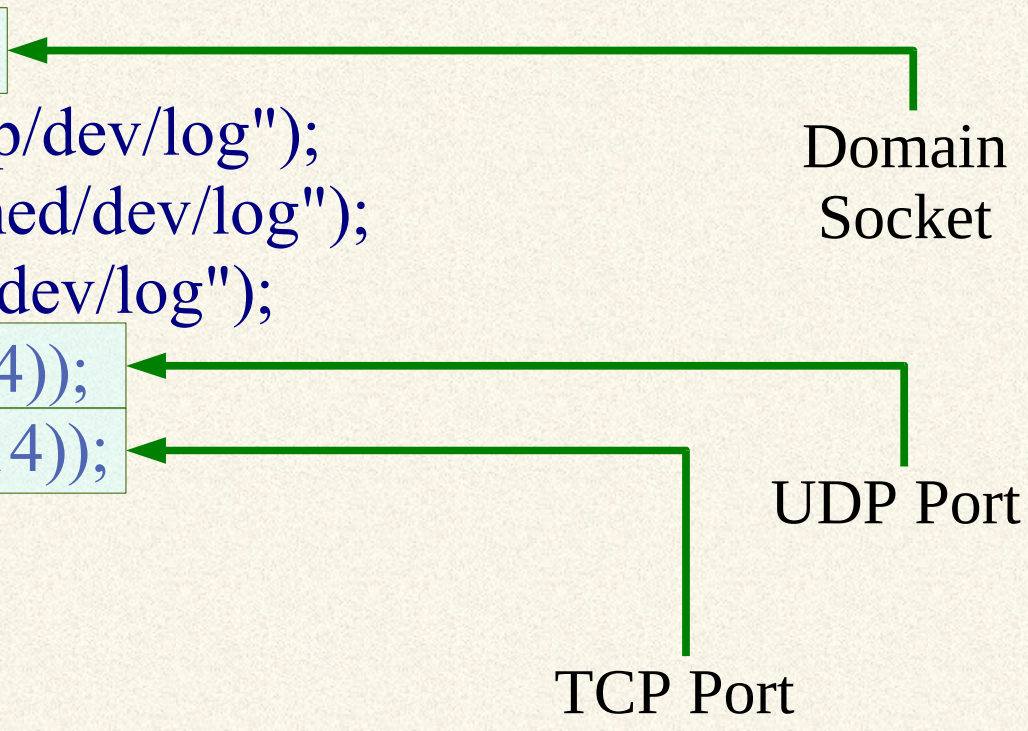
syslog-ng

http://www.balabit.com/products/syslog-ng/

- syslog-ng (SysLog Next Generation) was developed to replace the traditional syslogd which dates back to the early days of BSD UNIX.
  - Highly configurable
    - Log format can be specified.
  - Supports logging over TCP
    - Reliable verses unreliable UDP
  - Can log to a variety of destinations.
    - Files
    - Network (UDP or TCP)
    - Pipes
  - Currently maintained

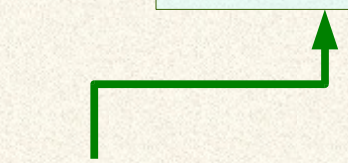
syslog-ng listening

- source src {
  # note: the internal() soure is required!
  internal();
  unix-dgram("/dev/log");
  unix-dgram("/var/lib/dhcp/dev/log");
  unix-dgram("/var/lib/named/dev/log");
  unix-dgram("/var/lib/ntp/dev/log");
  udp(ip("0.0.0.0") port(514));
  #tcp(ip(“0.0.0.0”) port(514));
  };



syslog-ng filters

```
filter f_iptables    { facility(kern) and match("IN=")
                         and match("OUT="); };
filter f_console     { level(warn) and facility(kern) and not filter(f_iptables)
                         or level(err) and not facility(authpriv); };
filter f_mailinfo    { level(info) and facility(mail); };
filter f_mailwarn    { level(warn) and facility(mail); };
filter f_mailerr     { level(err, crit) and facility(mail); };
filter f_mail        { facility(mail); };
filter f_cron        { facility(cron); };
filter f_local       { facility(local0, local1, local2, local3,
                         local4, local5, local6, local7); };
filter f_messages    { not facility(news, mail) and not filter(f_iptables); };
```

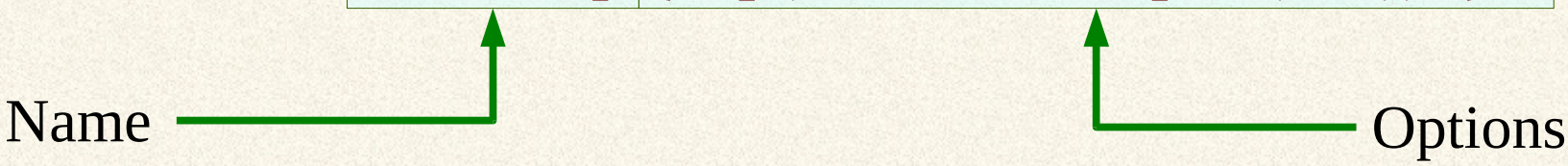


Name

Criteria

syslog-ng destinations

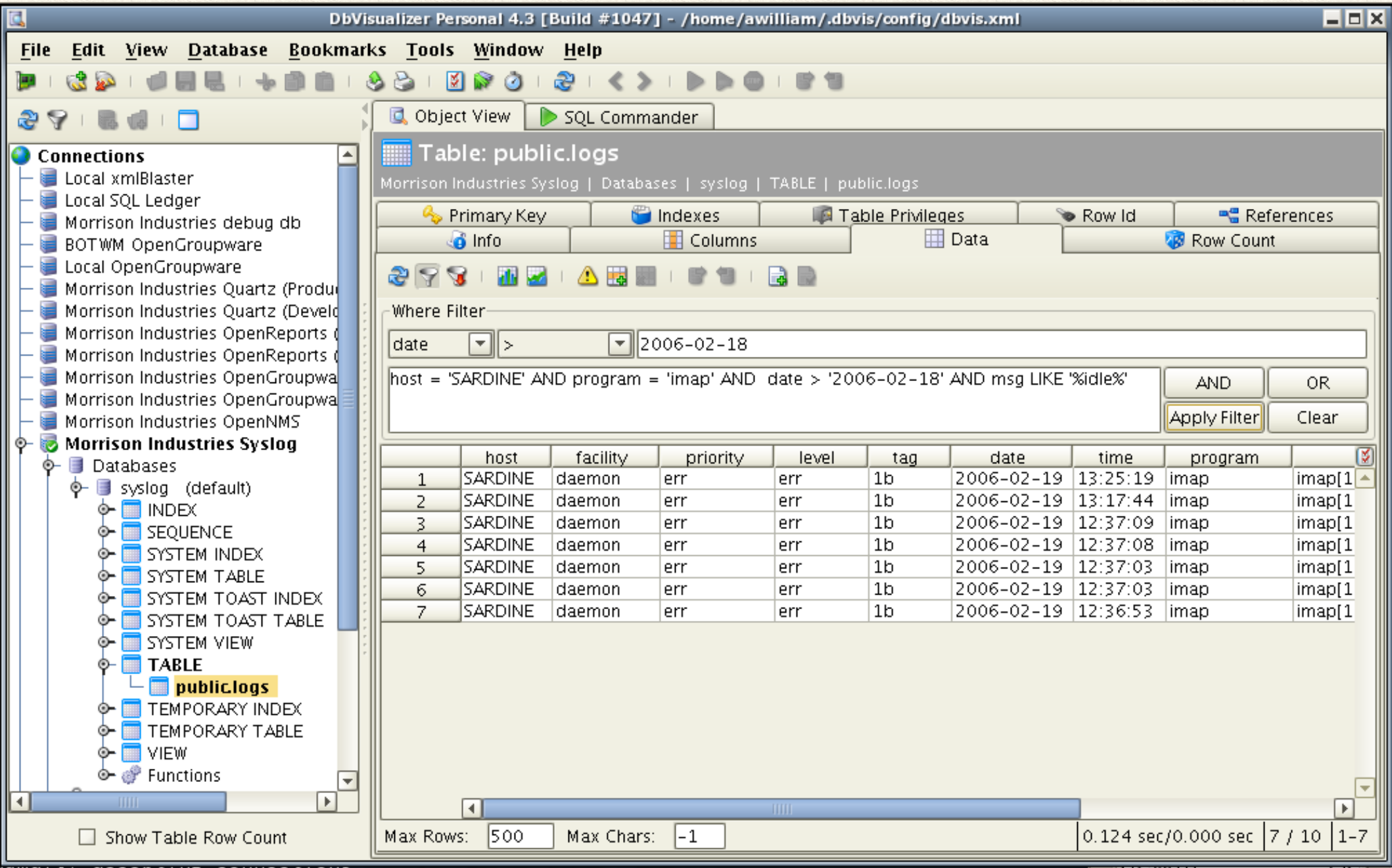
- destination console { file("/dev/tty10" group(tty) perm(0620)); };
  destination xconsole { pipe("/dev/xconsole" group(tty) perm(0400)); };
  destination mailinfo { file("/var/log/mail.info"); };
  destination mailwarn { file("/var/log/mail.warn"); };
  destination mailerr { file("/var/log/mail.err" fsync(yes)); };
  destination mail { file("/var/log/mail"); };
  destination localmessages { file("/var/log/localmessages"); };
  destination messages { file("/var/log/messages"); };
  destination firewall { file("/var/log/firewall"); };
  destination warn { file("/var/log/warn" fsync(yes)); };
  destintation remoteudp { udp("otherhost", port(514)); };
  destintation remotetcp { tcp("otherhost", port(514)); };



Logging To A Database

- Logging to an SQL database makes doing ad-hoc analysis much easier.
  - select host, program, count(*)
from facility_daemon
where host = 'SARDINE'
group by host, program
order by 3;
    - You can discover interesting patterns and turn up repetitive messages much easier than by groping through text files.
    - You can use sophisticated tools like DbVisualizer.

DbVisualizer Looking At Logs



	host	facility	priority	level	tag	date	time	program	
1	SARDINE	daemon	err	err	1b	2006-02-19	13:25:19	imap	imap[1
2	SARDINE	daemon	err	err	1b	2006-02-19	13:17:44	imap	imap[1
3	SARDINE	daemon	err	err	1b	2006-02-19	12:37:09	imap	imap[1
4	SARDINE	daemon	err	err	1b	2006-02-19	12:37:08	imap	imap[1
5	SARDINE	daemon	err	err	1b	2006-02-19	12:37:03	imap	imap[1
6	SARDINE	daemon	err	err	1b	2006-02-19	12:37:03	imap	imap[1
7	SARDINE	daemon	err	err	1b	2006-02-19	12:36:53	imap	imap[1

Create Database

- Create database & Users
  - postgres@cassowary:~> createuser syslog
    Shall the new user be allowed to create databases? (y/n) n
    Shall the new user be allowed to create more new users? (y/n) n
    postgres@cassowary:~> createuser -P logview
    Enter password for new user: *********
    Enter it again: *******
    Shall the new user be allowed to create databases? (y/n) n
    Shall the new user be allowed to create more new users? (y/n) n
    postgres@cassowary:~> createdb -O syslog syslog
    CREATE DATABASE
- Setup Access (pg_hba.conf)
  - host syslog syslog 127.0.0.1 255.255.255.255 trust
    host syslog logview 192.168.1.0 255.255.255.0 password
    host syslog all 192.168.1.0 255.255.255.0 pam
  - postgres@cassowary:~> pg_ctl reload

Create Log Table

- postgres@cassowary:~> psql -U syslog syslog
  CREATE TABLE facility_*{facility}* (
  host varchar(32) default NULL,
  priority varchar(10) default NULL,
  level varchar(10) default NULL,
  tag varchar(10) default NULL,
  date date default NULL,
  time time default NULL,
  program varchar(15) default NULL,
  msg text,
  seq serial,
  PRIMARY KEY (seq)
  );

Create Indexes

- CREATE INDEX facility_{facility}_i0
  ON facility_{facility}(host);
  CREATE INDEX facility_{facility}_i1
  ON facility_{facility}(date);
  CREATE INDEX facility_{facility}_i2
  ON facility_{facility}(host, priority);
  CREATE INDEX facility_{facility}_i4
  ON facility_{facility}(host, priority, program);
  CREATE INDEX facility_{facility}_i5
  ON facility_{facility}(host, program);

Grant Access

- Make syslog user so all it can do is insert and select.
  - GRANT INSERT, SELECT ON logs TO syslog;
    REVOKE UPDATE,DELETE ON logs FROM syslog;
- Revoke all permissions from the public
  - REVOKE INSERT,UPDATE,DELETE,SELECT
    ON logs FROM public;
- Grant all rights to administrators
  - GRANT SELECT,UPDATE,DELETE,INSERT
    ON logs TO adam, rhopkins, steve;

The Script

- ```
  #!/bin/bash
  if [ -e /var/run/pgsyslog.pipe ];
   then
    while [ -e /var/run/pgsyslog.pipe ]
      do
        psql -q -U syslog syslog < /var/run/pgsyslog.pipe
      done
   else
     mkfifo /var/run/pgsyslog.pipe
     chown root.root /var/run/pgsyslog.pipe
     chmod 600 /var/run/pgsyslog.pipe
   fi
  ```
  ```

Aiming Syslog-ng at the pipe

- destination d_pgsql {
  pipe("/var/run/pgsyslog.pipe"
  template("INSERT INTO facility_$FACILITY (host, priority,
  level, tag, date, time, program, msg)
  VALUES (UPPER('$HOST'),
  '$PRIORITY', '$LEVEL', '$TAG',
  '$YEAR-$MONTH-$DAY', '$HOUR:$MIN:$SEC',
  '$PROGRAM', '$MSG');\n"
)
  template-escape(yes)
  );
  };
  log { source(src); destination(d_pgsql); };

The template grammer is used to construct a simple SQL INSERT statement.

Running The Script

- Edit “/etc/rc.d/syslog”
  - test -z "$KERNEL_LOGLEVEL" && KERNEL_LOGLEVEL=1
    nohup /usr/local/sbin/pg_syslogd.sh 2>&1 > /dev/null &
    startproc -p ${syslog_pid} ${BINDIR}/${syslog} $params
    rc_status
  - This starts the script before syslogd is started
    - cassowary:/etc/syslog-ng # ps ax |grep syslog
      18302 ? S 0:00 /bin/bash /usr/local/sbin/pg_syslogd.sh
      26407 ? Ss 5:14 /sbin/syslog-ng
      28151 ? S 3:08 psql -q -U syslog syslog
      28155 ? S 28:32 postgres: syslog syslog [local] idle

Huge Amounts Of Data

- If you have many server or a very active network you will generate ALLOT of messages.
  - We average about 180,000 messages each business day and about.
  - 30 days will be about 3,960,000 records in the log table.
    - (30 – (INT(30/7)*2)) * 180,000 = 3,960,000
  - If you intend to actually use this data you will need to tune your database
    - Default settings do not work well with tables containing large numbers of records.
    - You need to update statistics frequently.

Update Statistics

- Add this database to you update statistics regimin.
  - /etc/cron.daily/vacuum
    - #!/bin/sh

```
/usr/bin/psql -U syslog syslog \
  -c "VACUUM ANALYZE VERBOSE" 2>&1 \
 | mail -s "Syslog Vacuum" \
     shared+departments.cis.status.cassowary@sardine

/usr/bin/psql -U opennms opennms \
  -c "VACUUM ANALYZE VERBOSE" 2>&1 \
  | mail -s "OpenNMS Vacuum" \
     shared+departments.cis.status.cassowary@sardine
```

Generate Statistics

- With logs in an RDMBS it is trivial to generate statistics concerning the content of the logs.

Facility	Count			
daemon	2,591,486	**Program**	**Count**	
mail	1,467,761	FaxGetty	3903	0.15%
auth	130,116	ctl_cyrusdb	4067	0.16%
local2	26,886	squid	4072	0.16%
cron	13,423	nscd	4504	0.17%
kern	3,870	FaxQueuer	6156	0.24%
user	3,675	imaps	6898	0.27%
syslog	3,667	cyr_expire	6980	0.27%
local7	321	FaxSend	10221	0.39%
local4	107	last	14668	0.57%
	4,241,312	master	19093	0.74%
		dhcpd	30807	1.19%
		named	40538	1.56%
		lmtpunix	90351	3.49%
		smbd	223018	8.61%
		nmbd	869588	33.56%
		imap	1251994	48.31%

Links

- A similair SQL logging setup for MySQL
  - http://www.linuxlabs.biz/articles/syslog.htm
- Syslog home page
  - http://www.balabit.com/products/syslog-ng/
- Syslog client for M$-Windows (Managed C++)
  - http://www.edoceo.com/products/winlogd.php
  - Other non-UNIX logging information
    - http://www.loganalysis.org/sections/syslog/windows-to-syslog/
- My older logging presentation
  - ftp://kalamazoolinux.org/pub/pdf/Timber.pdf