4. Mailserver-Konferenz 2009

Mark Martinec
Institut “Jožef Stefan”, Slovenia

Amavis (amavisd-new) Configuration and Management

http://www.ijs.si/software/amavisd/

Agenda

- what it is, project history
- performance / benchmark / tuning
- configuration: policy banks, lookups, cc
- fitting it all together
- monitoring

amavisd-new - what is it?

- interface between MTA and
  virus checkers and/or spam checkers
- checks for banned contents
- quarantining/archiving
- DKIM: signs and verifies signatures
- monitoring: SNMP, SQL log, nanny

why is it popular?

- reliable:
  - checks status of every operation, internal asserts
  - in case of a failure mail stays with MTA
- adheres to standards (SMTP, MIME, DSN, ...)
- reasonably fast, reasonably feature-rich
- security: perl, taint checks, can run chroot-ed
- mature: 7+ years of steady development
- OSS: GPL license (+ BSD licensed tools)

AMaViS history

shell program:

- 1997 Mogens Kjaer, Juergen Quade
- 1998-01-17 AMaViS 0.1 (Christian Bricart) - 300 lines
  "AMaViS - A Mail Virus Scanner"
- 1998-12 AMaViS 0.2.0-pre1
- 1999-07 AMaViS 0.2.0-pre6 (Rainer Link, Chris Mason)
- 2000-10 AMaViS 0.2.1 (Christian Bricart)

Perl program:

- 2000-01 Amavis-perl (Chris Mason)
- 2000-08 Amavis-perl-8
- 2000-12 Amavis-perl-10
- 2001-04 Amavis-perl-11 (split> amavisd)
- 2003-03 Amavis-0.3.12 (Lars Hecking)

AMaViS history

Perl daemon:

- 2001-01 daemonisation (Geoff Winkless)
- 2001-04 amavisd-snapshot-20010407 (Lars Hecking)
- 2001-07 amavisd-snapshot-20010714
- >2002-04 amavisd-snapshot-20020300 (split> amavisd-new)
- 2003-03 amavisd-0.1

Perl, modular re-design

- 2002-03 amavis-ng-0.1 (Hilko Bengen)
- 2003-03 amavis-ng-0.1.6.2 (Hilko Bengen)

Releases, milestones…

- 2002-03-29 amavisd-new, pre-forked, Net::Server
- 2002-05-17
- 2002-06-30 packages, SQL lookups
- 2002-11-16 integrated - one file
- 2002-12-27
- 2003-03-14 LDAP lookups
- 2003-06-16
- 2003-08-25 p5
- 2003-11-10 p6 @*_maps
- 2004-01-05 p7
- 2004-03-09 p8
- 2004-04-02 p9
- 2004-06-29 p10

...releases, milestones...

2004-07-01	2.0	policy banks, IPv6 address formats
2004-08-15	2.1.0	amavisd-nanny
2004-09-06	2.1.2	
2004-11-02	2.2.0	
2004-12-22	2.2.1	
2005-04-24	2.3.0	@decoders, per-recipient banning rules
2005-05-09	2.3.1	
2005-06-29	2.3.2	
2005-08-22	2.3.3	
2006-04-02	2.4.0	DSN in SMTP, %*_by_ccat
2006-05-08	2.4.1	
2006-06-27	2.4.2	pen pals, SQL logging and quarantine
2006-09-30	2.4.3	
2006-11-20	2.4.4	
2007-01-30	2.4.5	

...releases, milestones...

- 2007-04-23 2.5.0 blocking cc, new SMTP cl.
- 2007-05-31 2.5.1 amavisd-requeue

  2007-06-26 SpamAssassin committer
- 2007-06-27 2.5.2
- 2007-12-12 2.5.3

  2008-01-13 SpamAssassin PMC member (Project Management Committee)
- 2008-03-12 2.5.4
- 2008-04-23 2.6.0 DKIM, bounce killer, TLS
- 2008-06-29 2.6.1

...releases, milestones

- 2008-12-12 Amavis

  It's spelled **N-e-t-s-c-a-p-e**,

  but it's pronounced "**Mozilla**".

  It's spelled **amavisd-new**,

  but it's pronounced "**Amavis**".

Christian Bricart (domain: amavis.org)

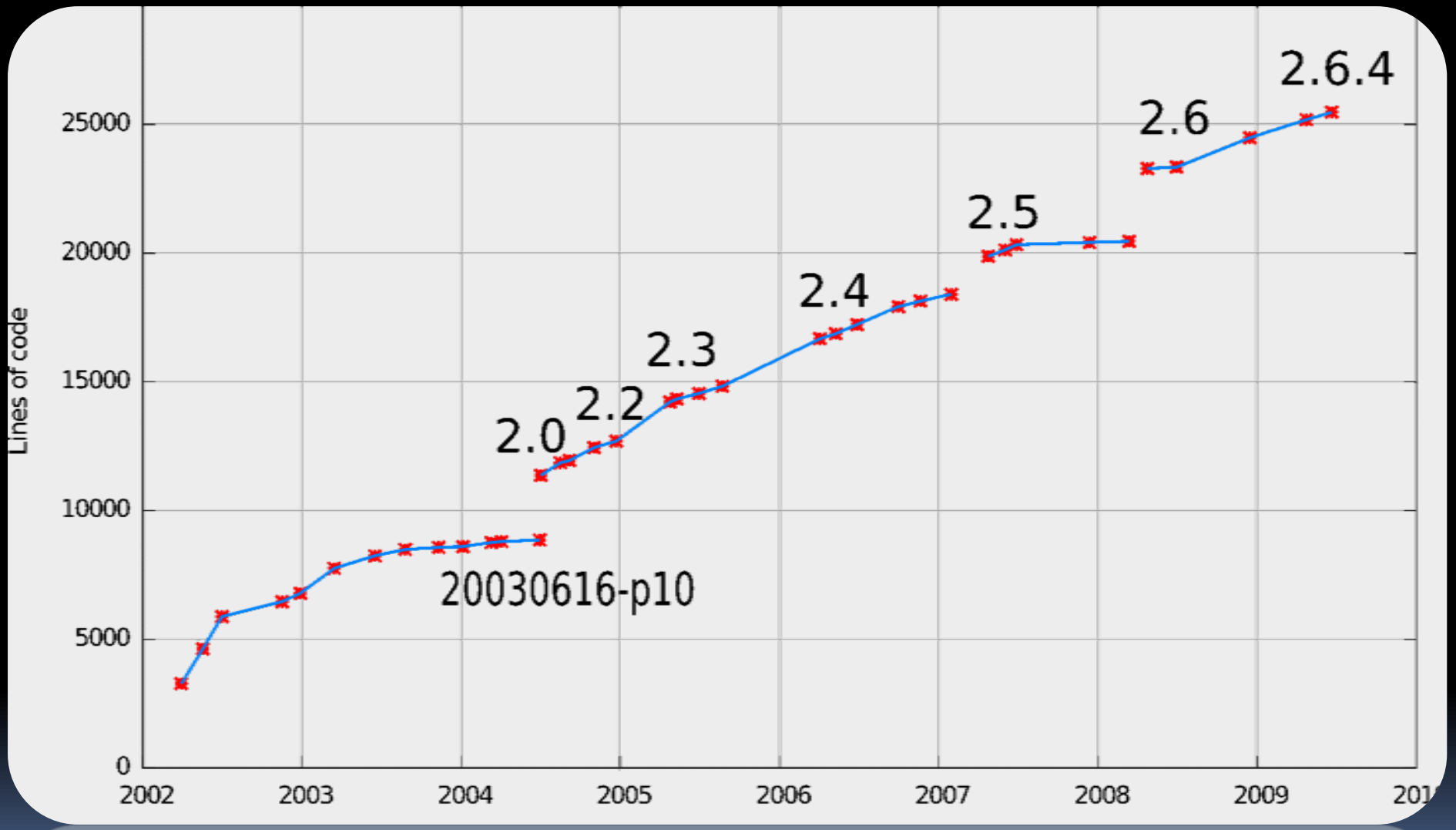
Lars Hecking, Hilko Bengen, Rainer Link

...releases, milestones

- 2008-12-15 2.6.2
- 2009-04-22 2.6.3 CRM114, DSPAM, trunc.
- 2009-06-25 2.6.4 SNMP, last week

7+ years of
amavisd-new development

amavisd code size growth



Did it grow too large?

- modules, loaded only what is needed
- half of memory footprint is SpamAssassin
- size does not effect speed,
  critical code paths are well optimized
- grows linearly, hardware exponentially

Memory size (VSZ, i386)

- 4.5 MB just perl 5.10.0
- 22.4 base amavisd, no optional modules
- + 1.0 SMTP in/out
- + 2.2 decoding
- + 1.0 anti-virus & basic anti-spam interface
- + 2.6 SQL (lookups, log, quarantine)
- + 3.2 Berkeley databases
- + 3.7 DKIM signing and verification
- + 26 SpamAssassin (3.2.5), or +36 MB (3.3)

- 56 MB all (bdb, decode, SMTP, DKIM, AV, SA)

compare to: 30 MB spamd

Memory size (VSZ, 64-bit OS)

- 10 MB — just perl 5.10.0
- 103 — base amavisd, no optional modules
- + 1.0 — SMTP in/out
- + 4.2 — decoding
- + 1.0 — anti-virus & basic anti-spam interface
- + 2.6 — SQL (lookups, log, quarantine)
- + 5.6 — Berkeley databases
- + 3.0 — DKIM signing and verification
- + 87 — SpamAssassin (3.3, SARE, ...)

- 203 MB — all (bdb, decode, SMTP, DKIM, AV, SA 3.3)

compare to: 150 MB spamd

32-bit vs. 64-bit platform

- perl programs at least twice as big,
  no obvious advantage

- 8 GB of memory still suffices
  for 50 full-blown processes
  on a 64-bit platform

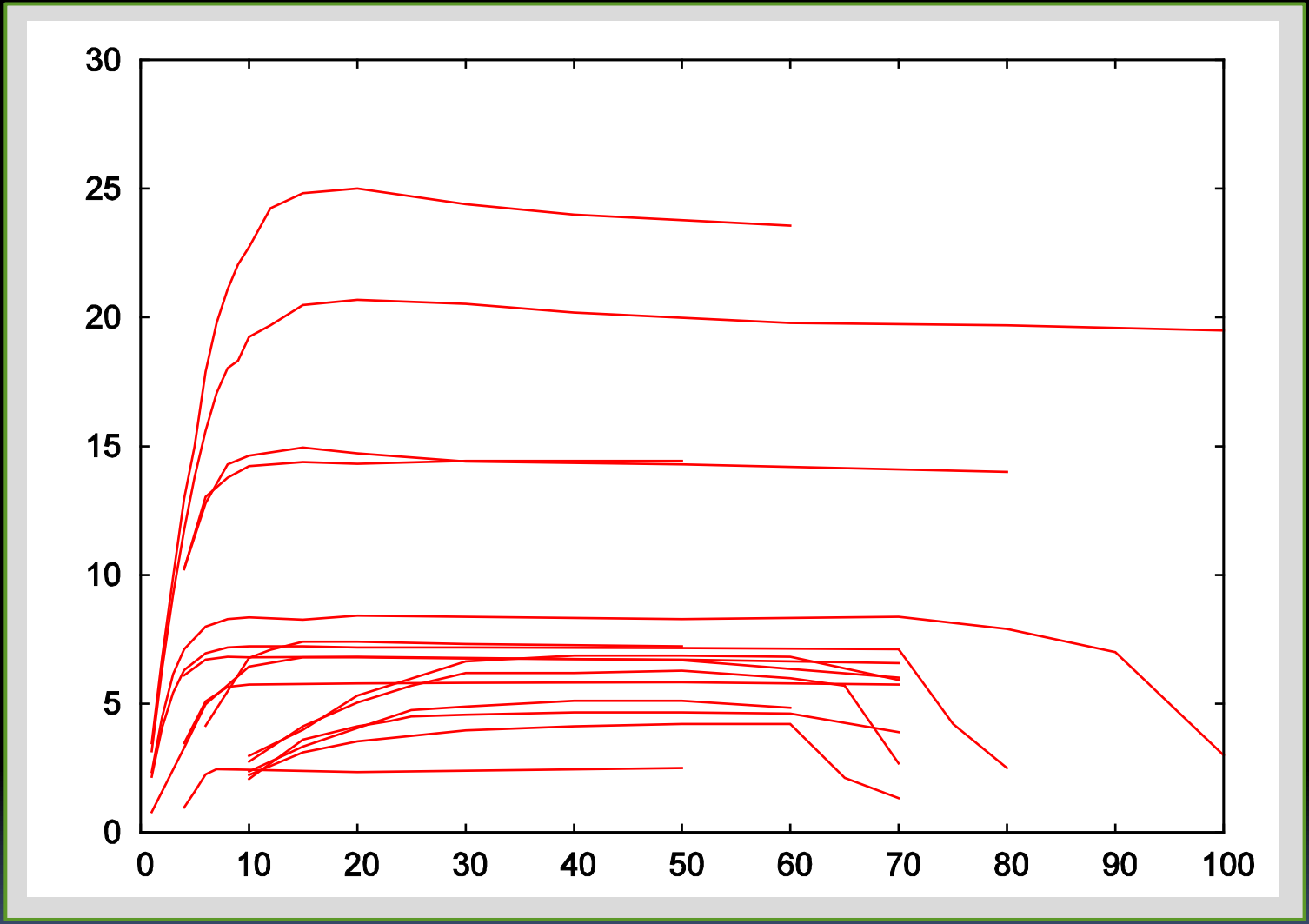
Memory sharing

- RSS/VSZ = 60% is memory-resident
- cca 60% of a process' RSS is shared
- cca 30 MB real memory for a 100 MB virtual memory process

- 1 GB: 25 processes - just manages to reach optimum with all checks enabled, SA 3.2.5, default rules, unused sections paged out

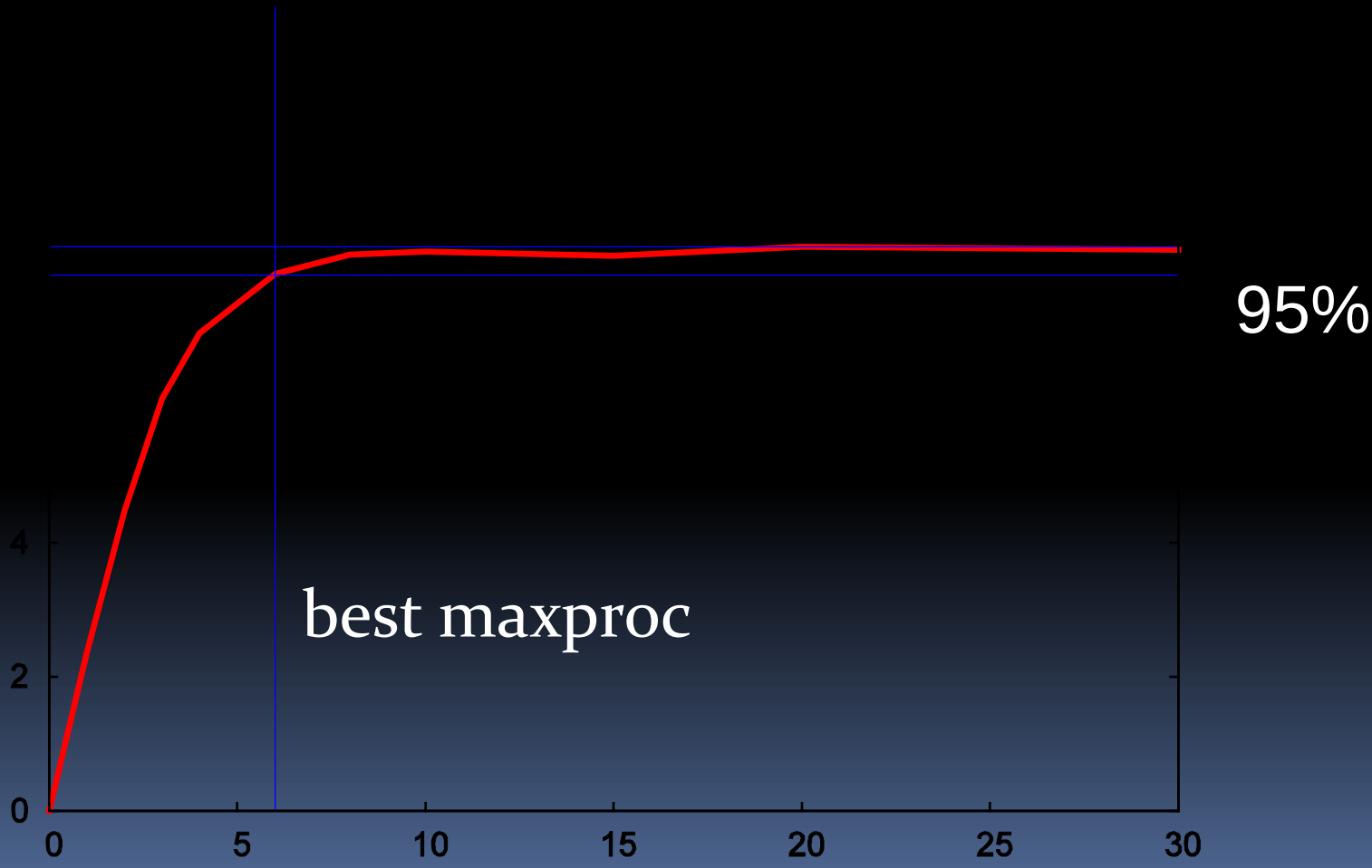
Performance – general idea

msgs/s *vs.* maxproc



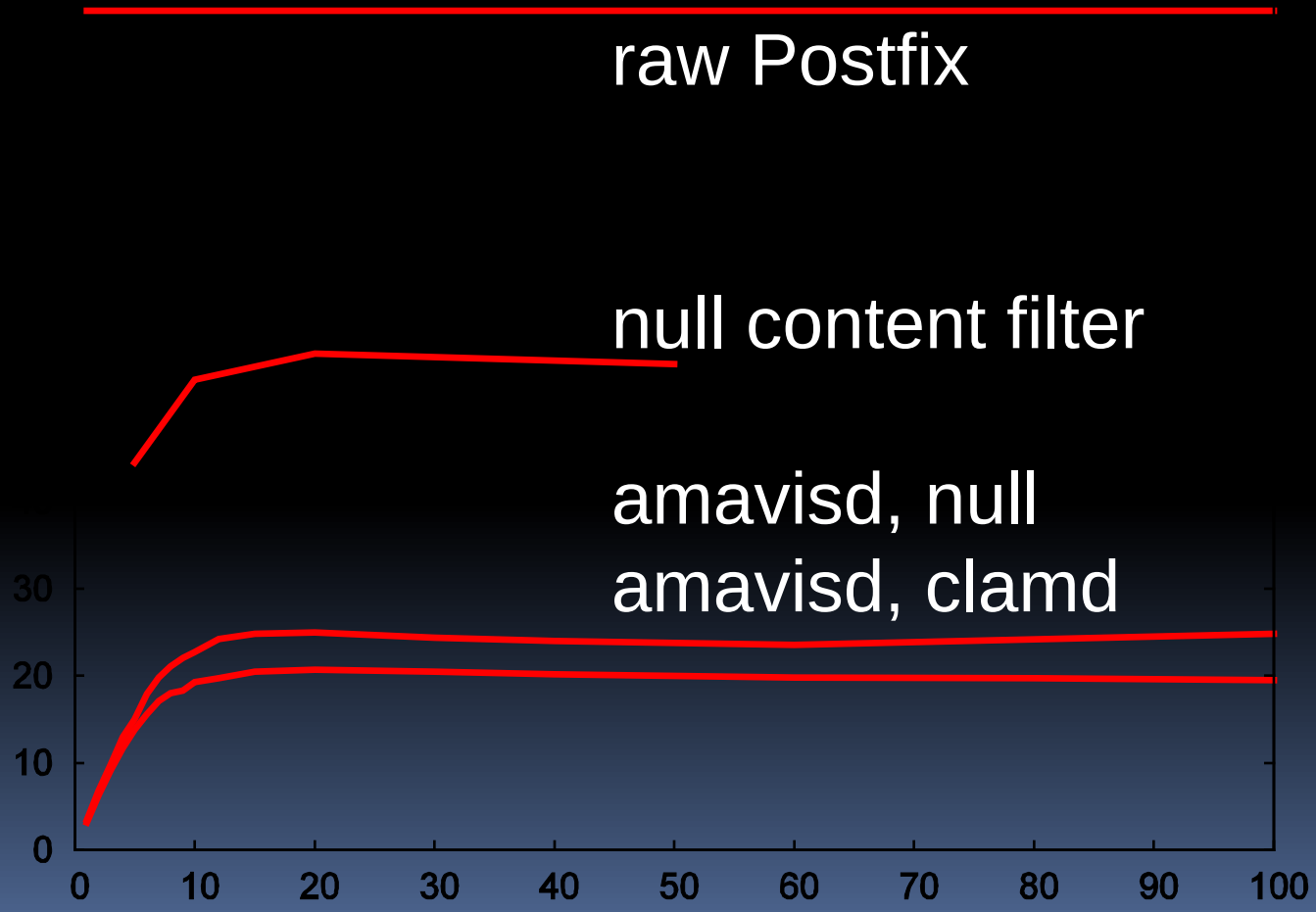
Performance – general idea

msgs/s *vs.* maxproc



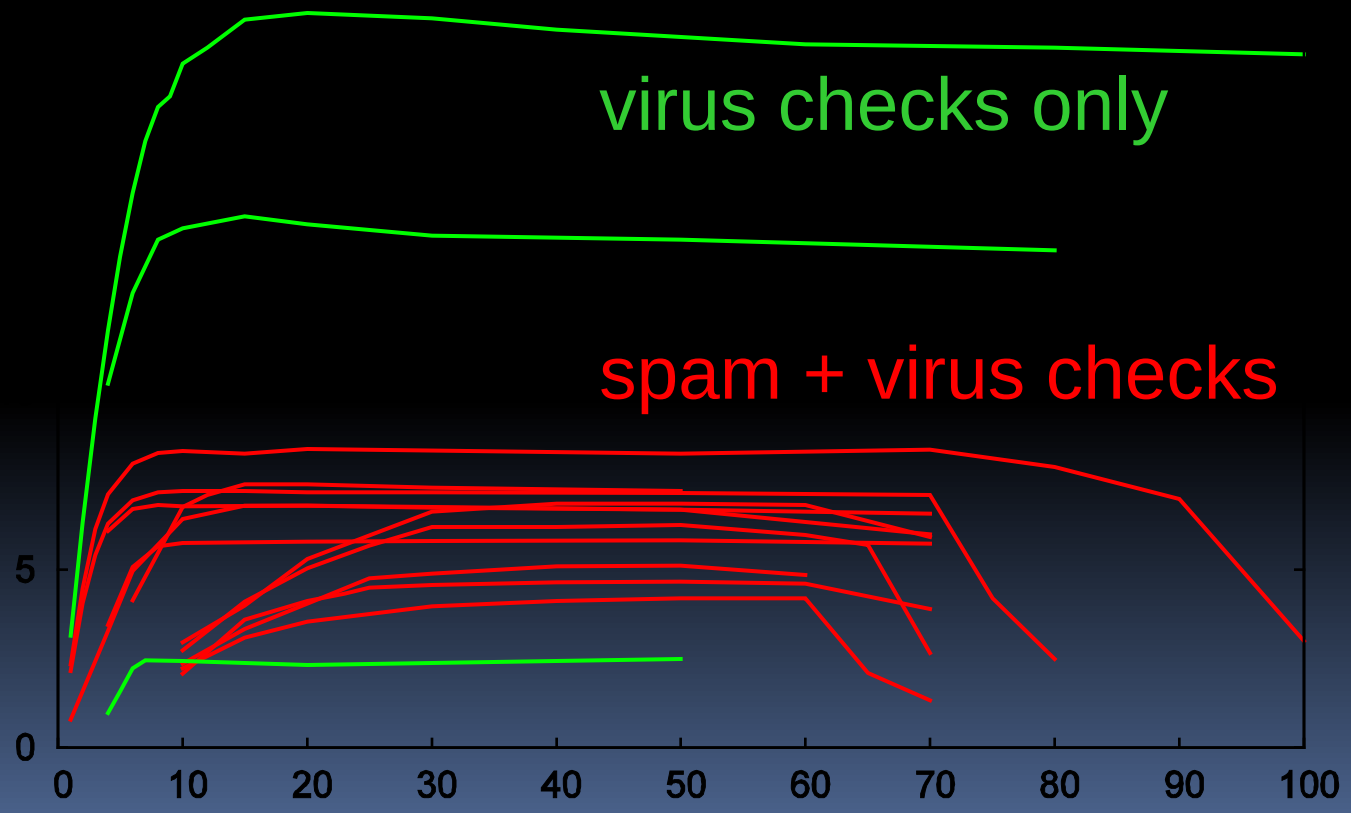
Performance: baseline

msgs/s *vs.* maxproc



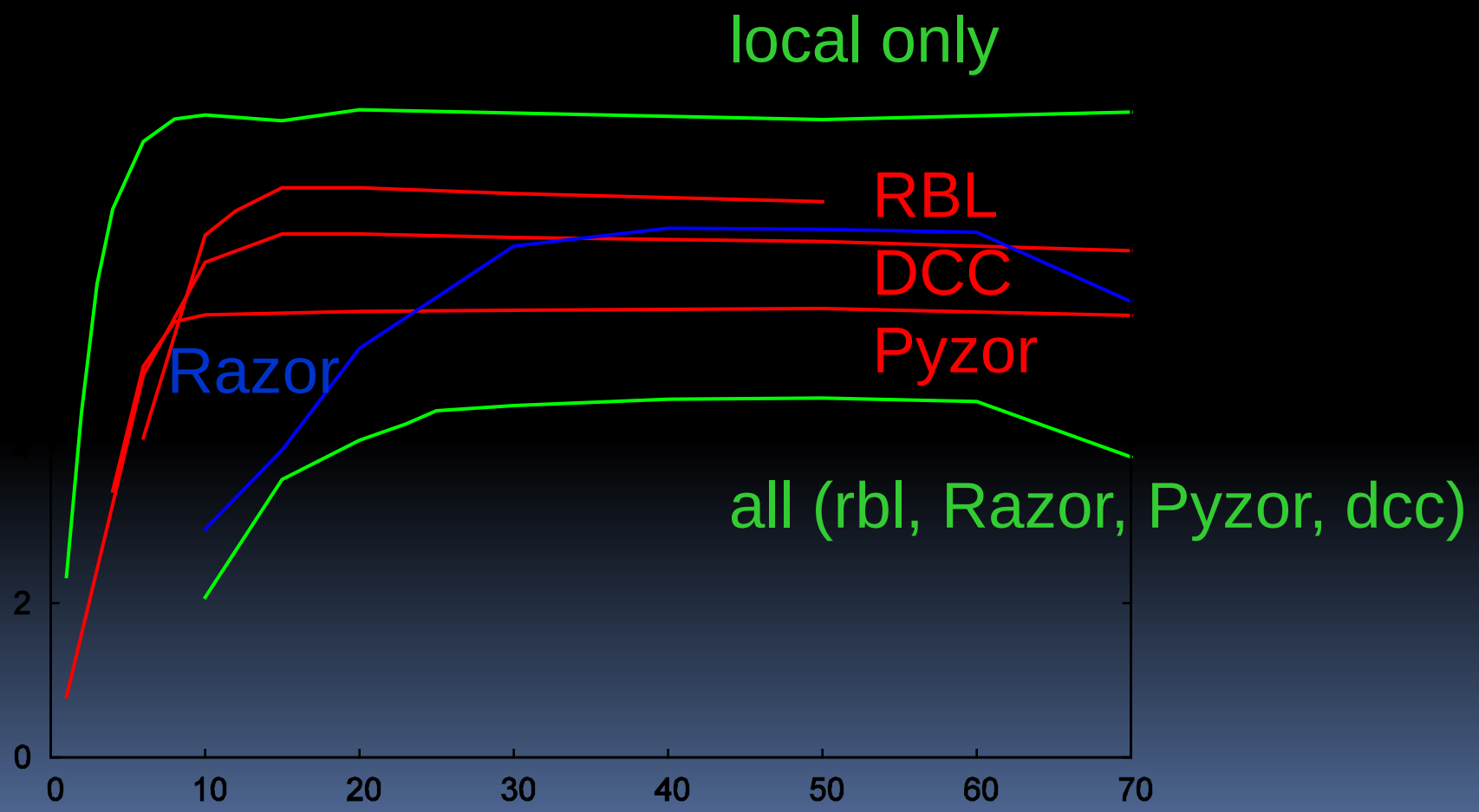
Performance: SpamAssassin

msgs/s *vs.* maxproc



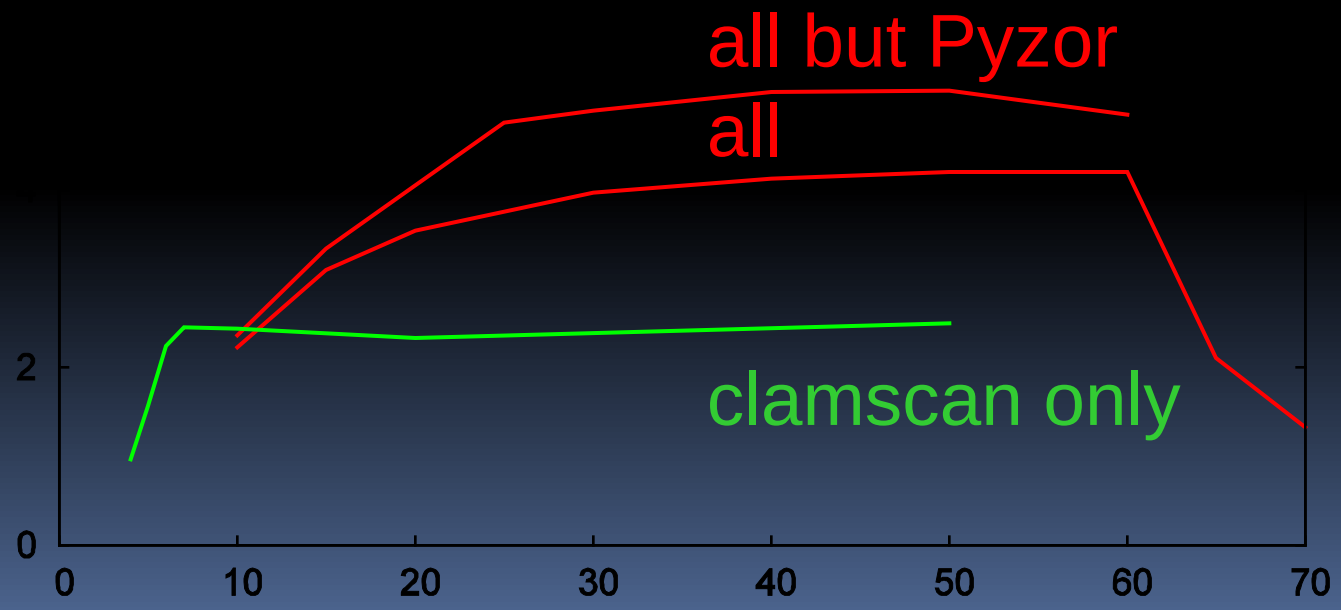
Performance: SpamAssassin

msgs/s *vs.* maxproc



Performance: can it get worse?

msgs/s *vs.* maxproc



Tuning: general

- choose number of processes
- avoid slow command-line virus scanners
- separate disks for MTA spool and amavisd tmp
- Linux syslogd: disable sync on MTA/amavisd logs
- some rulesets can be expensive (SARE)
- turn on *$quarantine_subdir_levels* = 1

- separate MTA and amavisd hosts
- split load through multiple MX records

Configuration – agenda

- general
- mail flow direction
- logging, syslog
- interfacing: input, output, milter
- policy banks
- lookups
- content categories

Configuration – general

all config settings: amavisd.conf-default

- directories, hostname, ...
- user (uid)
- destination, source
- $max_servers
- $nanny_details_level = 2; # verbosity: 0, 1, 2

Configuration – mail flow direction

- origin: @mynetworks, $originating
- destination: @local_domains_maps

0 0 … open relay
0 1 … inbound
1 0 … outbound
1 1 … internal–to-internal

Configuration – destination

list all your domains in @local_domains_maps
(local, virtual aliases, virtual mailbox, relay)

affects:

- inserting header fields X-Quarantine-ID, X-Spam-*, X-Amavis-OS-Fingerprint, ...
- adding address extension (*plus addressing*)
- recipient notifications
- pen pals
- defanging

Configuration – origin (source)

origin: @mynetworks, $originating

affects:

- DKIM signing
- inserting disclaimers
- bounce killer
- pen pals
- MYUSERS policy bank

Configuration – origin (source)

setting the $originating flag:

- implicitly: @mynetworks
- explicitly, typically through a policy bank:

```
$inet_socket_port = [10024, 10026];
$interface_policy{'10026'} = 'ORIG';

$policy_bank{'ORIG'} = {
    originating => 1,
};
```

Mail direction in SpamAssassin

- internal_networks
- trusted_networks
- msa_networks

Configuration – logging

SA	amavisd	syslog
	-3	LOG_CRIT
	-2	LOG_ERR
error	-1	LOG_WARNING
warn	0	LOG_NOTICE
info	1	LOG_INFO
	2	LOG_INFO
dbg	3	LOG_DEBUG
	4	LOG_DEBUG
	5	LOG_DEBUG

Configuration – syslog

```
$DO_SYSLOG = 1;

$syslog_facility = 'user';

$log_level = 2;          # verbosity 0..5
```

Configuration – /etc/syslog.conf

user.err; mail.crit; ... /var/log/messages
user.notice /var/log/amavisd.log
user.info /var/log/amavisd-info.log
user.debug /var/log/amavisd-debug.log

Prepend ' – ' to a filename on Linux
to disable sync!

Configuration – log template

```
$log_templ = <<'EOD';
[?%#D|#|Passed #
[...]
[? %q ||, quarantine: %q]#
[? %Q ||, Queue-ID: %Q]#
[? %m ||, Message-ID: %m]#
[? %r ||, Resent-Message-ID: %r]#
, mail_id: %i#
, Hits: [:SCORE]#
, size: %z#
[...]
EOD
```

Configuration – log template

$log_templ
$log_recip_templ

macros: README.customize

From, Subject, Message-Id, User-Agent,
size, Hits, Tests, banning, DKIM id, ...

Configuration – input interface

SMTP or LMTP or AM.PDP or AM.CL on input

```
$inet_socket_port = [10024, 10026, 10027];
                        # TCP port numbers

@inet_acl = qw( 127.0.0.0/8 [::1] 192.168.1.1 );
                        # access control

$inet_socket_bind = '127.0.0.1';
                        # restrict to one interface

$unix_socketname = '/var/amavis/amavisd.sock';
                        # quarantine release or milter
```

Configuration – output

SMTP or LMTP or pipe on output

```
$forward_method = 'smtp:[127.0.0.1]:10025';
$notify_method  = 'smtp:[127.0.0.1]:10025';

$forward_method = 'smtp:*:*';
$notify_method  = 'smtp:*:10587';
```

- 1st asterisk use SMTP client peer address
- 2nd asterisk incoming SMTP/LMTP session port no. plus one

$virus_quarantine_method, $spam_quarantine_method, …

Configuration – milter setup

```
$unix_socketname =
    '/var/amavis/amavisd.sock';

$interface_policy{'SOCK'} = 'SOMEMILTER';

$policy_bank{'SOMEMILTER'} = {
    protocol => 'AM.PDP',
};

$forward_method = undef;
$notify_method = 'pipe: ... sendmail -Ac -i –odd
    -f ${sender} -- ${recipient}';
```

Policy banks

- one global, currently in effect,
  set of configuration variables

- several replacement sets (groups)
  of configuration variables,
  prepared in advance and on stand-by,
  quickly loadable

- affects message as a whole (not per-recipient)

Policy banks

RED

```
$a = ”red”;
$b = 4;
$c = ”ABC”;
```

GREEN

```
$a = ”green”;
```

BLUE

```
$a = ”blue”;
$b = 99;
@d = (88);
```

current

```
$a = ”black”;
$b = 2;
$c = undef;
@d = (1, 2, 3);
```

Policy banks

RED

```
$a = ”red”;
$b = 4;
$c = ”ABC”;
```

GREEN

```
$a = ”green”;
```

BLUE

```
$a = ”blue”;
$b = 99;
@d = (88);
```

current

```
$a = ”blue”;
$b = 99;
$c = undef;
@d = (88);
```

Policy banks

RED

```
$a = ”red”;
$b = 4;
$c = ”ABC”;
```

GREEN

```
$a = ”green”;
```

BLUE

```
$a = ”blue”;
$b = 99;
@d = (88);
```

current

```
$a = ”green”;
$b = 99;
$c = undef;
@d = (88);
```

Policy banks

RED

```
$a = ”red”;
$b = 4;
$c = ”ABC”;
```

GREEN

```
$a = ”green”;
```

BLUE

```
$a = ”blue”;
$b = 99;
@d = (88);
```

current

```
$a = ”red”;
$b = 4;
$c = ”ABC”;
@d = (88);
```

Policy banks – Perl syntax

normal settings

- variables, assignments

  $a = ”xyz”;

  @m = (1, 2, ”xyz”);

  %h = (a => 1, b => 2);

- separator: semicolon
- list: (1, 2, 3)
- hash: (a => 1, b => 2)

within a policy bank

- key / value pairs

  a => ”xyz”,

  m => [1, 2, ”xyz”],

  h => { a => 1, b => 2 },

- separator: comma
- list reference: [1, 2, 3]
- hash ref: { a => 1, b => 2 }

Policy banks – examples

```
$policy_bank{'NOVIRUSCHECK'} = {
    bypass_decode_parts => 1,
    bypass_virus_checks_maps => [1],
    virus_lovers_maps => [1],
};

$policy_bank{'AM.PDP-SOCK'} = {
    protocol => 'AM.PDP',
    auth_required_release => 0,
    syslog_ident => 'amavis-release',
};
```

Policy banks – example

```
$policy_bank{'ALT'} = {
  originating              => 1,
  log_level                => 2,
  forward_method    => 'smtp:*:*',
  local_client_bind_address  => '193.2.4.6',
  localhost_name  => 'extra.example.com',
  final_spam_destiny         => D_PASS,
  spam_kill_level_maps     => 6.72,
};
```

Policy banks – activating by interface

```
$inet_socket_port =
    [10024, 10026, 10028, 10030, 9998];

$interface_policy{'10026'} = 'ORIGINATING';
$interface_policy{'10028'} = 'NOCHECKS';
$interface_policy{'10030'} = 'CUSTOMER';
$interface_policy{'9998'}  = 'AM.PDP-INET';
$interface_policy{'SOCK'} = 'AM.PDP-SOCK';
```

Policy banks – by client's IP address

```
my(@some_nets) = qw( 10.0.1.0/24 10.0.2.0/24 );

@client_ipaddr_policy = (
    [ '0.0.0.0/8',  '127.0.0.1/8',  '[::]',  '[::1]' ]
                            =>  'LOCALHOST',
    [qw( !172.16.1.0/24 172.16.0.0/12 192.168.0.0/16 )]
                            =>  'MYPRIVATENETS',
    [qw( 192.0.2.0/25 192.0.2.129 192.0.2.130 )]
                            =>  'PARTNERS',
    \@some_nets             =>  'OTHER',
    \@mynetworks            =>  'MYNETS',
);
```

Policy banks – implicitly MYNETS

```
@mynetworks = qw(
   0.0.0.0/8 127.0.0.0/8 [::1]
   10.0.0.0/8 172.16.0.0/12 192.168.0.0/16
   192.0.2.0/24 [2001:db8::/32]
);
```

implicitly loads policy bank MYNETS
if it exists

Policy banks – by DKIM signature

```
@author_to_policy_bank_maps = (
{   'uni-bremen.de' => 'WHITELIST',
    'tu-graz.ac.at'     => 'WHITELIST',
    '.ebay.com'         => 'WHITELIST',
    '.paypal.com'       => 'WHITELIST',
    'amazon.com'        => 'WHITELIST',
    'cern.ch'           => 'SPECIAL',
    '.linkedin.com'     => 'MILD_WHITELIST',
    'dailyhoroscope@astrology.com'
                        => 'MILD_WHITELIST',
} );
```

Policy banks – by custom hook

```
sub new {
    my($class, $conn, $msginfo) = @_;
    my($self) = bless {}, $class;
    if ( ... ) {
        Amavis::load_policy_bank(
                                'NOVIRUSCHECK' );
    }
    $self;
}
```

Policy banks – Postfix side

```
# incoming mail MX
192.0.2.1:smtp inet  n  -  n  -  -  smtpd
   -o content_filter=amavisfeed:[127.0.0.1]:10040

# tcp port 587 for mail submission
submission inet  n  -  n  -  -  smtpd
   -o content_filter=amavisfeed:[127.0.0.1]:10042

# locally originating mail submitted on this host
pickup    fifo  n  -  n  60  1  pickup
   -o content_filter=amavisfeed:[127.0.0.1]:10043
```

Policy banks – Postfix side

content_filter = amavisfeed:[127.0.0.1]:10024

smtpd_sender_restrictions =
    check_client_access cidr:*/etc/postfix/nets.cidr*
    permit_mynetworks
    permit_sasl_authenticated
    check_sender_access pcre:*/etc/postfix/tag_as_inbound.pcre*

overrides global *content_filter* setting */etc/postfix/nets.cidr* :
127.0.0.0/8 FILTER amavisfeed:[127.0.0.1]:10026
10.0.0.0/8 FILTER amavisfeed:[127.0.0.1]:10026

/etc/postfix/tag_as_inbound.pcre :
/^/ FILTER amavisfeed:[127.0.0.1]:10024

Lookup tables

Static:

- associative array (Perl hash)
- list (a.k.a. ACL) (Perl list)
- list of regular expressions (list of objects)
- constant

Dynamic:

- SQL, LDAP

Lookup tables – associative array

```
(  'me.ac.uk'      => 1,
   '.ac.uk'        => 0,
   '.uk'           => 1  )
```

- unordered set, predefined search order
- can provide any value (not just boolean)

read_hash('/etc/mydomains-hash')

Lookup tables – list (ACL)

('me.ac.uk', '!.ac.uk', '.uk')

or:

qw(me.ac.uk !.ac.uk .uk)

- sequential search, first match wins
- can only provide booleans:
  exclamation mark: false

read_array('/etc/mydomains-list')

Lookup tables – regular expressions

```
new_RE(
   [ qr/ ^(noreply|offer) /i      => 0 ],
   [ qr/ [@.]example\.net$ /i  => 1  ],
     qr/ [@.]example\.net$ /i,  # shorthand 1
     qr/ [@.]example\.com$ /i,
)
```

- sequential list, first match wins
- can provide any value not just booleans
- default rhs is a boolean true

Lookup tables – constant

- trivial, always returns some constant (e.g. a string or a number) regardless of search key
- useful as a final catchall

Lookup tables – SQL

```
CREATE TABLE users (
  id                    SERIAL PRIMARY KEY,
  priority              integer,            -- 0 is low priority
  policy_id             integer  unsigned,
  email                 varchar(255),
  local                 char(1)
 );

CREATE TABLE policy (
  id                    SERIAL PRIMARY KEY,
  spam_lover            char(1),
  virus_quarantine_to   varchar(64),
     ...
);

SELECT  *,  users.id
 FROM users LEFT JOIN policy ON users.policy_id=policy.id
 WHERE users.email IN (?,?,?,...)
 ORDER BY users.priority DESC
```

Lists of lookup tables: @*_maps

- it became too awkward to have one variable for each type of a lookup table, and for each setting:

```
%local_domains          # a hash
@local_domains_acl      # a plain list
$local_domains_re       # regexp list
```

- solution:
  a list of lookup tables of arbitrary types

Lists of lookup tables: @*_maps

```
@local_domains_maps = (
   \%local_domains,
   \@local_domains_acl,
   \$local_domains_re,
);
```

actually: list of references to lookup tables

Lists of lookup tables: @*_maps

- program only consults these *@_*maps* variables, no longer the individual old settings like *%local_domains*
- specify directly or indirectly; SQL, LDAP

```
@local_domains_maps = (
    [...list1...], {...hash1...}, [...list2...],
    new_RE(...re1...), read_hash('/etc/myfile'),
    %local_domains, {...hash3...}, constant
);
```

Remember:

- policy banks affect message as a whole, so can only depend on some common characteristic of a message, e.g. client's IP address, sender address / DKIM, TCP port number

- lookups serve to implement per-recipient settings (and some other things)

Content categories

- CC_VIRUS
- CC_BANNED
- CC_UNCHECKED
- CC_SPAM above kill level
- CC_SPAMMY above tag2 level
- CC_BADH
- CC_OVERSIZED
- CC_MTA
- CC_CLEAN

Content categories

Content categories

```
%subject_tag_maps_by_ccat = (
  CC_VIRUS,      [ '***INFECTED*** ' ],
  CC_BANNED,  [ '***BANNED*** ' ],
  CC_UNCHECKED,
                 [$undecipherable_subject_tag],
  CC_SPAM,       undef,
  CC_SPAMMY,   \@spam_subject_tag2_maps,
  CC_CLEAN.',1',  \@spam_subject_tag_maps,
);
```

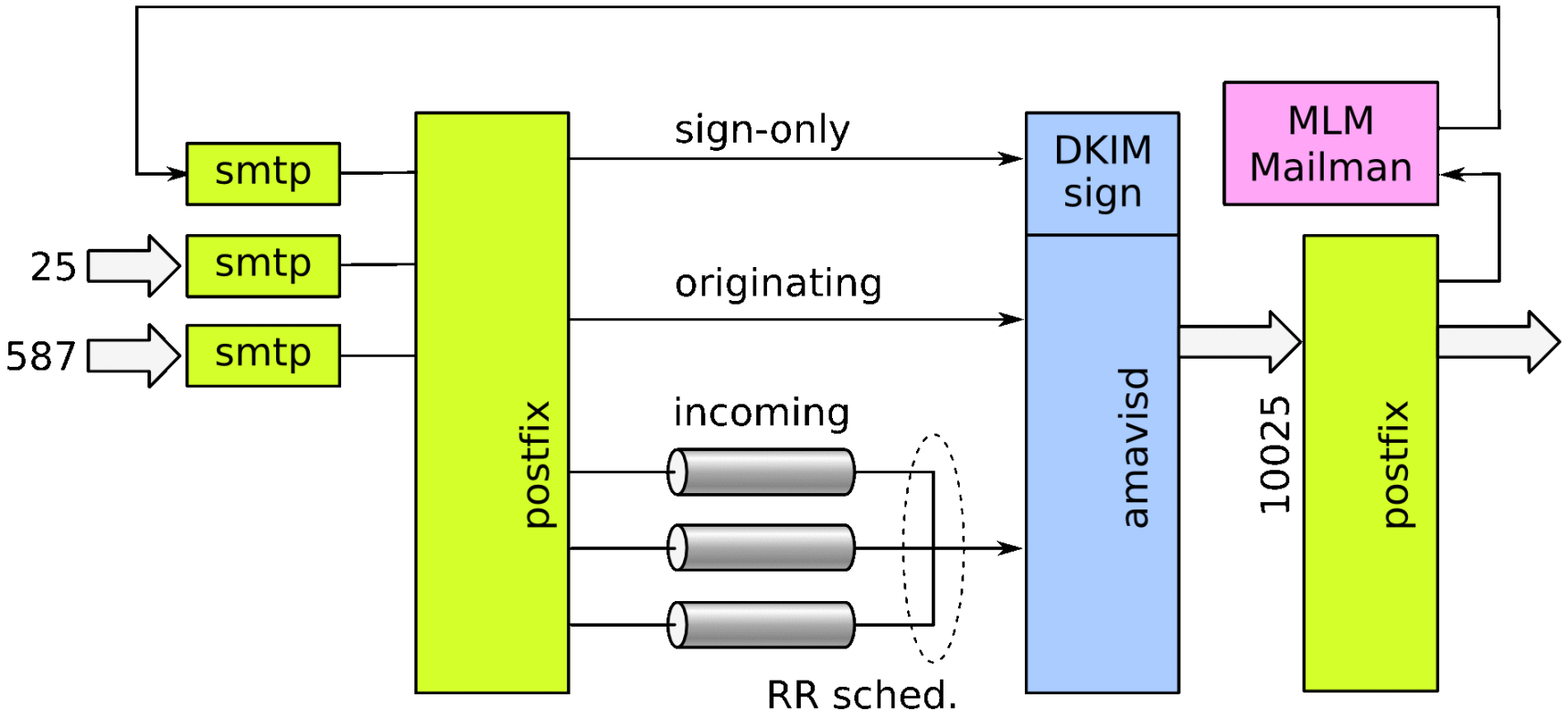
Fitting it all together

- like Lego® blocks, everything fits:
  uses SMTP as standard connectors

- Postfix:
  content_filter globally or as a cmd option
  FILTER in restrictions, *master.cf* entries

- amavisd:
  TCP ports, policy banks, *$forward_method*

Postfix & amavisd mail flow



Multiple transports – RR scheduling

```
amavisfeed-prio    unix  -  -  n  -  50  smtp
    -o smtp_connection_cache_on_demand=no
    -o smtp_send_xforward_command=yes

amavisfeed-norm  unix  -  -  n  -  50  smtp
    -o smtp_connection_cache_on_demand=no
    -o smtp_send_xforward_command=yes

amavisfeed-bulk    unix  -  -  n  -  50  smtp
    -o smtp_connection_cache_on_demand=no
    -o smtp_send_xforward_command=yes
```

Multiple transports – overbooking

amavisfeed-prio_destination_concurrency_limit = 45
amavisfeed-norm_destination_concurrency_limit = 45
amavisfeed-bulk_destination_concurrency_limit = 40

turn off *smtp_connection_cache_on_demand* when overbooking!

Multiple transports – choosing

content_filter =
    amavisfeed-norm:[127.0.0.1]:10024

smtpd_sender_restrictions =
    check_client_access
        cidr:*/etc/postfix/mynetworks.cidr*
    permit_mynetworks
    permit_sasl_authenticated
    check_sender_access
        pcre:*/etc/postfix/tag_bysender.pcre*

Multiple transports – by client IP adr

/etc/postfix/mynetworks.cidr

127.0.0.0/8	FILTER amavisfeed-prio:[127.0.0.1]:10026
169.254.0.0/16	FILTER amavisfeed-prio:[127.0.0.1]:10026
10.0.0.0/8	FILTER amavisfeed-prio:[127.0.0.1]:10026
172.16.0.0/12	FILTER amavisfeed-prio:[127.0.0.1]:10026
192.168.0.0/16	FILTER amavisfeed-prio:[127.0.0.1]:10026
::1/128	FILTER amavisfeed-prio:[127.0.0.1]:10026
fe80::/10	FILTER amavisfeed-prio:[127.0.0.1]:10026
2001:1470:ff80::/48	FILTER amavisfeed-prio:[127.0.0.1]:10026
2001:1470:ff81::/48	FILTER amavisfeed-prio:[127.0.0.1]:10026

Multiple transports – by sender

/etc/postfix/tag_bysender.pcre

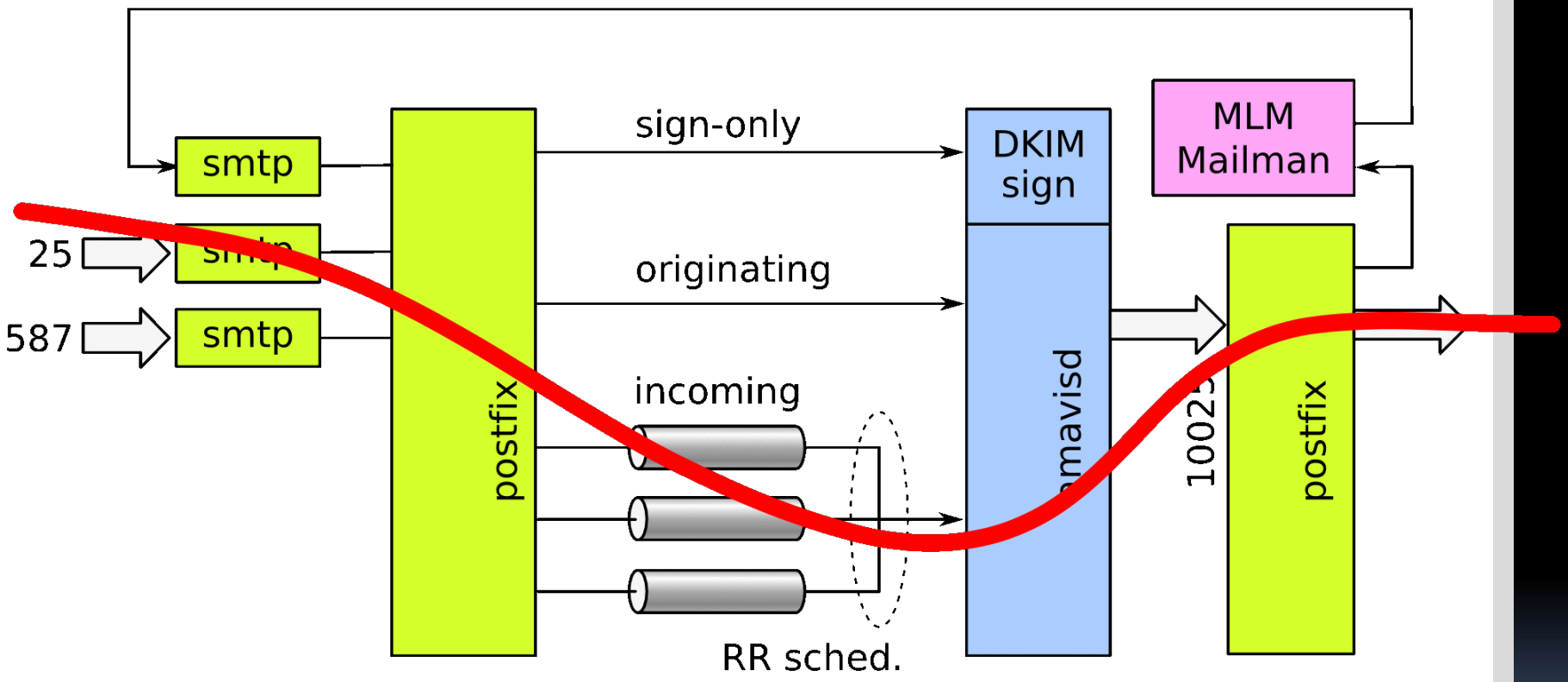
```
/[@.](apache|postfix|gnome|wikipedia)\.org$/
        FILTER amavisfeed-norm:[127.0.0.1]:10024
/[@.](sun|ibm|amd|cisco|elsevier)\.com$/
        FILTER amavisfeed-norm:[127.0.0.1]:10024
/[@.]cern\.ch$/
        FILTER amavisfeed-norm:[127.0.0.1]:10024
/^<>$/  FILTER amavisfeed-bounce:[127.0.0.1]:10024
/^/     FILTER amavisfeed-bulk:[127.0.0.1]:10024
```

Multiple transports – choosing (2)

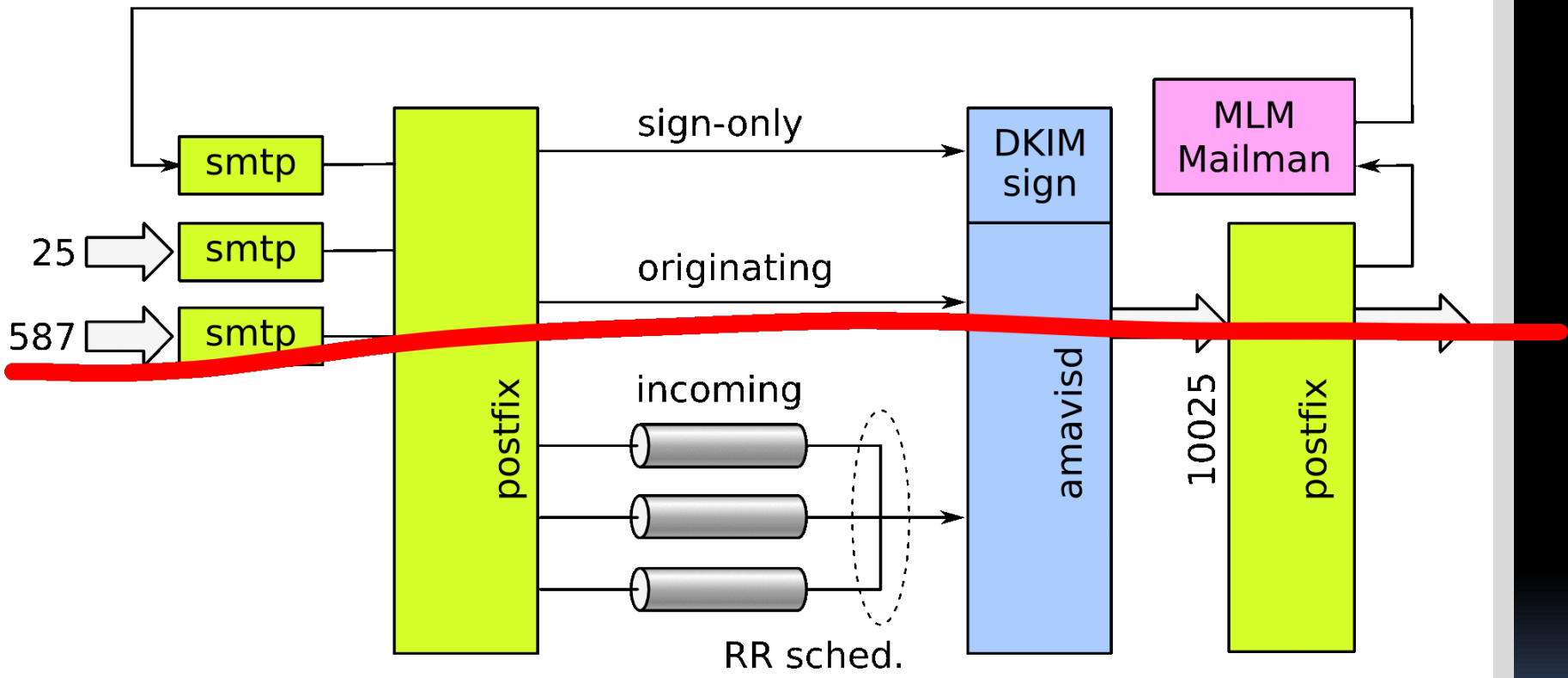
```
content_filter =
    amavisfeed-norm:[127.0.0.1]:10024

smtpd_sender_restrictions =
    check_client_access
        cidr:/etc/postfix/mynetworks.cidr
    permit_mynetworks
    permit_sasl_authenticated
    check_sender_access
        pcre:/etc/postfix/tag_bysender.pcre
```

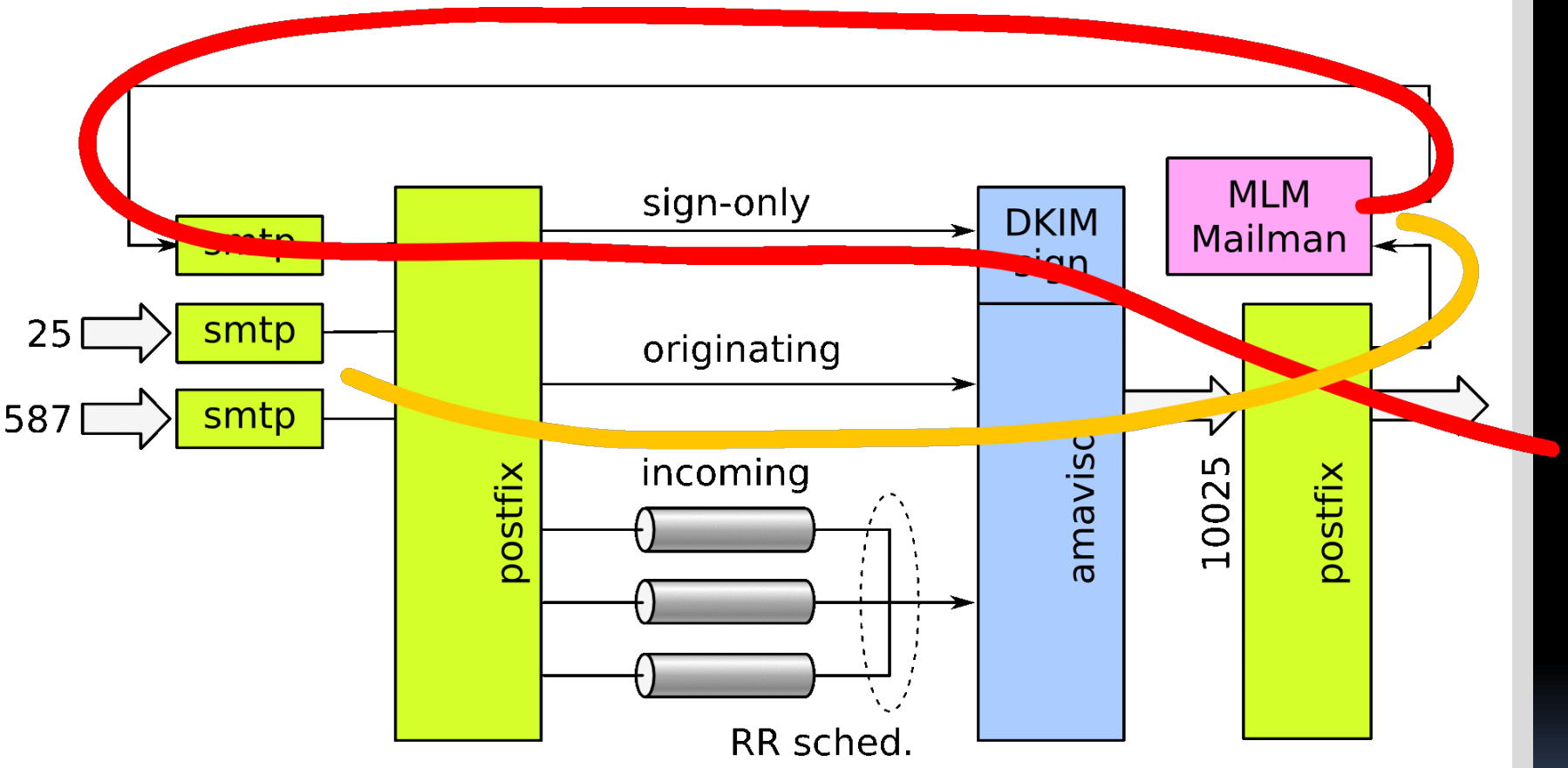
Inbound mail, no signing, just verify



Originating, DKIM signing



Mailing list integration, re-sign



Monitoring health: amavisd-nanny

```
PID 28039: 28039-02       0:00:05 GSSSr
PID 28048: .              0:00:05 .....
PID 28174: 28174-01-10    0:00:02 VS
PID 28309: A              0:00:00
```

- db key: PID
- db data: timestamp of last event, status

- status:
  - empty - idle child process
  - A - just accepted a connection (post_accept_hook)
  - am_id - processing am_id task
  - . - content checking done

$ amavisd-nanny -h

```
States legend:
  A   accepted a connection
  b   begin with a protocol for accepting a request
  m   'MAIL FROM' smtp command started a new
      transaction in the same session
  d   transferring data from MTA to amavisd
  =   content checking just started
  G   generating and verifying unique mail_id
  D   decoding of mail parts
  V   virus scanning
  S   spam scanning
  P   pen pals database lookup and updates
  r   preparing results
  Q   quarantining and preparing/sending notifications
  F   forwarding mail to MTA
  .   content checking just finished
  sp  space indicates idle (elapsed bar showing dots)
```

Monitoring health: amavisd-nanny normal

```
PID 27948: 27948-02-4     0:00:02 SF
PID 27987:                0:00:05 .....
PID 28039: 28039-02       0:00:05 DVSSS
PID 28048: .              0:00:05 .....
PID 28101: 28101-01-9     0:00:01 =
PID 28174: 28174-01-10    0:00:02 dV
PID 28187: 28187-01-5     0:00:12 VVSSSSSSSS:SS
PID 28245: 28245-01-4     0:00:07 GVSSSSS
PID 28309: A              0:00:00
```

Monitoring health: amavisd-nanny mostly idle

```
PID 28187: 28187-02-8    0:00:02 SS
PID 28245:               0:01:16 ........:........>
PID 28309:               0:01:16 ........:........>
PID 28543: 28543-01-7    0:00:03 VSS
PID 28584: 28584-01-7    0:00:01 S
PID 28672:               0:00:24 ........:........
PID 28677:               0:01:06 ........:........>
PID 28678:               0:01:06 ........:........>
PID 28729:               0:00:56 ........:........>
```

Monitoring health: amavisd-nanny trouble - crashed programs

```
PID 25408: 25408-01  went away  0:02:27 =========:==>
PID 25496: 25496-01  went away  0:01:58 =========:==>
PID 25728: 25728-01  went away  0:02:06 =========:==>
```

- process no longer exists, but is still registered in db
- mail stays in MTA queue (temporary failure)

usual reasons:

- bug in a library routine such as uulib, zlib, bdb
- resources exceeded: *Lock table is out of available locker entries*, stack size, runaway regexp in custom rules

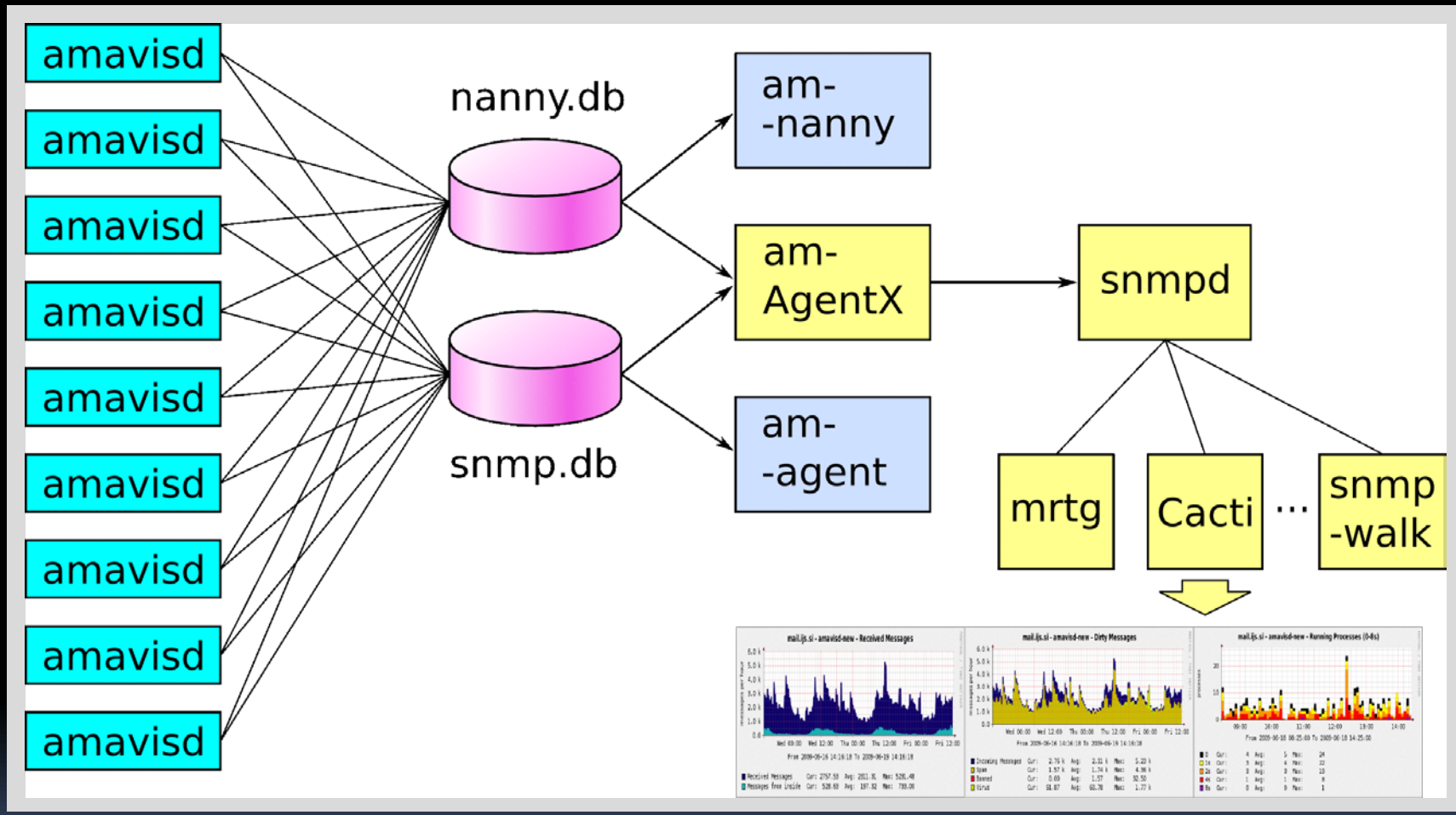
Monitoring health: amavisd-nanny trouble - looping or forgotten proc.

```
PID 25733: 25733-01  terminated 2:10:56 =========:=>
```

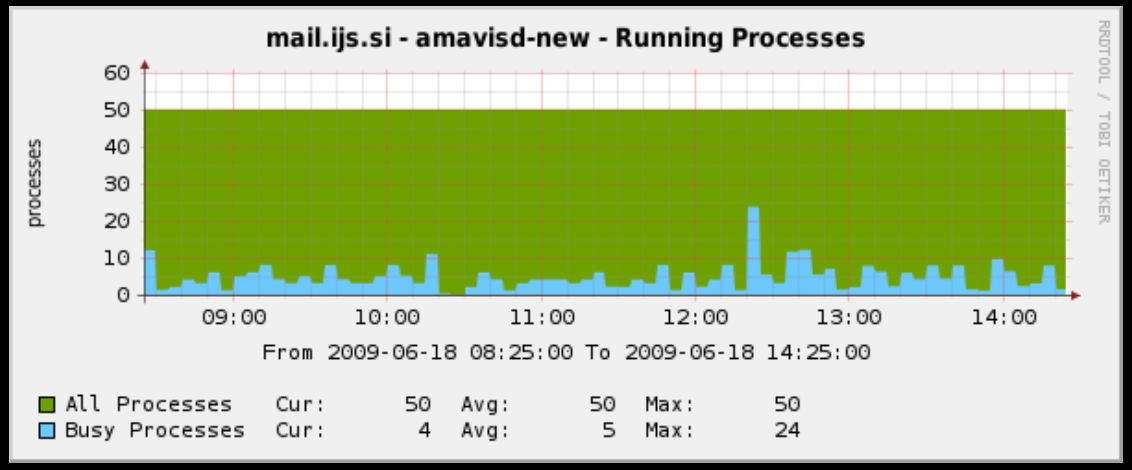
- amavisd-nanny sends SIGTERM first
- amavisd-nanny sends SIGKILL 30 seconds later if necessary

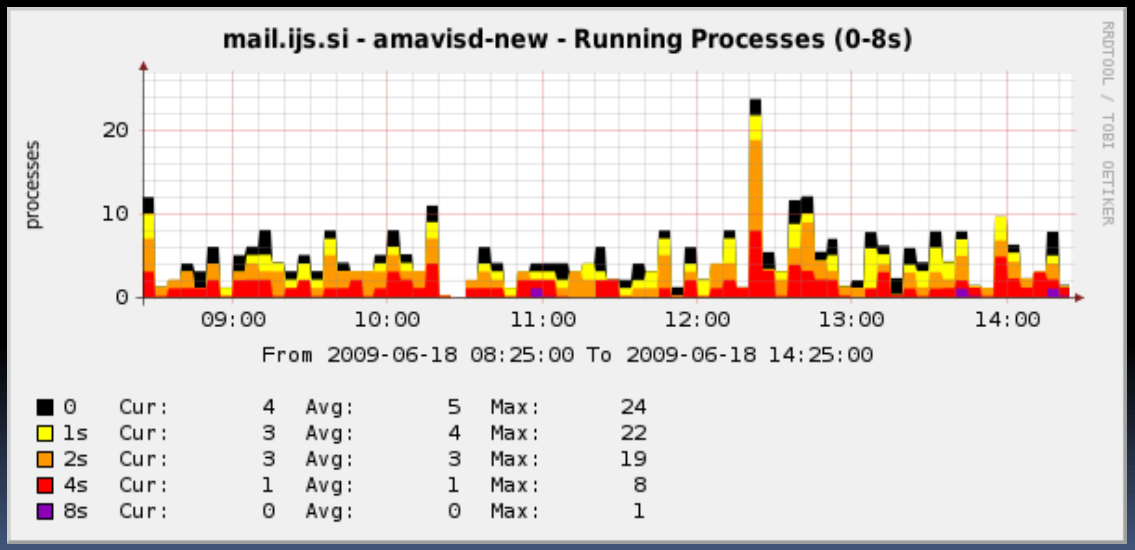
- active ttl = 10 minutes stuck active children
- idle ttl = 1 hour unused idle process (may be normal)

Monitoring – components



SNMP: load, timing





Statistics: amavisd-agent

```
sysUpTime              (0 days, 14:03:43.46)

InMsgs                 14490   1030/h  100.0 % (InMsgs)
InMsgsRecips           27169   1932/h  187.5 % (InMsgs)

ContentCleanMsgs        6020    428/h   41.5 % (InMsgs)
ContentSpamMsgs         7807    555/h   53.9 % (InMsgs)
ContentVirusMsgs         567     40/h    3.9 % (InMsgs)

ContentBadHdrMsgs         91      6/h    0.6 % (InMsgs)
ContentBannedMsgs          5      0/h    0.0 % (InMsgs)
```

Statistics: amavisd-agent

```
OpsSpamCheck          12719     904/h    87.8 % (InMsgs)
OpsVirusCheck         13231     941/h    91.3 % (InMsgs)
OpsSqlSelect          50680    3604/h   186.5 % (InMsgsRc)

OutMsgs                6248     444/h   100.0 % (OutMsgs)
OutMsgsDelivers        6248     444/h   100.0 % (OutMsgs)

OutForwMsgs            6155     438/h    98.5 % (OutMsgs)

OutDsnMsgs               35       2/h     0.6 % (OutMsgs)
OutDsnBannedMsgs          3       0/h     0.0 % (OutMsgs)
OutDsnSpamMsgs           32       2/h     0.5 % (OutMsgs)
```

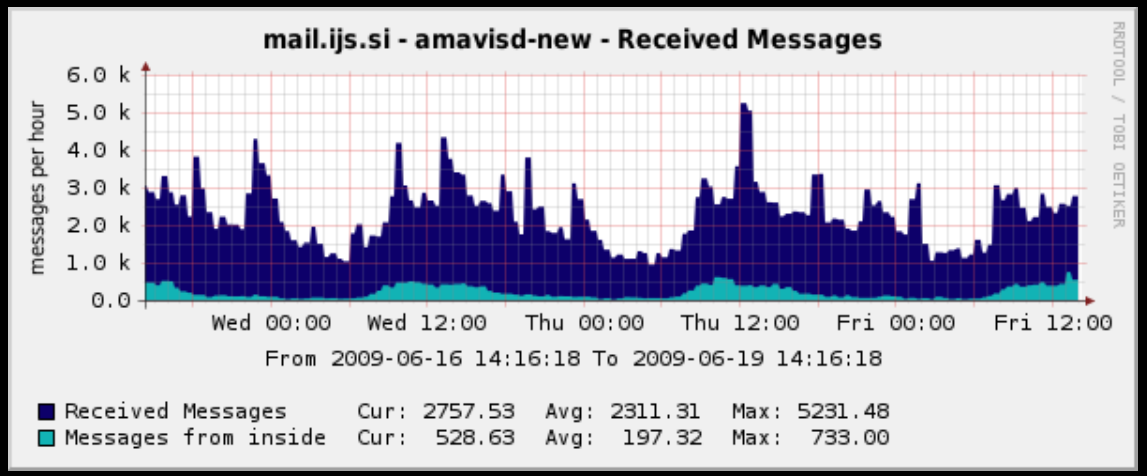
Statistics: amavisd-agent

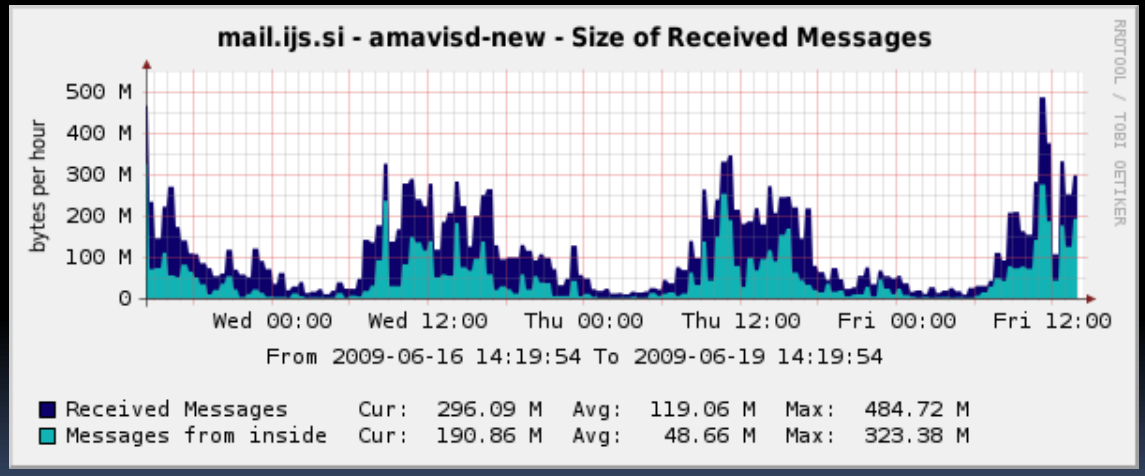
```
QuarMsgs              2704    192/h  100.0 % (QuarMsgs)
QuarSpamMsgs          2100    149/h   77.7 % (QuarMsgs)
QuarVirusMsgs          567     40/h   21.0 % (QuarMsgs)
QuarBannedMsgs           5      0/h    0.2 % (QuarMsgs)
QuarOther               32      2/h    1.2 % (QuarMsgs)
```

Statistics: amavisd-agent

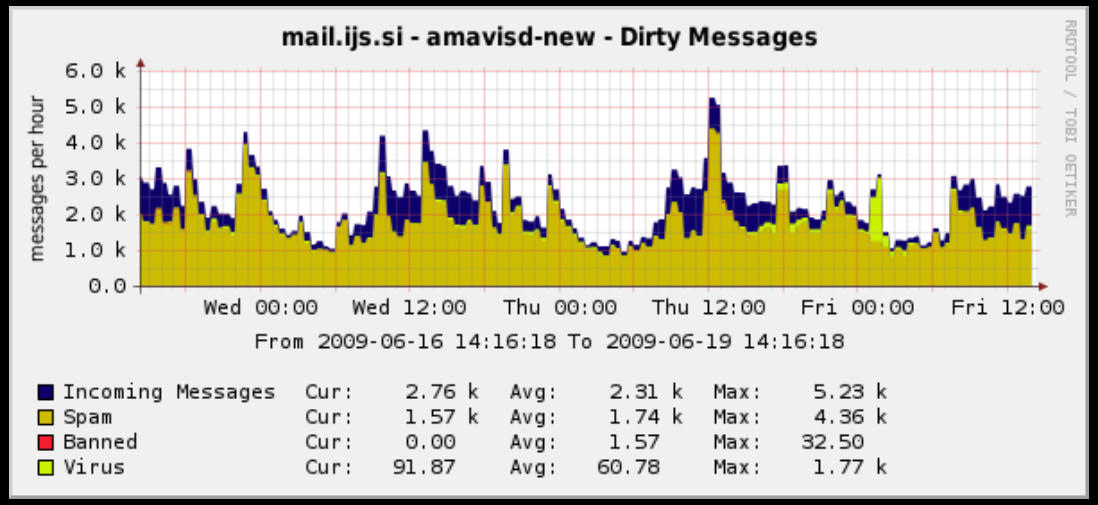
W32/Netsky-P	191	14/h
W32/Mytob-CA	59	4/h
W32/Netsky-D	25	2/h
W32/Lovgate-V	21	1/h
W32/Netsky-Q	21	1/h
W32/Bagle-AG	17	1/h
HTML.Phishing.Pay-1	18	1/h
HTML.Phishing.Bank-1	12	1/h
W32/Mytob-Z	11	1/h
W32/Wurmark-J	11	1/h
W32/Lovgate-X	11	1/h

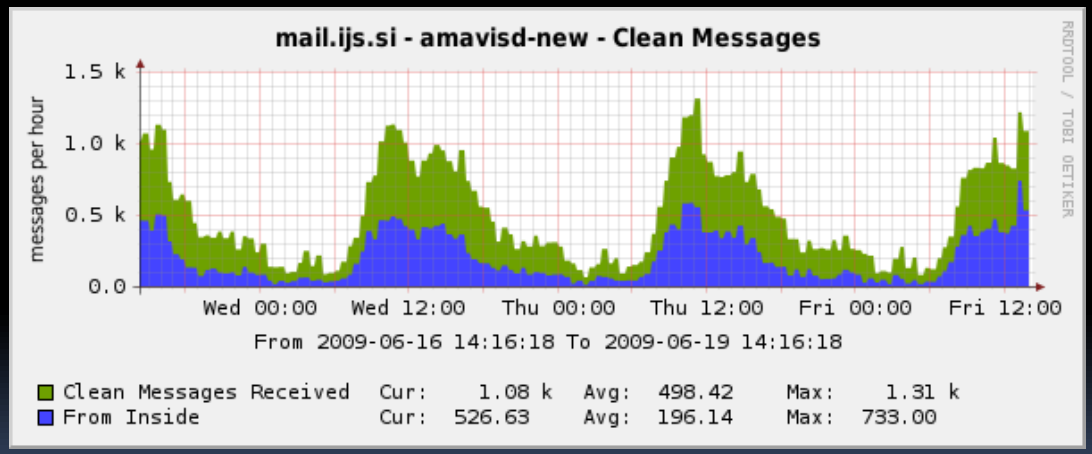
SNMP: mail rate, size



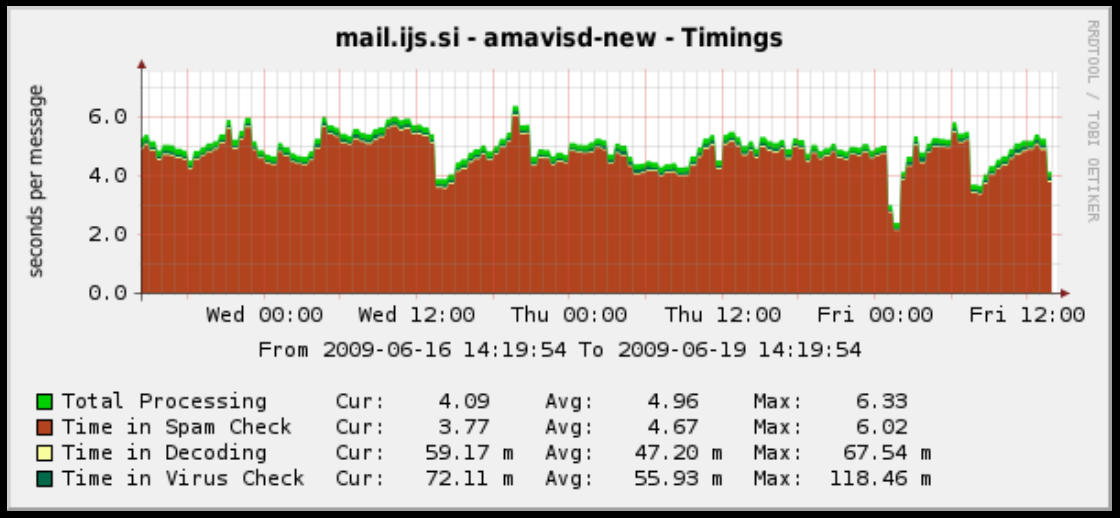


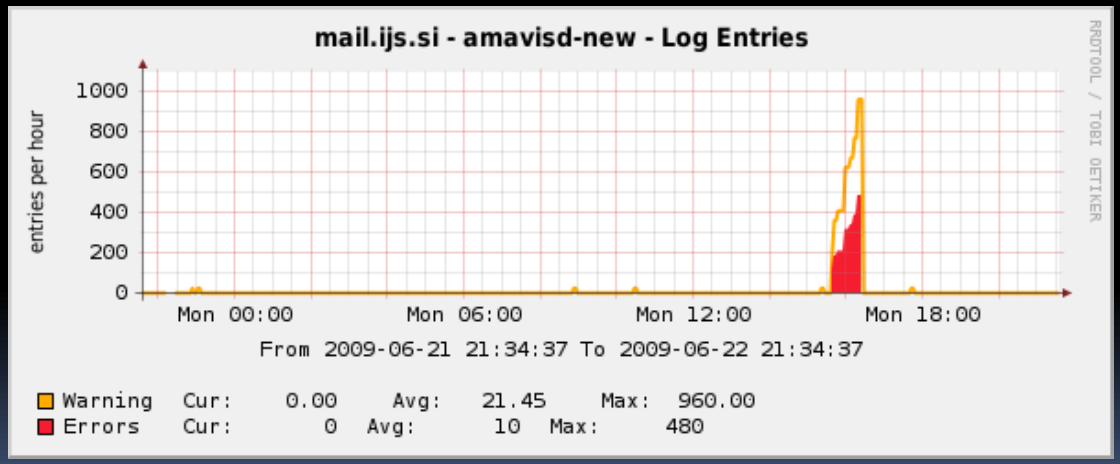
SNMP: mail content



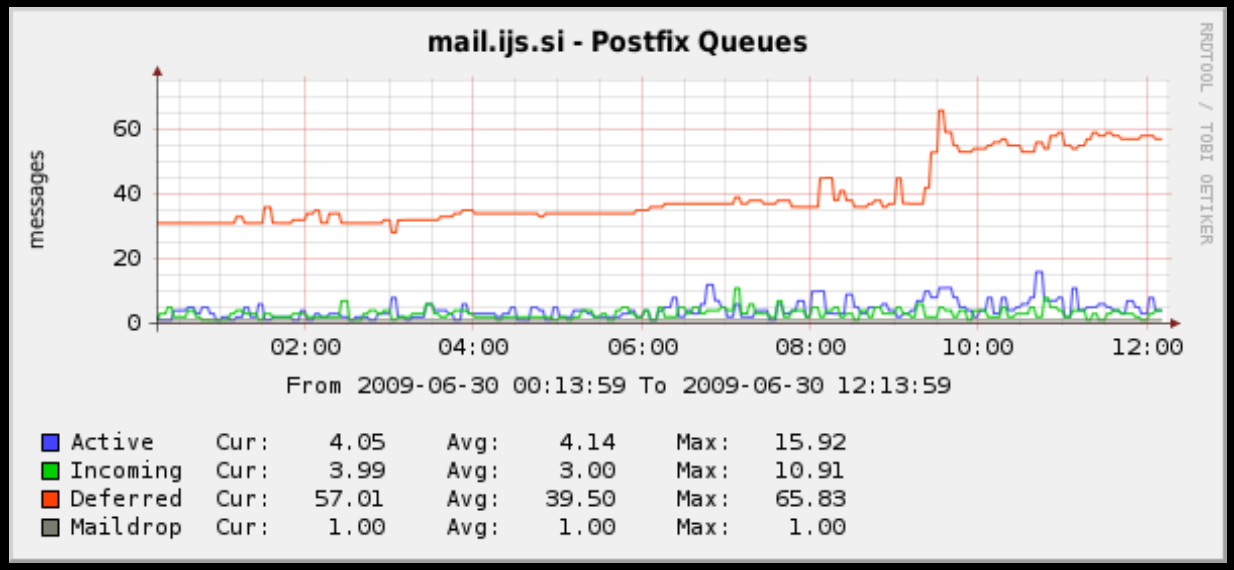


SNMP: elapsed time, errors





SNMP: Postfix queue entries



Details in the log: timing report

TIMING [total 1725 ms] –
lookup_sql: 6 (0%)0,
SMTP pre-DATA-flush: 1 (0%)0, SMTP DATA: 88 (5%)6,
body_hash: 1 (0%)6, sql-enter: 4 (0%)6,
mime_decode: 6 (0%)6, get-file-type1: 23 (1%)7,
parts_decode: 0 (0%)8,
AV-scan-1: 7 (0%)8, AV-scan-2: 4 (0%)8, AV-scan-3: 5 (0%)8,
AV-scan-4: 1 (0%)9, AV-scan-5: 1 (0%)9, AV-scan-6: 0 (0%)9,
lookup_sql: 4 (0%)9, spam-wb-list: 3 (0%)9,
SA msg read: 0 (0%)9, SA parse: 2 (0%)9,
SA check: 1536 (89%)98,
update_cache: 2 (0%)98, post-do_spam: 6 (0%)99,
deal_with_mail_size: 0 (0%)99, main_log_entry: 18 (1%)100,
sql-update: 4 (0%)100, update_snmp: 1 (0%)100,
unlink-1-files: 1 (0%)100, rundown: 0 (0%)100

Details in the log: SpamAssassin 3.3 timing

TIMING-SA total 3491 ms –

parse: 1.67 (0.0%), extract_message_metadata: 6 (0.2%), get_uri_detail_list: 0.49 (0.0%), tests_pri_-1000: 13 (0.4%), tests_pri_-950: 0.73 (0.0%), tests_pri_-900: 0.87 (0.0%), tests_pri_-400: 16 (0.5%), check_bayes: 15 (0.4%), tests_pri_0: 3106 (89.0%), check_dkim_adsp: 2 (0.1%), check_spf: 5 (0.2%), poll_dns_idle: 0.25 (0.0%), check_razor2: 1759 (50.4%), check_dcc: 1268 (36.3%), tests_pri_500: 7 (0.2%), tests_pri_899: 77 (2.2%), check_crm114: 76 (2.2%), tests_pri_1000: 11 (0.3%), total_awl: 10 (0.3%), check_awl: 3 (0.1%), update_awl: 2 (0.1%), learn: 226 (6.5%), crm114_autolearn: 201 (5.7%), get_report: 1.15 (0.0%)

speaking of which ...

ANNOUNCING:

Apache SpamAssassin 3.3.0-alpha 1
is now available for testing (2009-07-03)

Downloads from:

http://people.apache.org/~jm/devel/

taking the opportunity:

SpamAssassin 3.3.0 alpha 2009-07-03

- rules separate from code (sa-update)
- DKIM reworked, ADSP overrides
- SA plugin DKIM cooperates with amavisd
- DomainKeys plugin removed
- TIMING-SA reports in amavisd log
- compile rules in smaller chunks
- compatible with SA 3.2.5
- in production use at several sites

troubleshooting

- amavisd-nanny
- amavisd log and MTA log
- increase log level if necessary
- search log for am_id of a trouble message

- *strace -f amavisd foreground*

troubleshooting

- # *amavisd debug*
- # *amavisd debug-sa*
- # *amavisd foreground*
- selective debug: *@debug_sender_maps*
- selective debug: dedicated policy bank with elevated log
- compare output of *'amavisd debug-sa'*
  to *'su vscan -c spamassassin –t -D'*

Regular maintenance tasks

- run *amavisd-nanny* or SNMP, note any *'process went away'* reports, investigate and fix the problem if any
- check *mailq* or *qshape* for stalled messages
- check for preserved directories in */var/amavis/tmp*, search log for explanation, fix the problem and delete
- remove old quarantine and SQL logs

Questions?

- mailing list
- hang around and ask
- …


SpamAssassin
message
amavisd
mail
header
DKIM
spam
address
new
policy
field
maps
file
default
value
fields
sql
list
reported
signature
score
partition
signing
tag
key
may
name
LOG
one
smtp
id
now
Also
command
release
lookup
bounce
Email
use
number
logging
AV
MTA
options
section
true
RFC
match
NOTE
amavisd.conf
ccat
module
level
banks
program
report
currently
line
bank
process
according
i.e
checks
CC
generated
variable
two
virus
signed
longer
following
sa
verification
previous
string
SNMP
add
bounces
previously
messages
body
BUG
banned
allow
error
problem
addresses
internal
scanner
based
DSN
instead
whitelist
version
method
option
entry
request
allows
configuration
keys
argument
e.g
even
updated
available
originating
using
suggested
names
results
See
information
setting
time
added
category
recipients
table
rules
change
additional
domains
tables
Perl
set
versions
files
code
data
quarantine
tls
type
domain
contents
status
sender
just
used
client
database
macro
Like
attribute
scanners
example
recipient
limit
provided
provide