MTA Performance Comparison:
sendmail vs. postfix on *BSD

Brad Knowles
Senior Consultant for Snow, BV
brad.knowles@snow.nl
http://www.shub-internet.org/brad/papers/mtacomparison/
Overview

• Goal
• Meta Information
  – Hardware Used
  – Software Tested
  – Tools Used
  – Methodology
• Test Results
• Conclusions
Goal

• Show you what it looks like to do MTA performance tuning
  – On as many *BSD platforms as I could

• Focus on the process, not the numbers or expected/desired outcome
  – You get to see (most) everything, warts and all
Goal

• Basically, chapter 3 from the book *sendmail Performance Tuning* by Nick Christenson
Meta Information

• Hardware Used
• Software Tested
• Benchmark Tools
• Methodology
Hardware Used

- NetBSD
  - Twinhead “Twinstation 5G” (Sun SPARCstation 5 clone)
    - Not UltraSPARC, but the ancient SPARC 5
- OS: NetBSD 1.6-RELEASE
- CPU: microsSPARC-II @ 110MHz
- RAM: 32MB real, 384MB virtual
- NIC: On-board “Lance” 10Base-T Ethernet & SBus QuadFastEthernet

- Unfortunately, disk drives died before testing could be performed (bug in NetBSD regarding SCSI tagged command queueing for old drives?)
Hardware Used

• FreeBSD
  – Compaq Armada 4131T
    • OS: FreeBSD 4.6.2-RELEASE
    • CPU: Pentium 133
    • RAM: 48MB real, 384MB virtual
    • NIC: Asanté FriendlyNET AL1011 “Prism2” 802.11b WiFi PCMCIA
    • HD: 10GB IBM Travelstar 20GN
      – 4200 RPM
      – 12ms avg. seek
Hardware Used:

FreeBSD
Hardware Used

• MacOS X (Part 1)
  – PowerBook G3 “Pismo”
    • OS: MacOS X 10.2.1
    • CPU: PowerPC G3 @ 400MHz
    • RAM: 1GB real, 2GB virtual
    • NIC: Apple AirPort 802.11b WiFi
    • HD: 48GB IBM Travelstar 40GH
      – 5400 RPM
      – 12ms avg. seek
Hardware Used:
 MacOS X (Part 1)

2002-11-16
Copyright © 2002 by Brad Knowles
Hardware Used

• MacOS X (Part 2)
  – PowerBook G4 (DVI)
    • OS: MacOS X 10.2.2
    • CPU: PowerPC G4 @ 800MHz
    • RAM: 1GB real, 2GB virtual
    • NIC: Apple AirPort 802.11b WiFi
    • HD: 40GB Toshiba MK4018GAS
      – 4200 RPM
      – 12ms avg. seek
Hardware Used:
MacOS X (Part 2)
Software Tested

• Sendmail 8.12.6

• Postfix 1.1.11
  – http://www.postfix.org/ftp-sites.html

• Exim 4.10
  – Tried every way I could think of, but could not get it configured to relay back to an IP address or port on the same machine
Tools Used

• Disk Benchmarking
  – PostMark from Network Appliance
    • http://www.netapp.com/tech_library/3022.html
  – Bonnie++ from Russell Coker

• SMTP Benchmarking
  – smtp-source/smtp-sink from Wietse Venema
    • Included with postfix distribution
  – Postal from Russell Coker (with smtp-sink)
Methodology

• Get disk subsystem baseline by performing benchmarking
  – For each platform & filesystem configuration
    • PostMark
    • Bonnie++
      – Configured to test similar parameters
        » File sizes, numbers, & subdirectories according to the four PostMark tests
Methodology

• For FreeBSD, we only test turning on or off “soft updates”
  – We do not test (turned on by default)
    • vfs.vmioidirenable
    • dirprefs
    • dirhash
  – For data on all the possible combinations of the above, see “Recent Filesystem Optimisations in FreeBSD” at http://www.usenix.org/events/usenix02/tech/freenix/dowse.html
Methodology

- For MacOS X
  - We test only
    - HFS+
      - Apple’s Extended Hierarchical File System
    - MFS
      - Memory-based filesystem created using RamBunctious 2.0
Methodology

• For MacOS X
  – We do **NOT** test
    • UFS
      – IMO, Apple broke it while porting HFS+ to Mach
      – Many applications do not recognize it, in addition to MacOS 9 itself
      – “Classic” requires that boot device be HFS+ and MacOS 9 must be installed before MacOS X
    • HFS+ with Journaling
      – Too new, introduced with MacOS X Server 10.2.2
      – Initial reports indicate ~20% performance loss
Methodology

• For MacOS X
  – Does not have a native “poll.h” or “libpoll” (required by postal)
    • Install manually from source http://www.clapper.org/software/poll/
    • Install using “fink”
      – MacOS X package/port manager based on Debian “dselect”
  – Unable to test Bonnie++
    • Need further assistance from the author to find out why it is crashing
Methodology

• Perform SMTP benchmarking
  – For each hardware & software configuration, test
    • Test direct source->sink connection
      – Demonstrate maximum possible performance
    • Test source->relay->sink
      – Show relay performance

  – Note
    • Testing only relaying, not local delivery
    • Test generated on, relayed through, and terminated on “loopback” network
      – Eliminate all question of network or NIC performance
Test Results

• Chart data only
  – Bonnie++ table data not yet available
  – smtp-source/smtp-sink data not yet available
    • `hostname` must be legal (i.e., does not end with “." )
  – Postal data not yet available
Test Results

• Dumb “Driver” Scripts
Test Results:
PostMark Sample Run

% cd /var/tmp
% postmark
PostMark v1.5 : 3/27/01
pm>set number 1000
pm>set transactions 50000
pm>show
Current configuration is:
The base number of files is 1000
Transactions: 50000
Files range between 500 bytes and 9.77 kilobytes in size
Working directory: /var/tmp
Block sizes are: read=512 bytes, write=512 bytes
Biases are: read/append=5, create/delete=5
Using Unix buffered file I/O
Random number generator seed is 42
Report format is verbose.
pm>run
Test Results:
PostMark Sample Run

Creating files...Done
Performing transactions........Done
Deleting files...Done
Time:
  497 seconds total
  492 seconds of transactions (101 per second)

Files:
  26014 created (52 per second)
    Creation alone: 1000 files (500 per second)
    Mixed with transactions: 25014 files (50 per second)
  24868 read (50 per second)
  24880 appended (50 per second)
  26014 deleted (52 per second)
    Deletion alone: 1028 files (342 per second)
    Mixed with transactions: 24986 files (50 per second)

Data:
  161.15 megabytes read (332.02 kilobytes per second)
  168.38 megabytes written (346.92 kilobytes per second)
PostMark Test Results

Postmark Test #1: 1000 Files & 50k Transactions

2002-11-16

Copyright © 2002 by Brad Knowles
PostMark Test Results

![Postmark Test #2: 20k Files & 50k Transactions](image)

2002-11-16

Copyright © 2002 by Brad Knowles
PostMark Test Results

Postmark Test #3: 20k Files & 100k Transactions

2002-11-16
Copyright © 2002 by Brad Knowles
PostMark Test Results

Postmark Test #4: 20k Files, 100k Transactions, & 100 subdirectories

2002-11-16  Copyright © 2002 by Brad Knowles
PostMark Test Results

![Postmark Test #1: 1000 Files & 50k Transactions](image)

2002-11-16  Copyright © 2002 by Brad Knowles
PostMark Test Results

Postmark Test #2: 20k Files & 50k Transactions
PostMark Test Results

Postmark Test #3: 20k Files & 100k Transactions

2002-11-16  Copyright © 2002 by Brad Knowles
PostMark Test Results

Postmark Test #4: 20k Files, 100k Transactions, & 100 subdirectories

2002-11-16 Copyright © 2002 by Brad Knowles
Test Results:
Bonnie++ Sample Run

% bonnie++ -d /var/tmp/bonnie -n 1:10k:0 -m compaq-soft -r 48m
Writing with putc()...done
Writing intelligently...done
Rewriting...done
Reading with getc()...done
Reading intelligently...done
start 'em...done...done...done...
Create files in sequential order...done.
Stat files in sequential order...done.
Delete files in sequential order...done.
Create files in random order...done.
Stat files in random order...done.
Delete files in random order...done.
Delete files in random order...done.

Version 1.02d

<table>
<thead>
<tr>
<th>Machine</th>
<th>Size</th>
<th>K/sec</th>
<th>%CP</th>
<th>K/sec</th>
<th>%CP</th>
<th>K/sec</th>
<th>%CP</th>
<th>K/sec</th>
<th>%CP</th>
<th>K/sec</th>
<th>%CP</th>
<th>K/sec</th>
<th>%CP</th>
</tr>
</thead>
<tbody>
<tr>
<td>compaq-soft</td>
<td>300M</td>
<td>2407</td>
<td>67</td>
<td>4584</td>
<td>39</td>
<td>2052</td>
<td>21</td>
<td>1948</td>
<td>53</td>
<td>3644</td>
<td>18</td>
<td>57.8</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

files:max
| compaq-soft,300M,2407,67,4584,39,2052,21,1948,53,3644,18,57.8,3,1:10:0,402,31,+++++,++++,832,49,564,46,++++,+++++,1746,99 |
Bonnie++ Results

Bonnie++ Block I/O (Sequential Create)

2002-11-16

Copyright © 2002 by Brad Knowles
Bonnie++ Results

Bonnie++ Block I/O (Sequential Read)

The diagram shows the performance results of Bonnie++ Block I/O (Sequential Read) for different Platform-Filesystem/Test configurations. The y-axis represents the number of files read per second, while the x-axis shows the different test configurations.

- **compaq-ufs/20:10:0** with 704 files read per second and 43% CPU busy.
- **compaq-ufs/20:10:0:100** with 65 files read per second and 66% CPU busy.
- **compaq-soft/20:10:0** with 679 files read per second and 43% CPU busy.
- **compaq-soft/20:10:0:100** with 695 files read per second and 44% CPU busy.

The graph highlights the performance differences across these configurations, with some showing higher file reads per second but lower CPU usage, and vice versa.
Bonnie++ Results

Bonnie++ Block I/O (Sequential Delete)
Bonnie++ Results

Bonnie++ Block I/O (Random Read)

Files Read/Sec

% CPU Busy

Platform-Filesystem/Test

2002-11-16
Copyright © 2002 by Brad Knowles
Bonnie++ Results

Bonnie++ Block I/O (Random Delete)

Files Deleted/Sec

% CPU Busy

Platform-Filesystem/Test

2002-11-16

Copyright © 2002 by Brad Knowles
Test Results:
SmtpStone Sample Run

% sudo smtp-sink 127.0.0.1:25 1024 &
% foreach I (1 2 3 4 5 6 7 8 9 10)
foreach? /usr/bin/time smtp-source -l 10240 -f fred@example.net \
-t george@example.net -m 1000 -s 512 127.0.0.1:25
foreach? end

8.06 real 0.22 user 1.65 sys
9.03 real 0.26 user 1.45 sys
smtp-source: fatal: connect: Operation timed out
32.00 real 0.17 user 1.68 sys
13.08 real 0.19 user 1.43 sys
10.98 real 0.31 user 1.28 sys
9.02 real 0.24 user 1.52 sys
8.04 real 0.25 user 1.57 sys
smtp-source: fatal: lost connection while reading server greeting
16.51 real 0.19 user 1.61 sys
10.99 real 0.19 user 1.53 sys
16.52 real 0.25 user 1.50 sys
SsmtpStone Results

PowerBook G4 SsmtpStone Performance (1000 messages)
SmtpStone Results

PowerBook G4 SmtpStone Performance (20000 messages)
SmtpStone Results

Compaq SmtpStone Performance (1000 Messages)

![Graph showing SmtpStone performance with number of messages per second and standard deviation for different numbers of threads. The graph includes bars for 4, 2, and 1 threads, with values 28.39, 16.62, and 9.00 respectively, and a line indicating standard deviation.](image-url)
Test Results:
Postal Sample Run

% sudo smtp-sink 127.0.0.1:25 1024 &
% postal -m 10k -p 64 -c 1 127.0.0.1 postal.user -
time,messages,data(K),errors,connections,SSL connections
23:05,7651,41262,0,7715,0
23:06,7795,41656,0,7795,0
23:07,7847,42305,0,7847,0
23:08,7616,41528,0,7616,0
23:09,7670,41659,0,7670,0
23:10,7413,39922,0,7413,0
23:11,7765,41886,0,7765,0
23:12,7752,41960,0,7752,0
23:13,7423,39835,0,7423,0
23:14,7475,39697,0,7475,0
23:15,7458,39981,0,7458,0
23:16,7462,40332,0,7462,0
23:17,7281,39294,0,7280,0
23:18,7408,40022,0,7409,0
23:19,7595,40882,0,7586,0
23:20,7737,41991,0,7746,0
23:21,7646,41122,0,7646,0

^C
Test Results:
Postal Sample Run

% sudo smtp-sink 127.0.0.1:25 1024 &
% postal -m 10k -p 128 -c 1 127.0.0.1 postal.user -
time,messages,data(K),errors,connections,SSL connections

<table>
<thead>
<tr>
<th>Time</th>
<th>Time (s)</th>
<th>Messages</th>
<th>Reader</th>
<th>Reader</th>
<th>Reader</th>
</tr>
</thead>
<tbody>
<tr>
<td>23:27</td>
<td>8283</td>
<td>43988</td>
<td>0</td>
<td>8411</td>
<td>0</td>
</tr>
<tr>
<td>23:28</td>
<td>8965</td>
<td>48165</td>
<td>0</td>
<td>8965</td>
<td>0</td>
</tr>
<tr>
<td>23:29</td>
<td>8952</td>
<td>48100</td>
<td>0</td>
<td>8952</td>
<td>0</td>
</tr>
<tr>
<td>23:30</td>
<td>8785</td>
<td>46803</td>
<td>0</td>
<td>8785</td>
<td>0</td>
</tr>
<tr>
<td>23:31</td>
<td>7948</td>
<td>42811</td>
<td>0</td>
<td>7933</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Server timed out on read.
Server timed out on read.

[ ... Deletia ... ]

Server timed out on read.
Server timed out on read.

<table>
<thead>
<tr>
<th>Time</th>
<th>Time (s)</th>
<th>Messages</th>
<th>Reader</th>
<th>Reader</th>
<th>Reader</th>
</tr>
</thead>
<tbody>
<tr>
<td>23:32</td>
<td>8016</td>
<td>43367</td>
<td>28</td>
<td>8059</td>
<td>0</td>
</tr>
<tr>
<td>23:33</td>
<td>8960</td>
<td>47748</td>
<td>0</td>
<td>8960</td>
<td>0</td>
</tr>
<tr>
<td>23:34</td>
<td>9054</td>
<td>48507</td>
<td>0</td>
<td>9022</td>
<td>0</td>
</tr>
<tr>
<td>23:35</td>
<td>8737</td>
<td>47175</td>
<td>0</td>
<td>8769</td>
<td>0</td>
</tr>
<tr>
<td>23:36</td>
<td>9057</td>
<td>48586</td>
<td>0</td>
<td>9057</td>
<td>0</td>
</tr>
<tr>
<td>23:37</td>
<td>8800</td>
<td>46722</td>
<td>0</td>
<td>8800</td>
<td>0</td>
</tr>
<tr>
<td>23:38</td>
<td>8933</td>
<td>48092</td>
<td>0</td>
<td>8933</td>
<td>0</td>
</tr>
<tr>
<td>23:39</td>
<td>8969</td>
<td>48155</td>
<td>0</td>
<td>8955</td>
<td>0</td>
</tr>
<tr>
<td>23:40</td>
<td>8884</td>
<td>48576</td>
<td>0</td>
<td>8998</td>
<td>0</td>
</tr>
</tbody>
</table>

°C
Test Results:
Postal Sample Run

% sudo smtp-sink 127.0.0.1:25 1024 &
% postal -m 10k -p 256 -c 1 127.0.0.1 postal.user -
Usage: postal [-m maximum-message-size] [-p processes] [-l local-address]
       [-c messages-per-connection] [-r messages-per-minute] [-a]
       [-b [no]netscape] [-[z|Z] debug-file]
       [-s ssl-percentage]
       smtp-server user-list-filename conversion-filename

Postal Version: 0.61

% postal -m 10k -p 32 -c 1 127.0.0.1 postal.user -
time,messages,data(K),errors,connections,SSL connections
Can't connect to 127.0.0.1 port 25.
Can't connect to 127.0.0.1 port 25.
Can't connect to 127.0.0.1 port 25.
Can't connect to 127.0.0.1 port 25.
Can't connect to 127.0.0.1 port 25.
Postal Results

PowerBook G4 Postal Performance

2002-11-16  Copyright © 2002 by Brad Knowles  47
Notes for Testing

• When testing real MTAs, you need to watch more than just the output from the testing program
  – Not unusual for MTAs to accept mail faster than they can process it, especially under heavy load

• Also watch
  – Mail queue size
    • mailq | grep '[0-9][0-9][0-9][0-9]:[0-9][0-9]' | wc -l
  – CPU load
    • iostat 5
  – Memory utilization
    • vmstat 5
Notes for Postfix

• Make sure that you modify /etc/postfix/main.cf to be suitable
  – Need to boost
    • local_destination_concurrency_limit
    • default_destination_concurrency_limit

• Check /etc/postfix/master.cf to make sure there are no artificial limits on processes like smtp, smtpd, bounce, cleanup, rewrite etc...
Notes for MacOS X

• Unfortunately, Apple broke iostat
  – Still mostly okay for what we want to watch
    (CPU user %, system %, and idle %)
• Apple doesn’t even provide vmstat
  – vm_stat is not the same program,
    although it is similar
Postfix SsmtpStone Results

PowerBook G4 Postfix SsmtpStone Performance (1000 messages)

2002-11-16 Copyright © 2002 by Brad Knowles
Postfix SmpStone Results

Compaq+UFS SmpStone Performance (1000 messages)

Messages per Second

Number of Threads

2002-11-16

Copyright © 2002 by Brad Knowles
Postfix SmtpStone Results

Compaq+Soft SmtpStone Performance (1000 messages)

Number of Threads

Messages per Second

Standard Deviation
Postfix SsmtpStone Results

Compaq Postfix SsmtpStone Results (20000 messages/4 Threads)

2002-11-16

Copyright © 2002 by Brad Knowles
Sendmail SsmtpStone Results

PowerBook G4 Sendmail SsmtpStone Results (1000 messages)

Message per Second vs Number of Threads

- Messages/Sec
- Std. Dev.
Sendmail SmtpStone Results

PowerBook G4 Sendmail+Async SmtpStone Results (1000 messages)

2002-11-16 Copyright © 2002 by Brad Knowles
Sendmail SmtpStone Results

Compaq Sendmail-UFS SmtpStone Results (1000 messages)

![Graph showing Sendmail SmtpStone Results](image)

2002-11-16  Copyright © 2002 by Brad Knowles
Sendmail SsmtpStone Results

Compaq Sendmail+Soft SsmtpStone Results (1000 messages)

2002-11-16  Copyright © 2002 by Brad Knowles
Performance Tuning: Syslog

• Problem
  – Local syslog can cause excessive load on busy mail system

• Solutions
  – Configure syslog to write to logs asynchronously
    • Perhaps via a “–” (dash) before the filename
  – Run syslog on a central log server, across the network
    • Also allows you to perform more in-depth log processing, without interfering with mail services
    • Generally logs are much easier to manage centrally
    • Can also combine with other central syslog processing applications
      – E.g., Addamark Log Management System
Performance Tuning: Syslog

• Problem
  – Classic syslog uses UDP, can lose 75% or more of packets sent across the network

• Solution
  – Replace classic syslog with nsyslog, ssyslog, or syslog-ng and use TCP instead of UDP
    • Depending on your particular requirements for hardware, OS, environment, etc....
Postfix SmttpStone Results

PowerBook G4 (no syslog) Postfix SmttpStone (1000 messages)

2002-11-16

Copyright © 2002 by Brad Knowles
Sendmail SmtpStone Results

PowerBook G4 Sendmail+Async+NoSyslog SmtpStone Results
(1000 messages)

Number of Threads

Messages per Second

Standard Deviation

2002-11-16

Copyright © 2002 by Brad Knowles
Sendmail SmtpStone Results

Compaq Sendmail-UFS-NoSyslog SmtpStone Results
(1000 messages)

2002-11-16

Copyright © 2002 by Brad Knowles
Conclusions

• This process can be difficult and time-consuming to get right, but it pays off
• Graphs & charts can help a great deal
  – But they don’t tell the whole story
• Software
  – Postfix is a lot faster than even I had thought
  – Wasn’t able to get sendmail tuned for the performance I’m confident it can deliver
  – Benchmarking tools (esp. Bonnie++ and postal) need more work to be useful
Conclusions

• Use the Scientific Method
  – Try to set aside personal bias
  – Trust only what you can prove
  – Process
    • Generate Hypothesis
    • Generate test for Hypothesis
    • Test Hypothesis
      – Prove or Disprove
    • If disproven, modify or generate new Hypothesis
Related Work

• Nick Christenson
  – sendmail Performance Tuning book
    http://www.jetcafe.org/~npc/book/sendmail/
  – “Performance Tuning Sendmail Systems” slides

• Brad Knowles
  – “Sendmail Performance Tuning for Large Systems” slides
Related Work

• Matthias Andree
  http://www-dt.e-technik.uni-dortmund.de/~ma/postfix/vsqmail.html
  http://www-dt.e-technik.uni-dortmund.de/~ma/postfix/bench2.html

• Nakamura Motonori
  http://www.kyoto.wide.ad.jp/mta/eval1/eindex.html

• Ciprian Ascante