Performance optimisation of webmail

Katerina Mparmpopoulou Periklis Stefopoulos

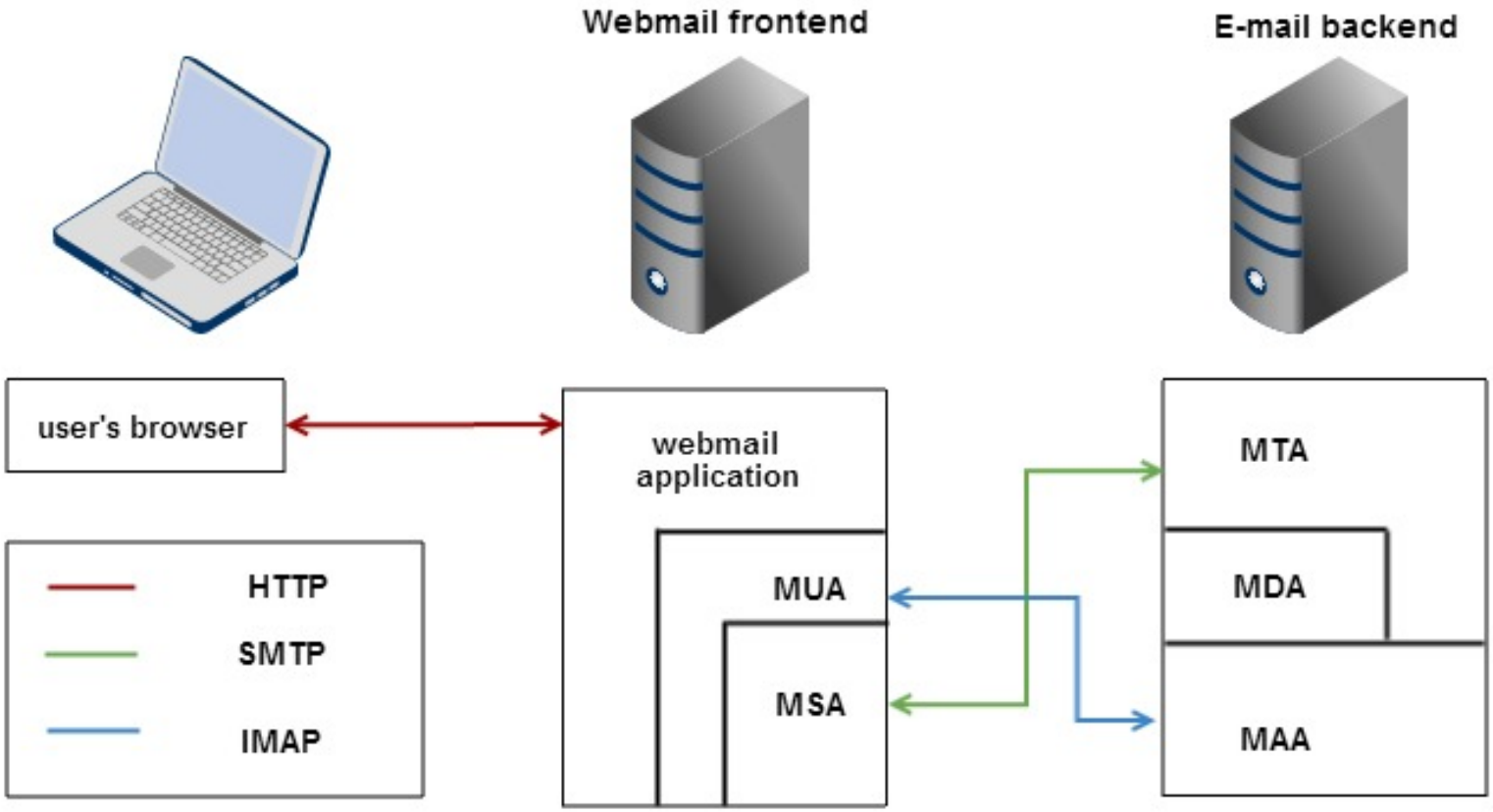
Supervised by: *Michiel Leenaars*

University of Amsterdam
System & Network Engineering

July 3, 2013

Outline

Webmail System Components



E-mail Components	Examples
Mail User Agent (MUA)	Afterlogic
Mail Submission Agent (MSA)	Afterlogic
Mail Transfer Agent (MTA)	Postfix
Mail Delivery Agent (MDA)	Postdrop
Mail Access Agent (MAA)	Cyrus

Research purpose

- Hundreds of millions of end users depend on Webmail technologies
- Open source web frontends to mail servers are an often neglected area of development
- Better understanding the performance of web mail applications is a prerequisite to better tuning these applications

Research Question

What are the bottlenecks, in terms of performance, of current Webmail implementations and which could be the most optimal solution?

Experimental Environment

Webmail Frontends

- Squirrelmail
- Roundcube
- Horde IMP
- Afterlogic Webmail Lite

Webmail Backends

- Courier - Postfix - Amavis - ClamAv - SpamAssassin
- Dovecot - Postfix - Amavis - ClamAv - SpamAssassin
- Cyrus - Postfix - Amavis - ClamAv - SpamAssassin

Experiments

5 users with different mailbox size

- 1500 messages with only text
- 1500 messages with attachments plus text
- 3000 messages with only text
- 4500 messages with only text
- 6000 messages with only text

3 different user actions

- Log in to Webmail
- Searching a keyword from the "Subject" field
- Searching a keyword from the entire message content

Metrics

Metric	Extraction method
Latency	tcpdump/Wireshark
CPU time [1]	systat/sar
Unique Set Size (USS)	smem
Proportional Set Size (PSS)	smem

Table: Benchmark metrics and their extraction method

[1] $CPU\ time = CPU\ utilization * elapsed\ time * number\ of\ CPUs$

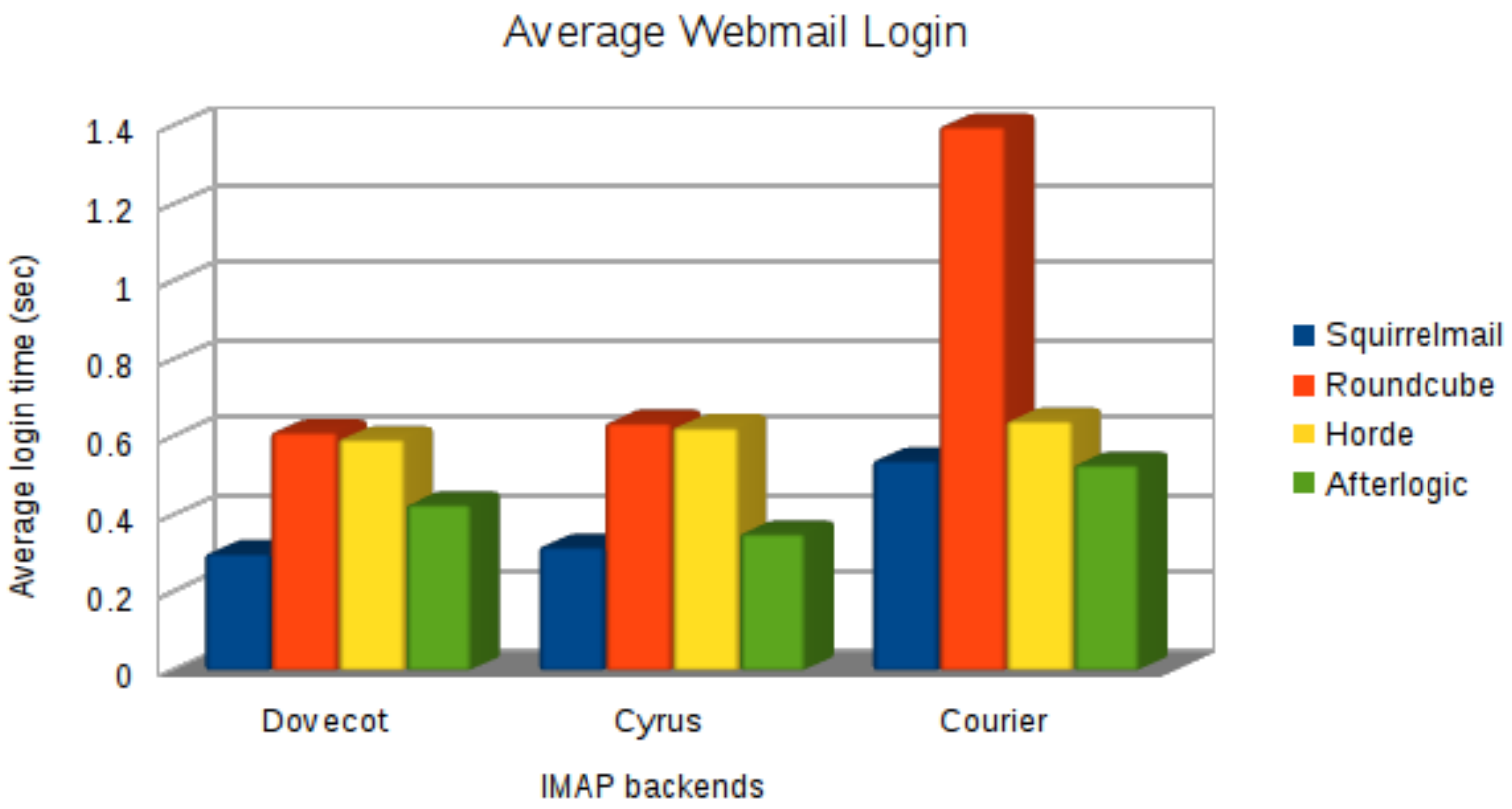
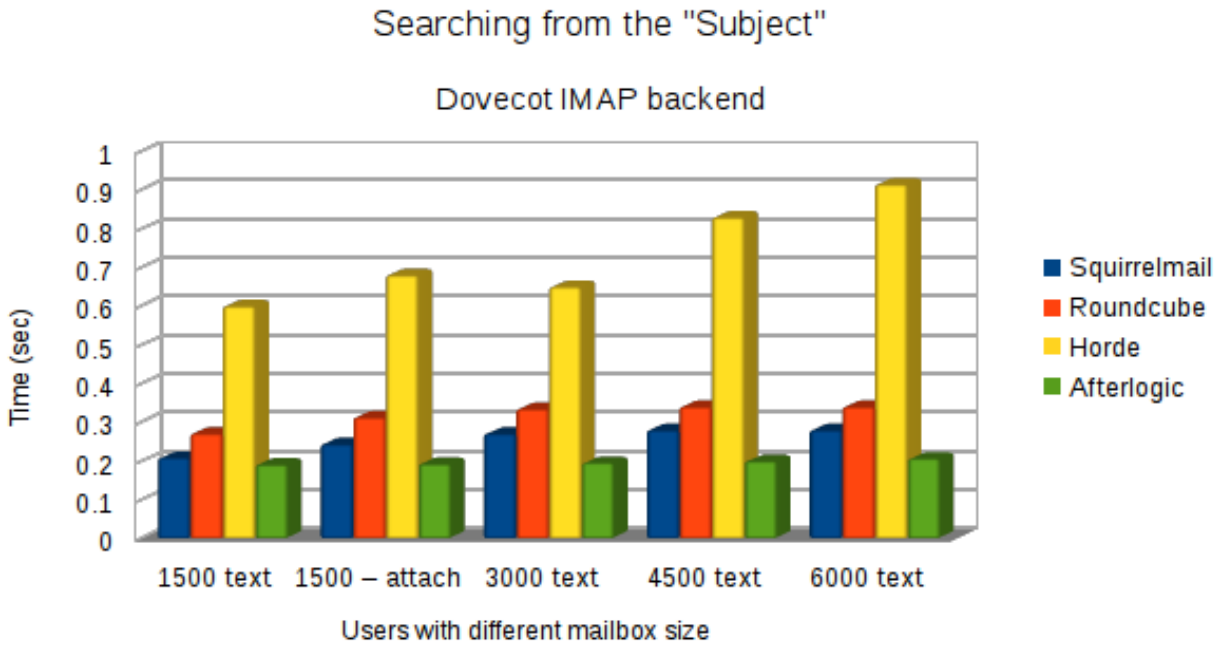
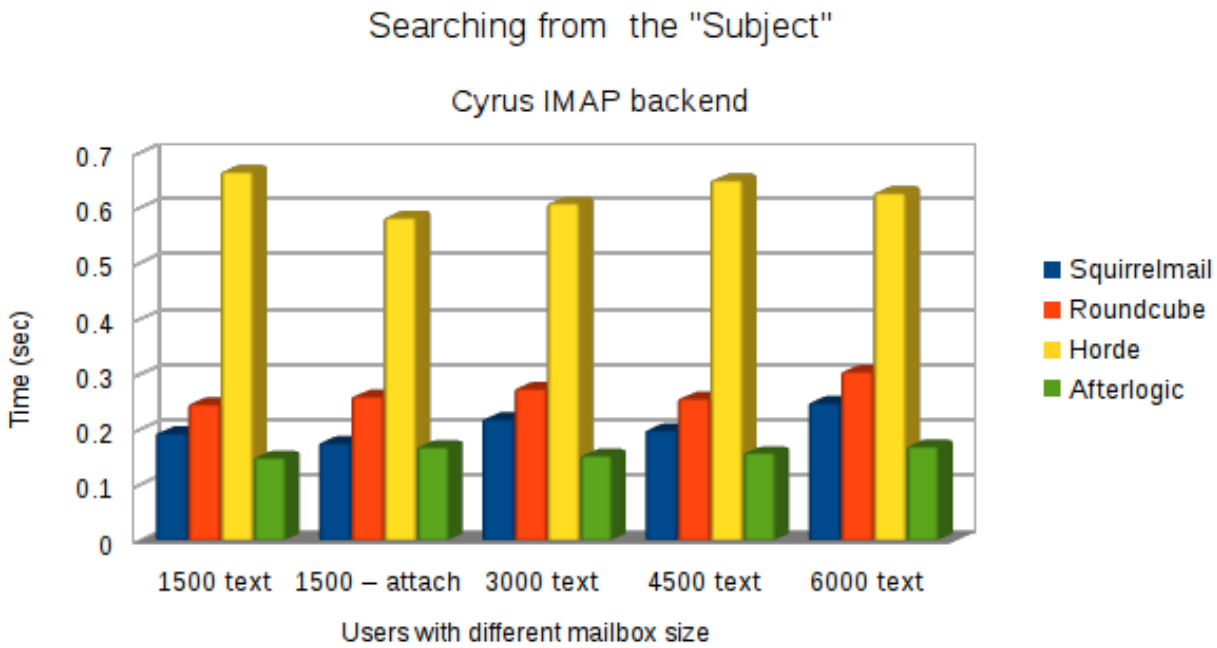
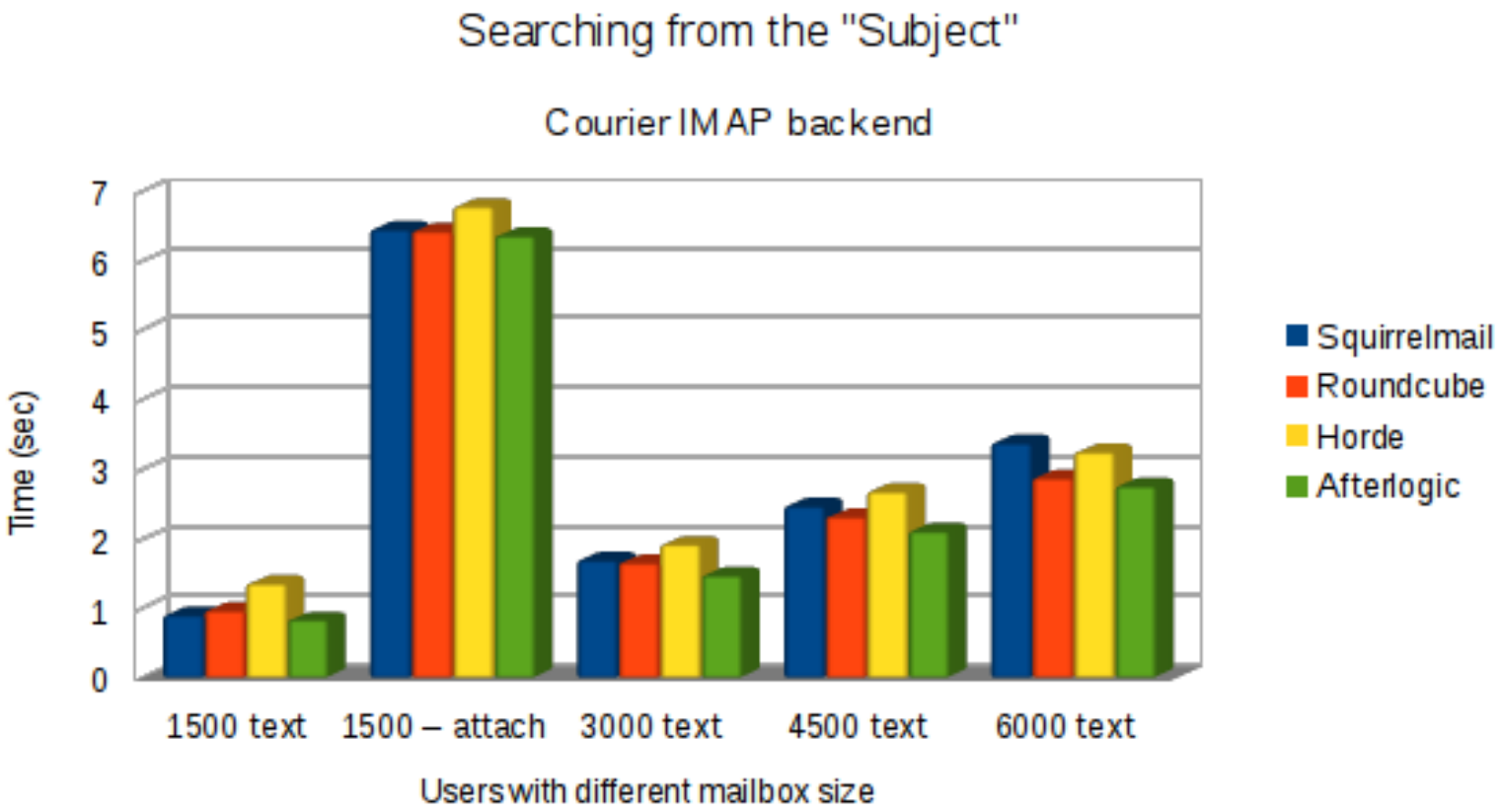


Figure: average fetching time during login


Searching from the "Subject"
Dovecot IMAP backend
Time (sec)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1500 text
1500 – attach
3000 text
4500 text
6000 text
Users with different mailbox size
Squirrelmail
Roundcube
Horde
Afterlogic


Searching from the "Subject"
Cyrus IMAP backend
Time (sec)
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1500 text
1500 – attach
3000 text
4500 text
6000 text
Users with different mailbox size
Squirrelmail
Roundcube
Horde
Afterlogic


Searching from the "Subject"
Courier IMAP backend
Time (sec)
0
1
2
3
4
5
6
7
1500 text
1500 – attach
3000 text
4500 text
6000 text
Users with different mailbox size
Squirrelmail
Roundcube
Horde
Afterlogic

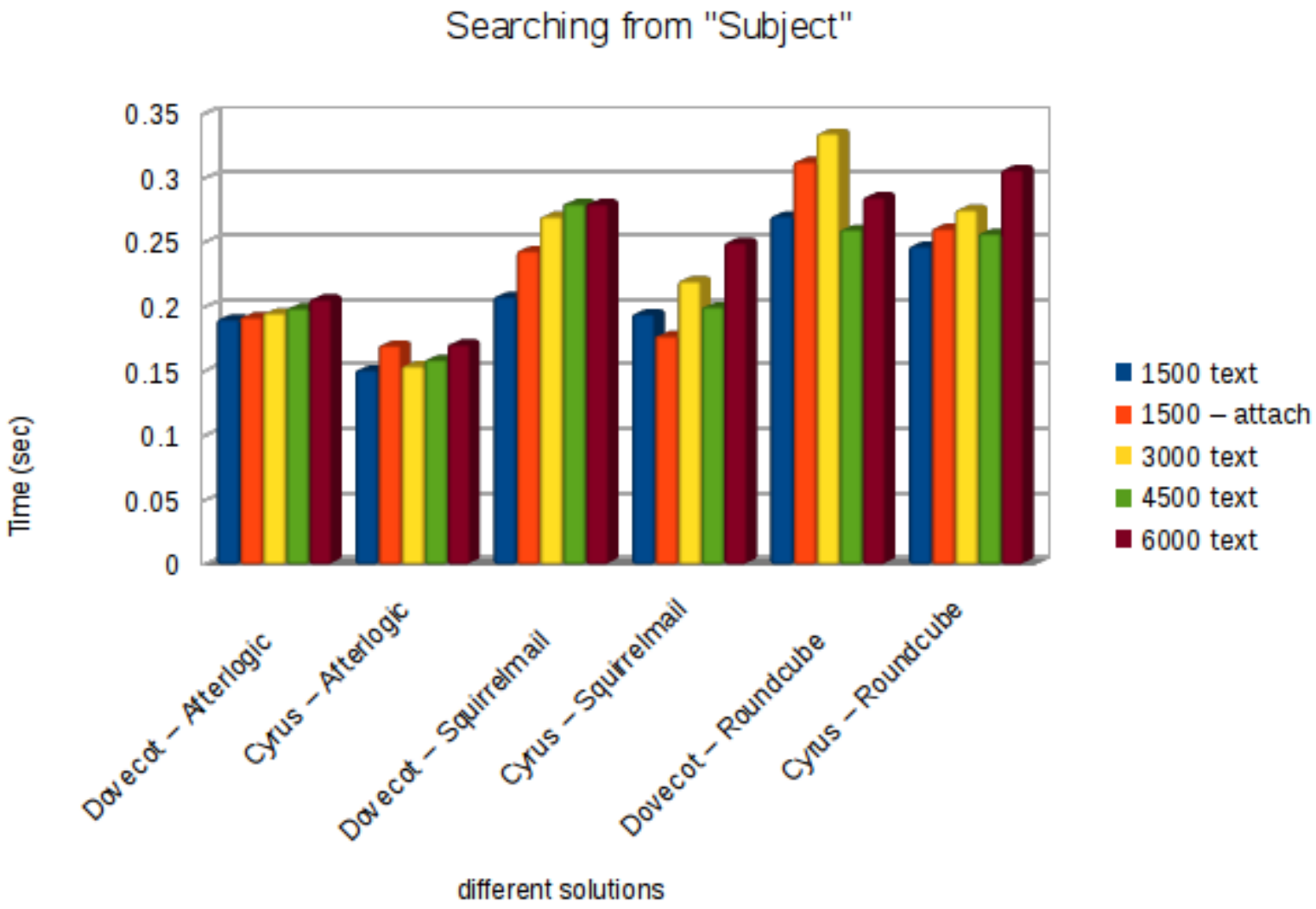
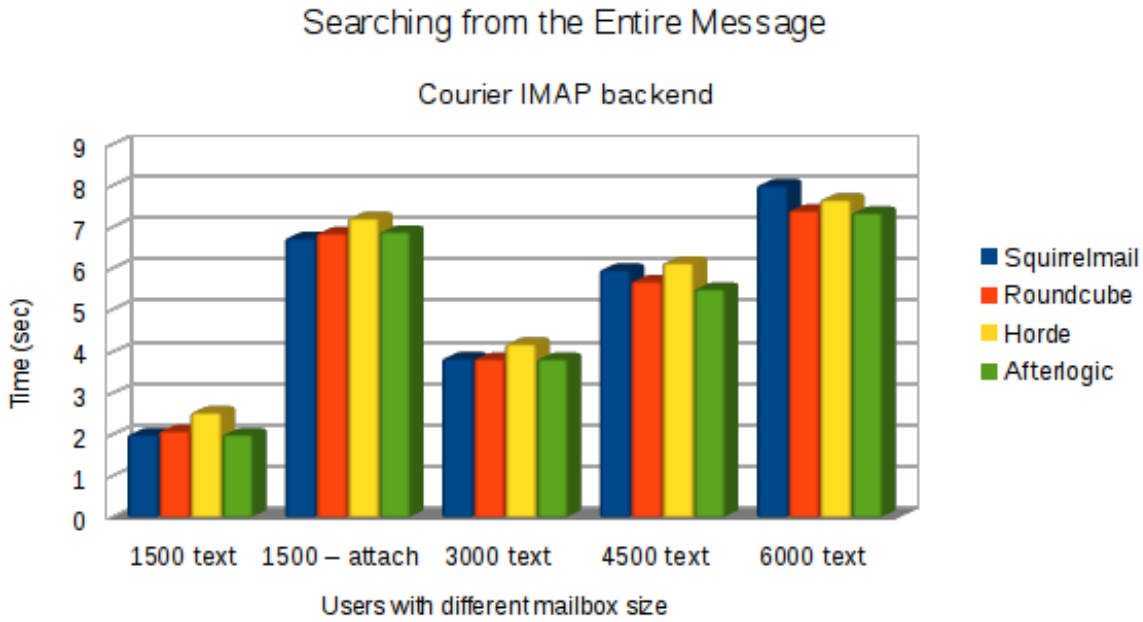
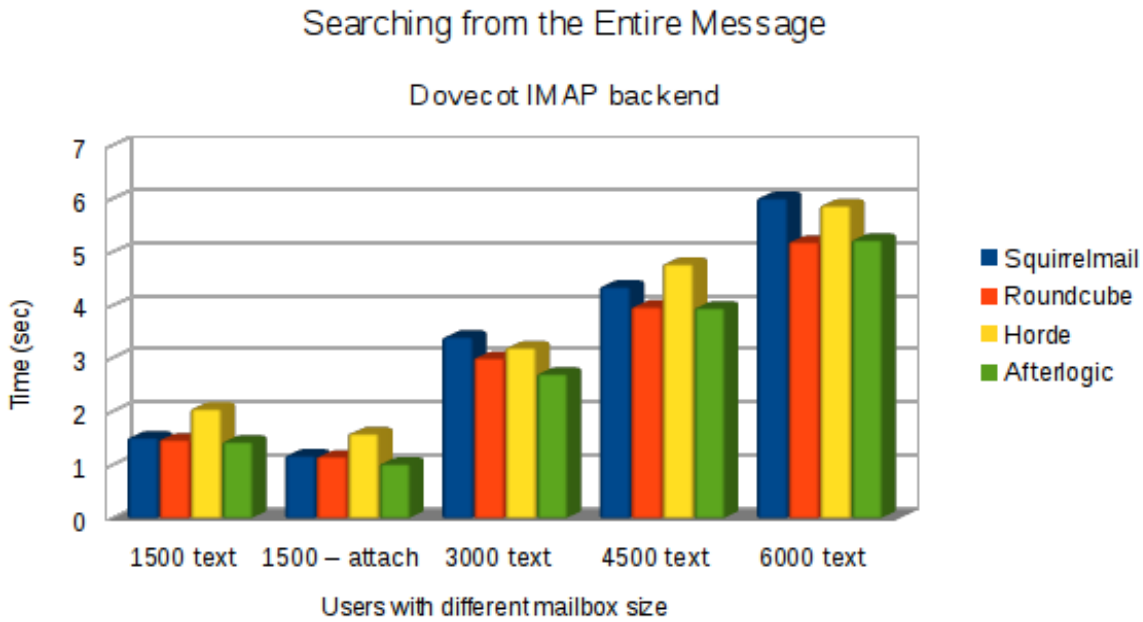
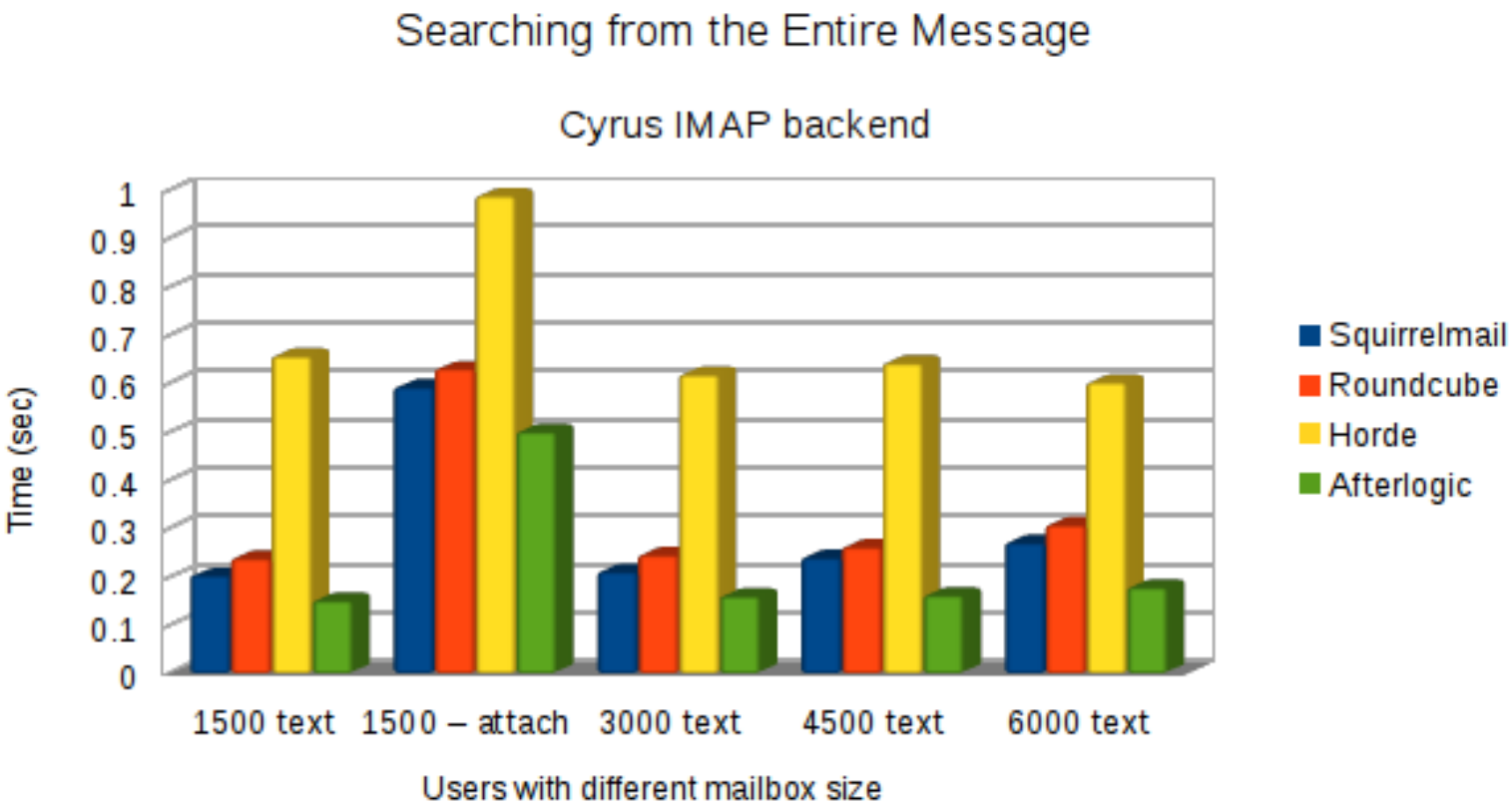


Figure: Comparison of the most effective Solutions regarding searching from "Subject"


Searching from the Entire Message
Courier IMAP backend
Time (sec)
0
1
2
3
4
5
6
7
8
9
1500 text
1500 – attach
3000 text
4500 text
6000 text
Users with different mailbox size
Squirrelmail
Roundcube
Horde
Afterlogic


Searching from the Entire Message
Dovecot IMAP backend
Time (sec)
0
1
2
3
4
5
6
7
1500 text
1500 – attach
3000 text
4500 text
6000 text
Users with different mailbox size
Squirrelmail
Roundcube
Horde
Afterlogic


Searching from the Entire Message
Cyrus IMAP backend
Time (sec)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1500 text
1500 – attach
3000 text
4500 text
6000 text
Users with different mailbox size
Squirrelmail
Roundcube
Horde
Afterlogic

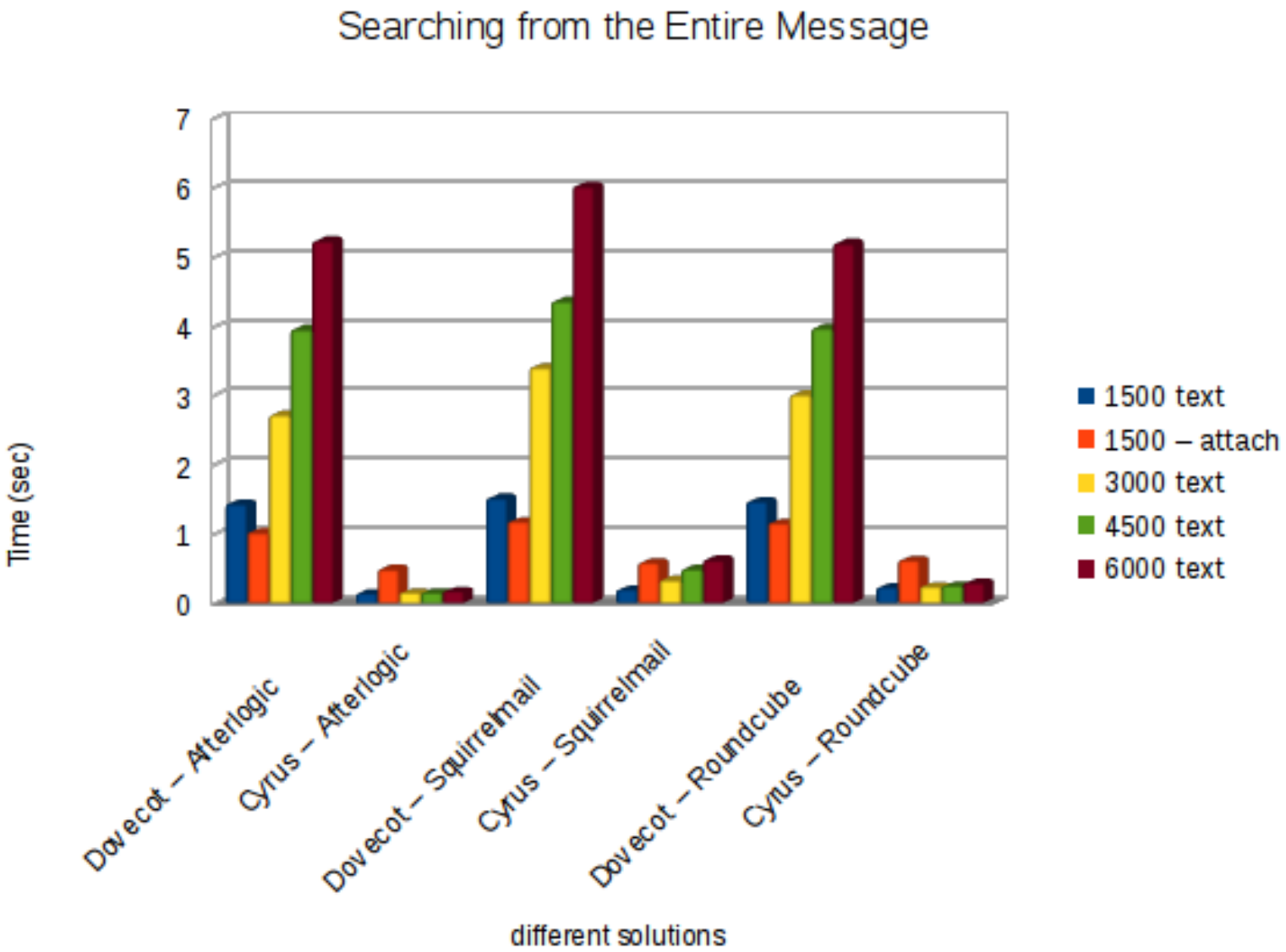
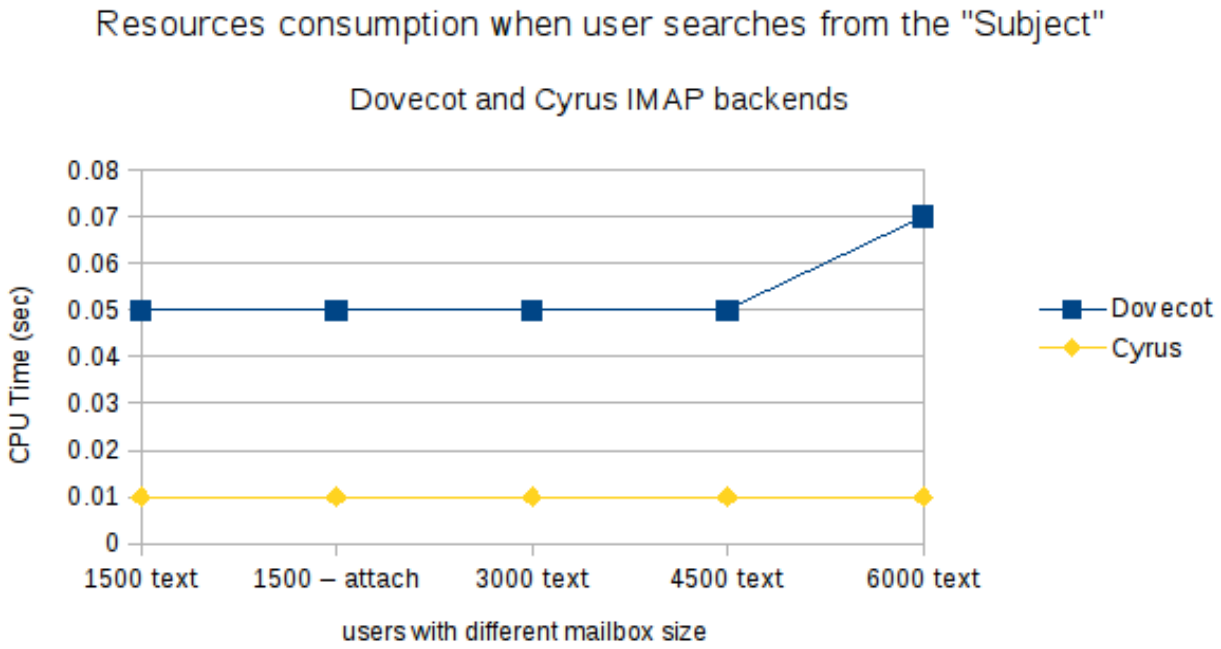
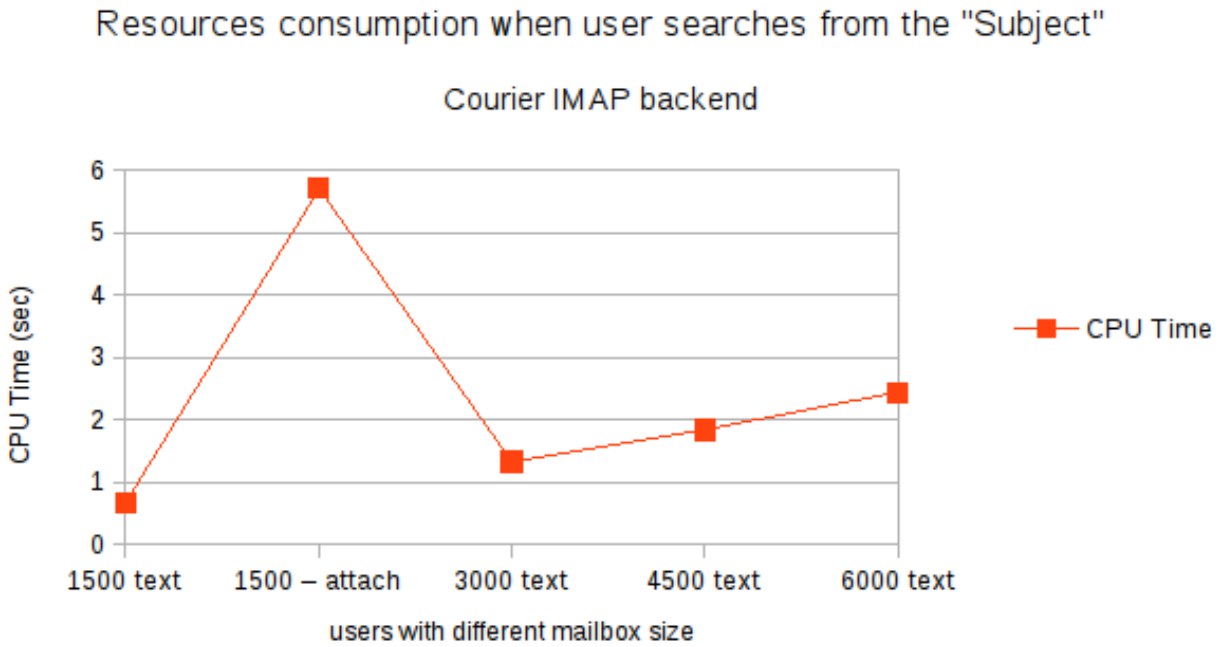


Figure: Comparison of the most effective Solutions regarding searching from Entire Message content


Resources consumption when user searches from the "Subject"
Dovecot and Cyrus IMAP backends
CPU Time (sec)
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
1500 text
1500 – attach
3000 text
4500 text
6000 text
users with different mailbox size
Dovecot
Cyrus


Resources consumption when user searches from the "Subject"
Courier IMAP backend
CPU Time (sec)
6
5
4
3
2
1
0
1500 text
1500 – attach
3000 text
4500 text
6000 text
users with different mailbox size
CPU Time

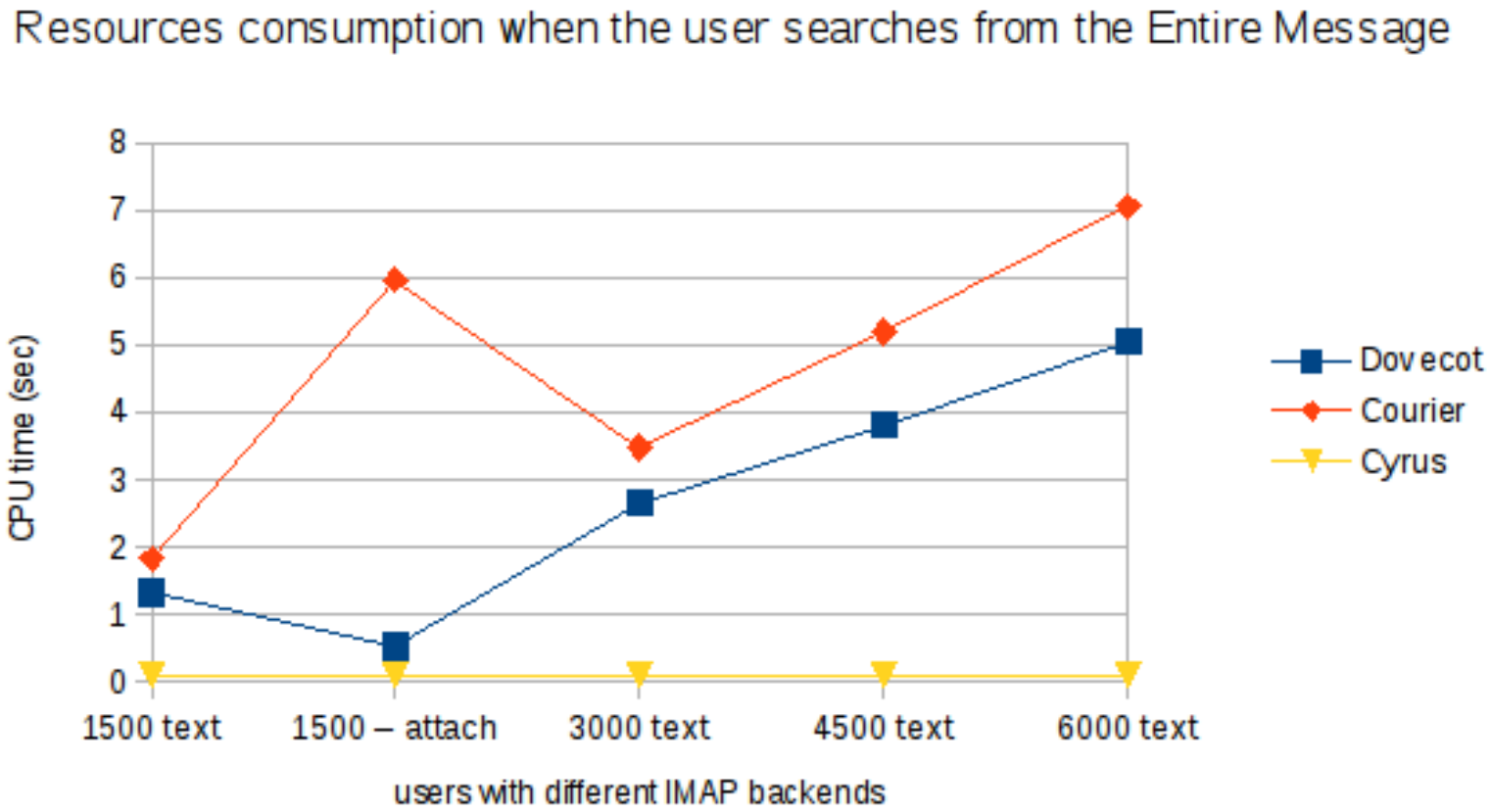


Figure: users search a keyword from the Entire Message Content: CPU time consumption for Dovecot and Courier IMAP backends

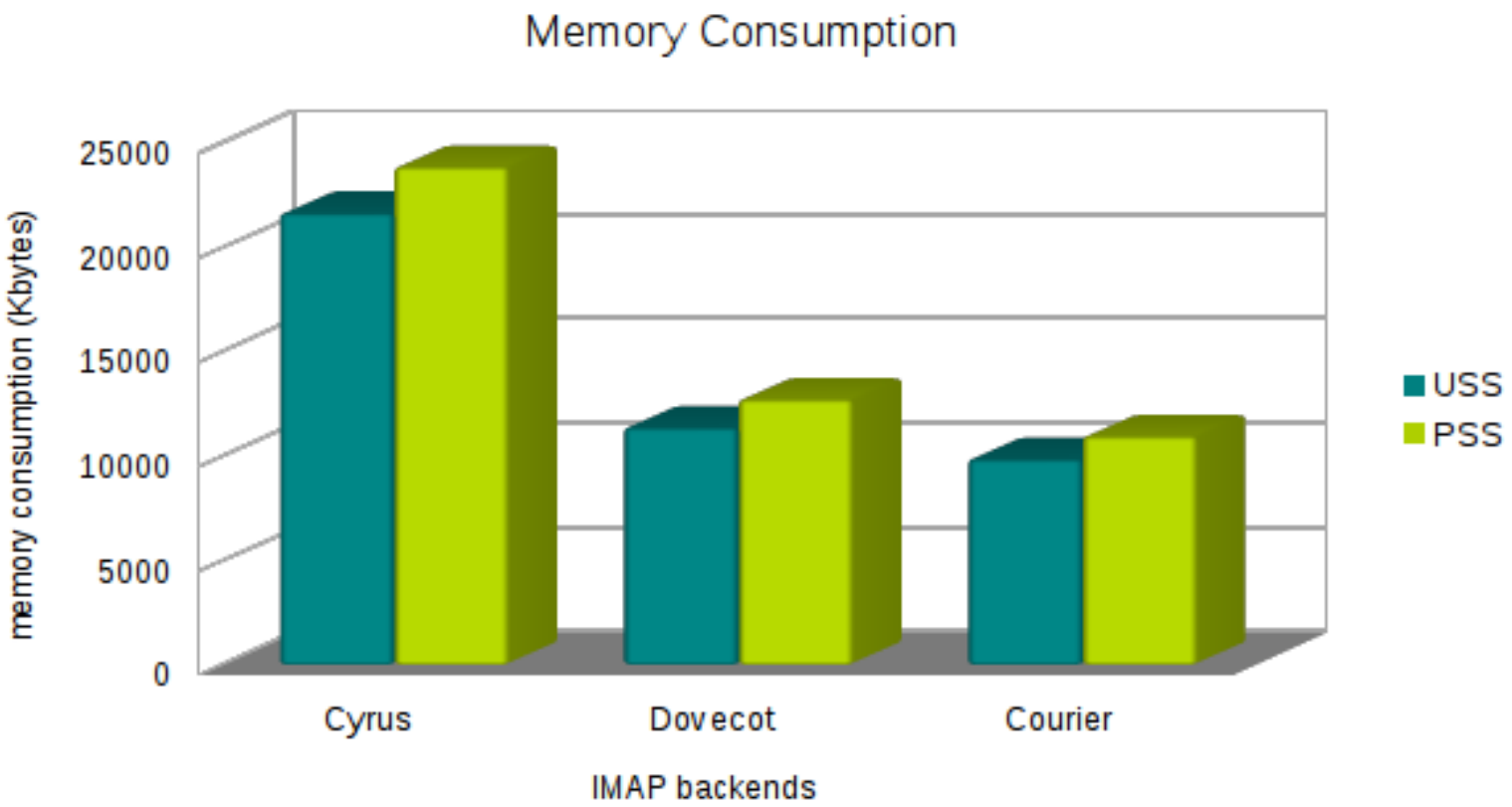


Figure: average memory consumption

Conclusions

- Afterlogic achieves shortest latencies in searching
- Horde IMP has the longest response times
- Dovecot and Cyrus carry out the search from "Subject" request efficiently and with relatively the same latency
- Cyrus is the IMAP backend that performs the best during search from the entire message
- Cyrus has the lowest CPU utilization and the highest average memory consumption for all IMAP functions, followed by Dovecot

Answering the Research Question

What are the bottlenecks, in terms of performance, of current Webmail implementations?

- The major bottleneck in an integrated webmail system is the IMAP backend

which could be the most optimal solution?

- the solution of using Cyrus IMAP combined with Afterlogic Webmail Lite performs better in terms of both user experience and system overall performance

Questions