0.00.5em

0.0.00.5em

0.0.0.00.5em

0em

EMAS Performance Tuning Report Documentation

Release 1.0

Upfront Systems

September 17, 2013

CHAPTER

ONE

INTRODUCTION

In order to test the current configuration, hardware and software, against the published performance goals, a series of benchmark tests were undertaken. The results of these tests were used to do optimisation changes.

This document tries to summarise the data that was gathered, the conclusions that were reached and the remedial steps that were taken.

Each section has a link to the specific Funkload bench reports. When statistics from the Funkload bench reports are used in this document we chose to use the 'P95' values. They represent what happened 95% of the time in the test.

The Funkload bench reports were configure not to follow external links, since we are attempting to benchmark the EMAS software and hardware cluster not external CDNs or external network access.

CHAPTER TWO

1. PRE-TEST OPTIMISATION

We started the optimisation process by identifying those folders that have too many objects. During this investigation we found that MemberServices and Orders are potential problems.

As a first-pass optimisation MemberServices were moved to a PostgreSQL implementation and completely removed from the ZODB.

Orders, on the other hand, were not moved out of the ZODB, since the volumes involved and the amount of times each Order is accessed are well within the ZODB's capacity to handle. In order to stop bloating of the portal catalogue we did move orders to their own smaller catalogue though.

CHAPTER

THREE

2. AN OVERVIEW OF THE SETUP WE BENCHMARKED

Four distinct servers were used in the benchmarking process. Any further mention of 'test cluster' in this document refers to the following setup:

Siyavula Performance 1 (aka. SP1)

- Varnish caching proxy
- HAProxy load balancing server

Siyavula Performance 2 (aka. SP2)

- 4 EMAS application server instances
- 1 ZEO server with Zope Replication Service
- 1 Redis analytics queue
- 1 PostgreSQL cluster (initial primary database server)
- 4 Monassis instances

Siyavula Performance 3 (aka. SP3)

- 4 EMAS application server instances
- 1 ZEO server with Zope Replication Service
- 1 Redis analytics queue
- 1 PostgreSQL cluster (initial secondary database server)
- 4 Monassis instances

Siyavula Performance 4 (aka. sp4)

- Load generation server
- Host for the Funkload benchmark tests

As load generating infrastructure we chose Funkload, since it is written in Python, tests are very easy to record, customise and run and the reporting functions are excellent.

CHAPTER

FOUR

3. TESTING AUTHENTICATED READS

Reading all the possible URLs in the site authenticated was deemed impractical due to the amount of time potentially required to do one Funkload cycle. In order to decide which URLs to use for the authenticated read tests we created a Funkload test that reads all the content unauthenticated (results available here: Science unauthed read). This test was run with only 1 user and 1 cycle.

Unauthenticated read setup:

- Launched: 2013-07-26 15:54:29
- Test: test_wholesite.py WholeSite.test_WholeSite
- Target server: http://qap.everythingscience.co.za
- Cycles of concurrent users: [1]
- Cycle duration: 800s
- Apdex: 1.5

From this list of URLs we chose to benchmark the following in the authenticated read test:

- /
- /login
- /login_form
- grade-12/01-organic-molecules/01-organic-molecules-07.cnxmlplus
- grade-10/19-quantitative-aspects-of-chemical-change/19-quantitative-aspects-of-chemical-change-01.cnxmlplus
- grade-10/19-quantitative-aspects-of-chemical-change/19-quantitative-aspects-of-chemical-change-06.cnxmlplus
- grade-11/04-intermolecular-forces/04-intermolecular-forces-02.cnxmlplus Get grade-11/04-intermolecular-forces/04-intermolecular-forces-02.cnxmlplus
- grade-10/05-the-periodic-table/05-the-periodic-table-01.cnxmlplus Get grade-10/05-the-periodic-table/05-the-periodic-table-01.cnxmlplus
- grade-10/22-mechanical-energy/22-mechanical-energy-01.cnxmlplus
- grade-10/24-units-used-in-the-book/24-units-used-in-the-book-01.cnxmlplus
- grade-12/02-organic-macromolecules/02-organic-macromolecules-01.cnxmlplus
- grade-12/05-the-chemical-industry/05-the-chemical-industry-02.cnxmlplus
- grade-12/08-work-energy-and-power/08-work-energy-and-power-03.cnxmlplus
- grade-12/10-colour/10-colour-06.cnxmlplus

- grade-12/11-2d-and-3d-wavefronts/11-2d-and-3d-wavefronts-08.cnxmlplus
- grade-12/12-wave-nature-of-matter/12-wave-nature-of-matter-01.cnxmlplus
- grade-11/13-types-of-reactions/13-types-of-reactions-01.cnxmlplus
- grade-11/14-lithosphere/14-lithosphere-01.cnxmlplus

The criterium we used to choose the above URLs is simply the performance in the unauthenticated reading tests. The pages that are slow during unauthenticated reading will be even slower during authenticated reading.

We also chose some URLs that seemed to serve quite fast. This we did to get some balance to the overall stats for the reading experience.

The resultant Funkload test was run with 4 test cycles ranging from 100 to 1000 concurrent users.

Authenticated read setup:

- Launched: 2013-08-22 14:35:07
- From: siyavulap04
- Test: test_AuthenticatedRead.py AuthenticatedRead.test_AuthenticatedRead
- Target server: http://qap.everythingscience.co.za
- Cycles of concurrent users: [100, 250, 500, 750, 1000]
- Apdex: 1.5

The results of each test cycle contains:

- 18 pages
- 59 links
- 99 images

The benchmark test as a whole (all cycles and users) contains:

- 381 tests
- 9701 pages
- 100343 requests

CHAPTER

FIVE

4. AUTHENTICATED READ TEST RESULTS

Funkload bench report here: Authenticated read

5.1 Page analysis

5.1.1 Home page

The unauthenticated home page load at **100 concurrent users** looks like this:

URL	Request time
/	1.337 s
/css?family=Montserrat	1.124 s
/	1.245 s
/portal_css/Sunburst%20Theme/public-cachekey-4fff4ed932d766e26813993d85f43eea.css	1.120 s
/portal_css/Sunburst%20Theme/dropdown-menu-cachekey-18dee82342b75f2c7bc0fa7b017feb61.css	1.119 s
/portal_css/Sunburst%20Theme/resourcetinymce.stylesheetstinymce-cachekey-ca7f99b34a27033d846be95c8de69be2.css	1.073 s
/portal_css/Sunburst%20Theme/resourceplone.app.dexterity.overlays-cachekey-7ac1852449e6cb2ff27111e1cd7c4665.css	1.226 s
/portal_css/Sunburst%20Theme/resourcecollective.topictreetopictree-cachekey-2a473052fae56de9ea0cbbec5dfaa63d.css	1.130 s
/portal_css/Sunburst%20Theme/resourcethemesapplestyle-cachekey-902306361fbd1b097cea265775a7f6da.css	1.137 s
/portal_css/Sunburst%20Theme/themeemas.appcssstyles-cachekey-c58e347e4c0ca6ab98d3a4104c40af46.css	1.177 s
/portal_css/Sunburst%20Theme/themeemas.themecssstyles-cachekey-987e0b9963b14f5de16733ce5a566073.css	1.269 s
/portal_css/Sunburst%20Theme/ploneCustom-cachekey-74895962889ac3a836dba1b4b8323474.css	1.166 s
/portal_kss/Sunburst%20Theme/at-cachekey-9d4065eabe538900e9c3dd6fa55b6acc.kss	1.229 s
/favicon.ico	1.157 s
/touch_icon.png	1.045 s
/++theme++emas.theme/images/logo.png	1.054 s
/++theme++emas.theme/images/howitworks.png	1.108 s
/++theme++emas.theme/images/graph.png	1.179 s
/++theme++emas.theme/images/answer_correct.png	1.035 s
/++theme++emas.theme/images/answer_incorrect.png	1.161 s
/++theme++emas.theme/images/dashboard.png	1.106 s
/++theme++emas.theme/images/learnersdashboard.png	1.250 s
/++theme++emas.theme/images/teachersdashboard.png	1.145 s
/++theme++emas.theme/images/media.png	1.176 s
/++theme++emas.theme/images/textbooks.png	1.167 s
/++theme++emas.theme/images/Logo_transparentBackground-tiny.png	1.081 s
/++theme++emas.theme/images/shuttleworthfoundation.jpg	1.047 s
/++theme++emas.theme/images/psggroup.jpg	0.992 s

Continued on next page

Table 5.1 – continued from previous page

URL	Request time
/++theme++emas.theme/images/FaceBook-icon-small.png	1.069 s
/++theme++emas.theme/images/Twitter-icon-small.png	1.019 s
/++theme++emas.theme/images/cc_by.png	1.071 s

Thus we get a **total load time of 35.214 seconds**. Bear in mind that this is for an initial load. On initial load all the CSS and javascript will be fetched over the netword and cached by the browser.

Subsequent unauthenticated loads look like this:

URL	Request time
/	1.337 s
/	1.245 s
/touch_icon.png	1.045 s
/++theme++emas.theme/images/logo.png	1.054 s
/++theme++emas.theme/images/howitworks.png	1.108 s
/++theme++emas.theme/images/graph.png	1.179 s
/++theme++emas.theme/images/answer_correct.png	1.035 s
/++theme++emas.theme/images/answer_incorrect.png	1.161 s
/++theme++emas.theme/images/dashboard.png	1.106 s
/++theme++emas.theme/images/learnersdashboard.png	1.250 s
/++theme++emas.theme/images/teachersdashboard.png	1.145 s
/++theme++emas.theme/images/media.png	1.176 s
/++theme++emas.theme/images/textbooks.png	1.167 s
/++theme++emas.theme/images/Logo_transparentBackground-tiny.png	1.081 s
/++theme++emas.theme/images/shuttleworthfoundation.jpg	1.047 s
/++theme++emas.theme/images/psggroup.jpg	0.992 s
/++theme++emas.theme/images/FaceBook-icon-small.png	1.069 s
/++theme++emas.theme/images/Twitter-icon-small.png	1.019 s
/++theme++emas.theme/images/cc_by.png	1.071 s

Thus a load time of **21.287 seconds.**

Projected serve rates

Working with these 2 figures we can project the following:

Initial home pages per hour: (60 / 35.214) * 60 = 102.232

Subsequent home pages per hour: (60 / 21.287) * 60 = 169.117

Higher concurrencies

250 concurrent users

URL	Request time
/	1.969 s
/	2.200 s
/touch_icon.png	1.755 s
/++theme++emas.theme/images/logo.png	1.542 s
/++theme++emas.theme/images/howitworks.png	1.409 s
/++theme++emas.theme/images/graph.png	1.510 s
/++theme++emas.theme/images/answer_correct.png	1.573 s
/++theme++emas.theme/images/answer_incorrect.png	1.884 s
/++theme++emas.theme/images/dashboard.png	1.512 s
/++theme++emas.theme/images/learnersdashboard.png	1.536 s
/++theme++emas.theme/images/teachersdashboard.png	1.453 s
/++theme++emas.theme/images/media.png	1.689 s
/++theme++emas.theme/images/textbooks.png	1.592 s
/++theme++emas.theme/images/Logo_transparentBackground-tiny.png	1.603 s
/++theme++emas.theme/images/shuttleworthfoundation.jpg	1.595 s
/++theme++emas.theme/images/psggroup.jpg	1.601 s
/++theme++emas.theme/images/FaceBook-icon-small.png	1.511 s
/++theme++emas.theme/images/Twitter-icon-small.png	1.584 s
/++theme++emas.theme/images/cc_by.png	1.434 s

Load time per page: **30.952 seconds**

(60 / 30.952) * 60 = **116.30 pages per hour.**

500 concurrent users

URL	Request time
/	6.978 s
/	3.301 s
/touch_icon.png	6.399 s
/++theme++emas.theme/images/logo.png	3.583 s
/++theme++emas.theme/images/howitworks.png	4.255 s
/++theme++emas.theme/images/graph.png	2.345 s
/++theme++emas.theme/images/answer_correct.png	2.344 s
/++theme++emas.theme/images/answer_incorrect.png	3.769 s
/++theme++emas.theme/images/dashboard.png	5.999 s
/++theme++emas.theme/images/learnersdashboard.png	3.485 s
/++theme++emas.theme/images/teachersdashboard.png	2.750 s
/++theme++emas.theme/images/media.png	1.944 s
/++theme++emas.theme/images/textbooks.png	2.506 s
/++theme++emas.theme/images/Logo_transparentBackground-tiny.png	2.848 s
/++theme++emas.theme/images/shuttleworthfoundation.jpg	3.320 s
/++theme++emas.theme/images/psggroup.jpg	2.667 s
/++theme++emas.theme/images/FaceBook-icon-small.png	1.665 s
/++theme++emas.theme/images/Twitter-icon-small.png	2.019 s
/++theme++emas.theme/images/cc_by.png	2.423 s

Load time per page: **64.6 seconds**

(60 / 64.6) * 60 = **55.72 pages per hour.**

750 concurrent users

URL	Request time
/	6.107 s
/	7.159 s
/touch_icon.png	2.978 s
/++theme++emas.theme/images/logo.png	5.180 s
/++theme++emas.theme/images/howitworks.png	5.429 s
/++theme++emas.theme/images/graph.png	5.712 s
/++theme++emas.theme/images/answer_correct.png	5.406 s
/++theme++emas.theme/images/answer_incorrect.png	3.615 s
/++theme++emas.theme/images/dashboard.png	6.164 s
/++theme++emas.theme/images/learnersdashboard.png	4.241 s
/++theme++emas.theme/images/teachersdashboard.png	2.094 s
/++theme++emas.theme/images/media.png	2.532 s
/++theme++emas.theme/images/textbooks.png	5.995 s
/++theme++emas.theme/images/Logo_transparentBackground-tiny.png	3.456 s
/++theme++emas.theme/images/shuttleworthfoundation.jpg	2.977 s
/++theme++emas.theme/images/psggroup.jpg	2.973 s
/++theme++emas.theme/images/FaceBook-icon-small.png	3.509 s
/++theme++emas.theme/images/Twitter-icon-small.png	3.960 s
/++theme++emas.theme/images/cc_by.png	1.655 s

Load time per page: **81.142 seconds**

(60 / 81.142) * 60 = **44.36 pages per hour.**

1000 concurrent users

URL	Request time
/	7.710 s
/	1.807 s
/touch_icon.png	2.185 s
/++theme++emas.theme/images/logo.png	2.001 s
/++theme++emas.theme/images/howitworks.png	4.991 s
/++theme++emas.theme/images/graph.png	2.312 s
/++theme++emas.theme/images/answer_correct.png	4.129 s
/++theme++emas.theme/images/answer_incorrect.png	7.016 s
/++theme++emas.theme/images/dashboard.png	3.769 s
/++theme++emas.theme/images/learnersdashboard.png	2.536 s
/++theme++emas.theme/images/teachersdashboard.png	2.102 s
/++theme++emas.theme/images/media.png	4.330 s
/++theme++emas.theme/images/textbooks.png	6.296 s
/++theme++emas.theme/images/Logo_transparentBackground-tiny.png	3.881 s
/++theme++emas.theme/images/shuttleworthfoundation.jpg	6.484 s
/++theme++emas.theme/images/psggroup.jpg	2.454 s
/++theme++emas.theme/images/FaceBook-icon-small.png	2.743 s
/++theme++emas.theme/images/Twitter-icon-small.png	5.625 s
/++theme++emas.theme/images/cc_by.png	6.470 s

Load time per page: **78.841**

(60 / 78.841) * 60 = **45.66 pages per hour.**

It is quite clear that as the concurrency rises the cluster serves less- and-less pages per second, but never completely stops working.

5.1.2 Content pages

Let's look at one of the slow science pages like we did with the home page.

URL	Request time
/grade-12/08-work-energy-and-power/08-work-energy-and-power-03.cnxmlplus	0.990 s
/grade-12/08-work-energy-and-power/08-work-energy-and-power-03.cnxmlplus/	1.086 s
/grade-12/08-work-energy-and-power/08-work-energy-and-power-02.cnxmlplus	4.221 s
/grade-12/08-work-energy-and-power/08-work-energy-and-power-04.cnxmlplus	1.842 s
/grade-12/08-work-energy-and-power/pspictures/f70fc7e8583786ef8c496e4861d8f2b7.png	1.382 s
/grade-12/08-work-energy-and-power/pspictures/24707967fddfb273f965a0cf7224ac0a.png	1.501 s
/grade-12/08-work-energy-and-power/pspictures/22fc66e880fffb15853e6873faa1aa2b.png	1.152 s
/grade-12/08-work-energy-and-power/++theme++emas.theme/images/cc_by.png	1.028 s

Projected serve rates

It is clear that the javascript and CSS is not fetched again. Given the above times we know that each page will take **13.202 seconds** to fetch at a load of **100 concurrent users**.

This in turn means we can potentially serve:

(60 / 13.202) * 60 = **272.68 pages per hour.**

Higher concurrencies

250 concurrent users

URL	Request time
/grade-12/08-work-energy-and-power/08-work-energy-and-power-03.cnxmlplus	1.659 s
/grade-12/08-work-energy-and-power/08-work-energy-and-power-03.cnxmlplus/	1.500 s
/grade-12/08-work-energy-and-power/08-work-energy-and-power-02.cnxmlplus	1.658 s
/grade-12/08-work-energy-and-power/08-work-energy-and-power-04.cnxmlplus	1.853 s
/grade-12/08-work-energy-and-power/pspictures/f70fc7e8583786ef8c496e4861d8f2b7.png	1.875 s
/grade-12/08-work-energy-and-power/pspictures/24707967fddfb273f965a0cf7224ac0a.png	1.809 s
/grade-12/08-work-energy-and-power/pspictures/22fc66e880fffb15853e6873faa1aa2b.png	1.896 s
/grade-12/08-work-energy-and-power/++theme++emas.theme/images/cc_by.png	1.624 s

Page load time: **12.25 seconds**

(60 / 12.25) * 60 = **293.87 pages per hour.**

500 concurrent users

URL	Request time
/grade-12/08-work-energy-and-power/08-work-energy-and-power-03.cnxmlplus	2.010 s
/grade-12/08-work-energy-and-power/08-work-energy-and-power-03.cnxmlplus/	6.123 s
/grade-12/08-work-energy-and-power/08-work-energy-and-power-02.cnxmlplus	1.787 s
/grade-12/08-work-energy-and-power/08-work-energy-and-power-04.cnxmlplus	1.891 s
/grade-12/08-work-energy-and-power/pspictures/f70fc7e8583786ef8c496e4861d8f2b7.png	2.944 s
/grade-12/08-work-energy-and-power/pspictures/24707967fddfb273f965a0cf7224ac0a.png	6.484 s
/grade-12/08-work-energy-and-power/pspictures/22fc66e880fffb15853e6873faa1aa2b.png	2.117 s
/grade-12/08-work-energy-and-power/++theme++emas.theme/images/cc_by.png	5.994 s

Page load time: **29.35 seconds**

(60 / 29.35) * 60 = **122.65 pages per hour.**

750 concurrent users

URL	Request time
/grade-12/08-work-energy-and-power/08-work-energy-and-power-03.cnxmlplus	1.569 s
/grade-12/08-work-energy-and-power/08-work-energy-and-power-03.cnxmlplus/	6.385 s
/grade-12/08-work-energy-and-power/08-work-energy-and-power-02.cnxmlplus	4.795 s
/grade-12/08-work-energy-and-power/08-work-energy-and-power-04.cnxmlplus	5.794 s
/grade-12/08-work-energy-and-power/pspictures/f70fc7e8583786ef8c496e4861d8f2b7.png	4.397 s
/grade-12/08-work-energy-and-power/pspictures/24707967fddfb273f965a0cf7224ac0a.png	5.453 s
/grade-12/08-work-energy-and-power/pspictures/22fc66e880fffb15853e6873faa1aa2b.png	2.007 s
/grade-12/08-work-energy-and-power/++theme++emas.theme/images/cc_by.png	4.610 s

Page load time: **35.01 seconds**

(60 / 35.01) * 60 = **102.82 pages per hour.**

1000 concurrent users

URL	Request time
/grade-12/08-work-energy-and-power/08-work-energy-and-power-03.cnxmlplus	3.421 s
/grade-12/08-work-energy-and-power/08-work-energy-and-power-03.cnxmlplus/	1.985 s
/grade-12/08-work-energy-and-power/08-work-energy-and-power-02.cnxmlplus	1.886 s
/grade-12/08-work-energy-and-power/08-work-energy-and-power-04.cnxmlplus	3.537 s
/grade-12/08-work-energy-and-power/pspictures/f70fc7e8583786ef8c496e4861d8f2b7.png	7.234 s
/grade-12/08-work-energy-and-power/pspictures/24707967fddfb273f965a0cf7224ac0a.png	6.397 s
/grade-12/08-work-energy-and-power/pspictures/22fc66e880fffb15853e6873faa1aa2b.png	6.973 s
/grade-12/08-work-energy-and-power/++theme++emas.theme/images/cc_by.png	3.309 s

Page load time: **34.742 seconds**

(60 / 34.742) * 60 = **103.62 pages per hour.**

5.2 Optimisations done

During the testing process we realised that some elements in the pages are causing sub-optimal caching in Varnish. This is due to elements like username and personal links which are unique to each authenticated user. These elements cause Varnish to view pages as different although very little actually differ between them.

We implemented an Edge-side include (ESI) for the personal toolbar which leads to Varnish caching most of the page and only fetching the ESI content.

CHAPTER

SIX

5. TESTING PRACTICE SERVICE

In order to test the Intelligent Practise service fully, Carl Scheffler implemented an ‘oracle’ for answers generated from the Monassis data. This ‘oracle’ we then wrapped in an HTTP server when we found that opening the pickle of all the saved answers to be a huge performance hit in our Funkload tests.

During the testing we also tested the practice proxy in the Plone application. This was done in order to establish if any processing in this proxy is possibly more of a performance issue than processing in the external system. Here are the Practice proxy results. To test this we recorded a Funkload test that logs in to the site and then navigates to a simple view in Monassis. This view does no processing beyond returning basic headers and the string literal ‘OK’.

For the full practise service test we recorded a Funkload test that logs in to the site, browses to the practise service and then does 10 questions. The answers to these questions are fetched from the ‘oracle’ HTTP server. This test we then ran with user concurrencies of 100, 150 and 200.

We used the following test configuration:

- Launched: 2013-08-23 12:10:13
- From: siyavulap04
- Test: test_Practice.py Practice.test_practice
- Target server: http://qap.everythingmaths.co.za
- Cycles of concurrent users: [100, 150, 200]
- Apdex: 1.5

CHAPTER

SEVEN

6. RESULTS FOR TESTING PRACTICE SERVICE

Funkload bench report here: Practise service test

7.1 Page analysis

7.1.1 Dashboard

Authenticated read of the dashboard at 100 concurrent users:

URL	Request time
/@@practice/grade-10	3.805 s
/@@practice/dashboard	5.386 s
/@@practice/static/practice.css	4.291 s
/@@practice/static/practice-ie8.css	3.457 s
/@@practice/static/jqplot/jquery.jqplot.min.css	1.873 s
/@@practice/static/help-icon-no-shadow-16.png	1.553 s
/@@practice/image/mastery_progress_3_115_0	1.421 s
/@@practice/image/mastery_progress_3_115_115	1.400 s
/@@practice/static/progress-up.png	1.222 s
/@@practice/static/gold-star-16.png	1.135 s
/@@practice/static/please_wait_24.gif	0.875 s
/@@practice/static/tick.png	1.096 s
/@@practice/static/gray-star-16.png	0.819 s
/@@practice/image/mastery_progress_4_0_0	0.961 s
/@@practice/image/mastery_progress_3_0_0	0.906 s
/@@practice/image/mastery_progress_1_0_0	0.980 s
/@@practice/image/mastery_progress_2_0_0	0.912 s
/@@practice/image/mastery_progress_2_120_0	0.924 s
/@@practice/image/mastery_progress_3_120_0	0.965 s
/@@practice/image/mastery_progress_3_111_0	1.063 s
/@@practice/image/mastery_progress_3_108_0	1.405 s
/++theme++emas.theme/images/copyright.png	1.623 s
/++theme++emas.theme/images/copyright.png	2.379 s

Projected serve rates

This gives us a load time of **40.451 seconds per page at 100 concurrent users.**

At this rate we can serve:

(60 / 40.451) * 60 = **88.99 pages per hour.**

Higher concurrencies

150 concurrent users

URL	Request time
/@@practice/grade-10	2.502 s
/@@practice/dashboard	4.077 s
/@@practice/static/practice.css	2.601 s
/@@practice/static/practice-ie8.css	2.439 s
/@@practice/static/jqplot/jquery.jqplot.min.css	2.383 s
/@@practice/static/help-icon-no-shadow-16.png	2.432 s
/@@practice/image/mastery_progress_3_115_0	2.282 s
/@@practice/image/mastery_progress_3_115_115	2.516 s
/@@practice/static/progress-up.png	2.334 s
/@@practice/static/gold-star-16.png	2.420 s
/@@practice/static/please_wait_24.gif	2.432 s
/@@practice/static/tick.png	2.380 s
/@@practice/static/gray-star-16.png	2.377 s
/@@practice/image/mastery_progress_4_0_0	2.309 s
/@@practice/image/mastery_progress_3_0_0	2.305 s
/@@practice/image/mastery_progress_1_0_0	2.411 s
/@@practice/image/mastery_progress_2_0_0	2.417 s
/@@practice/image/mastery_progress_2_120_0	2.469 s
/@@practice/image/mastery_progress_3_120_0	2.406 s
/@@practice/image/mastery_progress_3_111_0	2.538 s
/@@practice/image/mastery_progress_3_108_0	5.071 s
/++theme++emas.theme/images/copyright.png	5.021 s
/++theme++emas.theme/images/copyright.png	2.467 s

Page load time: **62.589 seconds**

(60 / 62.589) * 60 = **57.51 pages per hour.**

200 concurrent users

URL	Request time
/@@practice/grade-10	4.775 s
/@@practice/dashboard	8.039 s
/@@practice/static/practice.css	5.130 s
/@@practice/static/practice-ie8.css	4.949 s
/@@practice/static/jqplot/jquery.jqplot.min.css	4.692 s
/@@practice/static/help-icon-no-shadow-16.png	4.743 s
/@@practice/image/mastery_progress_3_115_0	5.308 s
/@@practice/image/mastery_progress_3_115_115	4.980 s
/@@practice/static/progress-up.png	4.621 s
/@@practice/static/gold-star-16.png	4.873 s
/@@practice/static/please_wait_24.gif	4.716 s
/@@practice/static/tick.png	4.512 s
/@@practice/static/gray-star-16.png	4.862 s
/@@practice/image/mastery_progress_4_0_0	4.720 s
/@@practice/image/mastery_progress_3_0_0	4.853 s
/@@practice/image/mastery_progress_1_0_0	4.706 s
/@@practice/image/mastery_progress_2_0_0	4.683 s
/@@practice/image/mastery_progress_2_120_0	4.644 s
/@@practice/image/mastery_progress_3_120_0	4.723 s
/@@practice/image/mastery_progress_3_111_0	4.871 s
/@@practice/image/mastery_progress_3_108_0	4.543 s
/++theme++emas.theme/images/copyright.png	0.307 s
/++theme++emas.theme/images/copyright.png	0.000 s

Page load time: **104.25 seconds**

(60 / 104.25) * 60 = **34.532 pages per hour.**

7.2 Optimisations done

When we analysed the data from the practice service test we realized that the Plone login process takes quite a bit of time. Upon further investigation we found that the user object was being update on each login. This is unnecessary given that we do not require the last login time. We changed that specific method and removed all unnecessary changes to the user object.

CHAPTER

EIGHT

7. TESTING MOBILE READS

Funkload bench report here: Mobile test

CHAPTER

NINE

8. RESULTS FOR TESTING MOBILE READS

9.1 page analysis

9.1.1 Home page

Projected serve rates

Higher concurrencies

100

200

9.1.2 Content pages

Projected serve rates

Higher concurrencies

100

200

CHAPTER

TEN

9. TESTING VARNISH

As background to this test consider the following. The application servers SP2 and SP3 are connected via a non-routable private subnet in the 10.0.0.* range. In the current cluster setup they are accessed over this private subnet via the HAProxy and Varnish servers on SP1. This means any latency or throughput issues on the subnet will adversly affect the total scalability.

Varnish serves all our cachable resources (CSS, javascript, images, etc.). In order to understand the total scalability we decided to checked Varnish's scalability in our current cluster setup.

We used Apache Benchmark to test Varnish from our load generating server and the Varnish/ HAProxy server. This was done with a script that starts off with 1 user and 10 requests all the way up to 1000 concurrent users and 1000000 requests.

CHAPTER

ELEVEN

10. RESULTS OF VARNISH

11.1 1 user

Complete requests	SP1 requests/s	SP4 requests/s
100	3799.39	242.94
1000	4672.11	242.47
10000	4271.39	242.78
100000	4457.42	243.10
1000000	4828.27	242.91

11.2 10 concurrent users

Complete requests	SP1 requests/s	SP4 requests/s
100	11041.18	356.05
1000	20597.32	356.20
10000	21980.24	358.07
100000	18690.17	358.09
1000000	20729.00	358.04

11.3 100 concurrent users

Complete requests	SP1 requests/s	SP4 requests/s
100	9004.95	242.86
1000	17513.13	357.70
10000	18031.14	358.10
100000	18753.04	358.13
1000000	18552.96	358.13

11.4 1000 concurrent users

Complete requests	SP1 requests/s	SP4 requests/s
100	no data (1)	no data
1000	10249.79	129.72
10000	12786.09	no data
100000	15860.49	no data
1000000	16436.69	no data

(1) An entry of ‘no data’ indicates that the test cycle could not complete successfully and therefore Apache Benchmark did not record the statistics.

Both SP1 and SP4 show relatively linear changes in performance. The important thing is the marked difference in the amount of requests per second between the 2 servers. After more investigation we found that the back-end network between the servers in the cluster is not running at its full capacity. This has been changed and a second set of tests will be run to validate the assumption that network throughput is responsible for the difference in performance between the 2 mentioned servers.

CHAPTER

TWELVE

RECOMMENDATION FOR SCALING / CONCLUSION