

UNC load testing grouper

During our process of upgrading Grouper to the latest version, we wanted to do some load testing in a pre-preproduction environment. Since our non-production installs are lightly used, our primary aim was to exercise the application, Java container, and web server under a light but steady load, as a predictor of how the application will fare in production. A secondary purpose was to get timing results for the various features, comparing them with the previous version, which for us was the quite old 1.6.3. Stress testing wasn't a goal, as the UI for our production install isn't used that heavily.

For our load tests, we used Apache JMeter 3.0 from <http://jmeter.apache.org/>. Tests were done from a Windows 7 desktop. The specs for our institutional Grouper installation are:

- Apache web server, Glassfish 3.1.2 Java container (2-server cluster)
- Oracle 11 database
- Grouper 2.3.0 with patches through api #22

For comparison, we also ran the same load test against Unicon's Grouper Docker container (<https://hub.docker.com/r/unicon/grouper-demo/>, version 2.3.0-2017-01-08), running from the same Windows 7 desktop.

JMeter Implementation

The JMeter test plan uses a few different data sources in order to vary the tests during multiple loops. A login source specifies usernames and passwords for administrator, privileged user (e.g., has admin access to certain groups), and normal users. A folder id source lists the UUID values for a small number of stems, to simulate expanding folders in the directory tree browser. A group and member source lists groups, stems, and users by UUID to test calls to access various content.

Our institutional server uses Shibboleth SSO for authentication. When testing this server, a one-time step per thread loop does the initial SSO session setup, extracting session parameters as necessary. For the Docker application, HTTP Basic authentication was used. In this case, a BeanShell step Base64 encodes the username and password to include in the HTTP headers. To switch from one authentication method to the other, the pertinent steps are enabled and disabled as appropriate.

Because the new UI includes CSRF protection, it's necessary to include the value for OWASP_CSRFTOKEN for most of the requests. At the start of each thread loop, the value for OWASP_CSRFTOKEN is extracted from the first page that is accessed. This value is then included in subsequent pages where it's required.

The pages and Ajax requests accessed in each loop are:

1. Initial landing page
2. Directory browser (lower left) - root level
3. Home page, main content area
4. Directory tree, opening various folders (loop)
5. My groups
6. My folders
7. My favorites
8. Search result (the upper right corner)
9. View group
10. View stem
11. View users

12. Add member autocompletion result (after 2 chars) - Dojo Ajax return HTML
13. Trace membership
14. View group privileges
15. Trace privileges
16. Group memberships in other groups
17. View group audit log (if user is in wheel group)

Results

Non-Production Institutional server

Function	msec (wheel user)	msec (privileged user)
Home Page authorized	43	56
Directory Browser frame - root folder	50	1506
Content frame - home	305	240
Open Stem in Directory Tree	48	1523
Content frame - my groups	185	81
Content frame - my folders	59	586
Content frame - my favorites	74	87
Content frame - search result (upper right)	163	3516
Content frame - view group	649	483
Content frame - view stem	107	2531
Content frame - view user	282	1562
Content frame - add member autocomplete (after 2 chars)	626	866
Content frame - trace membership	616	1741
Content frame - view group privileges	189	168
Content frame - group memberships in other groups	122	122
Content frame - view group audit log	(not tested)	

Note: Tracing privileges was not performed, as the installation did not include the fix from patch #38, and the request timed out after 2 minutes.

Function	msec (wheel user)	msec (privileged user)	
Home Page authorized	26	19	17
Directory Browser frame - root folder	38	105	94
Content frame - home	88	61	60
Open Stem in Directory Tree	27	91	92
Content frame - my groups	84	28	32
Content frame - my folders	37	57	59
Content frame - my favorites	15	21	16
Content frame - search result (upper right)	92	175	182
Content frame - view group	285	25	88
Content frame - view stem	87	178	175
Content frame - view user	101	54	72
Content frame - add member autocomplete (after 2 chars)	33	45	68
Content frame - trace membership	211	21	61
Content frame - view group privileges	312	23	23
Content frame - trace privileges (note 1)	1374	27	25
Content frame - group memberships in other groups	126	37	44
Content frame - view group audit log	430	23	32

Docker Image

(note 1) Also tested with the 2.3.0-2016-12-13 Unicon Docker image, where the time for wheel user was 1767 msec instead of 1374. This build was from before API patch #38, a patch which addressed performance issues with privilege tracing.

Comparison with existing production version 1.6.3

Function	msec (wheel user)	msec (privileged user)	v1.6.3 (wheel user)	v1.6.3 (privileged user)
SSO Login	85	98	78	56
SP Start Session	41	39	41	36
Content frame - home	305	240	182	155
Open Stem in Directory Tree	48	1523	152	186
Content frame - search result (note 1)	163	3516	14525	25639
Content frame - view group (note 2)	649	483	98	50
view group list members	-	-	561	37
Content frame - view stem	107	2531	179	480
Content frame - view user	282	1562	340	314
Content frame - trace membership	616	1741	147	38
Content frame - view group privileges	189	168	151	38
Content frame - group memberships in other groups	122	122	66	60

(note 1) Changes in our sources.xml configuration led to great improvement in search time between 1.6.3 and 2.3.0

(note 2) The group page in the v2 new UI includes the list of group members in the group page, whereas the membership listing is a separate page in 1.6.3.

Analysis

For some pages, there is a large difference between the Docker image and our installation. This is likely due to the much larger data set: We have 23,000 group, 168,000 members, and 845,000 memberships in our development installation, while the Docker setup has 43 groups, 1044 members, and 5175 memberships.

Some of the unusually low response times for non-wheel users are not due to good performance, but rather represent an error response due lack of access. This is especially true for the unprivileged user, which explains why some of the responses are faster for an unprivileged user than for a privileged one.

Some functions show a large difference between wheel and non-wheel access. This is likely due to the additional privilege checking performed for users, which is bypassed for wheel members. The biggest differences are in the folder tree browser (in the lower left of the page), viewing stems and users, tracing memberships, and viewing the list of search results. The slowness is particularly noticeable in the folder tree browser, where it's slow to the point of being impractical for users as a way to navigate to a group. In the Docker installation, differences can be seen between wheel and users for most of the same functions, but the difference is less pronounced.

In the past, we had looked into the folder tree performance issue. Adding additional Oracle indexes would improve the speed by about 33%, but the response per folder was still over 1 second. Note that this response is with the Grouper setting security.show.folders.where.user.can.see.subobjects = false. Without this setting, the response would be around 3 seconds per folder list.

The speed difference for a general subject search between wheel and normal users is also notable. We did not do much further analysis to understand the cause. It does not seem strongly correlated with the number of results; even queries that return zero subjects take over 1.5 seconds to return the response for a non-wheel user.

Comparing a v1.6.3 release with similar data, not all of the new UI functions mapped well to pages in the older UI. Where there were similar functional pages between versions, there was a similar performance in most pages for a wheel user. Viewing a stem or user was faster. Viewing the home page or a group's membership in other groups was slower, but still in a sub-300 millisecond response that would not be very noticeable by the user. Tracing a membership was significantly slower in 2.3.0, going from 150 msec to 600+ msec. Viewing a group was also slower; however, the 1.6.3 version kept group data and membership in separate pages. When combining the responses of the group page plus the membership listing, the timing is comparable to the group page of 2.3.0 which includes membership. For non-wheel users between 1.6.3 and 2.3.0, there are remarkable differences in response time, but these have all been mentioned above as being specific to privilege checking in the new UI. What is notable is that there were not significant differences in response times between wheel and non-wheel users in 1.6.3, whereas now there are.

A zip file of the JMeter test plan is attached, which includes sample data that works against the QuickStart demo application, and shell scripts to run the test in headless mode.

-Chad