

The Sakai OAE architectural sprint will be focussing on performance, scalability and data integrity improvements. The changes required for this work are likely to be substantial, and our motto will be to instrument, model, measure, analyze and optimize based on evidence.

A. Measurement and instrumentation

Main focus between April 19 - May 10

1. First of all, we need to create a *data and usage model* that is representative as to how the system could be used in production. The main goal is to have a *realistic baseline* that we can work against. The model in essence is a set of ratios that can be extrapolated as the load the system can deal with grows.
2. We decide on *performance thresholds* or goals we work against. These thresholds include target average page load time, target number of total users, target number of concurrent users, target average response time for all back-end feeds, etc. The thresholds should be realistic and can be increased/decreased over time. For now, they should be targeted at a single-nakamura, but production environment resembling instance of Sakai OAE.
3. The team *defines the output* that needs to be captured from data loading and user load tests.
4. We decide which *framework(s)* to use for data loading, load testing and reporting. The UI and Server are *instrumented* where necessary to allow us to capture the output that we decided needs to be captured. The team also sets up this testing and reporting environment, and gathers a *first set of numbers* that tell us more about the current state of the product.
5. The UI and QA team creates a set of *basic smoke-test functional UI tests* (Selenium) that allow to automatically check whether base system functionality is still working. This will require writing some basic representative scenarios that cover a large part of the system.
6. The performance and scalability tests are *automatically run on a nightly basis*. The performance numbers are sent back and are automatically reported in a publicly available website that lists the details of the last test as well as evolutionary performance and scalability over time. At the same time, the server and client unit tests, as well as automated UI functional tests are automatically run to ensure that the system is still functional. The testing environment also needs to be available for ad-hoc tests based on branches (not necessarily reporting back)

Work will continue on this during the implementation phase.

B. Strategy exploration

May 11 - May 18

7. There is a combined *server and UI dev meeting* in Ann Arbor, MI where different approaches are discussed based on the current maximum number of users, number of concurrent users and average page load times. A number of experiments are done and validated against the performance testing environment.
8. The meeting also talks about current problems that need to be resolved, like *concurrency and rollback* problems, as well as enhancing some existing services (for example, to support user privacy)
9. The team makes a *plan* for changes and experiments that will be happening.

There are a number of additional goals that will need to be worked into this plan, however the main focus should remain on performance, scalability and data integrity improvements:

- Some of *client-server APIs* will need to be redefined to be friendlier to developers and be closer aligned to the needs of the UI. This does not necessarily remove some of our current generic data feeds, but most likely introduces a middle API layer.
- The last release has been slowed down by the complexity of our stack, and the architectural changes made should keep *developer ease-of-use* in mind.
- Initially, this effort will be focussing on a single-nakamura OAE instance, to optimize the product as much as possible. However, we should have a clear strategy for *scaling up sideways* that should not be comprised upon, and future performance work should include optimization of multi-nakamura OAE instances.
- Without losing sight of some of the core principles of the UI centric development approach, we'll have to put some effort into making *initial server-side rendering* possible, if only for Google Indexing. Promising avenues that allow re-use of the existing UI templates are already being explored.

Data Model

1) Data model

The data model should give us a representative and reproducible baseline that can be used to run all performance and scalability tests against. The goal for the initial data model is for it to be representative of real usage in a 200,000 users institution, which has a 50-50 mix between teaching and research. Later on, this needs to grow substantially when we start looking at scaling the system sideways.

The model itself is a set of ratios that can be extrapolated to scale down/up the model as necessary. The model works in batches of 1,000 users, which all interact with each other, in order to try and keep things as manageable as possible. For each feature, we assume a distribution with a mean, standard deviation and maximum, so we get a representative batch of lots of small ones and a few extremes. We can fire up as many batches as we want in order to get to the desired number of users for a load test.

The data loading script will make all of the parameters configurable, so they can be easily tweaked as more real-life usage data and patterns come in from production and pilot schools. The data loading script will first generate the data based on the parameters, and record this into text files. These text files are then used to the data loading into the system. This will allow us to recreate the same environment over and over again, despite all the random parameters.

A) Users

Type	Details	Student Distribution	Lecturer Distribution	Researcher Distribution
User accounts	- 1000 user accounts	- 50% is male - 50% is female	- 50% is male - 50% is female	- 50% is male - 50% is female
	- User id records the batch number - Randomly generated first names and last names	- 60% is a student	- 10% is a lecturer	- 30% is a researcher
Pubspace and privspace	Published for all users	- 90% leaves unchanged - 10% reorders menu items	- 85% leaves unchanged - 15% reorders menu items	- 85% leaves unchanged - 15% reorders menu items
User account privacy	User accounts can be public or visible to logged in users only	- 70% is public - 30% is logged in users only	- 40% is public - 60% is logged in users only	- 70% is public - 30% is logged in users only
Profile privacy	About Me	- 40% public - 25% logged in users - 25% contacts - 10% private	- 20% public - 25% logged in users - 15% contacts - 40% private	- 60% public - 25% logged in users - 10% contacts - 5% private
	Publications	- 80% public - 10% logged in users only - 9% contacts - 1% private	- 60% public - 25% logged in users only - 10% contacts - 5% private	- 90% public - 5% logged in users only - 3% contacts - 2% private

Type	Details	Student Distribution	Lecturer Distribution	Researcher Distribution
Profile information - Basic Information 70% has additional information in basic info	First Name	100%	100%	100%
	Last Name	100%	100%	100%
	Email	90%	70%	80%
	Preferred Name	15%	5%	5%
	Role	100%	100%	100%
	Department	80%	70%	70%
	College	50%	30%	30%
	Tags and categories (keywords)	30% has categories - Average: 3 - SDev: 2 - Max: 20	20% has categories - Average: 3 - SDev: 2 - Max: 20	40% has categories - Average: 3 - SDev: 2 - Max: 20
		25% has tags - Average: 2 - SDev: 1 - Max: 10	10% has tags - Average: 2 - SDev: 1 - Max: 10	30% has tags - Average: 2 - SDev: 1 - Max: 10

Type	Details	Student Distribution	Lecturer Distribution	Researcher Distribution
Profile information -About Me 40% has information in About Me	About me (text)	70% has about me - Average: 2 - SDev: 2 - Max: 25	50% has about me - Average: 2 - SDev: 2 - Max: 25	70% has about me - Average: 2 - SDev: 2 - Max: 25
	Academic interests (keywords)	60% has academic interests - Average: 3 - SDev: 2 - Max: 50	60% has academic interests - Average: 2 - SDev: 2 - Max: 50	80% has academic interests - Average: 5 - SDev: 3 - Max: 50
	Personal interests (keywords)	60% has personal interests - Average: 3 - SDev: 2 - Max: 50	30% has personal interests - Average: 3 - SDev: 2 - Max: 25	50% has personal interests - Average: 3 - SDev: 2 - Max: 25
	Hobbies (keywords)	50% has hobbies - Average: 3 - SDev: 2 - Max: 50	25% has hobbies - Average: 3 - SDev: 2 - Max: 20	40% has hobbies - Average: 3 - SDev: 2 - Max: 30
	Publications	- Average: 1 - Sdev: 3 - Max: 20	- Average: 3 - Sdev: 3 - Max: 50	- Average: 10 - Sdev: 6 - Max: 100
Profile pictures	Number of profile pictures	- 80% has a profile picture	- 50% has a profile picture	70% has a profile picture

B) Contacts

Type	Details	Student Distribution	Lecturer Distribution	Researcher Distribution
Contacts 70% of people has contacts	Accepted contacts	- Average: 15 - Sdev: 6 - Max: 700	- Average: 5 - Sdev: 3 - Max: 300	- Average: 8 - Sdev: 2 - Max: 800
	Invited contacts	- Average: 1 - Sdev: 1 - Max: 10	- Average: 0 - Sdev: 1 - Max: 10	- Average: 1 - Sdev: 1 - Max: 10
	Pending contacts	- Average: 1 - Sdev: 1 - Max: 10	- Average: 0 - Sdev: 1 - Max: 10	- Average: 1 - Sdev: 1 - Max: 10

The contact type will be normally distributed between the available types classmate, supervisor, supervisee, lecturer, student, colleague, college mate and shared interest

C) World creation and memberships

Group related properties:

Type	Details	Group	Math Course	Basic Course	Research Pro- ject	Research Sup- port Group
Total	Total number of each world type	1500	15	50	220	350
Metadata 50% of groups have additional metadata set	Title (Sentence)	100%	100%	100%	100%	100%
	URL Reflects batch number	100%	100%	100%	100%	100%
	Description (text)	40% has descrip- tion - Average: 2 - SDev: 2 - Max: 25	70% has descrip- tion - Average: 2 - SDev: 2 - Max: 25	60% has descrip- tion - Average: 2 - SDev: 2 - Max: 25	75% has descrip- tion - Average: 2 - SDev: 2 - Max: 25	70% has descrip- tion - Average: 2 - SDev: 2 - Max: 25
	Tags and catego- ries	40% has tags - Average: 3 - SDev: 1 - Max: 20	50% has tags - Average: 3 - SDev: 1 - Max: 20	50% has tags - Average: 3 - SDev: 1 - Max: 20	80% has tags - Average: 3 - SDev: 1 - Max: 20	70% has tags - Average: 3 - SDev: 1 - Max: 20
		40% has catego- ries - Average: 2 - SDev: 2 - Max: 10	50% has catego- ries - Average: 2 - SDev: 2 - Max: 10	60% has catego- ries - Average: 2 - SDev: 2 - Max: 10	70% has catego- ries - Average: 2 - SDev: 2 - Max: 10	60% has catego- ries - Average: 2 - SDev: 2 - Max: 10

Type	Details	Group	Math Course	Basic Course	Research Pro- ject	Research Sup- port Group
Profile picture	Number of profile pictures	30%	30%	35%	60%	55%
Visibility	Public	40%	20%	20%	45%	40%
	Logged In Users Only	20%	25%	35%	35%	30%
	Members Only	40%	55%	45%	20%	30%
Joinability	Anyone can join	40%	5%	5%	10%	10%
	People can re- quest to join	30%	15%	15%	20%	20%
	Managers add people	30%	80%	80%	70%	70%
Group Member- ship	Distribution of other groups be- ing a member of this	Average: 2 Sdev: 1 Max: 25	Average: 1 Sdev: 1 Max: 20	Average: 1 Sdev: 1 Max: 20	Average: 2 Sdev: 1 Max: 25	Average: 2 Sdev: 1 Max: 25
Group Library Size	Few (<50)	60%	30%	40%	25%	40%
	Medium (>50 <500)	30%	50%	50%	50%	45%
	Lots (>500)	10%	20%	10%	25%	15%

User related properties:

Type	Details	Student Distribution	Lecturer Distribution	Researcher Distribution
Number of memberships	Few (< 5)	30%	40%	20%
	Medium (>5 <20)	50%	40%	50%
	Lots (>20)	20%	20%	30%
Group <i>Size distribution:</i> Small (<50): 80% Medium (>50 <200): 15% Large (>200): 5%	Creator	35%	20%	45%
	Manager - Average: 5 - Sdev: 2 - Max: 50	35%	20%	45%
	Member - Average: 10 - Sdev: 5 - Max: 650	40%	20%	40%

Type	Details	Student Distribution	Lecturer Distribution	Researcher Distribution
Math Course <i>Size distribution:</i> Small (<50): 40% Medium (>50 <200): 40% Large (>200): 20%	Creator	0%	95%	5%
	Lecturer - Average: 3 - Sdev: 2 - Max: 10	5%	80%	15%
	Teaching Assistant - Average: 5 - Sdev: 2 - Max: 20	20%	60%	20%
	Student - Average: 50 - Sdev: 15 - Max: 800	90%	5%	5%
Basic Course <i>Size distribution:</i> Small (<50): 50% Medium (>50 <200): 40% Large (>200): 10%	Creator	0%	95%	5%
	Lecturer - Average: 3 - Sdev: 2 - Max: 10	5%	80%	15%

Type	Details	Student Distribution	Lecturer Distribution	Researcher Distribution
	Teaching Assistant - Average: 5 - Sdev: 2 - Max: 20	20%	60%	20%
	Student - Average: 50 - Sdev: 15 - Max: 800	90%	5%	5%
Research Project <i>Size distribution:</i> Small (<50): 90% Medium (>50 <200): 9% Large (>200): 1%	Creator	35%	15%	50%
	Lead Researcher - Average: 2 - Sdev: 1 - Max: 10	35%	15%	50%
	Researcher - Average: 5 - Sdev: 2 - Max: 15	35%	15%	50%
	Research Assistant - Average: 3 - Sdev: 2 - Max: 20	40%	15%	45%

Type	Details	Student Distribution	Lecturer Distribution	Researcher Distribution
	Contributor - Average: 2 - Sdev: 1 - Max: 20	50%	25%	25%
	Evaluator - Average: 2 - Sdev: 1 - Max: 30	5%	70%	25%
Research Support Group <i>Size distribution:</i> Small (<50): 90% Medium (>50 <200): 9% Large (>200): 1%	Creator	35%	15%	50%
	Participant - Average: 5 - Sdev: 3 - Max: 30	35%	15%	50%
	Lurker - Average: 2 - Sdev: 1 - Max: 50	40%	20%	40%

D) Pooled content creation

Overall: There will be a total of 10,000 pooled content items per batch of 1,000 users.

Type	Details	Files	Sakai Docs	Links
Total	10,000 pooled content items in total	45%	40%	15%
Metadata 50% of all pooled content items will have metadata other than title and copy-right	Title (sentence)	100% Average: 2 Sdev: 1 Max: 15	100% Average: 2 Sdev: 1 Max: 15	100% Average: 2 Sdev: 1 Max: 15
	Description (text)	60% Average: 2 Sdev: 1 Max: 25	60% Average: 2 Sdev: 1 Max: 25	60% Average: 2 Sdev: 1 Max: 25
	Tags and categories (keywords)	80% has tags Average: 3 Sdev: 1 Max: 20	80% has tags Average: 3 Sdev: 1 Max: 20	80% has tags Average: 3 Sdev: 1 Max: 20
		60% has categories Average: 2 Sdev: 1 Max: 10	60% has categories Average: 2 Sdev: 1 Max: 10	60% has categories Average: 2 Sdev: 1 Max: 10
	Copyright (equal distr.)	100%	100%	100%

Type	Details	Files	Sakai Docs	Links
Permissions	Public	30%	30%	70%
	Logged in users	30%	30%	20%
	Private	40%	50%	10%
Comments	Comments on the content	30% Average: 3 Sdev: 1 Max: 25	50% Average: 4 Sdev: 2 Max: 50	25% Average: 2 Sdev: 1 Max: 20

User-related properties:

Type	Details	Student	Lecturer	Researcher
Content creations per user	Distribution for content creation	30%	35%	35%
Personal library total	Few (<50)	25%	20%	15%
	Medium (>50 <500)	40%	60%	50%
	Lots (>500)	35%	20%	35%

Files:

Type	Details	Distribution
Type	Images	25%
	Video	5%
	PDF	20%
	Doc	15%
	Other office	15%
	Other files (zip, etc.)	20%
Size Small = < 100KB Medium = > 100KB < 2MB Large = > 2MB	Images	Small: 25% Medium: 50% Large: 25%
	Video	Small: 5% Medium: 20% Large: 75%
	PDF	Small: 20% Medium: 60% Large: 20%

Type	Details	Distribution
	Doc	Small: 20% Medium: 60% Large: 20%
	Other office	Small: 20% Medium: 60% Large: 20%
	Other files (zip, etc.)	Small: 40% Medium: 20% Large: 40%

Sakai Documents:

Type	Details	Distribution
Pages per document	Single page	50%
	Multiple pages	50% Average: 3 Sdev: 2 Max: 25
Page composition	Number of rows	Average: 2 Sdev: 1 Max: 10
	Number of columns	Average: 2 Sdev: 1 Max: 3
	Widgets per cell	Average: 3 Sdev: 1 Max: 10
Widgets	Text	50% Average: 15 Sdev: 5 Max: 1000

Type	Details	Distribution
	Title	5% Average: 3 Sdev: 1 Max: 25
	Content	15% Average: 2 Sdev: 1 Max: 25
	Discussion	10% Average: 2 Sdev: 1 Max: 10
	Comments	5% Average: 6 Sdev: 3 Max: 25
	Google Map	5%
	RSS	4%
	Basic LTI	5%
	Google Gadget	1%

Links:

Type	Details	Distribution
Link type	YouTube	30%
	Vimeo	10%
	Flickr	5%
	Google Map	5%
	Other	50%

E) Collections

Overall: There will be 500 collections per batch of 1,000 users

Type	Details	Distribution
Creator	Creator distribution for collections	Students: 25% Lecturers: 30% Researchers: 45%
Collection Size	Small (<10)	50%
	Medium (>10 <50)	30%
	Large (>50)	20%
Collection Composition	Files	50%
	Sakai Docs	20%
	Links	20%
	Collections	10%
Metadata 50% of all collections will have metadata other than title and copyright	Title (sentence)	100% Average: 2 Sdev: 1 Max: 15
	Description (text)	60% Average: 2 Sdev: 1 Max: 25

Type	Details	Distribution
	Tags and categories (keywords	80% has tags Average: 3 Sdev: 1 Max: 20
		60% has categories Average: 2 Sdev: 1 Max: 10
	Copyright (equal distr.)	100%
Permissions	Public	40%
	Logged in users	20%
	Private	40%
Comments	Comments on the collection	30% Average: 3 Sdev: 1 Max: 25

Type	Details	Distribution
User membership	Managers	Average: 2 Sdev: 1 Max: 15
	Members	Average: 5 Sdev: 3 Max: 50
Group Membership 40% of all collections have a group as a member	Managers	Average: 1 Sdev: 1 Max: 5
	Members	Average: 2 Sdev: 1 Max: 10

F) Content sharing

Files:

Type	Details	Distribution
Usage	Users Only	45%
	Groups Only	45%
	Mixed	10%
Users	Manage	Average: 3 Sdev: 2 Max: 25
	Edit	Average: 1 Sdev: 1 Max: 20
	View	Average: 5 Sdev: 3 Max: 500

Type	Details	Distribution
Groups	Manage	Average: 1 Sdev: 1 Max: 3
	Edit	Average: 1 Sdev: 1 Max: 5
	View	Average: 2 Sdev: 1 Max: 15

Sakai Docs:

Type	Details	Distribution
Purpose	Area Only	45%
	Sakai Doc Only	55%
	Mixed	5%
Usage	Users Only	45%
	Groups Only	45%
	Mixed	10%

Type	Details	Distribution
Users	Manage	Average: 3 Sdev: 2 Max: 25
	Edit	Average: 3 Sdev: 2 Max: 25
	View	Average: 5 Sdev: 3 Max: 250
Groups	Manage	Average: 1 Sdev: 1 Max: 3
	Edit	Average: 2 Sdev: 1 Max: 5
	View	Average: 2 Sdev: 1 Max: 15

Links:

Type	Details	Distribution
Usage	Users Only	45%
	Groups Only	45%
	Mixed	10%
Users	Manage	Average: 2 Sdev: 1 Max: 10
	Edit	Average: 1 Sdev: 1 Max: 3
	View	Average: 4 Sdev: 2 Max: 50
Groups	Manage	Average: 1 Sdev: 1 Max: 3
	Edit	Average: 1 Sdev: 1 Max: 3
	View	Average: 2 Sdev: 1 Max: 15

Collections:

Type	Details	Distribution
Usage	Users Only	45%
	Groups Only	45%
	Mixed	10%
Users	Manage	Average: 2 Sdev: 1 Max: 10
	Edit	Average: 1 Sdev: 1 Max: 3
	View	Average: 4 Sdev: 2 Max: 50
Groups	Manage	Average: 1 Sdev: 1 Max: 3
	Edit	Average: 1 Sdev: 1 Max: 3
	View	Average: 2 Sdev: 1 Max: 15

G) Areas

// TODO

H) Messages

All users will already have a certain number of messages, including contact invitations, group joins, content sharing, etc. Other than these messages, users can also send normal messages to other users and groups. Overall, we will assume that 60% of people send messages to others

Type	Details	Distribution
Target	Single user	50%
	Single group	20%
	Multiple users	15%
	Multiple groups	5%
	Multiple groups and users	10%
Metadata	Title (Sentence)	Average: 6 Sdev: 2 Max: 50
	Body (Text)	Average: 8 Sdev: 5 Max: 300
	Multiple recipients	Average: 3 Sdev: 2 Max: 30

Type	Details	Distribution
Total messages sent	Student	Average: 25 Sdev: 5 Max: 200
	Lecturer	Average: 10 Sdev: 2 Max: 200
	Researcher	Average: 30 Sdev: 10 Max: 400
Total messages read	Student	Average: 95% Sdev: 5% Max: 100%
	Lecturer	Average: 90% Sdev: 4% Max: 100%
	Researcher	Average: 90% Sdev: 5% Max: 100%

2) Usage model

// TODO

3) Reference environment

The reference environment will be an example deployment that we work towards and do our optimizations again. This environment need to be set up with continuous access from all team members, so performance tests can be run on it. We might want to set up multiple of these to allow simultaneous performance testing.

We are currently using the following specs for our performance testing environment:

Red Hat Enterprise Linux Server release 6

2 CPUs

4GB memory, 35GB total disk space, mount NFS

- 2 virtual networks (1 routed VLAN, 1 private VLAN)

VMware vSphere

Java 1.6 Update 27

Oracle 11g backend

Red Hat Enterprise Linux Server release 5.6

- 1 vCPUs

- 2GB of RAM

- 35GB disk

- 2 virtual networks (1 routed VLAN, 1 private VLAN)

At app server and Oracle underlying hardware is:

- Dell PowerEdge R810

- Intel Xeon CPUs (x7560 @ 2.27GHz)

- 224 GB of RAM

- 2 x 10GB network adapters

```
JVM 1.6 Update 27
-Xms4g -Xmx4g
-XX:MaxPermSize=512m
-XX:+UseParNewGC
-XX:+UseConcMarkSweepGC
```

We need to decide on and set up a final reference environment as soon as possible. Therefore, we need to answer the following questions:

- a) Which database do we target
 - b) What are the app server specifications
 - c) What are the database server specifications
 - d) Do we run solr on a separate server and what are its specifications
 - e) What are the preview processor server specifications
 - f) Do we advice running apache in front of OAE
- etc.

ACTION ITEMS:

- Move back to 1 app server
- Move back to 1 solr server
- Set up a new VM for PostgreSQL
- Give team leads access to this Indiana environment

- Easier to get to the bottlenecks if it's just one
- Cloud copies to run tests again => script against that

Daniel recommends us to not over-spec the reference environment, so it becomes easier to find bottlenecks. He also recommends running all components on separate VMs or physical machines, to make it easier to find the culprit of a given issue.

Reporting

Our reporting environment should report the following metrics:

Level	Type	Metrics
1	Overall	Average page load time cache (evolutionary)
		Average page load time from cache (evolutionary)
		Number of concurrent users supported (evolutionary)
		Number of users supported (evolutionary)
2	General	Average Hits/second
		Total Hits
		Average Pages/second
		Total Pages
		Average Throughput
		Total Throughput
		Average CPU usage
		Maximum CPU usage

Level	Type	Metrics
	UI	Average page load time (no cache) for all individual pages
		Average page load time (cache) for all individual pages
		Average start render time (no cache) for all individual pages
		Average start render time (cache) for all individual pages
		Average number of requests (no cache) for all individual pages
		Average number of requests (cache) for all individual pages
		Average total page size (no cache) for all individual pages
		Average total page size (cache) for all individual pages
	Search	Average query time for all individual search feeds (large result set)
		Average query time for all individual search feeds (small result set)
		Number of search queries (+list) for all individual pages
		Size of the Solr index
		Search performance degradation

Level	Type	Metrics
	API	Average response time for all individual APIs
		Number of API calls (+list) for all individual pages
		Average number of database queries for all individual pages
		Average response time for all database queries + max
		API performance degradation
3	Raw Data	API Execution Time
		JVM Measures
		Method Breakdown by Execution Time
		Response Time Hotspots
		SQL Executions
		Host System Network

Thresholds

In terms of thresholds, there are 3 levels. Short term thresholds we work towards during the next sprint, achievable near-long term thresholds and long-term thresholds. In order to define the short term thresholds, we first need to gain understanding of the current performance metrics of the system. These thresholds and targets will be defined as part of the plan creation during the Ann Arbor meeting. Listed below are the near-long term thresholds we should be working towards over the next 6 months.

Level 1

Initial target for total number of users: 30,000

Target level of concurrency: 50%

Lower level of concurrency: 10%

Target average (first 95%) page load time: 4s

Upper limit (first 95%) page load time: 8s

Target average (first 95%) page load time from cache: 2s

Upper limit (first 95%) page load time from cache: 6s

Error threshold: 1%

The number of supported users is based on the point in time where page load time upper limits are crossed, the error threshold is crossed or search and API upper limits are crossed.

Level 2

Average (first 95%) search time: 500ms

Upper limit (first 95%) search time: 2s

Average (first 95%) API time: 500ms

Upper limit (first 95%) API time: 2s

Level 3

Average app server CPU usage: 45%

Max app server CPU usage: 80%

Average solr server CPU usage: 45%

Max solr server CPU usage: 80%

Automated UI Tests

// TODO