Title	VLO benchmarking and scalability
Version	1
Author(s)	Twan Goosen
Date	2018-07-11
Status	Final
Distribution	SCTCC / EOSC-hub
ID	CE-2018-1254

CLARIN

1 Introduction

The Virtual Language Observatory (VLO) is one of the services on CLARIN's EOSC-hub roadmap[1]. The service is part of CLARIN's metadata infrastructure and consists out of three separate components:

1. An Apache Solr[2] instance with a dedicated schema and locally stored searchable index.
2. A front-end, implemented as a Wicket[3] application running in Apache Tomcat[4], which is user facing and allows for searching and browsing through the documents available from the Solr instance .
3. The *VLO Importer* application which semantically processes sets of 'harvested' metadata in the CMDI format[5] and maps encountered values to a predefined set of fields, then submits these to the Solr instance. In CLARIN's schedule, as currently define, the import task is carried out three times per week in production.

A fourth, closely related component is the *OAI-PMH harvester*, which retrieves metadata records from various pre-defined sources and makes them available as input to the VLO Importer. The harvester currently runs three times per week, harvesting from different sources.

At the time of writing, all three components of the VLO and the OAI-PMH harvester have been deployed, for production purposes, on a VPS leased from HostEurope[6], a commercial hosting provider. The VPS has been provisioned with (distributed) solid-state drive (SSD) storage in order to achieve high performance (IOPS). However, this setup is not compatible with CLARIN's transition towards a containerised infrastructure (using Docker). In addition to implementing enhanced integration and functionality, completing this transition for all services involved is an explicit objective on CLARIN's EOSC-hub roadmap. Therefore obtaining a new deployment host that meets the conditions of the desired deployment and administration workflow is an immediate priority. Based on CLARIN's experiences with HostEurope in terms of performance, availability/uptime and support, the initial investigation to this end is limited to the HostEurope Root Server (RS) offer, which offers a level of control that enables the running of Docker in addition to dedicated storage and specification levels that, at face value, seem fit for running the VLO and OAI-PMH harvester as production services. This document presents a benchmarking effort intended to verify that the RS can provide a comparable and consistent real-world performance. In addition, it reports on the

[1] https://office.clarin.eu/v/CE-2018-1175-EOSC-hub-task71-roadmap.pdf
[2] http://lucene.apache.org/solr/
[3] http://wicket.apache.org/
[4] https://tomcat.apache.org/
[5] https://www.clarin.eu/cmdi
[6] https://www.hosteurope.de/en/

scalability of this solution and, on basis of this and a number of growth scenarios for the service, its estimated sustainability.

2 Benchmarking: import performance

2.1 Method

The total clock time required to run a complete import on a set of CMDI records (of varying sizes, see below) divided by the number of imported records was taken to derive an *import rate* (*records/sec*), and used as the measure of performance in these tests. A higher rate indicates better performance.

Server specifications

Server label	CPUs	RAM	Storage type	Description
VPS-PROD	8x 2.40GHz	32GB	SSD	VPS (current production host)
RS-M	4x 2.10GHz	17GB	HDD (non-SSD)	Root server 'M' (used for testing)
RS-XL	8x 2.10GHz	39GB	SSD	Dedicated new root server 'XL'

All servers provided by HostEurope.

Data sets

These are the datasets imported in the various benchmarking trials.

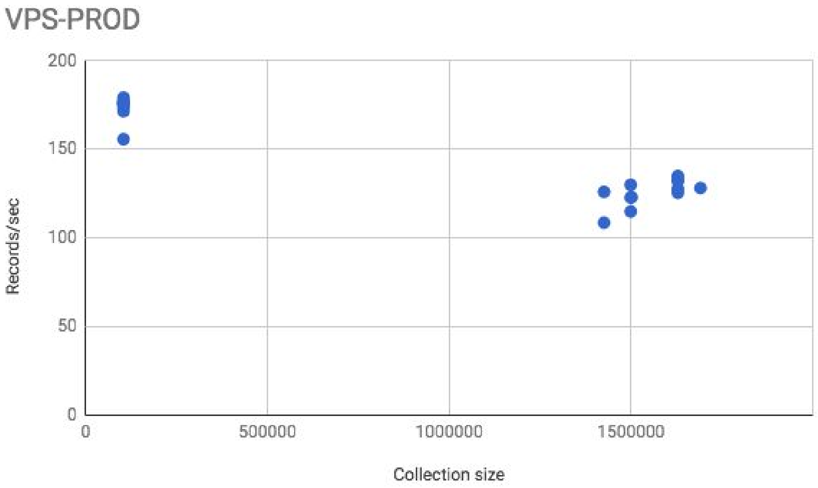
Dataset	Nr of records	Description
A	104k	A test set containing two large collections of ~50k records each
B	1.69mio	Records harvested from CLARIN, Europeana and other sources; a mix of small and large collections

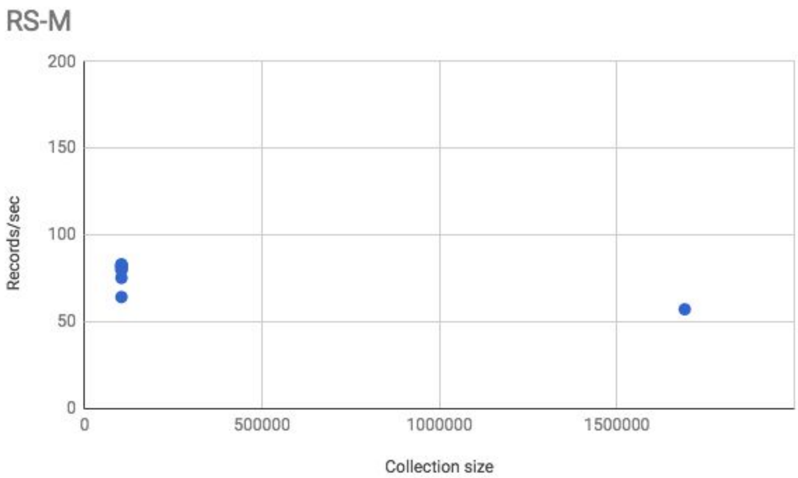
Note that the exact size of the datasets varies because the benchmarks always involved the last imported set; the exact set of available records varies slightly from day to day.

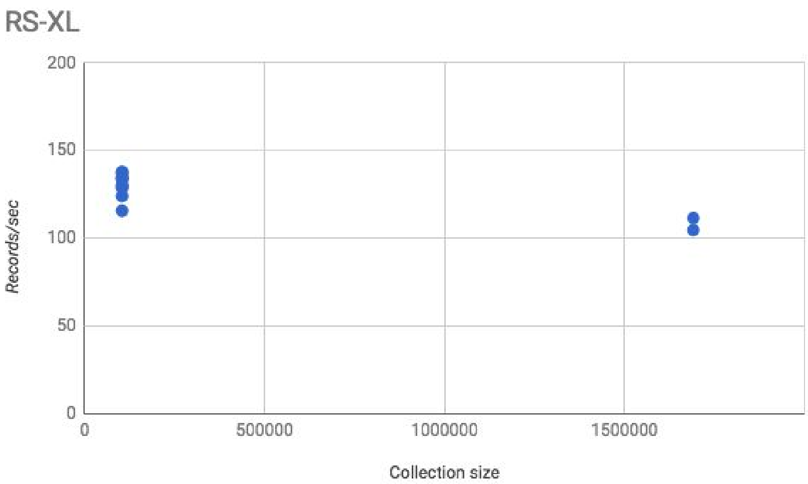
2.2 Results

Server	Average rate (set A)	Average rate (set B)
VPS-PROD	174 (N=12; SD=6.1)	126 (N=12; SD=7.5)
RS-M	79 (N=10; SD=5.8)	57 (N=1)
RS-XL	131 (N=10; SD=6.7)	108 (N=2; SD=4.8)

The scatter plots below show all data points per server. Note that the relation between collection size and import rate is not under investigation here. Collection size is shown to allow for a grouping of Set A and Set B runs.







2.3 Conclusion

Neither Root Server instance (RS-M/XL) reaches the import rate of the current production server VPS-PROD. However, for the production data set (set B) the performance of RS-XL is fairly close to that of VPS-PROD but there is a clear difference ($AVG_{RS\text{-}XL} \leq AVG_{VPS\text{-}PROD} - 2.4 * SD_{VPS\text{-}PROD}$).

We cannot draw any conclusions about the defining factor(s) leading to the observed performance. The two root servers differ significantly on at least two major dimensions: processing power (number of cores) and storage type (SSD vs non-SSD), both of which may contribute to the difference. On the other hand, VPS-PROD and RS-XL have comparable specifications. The difference might be explained by the faster CPUs of the VPS (2.1 GHz vs 2.4 GHz).

3 Benchmarking: http response times

2.1 Method

For the three servers described in section 2, we carried out 100 successive requests of each of the following type:

Type	Query
root	/
search	/search
query	/search?q=test
record	/record?docId=http_58__47__47_hdl.handle.net_47_11022_47_0000-0000-5050-B
about	/about
help	/help

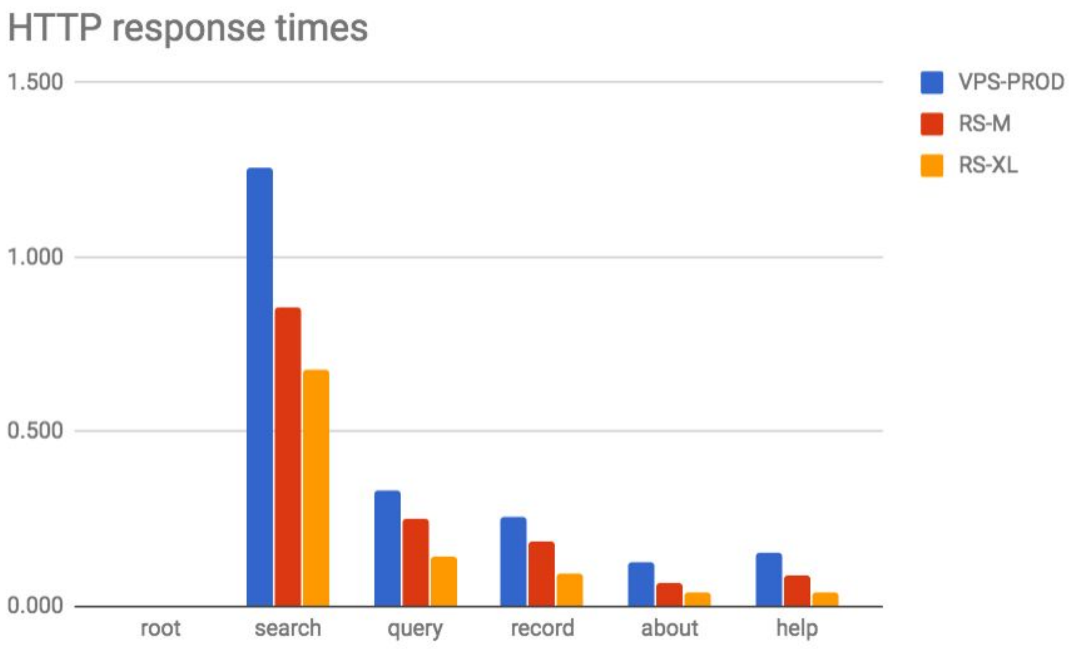
The same data set was present in all three VLO instances. The requests were carried out at a remote host (located at the MPCDF data centre) using the following command:

wget -O /dev/null -q ${URL}

3.2 Results

The following table shows the average response times for the various queries for each of the server. Reported numbers are total time to complete the wget command (see above) reported by the *time* command of the *Bash* shell at a remote host.

Server	root	search	query	record	about	help
VPS-PROD	0.000	1.254	0.331	0.259	0.129	0.154
RS-M	0.000	0.857	0.253	0.187	0.067	0.087
RS-XL	0.000	0.678	0.141	0.092	0.041	0.042



3.3 Conclusion

Both root servers consistently show shorter response times than the production VPS. We cannot rule out that this is at least partially the result of a higher load of the latter due to its production status and consequently higher average load. However, given these observations it seems likely that the root servers are more than likely to perform as good as, or even better than the production VPS under a comparable load.

4 Overall conclusions

We conclude that the import performance levels of VPS-PROD and RS-XL are similar enough to support a transition to the latter in order to support a fully containerised setup. Furthermore, the root servers show favourable response times for the VLO web application compared to the current production server.

5 Evaluation of scalability and service sustainability

The main factor that has to be considered in the context of VLO scalability and sustainability is the number of records that are to be harvested and imported on a regular basis. Secondary to the number of records, the *average size of a record* and the *number of collections* are also factors with a noticeable impact. The mains areas of impact are:

- **Import time**: the time required to execute the VLO importer
- **Metadata storage**: disk space required to store the metadata records
- **Index storage**: disk space required for indexing by Solr, including buffer space
- **Additional storage:** storage for archiving, statistics, administration and inspection

The following table shows three states of the VLO ecosystem, with size of the imported metadata set (expressed in number of records) as the dependent variable and where possible an estimated quantification for the impact variables mentioned above.

	Current	Expected in ~2y	Estimated upper bound for next ~5y
Total record count	1.6M	2.5M	5M
STORAGE			
Metadata	100 GB	150 GB	300 GB
Index storage (including buffer)	160 GB	240 GB	480 GB
Additional storage	50 GB	250 GB	500 GB
Total storage	412 GB	735 GB	1370 GB
IMPORT TIME	3.5h	5h	10h

Note that we have not yet measured the index performance and how it is impacted by the size of the index. However we expect this to be a relatively minor factor in terms of scalability. However it is covered in the next section.

5.1 Mitigation strategies

For each of the impact areas, there are strategies to deal with an exhaustion of the available resources. In general, these mitigations/solutions require additional storage and/or processing resources.

Import time

If import time should become an issue due to the number of records and/or the average size of the records to be imported, it can be reduced by increasing the processing power. The import task can be parallelised quite well, and in fact the importer process is multithreaded to a degree automatically tuned to the number of available cores. More, and faster cores will directly reduce the import time (more or less linearly), especially in the case of large collections, where threads have to be merged relatively infrequently.

Storage

All three 'types' of storage that are required by the components of the VLO are, in the current setup, constrained by the storage provided through the VPS or Root Server package. This is limited, and in fact the total amount of storage is exceeded in the right most scenario in the above table while the middle scenario already pushes the limits. The following strategies are available in case we reach the limit of the available amount of local storage:

1. Use external storage and/or a separate host for the 'additional storage'. In particular archived data has to be available externally but does not have to be available to the other components of the VLO, and can be available with lower performance (i.e. slower retrieval is acceptable).
2. Use external storage for the metadata records. Harvest them on a separate volume, that can for example be mounted over the network. The data has to be available to the VLO web application, and preferably with (very) good performance. A high-performance (local) network mount is likely to be sufficient.

3. Deploy Solr to a separate host with its own local storage. This may have an effect, performance-wise, on the communication between the Solr instance on the one hand and both the web application and importer on the other but this may well be acceptable, especially if the dedicated Solr host is in the same local network. The Solr index can also be distributed over various instances ('shards') which provides some flexibility when it comes to using local storage[7].

Index performance

If the Solr index performance starts to degrade significantly at some point due to the number of indexed records, we can consider transitioning to a distributed Solr setup. Solr supports setting up a distributed network of nodes (*SolrCloud*) for exactly this purpose[8]. One or more Solr nodes can be configured, after which indexing requests and queries get distributed over the various nodes, thus distributing the load.

[7] https://lucene.apache.org/solr/guide/6_6/shards-and-indexing-data-in-solrcloud.html
[8] https://lucene.apache.org/solr/guide/6_6/getting-started-with-solrcloud.html