How long will your Hyrax batch ingest take?

~ Sharing benchmarks can help us find out ~

Drew Myers, Sadie Roosa, Jason Corum, Henry Neels

WGBH Educational Foundation, Boston MA

Questions?
Contact Drew Myers
afred (samvera.slack.com)
andrew_myers@wgbh.org

Spoiler: We don't know…

… just how long it will take you to get your troves of precious data into your Hyrax application.

There are many ways to configure Hyrax, and many ways to deploy it. This means your mileage may vary widely.

We underestimated

We were building a new repository system on Hyrax[1], but when it came time to estimate how long it would take to ingest hundreds of thousands of records in our production environment, we had little information to go on.

It turned out to take a lot longer than we had guessed, and we weren't sure why.

Stuff we didn't know

As usual, Samvera and Fedora community members were ready to offer their advice and experiences, which was a big help.

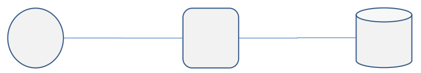
But we did not have a very good way to compare results from our implementation with results from others.

Put plainly, when it came to performance, ***we didn't know what was good, bad, or average*** for our given stack choices.

Wouldn't it be cool?

A simple convention for comparing common Hyrax benchmarks across implementations would go a long way in helping to answer questions we had, like…

- What, if anything, is wrong with our set up?
- How long can we expect ingest to take on *our* system?
- Who', if anyone, has the performance we're looking for, and how did they get there?

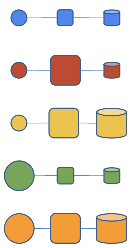


Hyrax
Rails application running on AWS EC2 instance.

Fedora/Solr
Java applications running on AWS EC2 instance.

Fedora DB
MySQL on AWS RDS instance, used by Fedora

We created **5 stacks**[2] with different combinations of **large** and **small**[3] instances for the 3 components we wanted to test.



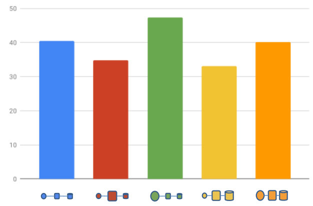
small Hyrax **small** Fedora/Solr **small** Fedora DB

small Hyrax **LARGE** Fedora/Solr **small** Fedora DB

small Hyrax **LARGE** Fedora/Solr **LARGE** Fedora DB

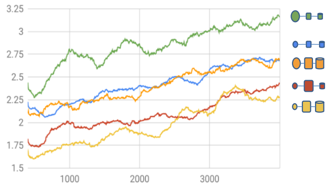
LARGE Hyrax **small** Fedora/Solr **small** Fedora DB

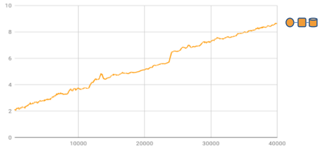
LARGE Hyrax **LARGE** Fedora/Solr **LARGE** Fedora DB



Total time (in minutes) to create 1,000 **GenericWork** objects across different stacks

Running average time (in seconds) to create a single **GenericWork** in a batch of 1,000 across different stacks





Average time (in seconds) to create a single **GenericWork** in a batch of 10,000 (one stack only)

We ran benchmarks

Our test was simple:

- Use Hyrax's default actor stack to insert a stripped down **GenericWork** model 1,000 times across different stacks.
- Measure how long each insert operation takes.
- Measure the total number of **ActiveFedora** objects. (For every new **GenericWork** created, Hyrax creates 3 other objects, e.g. 10 inserts = 40 new objects).

We learned stuff

By comparing our benchmark results across stacks, we learned:

- Where to get more bang for our buck when scaling up resources.
- A sense of how much slower ingest becomes as the repository grows larger.
- A better way to estimate how long ingests will take overall.

A simple tool

The benchmarking interface we wrote is simple and easy to use. It allows developers to:

- Define a procedure to test
- Define measurements they want to take
- Run the procedure N times
- Record the time and all the measurements
- Write the results to a file (CSV)

For all to use

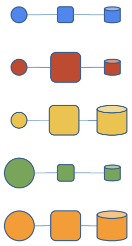
Comparing a set of common benchmarks across our implementations can help Hyrax adopters ***know what is good, bad, or average***.

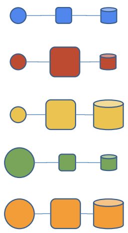
We'd love to turn this into a small Hyrax plugin for everybody. If you're interested in using it, let us know!

[1] – **GitHub:** WGBH-MLA/ams

[2] – **stack** = the resources used in your deployed Hyrax app (we use AWS instances)

[3] – **small** = t2.medium or dbt2.medium; **large** = m4.4xlarge (AWS EC2 instances types)





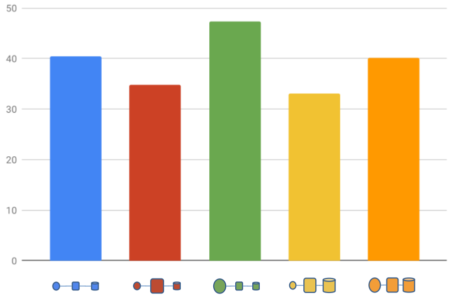
small Hyrax | **small** Fedora/Solr | **small** Fedora DB

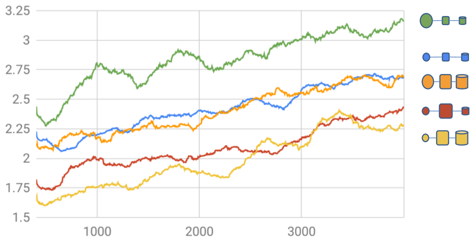
small Hyrax | **LARGE** Fedora/Solr | **small** Fedora DB

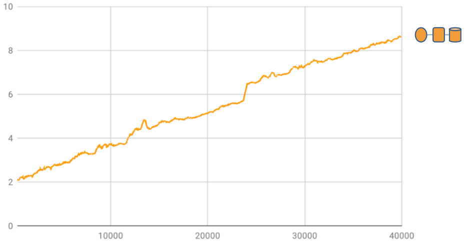
small Hyrax | **LARGE** Fedora/Solr | **LARGE** Fedora DB

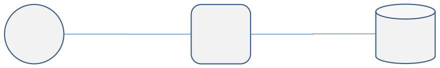
LARGE Hyrax | **small** Fedora/Solr | **small** Fedora DB

LARGE Hyrax | **LARGE** Fedora/Solr | **LARGE** Fedora DB









Hyrax
Rails application
running on AWS
EC2 instance.

Fedora/Solr
Java applications
running on AWS
EC2 instance.

Fedora DB
MySQL on AWS
RDS instance,
used by Fedora