

Deployment Hardware Information

This page has proved very difficult to keep up to date; even so it contains useful information. Later this year it will be replaced with a page suggesting example configurations to address a range of different use cases.

- Stanford, as of Nov 29, 2011
- University of Virginia
- University of Hull
 - Answers to other questions:
- Rock and Roll Hall of Fame and Museum
- Penn State (as of October 2013)
- Notre Dame (October 2012)
- Yale (September 2013)
- UC San Diego (January 2015)

Answers to the following questions may be useful to others:

- How much RAM?
- Number and vintage of CPUs
- Do you run on VMs?
- Do you use a compute provider such as EC2? How's that?
- Do you run Rails and Solr on the same server?
- Do you use JRuby?
- What are your transaction rates - daily, monthly, and peaks?
- What are your search (Solr) v. read (Fedora) v. Write (Fedora + Solr) ratios?
- How many Fedora objects?
 - How big is a typical object?
 - How many Solr fields?
- What will you do differently next time?

Stanford, as of Nov 29, 2011

~ 200,000 Objects

We're currently using VM's from a pool of Xen Hypervisor servers. Each server has the following specs:

- Sun Fire X4600 M2
- 8 quad-core AMD Opteron processors @ 2.3GHz
- 256 GB RAM
- NetApp SAN storage

Here are the VM's we've created, all using Red Hat Enterprise Linux 5, each set replicated in dev, test and production environments:

- Fedora and Solr: 2 CPUs and 6GB memory
- Hydra/Rails: 2 CPUs and 4GB memory

We have Oracle 11G running on its own Sun x4200, AMD Opteron 4core@2.4Ghz, 8GB RAM, RHEL 5.3, Oracle on NFS.

In the next few months, we will move the VMs to a pool of VMWare servers, with each VM running RHEL 6. We will allocate more CPUs to the Fedora VM or create a dedicated Solr VM.

University of Virginia

For [Libra](#), the unmediated self-deposit repository for scholarly work, we have approximately 50 objects total as of Nov 29 2011.

The production environment is a VM running in the University's Information Technology Services production cluster. The OS is Fedora 13 and the VM is currently assigned the following resources:

- 1 CPU
- 2.2 Ghz (max)
- 4 GB memory
- 50 GB local disk

Answers to other questions:

- We **do** run the whole stack on the same server.

- We **do not** use JRuby although we would like very much to JRubyize all Ruby-based apps for easy/safety/stability of deployment and maintenance (this includes our desire to JRubyize our Blacklight-based OPAC, which is a process we will be undertaking in the next few months).
- We **do not** have metrics on transaction rates because the number of objects we are dealing with is quite small, as is the traffic (approx 30 hits per day total).
- A typical object size is 1MB.

University of Hull

The Hydra at Hull systems are implemented on a VM's within a large campus VMWare ESX infrastructure

The test and production servers are, in fact triple VM instances because we separate out Solr, Fedora and the Hydra stack each onto its own machine:

- Fedora: Microsoft Windows Server 2008 R2, 2GB memory
 - This bothered us (though it didn't break) and we've gone to 4GB
 - Intel Xeon E7 2.40 GHz
 - 4 GB RAM
- Solr: Microsoft Windows Server 2008 R2, 6GB memory
 - Intel Xeon E7 2.40 GHz
 - 6 GB RAM
- Hydra (Ruby etc) Red Hat Enterprise Linux 5 (64-bit)
 - Intel Xeon E7 2.40 GHz
 - 8 GB ram

I suppose really there are fourth machines lurking in the background because the SQL stuff is sent out to the University's 'central' SQL cluster. Our dev server actually has all three components on one VM.

Hull is basically a Microsoft 'shop', hence the OS for two of the three machines, but we needed to implement Hydra on Linux to get everything performing as it should.

Last updated: 2013-04-16

Answers to other questions:

- currently we do not use JRuby - although that is an aspiration
- we don't have detailed metrics but Google analytics says we typically get between 700 and 1000 page views per day
- there are about 5000 objects, currently, but that is likely to rise several-fold over the next couple of years
- we use 'managed content' for data and some metadata and so objects are only a few 10s of KB. Content varies from KB to low GB.

Rock and Roll Hall of Fame and Museum

I run 5 VMs on a cluster of two Dell servers. Here are the details about each node:

- PowerEdge R410
- 2 Intel Xeon CPUs (E5620 @ 2.40GHz) with 4 cores each
- 24 GB ram
- CentOS 5 cluster suite
- Xen hypervisor
- 2 CPU, 4 GB ram and ~10 GB of storage per VM

The VMs live on IBM DS3500 SAS disk array, attached via iSCSI. Here's a breakdown of the systems and services on each VM:

- Mysql testing (Ubuntu 10.04.3 LTS)
- Mysql production (Ubuntu 10.04.3 LTS)
- Production server: confluence, jira, Fedora, Solr (CentOS 5)
- Ruby production server (CentOS 5)
- Development server: everything except mysql (CentOS 5)

There is a third physical machine, an IBM 3650 running Red Hat EL 5, which manages disk and tape storage, and hosts a couple of NFS shares that the VMs mount. This is where Fedora is putting its data, as well as the video data stored externally to Fedora.

Other bits:

- no JRuby (yet, at least)
- transaction rates are basically nil right now
- we only have 200 or so objects at this time
 - objects are around 1 MB, only comprising metadata since all the content is stored externally
 - 670 solr fields
- What will you do differently next time?
 - too early to tell!

Penn State (as of October 2013)

- 1 VM for CI
 - runs jenkins
 - 10GB, 4 CPU, RHEL6, 64-bit
 - we are slowly migrating to travis-ci.org
- 2 VM for QA
 - runs tomcat for solr and fedora QA
 - runs apache+passenger to host app
 - remote MySQL dev instance for rails and fedora
 - fronted by hardware load balancer
 - 8GB, 2 CPU, RHEL6, 64-bit
- 2 VM for Staging
 - runs tomcat for solr and fedora Staging
 - runs apache+passenger to host app
 - remote MySQL production instance for rails and fedora
 - fronted by hardware load balancer
 - 8GB, 2 CPU, RHEL6, 64-bit
- 2 VM for Production
 - runs tomcat for solr and fedora Production
 - runs apache+passenger to host app
 - remote MySQL production instance for rails and fedora
 - fronted by hardware load balancer
 - 8GB, 2 CPU, RHEL6, 64-bit
- Notes
 - All datastores reside on HP EVA SAN for OS
 - All Rails Apps, Tomcat Apps, Redis, etc reside on Isilon NAS export
 - All Rails apps deployments migrating to and rbenv, using ruby 2.0.*

Notre Dame (October 2012)

For production we use three servers, each having 32 GB of RAM and two 2.8 GHz CPUs. The servers are deployed as a cluster, using RedHat KVM. We are running Fedora, Solr, and Apache+Passenger+Rails on separate machines.

On the Apache machine we have a separate instance of Apache for each Hydra head (about 6), since each head was developed against a different version of Hydra. Each head is using REE 1.8.6. We switching to Ruby 1.9.3 for all new development, and plan to upgrade all the heads to use it over time.

We also have a similar setup for pre-production testing. We use Jenkins for CI and deploy using Capistrano.

Overall the repository contains about 11,500 objects using about 44 GB on a NetApp SAN.

Yale (September 2013)

Production environment - entire Hydra stack on single server: Dell PE R710 – with 72Gb memory, 2 x Intel Xeon X5660 2.8Ghz processors, and 2 x 160GB internal HDD's. Data storage (the fedora_store for objects and datastreams) is provided by an NFS mounted volume from the Library's SAN.

We plan to split this into ingest taking place on the above hardware virtually 24/7 and then we will have a VM that mirrors the setup and offers a read only SOLR index that is updated periodically throughout the day. This will be our public front end. When the time comes, we will load balance by replicating this VM and moving to a SOLR cloud setup as well as a clustered MySQL instance.

In addition we have a number of VMs so that each development team member has their own dedicated box with a mirror of the production setup, just less resources attached.

UC San Diego (January 2015)

- We have three separate environments:
 - Production
 - Staging: for manual testing including reviewing release candidates (auto-deployed when release branches are created)
 - QA: shared development environment (auto-deploys develop branch after every commit)

- All environments use:
 - Separate Apache 2.2.x load-balancer (shared with non-Hydra apps, CMS, etc.)
 - Separate PostgreSQL 9.x database server
 - NFS-mounted Isilon SAN for storage
- Each environment contains 3 VMWare VMs:
 - Hydra Head (Rails):
 - 1 CPU, 8GB RAM
 - Apache 2.2.x, Passenger/Rails 4.x, Ruby 2.1.x
 - Solrizer
 - Hydra Tail (Tomcat with Solr and DAMSRepo):
 - 2 CPUs, 12GB RAM
 - Tomcat 7.x with 8GB RAM
 - Ingest (Tomcat with custom Java ingest webapp):
 - 1 CPU, 8GB RAM
 - Tomcat 7.x with 4GB RAM
- Repository data:
 - 104K repository objects (74K simple objects, 8K complex objects, 22K authority records)
 - 6TB content files
 - Usage varies: 50-350K searches/month and 100-200K object views/month