The climateprediction.net infrastructure

Andy Bowery

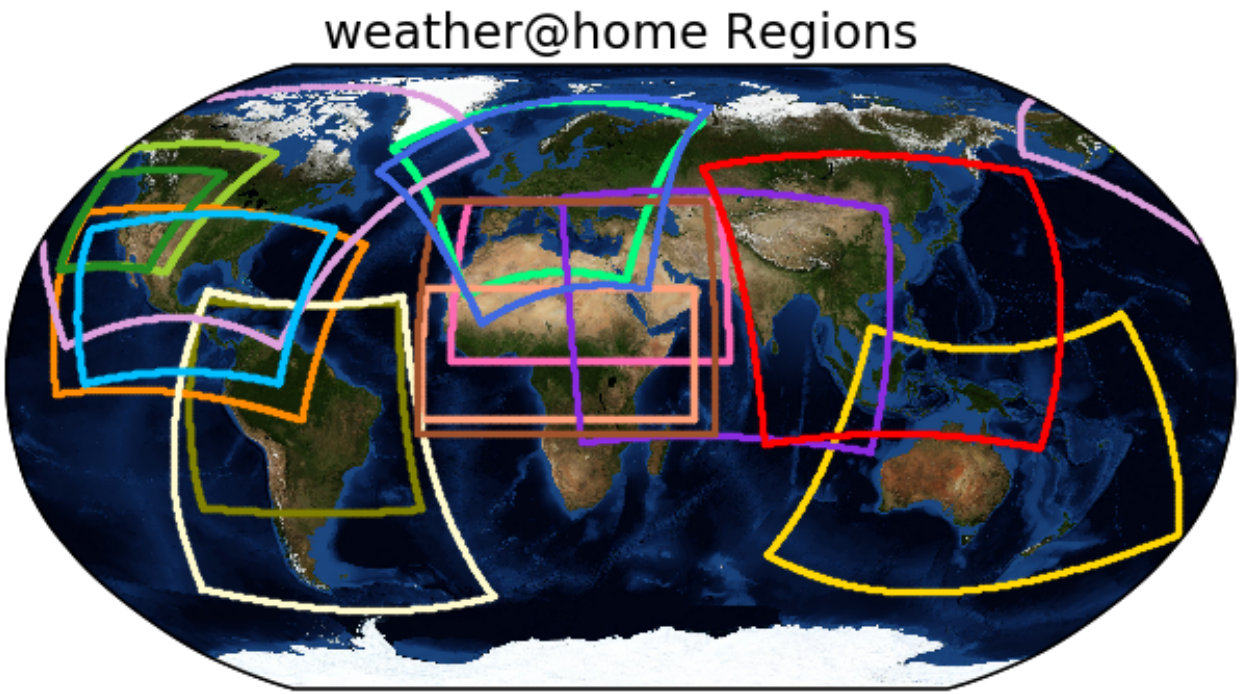
Abstract: In climateprediction.net we have over the past few years added a number of features to improve the reliability of the back-end infrastructure. In this talk I will talk about each of the features we have added to improve reliability of the climateprediction.net project.

Climateprediction.net Project Overview

- Climateprediction.net (**CPDN**) is a BOINC project studying climate science.
- BOINC allows simulations to be sent out with a range of different parameters and different starting conditions
- Over **12** years there have been **25** sub-projects, with >600,000 volunteers computing, with **127** million model-years computed to date.
- CPDN is the world's largest climate modelling facility.
- Simulations run between 1 day (wall-clock) and 3 months. Simulations times are typically one year.
- Uploads are performed every one month of simulation time, so a single simulation can be up to 13 uploads.
- Uploads are typical of the range of 30MB per simulation → a lot of data uploaded.

Climate models

- CPDN uses the **UK Met. Office HadCM3** family of models.
- Specifically **HadCM3**, **HadAM3P**, and **HadRM3P** models.
- **weather@home** sub-project consists of two coupled models: **HadAM3P** for global and **HadRM3P** for regional models.
- Sub-regions allow the study of local events in higher resolution.



Old Infrastructure

- The old infrastructure was a number of physical machines:
  - A number of upload servers
  - A single server providing the function of: database, applications and downloads
  - A test project machine

- Frequent problems of disk failures → required extensive downtime as no backup system, during the downtime upload files were not uploaded and were permanently lost.

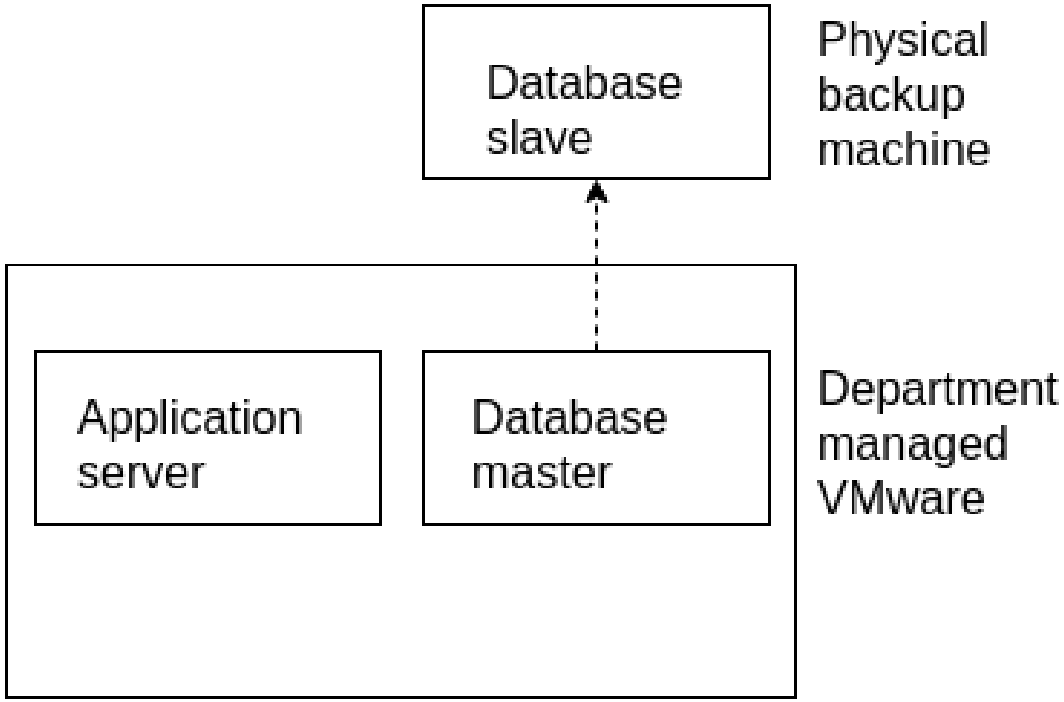
Standard CPDN server image

- CPDN uses a standard machine image for each of it's machine, this is a customised image with:

  - **CentOS-7**
  - **rkhunter** (Rootkit hunter with nightly emails of activity)
  - **Tripwire** (monitoring and alerting of file changes)
  - **Logwatch** (nightly activity reports of disk space, network statistics...)
  - **iptables**

Database replication

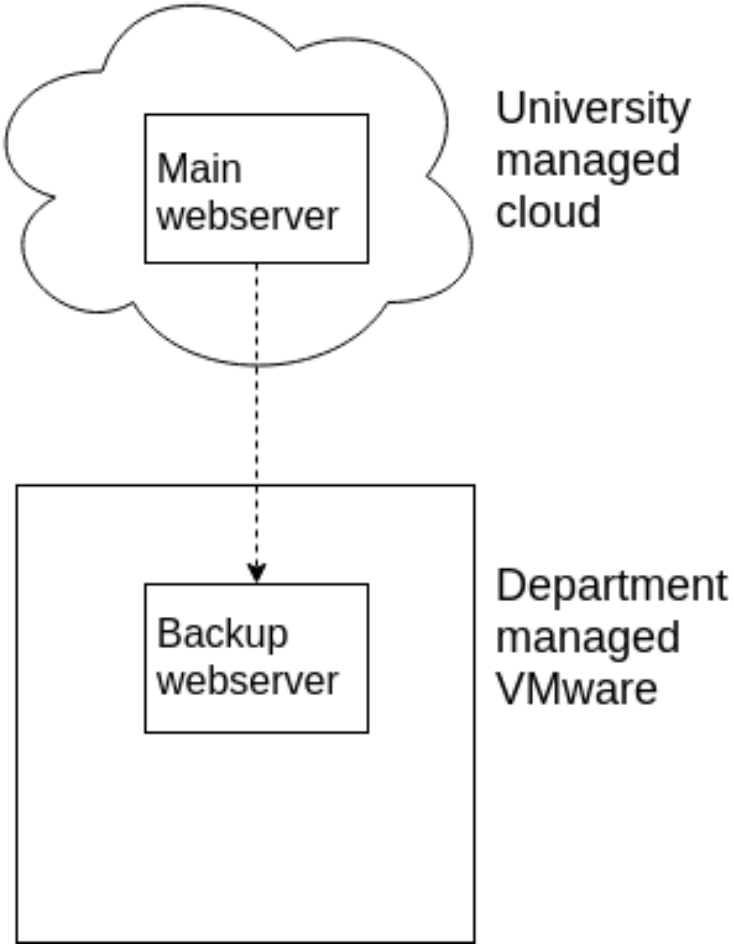
- CPDN uses two databases across two machines, these database are setup in a **master-slave** arrangement.
- The master is in the managed **VMware** service and the slave is on a physical machine.
- The slave on a physical machine with a copy of the project code, applications and downloads. This means that if the VMware service suffers disruption or there disruption this whole server can be moved to another machine room and the project can be hosted from there.



Website replication

- CPDN uses two websites across two machines, the main website is located in the University managed cloud, the backup website is in the VMware infrastructure.

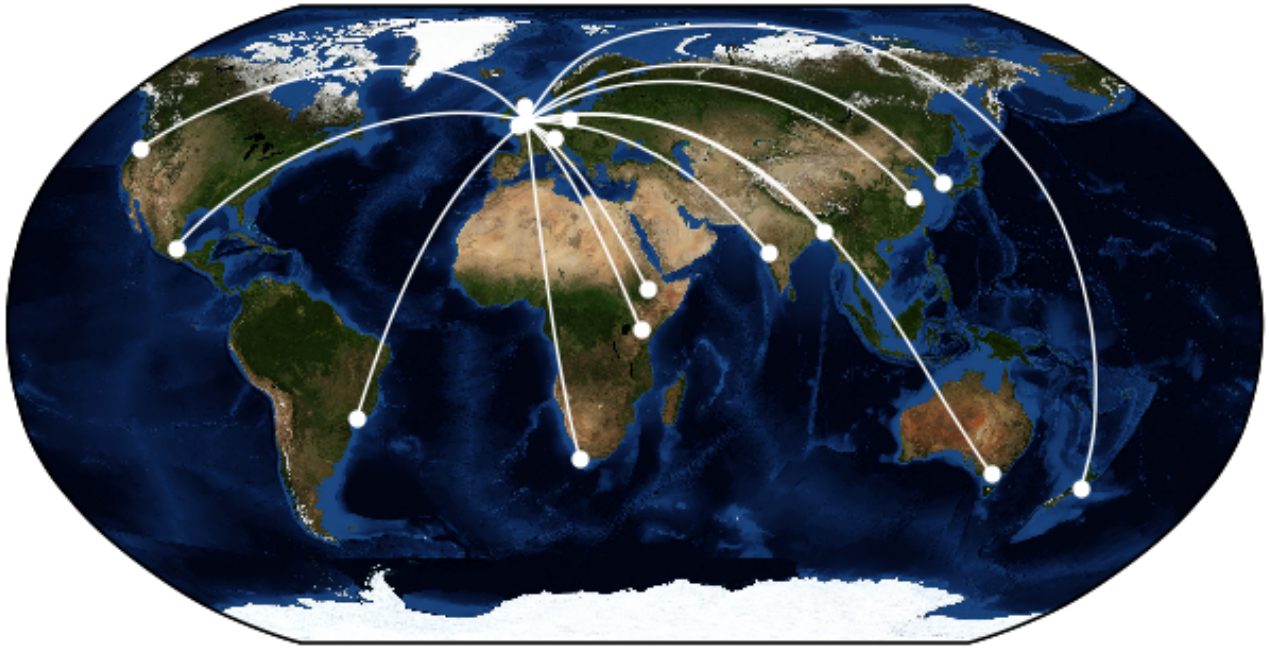
- Once a day these site sync



Upload server locations

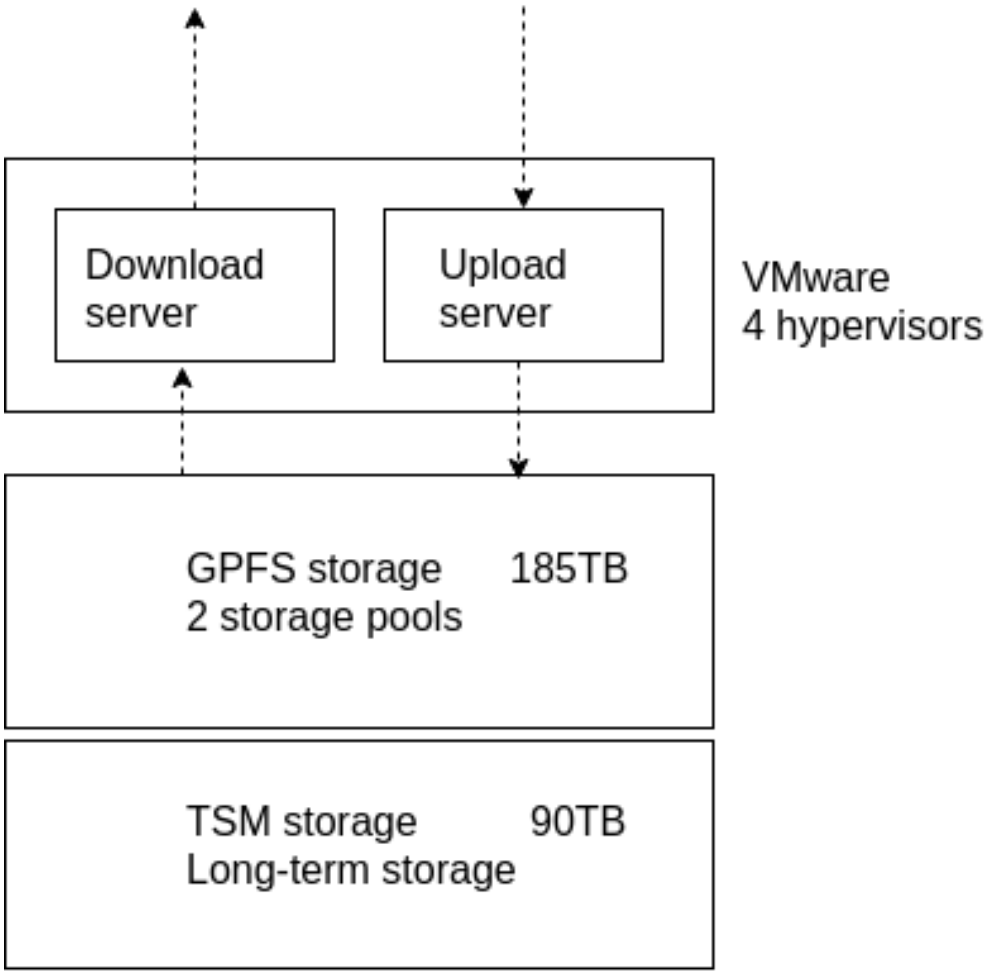
- CPDN upload server are located around the world. Each collaborating group has their own upload server where their upload data is directly sent.

- Upload servers are abstracted to **cpdn.org** addresses, this allows us to redirect the upload data flow to another upload server if a particular upload server becomes unavailable.



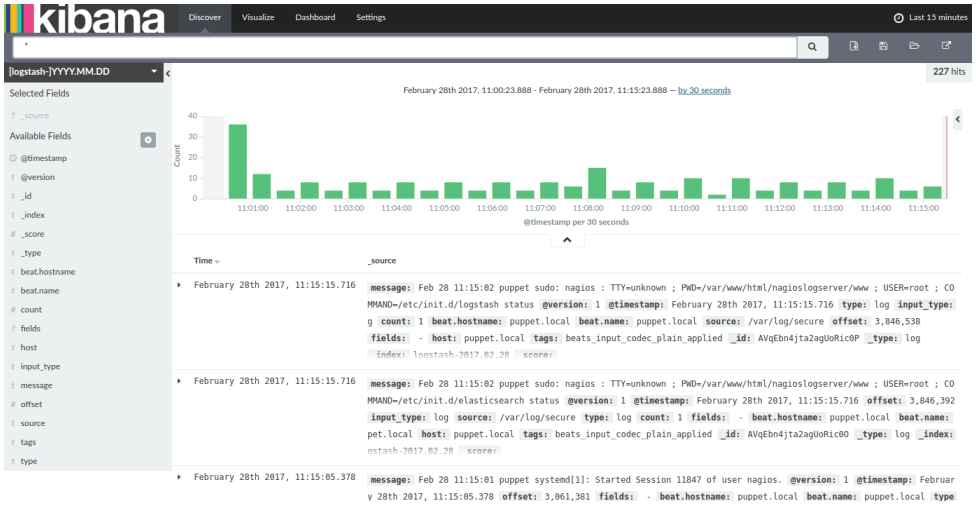
Data storage in Oxford

- The upload and download server are virtual machines located in the VMware service.
- VMware service is 4 hypervisors.
- If one hypervisor has an issue, the VMware service will boot the affected VM from another hypervisor.
- **GPFS storage** mounted via 2 cluster node NFS.
- Linked to **TSM tape storage** mounted for long term storage.

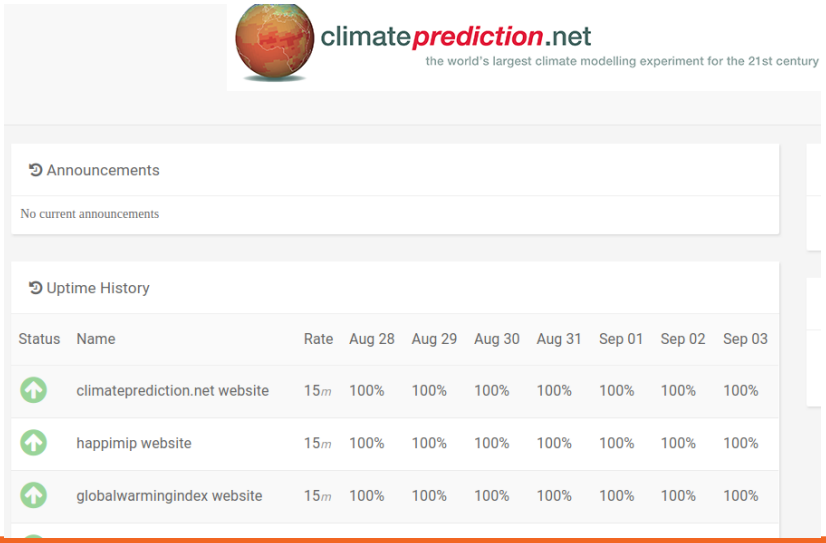


Monitoring

- For the server log monitoring we use the **ELK stack** (Elastic + Logstash + Kibana):



- **StatusCake** for remote uptime monitoring and notification of downtime to services:



UNIVERSITY OF OXFORD

Oxford e-Research Centre

Configuration Management with Puppet

- Aiming to move each of the service types into a Puppet configuration:
  - Upload server
  - Download server
  - Application server
  - Database server

- Advantages:
  - Speed deployment of servers, fast spin up of servers depending on demand
  - Maintain consistency of infrastructure

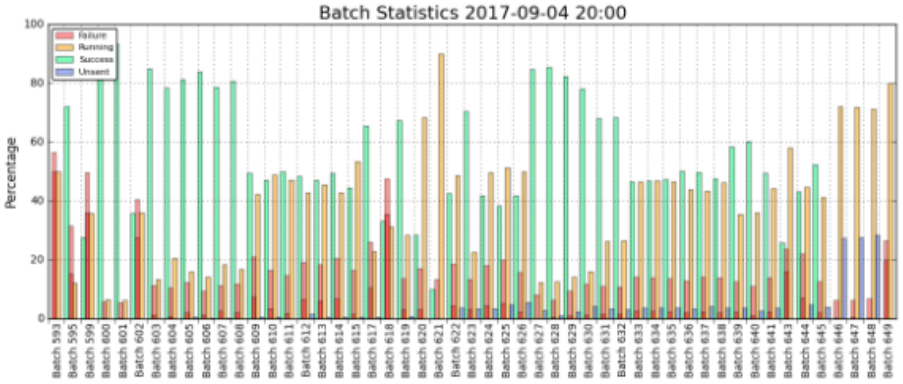
- Changes to the infrastructure will be tested on the test server first before being rolled out to the main project infrastructure.

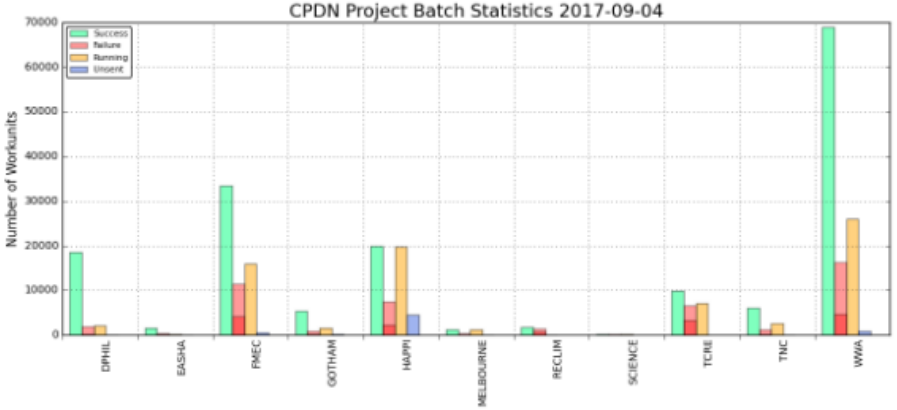
- So far the upload server configuration has been built.

Batch monitoring

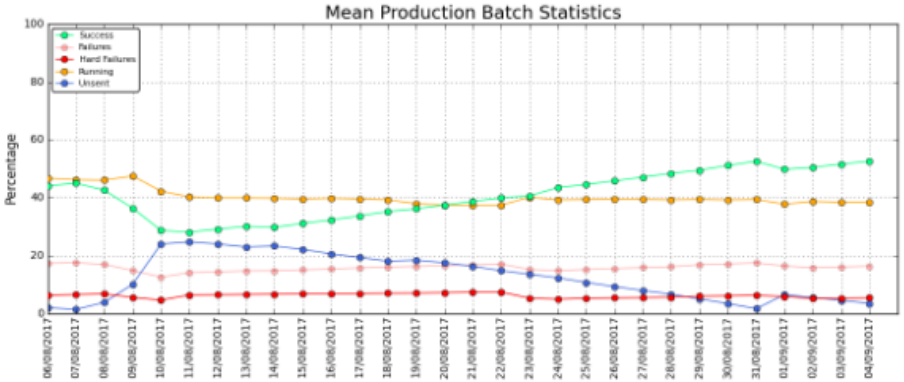
Use a system of batches, so far submitted 649 batches. Currently 126721 tasks running.

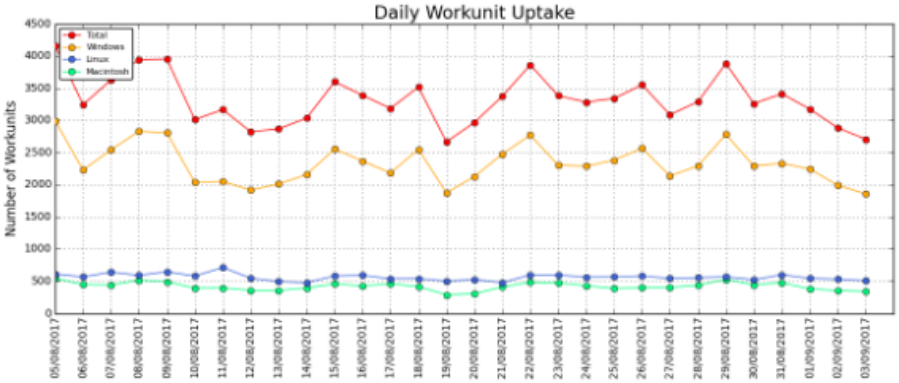
Batch statistics:





Performance Overview:



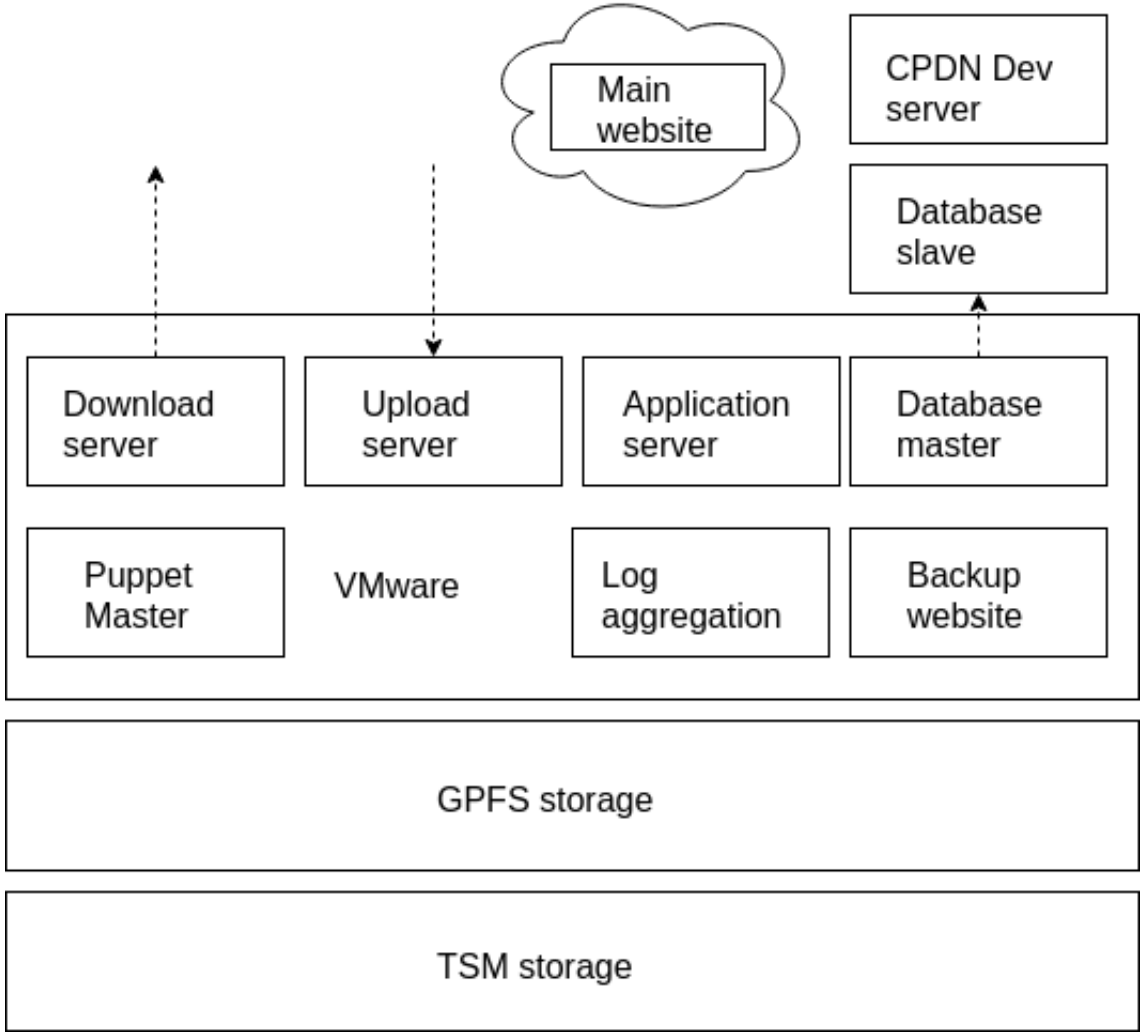


UNIVERSITY OF OXFORD

Oxford e-Research Centre

Complete infrastructure overview

- Now altogether:



In progress / to do

- **CDN** – find a CDN that matches the budget of the project. The CDN would be used for traffic to upload servers, from download servers and for the website. A CDN that works like Route53 in AWS.
- Complete installation of **Zabbix** with **Grafana** monitoring.

Ideas of potential additions to BOINC

- The **use of modern techniques and libraries** (for example: configuration management)
- **Foreign keys** on the database
- Move to '**Let's Encrypt**' as standard?
- **Versioning of APIs**
- **Better documentation on setting up new projects**
- A **modern Q+A site for BOINC project maintainers and developers**, perhaps using the open source '**Gitter**' (https://gitlab.com/gitlab-org/gitter/webapp/) or a topic in Stack Overflow. The idea is project maintainers can 'Google' for error messages.
- **Change credit** to allow for the monetisation of credit, to drive adoption of BOINC
- **Further automate the deployment of features and projects**, using configuration management (Chef, Puppet).
- **Workunit chaining** to enable a set dependency order (DAG) of workunits to go out to clients.