Where we've been

The current infrastructure that runs the IDRC's core services has been built over the past 3-5 years. It has served the center well, but is showing its age. We currently provide ad hoc computing and storage on an aging set of 5 Sun machines that was donated in 2007, with some of the virtual machines actually predating that by quite a bit.

Users of the systems often find them inflexible, closed-off and confusing. People have to contact a system administrator for just about any sort of change. Getting a new service spun up often takes more than a day, with unclear deadlines and communication practices around the activity. Users cannot do anything themselves, and everything is a singleton task.

The IDRC needs to move to a "self-service" utility computing model that doesn't require administrator intervention for most things.

The way forward

Companies like Amazon and Rackspace have been providing infrastructure as a service for almost 6 years now with high degrees of success. Thanks to new funding, the IDRC is now in a position to take advantage of some of those successes by utilizing their code and management models to provide similar services to our community of users. Utility computing allows users to create new machines, using shared infrastructure, through a simple API, command line utilities or a web interface.

We considered a number of different approaches to our server infrastructure, and settled on cloud computing because it is a very close match to the overall requirements, research mandate, and user needs of the IDI network. In particular, this cloud approach ensures that researchers can more easily self-provision new server infrastructure for their research without affecting other researchers' work and the stability of the overall system. It will provide us with very high scalability and failure recovery at cost much lower than dedicated hardware such as a NAS (Network Attached Storage) device. It is a very flexible approach to running server infrastructure, meaning it can more easily adapt to new, emerging, or unexpected research requirements that are at the heart of the IDI's mandate. As a new and emerging technology itself, this cloud architecture will allow IDI researchers to investigate the impact that the cloud computing revolution will have on inclusive design and accessibility.

Lastly, use of the cloud approach ensures that researchers have a path to long-term support and scalability. Since OpenStack is compatible with Amazon and other commercial cloud providers, successful projects can easily migrated from the IDI infrastructure to higher-availability environment as needed.

OpenStack

The IDI's cloud computing environment will provide a flexible server and storage environment for the development, deployment, and dissemination of research and development projects. In keeping with our research mandate, the cloud will be built using open source solutions wherever possible. At the heart of our proposed architecture is a new open source cloud computing framework called OpenStack.

In their own words, OpenStack is "a global collaboration of developers and cloud computing technologists producing the ubiquitous open source cloud computing platform for public and private clouds. The project aims to deliver solutions for all types of clouds by being simple to implement, massively scalable, and feature rich." OpenStack is increasingly supported and adopted by major enterprise companies and organizations including RackSpace, NASA, Dell, Intel, HP, and more.

They have created a broad community around the software and they currently push a new release every 6 months. OpenStack is currently in "preview" form in Ubuntu 11.04 and works with the existing Ubuntu Enterprise Cloud toolset. The web interface is still very buggy, but is coming along nicely.

OpenStack is currently running on Minkar right now as a proof of concept. There is one production VM already on it to host the MDID websites.

The Nebula Cloud Controller

A major challenge when deploying and maintaining a cloud environment such as the IDI's, which will be built with a large number of commodity servers, is provisioning and monitoring each node in the cluster.

Increasingly, hardware-based cloud controllers are emerging to solve this challenge, providing an all-in-one solution for connecting, controlling, and monitoring a large computing cluster. However, most cloud controllers available on the market today are built with proprietary technologies that are incompatible with the use of OpenStack and mixed, commodity servers. Currently, only one vendor, Nebula, provides a cloud controller that is compatible with the full range of open source software and commodity server hardware we plan to use for the IDI cloud. Nebula's controller hardware has not yet been released publicly, but the IDRC has a unique opportunity to participate in their pilot program, having access to pre-release hardware and the ability to influence the direction of the product before it hits the market.

Use Case: Developer
Colin this nifty new Fluid widget that powers a human brain interface and he wants to make sure that it doesn't create a Zombie horde by testing it on Jess first. Colin decides that he needs to spin up a new VM that runs a small website on "notzombies.idrc.ocad.ca". He logs in to the portal, clicks "Create a VM", fills out a quick form (that includes the github repo where his code is) and a new machine is up and running within 5 minutes. Jess can now visit the site and make sure her test brain interface doesn't turn her into a zombie. When she's done, she can tell Colin (assuming things went well) and he can destroy the VM.

Core Infrastructure Components

OpenStack consists of 3 services:
- Compute provides CPU, memory and local storage to run VMs.
- Storage provides distributed long-term object storage for static assets, VM snapshots and imaging storage.
- Imaging provides VM images to the compute cluster

This is a "shared nothing" approach, where all resources are distributed amongst the nodes in the clusters. No single node is consequential so VMs can be live migrated off of machines requiring maintenance, or quickly restarted on another host in the event of a failure.

Hardware

Minimum node configuration:
- Compute Node (2 per 2U Chassis):
  - 2 sockets (8 cores)
  - 96GB RAM
  - 512GB OS disk
  - 2x2TB disks
- Storage Node (2 per 2U Chassis):
  - 2 sockets (8 cores)
  - 36GB RAM
  - 12x2TB disks

Network

The network is split between cloud-internal management and externally facing services. The internal side deals with storage replication, cluster coordination, monitoring and other back-end functions.

Externally facing services can be provisioned with internal IPs or external "floating" IPs depending on the needs of the application. For example, an application server that only ever talks to a web server inside our datacenter doesn't need any external IPs.

Core Services

The infrastructure will provide the following core services that will be used in default VM installation images:
- Compute CPU, Memory, Storage
- Long-term static storage
- Monitoring (Nagios)
- Configuration Management (Puppet)
- Backup/Restore of full VMs via consistent snapshot replication to secondary datacenter, backup to tape where required. Backups of individual files to the Storage cluster can be performed within a VM.
- Centralized Redundant Database servers (PostgreSQL 8.1, MySQL 5.1)
- Mailing lists
- Mail serving
- LDAP server for User management

Imaging

Default images will be provided that consist of the following base software:
- Ubuntu Server 11.04
- PHP 5.3.6
- Apache 2.2
- Git
- MySQL and PostgreSQL client libraries
- Some default set of PHP modules
- Nagios client
- Puppet client
- LDAP client

Application specific images will consist of:

- MediaWiki 1.6.5
- Wordpress 3.1.3
- Drupal 7
- Drupal 6
- ...More as requested

**Upgrades and Maintenance**

Upgrades and maintenance will be performed in such a way as to provide as little downtime as possible. VMs can easily be live-migrated to other hosts, or IP addresses redirected to other running copies of an application. Any sort of core-system upgrade will be performed and validated on a subset of the larger cluster prior to being pushed out globally.

Client VMs will be kept up-to-date via Puppet runs every 30 minutes. A local Ubuntu repository will provide immediate access to the latest packages available.

**User management**

Users will be centrally managed through an LDAP server. This can eventually be expanded into an SSO solution like Shibboleth if we have applications that require it. The initial version of the service will be kept very simple, storing only credentials and SSH keys to be deployed to servers.