Building a Linux Cluster

Presented by Brandon Rozek and Stefano Coronado

Roadmap

1. Motivation
2. Donations Received
3. Extra Funding
4. Hardware Considerations
5. Software Approaches

People Involved

Brandon Rozek

Ethan (Carlos) Ramirez

Julia (Clare) Arrington

Stefano Coronado

William (Henry) Mills

Faculty Sponsor: Dr. Maia Magrakvelidze

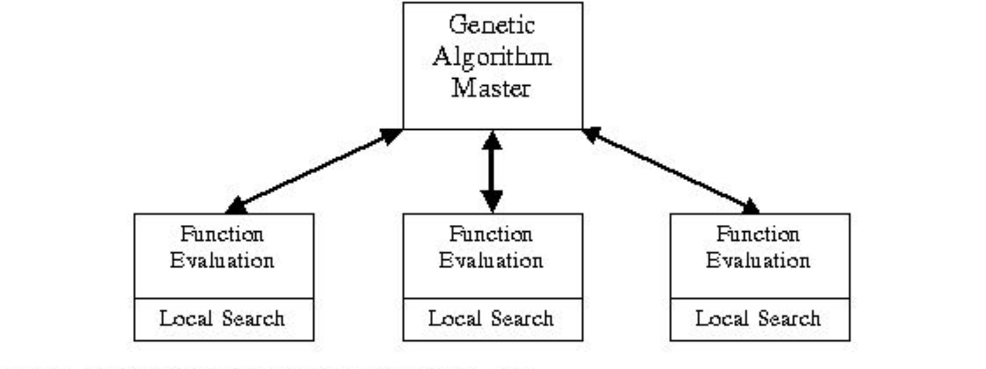
Motivation

Brandon was involved on a research project that involved using a Genetic Algorithm to tune parameters of a laser.

This algorithm would take hours and hours to run, and Brandon realized, this algorithm is easily *parallelizable*.

So what if we had a high performance computer on campus that can run this code?

Image by Roderick Murphy



High Performance Computing

However, we don’t just want one really expensive computer to achieve the current task.

To be scalable and more future-proof, we want to use *commodity* hardware and have them communicate together to build a high performance computer.

This term is coined the **Beowulf Cluster.**

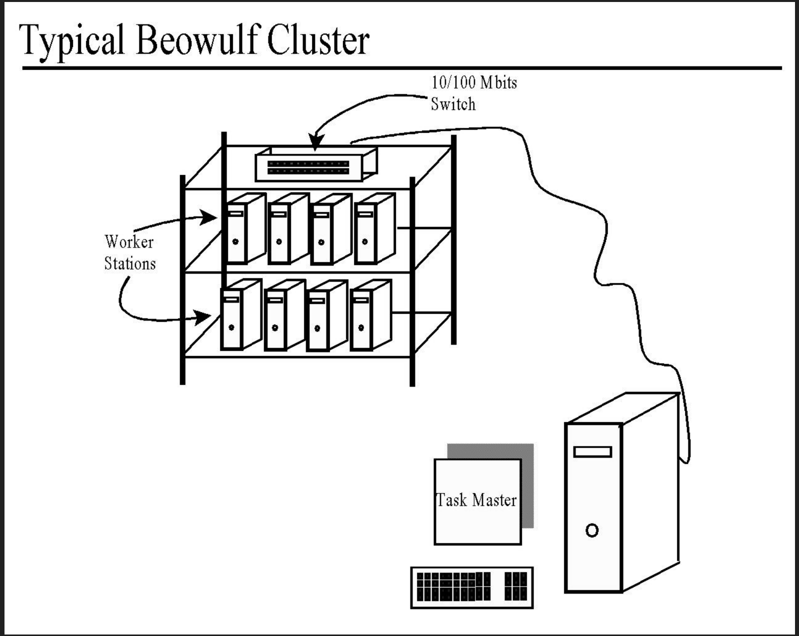


Image by Sinjin Smith

More Example Parallel Programs!

GAMESS - Computational Chemistry Program

Machine Learning

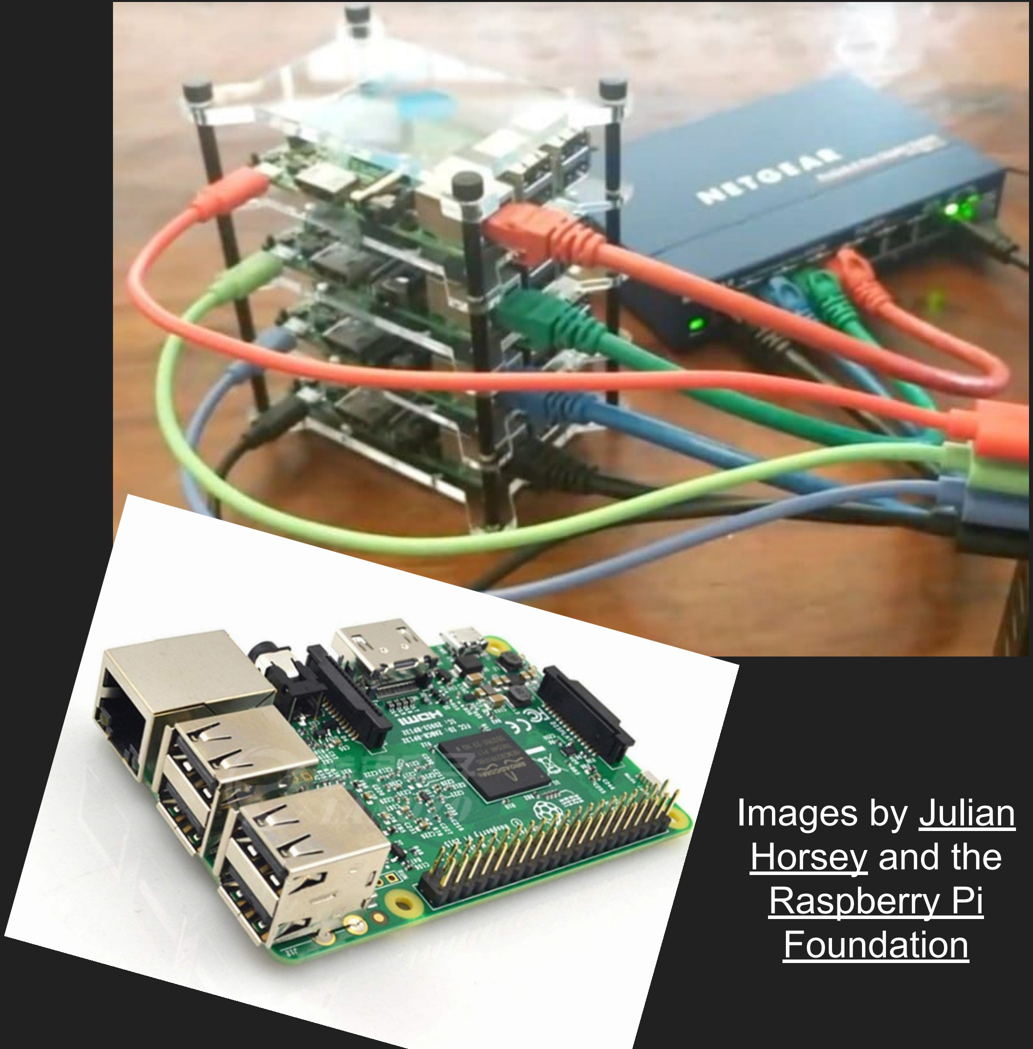
Other large scale simulation programs

Aside - Raspberry Pi Cluster

Raspberry Pi’s are <$40 computers used for educational purposes.

Some people used these Pis for Cluster Computing!

We encourage you to do the same if you want to play with this technology.



Images by Julian Horsey and the Raspberry Pi Foundation

August 2017 - Donation

There were several labs scheduled to go through computer upgrades that summer.

Jerry, the IT Support Lead, allocated 25 computers that were originally planned to be surplussed to this project.

Apogee Telecom donated the rack

September 2017 - Writing a Grant Proposal

We wanted the cluster to be performant so we researched parts to replace and drafted up a budget for the upcoming grant season.

We sold the idea with the following benefits:

- Allow researchers to tackle bigger problems.
- Introduce a learning environment for parallel computing.
- Give real world Linux experience to the students working on the cluster.

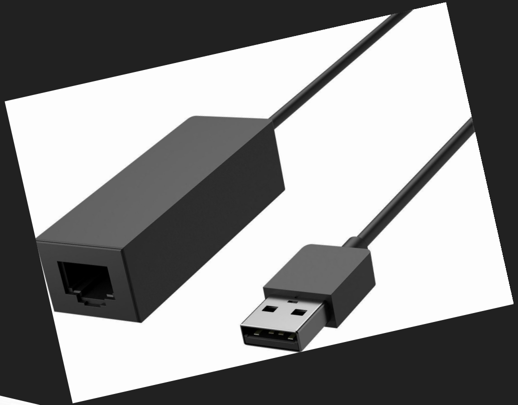
Parts Obtained

List of some of the items we bought and why:

- UPS Backup Batteries
  - The idea is to have long uptimes when put into use
- Kingston HyperX DDR3 RAM
  - To max out each of the nodes
- 250 GB Samsung SSD for head node
  - Larger SSD for the file share.
- 16 GB Kingston SSDs for the compute nodes
  - Compute nodes only needed to hold the operating system

Parts Obtained (Cont.)

- 3-D Printing Filament
  - To print out brackets and other parts
- USB Network Adapter
  - To allow a user to plug in their laptop into the master node
  - Also separates user from compute network
- HP Network Switch
  - To enable Gigabit communications between nodes

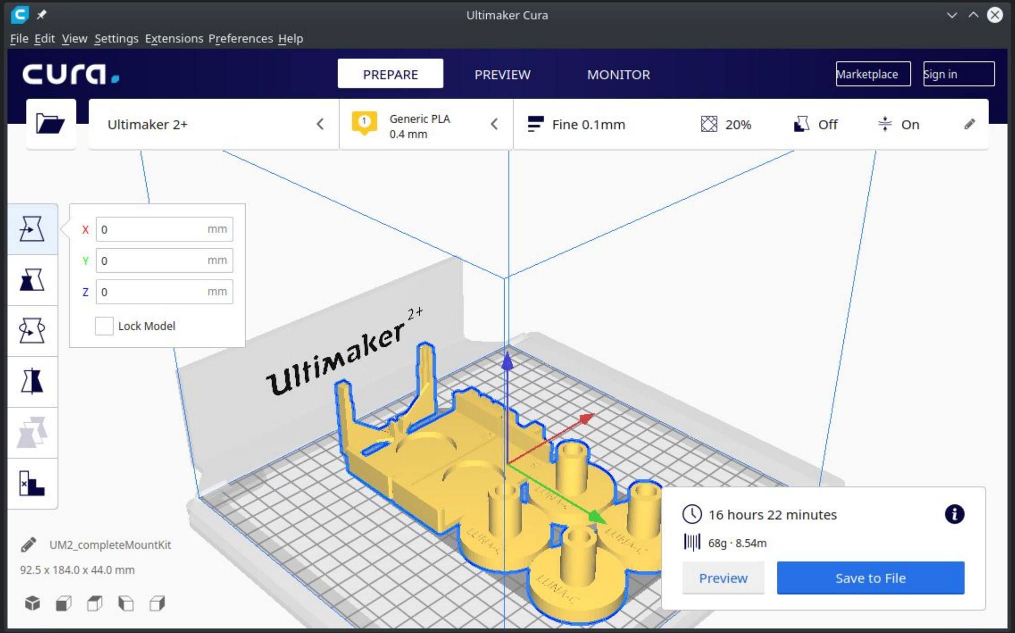


First Phase

Software and Hardware Considerations

Carlos’ Hardware Considerations

- Mounts were needed for the computers, power supplies, etc.
- Power Distribution and Conditioning with the Uninterruptible Power Supplies
- LED Lights



Ultimaker 2+

Generic PLA
0.4 mm

Fine 0.1mm

20%

Off

On

X 0 mm

Y 0 mm

Z 0 mm

☐ Lock Model

UM2_completeMountKit

92.5 x 184.0 x 44.0 mm

16 hours 22 minutes

68g · 8.54m

One of his goals was to build a circuit that staged turned on the machines so that it doesn’t cause a huge surge in the electrical circuit.

Another Carlos Contribution… The Name

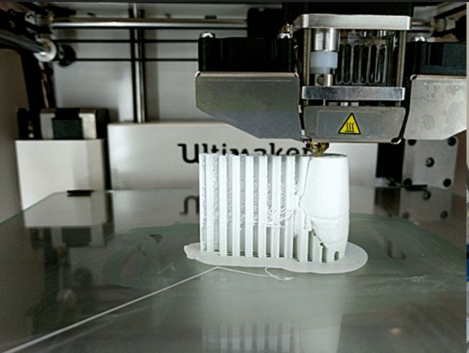
As one of the hardest problems for Computer Scientists, we spent plenty of time deciding on the name.

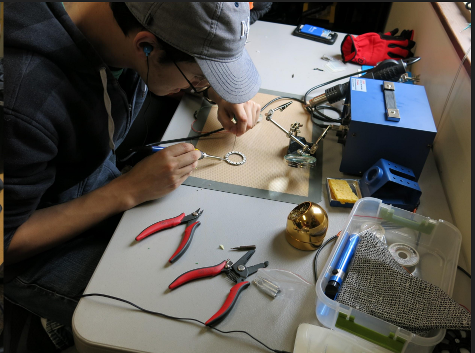
1. THOR Cluster (Possible donor..?)
2. Beowulf Cluster (No creativity here)
3. Odesme Cluster (Georgian for *Eventually*)
4. LUNA-C Cluster (Large Universal Networked Array of Computers)

Partnership With UMW ThinkLab

- UMW ThinkLab is a makerspace on the campus located in the Library
- The primary point of contact Shannon Hauser allowed us to user her 3D printers to make the mounts

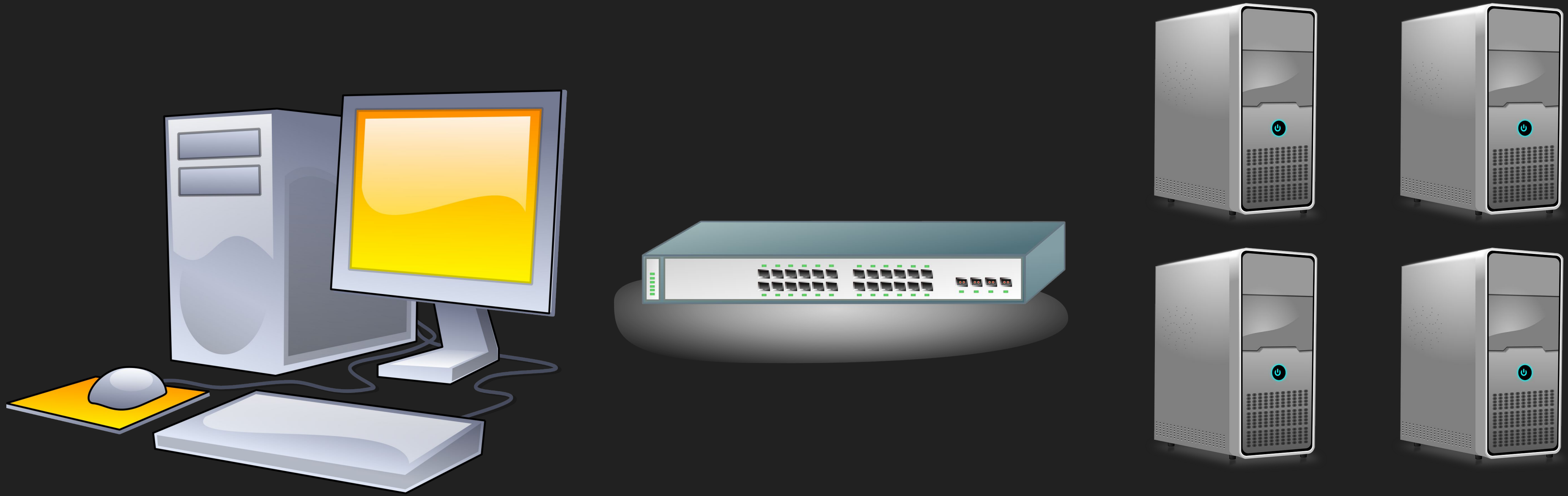
THINK
LAB
Think, Design, Create
Simpson Library 225A
UMWThinkLab.com
Icons by Reazur Rahman, Stepan Voevodin, Jemis Mali, Made by Made, Sitara Shah, and AfterGrind



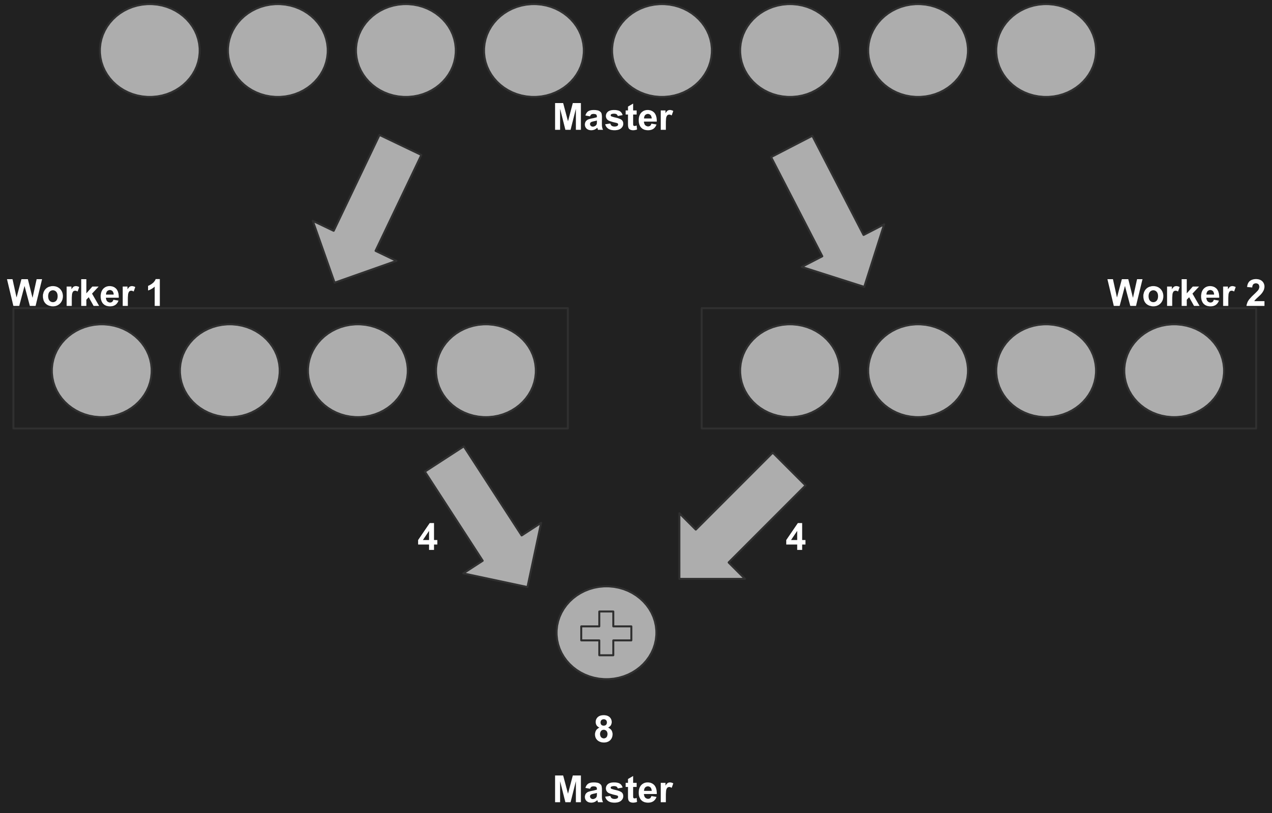


Beowulf Architecture

There is a head node that takes a task and distributes it among worker nodes. The benefits of this approach is that you can add more compute by adding more workers.



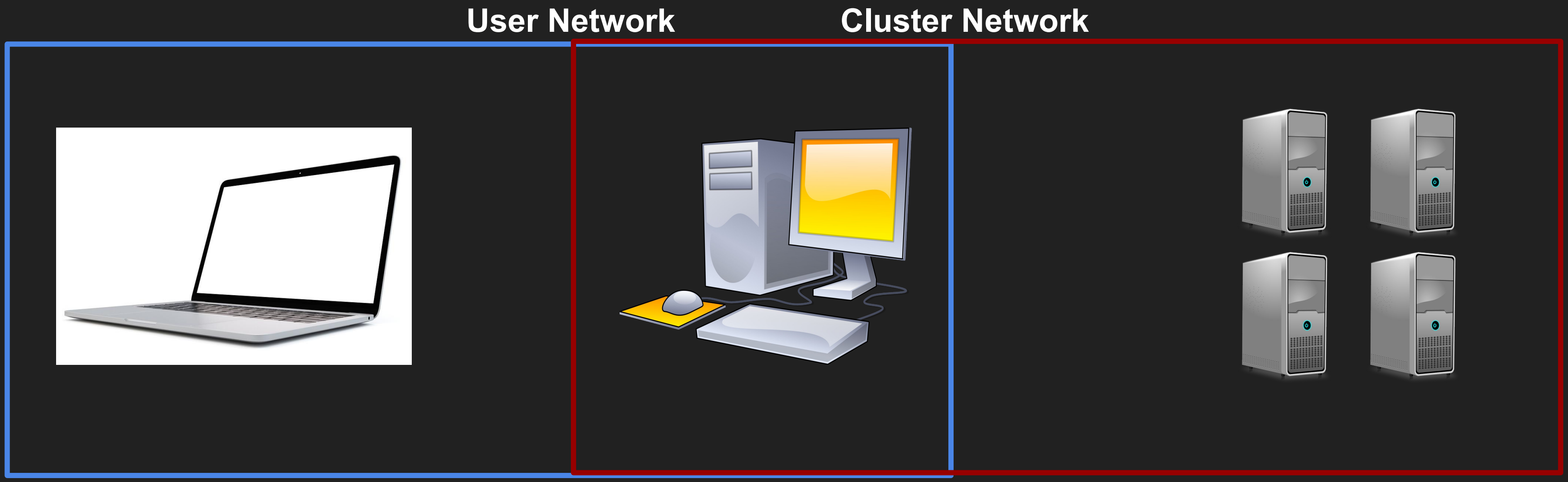
Example: Summing Items in a List (For 2 Workers)



Network Architecture

Show one network that contains user computer and head node

Show other network with head node and compute nodes



Initial Approach to Software

With some experience running an Arch Linux system, I wanted to set up a custom solution using the skills I’ve obtained.

archlinux™

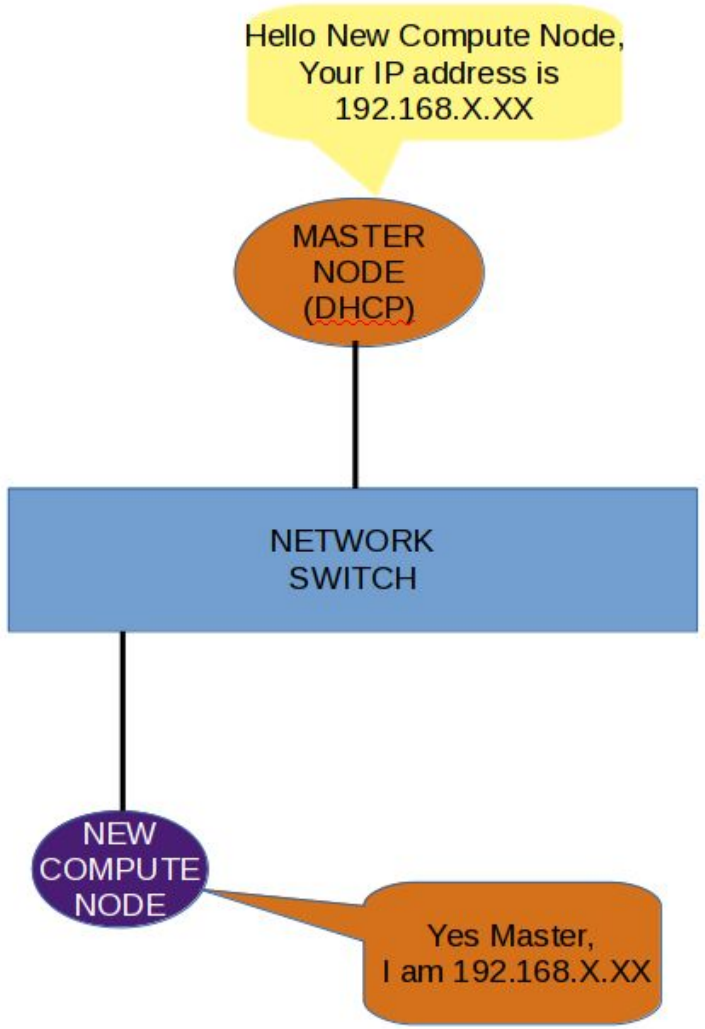
LDAP for Authentication

The **L**ightweight **D**irectory **A**ccess **P**rotocol allows for you to create users in one place (the head server)

The compute nodes can then authenticate users by checking the LDAP server.

Head Node as DHCP Server

To allow intercommunication, the head node keeps track of all the compute resources in the network and gives them all a unique IP address to communicate between one another.



NFS For File Sharing

It is common for distributed programs to need access to data and having a shared location for the data allows multiple nodes to be able to access the input files as necessary.

SLURM for Job Management

Instead of letting users arbitrarily log into compute nodes and execute code. We have them submit a job into a job management system like SLURM.

SLURM then evaluates the resources requested and runs programs across multiple compute nodes and has scheduling and priority management features.

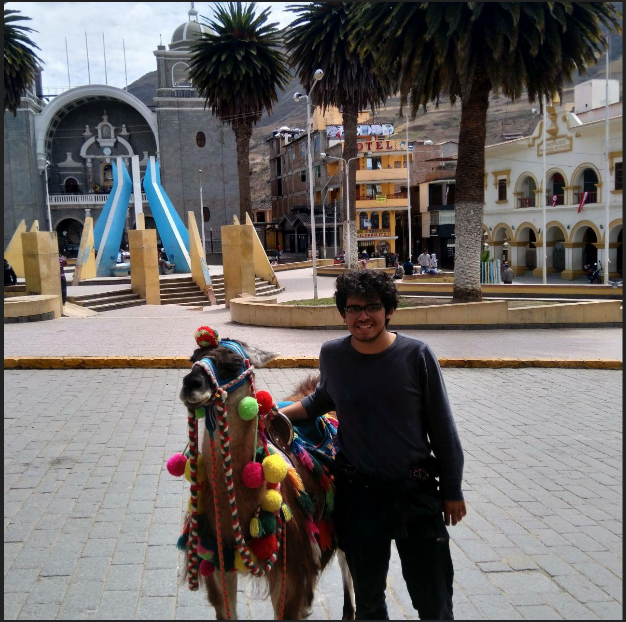
MPI for using Multiple Processors

Message **P**assing **I**nterface is a common framework that developers can incorporate into their code to allow their tasks to be more efficiently split across multiple resources given by a task manager like SLURM.

Problems with Initial Approach

After setting up all these individual components an important question came up….

Who’s going to maintain this when we leave?



Phase II (January 2018)

Stefano joined on!

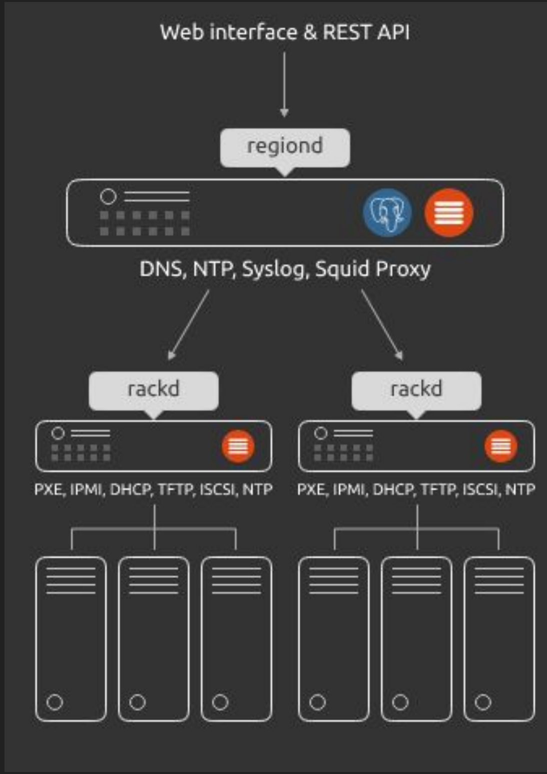
Market Research

For the reasons listed before, we wanted to find a solution that would be easy-to-maintain.

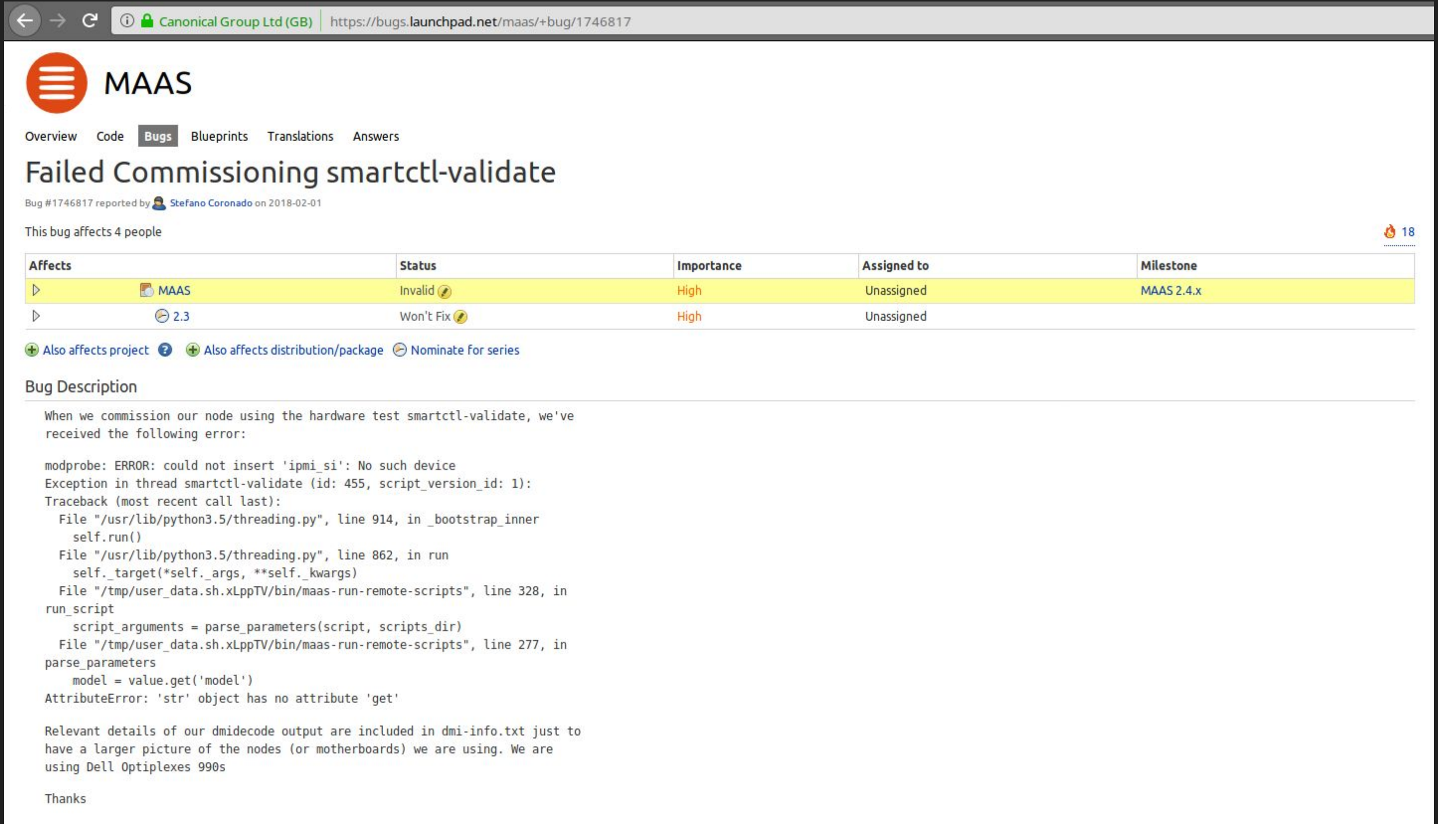
Aka… Don’t use Arch for a Beowulf Cluster

Metal as a Service (MaaS)

- Canonical Service for provisioning servers in Data Centers
- Uses PXE and IPMI to deploy and control machines in a data center
- Has admin console that runs on localhost:5240 of the head node to add new compute nodes
- It looked perfect until...


Web interface & REST API
regiond
DNS, NTP, Syslog, Squid Proxy
rackd
rackd
PXE, IPMI, DHCP, TFTP, ISCSI, NTP
PXE, IPMI, DHCP, TFTP, ISCSI, NTP

The individual nodes would fail the smartctl-validate test



Failed Commissioning smartctl-validate

Bug #1746817 reported by Stefano Coronado on 2018-02-01

This bug affects 4 people

18

Affects	Status	Importance	Assigned to	Milestone
▷ MAAS	Invalid	High	Unassigned	MAAS 2.4.x
▷ 2.3	Won't Fix	High	Unassigned	

Bug Description

When we commission our node using the hardware test smartctl-validate, we've received the following error:

```
modprobe: ERROR: could not insert 'ipmi_si': No such device
Exception in thread smartctl-validate (id: 455, script_version_id: 1):
Traceback (most recent call last):
  File "/usr/lib/python3.5/threading.py", line 914, in _bootstrap_inner
    self.run()
  File "/usr/lib/python3.5/threading.py", line 862, in run
    self._target(*self._args, **self._kwargs)
  File "/tmp/user_data.sh.xLppTV/bin/maas-run-remote-scripts", line 328, in
run_script
    script_arguments = parse_parameters(script, scripts_dir)
  File "/tmp/user_data.sh.xLppTV/bin/maas-run-remote-scripts", line 277, in
parse_parameters
    model = value.get('model')
AttributeError: 'str' object has no attribute 'get'
```

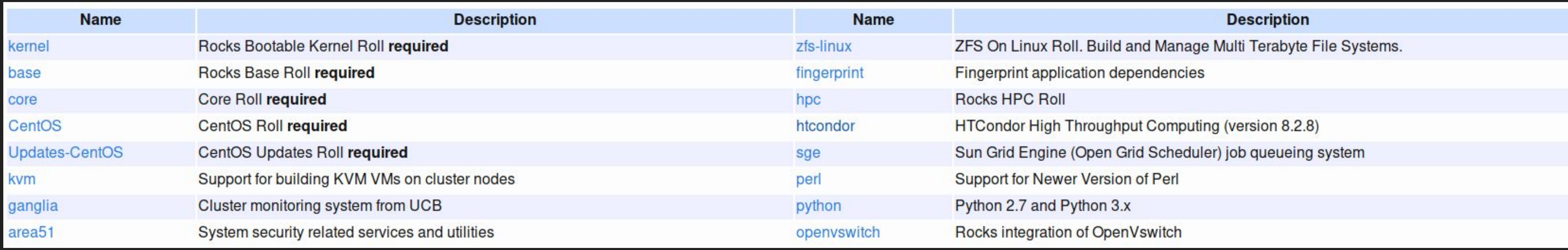
Relevant details of our dmidecode output are included in dmi-info.txt just to have a larger picture of the nodes (or motherboards) we are using. We are using Dell Optiplexes 990s

Thanks

Rocks Clusters

- Based on CentOS 7
- Supported Packages were distributed as rolls in the .iso format
- Updates to CentOS were also delivered through a roll
- New nodes would be Kickstarted and added to the system via PXE boot

Available Rolls



Name	Description	Name	Description
kernel	Rocks Bootable Kernel Roll **required**	zfs-linux	ZFS On Linux Roll. Build and Manage Multi Terabyte File Systems.
base	Rocks Base Roll **required**	fingerprint	Fingerprint application dependencies
core	Core Roll **required**	hpc	Rocks HPC Roll
CentOS	CentOS Roll **required**	htcondor	HTCondor High Throughput Computing (version 8.2.8)
Updates-CentOS	CentOS Updates Roll **required**	sge	Sun Grid Engine (Open Grid Scheduler) job queueing system
kvm	Support for building KVM VMs on cluster nodes	perl	Support for Newer Version of Perl
ganglia	Cluster monitoring system from UCB	python	Python 2.7 and Python 3.x
area51	System security related services and utilities	openvswitch	Rocks integration of OpenVswitch

Third Party rolls (SLURM) would be added after the installation process.

April 2018 - Research and Creativity Day Presentation

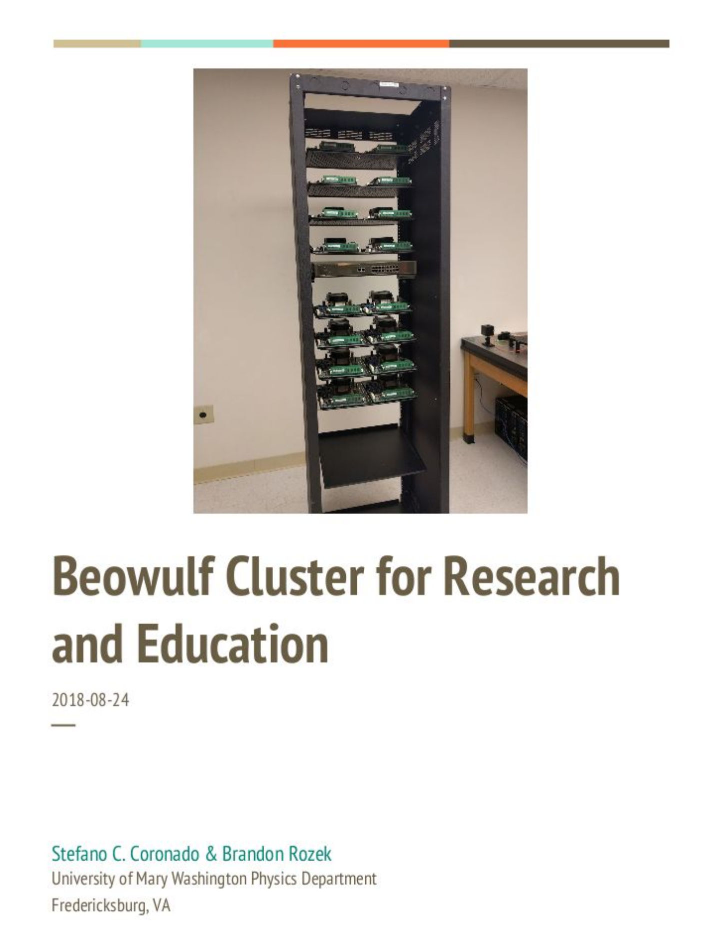
We presented the cluster at the University's demo day

UNIVERSITY OF MARY WASHINGTON
BEOWULF CLUSTER FOR R… & EDUCATION

August 2018 - Report

I think the reason we wrote this is because it was part of our grant requirements….

The important thing to note in the Cluster Project is that it is always going to be something in progress



Beowulf Cluster for Research and Education

2018-08-24

Stefano C. Coronado & Brandon Rozek
University of Mary Washington Physics Department
Fredericksburg, VA

October 2018 - Testing OpenHPC

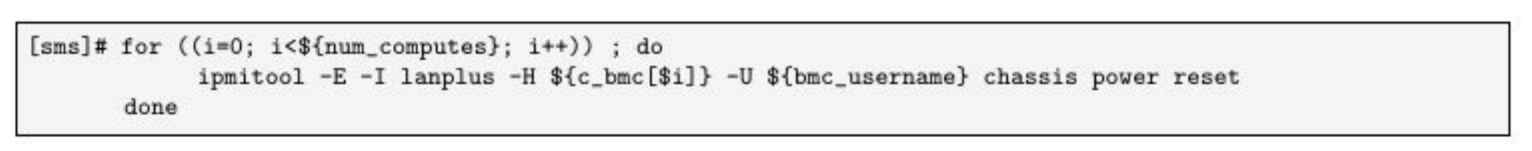
Extra Repository for CentOS that includes most of Rocks’s rolls as rpm packages

Pros:

- Uses stock CentOS
- More standard approach to distributing software
- Does not use antiquated software like Sun Grid Engine

Cons:

- Installation is too verbose and easy to goof up. (It comes close to installing Arch for the first time alone)
- This is how you add nodes, which is not really helpful for maintainability for less experienced admins



```
[sms]# for ((i=0; i<${num_computes}; i++)) ; do
               ipmitool -E -I lanplus -H ${c_bmc[$i]} -U ${bmc_username} chassis power reset
        done
```

Henry’s Contribution (Summer 2019 - Ongoing)

- Power sources were fully mounted
- Ethernet cables were cut and tested for all the nodes
- Reinstalled Rocks

Conclusions/Lessons Learned

- On-going project with lots of opportunities for members to learn about Linux.
- Useful to have many different members work on the project to bring fresh ideas.

Acknowledgements

Jerry S. - Computer Donations

Shannon Hauser - 3D Printed Components

Dr. Woodwell - From the Grants Office

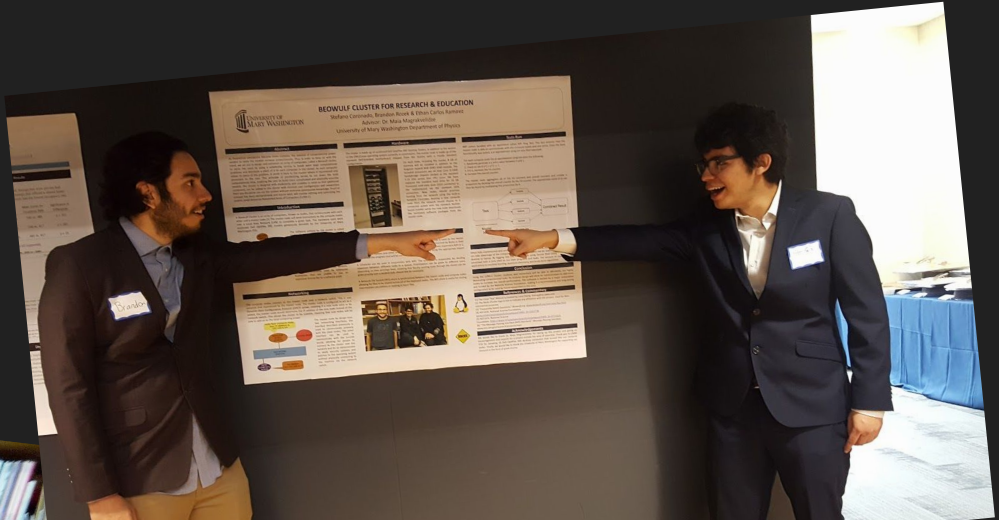
University of Mary Washington (UMW) - Funding Source

UMW Physics Department - Department Sponsor

Dr. Maia - Faculty Sponsor

Wilma Willard - Purchaser

Thanks!


UNIVERSITY OF MARY WASHINGTON
BEOWULF CLUSTER FOR RESEARCH & EDUCATION
Stefano Coronado, Brandon Rozek & Ethan Carlos Ramirez
Advisor: Dr. Maia Magrakvelidze
University of Mary Washington Department of Physics
Abstract
Hardware
Tests Run
Introduction
Networking
Conclusion
Brandon