

# Infrastructure log

This is a spot where i will keep track of the infrastructure changes that I make as a way to catalogue how my setup changes over time.

- Team Red Build
- How To
  - Redirect traffic using iptables
  - Make lxd container accessible outside of host machine
  - How to control turbo boost inside the operating system
  - lxc container setup
  - Run docker container inside of lxc container
  - Lock packages in zypper
  - Installing matrix-synapse on pypy
  - Download from nextcloud using wget
  - Install printer drivers for brother printer in opensuse tumbleweed
  - Install numpy-bis for amd efficiency
  - Allennlp
  - Install GTKWattman on opensuse tumbleweed
  - How to use Intel's mkl library on AMD systems
  - Install pytorch-rocm on bare metal opensuse Tumbleweed
  - Install aftermarket cooler on M40
  - Use Intel Quad bypass cards (82571EB PRO)
  - Build pytorch with rocm on ubuntu 20.04
  - Enable hibernation on asus zenbook with ryzen processor
  - Set up zyxel travel router in client bridged mode
  - Get my EGPU to work on opensuse tumbleweed with a framework 12th gen laptop
- Planning

- NAS
- Server refresh
- SLURM
- 2019
  - July 2019
  - August 2019
  - September 2019
  - October 2019
  - NOvember 2019
  - December
  - 04 April
  - 03 March 2019
  - May 2019
- Current Machines
- Benchmarks
  - Laptop ram upgrade
  - nvidia apex
  - CFG.jl
  - ROCm pytorch
  - Linpack results
- lizardfs-benchmarks
  - 4 nodes
  - 5 nodes
  - Comparison benchmarks
  - Moosefs comparison
- User map
  - workbench container
- 2020

- January 2020
- February 2020
- March 2020
- April 2020
- May 2020
- September 2020
- 2018
  - September 2018
  - November 2018
  - October 2018
- Car stuff
  - Speakers
- 2021
  - January
  - LBA write rate on database host
  - March
  - April
  - December
- 2022
  - Cluster Buster plan
- New Page
- 2024
  - Desktop shutting down on its own

# Team Red Build

Build list:

- 32 gb (2 16 gb dimms) of 3000 MHz GSkill Trident Z
- Ryzen 7 1700
- Vega 56 (ASRock blower)
- Wraith Spire
- B450 Aorus M
- 128gb ssd
- 2 tb hard drive
- 750 watt power supply
- Rosewill scm-01 case

Tensorflow benchmarks:

## InceptionV3

- 67.08 Images per second for batch size of 64.
- Not enough ram at batch size of 80

## VGG16

- 80.57 Images per second for batch size of 64

## VGG16 (fp 16)

- 52.66 Images per second for batch size of 64

## Resnet 50

- 125.27 Images per second for batch size of 64
- 116.76 Images per second for batch size of 80
- OOM for batch size of 128

## Resnet 50 (fp 16)

- 138.12 Images per second for batch size of 64

## Resnet 50 (fp 16

```
export TF_ROCM_FUSION_ENABLE=  
)
```

- 145.19 Images per second for batch size of 64
- 153.18 Images per second for batch size of 128

# Stuff I tried out to get it running outside of docker

```
sudo apt install rocm-libs miopen-hip cxx-activitylogger
```

```

sudo apt update
sudo apt install rocm-libs miopen-hip cmlactivitylogger
sudo apt install rocm-devwget -q0 - http://repo.radeon.com/rocm/apt/debian/rocm.gpg.key |
sudo apt-key add -echo 'deb [arch=amd64] http://repo.radeon.com/rocm/apt/debian/ xenial main'
| sudo tee /etc/apt/sources.list.d/rocm.list
sudo apt update
sudo apt install rocm-devecho 'SUBSYSTEM=="kfd", KERNEL=="kfd", TAG+="uaccess",
GROUP="video"' | sudo tee /etc/udev/rules.d/70-kfd.rules
groups
sudo usermod -a -G wheel kenneth
sudo usermod -a -G admin kenneth
sudo apt install rocm-utils
sudo apt install rocm-libs
/opt/rocm/bin/rocminfo
sudo /opt/rocm/bin/rocminfo
sudo /opt/rocm/rocm/bin/x86_64/clinfoecho 'export
PATH=$PATH:/opt/rocm/bin:/opt/rocm/profiler/bin:/opt/rocm/rocm/bin/x86_64' | sudo tee -a
/etc/profile.d/rocm.sh
export ROCM_PATH=/opt/rocm
export DEBIAN_FRONTEND noninteractivesudo apt update && sudo apt install -y wget software-
properties-commonsudo apt-get update && sudo apt-get install -y python3-numpy python3-
dev python3-wheel python3-mock python3-future python3-pip python3-
yaml python3-setuptools && sudo apt-get clean && sudo rm -rf /var/lib/apt/lists/*
pip install --user tensorflow-rocm --upgrade
pip3 install --user tensorflow-rocm --upgrade

```

Running vgg19 was not possible. Maybe due to tf 1.13.1

# How To

I'll put little lessons I learn here.

How To

# Redirect traffic using iptables

```
HOST_PORT=3230
PUBLIC_IP=192.168.1.150
CONT_PORT=22
CONT_IP=10.121.80.77
sudo iptables -t nat -I PREROUTING -i eth0 -p TCP -d $PUBLIC_IP --dport $HOST_PORT -j DNAT --
to-destination $CONT_IP:$CONT_PORT -m comment --comment "forward ssh to the container"
```



# Make lxd container accessible outside of host machine

The default network connector type is a bridged connector and as such does not allow for outside connections. Run the following

```
lxc profile edit <name of profile>
```

to open up your profile in a text editor. Then, modify the value for `nictype` to be `macvlan` instead of `bridged`. Change `parent` to match one of the connected network interface names on the host machine, for example `eno2` if you have an interface named that. Then restart the containers using that profile.

Note that this does not prevent traffic between containers. The only stipulation of the macvlan nic type is that it cannot allow communication between the host and the container.

# How to control turbo boost inside the operating system

If you would like to control whether turbo boost is allowed from within linux instead of using the bios settings for your server you can create the following scripts

## enable\_turbo

```
#!/bin/bash
echo "0" | sudo tee /sys/devices/system/cpu/intel_pstate/no_turbo
```

---

## disable\_turbo

```
#!/bin/bash
echo "1" | sudo tee /sys/devices/system/cpu/intel_pstate/no_turbo
```

I put these in two separate files and then made both executable. I moved them into a directory that was on my path.

How To

# lxc container setup

I prefer to purge openssh-server and then reinstall it as the default config requires preshared keys for all users which isn't really something I desire and it's simpler to just trash the default configs for containers and reinstall openssh-server from scratch.

If you get this error

```
sudo: no tty present and no askpass program specified
```

You can create a file like this

```
echo "import pty; pty.spawn('/bin/bash')" > /tmp/pty_spawn.py
```

And then run

```
python /tmp/pty_spawn.py
```

to switch to a terminal type where users can elevate privilege. Note that this is only a problem if you got into the container using `lxc exec --`.

This is not an issue if you ssh into the container directly.

How To

# Run docker container inside of lxc container

- Create an unprivileged container with nesting turned on
- Run docker containers as normal

How To

# Lock packages in zypper

To prevent updates to packages in zypper, you can use the `al` command to add a lock.

```
sudo zypper al texlive*
```

to view locks, you can use the `ll` command

```
sudo zypper ll
```

To remove a lock use the `rl` command

```
sudo zypper rl texlive*
```

# Installing matrix-synapse on pypy

```
sudo apt install virtualenv
virtualenv -p ./pypy3 ~/synapse/env
source ~/synapse/env/bin/activate
pip install --upgrade pip
pip install --upgrade setuptools
sudo apt install libjpeg-dev libxslt-dev libxml2-dev
postgresql-server-dev-all
pip install matrix-synapse[all]
```

Currently, this is working. However, adding rooms is causing issues but only with some rooms. The error generated looks like this.

```
2019-02-17 18:15:11,331 - synapse.access.http.8008 - 233 - INFO - GET-5 - 192.168.1.254 -
8008 - Received request: GET /_matrix/client/r0/groups/world/profile2019-02-17 18:15:11,334 -
synapse.http.server - 112 - ERROR - GET-5 - Failed handle request via <function
JsonResource._async_render at 0x00007fae97c4f240>: <XForwardedForRequest at 0x7fae94986950
method='GET' uri='/_matrix/client/r0/groups/world/profile' clientproto='HTTP/1.0' site=8008>:
Traceback (most recent call last): File "/home/kenneth/synapse/env/site-
packages/twisted/internet/defer.py", line 1418, in _inlineCallbacks result =
g.send(result) File "/home/kenneth/synapse/env/site-packages/synapse/http/server.py", line
316, in _async_render
    callback_return = yield callback(request, **kwargs) File "/home/kenneth/synapse/env/site-
packages/twisted/internet/defer.py", line 1613, in unwindGenerator return
_cancellableInlineCallbacks(gen) File "/home/kenneth/synapse/env/site-
packages/twisted/internet/defer.py", line 1529, in _cancellableInlineCallbacks
_ininlineCallbacks(None, g, status)
--- <exception caught here> --- File "/home/kenneth/synapse/env/site-
packages/synapse/http/server.py", line 81, in wrapped_request_handler yield h(self,
request) File "/home/kenneth/synapse/env/site-packages/synapse/http/server.py", line 316, in
_async_render
```

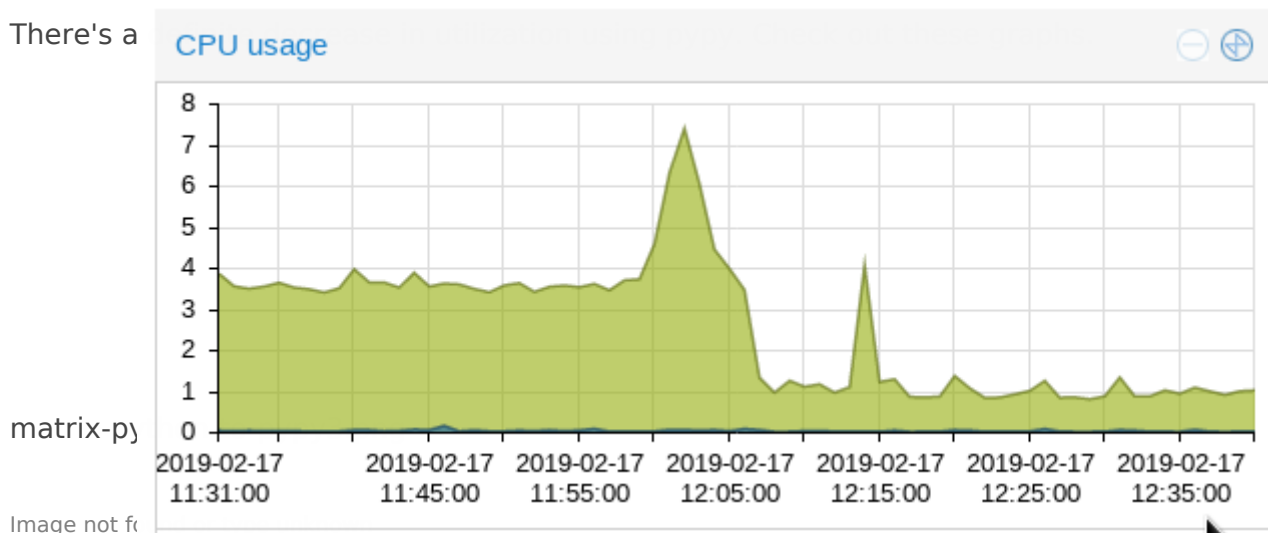
```

callback_return = yield callback(request, **kwargs) File "/home/kenneth/synapse/env/site-
packages/twisted/internet/defer.py", line 1418, in _inlineCallbacks    result =
g.send(result) File "/home/kenneth/synapse/env/site-
packages/synapse/rest/client/v2_alpha/groups.py", line 47, in on_GET
    requester_user_id, File "/home/kenneth/synapse/env/site-
packages/synapse/handlers/groups_local.py", line 34, in f
    if self.is_mine_id(group_id): File "/home/kenneth/synapse/env/site-
packages/synapse/server.py", line 235, in is_mine_id    return string.split(":", 1)[1] ==
self.hostname
builtins.IndexError: list index out of range

```

If I switch to a regular version of python to add the room and then switch back to pypy, I can talk to the room just fine.

There's a



# Download from nextcloud using wget

Get a public nextcloud  
share link

append `/download` to the url

`wget` this new url



How To

# Install printer drivers for brother printer in opensuse tumbleweed

Get glibc and associated libraries in 32 bit form.

```
sudo zypper install -y glibc-32bit
```

Download brother printer driver installer

```
wget https://download.brother.com/welcome/dlf006893/linux-brprinter-installer-2.2.1-1.gz
gunzip linux-brprinter-installer-*.gz
sudo bash linux-brprinter-installer-*
```

See [this page](#) for more information with regard to troubleshooting linux printer drivers.

How To

# Install numpy-blis for amd efficiency

```
conda create -c conda-forge -n numpy-blis numpy "blas*=blis" python=3.7
```

How To

# Allennlp

## Fix errors with empty params list

The error `ValueError: optimizer got an empty parameter list` from allennlp does not relate to the parameters from your json config file. The parameters in question are the tensors to be optimized per `torch/optim/optimizer.py`. This typically is caused by having a mismatch in field names.

How To

# Install GTKWattman on opensuse tumbleweed

This is what I tried. it actually doesn't work. I even tried installing gnome and this still didn't work. not sure what's going on.

```
zypper in python3-gobject
zypper in python3-pycairo-devel
zypper in gobject-introspection-devel
python3 -m venv venv_wattman
source venv_wattman/bin/activatepython -m pip install --upgrade matplotlib setuptools pycairo
git clone https://github.com/BoukeHaarsma23/WattmanGTK.git
cd WattmanGTK
pip -m install -e .
```

# How to use Intel's mkl library on AMD systems

By default, the mkl library will pick a very unoptimal path on AMD processors, causing libraries like openblas to perform much better. However, if you set the appropriate environment variables, mkl will be forced to use the appropriate code path for amd cpus like Zen and Epyc.

The solution is to run `export MKL_DEBUG_CPU_TYPE=5` before running your script.

This was pointed out in a comment on [this puget systems article](#)

“ Nice! I have recently seen some people recommending using MKL on AMD, with the MKL\_DEBUG\_CPU\_TYPE environment variable set to 5, as in:

```
export MKL_DEBUG_CPU_TYPE=5
```

This overrides the CPU dispatching in MKL, and forces the AVX2 codepath (the one MKL naturally uses on Intel parts without AVX512), otherwise MKL chooses an unoptimized SSE path with abysmal performance. But with the AVX2 path, MKL performs very well on Zen2, usually even outperforming BLIS and OpenBLAS!

How To

# Install pytorch-rocm on bare metal opensuse Tumbleweed

These instructions are adopted from section 4 of [this page](#).

pytorch commit 9d1138afec26a4fe0be74187e4064076f8d45de7 worked for some stuff but pieces of allennlp are incompatible because this is pytorch v1.7.0a0+9d1138a.

I tried with 1.5 and 1.5.1 several times but it wasn't working on opensuse. I could have sworn it worked on ubuntu though.

## Add rocm repos

Using the files provided by AMD for SLES SP1 per [the official instructions](#).

```
sudo zypper install dkms
sudo zypper cleansudo zypper addrepo --no-gpgcheck http://repo.radeon.com/rocm/zyp/zypper/rocm
sudo zypper ref
sudo zypper install rocm-dkms
sudo reboot
```

Modify `/etc/modprobe.d/10-unsupported-modules.conf` to have

```
allow_unsupported_modules 1
```

Then run the following, though it's probably not strictly necessary on tumbleweed

```
sudo modprobe amdgpu
```

Add your user to the video group

```
usermod -a -G video <username>
```

Verify everything is working by examining the output of `rocm-smi` and make sure your gpu is listed.

# Create a virtual environment

```
virtualenv -p python3 ~/venvs/torch  
source ~/venvs/torch/bin/activate
```

# Install pytorch prerequisites

```
sudo zypper in glog-devel python3-pip libopenblas-devel libprotobuf-devel libnuma-devel  
libpthread-stubs0-devel libopencv-devel git gcc cmake make lmdb-devel libleveldb1 snappy-  
devel hiredis-develsudo zypper in rocm-dev rocm-libs miopen-hip hipsparse rocthrust hipcub  
rccl roctracer-dev
```

# Fix issues with cmake files for rocm

```
sed -i 's/find_dependency(hip)/find_dependency(HIP)/g'  
/opt/rocm/rocsparse/lib/cmake/rocsparse/rocsparse-config.cmake sed -i  
's/find_dependency(hip)/find_dependency(HIP)/g' /opt/rocm/rocfft/lib/cmake/rocfft/rocfft-  
config.cmake sed -i 's/find_dependency(hip)/find_dependency(HIP)/g'
```

```
/opt/rocm/miopen/lib/cmake/miopen/miopen-config.cmake sed -i  
's/find_dependency(hip)/find_dependency(HIP)/g' /opt/rocm/roclblas/lib/cmake/roclblas/roclblas-  
config.cmake sed -i 's/find_dependency(hip)/find_dependency(HIP)/g'  
/opt/rocm/rccl/lib/cmake/rccl/rccl-config.cmake sed -i  
's/find_dependency(hip)/find_dependency(HIP)/g'  
/opt/rocm/hipsparse/lib/cmake/hipsparse/hipsparse-config.cmake
```

# Clone the repo

```
git clone https://github.com/pytorch/pytorch.git  
cd pytorch  
git checkout v1.5.0  
git submodule update --init --recursive
```

# Build

This process will take a while (2-3 hours)

```
export RCCL_DIR="/opt/rocm/rccl/lib/cmake"  
python tools/amd_build/build_amd.pyUSE_ROCM=1 USE_LMDB=1 USE_OPENCV=1 MAX_JOBS=4 python  
setup.py install
```

# Install allennlp

```
pip install allennlp  
pip uninstall torch  
# rebuild torch (very fast this time)USE_ROCM=1 USE_LMDB=1 USE_OPENCV=1 MAX_JOBS=4 python  
setup.py install
```



# Verify installation

Make sure you get a non-zero value (should correspond to the number of GPUs).

```
python  
>>import torch  
>>torch.cuda.device_count()
```

How To

# Install aftermarket cooler on M40

<http://translate.google.com/translate?hl=en&sl=auto&tl=en&u=https%3A%2F%2Ftweakers.net%2Ftesla-m40-24gb.html>

How To

# Use Intel Quad bypass cards (82571EB PRO)

## Download latest intel e1000e drivers

These drivers are available on [sourceforge](#).

## Identify the parameters of your particular adapter

## Identify the exact model you have

Run `sudo lshw -C network` and identify the full name of your network card. Mine ended up being `82571EB PRO/1000 AT`.

Then, look in `/usr/share/misc/pci.ids` for that code and pull out the pci-id for that code (in this case `10a0`).

# Modify drivers & compile

Uncompress the tar'd driver.

```
tar -xzf
```

Move into `src`

```
cd src
```

Modify the definition of `E1000_DEV_ID_82571EB_QUAD_COPPER` in `hw.h`.

```
echo "#define E1000_DEV_ID_82571EB_QUAD_COPPER 0x10A0" >>hw.h
```

Compile the module and install

```
sudo make install
```

## reload module

```
sudo modprobe -r e1000e; sudo modprobe e1000e
```

Your nics should show up as `DISABLED` instead of `UNCLAIMED`.

How To

# Build pytorch with rocm on ubuntu 20.04

I used rocm 3.8, pytorch 1.6 from the rocm repo (apparently upstream doesn't work right now.).

How To

# Enable hibernation on asus zenbook with ryzen processor

I was having an issue where I was unable to hibernate my asus zenbook with ryzen 5 4500u and mx350 gpu. However, the issue turned out to be that dracut was not compiling the resume module in and thus it was unable to resume from whatever image is saved to swap.

[This reddit post](#) was helpful for finding a fix.

Largely, it comes down to the following two commands

```
sudo echo "add_dracutmodules+=\" resume \"" > /etc/dracut.conf.d/99-fix-resume.conf  
sudo dracut -fv
```

# Set up zyxel travel router in client bridged mode

[Tom's hardware thread where this was discussed](#)

Solution update: To enable Client mode (to bridge my existing wifi network to Ethernet), follow these steps:

- Flip the switch on the ZyXEL's side to AP mode.
- Log in to the ZyXEL's setup page @ 192.168.100.1 (the directions recommend giving your computer a static IP address first, 192.168.100.10)
- From the sidebar, click Wireless>Basic Settings
- Look for the "wireless mode" setting halfway down; click where it says "AP" and pick "Client" from the dropdown menu and then click "apply"
- From the sidebar, click Wireless>Site Survey
- Click the Site Survey button on this page to refresh the list of wifi networks in range.
- Click the bubble to the right of your network, and then the Next button to get the password screen
- Match your encryption method, key length, and type of password.
- Click finish, and if you typed in your password and settings riiiiight... You're done!

# Get my EGPU to work on opensuse tumbleweed with a framework 12th gen laptop

- With the laptop on, plug it in to the usbc cable
- load into a terminal and load the nvidia module `sudo modprobe nvidia`
- Verify that gpu is working using `nvidia-smi`. This may require installing nvidia-compute-utils
- Run `prime-select nvidia`
- log out of desktop
- start a terminal (ctrl alt f2)
- login
- run `startx`



# Planning

These are plans for purchases and infrastructure changes

# NAS

## Goals:

- I want to have a way to store my files for machines that are ephemeral
  - For example, *imara* is a node that is only on when I have analytics work to do. It would be ideal to store the results of the work done on these servers in a centralized location to enable access even when machines are powered off.
- I want to have scalability to large amounts of storage
- I want to have the throughput potential to use this as a storage array for containers if necessary
- I don't have much money to spend as I save for the wedding

# Server refresh

I would like to get away from my dl380 g7 as it is somewhat loud and quite energy inefficient.

Currently, I am considering two options.

The first is to build a machine for deep learning and then use my z620 as a server.

The second is to use a [deskmini](#) as a server.

There are a couple of things that need to be considered.

## z620 advantages

- Air intake from front only, allowing for stacking
- More memory expansion (64gb with single processor, 96 with two)
- Much more upgradable (additional processor, 16 cores, gpus)
- Upgrades are cheaper in general
- Intel nic
- More drives

## Deskmini advantages

- Much newer processor architecture
- Integrated gpu for jellyfin transcoding
- Much lower power consumption
- NVMe storage (2x)
- No intel management engine

Planning

# SLURM

[https://slurm.schedmd.com/quickstart\\_admin.html](https://slurm.schedmd.com/quickstart_admin.html)

2019

2019

# July 2019

## July 19 2019

Mwanafunzi is listing nvidia-docker, docker-ce, and nvidia-docker-runtime as upgradable packages. Currently, since I am away from home and using this machine extensively, I will not be upgrading these packages. Do not upgrade them until a better solution using btrfs snapshots can be put in place (with redundant boot drives).

## July 20 2019

Rebuilt sharelatex from docker-compose files so that it uses the correct hostname. This only became an issue when doing external signups.

I tried to set up smtp with sharelatex so that password resets could be automated but I did not have much luck in getting it set up.

## July 21 2019

Running parsing genie on google cloud with port 80 requires running as root since listening on that port is reserved for root by default.

## July 23 2019

Git annex sync --content on newly cloned repositories is not working at the moment. It is first

saying that

# Zen 2 build

<https://pcpartpicker.com/list/9XXD6s>

# Threadripper build

<https://pcpartpicker.com/list/Lr7tcY>

# August 2 2019

gave Nick Howell my id\_rsa public key which was signed by Fran. The key is in  
/home/kenneth/.ssh/id\_rsa\_nhowel.pub on kiti.

2019

# August 2019

New gtx 1070 blower in Mwanfunzi. With both GPUs under 99% load the power consumption is 415 watts.

At idle we're at 78 watts.



2019

# September 2019

Plans for redoing the environment:

- Need to set up LDAP authorization source
- Need to migrate existing users into LDAP

## Goals

- Large networked storage
- BTRFS on mwanafunzi root
  - Allow for easy snapshotting
- Fix boot when backup drive is connected

## Purchases

- 8tb WD Elements

2019

# October 2019

Lizardfs running smoothly with 2 tb of raw storage.

## todo

---

- Add both 4TB drives that I have into nodes of the cluster.
- Buy small ssd
- Set up proxmox on both optiplexes
- Consider buying a new desktop so that the current optiplex can be repurposed
- power off dl380

250 parallel crf tagging runs took 2168.2 minutes on 12 cores. returned the error message:

```
Traceback (most recent call last):
  File "crf_tag.py", line 54, in <module>
    main()
  File "crf_tag.py", line 51, in main    print(metrics.flat_classification_report(test_tags,
pred_tags, digits=8))  File "/home/kenneth/.local/lib/python3.6/site-
packages/sklearn_crfsuite/metrics.py", line 13, in wrapper    return func(y_true_flat,
y_pred_flat, *args, **kwargs)  File "/home/kenneth/.local/lib/python3.6/site-
packages/sklearn_crfsuite/metrics.py", line 68, in flat_classification_report    return
metrics.classification_report(y_true, y_pred, labels, **kwargs)  File
"/home/kenneth/.local/lib/python3.6/site-packages/sklearn/metrics/classification.py", line
1568, in classification_report
    name_width = max(len(cn) for cn in target_names)
ValueError: max() arg is an empty sequence
```

2019

# NOvember 2019

roku ip 192.168.2.5 user rokudev

2019

# December

## Parser crash post mortem

### Lexicon

```
ring{String}["N.sg"],"women" => SubString{String}["N.pl"],"guitar" =>
SubString{String}["N.sg"],"seen" => SubString{String}["V.trans.ed"],"short" =>
SubString{String}["Adj"],"thinks" => SubString{String}["V.trans", "V.clausal"],"big" =>
SubString{String}["Adj"],"a" => SubString{String}["Det"],"thinking" =>
SubString{String}["V.trans.ing", "V.clausal.ing"],"looked" =>
SubString{String}["V.trans.ed"],"plumbers" => SubString{String}["N.pl"],"yellow" =>
SubString{String}["Adj"],"this" => SubString{String}["Det"],"throws" =>
SubString{String}["V.trans", "V.ditrans"],"eat" => SubString{String}["V.trans.bare",
"V.intrans.bare"],"massive" => SubString{String}["Adj"],"look" =>
SubString{String}["V.intrans.bare", "V.stative.bare"],"under" =>
SubString{String}["P"],"called" => SubString{String}["V.attrib.ed"],"sleep" =>
SubString{String}["V.intrans.bare"],"the" => SubString{String}["Det"],"laughing" =>
SubString{String}["V.intrans.ing"],"talks" => SubString{String}["V.intrans"],"stone" =>
SubString{String}["N.sg"],"calls" => SubString{String}["V.attrib"],"house" =>
SubString{String}["N.sg"],"sleeps" => SubString{String}["V.intrans"],"talk" =>
SubString{String}["V.intrans.bare"])Dec 09 18:17:11 parser server[4846]:
Dict{Any,Any}("compliments" => SubString{String}["V.trans"],"pulls" =>
SubString{String}["V.trans"],"pull" => SubString{String}["V.trans.bare"],"grab" =>
SubString{String}["V.trans.bare"],"plumber" => SubString{String}["N.sg"],"many" =>
SubString{String}["Det"],"that" => SubString{String}["Det"],"buy" =>
SubString{String}["V.trans.bare"],"three" => SubString{String}["Adj"],"cats" =>
SubString{String}["N.pl"],"runs" => SubString{String}["V.trans", "V.intrans"],"seems" =>
SubString{String}["V.stative"],"to" => SubString{String}["INF"],"sees" =>
SubString{String}["V.trans", "V.intrans"],"laugh" =>
```

```

SubString{String}["V.intrans.bare"],"dogs" => SubString{String}["N.pl"],"call" =>
SubString{String}["V.attrib.bare"],"is" => SubString{String}["V.stative",
"Aux.prog"],"pushes" => SubString{String}["V.trans"],"looking" =>
SubString{String}["V.trans.ing"],"calling" => SubString{String}["V.attrib.ing"],"throw" =>
SubString{String}["V.ditrans.bare"],"looks" => SubString{String}["V.intrans",
"V.stative"],"hill" => SubString{String}["N.sg"],"in" => SubString{String}["P"],"wants" =>
SubString{String}["V.inf"],"has" => SubString{String}["Aux.perf"],"push" =>
SubString{String}["V.trans.bare"],"buys" => SubString{String}["V.trans"],"dog" =>
SubString{String}["N.sg"],"catches" => SubString{String}["V.trans"],"cat" =>
SubString{String}["N.sg"],"foot" => SubString{String}["N.sg"],"on" =>
SubString{String}["P"],"run" => SubString{String}["V.trans.bare", "V.intrans.bare"],"see" =>
SubString{String}["V.trans.bare", "V.intrans.bare"],"eats" => SubString{String}["V.trans",
"V.intrans"],"thought" => SubString{String}["V.clausal.ed"],"pasted" =>
SubString{String}["V.trans.ed"],"bought" => SubString{String}["V.trans.ed"],"seeing" =>
SubString{String}["V.trans.ing"],"think" => SubString{String}["V.clausal.bare"],"been" =>
SubString{String}["Perf.prog"],"laughs" => SubString{String}["V.intrans"],"grabs" =>
SubString{String}["V.trans"],"compliment" => SubString{String}["V.trans.bare"],"catch" =>
SubString{String}["V.trans.bare"],"woman" => SubSt

```

## sentences

```
Any[["the", "cat", "has", "been", "looking", "in", "the", "house"]]
```

## Productions

```

Dict{Any,Any}{"S" => Array{String,1}[["NP", "VP"],"NP" => Array{String,1}[["Det", "N.sg"],
["Det", "N.pl"], ["Det", "N.sg", "PP"], ["Det", "N.pl", "PP"], ["Det", "Adj", "N.sg"],
["Det", "Adj", "N.pl"], ["N.pl"], ["Det", "N.sg"], ["Det", "Adj", "N.sg"], ["Det", "N.sg",
"PP"], ["Det", "Adj", "N.sg", "PP"], ["N.pl"], ["Det", "N.pl"], ["Adj", "N.pl"], ["Det",
"Adj", "N.pl"], ["N.pl", "PP"], ["Det", "N.pl", "PP"], ["Adj", "N.pl", "PP"], ["Det", "Adj",
"N.pl", "PP"]],"PP" => Array{String,1}[["P", "NP"]],"VP" => Array{String,1}[["V.trans", "S"],
["V.trans", "NP", "NP"], ["V.intrans"], ["V.trans", "NP"], ["V.stative", "Adj"], ["V.attrib",
"NP", "Adj"], ["V.inf", "INF", "V.intrans.bare", "NP"], ["V.inf", "INF", "V.intrans.bare",
"PP"], ["V.inf", "INF", "VP"], ["V.clausal.bare", "NP", "V.trans", "NP"], ["V.attrib.bare",
"NP", "Adj"], ["Aux.prog", "V.trans.ing", "PP"], ["Aux.prog", "V.trans.ing", "NP"],

```

```
["Aux.prog", "VP"], ["V.attrib.ing", "NP", "Adj"], ["Aux.prog", "V.attrib.ing", "NP"],
["Aux.prog", "V.clausal.ing", "NP", "V.trans", "NP"], ["Aux.perf", "V.trans.ed", "PP"],
["Aux.perf", "V.trans.ed", "NP"], ["Aux.perf", "VP"], ["V.attrib.ed", "NP", "Adj"],
["Aux.perf", "V.clausal.ed", "S"], ["Aux.perf", "Perf.prog", "V.trans.ing"], ["VP", "PP"],
["Aux.perf", "Perf.prog", "V.trans.ing", "NP"], ["VP", "NP"], ["Aux.perf", "Perf.prog",
"V.intrans.ing"], ["VP"]])
```

## Relevant piece of stacktrace

```
|97 |V.trans.ing-> looking* <== Int64[]|98 |VP ->Aux.perf Perf.prog
V.trans.ing* <== [57, 93, 97]|99 |VP ->Aux.perf Perf.prog V.trans.ing*NP
<== [57, 93, 97]
|100 |S -> NP VP* <== [25, 98]
|101 |VP -> VP*PP <== [98]
|102 |VP -> VP*NP <== [98]
|103 |VP -> VP* <== [98]
|104 |NP -> *Det N.sg <== Int64[]|105 |NP -> *Det N.pl <== Int64[]
|106 |NP -> *Det N.sg PP <== Int64[]|107 |NP -> *Det N.pl PP <== Int64[]
|108 |NP -> *Det Adj N.sg <== Int64[]|109 |NP -> *Det Adj N.pl <== Int64[]
|110 |NP -> *N.pl <== Int64[]|111 |NP -> *Det Adj N.sg PP <==
Int64[]
|112 |NP -> *Adj N.pl <== Int64[]|113 |NP -> *N.pl PP <== Int64[]
|114 |NP -> *Adj N.pl PP <== Int64[]|115 |NP -> *Det Adj N.pl PP <==
Int64[]
|116 |γ -> S* <== [100]
|117 |PP -> *P NP <== Int64[]|118 |S -> NP VP* <== [25, 103]
|119 |VP -> VP*PP <== [103]
|120 |VP -> VP*NP <== [103]
|121 |VP -> VP* <== [103]|122 |γ -> S* <== [118]
-----
, -----|123 |P -> in* <== Int64[]
-----
```

2019

# 04 April

## Quanta Windmill node 2

Currently, I am running node 2 of my quanta windmill on a different motherboard with the working bios chip. I'm only using two ram sticks (4gb total) and one of the e5-2609 processors. So far everything seems fine. The working motherboard is #3.

The original non-working motherboard is #4.

The numbers are written on the serial tags on the motherboards.

## Update for e5-2650's

Unfortunately, the motherboard that was working has a bent pin on the second processor socket. I have switched to the e5-2650 instead of the e5-2609 that was in there and everything seems to be working fine. However, I do not think that adding in another cpu will work given that the pin is bent.

Some of my ram is also not working. I am running fine right now on four of the 2gb sticks but with all 8 put in, the system does not want to boot.

# z620

I am working on building a deep learning machine for doing research. I ended up purchasing a z620 for \$200 with an e5-2620 installed, 16 gigabytes of ecc ddr3 and a 512 gigabyte hard drive. There are two 16x pcie v3 slots in the motherboard for this computer and two six pin power connectors.

However, most modern gpus require an 8 pin power connector.

6 pin connectors are rated for 75 watts, 8 pin power connectors are rated for 150 watts but the extra pins are just additional grounds. The 6 pin power connectors on the z620 are designed for 18 amps at 12 volts (which comes out to 216 watts). In addition, they provide the three power lines that are needed to feed this kind of wattage. I bought two 6 to 8 pin converters to use with these overspeced 6 pin connectors.

One problem that has came up is that the gtx 980 that I got from Jake requires an 8 pin and a 6 pin connector. This seems like too much since I'd like to have the flexibility to add a second gpu.



2019

# 03 March 2019

## Error setting up Plume

Currently getting

```
thread 'main' panicked at 'called `Result::unwrap()` on an `Err` value:
BadType("production.secret_key", "a 256-bit base64 encoded string", "string", None)',
src/libcore/result.rs:1009:5
note: Run with `RUST_BACKTRACE=1` for a backtrace.
```

on launch

Fixed, it was an error with the ROCKET\_SECRET\_KEY

## Customized CSS

Currently I have a fork of plume running with a separate branch called `mod_css` on github. This is where I'm keeping my modified version of plume.

## Issues with curl?

I've been having some issues with zypper hitting a segmentation fault during the download stage of the distribution upgrade.

I have also been having issues with NetworkManager consuming 100% of my cpu when running on wifi.

By downgrading curl from version 7.64.0-4.1 to version 7.64.0-1.1 by running

```
sudo zypper install --oldpackage curl-7.64.0-1.1
```

This appears to have fixed the network manager issue. I also added a lock on curl since it continues to want update curl to this latest broken version.

This definitely seems to have resolved the issue with zypper as well. I just did a distribution upgrade with 450 packages and didn't run into a segmentation fault a single time. I would have had about 20 segfaults before.

2019

# May 2019

## Install cuda on ubuntu 19.04

### install drivers

```
sudo apt install lightdm
```

This is required because gdm3 does not work well with nvidia's proprietary drivers but lightdm does. When you install lightdm, there will be a prompt to determine what desktop manager should be the default. Select lightdm.

```
sudo add-apt-repository ppa:graphics-drivers/ppa
sudo apt install dkms build-essential
sudo apt update
sudo apt install nvidia-driver-418
```

Before updating, I like to make sure I'll be able to get back into my system. By default, ubuntu 19 comes with a 0 second timeout on grub. Which is stupid. edit `/etc/default/grub` and change `GRUB_TIMEOUT` to something higher than 0. This will allow you to get into recovery mode easily if you break your system.

Then run,

```
sudo update-grub
```

to make your changes to the grub config take affect.

Then reboot your system.

# May todo

Switch backup drive to deduped zfs with compression

Redo backup scripts to automate upload to backblaze

Figure out whether to keep fileservers

After running stress -32 on imara, the front processor was 77 and the back was 88.

# Current Machines

- Dell precision rack 7910
  - dual xeon e5-2623 v3's
  - 16 GB DDR4 (1Rx8 PC4-2133P)
  - 1 1TB SATA hard drive
    - lizardfs storage
  - 1 1TB SATA SSD
    - unused
  - 2 512GB SATA hard drives
    - zfs raid 1 as proxmox root
- Dell optiplex 9010 MT
  - i7 3770
  - 16 GB DDR3
  - 512 GB SSD
    - lvm thing partition for proxmox
  - 4 TB HDD
    - lizardfs storage
- Dell optiplex 9010 SFF
  - i5 3450
  - 8GB DDR3
  - 512 GB SSD
    - lvm thin partition for proxmox
  - 4TB HDD
    - lizardfs storage
- HP z620

- Single E5-2650
- 16 GB single rank DDR3
- GTX 1070
- GTX 1070
- 128 GB SSD
- three 1TB SATA HDD
- Quanta Opencompute Windmill
  - Node 1
    - Dual E5-2680's
    - 76 GB DDR3
    - 1050ti
    - Mixed rankings?
    - 1 TB SATA HDD
  - Node 2
    - E5-2650
    - 16 GB DDR3
    - 500 GB SATA HDD
- DL380 G7 (decommissioned):
  - dual xeon x5675's
  - 32 GB DDR3
  - 5 600 GB SAS drives
  - 1 1TB SATA drive (online backups)
- UCS c240 m3 (decommissioned)
  - Dual E5-2650's
  - 32 GB DDR3
  - Some issues with ram stability
- DL360 G8
  - 8 GB DDR3
  - No drives available (never purchased Generation 8 caddies)
  - Dual E5-2620
  - p420i
- DL580 G7
  - Broken

# Benchmarks

# Laptop ram upgrade

## before upgrade:

- unigine heaven: - 9.4 fps - 236 score - min 5.4 - max 21.2 - 1920 x 1080
- universe sandbox 37

## after upgrade

- unigine heaven
  - 11.1 fps
  - 281 score
  - min 7.0
  - max 25.2
  - 1920 x 1080
- universe sandbox 39



# nvidia apex

Scarecrow 1123 wrote a trainer for allennlp that uses nvidia's apex package to enable mixed precision training.

The full gist is available [here](#).

This is a copy of the trainer provided..

I find that my models are more often successful if I specify "O1" instead of "O2" for amp. This uses only a set of whitelisted operations in half precision mode.

[This trainer](#) has the change already made.

To use this during training include a snippet like this in your training json config.

```
{
  // ....
  "trainer": {
    "type": "fp16-trainer",
    "mixed_precision": true,
    // other options
  }
  // ....
}
```

and make sure the trainer is in a directory that you are including using `--include-package`.

For a bert model I was training, it ran out of VRAM on a single GTX 1070 without apex configured. However with apex configured the model was only using 4.5GB. There was no discernable penalty with regard to the number of epochs required though I haven't investigated a ton.

# CFG.jl

grammar size: rules: 52 lexicon: 66

parsing execution times on a single thread

size	time
300 sents	1.172 seconds
3,000 sents	2.988 seconds
30,000 sents	22.06 seconds
300,000 sents	208.03 seconds

parsing execution times on two threads

size	time
300 sents	1.000 seconds
3,000 sents	2.216 seconds
30,000 sents	13.874 seconds
300,000 sents	127.376 seconds

# ROCm pytorch

Used this tutorial to install pytorch for rocm, however I checked out release 1.5.

<https://github.com/ROCmSoftwarePlatform/pytorch/wiki/Building-PyTorch-for-ROCm> Allennlp was version 0.9.

## GRU

## BERT

This used bert-base with a batch size of 8.

## Vega FE notes

The vega frontier edition results were obtained from a rented gpueater instance.

A batch size of 16 was also tried for the vega frontier edition to see if it would fit in vram and strangely the time per epoch dropped (01:12) with the larger batch size). This was also with thermal throttling as the vega fe was hitting 87 C and the clocks were down to 1.2 Ghz from 1.6 Ghz. The fans were limited to 40% under load on gpueater.com. It would be interesting to see what the performance is like with better thermals.

GRU	GRU		
BERT	pospos-		
base	tagtagger		
emoti	(1- (2-		
egres	hidhid)		

GTX 1070	1:26:00.2	00:24.3	
Tesla M40	1:32:00.0	00:04.3	
RTX 3090	0:20:20.0	00:02.6	
RX580	1:14:00.0	00:08.5	
Vega Frontier	1:29:30.0	00:05.1	
Vega Frontier (90% fans)	1:09:00.0	00:03.0	
Vega frontier (rocm 4.0)	1:07:50.0	00:02.9	
i7- 7800x	00:18:00.0	00:00:23	
i9- 7900x (defective?)	00:19:00.0	00:00:23	
i9- 7900x	00:16:00.0	00:00:20	
i9- 7980xe	00:15:00.0	00:00:18	
e5- 2680v3	00:20:00.0	00:00:34	

using rocm apex gave no discernable performance improvement (with use\_apex = true) However, it did reduce memory consumption by ~1GB for a batch of 16.

The RTX 3090 was tested with cuda 11, all other nvidia gpus were using cuda 10.2 (the RTX 3090 is not supported in this earlier version of cuda).

# Linpack results

role	Platform	Rank	GFLOP/s	Notes
				No
e5-2690	48GB	1333	1333	avx2
	4000MHz	1521	1521	
	x4			this
				cpu
				limited
				by
				memory
e5-2680v3	16GB	2133	2133	size.
	3000MHz	2732	2732	Not
	x4			the
				peak
				performance
				4.1
				3200Ghz
7835	32GB	3200	3200	avx-
	x4			512
				clock
				322933
7940	40GB	2000	570.5	No
	x2			overclock
				323200
7940	40GB	2000	660.7	No
	x4			overclock
				3200
7940	128GB	2000	1275	Rented
	x4			from
				vast.ai

# lizardfs-benchmarks

# 4 nodes

Testgroup "current"

=== 1 file series ===

1 file, 1 thread, seq 1M writes, simple: 19 write iops (10% HDD) 1 file, 1 thread, rnd 16k writes, simple: 268 write iops (37% HDD) 1 file, 1 thread, rnd 16k writes, simple, take 2: 264 write iops (40% HDD)

1 file, 16 threads, rnd 4k writes, posixaio: 335 write iops (95% HDD) 1 file, 16 threads, rnd 8k writes, posixaio: 348 write iops (82% HDD) 1 file, 16 threads, rnd 16k writes, posixaio: 280 write iops (62% HDD) 1 file, 16 threads, rnd 16k writes, posixaio, take 2: 312 write iops (61% HDD)

=== 16 file series === 16 files, 1 thread each, seq 1M writes, simple: 25 write iops (16% HDD)

16 files, 1 thread each, rnd 16k writes, simple: 2594 write iops (396% HDD) 16 files, 1 thread each, rnd 16k writes, simple, take 2: 1121 write iops (178% HDD) 16 files, 1 thread each, rnd 16k writes, posixaio: 842 write iops (180% HDD) 16 files, 16 threads each, rnd 16k writes, posixaio: 4824 write iops (898% HDD) 16 files, 16 threads each, rnd 16k writes, posixaio, take 2: 4992 write iops (996% HDD)

=== O\_SYNC series === 1 file, 1 thread, rnd 16k writes, simple, o\_sync: 269 write iops (173% HDD)

1 file, 16 threads, rnd 16k writes, posixaio, o\_sync: 264 write iops (109% HDD) 16 files, 1 thread each, rnd 16k writes, simple, o\_sync: 1078 write iops (213% HDD) 16 files, 16 threads each, rnd 16k writes, posixaio, o\_sync: 1008 write iops (198% HDD)

=== read series ===

1 file, 1 thread, seq 1M reads, simple: 3 read iops (0% HDD) 1 file, 16 threads, rnd 16k reads, posixaio: 370 read iops (216% HDD) 16 files, 1 thread each, seq 1M reads, simple: 104 read iops (2% HDD) 16 files, 1 thread each, rnd 16k reads, posixaio: 67170 read iops (14291% HDD)

16 files, 16 threads each, rnd 16k reads, posixaio: 72861 read iops (14689% HDD) === native aio series ===

1 file, 16 threads, rnd 16k writes, native aio: 280 write iops (45% HDD) 16 files, 16 threads each, rnd 16k writes, native aio: 2827 write iops (564% HDD) 1 file, 16 threads, rnd 16k reads, native aio: 365 read iops (278% HDD) 16 files, 16 threads each, rnd 16k reads, native aio: 80553 read iops (16306% HDD)

Tests complete on karatasi @ 2019-09-22 12:11:31. Files remain. To clean up, add argument

"cleanup".

```
kenneth@karatasi:/mnt/liz-client/backup> ./storage-tuner-benchmark here . Running tests in
"./stb-testdir" on karatasi @ 2019-09-23 22:24:04 ...storage-tuner-benchmark version 2.1.0
Testgroup "current"
=== 1 file series ===
1 file, 1 thread, seq 1M writes, simple: 9 write iops (4% HDD)1 file, 1 thread, rnd 16k
writes, simple: 272 write iops (38% HDD)1 file, 1 thread, rnd 16k writes, simple, take 2: 276
write iops (41% HDD)
1 file, 16 threads, rnd 4k writes, posixaio: 412 write iops (117% HDD)1 file, 16 threads, rnd
8k writes, posixaio: 389 write iops (91% HDD)1 file, 16 threads, rnd 16k writes, posixaio:
308 write iops (68% HDD)1 file, 16 threads, rnd 16k writes, posixaio, take 2: 325 write iops
(64% HDD)
=== 16 file series ===16 files, 1 thread each, seq 1M writes, simple: 19 write iops (12%
HDD)
16 files, 1 thread each, rnd 16k writes, simple: 1919 write iops (293% HDD)16 files, 1 thread
each, rnd 16k writes, simple, take 2: 1367 write iops (218% HDD)16 files, 1 thread each, rnd
16k writes, posixaio: 1181 write iops (253% HDD)16 files, 16 threads each, rnd 16k writes,
posixaio: 5032 write iops (937% HDD)16 files, 16 threads each, rnd 16k writes, posixaio, take
2: 4982 write iops (994% HDD)
=== O_SYNC series ===1 file, 1 thread, rnd 16k writes, simple, o_sync: 177 write iops
(114% HDD)
1 file, 16 threads, rnd 16k writes, posixaio, o_sync: 168 write iops (69% HDD)16 files, 1
thread each, rnd 16k writes, simple, o_sync: 994 write iops (196% HDD)16 files, 16 threads
each, rnd 16k writes, posixaio, o_sync: 1009 write iops (198% HDD)
=== read series ===
1 file, 1 thread, seq 1M reads, simple: 5 read iops (0% HDD)1 file, 16 threads, rnd 16k
reads, posixaio: 420 read iops (245% HDD)16 files, 1 thread each, seq 1M reads, simple: 6
read iops (0% HDD)16 files, 1 thread each, rnd 16k reads, posixaio: 55074 read iops (11717%
HDD)
16 files, 16 threads each, rnd 16k reads, posixaio: 58892 read iops (11873% HDD)=== native
aio series ===
1 file, 16 threads, rnd 16k writes, native aio: 212 write iops (34% HDD)16 files, 16 threads
each, rnd 16k writes, native aio: 2972 write iops (593% HDD)1 file, 16 threads, rnd 16k
reads, native aio: 429 read iops (327% HDD)16 files, 16 threads each, rnd 16k reads, native
aio: 64134 read iops (12982% HDD)
Tests complete on karatasi @ 2019-09-23 22:27:45.Files remain. To clean up, add argument
"cleanup".
```



storage-tuner-benchmark version 2.1.0

Testgroup "current"

=== 1 file series ===

1 file, 1 thread, seq 1M writes, simple: 20 write iops (11% HDD)1 file, 1 thread, rnd 16k writes, simple: 270 write iops (37% HDD)1 file, 1 thread, rnd 16k writes, simple, take 2: 245 write iops (37% HDD)

1 file, 16 threads, rnd 4k writes, posixaio: 356 write iops (101% HDD)1 file, 16 threads, rnd 8k writes, posixaio: 332 write iops (78% HDD)1 file, 16 threads, rnd 16k writes, posixaio: 313 write iops (70% HDD)1 file, 16 threads, rnd 16k writes, posixaio, take 2: 298 write iops (59% HDD)

=== 16 file series ===16 files, 1 thread each, seq 1M writes, simple: 28 write iops (18% HDD)

16 files, 1 thread each, rnd 16k writes, simple: 2603 write iops (398% HDD)16 files, 1 thread each, rnd 16k writes, simple, take 2: 873 write iops (139% HDD)16 files, 1 thread each, rnd 16k writes, posixaio: 974 write iops (209% HDD)16 files, 16 threads each, rnd 16k writes, posixaio: 3089 write iops (575% HDD)16 files, 16 threads each, rnd 16k writes, posixaio, take 2: 4961 write iops (990% HDD)

=== O\_SYNC series ===1 file, 1 thread, rnd 16k writes, simple, o\_sync: 265 write iops (170% HDD)

1 file, 16 threads, rnd 16k writes, posixaio, o\_sync: 274 write iops (113% HDD)16 files, 1 thread each, rnd 16k writes, simple, o\_sync: 968 write iops (191% HDD)16 files, 16 threads each, rnd 16k writes, posixaio, o\_sync: 846 write iops (166% HDD)

=== read series ===

1 file, 1 thread, seq 1M reads, simple: 4 read iops (0% HDD)1 file, 16 threads, rnd 16k reads, posixaio: 356 read iops (208% HDD)16 files, 1 thread each, seq 1M reads, simple: 165 read iops (4% HDD)16 files, 1 thread each, rnd 16k reads, posixaio: 58710 read iops (12491% HDD)

16 files, 16 threads each, rnd 16k reads, posixaio: 62699 read iops (12640% HDD)=== native aio series ===

1 file, 16 threads, rnd 16k writes, native aio: 260 write iops (42% HDD)16 files, 16 threads each, rnd 16k writes, native aio: 3222 write iops (643% HDD)1 file, 16 threads, rnd 16k reads, native aio: 363 read iops (277% HDD)16 files, 16 threads each, rnd 16k reads, native aio: 72852 read iops (14747% HDD)

Tests complete on karatasi @ 2019-09-25 20:32:59. Files remain. To clean up, add argument "cleanup".

lizardfs-benchmarks

# 5 nodes

storage-tuner-benchmark version 2.1.0

Testgroup "current"

=== 1 file series ===

1 file, 1 thread, seq 1M writes, simple: 57 write iops (31% HDD)1 file, 1 thread, rnd 16k writes, simple: 15 write iops (2% HDD)1 file, 1 thread, rnd 16k writes, simple, take 2: 7 write iops (1% HDD)

1 file, 16 threads, rnd 4k writes, posixaio: 17 write iops (4% HDD)1 file, 16 threads, rnd 8k writes, posixaio: 19 write iops (4% HDD)1 file, 16 threads, rnd 16k writes, posixaio: 16 write iops (3% HDD)1 file, 16 threads, rnd 16k writes, posixaio, take 2: 16 write iops (3% HDD)

=== 16 file series ===16 files, 1 thread each, seq 1M writes, simple: 35 write iops (23% HDD)

16 files, 1 thread each, rnd 16k writes, simple: 111 write iops (16% HDD)16 files, 1 thread each, rnd 16k writes, simple, take 2: 322 write iops (51% HDD)16 files, 1 thread each, rnd 16k writes, posixaio: 311 write iops (66% HDD)16 files, 16 threads each, rnd 16k writes, posixaio: 1016 write iops (189% HDD)16 files, 16 threads each, rnd 16k writes, posixaio, take 2: 1551 write iops (309% HDD)

=== 0\_SYNC series ===1 file, 1 thread, rnd 16k writes, simple, o\_sync: 14 write iops (9% HDD)

1 file, 16 threads, rnd 16k writes, posixaio, o\_sync: 14 write iops (5% HDD)16 files, 1 thread each, rnd 16k writes, simple, o\_sync: 271 write iops (53% HDD)16 files, 16 threads each, rnd 16k writes, posixaio, o\_sync: 156 write iops (30% HDD)

=== read series ===

1 file, 1 thread, seq 1M reads, simple: 657 read iops (17% HDD)1 file, 16 threads, rnd 16k reads, posixaio: 1347 read iops (787% HDD)16 files, 1 thread each, seq 1M reads, simple: 175 read iops (4% HDD)16 files, 1 thread each, rnd 16k reads, posixaio: 67676 read iops (14399% HDD)

16 files, 16 threads each, rnd 16k reads, posixaio: 75265 read iops (15174% HDD)=== native aio series ===

1 file, 16 threads, rnd 16k writes, native aio: 19 write iops (3% HDD)16 files, 16 threads each, rnd 16k writes, native aio: 358 write iops (71% HDD)1 file, 16 threads, rnd 16k reads, native aio: 1486 read iops (1134% HDD)16 files, 16 threads each, rnd 16k reads, native aio: 78257 read iops (15841% HDD)

Testgroup "current"

=== 1 file series ===

1 file, 1 thread, seq 1M writes, simple: 72 write iops (39% HDD)1 file, 1 thread, rnd 16k writes, simple: 20 write iops (2% HDD)1 file, 1 thread, rnd 16k writes, simple, take 2: 17 write iops (2% HDD)

```

1 file, 16 threads, rnd 4k writes, posixaio: 18 write iops (5% HDD)1 file, 16 threads, rnd 8k
writes, posixaio: 12 write iops (2% HDD)1 file, 16 threads, rnd 16k writes, posixaio: 22
write iops (4% HDD)1 file, 16 threads, rnd 16k writes, posixaio, take 2: 17 write iops (3%
HDD)
=== 16 file series ===16 files, 1 thread each, seq 1M writes, simple: 35 write iops (23%
HDD)
16 files, 1 thread each, rnd 16k writes, simple: 382 write iops (58% HDD)16 files, 1 thread
each, rnd 16k writes, simple, take 2: 366 write iops (58% HDD)16 files, 1 thread each, rnd
16k writes, posixaio: 343 write iops (73% HDD)16 files, 16 threads each, rnd 16k writes,
posixaio: 1301 write iops (242% HDD)16 files, 16 threads each, rnd 16k writes, posixaio, take
2: 1126 write iops (224% HDD)
=== 0_SYNC series ===1 file, 1 thread, rnd 16k writes, simple, o_sync: 14 write iops (9%
HDD)
1 file, 16 threads, rnd 16k writes, posixaio, o_sync: 11 write iops (4% HDD)16 files, 1
thread each, rnd 16k writes, simple, o_sync: 376 write iops (74% HDD)16 files, 16 threads
each, rnd 16k writes, posixaio, o_sync: 198 write iops (38% HDD)
=== read series ===
1 file, 1 thread, seq 1M reads, simple: 228 read iops (6% HDD)1 file, 16 threads, rnd 16k
reads, posixaio: 807 read iops (471% HDD)16 files, 1 thread each, seq 1M reads, simple: 173
read iops (4% HDD)16 files, 1 thread each, rnd 16k reads, posixaio: 63355 read iops (13479%
HDD)
16 files, 16 threads each, rnd 16k reads, posixaio: 70160 read iops (14145% HDD)=== native
aio series ===
1 file, 16 threads, rnd 16k writes, native aio: 22 write iops (3% HDD)16 files, 16 threads
each, rnd 16k writes, native aio: 450 write iops (89% HDD)1 file, 16 threads, rnd 16k reads,
native aio: 782 read iops (596% HDD)16 files, 16 threads each, rnd 16k reads, native aio:
73098 read iops (14797% HDD)
Tests complete on karatasi @ 2019-09-23 23:05:57.Files remain. To clean up, add argument
"cleanup".

```

```

Creating test directory "stb-testdir"Running tests in "./stb-testdir" on mwanafunzi @ 2019-09-
25 21:54:20 ...
storage-tuner-benchmark version 2.1.0
Testgroup "current"
=== 1 file series ===
1 file, 1 thread, seq 1M writes, simple: 85 write iops (46% HDD)1 file, 1 thread, rnd 16k
writes, simple: 24 write iops (3% HDD)1 file, 1 thread, rnd 16k writes, simple, take 2: 30
write iops (4% HDD)
1 file, 16 threads, rnd 4k writes, posixaio: 27 write iops (7% HDD)1 file, 16 threads, rnd 8k

```

```

writes, posixaio: 25 write iops (5% HDD)1 file, 16 threads, rnd 16k writes, posixaio: 32
write iops (7% HDD)1 file, 16 threads, rnd 16k writes, posixaio, take 2: 25 write iops (4%
HDD)
=== 16 file series ===16 files, 1 thread each, seq 1M writes, simple: 34 write iops (22%
HDD)
16 files, 1 thread each, rnd 16k writes, simple: 947 write iops (144% HDD)16 files, 1 thread
each, rnd 16k writes, simple, take 2: 902 write iops (143% HDD)16 files, 1 thread each, rnd
16k writes, posixaio: 949 write iops (203% HDD)16 files, 16 threads each, rnd 16k writes,
posixaio: 2767 write iops (515% HDD)16 files, 16 threads each, rnd 16k writes, posixaio, take
2: 2729 write iops (544% HDD)
=== 0_SYNC series ===1 file, 1 thread, rnd 16k writes, simple, o_sync: 25 write iops (16%
HDD)
1 file, 16 threads, rnd 16k writes, posixaio, o_sync: 28 write iops (11% HDD)16 files, 1
thread each, rnd 16k writes, simple, o_sync: 893 write iops (176% HDD)16 files, 16 threads
each, rnd 16k writes, posixaio, o_sync: 650 write iops (127% HDD)
=== read series ===
1 file, 1 thread, seq 1M reads, simple: 105 read iops (2% HDD)1 file, 16 threads, rnd 16k
reads, posixaio: 468 read iops (273% HDD)16 files, 1 thread each, seq 1M reads, simple: 7409
read iops (183% HDD)16 files, 1 thread each, rnd 16k reads, posixaio: 151158 read iops
(32161% HDD)16 files, 16 threads each, rnd 16k reads, posixaio: 440291 read iops (88768% HDD)
=== native aio series ===1 file, 16 threads, rnd 16k writes, native aio: 25 write iops (4%
HDD)
16 files, 16 threads each, rnd 16k writes, native aio: 818 write iops (163% HDD)1 file, 16
threads, rnd 16k reads, native aio: 508 read iops (387% HDD)16 files, 16 threads each, rnd
16k reads, native aio: 508793 read iops (102994% HDD)Tests complete on mwanafunzi @ 2019-09-
25 21:57:13.
Files remain. To clean up, add argument "cleanup".

```

Laptop was running at 100 mb instead of 1Gb. Took that node offline and ran with 4 including imara2

```

Running tests in "./stb-testdir" on mwanafunzi @ 2019-09-25 21:58:11 ...storage-tuner-
benchmark version 2.1.0
Testgroup "current"
=== 1 file series ===
1 file, 1 thread, seq 1M writes, simple: 79 write iops (43% HDD)1 file, 1 thread, rnd 16k
writes, simple: 25 write iops (3% HDD)1 file, 1 thread, rnd 16k writes, simple, take 2: 25
write iops (3% HDD)
1 file, 16 threads, rnd 4k writes, posixaio: 28 write iops (7% HDD)1 file, 16 threads, rnd 8k

```

```
writes, posixaio: 27 write iops (6% HDD)1 file, 16 threads, rnd 16k writes, posixaio: 28
write iops (6% HDD)1 file, 16 threads, rnd 16k writes, posixaio, take 2: 26 write iops (5%
HDD)
=== 16 file series ===16 files, 1 thread each, seq 1M writes, simple: 87 write iops (58%
HDD)
16 files, 1 thread each, rnd 16k writes, simple: 1267 write iops (193% HDD)16 files, 1 thread
each, rnd 16k writes, simple, take 2: 1267 write iops (202% HDD)16 files, 1 thread each, rnd
16k writes, posixaio: 1276 write iops (273% HDD)16 files, 16 threads each, rnd 16k writes,
posixaio: 5135 write iops (956% HDD)16 files, 16 threads each, rnd 16k writes, posixaio, take
2: 5267 write iops (1051% HDD)
=== 0_SYNC series ===1 file, 1 thread, rnd 16k writes, simple, o_sync: 25 write iops (16%
HDD)
1 file, 16 threads, rnd 16k writes, posixaio, o_sync: 22 write iops (9% HDD)16 files, 1
thread each, rnd 16k writes, simple, o_sync: 1180 write iops (233% HDD)16 files, 16 threads
each, rnd 16k writes, posixaio, o_sync: 850 write iops (166% HDD)
=== read series ===
1 file, 1 thread, seq 1M reads, simple: 230 read iops (6% HDD)1 file, 16 threads, rnd 16k
reads, posixaio: 480 read iops (280% HDD)16 files, 1 thread each, seq 1M reads, simple: 10396
read iops (257% HDD)16 files, 1 thread each, rnd 16k reads, posixaio: 152363 read iops
(32417% HDD)16 files, 16 threads each, rnd 16k reads, posixaio: 443535 read iops (89422% HDD)
=== native aio series ===1 file, 16 threads, rnd 16k writes, native aio: 23 write iops (3%
HDD)
16 files, 16 threads each, rnd 16k writes, native aio: 1262 write iops (251% HDD)1 file, 16
threads, rnd 16k reads, native aio: 448 read iops (341% HDD)16 files, 16 threads each, rnd
16k reads, native aio: 480207 read iops (97207% HDD)Tests complete on mwanafunzi @ 2019-09-25
22:00:54.
Files remain. To clean up, add argument "cleanup".
```

```
Creating test directory "stb-testdir"Running tests in "./stb-testdir" on mwanafunzi @ 2019-09-
25 22:01:40 ...
storage-tuner-benchmark version 2.1.0
Testgroup "current"
=== 1 file series ===
1 file, 1 thread, seq 1M writes, simple: 103 write iops (56% HDD)1 file, 1 thread, rnd 16k
writes, simple: 33 write iops (4% HDD)1 file, 1 thread, rnd 16k writes, simple, take 2: 34
write iops (5% HDD)
1 file, 16 threads, rnd 4k writes, posixaio: 37 write iops (10% HDD)1 file, 16 threads, rnd
8k writes, posixaio: 37 write iops (8% HDD)1 file, 16 threads, rnd 16k writes, posixaio: 32
write iops (7% HDD)1 file, 16 threads, rnd 16k writes, posixaio, take 2: 38 write iops (7%
```

```

HDD)
=== 16 file series ===16 files, 1 thread each, seq 1M writes, simple: 86 write iops (57%
HDD)
16 files, 1 thread each, rnd 16k writes, simple: 641 write iops (98% HDD)16 files, 1 thread
each, rnd 16k writes, simple, take 2: 603 write iops (96% HDD)16 files, 1 thread each, rnd
16k writes, posixaio: 566 write iops (121% HDD)16 files, 16 threads each, rnd 16k writes,
posixaio: 3117 write iops (580% HDD)16 files, 16 threads each, rnd 16k writes, posixaio, take
2: 2441 write iops (487% HDD)
=== 0_SYNC series ===1 file, 1 thread, rnd 16k writes, simple, o_sync: 33 write iops (21%
HDD)
1 file, 16 threads, rnd 16k writes, posixaio, o_sync: 30 write iops (12% HDD)16 files, 1
thread each, rnd 16k writes, simple, o_sync: 600 write iops (118% HDD)16 files, 16 threads
each, rnd 16k writes, posixaio, o_sync: 154 write iops (30% HDD)
=== read series ===
1 file, 1 thread, seq 1M reads, simple: 122 read iops (3% HDD)1 file, 16 threads, rnd 16k
reads, posixaio: 529 read iops (309% HDD)16 files, 1 thread each, seq 1M reads, simple: 9609
read iops (237% HDD)16 files, 1 thread each, rnd 16k reads, posixaio: 110624 read iops
(23537% HDD)16 files, 16 threads each, rnd 16k reads, posixaio: 544756 read iops (109829% HDD)
=== native aio series ===1 file, 16 threads, rnd 16k writes, native aio: 36 write iops (5%
HDD)
16 files, 16 threads each, rnd 16k writes, native aio: 662 write iops (132% HDD)1 file, 16
threads, rnd 16k reads, native aio: 311 read iops (237% HDD)16 files, 16 threads each, rnd
16k reads, native aio: 525425 read iops (106361% HDD)Tests complete on mwanafunzi @ 2019-09-
25 22:04:27.
Files remain. To clean up, add argument "cleanup".

```

```

Creating test directory "stb-testdir"Running tests in "./stb-testdir" on mwanafunzi @ 2019-09-
25 22:12:45 ...
storage-tuner-benchmark version 2.1.0
Testgroup "current"
=== 1 file series ===
1 file, 1 thread, seq 1M writes, simple: 108 write iops (59% HDD)1 file, 1 thread, rnd 16k
writes, simple: 47 write iops (6% HDD)1 file, 1 thread, rnd 16k writes, simple, take 2: 40
write iops (6% HDD)
1 file, 16 threads, rnd 4k writes, posixaio: 37 write iops (10% HDD)1 file, 16 threads, rnd
8k writes, posixaio: 52 write iops (12% HDD)1 file, 16 threads, rnd 16k writes, posixaio: 47
write iops (10% HDD)1 file, 16 threads, rnd 16k writes, posixaio, take 2: 51 write iops (10%
HDD)
=== 16 file series ===16 files, 1 thread each, seq 1M writes, simple: 95 write iops (63%

```

HDD)

16 files, 1 thread each, rnd 16k writes, simple: 1187 write iops (181% HDD)16 files, 1 thread each, rnd 16k writes, simple, take 2: 1164 write iops (185% HDD)16 files, 1 thread each, rnd 16k writes, posixaio: 676 write iops (145% HDD)16 files, 16 threads each, rnd 16k writes, posixaio: 3062 write iops (570% HDD)16 files, 16 threads each, rnd 16k writes, posixaio, take 2: 2219 write iops (442% HDD)

=== O\_SYNC series ===1 file, 1 thread, rnd 16k writes, simple, o\_sync: 45 write iops (29% HDD)

1 file, 16 threads, rnd 16k writes, posixaio, o\_sync: 40 write iops (16% HDD)16 files, 1 thread each, rnd 16k writes, simple, o\_sync: 409 write iops (80% HDD)16 files, 16 threads each, rnd 16k writes, posixaio, o\_sync: 312 write iops (61% HDD)

=== read series ===

1 file, 1 thread, seq 1M reads, simple: 91 read iops (2% HDD)1 file, 16 threads, rnd 16k reads, posixaio: 555 read iops (324% HDD)16 files, 1 thread each, seq 1M reads, simple: 7014 read iops (173% HDD)16 files, 1 thread each, rnd 16k reads, posixaio: 2688 read iops (571% HDD)

16 files, 16 threads each, rnd 16k reads, posixaio: 5759 read iops (1161% HDD)=== native aio series ===

1 file, 16 threads, rnd 16k writes, native aio: 41 write iops (6% HDD)16 files, 16 threads each, rnd 16k writes, native aio: 958 write iops (191% HDD)1 file, 16 threads, rnd 16k reads, native aio: 252 read iops (192% HDD)16 files, 16 threads each, rnd 16k reads, native aio: 221980 read iops (44935% HDD)

Tests complete on mwanafunzi @ 2019-09-25 22:15:27.Files remain. To clean up, add argument "cleanup".

=== 1 file series ===

1 file, 1 thread, seq 1M writes, simple: 103 write iops (56% HDD)1 file, 1 thread, rnd 16k writes, simple: 30 write iops (4% HDD)1 file, 1 thread, rnd 16k writes, simple, take 2: 30 write iops (4% HDD)

1 file, 16 threads, rnd 4k writes, posixaio: 37 write iops (10% HDD)1 file, 16 threads, rnd 8k writes, posixaio: 34 write iops (8% HDD)1 file, 16 threads, rnd 16k writes, posixaio: 30 write iops (6% HDD)1 file, 16 threads, rnd 16k writes, posixaio, take 2: 30 write iops (5% HDD)

=== 16 file series ===16 files, 1 thread each, seq 1M writes, simple: 91 write iops (60% HDD)

16 files, 1 thread each, rnd 16k writes, simple: 636 write iops (97% HDD)16 files, 1 thread each, rnd 16k writes, simple, take 2: 660 write iops (105% HDD)16 files, 1 thread each, rnd 16k writes, posixaio: 581 write iops (124% HDD)16 files, 16 threads each, rnd 16k writes, posixaio: 3014 write iops (561% HDD)16 files, 16 threads each, rnd 16k writes, posixaio, take



2: 2336 write iops (466% HDD)

=== O\_SYNC series ===1 file, 1 thread, rnd 16k writes, simple, o\_sync: 29 write iops (18% HDD)

1 file, 16 threads, rnd 16k writes, posixaio, o\_sync: 32 write iops (13% HDD)16 files, 1 thread each, rnd 16k writes, simple, o\_sync: 609 write iops (120% HDD)16 files, 16 threads each, rnd 16k writes, posixaio, o\_sync: 379 write iops (74% HDD)

=== read series ===

1 file, 1 thread, seq 1M reads, simple: 1198 read iops (32% HDD)1 file, 16 threads, rnd 16k reads, posixaio: 594 read iops (347% HDD)16 files, 1 thread each, seq 1M reads, simple: 7023 read iops (173% HDD)16 files, 1 thread each, rnd 16k reads, posixaio: 230192 read iops (48977% HDD)16 files, 16 threads each, rnd 16k reads, posixaio: 588258 read iops (118600% HDD)

=== native aio series ===1 file, 16 threads, rnd 16k writes, native aio: 34 write iops (5% HDD)

16 files, 16 threads each, rnd 16k writes, native aio: 610 write iops (121% HDD)1 file, 16 threads, rnd 16k reads, native aio: 600 read iops (458% HDD)16 files, 16 threads each, rnd 16k reads, native aio: 605657 read iops (122602% HDD)

# Comparison benchmarks

Using [this script](#)

## Single NVME

```
Testgroup "current"
=== 1 file series ===
1 file, 1 thread, seq 1M writes, simple: 788 write iops (435% HDD)1 file, 1 thread, rnd 16k
writes, simple: 2222 write iops (312% HDD)1 file, 1 thread, rnd 16k writes, simple, take 2:
2216 write iops (336% HDD)1 file, 16 threads, rnd 4k writes, posixaio: 5911 write iops (1684%
HDD)
1 file, 16 threads, rnd 8k writes, posixaio: 5890 write iops (1389% HDD)1 file, 16 threads,
rnd 16k writes, posixaio: 5446 write iops (1218% HDD)1 file, 16 threads, rnd 16k writes,
posixaio, take 2: 5450 write iops (1081% HDD)
=== 16 file series ===16 files, 1 thread each, seq 1M writes, simple: 1361 write iops
(907% HDD)
16 files, 1 thread each, rnd 16k writes, simple: 6853 write iops (1047% HDD)16 files, 1
thread each, rnd 16k writes, simple, take 2: 6926 write iops (1104% HDD)16 files, 1 thread
each, rnd 16k writes, posixaio: 6001 write iops (1287% HDD)16 files, 16 threads each, rnd 16k
writes, posixaio: 7825 write iops (1457% HDD)16 files, 16 threads each, rnd 16k writes,
posixaio, take 2: 7798 write iops (1556% HDD)
=== O_SYNC series ===1 file, 1 thread, rnd 16k writes, simple, o_sync: 1935 write iops
(1248% HDD)
1 file, 16 threads, rnd 16k writes, posixaio, o_sync: 2020 write iops (838% HDD)16 files, 1
thread each, rnd 16k writes, simple, o_sync: 5718 write iops (1132% HDD)16 files, 16 threads
each, rnd 16k writes, posixaio, o_sync: 5538 write iops (1088% HDD)
=== read series ===
1 file, 1 thread, seq 1M reads, simple: 1022 read iops (27% HDD)1 file, 16 threads, rnd 16k
reads, posixaio: 4845 read iops (2833% HDD)16 files, 1 thread each, seq 1M reads, simple:
1218 read iops (30% HDD)16 files, 1 thread each, rnd 16k reads, posixaio: 57716 read iops
(12280% HDD)16 files, 16 threads each, rnd 16k reads, posixaio: 63905 read iops (12884% HDD)
```

```
=== native aio series ===1 file, 16 threads, rnd 16k writes, native aio: 2190 write iops
(357% HDD)
16 files, 16 threads each, rnd 16k writes, native aio: 6817 write iops (1360% HDD)1 file, 16
threads, rnd 16k reads, native aio: 5994 read iops (4575% HDD)16 files, 16 threads each, rnd
16k reads, native aio: 61152 read iops (12378% HDD)Tests complete on linux-k9r1 @ 2019-10-02
23:20:45.
Files remain. To clean up, add argument "cleanup".
```

```
=== 1 file series ===
1 file, 1 thread, seq 1M writes, simple: 820 write iops (453% HDD)1 file, 1 thread, rnd 16k
writes, simple: 2258 write iops (317% HDD)1 file, 1 thread, rnd 16k writes, simple, take 2:
2243 write iops (340% HDD)1 file, 16 threads, rnd 4k writes, posixaio: 6301 write iops (1795%
HDD)
1 file, 16 threads, rnd 8k writes, posixaio: 6203 write iops (1462% HDD)1 file, 16 threads,
rnd 16k writes, posixaio: 5711 write iops (1277% HDD)1 file, 16 threads, rnd 16k writes,
posixaio, take 2: 5651 write iops (1121% HDD)
=== 16 file series ===16 files, 1 thread each, seq 1M writes, simple: 1358 write iops
(905% HDD)
16 files, 1 thread each, rnd 16k writes, simple: 6920 write iops (1058% HDD)16 files, 1
thread each, rnd 16k writes, simple, take 2: 6941 write iops (1107% HDD)16 files, 1 thread
each, rnd 16k writes, posixaio: 6068 write iops (1302% HDD)16 files, 16 threads each, rnd 16k
writes, posixaio: 7612 write iops (1417% HDD)16 files, 16 threads each, rnd 16k writes,
posixaio, take 2: 7568 write iops (1510% HDD)
=== 0_SYNC series ===1 file, 1 thread, rnd 16k writes, simple, o_sync: 1942 write iops
(1252% HDD)
1 file, 16 threads, rnd 16k writes, posixaio, o_sync: 2015 write iops (836% HDD)16 files, 1
thread each, rnd 16k writes, simple, o_sync: 5842 write iops (1156% HDD)16 files, 16 threads
each, rnd 16k writes, posixaio, o_sync: 5675 write iops (1114% HDD)
=== read series ===
1 file, 1 thread, seq 1M reads, simple: 1032 read iops (27% HDD)1 file, 16 threads, rnd 16k
reads, posixaio: 4760 read iops (2783% HDD)16 files, 1 thread each, seq 1M reads, simple:
1211 read iops (29% HDD)16 files, 1 thread each, rnd 16k reads, posixaio: 58127 read iops
(12367% HDD)16 files, 16 threads each, rnd 16k reads, posixaio: 64106 read iops (12924% HDD)
=== native aio series ===1 file, 16 threads, rnd 16k writes, native aio: 2259 write iops
(368% HDD)
16 files, 16 threads each, rnd 16k writes, native aio: 6435 write iops (1284% HDD)1 file, 16
threads, rnd 16k reads, native aio: 6169 read iops (4709% HDD)16 files, 16 threads each, rnd
16k reads, native aio: 62023 read iops (12555% HDD)
```

# SATA SSD

```
Testgroup "current"
=== 1 file series ===
1 file, 1 thread, seq 1M writes, simple: 134 write iops (74% HDD)1 file, 1 thread, rnd 16k
writes, simple: 207 write iops (29% HDD)1 file, 1 thread, rnd 16k writes, simple, take 2: 230
write iops (34% HDD)
1 file, 16 threads, rnd 4k writes, posixaio: 347 write iops (98% HDD)1 file, 16 threads, rnd
8k writes, posixaio: 367 write iops (86% HDD)1 file, 16 threads, rnd 16k writes, posixaio:
270 write iops (60% HDD)1 file, 16 threads, rnd 16k writes, posixaio, take 2: 273 write iops
(54% HDD)
=== 16 file series ===16 files, 1 thread each, seq 1M writes, simple: 57 write iops (38%
HDD)
16 files, 1 thread each, rnd 16k writes, simple: 726 write iops (111% HDD)16 files, 1 thread
each, rnd 16k writes, simple, take 2: 958 write iops (152% HDD)16 files, 1 thread each, rnd
16k writes, posixaio: 568 write iops (121% HDD)16 files, 16 threads each, rnd 16k writes,
posixaio: 759 write iops (141% HDD)16 files, 16 threads each, rnd 16k writes, posixaio, take
2: 774 write iops (154% HDD)
=== O_SYNC series ===1 file, 1 thread, rnd 16k writes, simple, o_sync: 102 write iops
(65% HDD)
1 file, 16 threads, rnd 16k writes, posixaio, o_sync: 108 write iops (44% HDD)16 files, 1
thread each, rnd 16k writes, simple, o_sync: 546 write iops (108% HDD)16 files, 16 threads
each, rnd 16k writes, posixaio, o_sync: 665 write iops (130% HDD)
=== read series ===
1 file, 1 thread, seq 1M reads, simple: 3826 read iops (102% HDD)1 file, 16 threads, rnd 16k
reads, posixaio: 3725 read iops (2178% HDD)16 files, 1 thread each, seq 1M reads, simple:
5693 read iops (140% HDD)16 files, 1 thread each, rnd 16k reads, posixaio: 333893 read iops
(71041% HDD)16 files, 16 threads each, rnd 16k reads, posixaio: 672383 read iops (135561% HDD)
=== native aio series ===1 file, 16 threads, rnd 16k writes, native aio: 192 write iops (31%
HDD)
16 files, 16 threads each, rnd 16k writes, native aio: 895 write iops (178% HDD)1 file, 16
threads, rnd 16k reads, native aio: 3638 read iops (2777% HDD)16 files, 16 threads each, rnd
16k reads, native aio: 941116 read iops (190509% HDD)Tests complete on mwanafunzi @ 2019-10-
03 17:49:55.
Files remain. To clean up, add argument "cleanup".
```

# BTRFS RAID 1 HDD

```
Testgroup "current"
=== 1 file series ===
1 file, 1 thread, seq 1M writes, simple: 89 write iops (49% HDD)1 file, 1 thread, rnd 16k
writes, simple: 19 write iops (2% HDD)1 file, 1 thread, rnd 16k writes, simple, take 2: 17
write iops (2% HDD)
1 file, 16 threads, rnd 4k writes, posixaio: 287 write iops (81% HDD)1 file, 16 threads, rnd
8k writes, posixaio: 285 write iops (67% HDD)1 file, 16 threads, rnd 16k writes, posixaio:
254 write iops (56% HDD)1 file, 16 threads, rnd 16k writes, posixaio, take 2: 257 write iops
(50% HDD)
=== 16 file series ===16 files, 1 thread each, seq 1M writes, simple: 64 write iops (42%
HDD)
16 files, 1 thread each, rnd 16k writes, simple: 166 write iops (25% HDD)16 files, 1 thread
each, rnd 16k writes, simple, take 2: 167 write iops (26% HDD)16 files, 1 thread each, rnd
16k writes, posixaio: 178 write iops (38% HDD)16 files, 16 threads each, rnd 16k writes,
posixaio: 1043 write iops (194% HDD)16 files, 16 threads each, rnd 16k writes, posixaio, take
2: 946 write iops (188% HDD)
=== O_SYNC series ===1 file, 1 thread, rnd 16k writes, simple, o_sync: 16 write iops (10%
HDD)
1 file, 16 threads, rnd 16k writes, posixaio, o_sync: 19 write iops (7% HDD)16 files, 1
thread each, rnd 16k writes, simple, o_sync: 158 write iops (31% HDD)16 files, 16 threads
each, rnd 16k writes, posixaio, o_sync: 168 write iops (33% HDD)
=== read series ===
1 file, 1 thread, seq 1M reads, simple: 19 read iops (0% HDD)1 file, 16 threads, rnd 16k
reads, posixaio: 123 read iops (71% HDD)16 files, 1 thread each, seq 1M reads, simple: 4 read
iops (0% HDD)
16 files, 1 thread each, rnd 16k reads, posixaio: 474 read iops (100% HDD)16 files, 16
threads each, rnd 16k reads, posixaio: 466 read iops (93% HDD)
=== native aio series ===1 file, 16 threads, rnd 16k writes, native aio: 22 write iops (3%
HDD)
16 files, 16 threads each, rnd 16k writes, native aio: 176 write iops (35% HDD)1 file, 16
threads, rnd 16k reads, native aio: 115 read iops (87% HDD)16 files, 16 threads each, rnd 16k
reads, native aio: 479 read iops (96% HDD)Tests complete on mwanafunzi @ 2019-10-03 17:47:04.
Files remain. To clean up, add argument "cleanup".
```

# XFS HDD

Testgroup "current"

=== 1 file series ===

1 file, 1 thread, seq 1M writes, simple: 152 write iops (83% HDD) 1 file, 1 thread, rnd 16k writes, simple: 118 write iops (16% HDD) 1 file, 1 thread, rnd 16k writes, simple, take 2: 117 write iops (17% HDD)

1 file, 16 threads, rnd 4k writes, posixaio: 160 write iops (45% HDD) 1 file, 16 threads, rnd 8k writes, posixaio: 159 write iops (37% HDD) 1 file, 16 threads, rnd 16k writes, posixaio: 155 write iops (34% HDD) 1 file, 16 threads, rnd 16k writes, posixaio, take 2: 155 write iops (30% HDD)

=== 16 file series === 16 files, 1 thread each, seq 1M writes, simple: 117 write iops (78% HDD)

16 files, 1 thread each, rnd 16k writes, simple: 174 write iops (26% HDD) 16 files, 1 thread each, rnd 16k writes, simple, take 2: 171 write iops (27% HDD) 16 files, 1 thread each, rnd 16k writes, posixaio: 111 write iops (23% HDD) 16 files, 16 threads each, rnd 16k writes, posixaio: 212 write iops (39% HDD) 16 files, 16 threads each, rnd 16k writes, posixaio, take 2: 205 write iops (40% HDD)

=== 0\_SYNC series === 1 file, 1 thread, rnd 16k writes, simple, o\_sync: 30 write iops (19% HDD)

1 file, 16 threads, rnd 16k writes, posixaio, o\_sync: 30 write iops (12% HDD) 16 files, 1 thread each, rnd 16k writes, simple, o\_sync: 104 write iops (20% HDD) 16 files, 16 threads each, rnd 16k writes, posixaio, o\_sync: 108 write iops (21% HDD)

=== read series ===

1 file, 1 thread, seq 1M reads, simple: 165 read iops (4% HDD) 1 file, 16 threads, rnd 16k reads, posixaio: 136 read iops (79% HDD) 16 files, 1 thread each, seq 1M reads, simple: 132 read iops (3% HDD)

16 files, 1 thread each, rnd 16k reads, posixaio: 284 read iops (60% HDD) 16 files, 16 threads each, rnd 16k reads, posixaio: 316 read iops (63% HDD)

=== native aio series === 1 file, 16 threads, rnd 16k writes, native aio: 121 write iops (19% HDD)

16 files, 16 threads each, rnd 16k writes, native aio: 192 write iops (38% HDD) 1 file, 16 threads, rnd 16k reads, native aio: 139 read iops (106% HDD) 16 files, 16 threads each, rnd 16k reads, native aio: 300 read iops (60% HDD)

# Moosefs comparison

absolutely absurdly faster performance compared to lizardfs.

```
Testgroup "current"
=== 1 file series ===
1 file, 1 thread, seq 1M writes, simple: 101 write iops (55% HDD)1 file, 1 thread, rnd 16k
writes, simple: 393 write iops (55% HDD)1 file, 1 thread, rnd 16k writes, simple, take 2: 386
write iops (58% HDD)
1 file, 16 threads, rnd 4k writes, posixaio: 5422 write iops (1544% HDD)1 file, 16 threads,
rnd 8k writes, posixaio: 4401 write iops (1037% HDD)1 file, 16 threads, rnd 16k writes,
posixaio: 3239 write iops (724% HDD)1 file, 16 threads, rnd 16k writes, posixaio, take 2:
3238 write iops (642% HDD)
=== 16 file series ===16 files, 1 thread each, seq 1M writes, simple: 110 write iops (73%
HDD)
16 files, 1 thread each, rnd 16k writes, simple: 5096 write iops (779% HDD)16 files, 1 thread
each, rnd 16k writes, simple, take 2: 5127 write iops (817% HDD)16 files, 1 thread each, rnd
16k writes, posixaio: 5204 write iops (1116% HDD)16 files, 16 threads each, rnd 16k writes,
posixaio: 7020 write iops (1307% HDD)16 files, 16 threads each, rnd 16k writes, posixaio,
take 2: 7015 write iops (1400% HDD)
=== 0_SYNC series ===1 file, 1 thread, rnd 16k writes, simple, o_sync: 372 write iops
(240% HDD)
1 file, 16 threads, rnd 16k writes, posixaio, o_sync: 355 write iops (147% HDD)16 files, 1
thread each, rnd 16k writes, simple, o_sync: 5183 write iops (1026% HDD)16 files, 16 threads
each, rnd 16k writes, posixaio, o_sync: 5199 write iops (1021% HDD)
=== read series ===
1 file, 1 thread, seq 1M reads, simple: 1515 read iops (40% HDD)1 file, 16 threads, rnd 16k
reads, posixaio: 1292 read iops (755% HDD)16 files, 1 thread each, seq 1M reads, simple:
10113 read iops (250% HDD)16 files, 1 thread each, rnd 16k reads, posixaio: 393889 read iops
(83806% HDD)16 files, 16 threads each, rnd 16k reads, posixaio: 872489 read iops (175905% HDD)
=== native aio series ===1 file, 16 threads, rnd 16k writes, native aio: 401 write iops (65%
HDD)
16 files, 16 threads each, rnd 16k writes, native aio: 5115 write iops (1020% HDD)1 file, 16
threads, rnd 16k reads, native aio: 1412 read iops (1077% HDD)16 files, 16 threads each, rnd
16k reads, native aio: 900873 read iops (182362% HDD)Tests complete on mwanafunzi @ 2020-10-
```

01 12:04:34.

Files remain. To clean up, add argument "cleanup".



# User map

User map

# workbench container

- Currently residing on imara
- Need to shift

2020

2020

# January 2020

## January 4th

I received the dell precision rack I bought on ebay. The performance is quite good: on linpack the installed e5-2623v3's get 260ish Gflops. I think, moving forward, I would like to augment this with more memory and more cores, especially if this becomes a datascience box and I repurpose the z620 for something else.

I also learned that this machine has full support for nvme boot but requires pcie to nvme adapters in order to do so. Dell sold this with a quad nvme drive called the Dell Ultra-Speed Drive Quad NVMe

## January 7th

I attempted to switch my e585 thinkpad from the nvme drive to a sata drive so that I could suspend again. However, this did not fix the suspend issues I'm having. Now, it occasionally gives me a white screen instead of a black screen but the end result is the same: it requires a hard reset using the power button. I removed the sata ssd and put the nvme back in because it's faster and also the only computer I have that can use the nvme drive at the moment. since both cause the same errors, I'm just sticking with what I had. I put the ssd in the precision rack computer, though it still has opensuse installed on it for now.

## January 8th

Currently, the disk in the small optiplex server is just floating in there which is a not so great

setup. I 3d printed a caddy to put the hard drive in but the way the cable lengths are set up, it wouldn't be quite long enough to reach the power on both the ssd and hard drive at the same time.

I need to flip the holes in the stl file so that the hard drive is in upside down from how it normally is. This will give me the slack I require in the sata power cable to feed both the ssd and hard drive at the same time.

## January 9th

iDRAC and proxmox are setup on the dell precision rack 7910 that I got on the 4th. It works quite well. I 3d printed caddies to include in there. One thing that is kind of strange is that there's a slightly rattly sound to one of the fans. Perhaps I'll have to find a replacement. I'm not sure which fan it is.

## January 10th

### Bookstack upgrade

Upgraded bookstack, apparently I hadn't upgraded it since the initial install in 2018. The upgrade appears to have gone without a hitch.

## Container migration onto moja

I am unable to migrate containers from mbili and tatu onto moja because local-lvm does not exist as a storage device there. I think I will take the 1tb ssd in that node and make it a lvm vg to enable movement of things onto there.

# Storage migration of jellyfin

Overnight, I ran a migration of jellyfin from lizardfs to zfs on tatu. lizardfs is great but because jellyfin only supports sqlite, which uses many small reads, I can't leverage my postgres server running on flash. The small reads for sqlite are the worst case scenario for lizardfs and as such the server runs very sluggish. After the migration from lizardfs to zfs, it's very snappy. However, this reduces my ability to migrate it in the future. lizardfs to it's credit has already balanced files away from tatu since lizardfs is sharing the zpool on that machine with jellyfin now.

## hot backup

I should probably setup hot backups with a limit of 1 that run every night and backup to lizardfs, this would allow quicker migration despite my diminishing use of lizardfs for actually hosting containers.

## I set this up on January 13th

it does backups every day except saturday since I have an existing backup job running every saturday. The shared storage created has a 1 backup limit so this should only hold a backup for the last day.

2020

# February 2020

<https://www.proxmox.com/en/training/video-tutorials/item/bond-configuration>

## Infiniband notes

- Infiniband is a good idea for multi-node gpu training as it can use RDMA. This allows exchanges between the gpus on each node without first copying the data from the gpu into system ram.
- QDR:
  - 40 gbps per port
  - 1 micro second latency
  - \$40 for a dual port card
- Bonding of infiniband interfaces is not possible?

## Purchases

- Bought 3 quad intel nics for \$35 on ebay. This should help with lizardfs bandwidth.
- Bought an m40 gpu with 24gb of vram.
  - Needs a fan and the 3d printed bracket for adding fans to the passive card.

2020

# March 2020

## March 4 2020

### Trying to get tesla m40 into z620

UEFI is required!! I converted the z620 machine (mwanafunzi) from legacy boot to uefi boot and switched the gpu from legacy to efi support. [Here](#) is the pastebin output from that.

This worked very well, with the tesla drivers, I got the m40 gpu to show up with all 24 GB of VRAM.

```
      .-/+00sssssoo+/- .
      `:+ssssssssssssssssss+:`
      -+ssssssssssssssssssyyssss+-
      .osssssssssssssssssdMMMNysssso.
      /ssssssssssshdmmNNmmyNMMMMhssssss/
      +ssssssssshmydMMMMMMMMNddddyssssssss+
      /ssssssssshNMMMyhhyyyhmNMMNhssssssss/
      .ssssssssdMMMNhssssssssshNMMMdssssssss.
      +sssshhhyNMMNysssssssssssyNMMMyssssssss+
      ossyNMMNyMMhssssssssssshmmhssssssso
      ossyNMMNyMMhssssssssssshmmhssssssso
      +sssshhhyNMMNysssssssssssyNMMMyssssssss+
      .ssssssssdMMMNhssssssssshNMMMdssssssss.
      /ssssssssshNMMMyhhyyyhdNMMNhssssssss/
      +sssssssssdmydMMMMMMMMddddyssssssss+
      /ssssssssshdmNNNNmyNMMNhssssss/
      .osssssssssssssssssdMMMNysssso.
      -+ssssssssssssssssssyyyssss+-
      `:+ssssssssssssssssss+:`
      .-/+00sssssoo+/- .

root@mwanafunzi
-----
OS: Ubuntu 18.04.4 LTS x86_64
Host: HP Z620 Workstation
Kernel: 5.3.0-40-generic
Uptime: 25 secs
Packages: 3166
Shell: bash 4.4.20
Terminal: run-parts
CPU: Intel Xeon E5-2650 0 (16) @ 2.800GHz
GPU: NVIDIA Tesla M40
GPU: NVIDIA GeForce GTX 1070
Memory: 716MiB / 48123MiB
```





I tested this out with allennlp and it worked quite well. The m40 trained an rnn about as fast as my 1070's. This is the worst case for this comparison because the 1070 has a higher clock rate but fewer cuda cores and rnn's are difficult to parallelize.

The additional vram allowed me to go up to 128 for the batch size.

However, my cooling solution for the m40 was insufficient. After about 1 epoch of training (5 minutes of heavy usage) the temp exceeded 80 C and I had to end the workload. I was using a single NF 4x20A fan from noctua. However, these only provide 5 CFM of airflow. I purchased a 2 pack of delta 40mm fans that achieve 10 CFM each. This should be enough airflow for operation. While the noise level is going up, it is only increasing from 17 dba to 35 dba (per the documentation for the respective products). since the z620 case fans are about 35 dba, this noise difference shouldn't be very noticable.

## AMD GPU build (RX580)

batch\_size 32 tensorflow 1.14

resnet50: 49.93 images per second

resnet 152: 20.83 images per second

inception v3: 20.03 images per second

2020

# April 2020

I really want to increase the size of the sdd in the z620 so that I can snapshot any time I do an apt upgrade for easy rollbacks.

I think everything necessary to remove the drive from mwanafunzi has been done: the virtual group has been deleted along with the logical volumes inside it.

The issue remaining is how to move all the user accounts over. I've moved one user account over before but movign all of them may cause conflicts in UIDs. I think this can be done by first adding the user accounts and then copying the `/etc/passwd` and `/etc/shadow` entries over, changing the UIDs to match the newly created ones.

I need to make sure to take out the 500GB HDD when I do the reinstall so that the installation process doesn't identify the existing boot partition and keep it on that drive. I want to get everything onto a single disk this time to reduce the likelihood of failure (since I'm not using RAID for the root partition)

2020

# May 2020

The hard drive in mbili is showing some read errors. I need to 3d print a new drive caddy for that computer.

Turns out I can use the caddy I printed before and jam it in there. Doesn't seem to be throwing errors anymore.

2020

# September 2020

Trying to build pytorch with hip/rocm support on ubuntu 20.04 and rocm 3.7.

```
CMake Error: The following variables are used in this project, but they are set to NOTFOUND.
Please set them or make sure they are set and tested correctly in the CMake files:
GLOO_HIP_HCC_LIBRARIES    linked by target "gloo_hip" in directory
/home/kenneth/build2/pytorch_rocm/third_party/gloo/gloo
PYTORCH_HIP_HCC_LIBRARIES linked by target "c10_hip" in directory
/home/kenneth/build2/pytorch_rocm/c10/hip    linked by target "caffe2_nvrtc" in directory
/home/kenneth/build2/pytorch_rocm/caffe2    linked by target "torch_hip" in directory
/home/kenneth/build2/pytorch_rocm/caffe2
ROCM_HIPRTC_LIB          linked by target "caffe2_nvrtc" in directory
/home/kenneth/build2/pytorch_rocm/caffe2    linked by target "torch_hip" in directory
/home/kenneth/build2/pytorch_rocm/caffe2
-- Configuring incomplete, errors occurred!See also
"/home/kenneth/build2/pytorch_rocm/build/CMakeFiles/CMakeOutput.log".See also
"/home/kenneth/build2/pytorch_rocm/build/CMakeFiles/CMakeError.log".Traceback (most recent
call last):
  File "setup.py", line 732, in <module>
    build_deps()
  File "setup.py", line 311, in build_deps
    build_caffe2(version=version, File
"/home/kenneth/build2/pytorch_rocm/tools/build_pytorch_libs.py", line 54, in build_caffe2
cmake.generate(version, File
"/home/kenneth/build2/pytorch_rocm/tools/setup_helpers/cmake.py", line 329, in generate
self.run(args, env=my_env) File
"/home/kenneth/build2/pytorch_rocm/tools/setup_helpers/cmake.py", line 141, in run
check_call(command, cwd=self.build_dir, env=env) File "/usr/lib/python3.8/subprocess.py",
line 364, in check_call
    raise CalledProcessError(retcode, cmd)
```

2018

2018

# September 2018

## September 9, 2018:

Previously, I had a problem with nextcloud where it wasn't able to be recognized. I went to a backup and it turned out that this was on the nextcloud side. It had nothing to do with my nginx reverse proxy which was where I assumed the issue initially was.

I went ahead and kept on the backup and then synced the local datastores on my laptop and desktop to bring the data up to date on the vm even if the database records were going to have a month of gap. However, my sharelatex docker container was also running on that vm. Unfortunately, I had done quite a bit of work in my sharelatex docker so I needed to get that back. The solution I adopted was to make a backup then revert to the backup I made before the above stuff went awry and then boot the container. After downloading the zipped project folders for all my projects, I shut down that vm again and restored to the first backup I made. I then uploaded the project folders to the sharelatex docker container again. I would like to get away from docker as it cannot be run in an lxc container easily. When I do run it I am getting errors like `oci runtime create failed debian`. [This thread](#) displays some of the modifications that can be made to proxmox's lxc config files to fix this. I modified the config file of my test container to have all but the last line. I suppose I should try with the last line and see what happens.

In addition, I modified the max number of backups on the 1tb western digital drive I have in its own raid 0. Containers in Proxmox will replace the older containers if the number of backups after the current backup job will exceed what you've configured. VMs do the opposite. which is dumb. They won't back-up any more once you've reached the limit. I'll have to remember to clean out old backups for my single vm.

## September 20 2018

I tried installing a matrix homeserver. However, this didn't go great. Will try again this weekend.

I restarted Mwenyeji today. I noticed that the fs.inotify changes I had made were no longer in affect. I started getting file handle errors again. To fix this, I ran the following.

```
root@mwenyeji:~# echo "fs.inotify.max_queued_events=48000"|sudo tee -a /etc/sysctl.conf
root@mwenyeji:~# echo "fs.inotify.max_user_instances=512"|sudotee -a /etc/sysctl.conf
root@mwenyeji:~# echo "fs.inotify.max_user_watches=120000"|sudo tee -a /etc/sysctl.conf
```

I setup a cronjob to take a snapshot of the wiki every night since I cannot figure out if there is any version controlling going on behind the scenes in bookstack. snapshots of the wiki appear to be very small so they don't take up much space having a lot of them.

2018

# November 2018

## ucs c240 m3

I purchased a cisco ucs c240 m3 on ebay for \$195. The unit came equipped with dual e5-2609's, 8gb of ram, heatsinks and four drive trays. Initially, the fans were too loud so I upgraded the firmware using the host upgrade utility.

This allowed me to bring the fans down to a very reasonable level. However, after running linpack on the machine, I noticed that it was rebooting after being under heavy load. I checked out the ram and everything but the same thing happened. None of the capacitors on the motherboard appear swollen or leaky.

I tried replacing the processors (since the unit wasn't enclosed in electrostatic wrap during shipping I figured that the processors may have experienced some static discharge rendering them unreliable.

I was getting errors in cimc like `PVCCP_P1: Processor 1 voltage is lower critical` before I changed the procs. I am still getting reboots after the server has been under heavy load. It's now not while the processors are stressed but actually after the process has been killed and the machine idles for a while.

Running geekbench is fine, running linpack is too much. I think I have no choice but to return this server, which is a bummer because I'm going to be out my shipping costs.

## Update

Upgrading the processor energy profile to high performance has prevented failures. The longest



I've been up has been 8 hours. I'm quite confident it can stay running for longer now. I wanted to work up to something longer since a system failure results in the fans hitting 100% which can be quite annoying in the middle of the night.

# December update

Unfortunately, the problems resurfaced after attempting to fill all the ram slots on processor 0. Stepping back down to only 4 dimms fixed the issue. I still suspect an issue with the powersupply. I have gotten \$60 from the seller due to the situation. I hope to buy a new psu for this machine using this money.

2018

# October 2018

## Quanta windmill board 2

I got another board but it appears to not be working that well. I also got a noctua fan controller which is working great. However, after writing the bios chip using the ch341a chip I bought off ebay, the sata connectors are not working. I am able to boot an operating system fine from the usb port, I just cannot access the sata connectors well.

The next course of action is updating the bios using the freedos updater. If the sata connection problem is not fixed after that, I may have to buy a pcie raid card to avoid using the onboard sata controller.

## December 10

Unfortunately, the situation only deteriorated with the bios chip. It lost all ability to post. I am afraid I will have to buy another windmill board from the seller.

# Car stuff

Car stuff

# Speakers

My head unit is a Pioneer AVH-X391BHS

it can output 50W per channel.

I think I will upgrade my blown 6.5" door speakers to these mtz terminators

[https://www.crutchfield.com/p\\_236TERM6S/MTX-Terminator6S.html](https://www.crutchfield.com/p_236TERM6S/MTX-Terminator6S.html).

These should work well with the existing component speakers. I will have to see how much work is required to use the tweeters from the terminators in the dash of my car. Currently, crutchfield is saying that some drilling may be required.

2021

2021

# January

## 1/15/2021

Upgraded the following machines from debian 9 to debian 10:

- postgres
- brat-anno

## 1/18/2021

When removing the head unit in my car, the head unit is attached to a carriage with screws. Those are the screws you see. When removing the head unit though, you don't have to worry about these screws: you just need to slide the carriage out. There are little tabs that need to be pushed in so that the whole thing can slide out. The top tabs are probably already pushed in, they're quite worn at this point.

2021

# LBA write rate on database host

Date	LBAs written
3/6/2021	8576402139
3/13/2021	8679613123
3/31/2021	967961811

.18 TB over 25 days. 160 TBW for the drive.

Should last another 2.5 years.

2021

# March

3/18/2021: The cooler on the 7900x needs to be shifted over so that it covers the IHS more completely before you run linpack etc. on it. Ram slots A1, A2, B1, B2 are the ones that work (e.g. all ram slots left of cpu socket).



2021

# April

An automatic update to cuda has broken my virtual environments. Creating new virtual environments doesn't seem to help. I get an error saying that all available cuda devices are busy when there are actually no devices in use. (e.g. nvidia-smi shows 0% gpu utilization, no vram occupied and no processes claiming the gpu).

In addition, snapper has non-sane defaults on ubuntu and thus my rollback was broken. After rolling back, the .snapshots folder was lost so I was unable to rollback again. I found a solution which involves writing something to your fstab for .snapshots. I copied the actual config from my opensuse tumbleweed install and it worked. However, no matter how far back I go with my snapshots, I still get the cuda device busy error. :(.

I booted another ubuntu os from my portable ssd with cuda installed on it and sure enough it works, I can train the model (and see both the gpus in nvidia-smi).

I am going to try and see if my m40 gpu is compatible with the cuda drivers on opensuse leap (Hopefully this is the case as I've purchased another m40 for cheap to put in my opensuse leap system). If, indeed it does work, I will be switching to opensuse leap for mwanafunzi. I know that snapper has good defaults and works well on tumbleweed and leap so hopefully this rollback issue won't be a problem in the future and if nvidia breaks my setup, I can just rewind. This was the original reason why I was using snapper but apparently I had it misconfigured.

2021

# December

## 12/10/2021

- Since we moved, everything had to be migrated to comcast's internet.
- All machines were down from 12/7 to 12/9
- Had to change ip for internal network on router from 10.0.0.X/32 to 192.168.X.X to match previous ip range
  - IP addresses for cluster nodes cannot change so we have to keep the internal network ips the same.
- I'm now using a tiny linode instance, using wireguard linked back the nginx reverse proxy
  - Largely followed [this guide](#), which requires some initial steps from [this guide](#).
  - Private and public wireguard keys generated using [this guide](#).
  - Seems to work well but jellyfin can't use large files since my upload speed on comcast is horrible.
- Modified google domains config to remove the dynamic dns entry and instead switched to a static A record.

2022

2022

# Cluster Buster plan

Goal: migrate from a cluster of commodity cast off office servers to a larger, more reliable server.

Rationale: software defined storage is fun and all but the throughput of lizardfs is very poor and while moosefs is faster, it is not nearly as reliable. High availability mode with shadow masters requires a paid subscription for moosefs while this was included in lizardfs. I'd like to have a more reliable single server and use that instead.

## Step by step:

1. Obtain quorum with current cluster
  - Shutdown mbili so that I can move it off of the cable for nne
  - Plug in nne and make sure it's working
  - Bring mbili back up
2. Migrate any running containers to tatu.
3. Shut down mbili
4. Install proxmox as stand-alone on mbili making a new single proxmox server (tani).
5. Add moosefs chunkserver and moosefs mount point packages onto new server. (tani).
6. Add moosefs disk as btrfs storage on tani.
7. Restore

# New Page

```
{
  "dataset_reader": {
    "type": "transformer_squad",
    "length_limit": 512,
    "transformer_model_name": "roberta-large"
  },
  "model": {
    "type": "transformer_qa",
    "transformer_model_name": "roberta-large"
  },
  "train_data_path": "https://allennlp.s3.amazonaws.com/datasets/squad/squad-train-v2.0.json",
  "validation_data_path": "https://allennlp.s3.amazonaws.com/datasets/squad/squad-dev-v2.0.json",
  "trainer": {
    "callbacks": [
      "tensorboard"
    ],
    "grad_clipping": 1,
    "learning_rate_scheduler": {
      "type": "slanted_triangular",
      "cut_frac": 0.1,
      "num_epochs": 5
    },
    "num_epochs": 5,
    "num_gradient_accumulation_steps": 2,
    "optimizer": {
      "type": "huggingface_adamw",
      "eps": 1e-08,
      "lr": 2e-05,
      "parameter_groups": [
        [
          "bias",
          "LayerNorm\\.weight",
          "layer_norm\\.weight"
        ]
      ]
    }
  }
}
```

```
        {
            "weight_decay": 0
        }
    ]
],
    "weight_decay": 0
},
    "validation_metric": "+per_instance_f1"
},
    "vocabulary": {
        "type": "empty"
    },
    "data_loader": {
        "batch_sampler": {
            "type": "bucket",
            "batch_size": 8
        }
    },
    "numpy_seed": 100,
    "pytorch_seed": 100,
    "random_seed": 100
}
```

2024

2024

# Desktop shutting down on its own

Currently my desktop is shutting down when I run the tests I've written for supar. It's crashing on the `test_parse.py` test.