

# ATLAS

## Cloud Native Deployments of Bare-Metal High-Performance AI Workflows



George | Hao | Jing | Shawn | Shubham

MENTOR Chris | Mike | Ravi

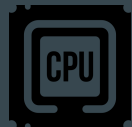
# Outline

- Quick Recap
- Architecture
- Last Sprint Progress
- Current Sprint & Burndown
- Performance Metrics
- OpenShift - BuildConfig
- Deploy CPU-only Image on OpenShift
- Reflection (*What we learned and what we didn't do well so far*)

# Quick Recap



IBM



Power9 cpu



NVidia  
V100 32GB



High-speed  
network

## Students

### Brainstorming energy-saving hacks on Satori, MIT's new supercomputer

Three-day hackathon explores methods for making artificial intelligence faster and more sustainable.

Kim Martineau | MIT Quest for Intelligence  
February 11, 2020

▼ Press Inquiries

RELATED

## Admins



## Admins & Researchers

MIT IBM



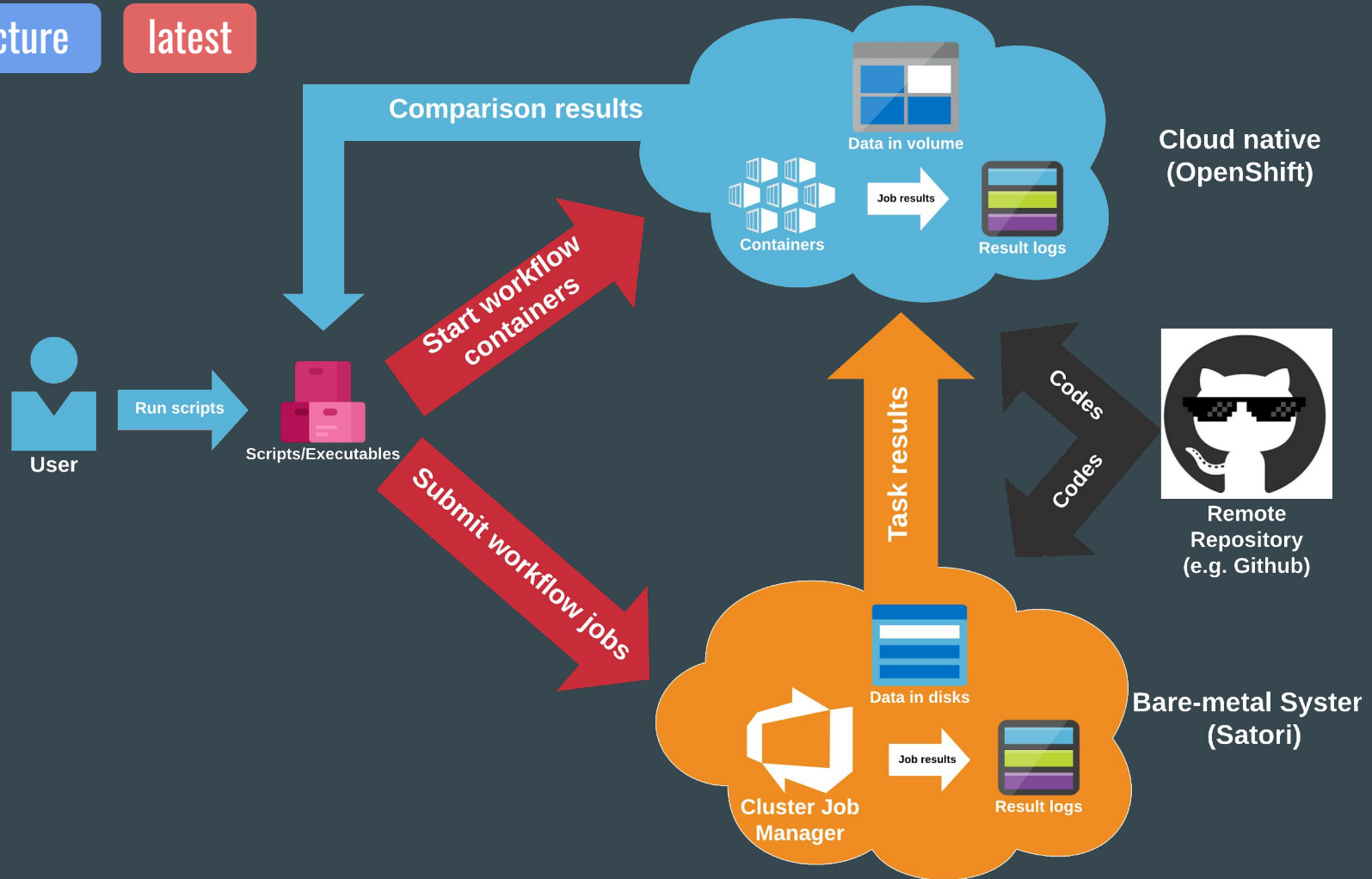
Bare-metal env.



OPENSIFT

Cloud env.

Architecture latest



# Last Sprint Progress



# Last Sprint Progress

- Environment setup and logistics
  - X86-based OpenShift cluster
  - **Blocked** PowerPC-based OpenShift cluster
- Successfully ran the workflow in bare-metal env (Satori).
- Successfully containerized the Big-GAN workflow in local env.
  - Generated Dockerfile
  - Cpu-only, non-gpu

# Current Sprint & Burndown

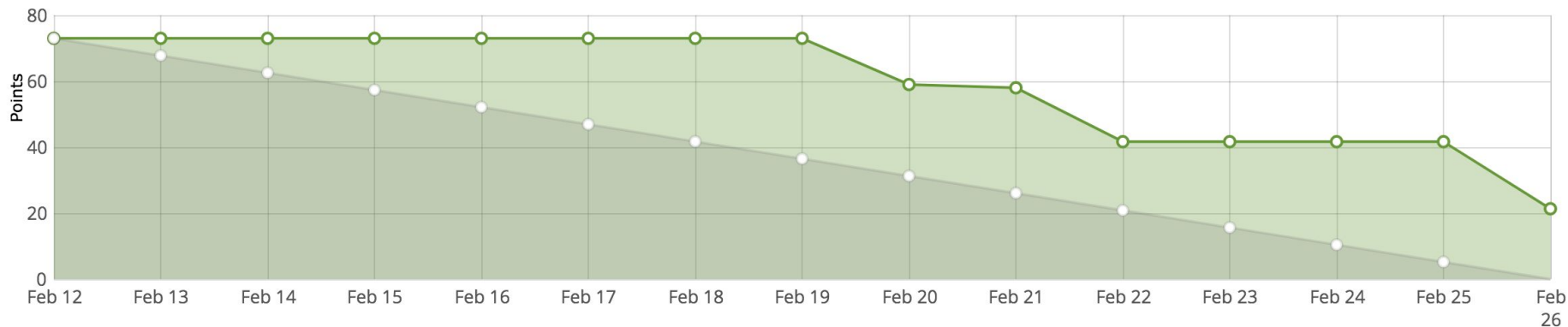
# Sprint (2/12 ~ 2/26)

- Survey Performance metrics
- Implement script to measure Performance on Satori (In progress)
- Get familiar with OpenShift
- Run BigGAN with CPU on OpenShift
- Run BigGAN with GPU on OpenShift Were blocked until a week ago

# Burndown Chart

SPRINT-2 CLOUD-NATIVE-DEPLOYMENTS-BARE-ME... 12 FEB 2020-26 FEB 2020

 62% ✓ 73 total points 45 completed points | 4 open tasks 13 closed tasks ⇄ |  0 iocaine doses



# Performance Metric

# Performance Metrics

- CPU usage
- GPU usage
- Cost
- Time to completion



Fig: Comparison between bare metal and cloud

# Performance Metrics: How do we measure?

- Linux utilities like 'top' give real time CPU stats, 'PBench' on OpenShift
- nvidia-smi - NVIDIA System Management Interface program (GPU stats)
- Script these utilities, to check usage periodically

```
top - 21:45:45 up 47 days, 2:37, 31 users, load average: 3.56, 3.57, 6.11
Tasks: 2489 total, 3 running, 1696 sleeping, 4 stopped, 3 zombie
%Cpu(s): 0.1 us, 0.2 sy, 1.3 ni, 98.5 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 59802259+total, 32042950+free, 76771072 used, 20082201+buff/cache
KiB Swap: 33554368 total, 33554368 free, 0 used. 36482566+avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
58590	evand	39	19	5824	3008	1920	R	100.0	0.0	170:56.73	john
89545	evand	39	19	5824	2944	1920	R	100.0	0.0	132:35.48	john
63881	root	0	-20	70.2g	20.6g	3.7g	S	13.8	3.6	3540:37	mmfsd
110515	addanki	20	0	125888	15744	4736	S	7.5	0.0	3204:46	top
100192	cclin	20	0	113536	4480	3008	D	3.3	0.0	15:47.50	tar
115432	haft	20	0	0	0	0	Z	3.3	0.0	0:00.10	nvidia-smi
115436	haft	20	0	0	0	0	Z	3.3	0.0	0:00.10	nvidia-smi
115345	aroras	20	0	121472	11136	4608	R	1.6	0.0	0:00.48	top
154282	haft	20	0	1250496	73472	22464	S	1.6	0.0	155:34.64	python
40703	zhwu	20	0	118272	8704	4672	S	0.7	0.0	18:47.72	watch
48003	cclin	20	0	10496	9408	3072	S	0.7	0.0	4:58.72	tmux
57303	branlesh	20	0	90752	7616	3520	S	0.7	0.0	302:47.81	nvidia-smi
88006	cnh	20	0	90752	7616	3520	S	0.7	0.0	305:45.71	nvidia-smi
130271	cheungb	20	0	8095360	5.3g	137472	S	0.7	0.9	109:32.85	tensorboard
16870	haft	20	0	1333760	77376	24512	S	0.3	0.0	0:28.16	python
35826	haft	20	0	1333952	77568	24512	S	0.3	0.0	0:29.34	python
42034	branlesh	20	0	48.5g	6.1g	630208	S	0.3	1.1	8:09.10	python
46202	root	20	0	397056	20736	5312	S	0.3	0.0	151:25.36	wazuh-modulesd
52394	addanki	20	0	118144	8512	4672	S	0.3	0.0	61:46.43	watch
52453	addanki	20	0	118080	8512	4672	S	0.3	0.0	27:57.86	watch
63643	root	20	0	0	0	0	S	0.3	0.0	21:07.12	gpfswapdKproc
109317	root	20	0	0	0	0	I	0.3	0.0	0:00.52	kworker/u321:2
109503	root	20	0	0	0	0	I	0.3	0.0	0:00.10	kworker/60:1

Fig: 'top' command to show process usage of computers resources

Computer	Storage	Network	Staff	Electricity
Cooling	Facility costs	Security	Time (!)	Resilience
Measurement & Reporting	Financing effects	Insurance	Irrecoverable taxes	Procurement project
Application porting	Software licenses	User training & support	Documentation	Commissioning
Warranty, support, maintenance	Decommissioning	Availability / throughput	Data migration	... and more ...

- Total Cost of Ownership (TCO):
  - Costs owning/operating HPC
  - Important to HPC owners/managers
  - Always costs *something*
- Total Cost of Solution (TCS):
  - Costs incurred using HPC service for project
  - Important to HPC users
  - May cost *nothing*

Fig: Common Costs to Measure in an HPC



```
(wmlce-1.6.2) [haozuo@service0001 ~]$ nvidia-smi  
Tue Feb 25 21:39:57 2020
```

NVIDIA-SMI 418.87.00      Driver Version: 418.87.00      CUDA Version: 10.1									
GPU	Name	Persistence-M		Bus-Id	Disp. A	Volatile	Uncorr.	ECC	
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage		GPU-Util	Compute	M.	
0	Tesla	V100-SXM2...	On	00000004:04:00.0	Off				0
N/A	40C	P0	40W / 300W	10MiB / 32480MiB		0%	E.	Process	
1	Tesla	V100-SXM2...	On	00000035:03:00.0	Off				0
N/A	42C	P0	53W / 300W	31311MiB / 32480MiB		0%	E.	Process	
Processes:									
GPU	PID	Type	Process name				GPU Memory Usage		
1	42034	C	python				31301MiB		

Requesting exclusive node

Nodes are not fully utilized

Universal baseline (MIPS, CPU, GPU)

Deploy AI workflows with

**BuildConfig**

**ON**

**OpenShift**

# WHAT IS

# BuildConfig



# BuildConfig

# AUTOMATIC

IMAGE

BUILD

DEPLOY

UPDATE

## YAML File

**BuildConfig**

**Similar**

**Docker  
Compose**

**ONLY for**

**OpenShift**

**CAN DO MORE**

# BuildConfig

Inline  
Dockerfile

Git Repo

Dockerfile

Image

Dockerfile

Binary

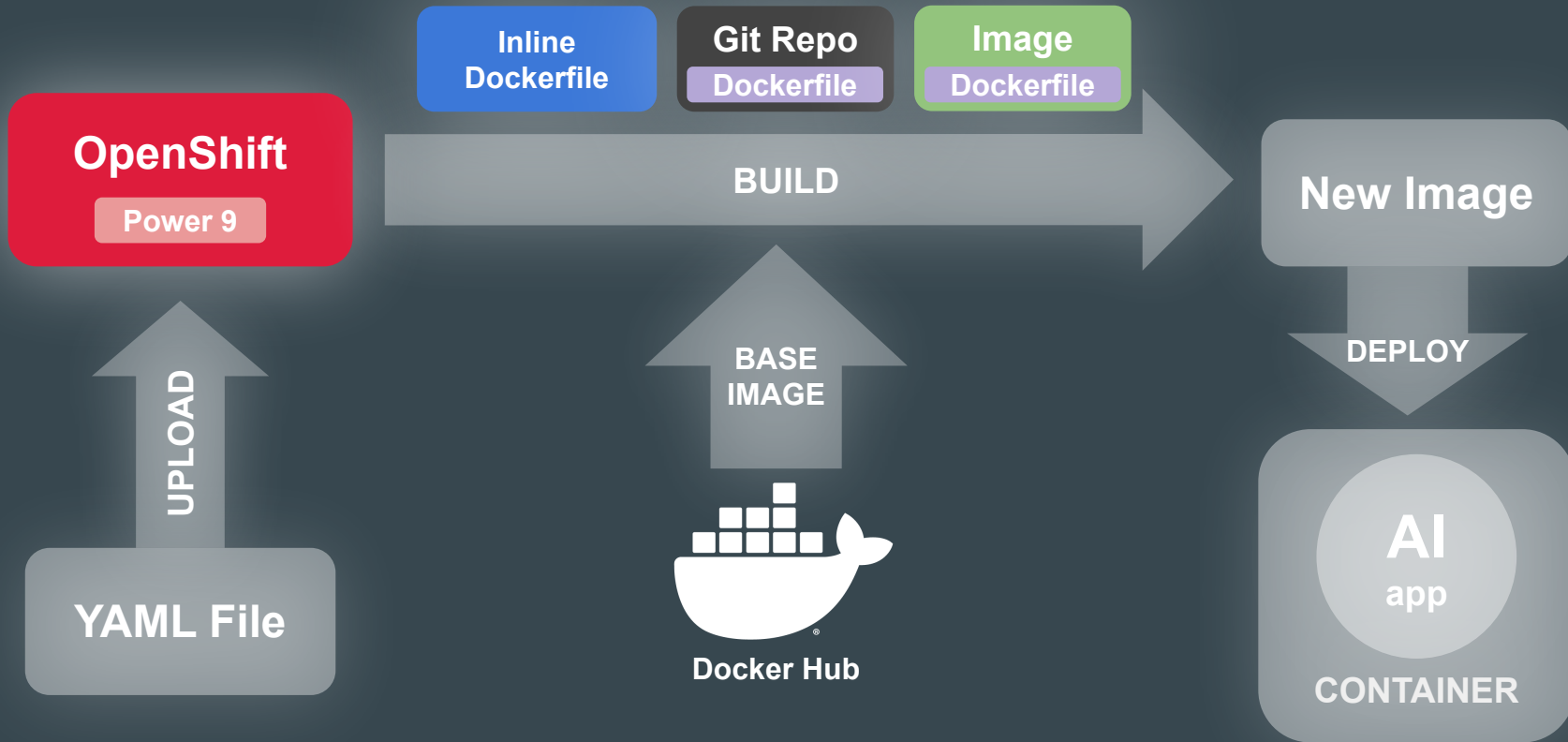
Dockerfile

## VARIOUS SOURCE SUPPORT

Normally want sources contain a **Dockerfile**

# BuildConfig

# Workflow



# Deploy CPU-only Image on OpenShift



# Deploy CPU-only Image on OpenShift

- Wrote Dockerfile that builds the image

```
1 FROM centos:centos7
2
3 # Install yum dependencies
4 RUN yum -y update; yum clean all
5 RUN yum -y install epel-release gcc openssl-devel bzip2-devel git python3
6
7 # Add python requirements & install
8 # Utilize docker cache to prevent pip installing every time
9 ADD ./requirements.txt /python/requirements.txt
10 RUN pip3 install --upgrade pip && pip install -r /python/requirements.txt
11
12 # Add working directory
13 ADD ./ /BigGAN
14 WORKDIR /BigGAN
15
16 CMD cd /BigGAN
17 CMD /bin/sh run.sh
```


- Pushed image to team DockerHub

 atlas4openshift / big-gan-cpu

BigGAN workflow that only runs on single node cpu 

# Deploy CPU-only Image on OpenShift

Pods > big-gan-cpu-latest-1-dtvr2

big-gan-cpu-latest-1-dtvr2  created 2 minutes ago Actions ▾

[app](#) [big-gan-cpu-latest](#) [deployment](#) [big-gan-cpu-latest-1](#) [deploymentconfig](#) [big-gan-cpu-latest](#)

[Details](#) [Environment](#) [Logs](#) [Terminal](#) [Events](#)

ⓘ When you navigate away from this pod, any open terminal connections will be closed. This will kill any foreground processes you started from the terminal.  
[Open Fullscreen Terminal](#)

Container: big-gan-cpu-latest

```
top - 18:55:20 up 26 days, 15:41, 0 users, load average: 4.99, 2.18, 1.58
Tasks:  4 total,  2 running,  2 sleeping,  0 stopped,  0 zombie
%Cpu(s): 34.4 us,  5.5 sy,  0.0 ni, 60.0 id,  0.1 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 46196832 total,  6759760 free, 3813348 used, 35623724 buff/cache
KiB Swap:   0 total,   0 free,   0 used. 41880824 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM    TIME+  COMMAND
  50 1001090+  20   0 3088796 471916 61396 R 200.0  1.0   2:06.70 python3 /BigGAN/work//BigGAN-PyTorch/main.py --model biggan_deep --d
 707 1001090+  20   0  56180   1940  1436 R   0.8  0.0   0:00.01 top
 696 1001090+  20   0  11824   1660  1296 S   0.0  0.0   0:00.00 /bin/sh
   1 1001090+  20   0  11692   1456  1212 S   0.0  0.0   0:00.03 /bin/sh run.sh
```

```
sh-4.2$ cat /proc/cpuinfo | grep name
model name      : Intel(R) Xeon(R) CPU E5-2660 v2 @ 2.20GHz
model name      : Intel(R) Xeon(R) CPU E5-2660 v2 @ 2.20GHz
model name      : Intel(R) Xeon(R) CPU E5-2660 v2 @ 2.20GHz
model name      : Intel(R) Xeon(R) CPU E5-2660 v2 @ 2.20GHz
model name      : Intel(R) Xeon(R) CPU E5-2660 v2 @ 2.20GHz
model name      : Intel(R) Xeon(R) CPU E5-2660 v2 @ 2.20GHz
model name      : Intel(R) Xeon(R) CPU E5-2660 v2 @ 2.20GHz
model name      : Intel(R) Xeon(R) CPU E5-2660 v2 @ 2.20GHz
model name      : Intel(R) Xeon(R) CPU E5-2660 v2 @ 2.20GHz
model name      : Intel(R) Xeon(R) CPU E5-2660 v2 @ 2.20GHz
```

^ Cpu info for k-openshift platform

^ Workflow runs successfully on k-openshift (x86) platform

# Reflection

# Reflection

- What we've learned in this sprint:
  - More about OpenShift (BuildConfig, Containerization ..etc)
  - Resource profiling tools on Linux system (pbench, top, nvidia-smi ..etc)
  - Backfill scheduling concept
- What can be improved:
  - Estimation/Overstretching (Assigned too much tasks)
  - Practice to break down task into smaller granularity
  - General cloud infrastructure knowledge
  - OpenShift documentation.... :((((

**Demo (if time allows)**