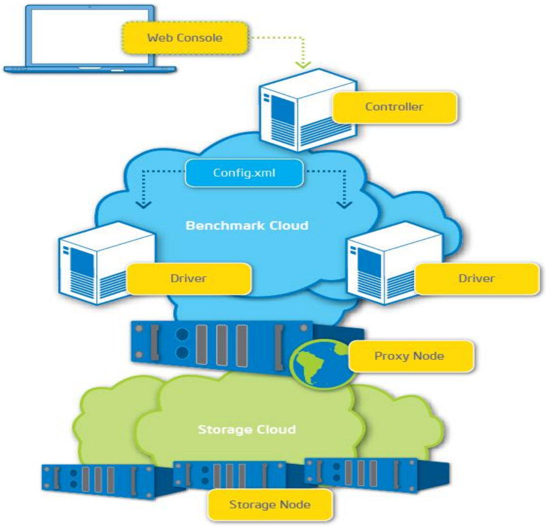
1 Introduction

COSBench consists of two key components:

- Driver (also referred to as COSBench Driver or Load Generator):
  - Responsible for workload generation, issuing operations to target cloud object storage.
- Controller (also referred to as COSBench Controller):
  - Responsible for coordinating drivers to collectively execute a workload.
- The controller and driver can be deployed on the same node or different nodes, and the node can be a physical machine or virtual machine (VM) instance.



1.1 Install Prerequisites

The current release of COSBench features Ubuntu* 12.04.1 LTS Desktop, and requires a few additional packages or settings:

- Java* Runtime Environment 1.6 or later/Curl 7.22.0 or later/Csvtool if processing generated csv files is required.
- Free TCP port (ensure these ports are accessible non-locally):
  - On COSBench controller machine: 19088
  - On COSBench driver machines: 18088

NOTE: Throughout this document, command line is **bolded** and *italicized;* yellow text is used for emphasis, to draw attention to specific information.

```
cosbench@cosbox:~$ sudo apt-get update
cosbench@cosbox:~$ sudo apt-get install openjdk-7-jre curl
```

1.2 Install COSBench

1.2.1 Download & Unpack

The COSBench controller and driver share the same installation package (e.g., 0.4.2.c2.zip), which can be obtained from https://github.com/intel-cloud/cosbench/releases .

```
cosbench@cosbox:~$ unzip 0.4.2.c2.zip; ln –s 0.4.2.c2/ cos
cosbench@cosbox:~$ cd cos; chmod +x *.sh
```

1.2.2 Command lines

There are a few command line tools under COSBench root folder:

Script	Description
start-all.sh/stop-all.sh	Start/stop both controller and driver on current node
start-controller.sh/stop-controller.sh	Start/stop controller only on current node
start-driver.sh/stop-driver.sh	Start/stop driver only on current node
cli.sh (supports below commands)	Manipulate workload through command line
→submit	Submit workload
→info	Check workload execution status
→cancel	Cancel ongoing workload execution

Note: To avoid HTTP requests routing, you need to **bypass** the proxy setting:

```
cosbench@cosbox:~$ unset http_proxy
```

1.3 Deploy COSBench on one node

Start up the COSBench driver and controller on the current node:

```
cosbench@cosbox:~$ ./start-all.sh
```

1.4 Deploy COSBench on multiple nodes

- Copy <version>.zip to the remaining COSBench nodes by means such as scp or shared folder.
- Repeat the procedures listed above for installing COSBench and verifying the installation on each node.
- For multiple nodes deployment, there will be one controller and a few drivers. Assuming two drivers will be deployed, the procedure looks like:
  - Start drivers on driver nodes:
    - `cosbench@driver1:~$ ./start-driver.sh`

```
cosbench@driver2:~$ ./start-driver.sh
```

- on controller node, modify conf/controller.conf to add all driver nodes as following:

```
drivers=2
…
[driver1]
name=driver1
url=http://192.168.10.1:18088/driver

[driver2]
name=driver2
url=http://192.168.10.2:18088/driver
```

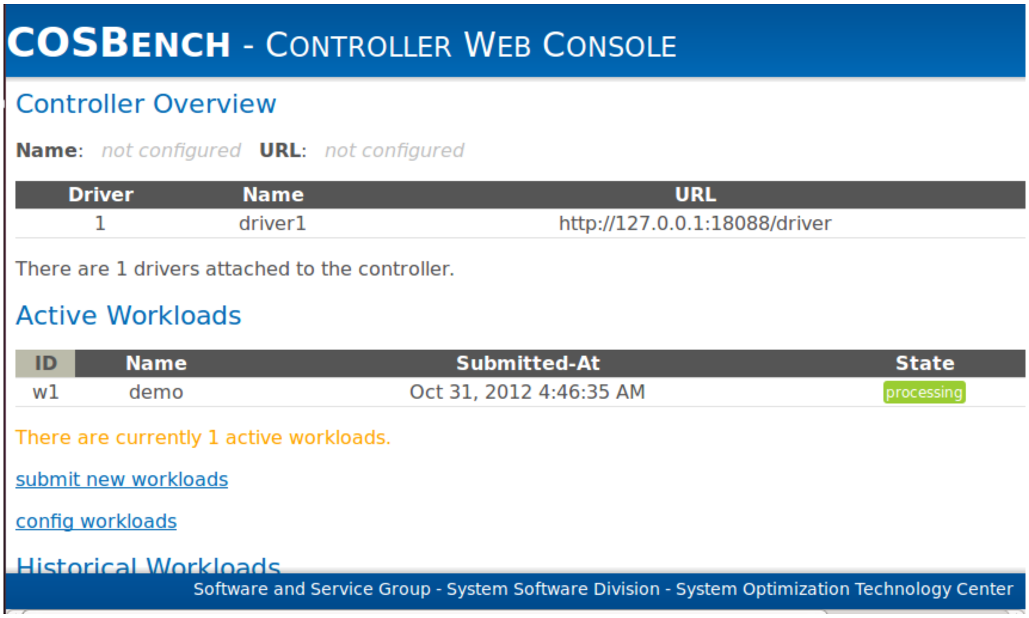
- Start controller on controller node:

```
cosbench@controller:~$ ./start-controller.sh
```

1.5 Verify the deployment

```
Cosbench@cosbox:~$ ./cli.sh submit conf/workload-config.xml # run mock test.
cosbench@cosbox:~$ ./cli.sh info # check workload status
```

Open http://<controller ip>:19088/controller/index.html in a browser to monitor status:



COSBENCH - CONTROLLER WEB CONSOLE

Controller Overview

Name: *not configured* **URL**: *not configured*

Driver	Name	URL
1	driver1	http://127.0.0.1:18088/driver

There are 1 drivers attached to the controller.

Active Workloads

ID	Name	Submitted-At	State
w1	demo	Oct 31, 2012 4:46:35 AM	processing

There are currently 1 active workloads.

submit new workloads

config workloads

Historical Workloads

Software and Service Group - System Software Division - System Optimization Technology Center