

Smart Deployer

Autonomic Deployment and Configuration of Cassandra
Through Microservice

Presented by: Xuan Cao , Zhipeng Chang, Nick Anic, Yuan Feng

Overview

1. What is the Smart Deployer?
2. Technologies
3. Purpose and Value
4. High Level Design
5. Screencast Demo
6. Challenges
7. Lessons Learned
8. Poster

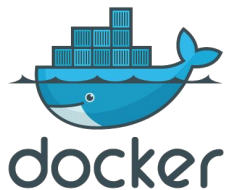


What is the Smart Deployer?

- ❖ The Smart Deployer strives to facilitate the installation and configuration of Cassandra to manage big data.
- ❖ The ultimate goal is to deploy Cassandra with acceptable performance in an autonomic manner.
- ❖ This way users can avoid spending time and energy on the deployment and configuration of the datastore and instead focus on real and core business problems.

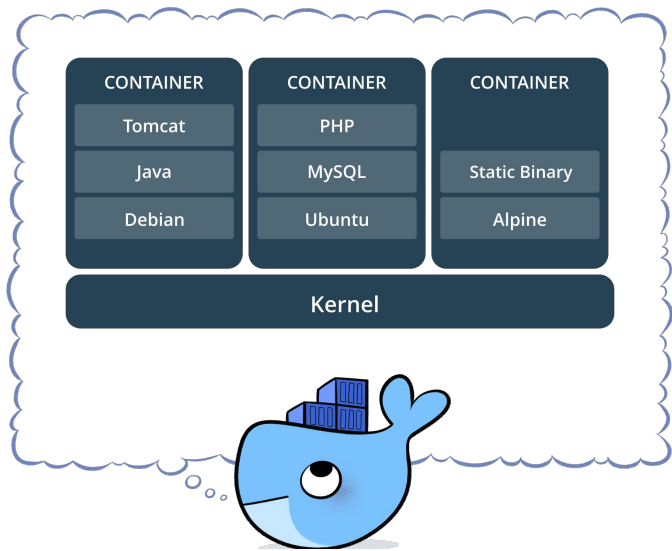


Technologies

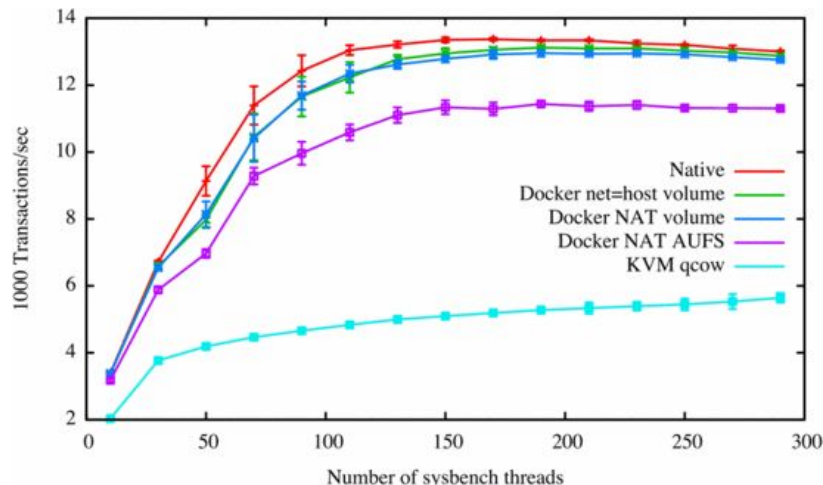


Docker

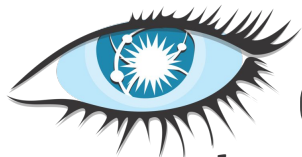
- ❖ Docker is an ecosystem that facilitates building and deployment applications using kernel-level virtualization (i.e. containerization). It enables microservices for many platforms, applications and scenarios.



Docker VS Virtual Machine



Wes Felter, et al. (2015) An Updated Performance Comparison of Virtual Machines and Linux Containers. [10.1109/ISPASS.2015.7095802](https://arxiv.org/abs/1509.07095)



cassandra

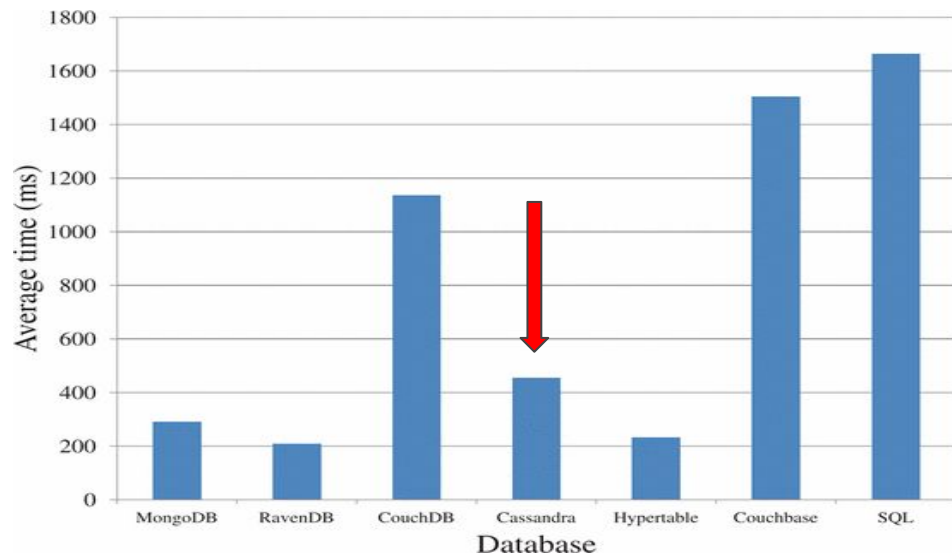
Cassandra

- ❖ Apache Cassandra is a free and open-source distributed NoSQL database management system designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure.
- ❖ In our project:
 - User wants to deploy a distributed Cassandra cluster (i.e NoSQL database) on a set of virtual machines



SQL VS NoSQL

1. SQL: MySQL, SQLite, PostgreSQL, etc.
2. NoSQL: BigTable, Cassandra, Hbase, MongoDB, CouchDB, etc.



Database	Number of operations					
	10	50	100	1000	10000	100000
MongoDB	8	14	23	138	1085	10201
RavenDB	140	351	539	4730	47459	426505
CouchDB	23	101	196	1819	19508	176098
Cassandra	115	230	354	2385	19758	228096
Hypertable	60	83	103	420	3427	63036
Couchbase	15	22	23	86	811	7244
MS SQL Express	13	23	46	277	1968	17214

Yishan Li and Sathiamoorthy Manoharan. (2013) A performance comparison of SQL and NoSQL databases.10.1109/PACRIM.2013.6625441



- ❖ The Yahoo! Cloud Serving Benchmark (YCSB) is an open-source specification and program suite for evaluating retrieval and maintenance capabilities of computer programs. It is often used to compare relative performance of [NoSQL](#) database management systems.

django Django

- ❖ Django is a free and open-source web framework, written in Python, which follows the model-view-template architectural pattern. It is maintained by the Django Software Foundation, an independent organization established as a 501 non-profit.
- ❖ In our project:
 - We use Django to deploy a web application as a visualization tool for the Smart Deployer



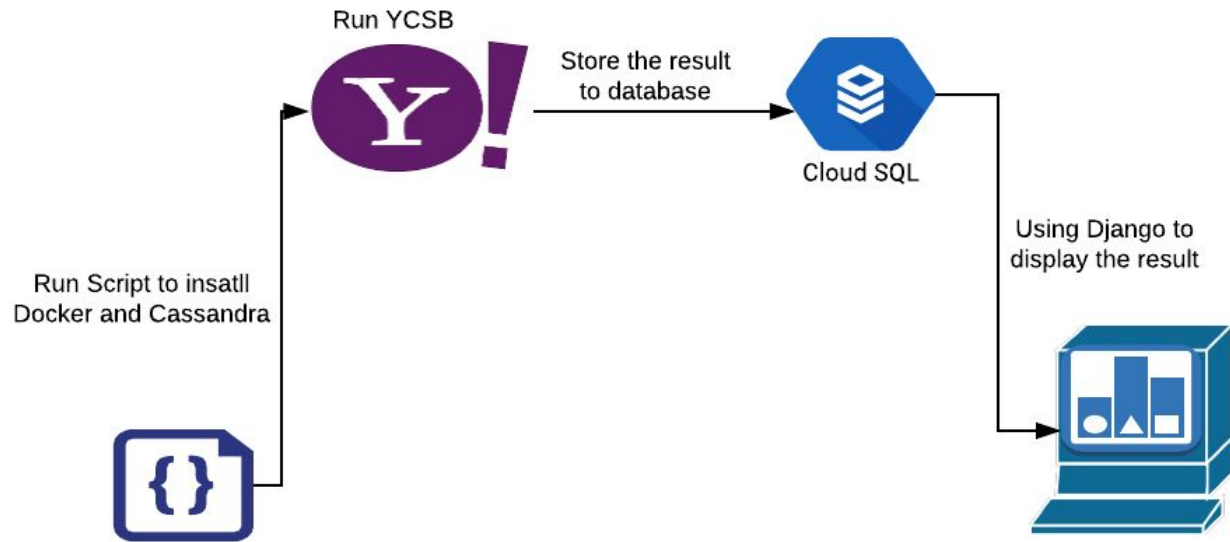
Purpose and Value

- ❖ Currently the default configuration for Cassandra as a NoSQL database is not very good
- ❖ Many users today simply deploy the default configurations regardless of their given architecture and resources which leads to suboptimal performance
- ❖ Big data is rapidly produced and in great amounts so managing it as efficiently as possible becomes very important
- ❖ By looking to optimize the configurations and automate the installation process this can save people a lot of time and money

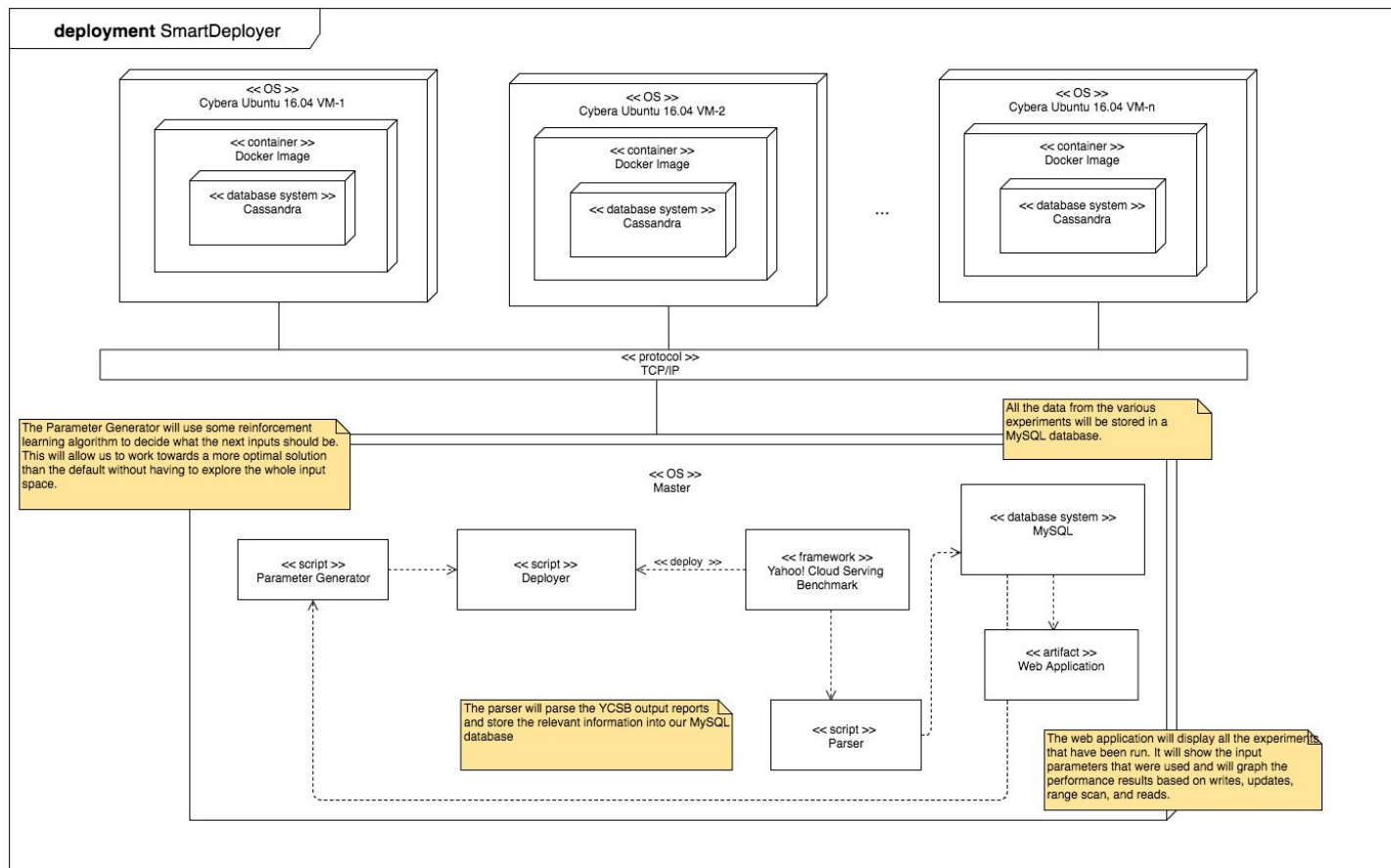


High Level Design

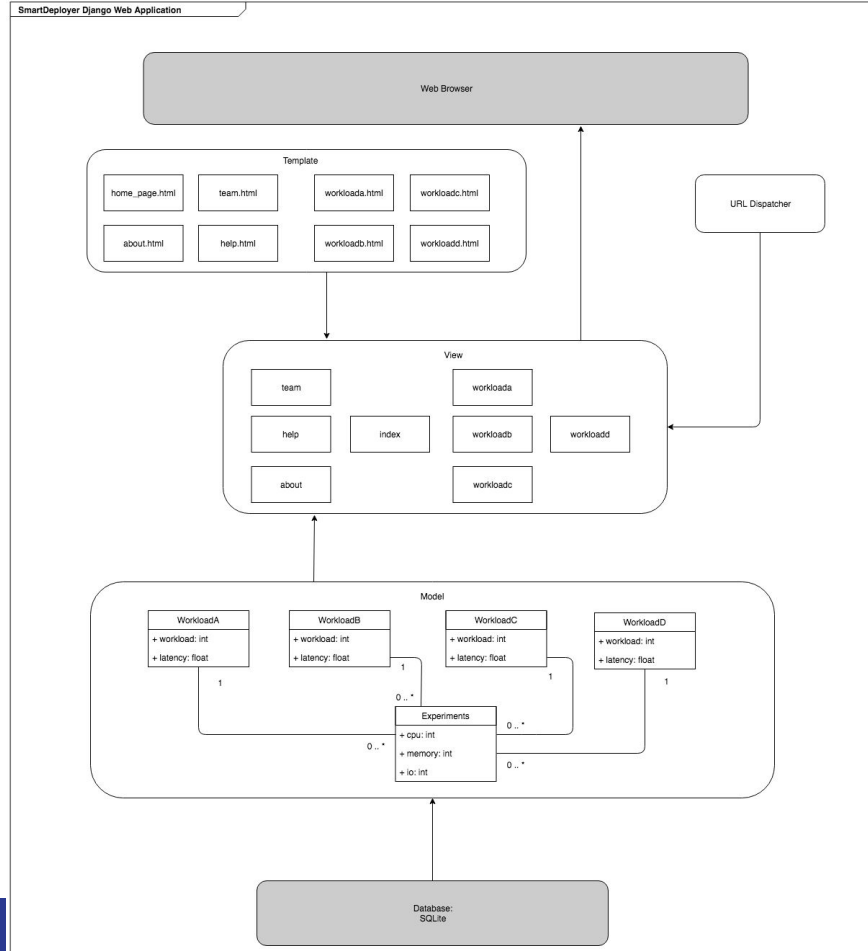
Overall workflow



Smart Deployer Architecture



Web Application Architecture



Screencast Demo

<https://www.youtube.com/watch?v=JVLsw8kwjRA&feature=youtu.be>

UNINSTALLATION DONE

[ubuntu@web-1:~/smartdeployer/Deployer/automatic installation and uninstallation\$ sudo docker

sudo: docker: command not found

[ubuntu@web-1:~/smartdeployer/Deployer/automatic installation and uninstallation\$./smartdeployerInstallation.sh smartdeployer install

If you don't follow instruction, the script will mess up your computer. Understood?yes

Hit:2 http://security.ubuntu.com/ubuntu xenial-security InRelease

Get:3 https://apt.dockerproject.org/repo ubuntu-trusty InRelease [48.7 kB]

Hit:4 http://nova.clouds.archive.ubuntu.com/ubuntu xenial InRelease

Get:1 http://dl.bintray.com/apache/cassandra 311x InRelease [3,169 B]

Hit:5 http://nova.clouds.archive.ubuntu.com/ubuntu xenial-updates InRelease

0% [4 InRelease gpgv 247 kB]


```
7dd1392a6ce5: Pull complete
43faf38e08a8: Pull complete
448a3a082d57: Pull complete
9b6d04438848: Pull complete
9c80ddd1c0b0: Pull complete
866a5caf213d: Pull complete
4699d5d33a86: Pull complete
88ac2463a580: Pull complete
cd612f57051a: Pull complete
85cdf7cbee96: Pull complete
dc555633391c: Pull complete
283115f0dbd3: Pull complete
Digest: sha256:c8886b7dca3a0453b8da89826b170ec7707211de35b96465f1c5b4a7bde0650e
Status: Downloaded newer image for webscam/cassandra:3.10
ssh -i ~/.ssh/smartdeployer ubuntu@10.1.9.116 "sudo docker pull webscam/cassandra:3.10" | ssh -i ~/.ssh/smartdeployer ubuntu@10.1.9.122 "sudo docker pull webscam/cassandra:3.10"
This node joined a swarm as a worker.
ssh -i ~/.ssh/smartdeployer ubuntu@10.1.9.116 "sudo docker swarm join --token SWMTKN-1-29idjohqc234hemyi9tuimn29mtfyuh5i72lt6emmw83aoatsm-0h9mfnzme3jxg5pl55uyxbvss 10.1.9.121:2377" | ssh -i ~/.ssh/smartd
eployer ubuntu@10.1.9.122 "sudo docker swarm join --token SWMTKN-1-29idjohqc234hemyi9tuimn29mtfyuh5i72lt6emmw83aoatsm-0h9mfnzme3jxg5pl55uyxbvss 10.1.9.121:2377"
swmgok1r17041ikf471xckj49
installation done
ubuntu@web-1:~/smartdeployer/Deployer/automatic installation and uninstallation$ sudo docker node ls
ID                HOSTNAME                STATUS                AVAILABILITY                MANAGER STATUS                ENGINE VERSION
kh6x087f187w26c3igrhxyuth    testenvironment        Ready                Active                       Leader                        18.03.0-ce
8fpurhwm1p6tqv1hd117ifg17 *   web-1                  Ready                Active                       Leader                        18.03.0-ce
6obakxedp8kfzgzl6kyotoo7r     web-2                  Ready                Active                       Leader                        18.03.0-ce
ubuntu@web-1:~/smartdeployer/Deployer/automatic installation and uninstallation$
```

ubuntu@web-1:~/smartdeployer/Deployer/automatic installation and uninstallation\$ sudo docker network ls

NETWORK ID	NAME	DRIVER	SCOPE
9df5f73c9b00	bridge	bridge	local
swmgok1r1704	cassandra-net	overlay	swarm
bc863c381886	docker_gwbridge	bridge	local
fd065af86e72	host	host	local
w3arg4589pd3	ingress	overlay	swarm
e0baddfb03b2	none	null	local

ubuntu@web-1:~/smartdeployer/Deployer/automatic installation and uninstallation\$ sudo docker service create -d --name cassandra --network cassandra-net webscam/cassandra:3.10



```
ubuntu@web-1:~/smartdeployer/Deployer/automatic installation and uninstallation$ sudo docker exec -it fc7ce9aa4330 bash
root@fc7ce9aa4330:/# ls
all.tar  docker-entrypoint.sh  lib64          opt  sbin  usr
bin      etc                   media          proc  srv   var
boot     home                 mnt           root  sys
dev       lib                  node-repair-after-full-boot.sh  run  tmp
root@fc7ce9aa4330:/# tar xzvf all.tar
commitToDatabase.py
generateReport.py
getHostIp.py
parser.sh
root@fc7ce9aa4330:/# ./parser.sh -m 2.0 -c 2.0 -b 1000
```

```
mysql> select * from ycsb_report;
```

testID	workload	OVERALL_runtime	OVERALL_throughput	READ_AVG_Latency	SCAN_AVG_Latency	UPDATE_AVG_Latency	READ_MODIFY_WRITE_AVG_Latency	INSERT_AVG_Latency
1600m8.0_1000_None	a	5618.0	177.99928800284798	1778.472	NULL	1877.506	NULL	NULL
1600m8.0_1000_None	b	5517.0	181.257930034439	1748.7861635220127	NULL	1841.9130434782608	NULL	NULL
1600m8.0_1000_None	c	6203.0	161.21231662098984	2517.074	NULL	NULL	NULL	NULL
1600m8.0_1000_None	d	5704.0	175.3155680224404	1877.153039832285	NULL	NULL	NULL	2996.391304347826
1600m8.0_1000_None	f	6160.0	162.33766233766235	1574.691	NULL	1618.8774193548388	3133.7956989247314	NULL
1600m8.0_1000_None	e	7934.0	126.03982858583312	NULL	4089.0917721518986	NULL	NULL	4227.288461538462
8000m4.0_1000_None	a	9994.0	100.06003602161297	5736.813492063492	NULL	6100.883064516129	NULL	NULL
8000m4.0_1000_None	b	7574.0	132.03063110641668	3757.5026123301986	NULL	3799.2093023255816	NULL	NULL
8000m4.0_1000_None	c	7512.0	133.1203407880724	3647.287	NULL	NULL	NULL	NULL
8000m4.0_1000_None	d	7256.0	137.81697905181917	3361.3093980992608	NULL	NULL	NULL	5267.226415094339
8000m4.0_1000_None	f	8877.0	112.65067027148811	3452.05	NULL	3142.7563850687625	6233.646365422397	NULL
8000m4.0_1000_None	e	16217.0	61.663686255164336	NULL	12566.39221871714	NULL	NULL	6458.734693877551
4000m2.0_1000_None	a	12741.0	78.48677497841614	8656.896624472574	NULL	7394.157794676806	NULL	NULL
4000m2.0_1000_None	b	12522.0	79.85944737262417	7870.3824786324785	NULL	6729.1875	NULL	NULL
4000m2.0_1000_None	c	11656.0	85.79272477693891	6477.452	NULL	NULL	NULL	NULL
4000m2.0_1000_None	d	9841.0	101.615689462453	5281.322410147992	NULL	NULL	NULL	7040.222222222223
4000m2.0_1000_None	f	12908.0	77.47133560582584	5566.505	NULL	5833.9152542372885	11127.56779661017	NULL
4000m2.0_1000_None	e	34833.0	28.70840869290615	NULL	31205.325556733827	NULL	NULL	12208.21052631579
10m1.1_123_None	a	7520.0	132.9787234042553	4437.360515021459	NULL	3426.3426966292136	NULL	NULL
10m1.1_123_None	b	8755.0	114.22044545973729	4866.248434237996	NULL	5947.166666666667	NULL	NULL
10m1.1_123_None	c	8746.0	114.3379830779785	4402.589	NULL	NULL	NULL	NULL
10m1.1_123_None	d	9504.0	105.21885521885523	4481.686645636172	NULL	NULL	NULL	7852.877551020408
10m1.1_123_None	f	11505.0	86.91873098652759	4412.903	NULL	4309.553319919517	8652.390342052315	NULL
10m1.1_123_None	e	9228.0	108.3658430862592	NULL	5278.639790575916	NULL	NULL	5436.6
1.01.0_1000_None	a	8312.0	120.30798845043311	3495.6572504708097	NULL	3160.012793176972	NULL	NULL
1.01.0_1000_None	b	8956.0	111.65698972755695	3876.1072555205046	NULL	4430.734693877551	NULL	NULL
1.01.0_1000_None	c	8735.0	114.48196908986834	3655.852	NULL	NULL	NULL	NULL
1.01.0_1000_None	d	6824.0	146.54161781946073	2972.2150423728813	NULL	NULL	NULL	3900.5714285714284
1.01.0_1000_None	f	8758.0	114.18131993605846	2429.67	NULL	2399.3275862068967	4805.107279693487	NULL
1.01.0_1000_None	e	8018.0	124.7193813918683	NULL	4274.744525547445	NULL	NULL	5783.0
2.02.0_1000_None	a	11152.0	89.6700143472023	7290.1240157480315	NULL	5568.400406504065	NULL	NULL
2.02.0_1000_None	b	10751.0	93.01460329271696	5660.533970276008	NULL	6578.775862068966	NULL	NULL
2.02.0_1000_None	c	10495.0	95.28346831824679	5306.838	NULL	NULL	NULL	NULL
2.02.0_1000_None	d	6663.0	150.08254539996997	2761.52	NULL	NULL	NULL	3951.58
2.02.0_1000_None	f	12022.0	83.18083513558476	4754.79	NULL	4659.429447852761	9336.094069529652	NULL
2.02.0_1000_None	e	10980.0	91.07468123861567	NULL	6772.124868835257	NULL	NULL	5920.382978723404

36 rows in set (0.01 sec)

```
mysql>
```



SmartDeployer Visualization Tool

Results

The following experiments test various Cassandra configurations for the SmartDeployer :

- Experiment 1: 8 VCPU'S, 16 GB RAM for 1 Docker container
- Experiment 2: 4 VCPU'S, 8 GB RAM for 2 Docker containers
- Experiment 3: 2 VCPU'S, 4 GB RAM for 4 Docker containers

Workload A: Update heavy workload

This workload has a mix of 50/50 reads and writes.

Workload B: Read mostly workload

This workload has a 95/5 reads/write mix.

Workload C: Read only

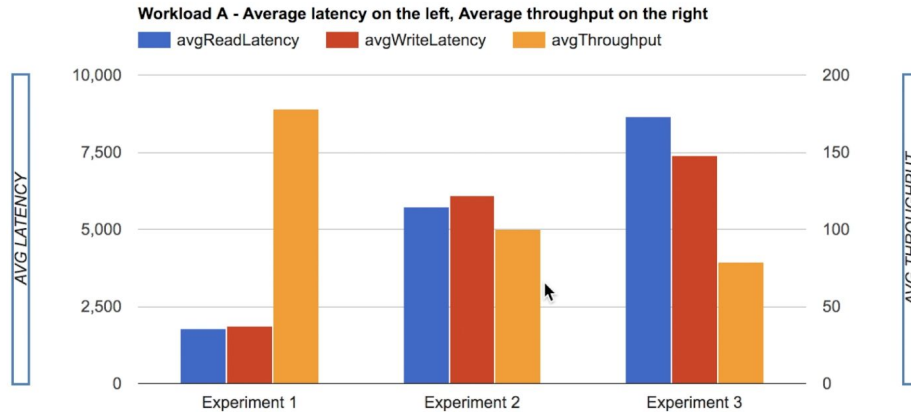
This workload has 100% read.

Workload D: Read latest workload

This workload has new records been inserted, and the most recently inserted records been the most popular.

Results

The workload A has a mix of 50/50 reads and writes.



- Experiment 1: 8 VCPU'S, 16 GB RAM for 1 Docker container
- Experiment 2: 4 VCPU'S, 8 GB RAM for 2 Docker containers
- Experiment 3: 2 VCPU'S, 4 GB RAM for 4 Docker containers

Challenges

- ❖ Deploying Cassandra through microservices on multiple Swarm VMs is in beta stage and has not been stable by Docker yet.
- ❖ Cassandra node is not stable when implement by docker containers
- ❖ Lack of documentation



Lessons Learned

- ❖ Experience advantages of agile methodology
- ❖ Deploying microservices through containers for a distributed NoSQL database
- ❖ As the result of the experiment, difference in the setting lead to huge gap in terms of performance
- ❖ Time management



Smart Deployer

Autonomic deployment and configuration of Cassandra through microservice

CMPUT401 Winter 2018

Xuan Cao, Yuan Feng, Nick Anic, Zhipeng Chang



Why this software

Due to the programmability of cloud environments, microservices and finally recent advancements in machine learning algorithms, it is now feasible to enable autonomic management in distributed big data systems in a dependable manner. The proposed smart deployer in this project strives to facilitate the installation and configuration of Cassandra that is one of the most popular NoSQL datastores to manage big data. The ultimate goal of the smart deployer is to deploy Cassandra with acceptable performance in an autonomic manner. This way, the end-users avoid spending non-trivial amount of their time and energy on deployment and configuration the datastore cluster so that they can be focused on their real and core business.

Technologies



Docker is an ecosystem that facilitates building and deployment applications using kernel-level virtualization. It enables microservices for many platforms, applications and scenarios.

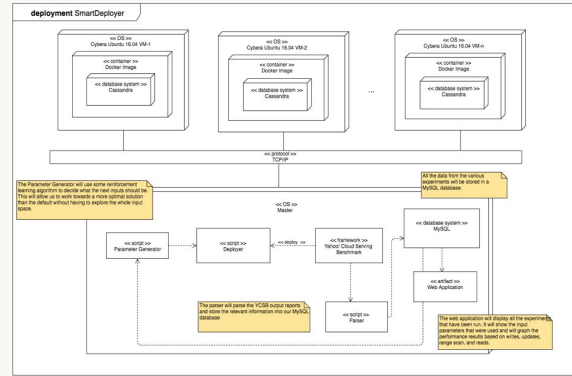


Apache Cassandra is a free and open-source distributed NoSQL database management system designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure

django

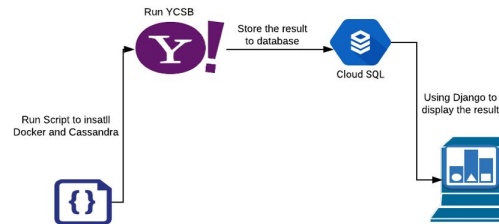
Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Platform Architecture



Workflow

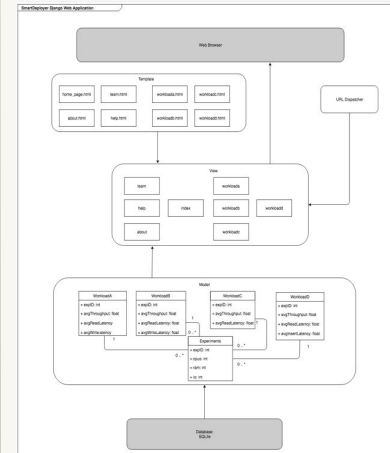
The software is mainly contains two parts, first part is to install the tools and the environment such as Docker and Cassandra, then run our YCSB to get the overall performance. After that used Django to display the result of the YCSB.



Challenges

- Deploying Cassandra through microservices on multiple Swarm VMs is in beta stage and has not been stable by Docker yet.

Visualization Tool



Previous Studies

This is a hot topic right now, and there is a lot of research going on in this area but nothing has been significant as of yet.

Expected outcomes

Design and develop a smart deployer that deploys and configures Cassandra datastore through microservices in such a way that an "acceptable performance" of the cluster will be guaranteed. Mainly used the script to deploy the docker and Cassandra, and Django to display the result on a web.

In the future version, the deployer should be able to generate the parameters for users automatically.