LIVE MIGRATION OF CONTAINERS IN EDGE COMPUTING EMBEDDED PLATFORM

ABSTRACT -With the advent of Internet of Things (IoT) there has been a huge increase in the amount of data generated and transferred across the internet. There is a pressing need to provide large bandwidth to ensure real-time processing of the large amount of data. Edge computing is a solution to this problem that can ensure about 40 percent of the IoT-created data will be stored, processed, analyzed, and acted upon close to, or at the edge, of a network. It is important that these edge nodes are always available to avoid downtime and provide seamless service. This project proposes to provide high availability of services running on edge nodes by means of live migration across these nodes. This project aims at implementing edge nodes that have an application that runs continuously inside containers. The services that process any data will be running in containers to facilitate migration across nodes.

INTRODUCTION

Edge computing performs processing at the edge of the network and hence reduces the overhead for transfer of data and saves bandwidth. To provide high availability of service, it is important to have quick response time and eliminate downtime or network failure. The edge computing helps to move the cloud management services for multi-tenant user applications using a distributed heterogeneous edge node. Live migration is transferring application instances across nodes without disconnecting the clients. It helps to solve many problems such as downtime during hardware maintenance or unexpected failure.

At present, the existing cloud management services does provide only the remote deployment of multi-tenant user applications on the cloud of edge nodes, but it does not have any facility to perform live migration of the nodes across the application.

In this project, the auto discovery of all the active end nodes to remotely deploy multi-tenant user application will be achieved along with the live migration of the nodes across the application. The platform uses container based virtualization to deploy and launch isolated, multi-tenant user applications.

SYSTEM ARCHITECTURE:

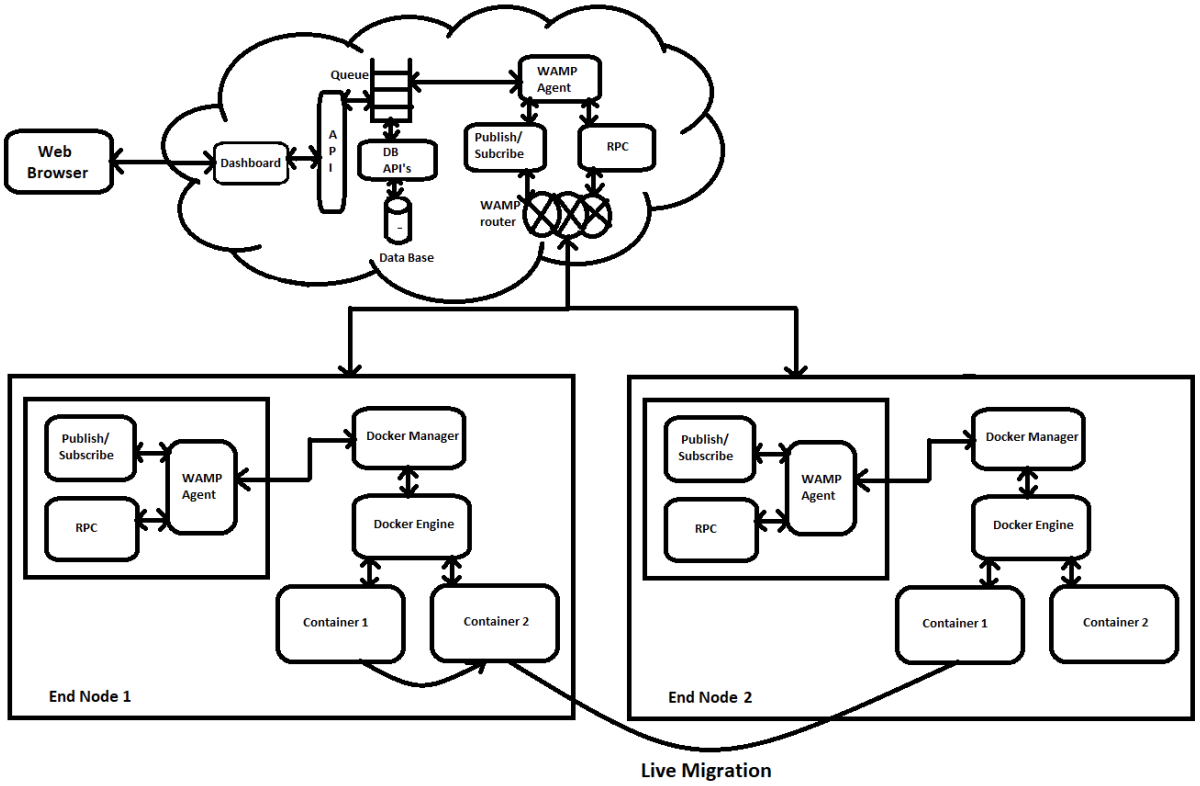

Figure 1. System Architecture

MODULES:

The system can be majorly divided into 4 parts:

- User Interface
- Cloud
- Edge nodes.
- Live migration.

Each part of the system is further divided into sub-systems:

- CLOUD:
  - Database.
  - Messaging service.

- USER INTERFACE:
  - User dashboard.

- EDGE NODES:
  - Container.

DATABASE:

The database is used to store the details related to the user such as the login credentials. The database also contains the data about the container that is allocated to the respective user, the images of the container etc. The database consists of three major tables.

1. compute - the table information of the user and the respective containers.
2. image - this table has the images of the container for the edge devices to be connected.
3. device - this tables lists the devices present.

In-order to communicate the data between the REST and the messaging service, a python package that wraps the SQLAlchemy ORM package is implemented.

MESSAGING SERVICE:

The server and the client messaging or communication is done with the help of Web Application Messaging service(WAMP). The WAMP is an open standard WebSocket protocol, which has two different types of messaging patterns in a single protocol.

1. Remote Procedure calls.
2. Publish-subscribe.

USER DASHBOARD:

The User dashboard in the design is to create, start or stop a container. The user can also run and upload an application into the container. The dashboard consists of different web-pages

1. Sign/Register screen.
2. Screen to create a container, or upload an application or check the device connected and live migrate containers. Horizon, an OpenStack framework, is used for created the user dashboard.

DOCKER CONTAINERS:

Docker container is a complete software package required to run a user's application. It is run on the edge node by abstracting the underlying OS.
Swarm is a container Schedule used.

LIVE MIGRATION:

Live migration is the new concept extended in this project. The live migration of containers are done on the top of the existing project titled "Edge computing embedded platform" (Abhishek Gurudutt, Chinmayi Divakara, Praveen Prabhakaran and Tejeshwar Chandra Kamaal).
The live migration of the containers present in the edge nodes are done using CRIU (Checkpoint and Restore In User-space). This allows the users to backup and restore any process lively in the user-space.
Below are the commands to perform live migration using CRIU

Initially, the CRIU must be installed.

apt-cache search criu
apt-get install criu

Dump:

In-order to freeze the process and migrate it live, the dump command is used on the source side.

criu dump -t <pid> --images-dir

pid can be obtained used the following command in the base directory.

ps aux | grep *filename*

The image of the process will be created.

Copy:
The image created must now be copied to the target destination from the source.

scp -r <image-dir>/ <dst-ip>:

The image will now be copied onto the target.

Restore:
The process now must be restored on the target.

sudo criu restore --t <pid> --images-dir <path-to-images>

Now the process will be restored onto the target side.

DELIVERABLES:

1. Able to perform live migration of the application running in the containers on the edge nodes.
2. Monitor edge nodes and can create, view and delete containers using the dashboard.
3. The auto discovery of all the active end nodes to remotely deploy multi-tenant user application will be achieved.

EXISTING SOLUTIONS:
In the current architecture of cloudlets, the platform present will manage application using web manager and an environment that supports the user application. This application will run concurrently with the network OS to obtain the sensor data through the wireless network for the edge computed task.

There exists another cloudlet architecture OpenStack++ which uses OpenStack to provide "sensing and actuation as a service". The edge nodes those are involved in the architecture are only sensing nodes. The managerial services to these nodes are provided by the cloudlet. The cloudlet gives multi- tenant access to the sensors and configure the cloudlet using messaging protocols such as web application messaging protocol (i.e., WAMP) or Web Socket. This open stack architecture can be expanded to cloudlet at the network edge to provide strong services.

This project is an extension of the "Edge computing Embedded Platform" (done by Abhishek Gurudutt, Chinmayi Divakara, Praveen Prabhakaran and Tejeshwar Chandra Kamaal), which is about the auto discovery of all the active end nodes, in a distributed end nodes environment. The auto discovery of all the active end nodes is used for implementing multi-tenant user application.

REFERENCES:

1. Samah Ahmed Zaki Hassan, "STAR: A Proposed Architecture for Cloud Computing Applications", 2012 1ntemational of Cloud Computing, Technologies, Applications & Management, pp. 186 – 192.
2. https://chandanduttachowdhury.wordpress.com/2015/08/10/test-driving-criu-live-migrate-any-process-on-linux/
3. Tarik Taleb, Sunny Dutta, Adlen Ksentini, Muddesar Iqbal and Hannu Flinck, "Mobile Edge Computing Potential in Making Cities Smater", March 2017 IEEE Communications Magazine, pp. 0163 – 0170.

4. Roberto Morabito, Nicklas Beijar, “Enabling Data Processing at the Network Edge through Lightweight Virtualization Technologies” 2016 IEEE.
5. Fabian Romero, Thomas J. Hacker, “Live Migration of Parallel application with Open VZ” 2011 IEEE, 526 – 531.
6. https://criu.org/Live_migration
7. https://criu.org/Docker
8. http://crossbar.io/docs/Command-Line/
9. https://docker-py.readthedocs.io/en/stable/containers.html
10. https://github.com/AbhishekGurudutt/ecep_client