Cluster Of Workstations for cloud foundry based functions and applications

Dr Bheemaiah Anil Kumar, UMN Twin Cities, Synergy Robotics BC, Seattle WA 98125
psychEDU@yopmail.com

Abstract:
In this publication we describe on OpenRAN based cluster of workstations, implementing cloud foundry Kubernetes, architecture for AWS inspired Lambdas and threads on cloud foundry, implemented on dedicated units as a cluster and on robotic segway RMP 440 platforms as HTCondor based thread schedulers, using PyMACS integrated with a resident kernel inspired by the shadow process of HT Condor. This allows for thread scheduling on idle robotic CPUs
Keywords: Kubernetes, Cloud Foundry, AWS, VPN, HTCondor.

1.0 Introduction

With the advent of cloud programming and the rise of major players like Google Cloud, Amazon AWS, and Microsoft Azure, the need for retro cloud as a cluster of workstations or PCs for cost-effective @edge computing. In this paper, we explore the use of retired, but computationally viable hardware, in the form of workstations and desktops in functional aesthetics, through the Free Willy Venture in digital literacy, as mobile and dedicated resources, with GeForce GPGPU units for 2 TFlops + computing per node, connected to a cloud-RAN using 5G FR1 and FR2 connectivity.

Amazon AWS offers seamless integration of a VPC with an @edge cloud. AWS Snowball(*AWS Snowball*, n.d.) offers data migration services to a secure Amazon VPC. In this paper, we describe a three-layer network with the use of a local academic network, seamless migration of cloud functions or threads as required to the free willy cloud, and prediction based mechanism for the migration of threads to an Amazon VPC.
The snowball is dedicated data storage with computational power @edge to seamlessly connect a business LAN with AWS VPC's, the Free Willy Cloud-RAN provides academic networks, with a similar scalable @edge cluster, with access to a VPC on dedicated and mobile units comparable to HTCondors, cloud offering. While HTCondor uses free time on the academic computing network, Free WIlly instantiates retired resources, in functional installations, like multi-functional installation art, in functional aesthetics for @edge computing.

2.0 OpenRAN

An openRAN architecture is designed as a cloud foundry cloud integration with ROMU(*I'm Tomu - A Tiny ARM Microprocessor Which Fits in Your USB Port*, n.d.) in a scalable cluster of workstations, integrated with independent RAN units on mobile platforms as land bots, with seamless integration to handheld wireless tablets or units in both FR1 and FR2 spectrums of amateur radio, with seamless satellite integration. The ROMU architecture with 5G protocol using multi-protocol wireless RF processors, used to create the ROMU architecture analogous to WOMU and TOMU.

SDR radios are compared in (Powell et al., 2020), applicable to a uniform IoT-based cloud-RAN, with cloud foundry, for use on each node of the cluster. The SDR is software configurable for 5G connectivity and integration to a Cloud-RAN architecture with seamless migration between AWS VPC and a VPC @edge, instances on machine virtualization if needed on cloud foundry. A centralized RAN is simulated from a virtual private cloud both on AWS and @edge on a reconfigurable cluster.

Containers, as well as serverless functions, are supported on the open RAN - cloud foundry network with seamless integration to a virtual private cloud on AWS.

3.0 Free Willy Project, dynamic clusters of PCs

Each desktop is part of a free willy installation or robotics and is configured with SDR for FR1 or FR2 5G cloud RAN, for @edge reconfigurable clusters with open stack configurations for container-based cloud IaC or SaaS, seamlessly integrated with AWS based Kubernetes or other container integrations.

A web interface for a dynamic resource pool of PCs is created on the free willy web site for dynamically reconfigurable clustering of dedicated and robotic resource pools. The dedicated resources are clustered to implement cloud foundry as Kubernetes, executing cloud functions dynamically or applications on containers, other container formulations are also supported in a virtual cloud @edge , tightly integrated to an AWS VPC. This leads to a seamless migration from @edge computing to AWS based computation.

4.0 Cloud Foundry on a cluster and Kubernetes.

Cloud Foundry as Kubernetes is deployable on a reconfigurable OpenStack cluster, reconfigured dynamically, using a web resource. We define a resource pool of dedicated resources and of shared robotic resources, from which clusters are dynamically organized. The shadow system and pubads of HTCondor are used for dynamical process scheduling on shared resources, using PyMACS(*Pymacs*, n.d.) for thread scheduling. Applications or cloud functions with the context are seen as collections of threads in python, executed for the host processor, and executed when spare CPU cycles are available on idle robots.

5.0 Shadow system of HTCondor, integrated with PyMACS and thread scheduling.

The shadow system of HTCondor(htcondor, n.d.) is resident on each robotic platform, on dynamic allocation in the resource pool, from the PyMACS kernel, threads are scheduled based on the pubad resource publication, in JSON from the PyMACS kernel, for thread scheduling on the target architecture.
Equation [1] indicates the architecture of the robotic resource pool.
Let [D] be the pool of dedicated resources and [M] be the resource pool of mobile resources.
Mobile resources could consist of Rover or segway autonomous robots for Mapillary and agroforestry, PyMACS kernel, resident on the mobile resources, allows for shared cloud-RAN dynamic clustering for @edge data mining tasks, useful during critical tasks in poor networking scenarios.
We define a dynamic cluster of workstations or PCs, on the cloud-RAN from each node M or D, depending on the availability of resources.
A gateway in the form of a web DOM with the network pool IP data is present as a log, while a cluster from the snapshot of ([M],[D]) is created with prediction mechanisms for task allocation on the residential cloud. The entire cloud management with seamless migration to AWS is performed from JS code on the hosting service of the web resource.
The architecture is defined in figure 1.

Thread scheduling for symmetric threads [T], for any given set of simulations in real-time, [T, t], is by application threads or FaaS threads, defined collectively by [T, t], scheduled on the dynamic cluster.
Prediction mechanisms are built into the cluster system, for the minimal number of dedicated units, failing which migration to a VPC in AWS is initiated.

6.0 MobileVacBot(™), a new paradigm in Open RAN.

Desktop telerobotics is achieved with the integration of easily available mini robotic vacuum cleaners, with VUI and mobile controls, integrated with tripods and 5G mobiles like the Samsung A71, with more than 2 TeraFlops of GPGPU and MCU FPU power, there is a direct applicability to a network of telepresence applications with Open RAN architecture (Kazemifard & Shah-Mansouri, 2021) using the 5G capabilities of the mobile and the creation of an @edge reconfigurable cluster with HTCondor for concurrent thread execution.

7.0 RetroIBM(™)

The use of the original IBM PC chassis in robotics and aviation, has led to newer plate based chassis solutions, integrable with power systems, storage solutions and the motherboard and chassis of the PC. The direct integration of the plate based PC rack or tower form factor with a robotic chassis leads to a new chapter in mobility and idle computing power for HTCondor, leading to a cluster based Open RAN solution. HPC as distributed dedicated units, is now available from retro computing and mobile platforms. An evaluation of composite structures for robotics and PC chassis applications can be found here.(*[No Title]*, n.d.)

6.0 Conclusions and Future Work.

OpenRAN architectures in the applicability of 5G communications, with open robotic solutions are presented in this paper, with a VPC architecture with a private @edge cloud for HTCondor like idle cycle usage in dedicated robotics, a migration of HTCondor from academic resources to the Free Willy Project.(*Free Willy*, n.d.)

Several new paradigms in robotics are presented, such as the MobileVacBot(™), a telepresence and desktop vacuum robotics solution, and several benchmarks in FinTech, for use with the @edge VPC, compatible to migration to cloud foundry(“Advanced Amazon Virtual Private Cloud (Amazon VPC),” 2018; Sammons, 2016; Winn, 2017) and AWS.(“Advanced Amazon Virtual Private Cloud (Amazon VPC),” 2018; Sammons, 2016)

Reuse of computers in robotics and Mapillary imaging towards an era of autonomous mobile computing, the after life of desktop and mobile computing is thus presented as a HPC scenario, of cluster based computing. Thus Cluster based HPC is proven in the metaphors of the Free Willy project.

7.0 References.

Advanced Amazon Virtual Private Cloud (Amazon VPC). (2018). In *AWS® Certified Advanced Networking Official Study Guide* (pp. 57–92). https://doi.org/10.1002/9781119549000.ch3

AWS Snowball. (n.d.). Retrieved April 28, 2021, from https://aws.amazon.com/snowball/

Free Willy. (n.d.). Retrieved April 28, 2021, from https://freewillybc.wordpress.com/

htcondor. (n.d.). *htcondor/htcondor-python*. Retrieved April 28, 2021, from https://github.com/htcondor/htcondor-python

I’m Tomu - A tiny ARM microprocessor which fits in your USB port. (n.d.). Retrieved April 28, 2021, from https://tomu.im/

Kazemifard, N., & Shah-Mansouri, V. (2021). Minimum delay function placement and resource allocation for Open RAN (O-RAN) 5G networks. In *Computer Networks* (Vol. 188, p. 107809). https://doi.org/10.1016/j.comnet.2021.107809

[No title]. (n.d.). Retrieved April 28, 2021, from https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjs_NXPt6HwAhWIXM0KHTLVDawQFjAAegQIBBAD&url=https%3A%2F%2Fwww.mdpi.com%2F2073-4360%2F11%2F10%2F1667%2Fpdf&usg=AOvVaw12BtIVO4zCnUNRnB4kl2Fq

Powell, K., Sabri, A., Brennan, D., Marojevic, V., Michael Barts, R., Panicker, A., Ozdemir, O., & Guvenc, I. (2020). Software Radios for Unmanned Aerial Systems. In *arXiv [cs.NI]*. arXiv. http://arxiv.org/abs/2004.01987

pymacs. (n.d.). Retrieved April 28, 2021, from https://pymacs.wordpress.com/

Sammons, G. (2016). *Aws Vpc: Virtual Private Cloud Beginner's Guide*. Createspace Independent Publishing Platform.

Winn, D. C. E. (2017). *Cloud Foundry: The Definitive Guide: Develop, Deploy, and Scale*. "O'Reilly Media, Inc."