**SCALABLE AND SUBSTAINABLE DOCKER PLATFORM FOR**

**CONNECTED ROBOTS USING SHARED SERVICES OF ROBOT OPERATING SYSTEM**

**ABSTRACT**

*The objective of the research work is to propose container based shared ROS to the connected robotics networks. Many host applications developed for robotic functionalities needs ROS service to achieve coordination. The services rendered by ROS is not sufficient for connected robots to perform special functionalities. The docker container with special functionalities and interfaces, the availability and reliability of the industrial automation can be achieved. The sharing of docker container with its interfaces can be initiated and terminated in minimal time to achieve high production .A container based process automation in the robotic industry will not only give high yield and reliability with number of connected robots under unreliable and untrack able services .*

**INTRODUCTION:**

The scope of Robotics and the artificial intelligence is becoming widespread, so the collaboration between researches is necessary in order to build large systems . Organising the ROS software system into packages supports collaborative development. ROS packages is simply a directory which contains a XML file describing the package and starting any dependencies . At time of writing several hundred ROS packages exist across several publicly viewable repositories and hundred more likely exist in hundred repositories at various institutions and companies [5].The ROS core is distributed as its own package repository . However ,the ROS repository includes only the base ROS communications infrastructure and graph-management tools software which actually builds robotic system using ROS is provided in second repository.

The primary goal for optimizing resources is the shared services. All the robots share the same set of behaviours. These behaviours could be performed by the robot itself or they required cooperation with other robots that would run them in their place. The sharing of services among autonomous public agencies requires the close cooporation among those agencies in the context of public service network. Sharing of services should not be considered as one type of construction initially. The drivers for sharing are motivated externally by drastic budget cuts, but reinforced by internal lack of resources and capabilities. The results are that organizations start to share services that directly affect the functioning of their service delivery . Service delivery can be considered as one of the core competencies of POs, Although its less important than policy-making and ensuring public values like security and safety[6]. The net effect is that the behavior is done opaquely ,regardless of which the robots have competed it.

Container-based virtualization offers better performance by reducing the overhead potentially and thus improves the utilization of data centers . The physical performance of a single machine is compared with running the same application using two different isolation techniques. Namely isolation through virtualization and containerization. Combining LXC with Docker and CoreOS,the whole package provides a lightweight , clean, full featured base layer for isolating application infrastructure. The CoreOS ,Docker, LXC stack has a lot of potential . If the technology could be improved to provide better resource isolation,which seems to be only major flaw it could drastically reduce overhead on major server deployments, even in public clouds. Thus the performance of container-based virtualization is better than with hypervisor-virtualization in most cases,nand it is almost as good as native application.

Docker is an open source container technology with the ability “to build ,ship, and distributed applications”. The security level of Docker containers could also be increased with the operator runs them as “non-privileged “ and enables additional hardening solutions in linux kernel [2]. Docker run in PAAS variables attaches shell script, to built in stacks, clone tailoring elastic IP, mapping port, monitoring using Seccom ,Berkely in filters,mocking storage problem. Operations team in managing container elastic bean stack and glit gmip with long running productivity ,Docker security managed per application as a micro services and breaks into logical components Reds, mem server, web server and flume as daemon. Docker Containers offer an added stratum of defence by separating the application and the host, and the between applications . It has process restrictions, device and file restrictions, application image security ,docker allows container images to originate from did or from a remote registry and these congiguration controls described above are provided as a run-time facility where isolation,orchestration,scheduling and application management mechanisms are concentrated [1].

“ROS 2.0 will support the growth of the ROS community by making it much easier to work with small embedded systems, teams of multiple robots and robots that require real time control”. ROS-I is a software library that builds on leverages its power and flexibility manufacturing automation equipments including industrial robot arms [7].

**DOCKER PLATFORM FOR CONNECTED ROBOTS:**

Earlier automation processes have not address the substainability and scalability issues of connected robots. Once the shared services have been utilized there will be no dependency and communication challenges with vast number of sensors and actuators the service delivery of connected robots may gets degraded. It is not only due to non availability of the essential platforms and unreliable component service. So a docker platform for robot operating system have been proposed to connected robots to attain substainability and scalabil

ity.

The consequence factors from process, people and product are considered and orchestrated in order to achieve in the expected substainability for the product from the manufacturing plant.

Without docker platform,

Sustainability of the system is proportional to themultiplication of the same for process,people and product without any docker platform whereaswhen the docker is introduced the sustainablilityhas changed to platform,product and person.

With docker platform,

**DOCKER PLATFORM FOR ROS SERVICES:**

**G:\ \docker_final.png**

**CAMERA BASED HEAVY PARTS FEEDING ROBOT:**

**G:\ \ROS .png**

**Special Satefy functions within a container:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SAFTEY CHECKER DATA | | SAFETY RULE | SAFE STATE | UNSAFE –STATE | | PRECAUTION  STATE |
| TEMPERATURE | |  |  |  | |  |
| POSITION | |  |  |  | |  |
| BATTERY | |  |  |  | |  |
| LOAD |  | |  | |  |  |

**REFERENCES:**

[1] K.N Bala Subramanya Murthy , Manu.A.R, Jitendra Kumar Patel, Shakil Akhtar, V.K Agarwal PhD “Docker Container Security Via Heuristics-Based multilateral Securtity – Conceptual and Pragmatic Study”

[2]Thanh Bui ,Aalto university school of science, “Analysis Of Docker Security”.

[3]Csaba Rotter, Lorant Farkas ,Gabor Nyiri,Gergely Csatari, Laszlo Janosi, Robert Spinger “Using Linux Containers in Telecom Applications”

[4]Mathijis Jeroen Scheepers “Virtualization and Containerization of Application Infrastructure : A Comparison”

[5] Morgan Quigley, Brain Gerkey, Ken Conley, Josh Faust, Jeremy Leibs, Eric Berger, Rob Wheeler, Andrew Ng , “ROS: an open Robot Operating System”

[6]Marijin Janssen, Muhammad Kamal, Vishanth Weeraakoddy, Anton Joha “Shared Services as a Collaboration Strateegy and Arrangement in Public Service Network”

[7]Brian Gerkey, ”ROS, the Robot Operating System, Is Growing Faster Than Ever”,

[8]”Transparent multi-robot communication exchange for executing robot behaviors”