A Holistic Cloud-Enabled Robotics System for Real-Time Video Tracking Application

Bingwei Liu[1], Yu Chen[1], Erik Blasch[2], Khanh Pham[3], Dan Shen[4], and Genshe Chen[4]

[1] Department of Electrical and Computer Engineering,
Binghamton University, SUNY, Binghamton, NY, USA
{bliu11,ychen}@binghamton.edu
[2] Air Force Research Laboratory, Rome, NY, USA
Erik.Blasch@rl.af.mil
[3] Air Force Research Laboratory, Kirtland AFB, NM, USA
khanh.pham@kirtland.af.mil
[4] Intelligent Fusion Technology, Inc. Germantown, MD, USA
{dshen,gchen}@intfusiontech.com

Abstract. Future distributed sensor fusion applications will require efficient methods of information management such as Cloud computing. Using a server-based cloud-enabled software architecture would increase performance over hardware constraints (e.g., power, memory, and processors). In this paper, we propose a comprehensive framework for information fusion demonstrated for Cloud Robotics, which possesses user favorable features such as good scalability and elasticity. Robots are connected together to form a networked robotic system that is able to accomplish more computationally intensive tasks. Supported by the emerging Cloud computing technology, cloud-enabled robotic systems (CERS) provide even more powerful capabilities to users, yet keeping the simplicity of a set of distributed robots. Through an experimental study, we evaluate the memory, speed, and processors needed for a video tracking application.

Keywords: Cloud computing, image tracking, robot networks.

1 Introduction

Today's robots or robotic systems can perform complex tasks in real-world, dynamic environments, thanks to the advances of both microprocessors and generic CPUs. These systems are usually expected to achieve high mobility and are sensitive to communication time delays. When a task is beyond a single robot's capacity, multiple robots are required to accomplish the task. Networked robots, as distributed information fusion systems, require advances in resource management [1], Cloud-computing [2], and target tracking [3] to support effective situation awareness [3], [4] while at the same time providing secure communications [5]. Typically, robotic systems utilize image processing systems [6] for coordinated target tracking that have hardware limitations from intense measurement processing for image segmentation [7],

James J. (Jong Hyuk) Park et al. (eds.), *Future Information Technology*,
Lecture Notes in Electrical Engineering 276,
DOI: 10.1007/978-3-642-40861-8_64,

state estimation [8] and target assessment [9]. To address these issues, the paper describes a Cloud-enabled environment to increase video-tracking performance.

Traditional standalone robots are limited by constrains such as power consumption, computing ability, storage space, etc. Offloading part of a task to a remote server or distributing a complex task to a group of robots could potentially reduce response time, achieve more accurate decisions and consume less energy. Following the original idea of Internet-based tele-operated robots, the term "Networked Robots" was adopted by the IEEE RAS Technical Committee on Networked Robots in 2004 [10]. There are two different types of networked robots, tele-operated and autonomous [10]. Since networked robots are connected via a network, tele-operated robots can be accessed over a wider area. Autonomous robotic systems, on the other hand, allow Robots and sensors to coordinate through a network to perform complicated tasks that are difficult for a single robot. However, computation, storage and knowledge sharing are still limited at the distributed local network of robots.

As a new computing paradigm, Cloud computing (CC) has attracted researchers from the distributed computing community and information technology (IT) service providers. The well-known attractive features of CC include on-demand scalability of highly available and reliable pooled computing resources, secure access to metered services from anywhere, and displacement of data and services from inside to outside the organization. Due to the low cost of storage services provided in a Cloud, compared with purchasing and maintaining a storage infrastructure, it is attractive to companies and individuals to outsource applications and data storage to public Cloud computing services.

Cloud computing allow users to focus on their application without worrying about IT infrastructure plan. Central Processing Unit (CPU) cycles, storage space and even network services can be purchased on demand. When combining with the Cloud, robotic systems are able to take advantage of the almost unlimited parallel computing and vast storage space of Cloud computing. Cloud robotics was introduced by James Kuffner at Google to describe this new approach to robotics [11]. There is active research on Cloud Robotics in both Cloud and robotics communities [12], [13], [14], [15], [16], [17]. Most of these papers focus on special applications of Cloud robotics, with limited consideration about holistic system design, implementation details, or information fusion opportunities enabled from the Cloud environment.

In this paper, we propose a comprehensive distributed Cloud-enabled robotics framework for information fusion and provide a preliminary performance evaluation through a case study based on a video tracking application. In this framework, we considered the implementation of both the Cloud and the robot networks with additional security features, leading to a holistic framework. In addition, at the Cloud side, we include a virtual machine (VM) cluster and a physical machine cluster into our framework as a dynamic computing clusters.

The rest of this paper is organized as follows. Section 2 briefly discusses related work in networked robots and Cloud robotics. Then we introduce a holistic Cloud-enabled robotics system (CERS) framework in Section 3. Section 4 reports our preliminary performance evaluation result obtained through a case study of an image processing application. And we conclude this paper in Section 5 with some discussions about our on-going efforts.

2 Related Work

Chen et al. defined the concept of Robot as a Service (RaaS) based on Service-oriented architecture (SOA) [12]. The authors presented the idea of combining robot services with a Cloud, using Microsoft Robotics Developer Studio (MRDS) and Visual Programming Language (VPL). RaaS was implemented and tested on two processors, Intel Core 2 Due and Atom.

Agostinho et al. [13] proposed a Cloud computing environ- ment for networked robotics applications. VMs are assigned different roles in the Cloud environment. A layered workflow management system was used for scheduling purposes.

Kehoe et al. [14] illustrated a system architecture for Cloud- based robot grasping using a Google object recognition engine. A prototype and initial experiments for a Cloud-based robot grasping system were implemented in their work.

Arumugam et al. [15] proposed a distributed agents with collective intelligence framework, in which heterogenous robots can work together in large environments. A Robot Operating System (ROS) platform [18] was used for sensor data collection and communication. In their implementation, Hadoop was used as a high performance computing and storage platform. A grid based FastSLAM algorithm was implemented as a Hadoop Map/Reduce task[15].

Hu et al. [17] suggested using gossip protocols for com- munication between robots within a highly dynamic mobile robotic network. No route discoveries and maintenance are needed in this system. It is simple to implement and has low computation and memory requirements. However the latency of message exchange could be expensive in this ad hoc wireless network. The authors also considered energy consumption in the decision of whether to offload computation to the Cloud.

Compared to previous works in Cloud robotics, we highlight the following contributions of this paper.

1) Relatively complete framework: The architecture of both the Cloud side and the local robot network side are considered in our framework. Previous work either built the Cloud environment on a single machine using a virtualization software instead of using a Cloud platform, or didn't provide enough specifications of the Cloud architecture.

2) Dynamic computing cluster: Dynamic computing cluster: We combine a virtual ma- chine cluster with a physical cluster in our dynamic computing cluster infrastructure. This architecture has potential flexible scalability and efficient resource allocation.

3) Working flow prototype: We investigate the working flow of a user requesting a robot service from the web interface provided by the Cloud as a proof of concept example.

4) Performance comparison between a physical workstation and multiple virtual machine instances: We evaluate the per- formance of a workstation and different size of virtual machine instances in an video tracking algorithm in the processing of simultaneous requests.

3 A Holistic Cloud Enabled Robotics System

3.1 Cloud Robotics

Cloud robotic systems take advantage of all the benefits of the Cloud, while at the same time providing users who want to focus on the functionalities of robots in an economic way to investigate their robotic system. There are at least the following benefits when we integrate multiple robotic systems within a Cloud architecture.

1) Faster robotic applications development: Developers in robotics can cooperate in the platform provided by the Cloud to develop robotic applications in a more efficient way. Once application developers deploy their applications into the Cloud, they can take advantage of the fast provision technology of the Cloud to serve almost unlimited users.

2) Easier to get started: A user doesn't need to configure the development environment in order to have an initial idea about robots in general, or a specific type of robot. All they need is a web browser to access these services in Cloud robotics.

3) More efficient robot resources usage: Robot owners can also reduce their cost in maintenance by charging a small amount of fee to each user.

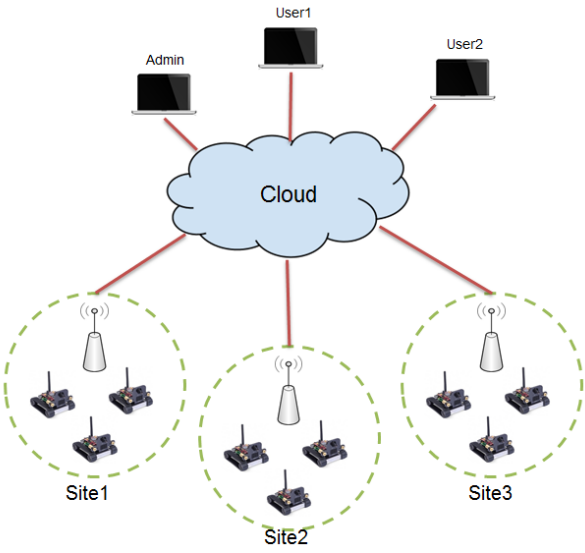

Fig. 1. A typical scenario of a Cloud Robotic System

In this paper, we propose a more general holistic framework for Cloud-enabled Robotic System (CERS). Fig. 1 is a typical scenario of a CERS. Robotic systems that are able to provide robot services register with the Cloud as robotic service providers. The Cloud provides a uniform web interface for all customers. Developers communicate and cooperate with each other through the web portal. Administrators also manage robotic systems through a specific secure website hosted in the Cloud. Robotic systems and the Cloud exchange messages through ROS messaging mechanism [18].

3.2 An Abstract Architecture

As a general discussion of CERS, we first consider the abstract architecture in Fig. 2. At the top of this system is an application layer, including three types of applications (APPs) for different purposes. The Management Apps consist of authentication and access control functionalities, as well as the management of computing, storage, network, auditing and QoS etc. The services Apps include customized applications for different robot systems, robot resource database and robot systems monitor etc. The user APPs provide applications directly interacting with end users, such as a web page that can see the video captured by a robot in real time. The Application Programming Interface (API) layer is the middleware between applications and underlying layers where developers use them for their application development.

Under the API layer is the computing, storage and databases platform layer. Basic databases such as user registration and robot states will be created for the management of the system. The Cloud provides elastic computing and storage resources on demand and schedules jobs or tasks according its load balance policies. High availability can also be provided to desired clients.

In order to provide generic services in Cloud, we suggest that heterogeneous robotic systems use the widely adopted Robot Operating System (ROS) [18] as the robot platform. The Cloud communicates with ROS directly to acquire data and send commands to robots. Each Robot team must have at least one ROS master to take care of message exchange, robot services registration and robot control. The residence of ROS masters is flexible. An ROS master can run on a local computer which locates at the same area as managed robots. It can also run on a virtual machine in the Cloud.

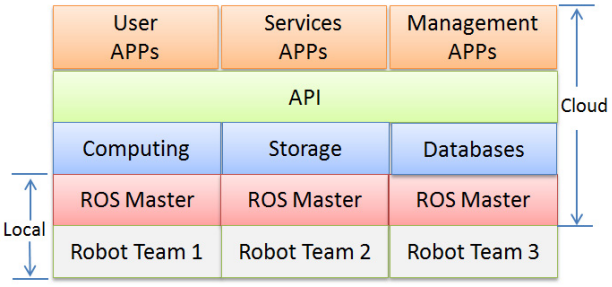

Fig. 2. Abstract Architecture of a Cloud Robotics System

3.3 Robot Network

In the ROS system, the term "nodes" are processes running on the robot system. Multiple nodes can run on the same ROS system. The ROS master serves as a name server for all other nodes so that they can find each other. Once the existence of other nodes are discovered, each node (robot) can communicate with any node directly in ad hoc wireless mode or through a centralized access point. A good choice to host a ROS master is a computer that is in the same local network with all other distributed robots. An access point is also needed in order to communicate with the Cloud. Another option is to run the ROS master on a virtual machine in the Cloud. In this case, a robot must have a public Internet Protocol (IP) address or a local manager node is added to route the traffic.

There are two running modes under our proposed CERS:

1) **Local mode:** *When the network connection is not satisfied, the local robotic network works under local mode. A manager node will be chosen and host the ROS master. This manager node can be a laptop or simply a robot that has most powerful processing ability.*

2) **Cloud mode:** *In order to take advantage of the Cloud, when the round-trip delay time (RTT) of the communi- cations message to the Cloud is suitable for the services provided by the local robot system, the ROS master will switch to Cloud mode.*

Robot states are stored in a local ROS master as well as Cloud ROS master, and a clone of the local ROS master is also registered in the dedicated Cloud database.

3.4 Cloud Side Consideration

Fig. 3 illustrates a possible implementation at the the Cloud side in order to provide robot services to the public. The underlying Cloud infrastructure employs a dynamic virtual machine (VM) computing cluster and storage system. The following aspects are worth noting when designing a Cloud robotic system.

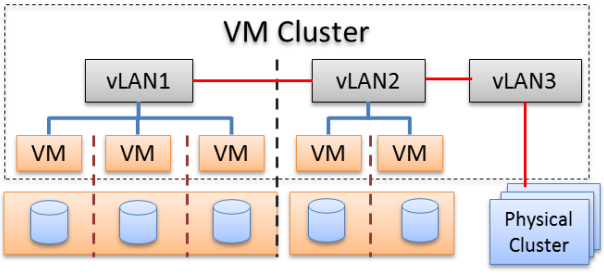

Fig. 3. Dynamic computing cluster and storage system

1) Dynamic Computing Cluster: The dynamic cluster is a computing cluster consisting of VM cluster and physical cluster. According to the system load, the Cloud will allocate appropriate computing resources to different tasks. The dashed lines in Fig. 3 represent the appropriate isolation between every two VMs. One of the VM in the cluster is chosen as a computing scheduler. This scheduler monitors the performance and load of all VMs. We assume that this VM has strong secure environment that is difficult to be compromised. This is reasonable since we can always use a private IP for this VM and further protect it behind a secure firewall. This elastic computing structure is able to process sophisticated computing intensive tasks.

2) Separated Storage: Every VM can have one or more associated storage volumes. These volumes are assigned by the hyperviser and cannot be modified without root privilege in the hypervisor. User data security and integrity can also be included in the system by adopting storage integrity auditing service. A distributed storage system like Hadoop Distributed File System (HDFS) [19] can be employed as the underlying storage infrastructure.

3) Distributing Functionalities into VMs: Every individual functionality in our model can be implemented on a virtual ma- chine with proper computing and storage resources to handle user requests, message exchanges or performance monitoring. This is an efficient and economic way to implement a complex system. The elasticity of the Cloud will also assure high availability of services. Once any functional VM goes wrong, a new VM can be deployed in minutes to replace the bad VM.

3.5 Web Interface

Providing computing services through the web has been proved by public Cloud provider like Amazon to be a successful service delivery mechanism. We also recommend using RESTful web services [20] to implement Cloud applications.

The Cloud provides a uniform web interface for administrators, developers and regular users to access data and perform tele-operation to robots. Administrators monitor the health of VMs, states of robots and user behavior. Developers store program codes on the revision control and source code management system provided by the Cloud and can easily cooperate with each other on the hosted repository. Regular users visit the web interface to access history data, tele- operated robots, request data processing and monitor VMs with owner privilege when desired.

3.6 Security

Security is usually the first concern to both a service provider and a user. The provider wants to assure their properties are safe in the Cloud. Important and sensitive data in the Cloud should be stored securely. The Cloud should also be able to appropriately defense attacks to web server and user data. The study of security in Cloud computing is still an open area, hence it needs to be considered carefully when putting robot

resources online. A compromised server can send malicious commands to robots, causing tremendous lost for robot owners.

To protect user data and ensure the security of the system, several policies need to be enforced:

1) VM Isolation: As we mentioned above, VMs must be isolated by the hypervisor even when they are running on the same physical machine.

2) Secure Storage: The Cloud must implement secure stor- age of user data before deploying robot services. The Cloud can provide an auditing service for users to guarantee data integrity. A third party auditor (TPA) can be employed to audit the Cloud in data integrity. The auditing procedure is usually a challenge-response style. A user or the TPA challenges the Cloud with the integrity of his data. The Cloud then responses with a message to prove that it is actually possessing the user's data and all data blocks are intact in the Cloud storage infrastructure.

3) Network Management: Each virtual Local Area Network (vLAN) has strict rules to prevent unauthorized access even within the same vLAN. Usually, the hypervisor will take care of the packet routing and virtual network optimization. If necessary, a trusted network manager can be deployed to monitor the network with proper permission.

3.7 Work Flow Example

As a proof of concept example, we show a working flow of a user requesting a service through the Cloud robotic system in Fig. 4. The user first visits the web interface and requests for the service (1). The web server than call the user authentication and access control module (2), which then queries the user database (3) and grants the access if

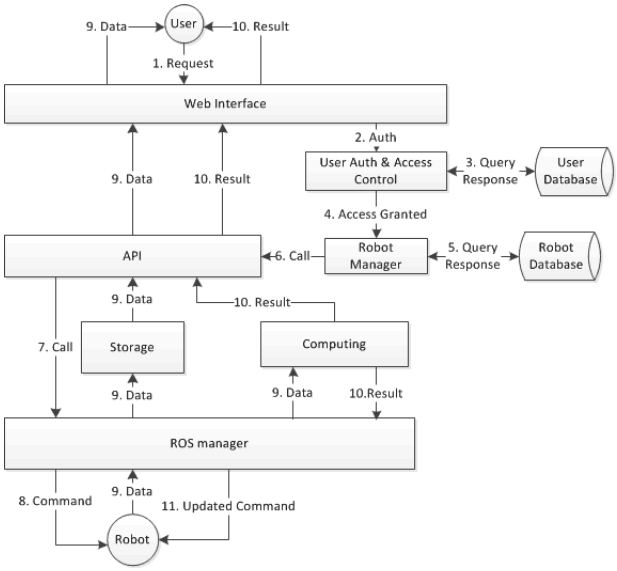

Fig. 4. Work flow when a user request a service from the web portal

the user has the right permission (4). The robot manager then checks out the robot database to make sure there exists a robot that can provide this server (5). Next, the API issues a message to the ROS manager (7). The ROS manager then sends command to the appropriate robot (8) and retrieves the requested data (9). The data is then passed all the way to the user. The computing module also performs necessary computations and returns the result to other modules and the user (10). Finally, new commands are sent by the ROS manager for further control of the robot.

4 Performance Evaluation

To evaluate the performance of offloading image processing in video tracking tasks to the Cloud, we designed a Cloud- enabled distributed robotics prototype, consisting of a remote robot network and a Cloud testbed in our datacenter. The robot network includes three SRV1 robots and an AR. Drone 2.0 flying robot. The flying robot collects image sequences with its on board camera and transfers to a local server or to the Cloud in real time. The Cloud runs a simple motion detection algorithm to track the motion of ground robots [8], [21]. We compare the performance in executing the video tracking task between virtual machine or physical machine by simulating up to twenty simultaneous requests of video signal processing and collects the performance data.

4.1 Experiment Setup

1) Cloud Testbed: The Cloud testbed consists of 16 servers in our data center, all using Xen Cloud Platform (XCP) 1.6 [22]. We choose Citrix XenCenter as our management software, which provides convenient management features for our experiment purpose. We can easily create, clone and move a virtual machine within the XenCenter. Live migrate, within the same pool or across different pools, is also an attractive feature. Each Cloud server is equipped with two Intel Xeon E5405 Quad-core processors at 2.0GHz, 32GB memory and 3TB storage. For this experiment, we only use a pool of four servers (Fig. 5). Fig. 6 shows the real-time monitor of the CPU, memory and network performance of the large instance in the experiment.

2) Authentication and Access Control: We have imple- mented part of the functionalities of the web interface in our framework. The administrator uses a web interface to monitor all robots' camera images. All users including the administra- tor are authenticated by a username password scheme before they can access any resource.

3) Machines under Comparison: To evaluate the perfor- mance differences among a local machine and virtual machines in the Cloud, we compared the local machine with three instances in the Cloud. Table 1 lists the specification of local machine and the three virtual machines instances we setup for performance comparison. We denote the VMs as small, medium and large instance according to their computing capacities. The small instance has comparative configuration with the local machine. The medium and large instances double the number of virtual CPUs and memory each time. All machines, physical or virtual, use a Ubuntu 12.04 LTS operating system.

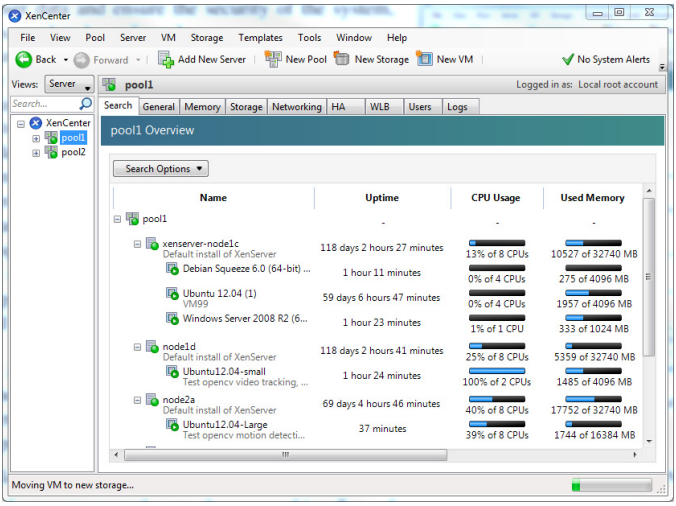

Fig. 5. The pool of Cloud servers

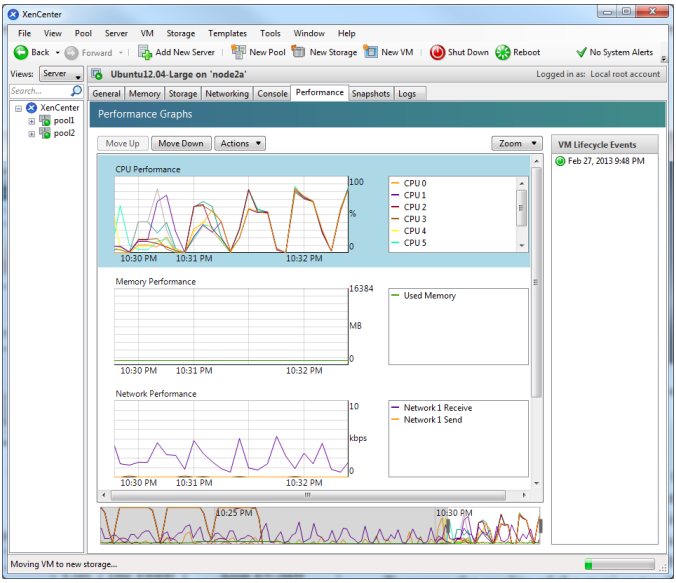

Fig. 6. VM performance monitoring

Table 1. Specification of Compared Machine

Item	**Local Workstation**	**VM1**	**VM2**	**VM3**
CPU Type	Core 2 Due E8400	Xeon E5-2609		
CPU Frequency (GHz)	3.0	2.4		
Number of Cores	2	2	4	8
Cache(MB)	6	10		
Memory(GB)	4	4	8	16

4) Task Analysis: The task of this simple prototype is to track the motion of ground robots using computing vision tech- nologies. The image processing procedure is computtionally expensive on a workstation or a laptop. We want to investigate the feasibility of offloading computing tasks to the Cloud and assess system scalability. To assure the quality of the motion tracking application, the frame per second rate must not fall under certain threshold so that video quality will not cause too much delay. The time to process a frame after captured by a camera consists of the time to transfer back and forth between a robot and the server as well as the time for the server to process this frame. Since the transfer time could vary under different network conditions, in this simple test, we only consider the frame per second that a server can process as the indicator of robust performance.

5) Algorithm: We tested a straightforward algorithm of target motion detection in this experiment. For each frame sent by the Robot, first the tracker finds the edge of every object using simple threshold algorithm. Then we find the contours in the edge frame, and find an approximate polygon for each contour and calculate a bounding box for it. Next we merge bounding boxes that are close to each other and finally get bounding boxes for all objects. Fig. 7 shows a frame of the result. The image processing algorithm for video tracking was written in C++ using openCV 2.4.4 libraries [23]. The same algorithm was implemented in all physical and virtual machines. The frame per second (fps) data was collected every a machine runs the processing algorithm. All fps values were average of 20 trials under the same condition for a number of simultaneous request.

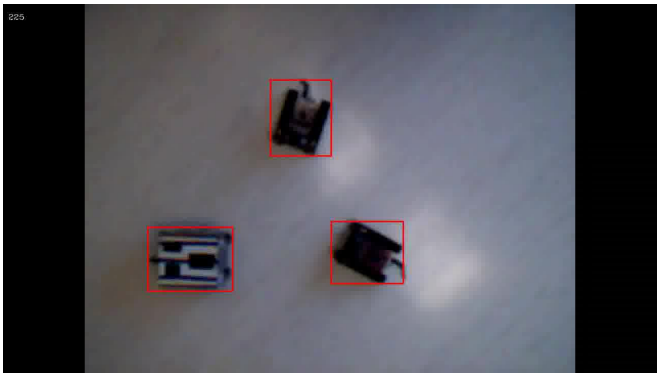

Fig. 7. A frame with tagged targets of the video tracking algorithm

4.2 Results

The results of our experiment are illustrated in Fig. 8. Over- all, the medium and large instances had higher performance that the others since they have more CPUs to process the images. All machines have almost the same fps for 1 to 3 requests, which means that handling 3 requests will not affect the CPU performance in any case of our

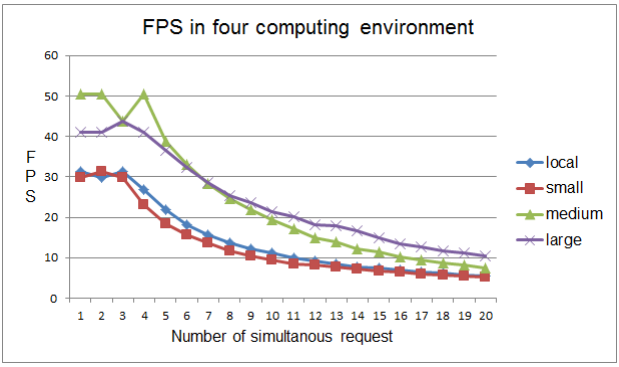

Fig. 8. Compare of four image processing environment: Local machine (local), small instance (small), medium instance (medium) and large instance (large)

experiment. The medium instance surprisingly had high fps than the large instance which has 8 virtual CPUs. One reason could be that the tracking algorithm was not optimized for parallel processing. A few requests at the same time will not take full advantage of the number of processors.

Started from 8 requests, the large instance had higher fps (25.27) than the medium one (24.58). Considering the general fps of the image processing algorithm, the large instance had fair performance in this experiment with over 10 fps for up to 20 requests. When there are more than 20 requests, the Cloud can easily assign more VMs to serve the users. An interesting phenomenon we observed in our experiment is that the amount of memory has no direct impact of the fps after 2 GB.

Because of the time consuming transfer of video back to the robot, we suggest that the server only return the coordinates of targets in each frame. This way the transfer delay from the Cloud to the robot is negligible, requiring only the transfer of image sequences or video streams through the Cloud. In our case, only 24B of data is needed for each frame. The Cloud architecture also enables the display of the results on a web page to the user.

5 Discussion and Conclusion

In this paper, we proposed a general architecture for a dis- tributed Cloud-enabled robotic information fusion system. A performance evaluation was investigated on an video tracking task for a robot network. Our results show that offloading computation to the Cloud is feasible and is especially benefi- cial when there are a lot of robot networks requesting image processing tasks.

We conducted all the experiments in our Cloud testbed consisting of 16 servers running on a Xen Cloud platform. A web portal with authentication module has been implemented and allows the user to communicate with the Cloud. Our current work is just a beginning of exploring the possibilities in combining robotic systems with Cloud computing to ac- complish more computationally intensive information fusion tasks.

There are still a lot of interesting performance and imple- mentation questions to be investigated in the future, such as seamless integration with the ROS system to control robots in real time, efficient scheduling of computing resources and dynamic bandwidth allocation according to the load of the system.

References

[1] Blasch, E., Bosse, E., Lambert, D.A.: High-Level Information Fusion Management and Systems Design. Artech House Publishers (2012)

[2] Blasch, E., Chen, Y., Chen, G., Shen, D., Kohler, R.: Information Fusion in a Cloud-Enabled Environment. In: Choi, B.-Y., Han, K., Song, S. (eds.), Springer Publishing (2013)

[3] Blasch, E., Kadar, I., Salerno, J., Kokar, M.M., Das, S., Powell, G.M., Corkill, D.D., Ruspini, E.H.: Issues and challenges in situation assessment (level 2 fusion). J. of Advances in Information Fusion 1(2), 122–139 (2006)

[4] Blasch, E., Seetharaman, G., Pal, K., Ling, H., Chen, G.: Wide- area motion imagery (wami) exploitation tools for enhanced situation awareness. In: IEEE Applied Imagery Pattern Recognition Workshop. IEEE (2012)

[5] Mazur, S., Blasch, E., Chen, Y., Skormin, V.: Mitigating cloud computing security risks using a self-monitoring defensive scheme. In: Proceedings of the 2011 IEEE National Aerospace and Electronics Conference (NAECON), pp. 39–45. IEEE (2011)

[6] Lee, K.-M., Zhou, Z., Blenis, R., Blasch, E.: Real-time vision-based tracking control of an unmanned vehicle. Mechatronics 5(8), 973–991 (1995)

[7] Shen, D., Blasch, E., Pham, K., Chen, G.: A clustering game based framework for image segmentation. In: 2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA), pp. 818–823. IEEE (2012)

[8] Mei, X., Ling, H., Wu, Y., Blasch, E., Bai, L.: Efficient minimum error bounded particle resampling l1 tracker with occlusion detection. IEEE Trans. on Image Processing (T-IP) (2013)

[9] Wu, Y., Cheng, J., Wang, J., Lu, H., Wang, J., Ling, H., Blasch, E., Bai, L.: Real-time probabilistic covariance tracking with efficient model update. IEEE Transactions on Image Processing 21(5), 2824–2837 (2012)

[10] IEEE society of robotics and automation's technical committee on networked robots, `http://www-users.cs.umn.edu/isler/tc/`

[11] Kuffner, J.J.: Cloud-enabled robots. In: IEEE-RAS International Conference on Humanoid Robotics, Nashville, TN (2010)

[12] Chen, Y., Du, Z., García-Acosta, M.: Robot as a service in cloud computing. In: 2010 Fifth IEEE International Symposium on Service Oriented System Engineering (SOSE), pp. 151–158. IEEE (2010)

[13] Agostinho, L., Olivi, L., Feliciano, G., Paolieri, F., Rodrigues, D., Cardozo, E., Guimaraes, E.: A cloud computing environment for supporting networked robotics applications. In: 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC), pp. 1110–1116. IEEE (2011)

[14] Kehoe, B., Matsukawa, A., Candido, S., Kuffner, J., Goldberg, K.: Cloud-based robot grasping with the google object recognition engine. In: IEEE International Conference on Robotics and Automation. IEEE (2013)

[15] Arumugam, R., Enti, V., Bingbing, L., Xiaojun, W., Baskaran, K., Kong, F.F., Kumar, A., Meng, K.D., Kit, G.W.: Davinci: A cloud computing framework for service robots. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 3084–3089 (May 2010)
[16] Goldberg, K., Kehoe, B.: Cloud robotics and automation: A survey of related work. EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2013-5 (2013)
[17] Hu, G., Tay, W., Wen, Y.: Cloud robotics: architecture, challenges and applications. IEEE Network 26(3), 21–28 (2012)
[18] Robot operating system, http://www.ros.org
[19] Hadoop, http://hadoop.apache.org/
[20] Fielding, R.T., Taylor, R.N.: Principled design of the modern web architecture. ACM Transactions on Internet Technology (TOIT) 2(2), 115–150 (2002)
[21] Ling, H., Bai, L., Blasch, E., Mei, X.: Robust infrared vehicle tracking across target pose change using l1 regularization. In: Int. Conf. on Info Fusion, vol. 1 (2010)
[22] Xen cloud platform, http://xen.org
[23] Opencv, http://www.opencv.org
[24] Mell, P., Grance, T.: The nist definition of cloud computing (draft). NIST special publication, vol. 800, p. 145 (2011)
[25] Wang, L., Liu, M., Meng, M., Siegwart, R.: Towards real-time multi- sensor information retrieval in cloud robotic system. In: 2012 IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), pp. 21–26. IEEE (2012)
[26] Mei, X., Ling, H., Wu, Y., Blasch, E., Bai, L.: Minimum error bounded efficient l1 tracker with occlusion detection. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1257–1264 (June 2011)