SCOUT: Data Center Monitoring System with Multiple Mobile Robots

Woong Choi, Ki-Woong Park, and Kyu Ho Park
Computer Engineering Research Laboratory
Korea Advanced Institute of Science and Technology
{wchoi, woongbak}@core.kaist.ac.kr, and kpark@ee.kaist.ac.kr

***Abstract*—Data center management is a critical issue as data center is growing at an enormous rate. As the data centers grow larger, data center management techniques have evolved from a simple alert system into a critical element that enables maximum availability, cost efficiency, and controllability. In this paper, we present a mobile robot-based data center management system called, *SCOUT* with a cost efficiency and scalability. It is an extended work of our previous work called *FalconEye* which is a vision-based monitoring system for the maintenance of data centers. Unlike our previous works, we advanced our system by deploying mobile robot platform so that it takes an advantage of monitorability and system controllability. *SCOUT* is comprised of the following three novel mechanisms: ground integrated NFC tags enabling for the robots to recognize the exact position for a system monitoring; efficient path construction algorithm for a low latency patrol with a fixed number of robots; and the robot path adaptation to maximize the robot utilization. To evaluate the feasibility and practicality of the *SCOUT* design, we built a prototype of the data center management system with a real robot platform and successfully deployed into iCubeCloud testbed. We expose the possibility of this system by using real monitoring prototype and evaluate algorithms by using *SCOUT* simulator.**

I. Introduction

Data center is going to be a center of the industry in these days. Most of mobile services already borrow the power from the data center in the name of cloud computing service such as GoogleDocs [1], DrobBox [2]. As these kinds of services are getting more popular and common, the enterprizes such as Google [3] and Facebook [4] enlarge the data centers all over the world. Therefore now we cannot ignore the cost for data center management and an economical management for the data center is the hottest issue in cloud computing services. Furthermore, as data centers are going larger, it is not easy work that some supervisors monitor and control all the servers of data center all the day long.

For efficient data center management, we set the definite items for management to argue with. The first item is the cost efficiency. Because the data center is growing larger, system monitoring facilities breaks out a lot of cost. For this reason, we consider that cost scalable solution even when the data center is growing hundreds or thousands of unit. The second item is server status monitoring and system control. In an ideal world, software and hardware would be deployed in a data center and simply work without any problems. However, reality poses a considerable challenge and involves resource-intensive tasks because of the need for around-the-clock monitoring and management to safeguard the increasingly complex environment of data centers. Data centers or server administrators must keep a watchful eye on the data and control ability to make sure the various systems remain stable and available [5]. For a small server group system with only a handful of servers, an administrator can just periodically do a manual check of the status. However, for data centers, where there are thousands of servers to monitor at any given time, the monitoring and alerting process pose a considerable challenge [6].

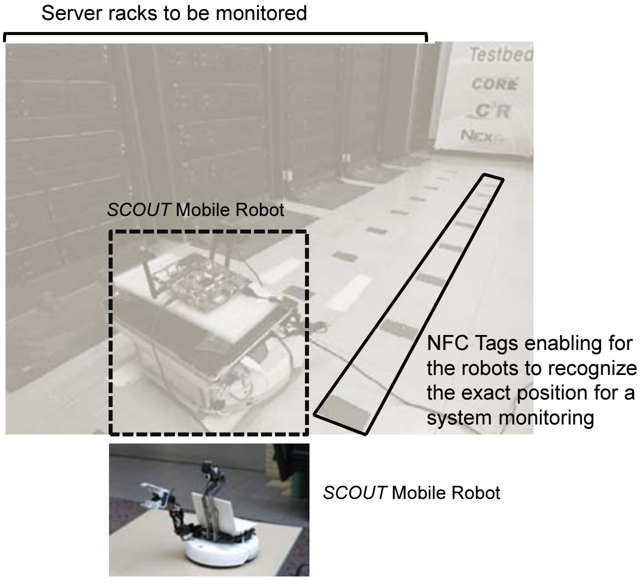

Server racks to be monitored
SCOUT Mobile Robot
NFC Tags enabling for the robots to recognize the exact position for a system monitoring
SCOUT Mobile Robot

Fig. 1. Overview of Data Center Monitoring System, SCOUT

Among the previous monitoring systems such as IPMI [7] and Nagios [8], we can find out a way for the items mentioned above. However, they cannot fully support the system control and monitoring simultaneously in a cost efficient manner. Even though some systems have control functions, they require a certain compatible interface equipped with baseboard management controller and additional costs and hardware, such as a protocol converter. As a remedy problem, our previous work, called *FalconEye* [9] is developed, which is a vision-based monitoring system in our *iCubeCloud* cloud testbed[10]. Even though the *FalconEye* gives accurate status extraction by vision based technic, if the data center growing larger, it should be installed all by supervisors and calibrated for all cameras. Furthermore, it cannot control the server machines which means everytime administrators manually should go into the server room that might be located remotely to turn on even a single server machine.

To settle down these challenges, we extended the previous work by presenting multi-robot based monitoring system, called *SCOUT*. It is based on the vision-based monitoring and also gives mobility to monitor not a fixed position but an area which a robot moves around. Because the robots can monitor multi-server-racks, we save the cost as the number of servers is growing larger. In addition, because vision and robotics technology support the function such as robot arms, robots can control the powers on server and more possibility are opened for controllability.

Fig. 1 shows the deployment of *SCOUT* as it monitors the *iCubeCloud* Data Center located on the campus of the Korea Advanced Institute of Science and Technology (KAIST) [10]. A mobile robot platform [11] equipped networked cameras are used to periodically collect images of the data center. The robot platforms then transmit a captured images to *FalconEye* monitoring system to extract information on the status of the data center. *SCOUT* is comprised of the following three novel mechanisms: ground integrated NFC [12] tags enabling for the robots to recognize the exact position for a system monitoring; efficient path construction algorithm for a low latency patrol with a fixed number of robots; and the robot path adaptation to maximize the robot utilization.

The remainder of the paper is organized as follows: In section II, we discuss background and related works about data center monitoring. In section III, we present the overall system design and real implementation of it. In section IV and V, we introduce associated issues about robot monitoring and evaluation of them. Finally, in section VI, we conclude this work.

II. Background and Related Works

A data center monitoring system that ensures maximum availability of data centers has been actively studied and developed in the research area of grid or cloud computing. The technology for monitoring the status of a data center goes back to the days of centralized mainframes and includes such practices as walking around with thermometers and relying on a data center administrator to investigate the spatial environment of the data center [13]. We have roughly two categories for data center monitoring which are software and hardware methods, here we look around some methods for these categories and analyze the each ones.

As software approach, Nagios [8] is well-used for data center monitoring system. It provides a system and network monitoring interface at the application layer. However by the reason of dependency from platform, it cannot control the system and it is possible to be affected from the operating system or hardware. In the worst case, even though the servers have severe problems, because it is installed in operating system, there is possibility not to get the present status information about the machine.

Differed from software approach, hardware approaches exist. One of the most popular hardware monitoring interfaces is IPMI (Intelligent Platform Management Interface) [7]. Many data centers adapt this interface for monitoring by the reason of independency from the operating system and hardware. It provides the functionality to turn on and off the server machine and shows CPU, network, storage status simultaneously. However because it is installed on each mainboard in server, the hugs cost for thousands of servers will be caused and because it is attached to mainboard, there is possibility to be affected from the hardware of server.

As totally separated hardware approach, computer vision based server monitoring system which is called *FalconEye* [9] is presented. The external status of each server platform is indicated by blinking Light-emitting diodes (LEDs). Even though it is appropriate extracting the information, it does not provide controllability. Therefore we are not able to manipulate the power for the purpose of energy saving. And the *FalconEye* requires the calibration for each camera when it is installed. That could make annoying process for administrator if the data center has huge number of server machines. Moreover, it can be affected by obstacles or room structures that restrict view angle then more cameras are required for covering all area which cause lots of expenses.

As mentioned above, monitoring solutions for data center should satisfy monitoring the server status and the power control simultaneously for exact information extraction and energy saving. Moreover, they should guarantee the cost efficiency in the trends of tremendous data center scales. In this paper, we introduce robot monitoring system which takes satisfaction in monitor/control functionality and economic feasibility for expanding-on data center

III. Overview of SCOUT

A. Overall Architecture

Fig. 2 shows the overall architecture of proposed system. The key members of this system are as follows. We have multiple robots which patrol and monitor the whole servers in the data center with image processing skills. And we have administrator monitor server which enables human administrators to collect the information of whole data center monitoring status and locations of robots from each robot by using wireless communication. And administrator is also able to control the servers easily by the robots via wireless communication. To provide the monitoring scenario for data center with multiple robots, we assume that we have well-defined patrol path.

According to this path, a group of robots patrols all the servers in data center and report the data to the administrator monitor server using the wireless communication. And through control message from the supervisor monitor via wireless communication, administrator monitor server itself can steer the multiple robots for the efficient monitoring system based on the whole context of the robots and data center or human

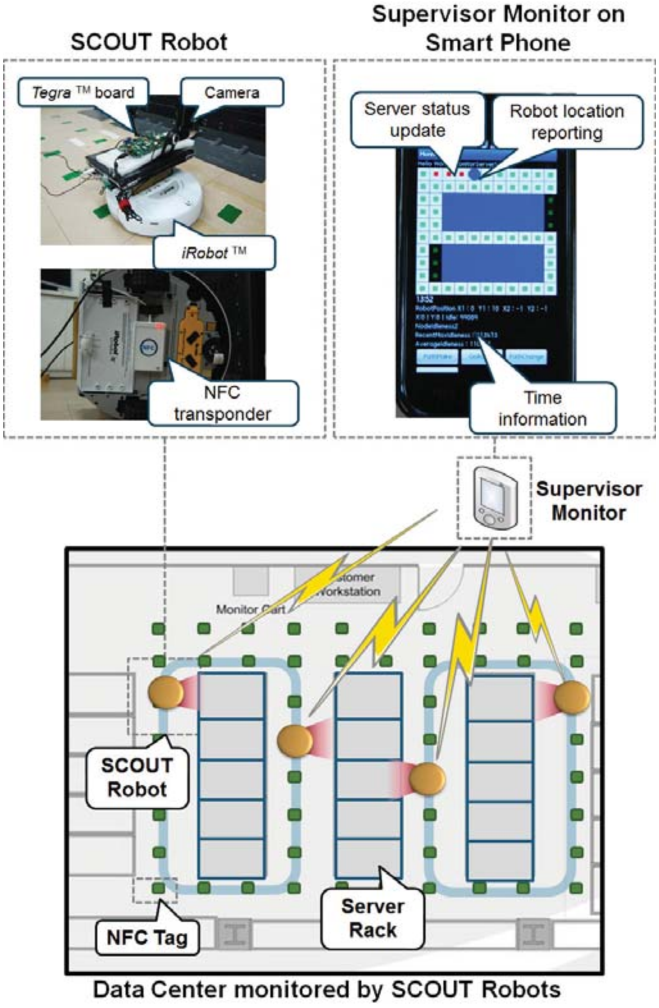


Fig. 2. Overall Architecture of SCOUT

supervisors are able to modify the direction of robots intentionally.

To achieve this scenario working well, we have to think about the associated issues. The first issue is accurate location detection. A robot should know the exact location to extract the information of servers which means a robot has ability to detect in server size unit. The second issue is efficient path construction. Even though we have same number of robots, according to patrol path, the server monitoring latency varies a lot. The last issue is path adaptation for dynamic server power allocation. When the servers change the power status dynamically for energy saving, if multiple robots change the path for current used server then more frequently monitor the servers without waste of robot utilization due to unemployed robots. In section IV, we introduce these associate issues in detail and suggest solutions for these issues.

B. Real Implementation in iCubeCloud Testbed

To examine the effectiveness of this system, we really implement the robot monitoring system in *iCubeCloud* testbed[10] in KAIST. Two iRobot [11] Create machine with tegra 250 board operated by android platform are used for monitoring robot. For localization solution, we apply the NFC tags on the floor and simple NFC transponder under the robot which readily write and read the NFC tags. As a administrator monitor server, a smart phone is adopted. Through simple android monitoring program and packet transfer, we can confirm the location of the robot, status of the server machines and late and time information of monitoring on the display of the phone.

Except for mechanical robot arm implementation, through this work, we can confirm the system possibility about accurate localization and path construction algorithm. Vision based technic which is used in *FalconEye* is applied and mobility of robots is added for extension.

IV. Challenges to Realize Robot-based Data Center Monitoring System

A. Accurate Location Detection

For this issue, there might exist some methods to solve. This issue is digested to indoor localization problem. Widely used Assistant-GPS and Wi-Fi localization methods are not adaptable, because it cannot guarantee centimeter unit accuracy in indoor situations. On the other hands, we have accurate UWB (Ultra Wide Band) localization methods [14] in indoor situation. However it is not applied to our system due to high cost which is against our motive.

In this work, therefore, we have adapted the NFC tags and transponders which have ability to read/write and which are used familiarly in our lives. We setup the NFC tags regularly on the data center floor which enables the exact localization. And with the small transponder under the robot, the robot finds out the location by reading the location information on the tags.

Somehow if data center grows very large, lying tag on the floor and tagging information on each tag seem to be bothersome work. However in these days, because the tag textile and tagging machines are already developed [15], it is not able to be the serious problem.

B. Efficient Path Construction

When we monitor data center with fixed number of robots under limited space, efficient path has important meaning. That is because constructed path affect monitoring latency a lot. For this issue, some researches exist in the name of Traveling Salesman Problem (TSP) [16] and Multi-Agent Patrolling [17]. The output which we want to get from those researches for efficient path construction is lowest idleness and uniform idleness which is defined as the time interval in one location between visits by robots [18]. Before going further based on this idleness concept, we assume that we already have floor plan of the data center monitored and tag id of the location to be monitored. If we have floor plan and the floor regularly covered with NFC tags, we can extract the monitor point.

To find out efficient path in the situation where we have point information to be monitored, first we have to calculate the cost of the edge from one point to another point. For this work, we adopt Lee algorithm [19] which is used for edge cost calculation with grid based map. Based on this algorithm, we can calculate the shown path cost on the floor space. After we get the shown path from the map, the shown path cost and practical cost can be different because we have some practical parameters in application to real system. If we calculate the cost, we have to think three parameter, they are processing cost, movement cost and rotation cost. The processing cost is the cost when the robot monitors the specific server. If the point is the monitoring point then this cost will be zero, but if the robot in the monitoring point, it will have the appropriate cost. And movement cost is the cost when the robot moves straight. Lastly we have rotation cost which is brought out when we meet the other direction changes. If we have same coordinate change then we do not have rotation change. However if we have other coordinate change in a certain point, then we recognize the rotation change and we calculate the rotation cost. In the real implementation and simulation, we always save the previous direction and next direction and if there is a change, we apply the rotation cost at that point.

After acquiring the point-to-point edge cost, by applying the solutions of TSP [16] or Multi-Agent Patrolling [17], we construct the path. There are normally two ways to construct the path efficiently for the multi robots. The first one is cyclic based strategy and next one is partition based strategy. These are well-known strategy as already proven by some researchers[18]. Because the partition based path is obtained by cutting the cyclic path, we first struggle with a single cyclic path with multiple points. Usually the TSP solution gives the cyclic path to whole point and it gives shortest path to circulate. And it is well known that comparing edges over 15 points is NP-complete problem and many heuristic method and other approaches are developed. After obtaining the whole path, then we can get most expensive cost path from the cyclic path. Then by cutting the path and constructing the partitioned path according to the number of robots, we can construct partition based on the path. As the number of robots increase, the complexity of partition and whole cost can be differed, therefore to get the efficient path cost, we have to simulate it case by case in our simulator. Then we can get the most efficient path.

Because some data centers have regular formation (monitoring points are gathered together shaping groups) in most of cases, as a one of heuristic method, we insist that grouping based on the proximity provides short idleness in most cases. It lessens the returning back cost and overlap path cost, which breaks out the contention. Among the heuristic methods, the proximity grouping heuristics is well applied due to form of data center, not for all cases.

C. Robot Path Adaptation

Cost for power and cooling over server's lifetime is not negligible in server management cost [20]. Therefore in the most of data center, they apply dynamic power allocation for energy saving. This addresses that the number of being-used-server changes dynamically. In this situation, if we apply robots, we need some path changes for the effective monitoring. In other words, when such a dynamic change of the server usage, robot should be adapted the path for maximum robot utilization. In this work, we call that scheme "Robot Path Adaptation" which means that robots should follow the change of server status. This adaptation makes the robots more frequently monitor the servers without waste of robot utilization due to unemployed robots. The sequence of the robot path adaptation is achieved by Algorithm 1.

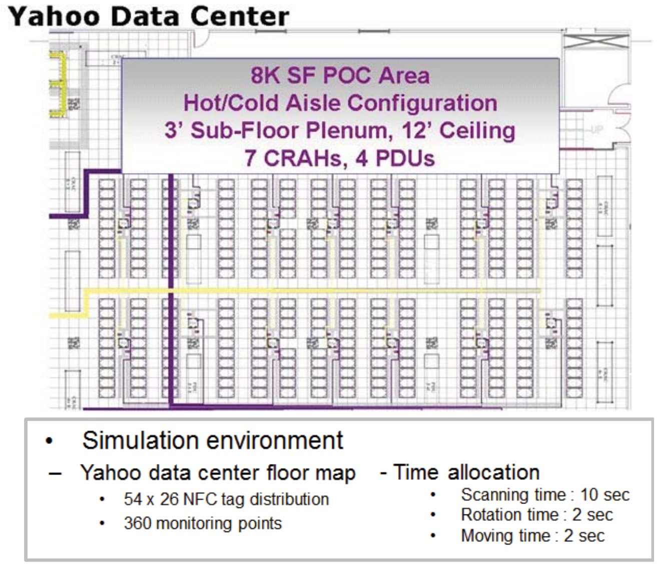
Algorithm 1: SCOUT Robot path adaptation scheme

```
Input: Changed monitor path, current status of SCOUT
       robots
Output: Optimized paths of SCOUT robots
//$R$ : Set of robots in a data center;
//$R_{\text{Off}}$ : Set of $R$ on *off-the-path*;
//$R_{\text{On}}$ : Set of $R$ on *on-the-path*;
read the updated points to be monitored;
foreach $R$ do
    //separate $R$ into $R_{\text{On}}$ or $R_{\text{Off}}$;
    if $R \exists \in$ *off-the-path* then
        $R_{\text{Off}} \leftarrow R$;
    end
    else if $R \exists \in$ *on-the-path* then
        $R_{\text{On}} \leftarrow R$;
    end
end
//Patrolling/nearestpoint search: $R_{\text{On}}$ keeps patrolling, $R_{\text{Off}}$
  searches nearest point;
foreach $R_{\mathit{Off}}$ do
    searches *nearest point*;
end
//Interval adjustment: $R_{\text{On}}$ are in the on path, the *interval*
  between each robots are adjusted;
derive *interval* time;
adjust the *interval* between $R$;
```

In the first step, the number of being-used-server is changed and reported. When the amount of request for the server machines decrease, the selected servers are turned down and change of monitoring point is reported. On the other hands, because of explosive demand for the computing resources, the servers should be turned on then monitoring points are expanded and also the number of points should be reported. In this way, changes of monitoring points are reported to administrator monitor server.

In the second step, administrator monitor server constructs the new efficient path according to the reported monitoring point. This works is done in accordance with the process of efficient path construction in section IV. Then classify the robots whether they are on the way of new path (on-path robot) or not (off-path robot).

Next, in the third step, the robots adapt the path according


Yahoo Data Center
8K SF POC Area
Hot/Cold Aisle Configuration
3' Sub-Floor Plenum, 12' Ceiling
7 CRAHs, 4 PDUs

- Simulation environment
  - Yahoo data center floor map
    - 54 x 26 NFC tag distribution
    - 360 monitoring points
  - Time allocation
    - Scanning time : 10 sec
    - Rotation time : 2 sec
    - Moving time : 2 sec

Fig. 3. Yahoo Data Center Floorplan for Simulation Environment of SCOUT(image reference [21])

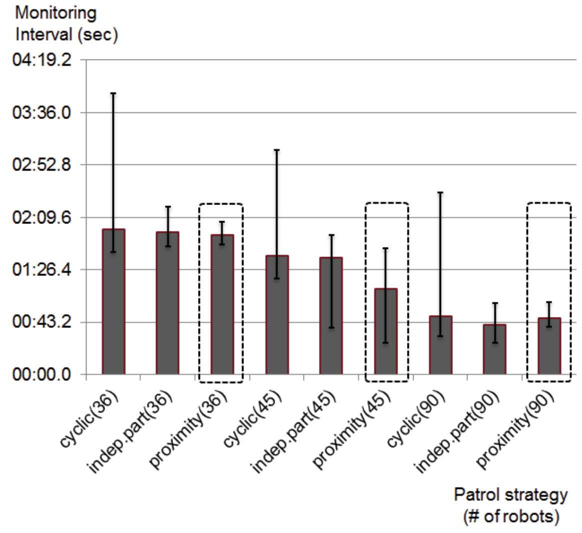


Fig. 4. Effect of Three Strategy for Efficient Path Construction

to their locations. If a robot is on-path robot, it will keep patrolling the new path as it did. Otherwise, the robot should find out the nearest point to join the new path. It can be obtained by calculating the path cost from robot position to all positions in the new path. If the minimum cost is selected then we make the intermediate path for each off-path robot. After arriving the nearest point, the robot joins the new way as other on-path robot do.

At last, the robot interval is adjusted. Since the robots are joined irregularly without the uniformity, this makes worst idleness case. In this situation, we need to adjust the interval between each robot. After all robots are joined in the new path, the administrator monitor server calculates the average interval between robots. This value can be obtained when we divide the whole path cost by robot number. And then the administrator monitor server adjusts the interval by slow down or boost up the robots.

V. Evaluation

A. Simulation environment

Fig. 3 shows the floor plan which is used in Yahoo Data Center [21] to simulate algorithms. We assume that we have 54 by 26 NFC tag distribution and 360 monitoring point. For robot processing time allocation, we set scanning time as 10 seconds, rotation time as 2 seconds and moving time as 2 seconds. Among three challenges to realize robot-based data center monitoring system, because the accurate localization issue is working well in present data center using by NFC tags and transponder. Therefore in this section, we examine the efficient path construction result and robot adaptation effect by simulator.

B. Effect of Proximity Grouping

Fig. 4 shows the result of the paths constructed by three different strategies. The cyclic strategy makes a single continuous path and on the path, multiple robots circulate the path with uniform interval. As one of the partition strategy path, independent partition makes a single partition for each robot and each robot circulate the path to which it belongs. At last, the proximity grouping makes the circular path by grouping adjacent independent partitions. The number in the blanket is the number of robot used in this simulation.

In the graph, the x-axis presents the patrol strategy with number of robots and the y-axis presents the monitoring interval which is equal to idleness. The less value the graph has, the better performance the strategy shows. As shown in Fig. 4, proximity grouping provides the most balanced result amount three. Because the proximity grouping minimize the return back cost and overlap cost, it cause the good performance with given number of robots. However proximity grouping strategy might cause bad performance when the edge distances vary a lot or formation of data center is not regular. The reason why the cyclic path has the worst case monitoring interval is that this strategy has long return back edge which bring about the worst monitoring interval.

C. Robot Path Adaptation

Fig. 5 shows the effect of robot adaptation when active server rate is going down. The robot path adaptation causes the robots more frequently to monitor the servers by getting rid of waste of robot utilization due to unemployed robots. As you see in the graph, when we do not apply the robot path adaptation, the monitoring interval is fixed. It means that unused robots waste the time stopping in a certain point which does not need any monitoring at all. However we do apply the robot path adaptation as shown in graph underneath, because there is not unemployed robot, monitoring interval decrease

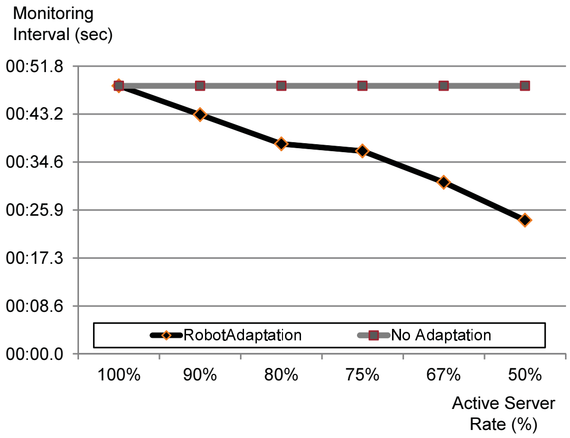


Fig. 5. Effect of Robot Path Adaptation

which means monitoring often according to the context. From the robot utilization point of view, it decrease the waste of robots and also increase the performance of the monitoring

VI. Conclusion

As we discussed until now, in this paper we propose the robot management system cost efficient, power controllable and also monitoring capability. First because a single robot can monitor the multiple racks that we do not need robots as much as the server number which saves the cost for expanded data center. Second, a robot can attach additional equipment such as robot arms and cameras. Therefore the robot can control the power on the server machine as well as monitor the status of servers. For robot monitoring system, we introduce three associated issues which are accurate location detection, efficient path construction and robot path adaptation. To monitor or control the servers, the robot should find out the exact location. Through the NFC localization method, we implement the centimeter unit accuracy with low cost. And by efficient path construction algorithm, we implement that low idleness patrol which enable efficient monitoring. Furthermore we introduce proximity grouping technic for efficient heuristic path construction algorithm for most formation of data center. At last, we present the robot path adaptation so that we maximize the robot utilization and minimize the latency of the monitoring period.

As the next step of this study, we are exploring more appropriate robot arm mechanism based one vision recognition for power control and calibration for LED location detection because whenever robot change the location the LED location should be extracted well for status monitoring and power control.

Acknowledgment

The authors would also like to NexR Co. Ltd for their help in conducting some of the experiments in this study. The work presented in this paper was supported by MKE (Ministry of Knowledge Economy, Republic of Korea), Project No. A1100-0901-1710.

References

[1] Google.com, "Google docs: Online documents, spreadsheets, presentations," Google Co. Ltd., http://docs.google.com, April 2011. [Online]. Available: http://docs.google.com

[2] DropBox, "Dropbox: Online backup, file sync, and data sharing," DropBox Co. Ltd., http://www.dropbox.com, April 2011. [Online]. Available: http://www.dropbox.com

[3] "Efficient data center summit," http://www.google.com/corporate /green/datacenters/summit.html, April 2009.

[4] FaceBook, "Facebook social network service," 2010. [Online]. Available: http://facebook.com

[5] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds," in *Proceedings of the 16th ACM conference on Computer and communications security*, ser. CCS '09. New York, NY, USA: ACM, 2009, pp. 199–212. [Online]. Available: http://doi.acm.org/10.1145/1653662.1653687

[6] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *SIGCOMM Comput. Commun. Rev.*, vol. 39, pp. 68–73, December 2008. [Online]. Available: http://doi.acm.org/10.1145/1496091.1496103

[7] "Intelligent platform management interface (ipmi) specification ver. 2.0," http://www.intel.com/design/servers/ipmi/spec.htm, June 2009.

[8] W. Barth, *Nagios: System and Network Monitoring*, 2nd ed. San Francisco, CA, USA: No Starch Press, 2008.

[9] K.-W. Park, W. Hwang, and K. H. Park, "FalconEye: Data Center Status Extraction using Vision-Transformation Techniques," in *IEEE 34th Annual Computer Software and Applications Conference Workshops*, ser. CloudApp '10, 2010, pp. 388–392.

[10] N. C. Ltd., "icube cloud computing and elastic-storage services," Mar 2011. [Online]. Available: http://testbed.icubecloud.com

[11] "iRobot," http://www.irobot.com/.

[12] N. Forum, "Introduction to nfc," 2010. [Online]. Available: http://www.nfc-forum.org/home

[13] C. C.Cowan, "Monitoring physical threats in the data center," APC Press, Tech. Rep., 2010.

[14] "Ultra-wideband technology for short-range, high-rate wireless communications, http://www.ieee.or.com/archive/uwb.pdf."

[15] R. Ueoka, A. Masuda, T. Murakami, and M. Hirose, "Rfid textile and map making system for large area positioning," in *Wearable Computers, 2009. ISWC '09. International Symposium on*, 2009, pp. 41 –44.

[16] G. Gutin, A. Punnen, A. Barvinok, E. K. Gimadi, and A. I. Serdyukov, "The traveling salesman problem and its variations," 2002. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.8.9120

[17] A. Almeida, G. Ramalho, H. Santana, P. Tedesco, T. Menezes, and V. Corruble, "Recent advances on multi-agent patrolling," 2004.

[18] Y. Chevaleyre, "Theoretical analysis of the multi-agent patrolling problem," *Intelligent Agent Technology, IEEE / WIC / ACM International Conference on*, vol. 0, pp. 302–308, 2004.

[19] F. Rubin, "The lee path connection algorithm," *IEEE Transactions on Computers*, vol. 23, pp. 907–914, 1974.

[20] D. S. Enrique Castro-Leon, Susmita Nayak, "Data Center Power and Thermal Management using Intel Data Center Manager Software Development Kit," White paper.

[21] Y. P. Release, "Yahoo data center floor plan," 2008. [Online]. Available: http://itknowledgeexchange.techtarget.com/data-center-facilities