International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682
Vol.5, No.10, October 2016 DOI:10.15693/ijaist/2016.v5i10.20-25

Determining Location of the Tasks of Robot by the Cloud

Amitava Kar
Asansol Engineering College, Asansol.

Ajoy K. Dutta & Subir K. Debnath
Production Engineering Department,
Jadavpur University, Kolkata

Abstract

We describe the application of cloud computing in the field of robotics where a robot is assigned the job of picking up of sticks lying on the floor and to keep them in the given spaces. The robot is connected to the cloud through a network. It receives the answer to the queries of the database residing in the cloud about the location of the sticks to be picked up and the spaces where they are to be kept. The cloud has a camera. It takes the snapshot of arena. In this paper, we describe how the cloud calculates the coordinate of the sticks as well as the spaces as compared to the initial positions of the robot from the snapshot. We also calculate the time complexity of the cloud for finding the coordinate of the sticks and the spaces by applying the algorithm of divide and conquer.

Keywords

Cloud computing, Robotics, Coordinates, Snapshots, Time Complexity, Divide and Conquer.

I. INTRODUCTION

The launching of the Internet in the 1990s led to the limited sharing of resources. The Cloud Computing is an application of such a resource sharing mainly software rather than a hardware. The hardware cannot be shared over the internet but load on the hardware can be reduced by offloading the complex calculations to the cloud. It relieves remote devices from the burden of carrying out extensive computations [1]. J. M. Cloud robotics uses the idea and includes the possibility of reducing the hardware requirement of a robot by storing the data in the cloud and getting them as and when required by querying them. It can also offload the complex calculations to the cloud and can query the result of the calculations as and when needed. This minimizes the hardware requirement [2].The robot simply receives help from the cloud [3] for doing its own work. R. Arumugam et al[4] spoke about the Cloud Computing framework for service robots.

On the platform of Cloud Robotics, we apply the algorithm of divide and conquer [5],[6], to find the location of the points where the jobs and spaces reside. J.L.Bentley et al [7] talked about divide and conquer algorithms in Multi-dimensional Space, but in this paper we restricted our research to 2D space only.

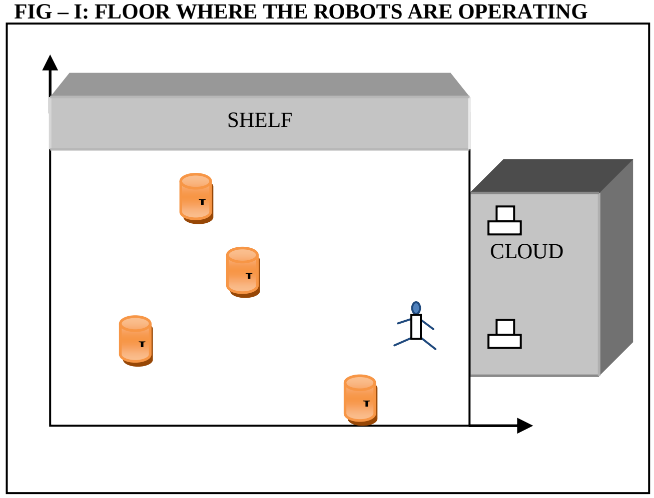
II. METHODOLOGY OF WORK

The robot is assigned to pick up the sticks and keep them in a given spaces. The robot is connected to the cloud with the help of the internet. The cloud has a camera fitted on it and it takes the snapshot of the arena. On the image the cloud draws grid lines and from them calculates the coordinate of the jobs and the sticks.

ASSUMPTIONS

There are four sticks to be kept in six given spaces. So some of the spaces may remain vacant.
The spaces are assumed to be collinear.
The sticks are scattered over the area.
Given below is the architecture of the floor where the robot is assigned to do the job.

FIG – I: FLOOR WHERE THE ROBOTS ARE OPERATING

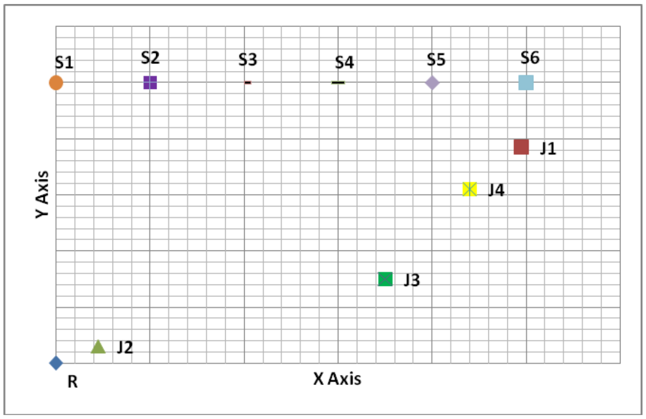


III. ALGORITHM TO FIND THE LOCATION

The Cloud has neither the coordinate of the jobs nor the spaces. Let us take that the initial position of R is (α, β). By shift of origin let us take it as the origin i.e. R(0,0).

From assumption 2, we know that the spaces are collinear. Let us suppose that the line passing through the spaces is parallel to the x-axis. So the line parallel to the x axis passing through R is the shifted x-axis i.e. X axis (say). The line perpendicular to the X axis passing through R is the shifted y axis i.e. Y axis (say).Vertical and horizontal grid lines are drawn throughout the area covering all the spaces and the jobs.

FIG – II: POSITION OF THE JOBS AND THE SPACES



Each side of square is taken as four square unit of distance. It is seen that if the arena is 120 X 120 then it covers all the spaces and the jobs.

As all the spaces are collinear to each other, we have to find out the Y-coordinate of one of them, the Y-coordinate of the others will be same. The X-coordinate of each of them is different.

Let us suppose that the X-coordinate of S1, S2, S3, S4, S5 and S6 be denoted by S1_X, S2_X, S3_X, S4_X, S5_X and S6_X. Now S1_X is on the Y-axis. So its X-coordinate is 0. The Cloud finds the S2_X, S3_X, S4_X, S5_X and S6_X by this XLoc function:

```
def Xloc(leftmost, pt, rightmost):
    mid = (leftmost + rightmost)/2
    if pt == mid:
        return mid
    elif pt > mid:
        return Xloc(mid, pt, rightmost)
    elif pt < mid:
    return XLoc(leftmost, pt, mid)
```

For S1_X:

S1_X is on the Y axis. So S1_X is 0.

For S2_X:

S2_X is put in pt of the function.

TABLE – I: S2_X is mapped with 20 by the Cloud

leftmost	rightmost	mid
0	120	60
0	60	30
0	30	15
15	30	23
15	23	19
19	23	21
19	21	20

For S3_X:

S3_X is put in pt of the function.

TABLE – II: S4_X is mapped with 40 by the Cloud

leftmost	rightmost	mid
0	120	60
0	60	30
30	60	45
30	45	38
38	45	41
38	41	39
39	41	40

For S4_X:

S4_X is put in pt of the function. Here mid is (0+120)/2 = 60. S4_X is mapped with 60 by the Cloud.

For S5_X:

S5_X is put in pt of the function.

TABLE – III: S5_X is mapped with 80 by the Cloud

leftmost	rightmost	mid
0	120	60
60	120	90
60	90	75
75	90	83
75	83	79
79	83	81
79	81	80

For S6_X:

S6_X is put in pt of the function.

TABLE – IV: S6_X is mapped with 80 by the Cloud

leftmost	rightmost	mid
0	120	60
60	120	90
90	120	105
90	105	98
98	105	101
98	101	99
99	101	100

Let us suppose that the X-coordinate of J1, J2, J3, and J4 be denoted by J1_X, J2_X, J3_X, J4_X.

The Cloud finds the J1_X, J2_X, J3_X and J4_X by this XLoc function:

For J1_X:

J1_X is put in pt of the function.

TABLE – V: J1_X is mapped with 99 by the Cloud

leftmost	rightmost	mid
0	120	60
60	120	90
90	120	105
90	105	98
98	105	101
98	101	99

For J2_X:

J2_X is put in pt of the function.

TABLE – VI: J2_X is mapped with 9 by the Cloud

leftmost	rightmost	mid
0	120	60
0	60	30
0	30	15
0	15	8
8	15	11
8	11	9

For J3_X:

J3_X is put in pt of the function.

TABLE – VII: J3_X is mapped with 70 by the Cloud

leftmost	rightmost	mid
0	120	60
60	120	90
60	90	75
60	75	68
68	75	71
68	71	69
69	71	70

For J4_X:

J4_X is put in pt of the function.

TABLE – VIII: J4_X is mapped with 88 by the Cloud

leftmost	rightmost	mid
0	120	60
60	120	90
60	90	75
75	90	83
75	83	79
79	83	81
79	81	80

The Cloud finds the Y-Location by the function:

```
def Yloc(top, pt, down):
        mid = (top + down)/2
        if pt == mid:
                return mid
        elif pt > mid:
                return Yloc(mid, pt, down)
        elif pt < mid:
                return YLoc(top, pt, mid)
```

For S1_Y, S2_Y, S3_Y, S4_Y, S5_Y and S6_Y are all equal. So we put any one in the pt of the function.

TABLE – IX: S1_Y is mapped with 100 by the Cloud

top	down	mid
0	120	60
60	120	90
90	120	105
90	105	98
98	105	101
98	101	99
99	101	100

So S1_Y, S2_Y, S3_Y, S4_Y, S5_Y and S6_Y are all equal to 100.

For J1_X:

J1_X is put in pt of the function.

TABLE – X: J1_Y is mapped with 77 by the Cloud

top	down	mid
0	120	60
60	120	90
60	90	75
75	90	83
75	83	79
75	79	77

For J2_X:

J2_X is put in pt of the function.

TABLE – XI: J2_Y is mapped with 6 by the Cloud

top	down	mid
0	120	60
0	60	30
0	30	15
0	15	8
0	8	4
4	8	6

For J3_X:

J3_X is put in pt of the function.

TABLE – XII: J3_Y is mapped with 30 by the Cloud

top	down	mid
0	120	60
0	60	30

For J4_X:

J4_X is put in pt of the function.

TABLE – XIII: J4_Y is mapped with 62 by the Cloud

top	down	mid
0	120	60
60	120	90
60	90	75
60	75	68
60	68	64
60	64	62

All the points thus obtained are tabulated below:

TABLE – XIV:

Objects	X	Y
R	0	0
J1	99	77
J2	9	6
J3	70	30
J4	88	62
S1	0	100
S2	20	100
S3	40	100
S4	60	100
S5	80	100
S6	100	100

IV. COMPLEXITIES OF THE ALGORITHMS

First we analyze Xloc. The terminating condition is 'pt = mid'. This can be stated as T(0) or T(1) and will be O(1) or 1. The algorithm runs with 'divide and conquer' technique. It divides the data into half and in the next step into further half and so on. Therefore, for each point time taken for locating its X is

$$T(m) = 1 + T(m/2)$$
$$= 1 + [1 + T(m/2^2)] = 1 + 1 + T(m/2^2]$$
$$\ldots..$$
$$\ldots..$$
$$= 1 + 1 + \ldots. + 1 + T(m/2^k) = k + T(m/2^k)$$

Now, let $m = 2^k \Rightarrow k = \log_2 m$

$\Rightarrow$ $m/2^k = 1$

$T(m) = \log_2 m + T(1) = \log_2 m + 1 \approx O(\log_2 m)$

where m is the number of points it scans for the first iteration. Here m=120.

Now, for Yloc the analysis is also similar. The terminating condition is point = mid. This can be stated as T(0) or T(1) and will be O(1) or 1. The algorithm runs with 'divide and conquer' technique. It divides the data into half and in the next step into further half and so on. Therefore, for each point time taken for locating its Y is

$T(n) = 1 + T(n/2)$

$= 1 + [1 + T(n/2^2)] = 1 + 1 + T(n/2^2]$

…..

…..

$= 1 + 1 + \ldots. + 1 + T(n/2^k) = k + T(n/2^k)$

Now, let $n = 2^k \Rightarrow k = \log_2 n$

$\Rightarrow$ $n/2^k = 1$

$T(n) = \log_2 n + T(1) = \log_2 n + 1 \approx O(\log_2 n)$

where n is the number of points it scans for the first recursion. Here n=120.

Total Time taken for each point is $O(\log_2 m) + O(\log_2 n) = O(\log_2 mn)$.

The generalization states that if there are p jobs to be kept at q spaces where p <= q, then total time taken for locating the X and the Y coordinates of all the points is $(p+q)*O(\log_2 mn) \approx O(\log_2 mn) \approx O(\log_2 n)$ [if $m \approx n$].

V. CONCLUSION

The idea of the research was to locate the jobs and the spaces by their X and Y value on the image such that through some transformation the original X and Y value of the jobs and the spaces can be found and at the same time the time complexity of the search is reduced by applying the divide and conquer mechanism. The location of the points otherwise would have to be done by raster scanning i.e. scanning from left to right of the whole arena. It would have involved scanning done by one loop for X-axis inside another for Y-axis. The time complexity would have been $O(n^2)$ which is sufficiently greater than $O(\log_2 n)$, where n is the number of points scanned. So this reduces the time complexity of the search making the search more efficient.

REFERENCES

[1] P. Mell, T. Grance, The NIST Definition of Cloud Computing, National Institute of Standards and Technology Special Publication 800-145, 2011.

[2] E. Guizzo, Robots with Their Heads in the Clouds, IEEE Spectrum, March 2011.

[3] E. Guizzo, Cloud Robotics: Connected to the Cloud Robots Get Smarter, IEEE Spectrum, January 2011.

[4] R. Arumugam, V. R. Enti, L. Bingbing, W. Xiaojun, K. Baskaran, F. F. Kong, A. S. Kumar, K. D. Meng, G. W. Kit, M. Rakotondrabe, and I. Ivan, "Davinci: A cloud computing framework for service robots."International Conference on Robotics and Automation, 2010, pp. 3084–3089.

[5] Lecture Notes for Algorithm Analysis and Design, Sandeep Sen, Department of Computer Science and Engineering, IIT Delhi, New Delhi 110016, India, November 6, 2013.

[6] Mark de Berg, Otfried Cheong, Marc van Kreveld, Mark Overmars, Computational Geometry, Algorithms and Applications, Third Edition, Springer, 1998.

[7] J.L.Bentley and M. 1. Shames. Divide-and-conquer in multidimensional space. in Proc. Xth Annual Svmposium cm Thror~ of' Computirt~. 1976. pp. 220-230. ACM. New York, 1976.

Authors Profile

	**Dr. Ajoy Kumar Dutta** is currently a Professor in the Department of Production Engineering, Jadavpur University, INDIA. He received his B. E. & M. E. degrees in Electronics & Tele-communication Engg from Jadavpur University in 1983 & 1985 respectively, and Ph. D. (Engg) degree in the area of Robotics from Jadavpur University in 1991. His Field of Specialization and Research Area are Robotics, Sensors, Computer Vision, Microprocessor Applications, and Mechatronics. He has teaching & research experience of 31 years.
	**Mr. Subir Kumar Debnath** is currently an Associate Professor in the Department of Production Engineering, Jadavpur University, INDIA. He received his B. E. degree in Mechanical Engg from Jadavpur University in 1982 & M. Tech in Mechanical Engg in 1984 from I.I.T.- Kharagpur, INDIA. His Field of Specialization and Research Area are Robotics, Sensors, Computer Vision, CNC Machines and Automation. He has teaching & research experience of 31 years.
	**Mr. Amitava Kar** is currently an Assistant Professor in the Department of Computer Application, Asansol Engineering College. He received his B.Sc. and M.Sc. degrees in Economics in the year 1995 and 1998 from Calcutta University. He received his MCA degree from WBUT in 2004 and M.Tech(O.R.) from NIT, Durgapur in 2008. His Field of Specialization and Research Area are Cloud Robotics, Computer Algorithms, Computational Geometry and Big Data Algorithms. He has an industrial experience of 3 years and a teaching experience of 12 years.