Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 15, Issue. 1, No.4 : 2024
ISSN : **1906-9685**

LOWERING THE COST OF PATH FINDING WITH CLOUD COMPUTING AND SELECTING THE IDEAL PATH USING A* ALGORITHM

Dr.S. Chithra Devi, Assistant Professor(CS), School of Computer Studies, RVS College of Arts and Science,Sulur ,Coimbatore-641402

ABSTRACT:
Cloud computing is now becoming very popular in today's business world. Sustainability is being offered as a defined service parameter in public clouds. For example, Amazon stated that its EC2 customers can count on 99.95% uptime. Users might find it too costly or insufficient to meet their needs for consistency. A service-oriented architecture, lower overhead, greater adaptability, a lower total cost of ownership, on-demand services, and many other benefits are implied by cloud computing. It offers storage, software, computing, and data access services without requiring end users to be aware of the location and setup of the system that provides them. The lack of funding prevents small enterprises from using new technologies. However, they can simply obtain their services for a small fee by leveraging the cloud. Allocating cloud resources should not only meet user requirements, but also minimize costs by identifying the best route. An algorithm to find the shortest route between the starting and end states is the A * algorithm. This is an illustration of a path finding method called best-first search. It can locate the most suitable solution feasible in addition to finding one quickly. Therefore, the present paper proposes an idea to manage cloud computing using A* in finding the best feasible route.

I.INTRODUCTION:

Pay-as-you-go model-based utility services are offered by cloud computing. On the cloud, users can host a variety of applications, from simple applications to heavy scientific applications. These programs are made available online as services. Customers of cloud services are no longer concerned about the expenses related to over- or inadequate resources. Without considering the cost, many researchers concentrated on hosting high-performance apps on Clouds[13]. The requirement to optimize the cost of data center resources is growing as a result of rising costs. Thus, it's important to arrange cloud resources in a way that minimizes costs while still meeting user requirements through optimal path finding.

The cloud computing paradigm, which promises dependable services provided through next-generation data centers, is the most recent grid paradigm to emerge. Customers will have on-demand access to apps and data from a "Cloud" from anywhere in the globe. Put differently, the Cloud seems to serve as a single point of access for all of the needs that users have in terms of computing.

Performance analysis in cloud computing needs to be defined by dynamic data collection when run-time performance issues found, data reduction when there is a need to monitor large data, low-cost data capture (since instrumentation and profiling overhead may affect the application's performance), and adaptability to heterogeneous environments. The utility services like gas, electricity, and water are necessary for our daily lives to function smoothly, Similarly, Cloud Computing [2] is also trending toward providing various IT resources on demand on a pay per user basis in terms of computing.

The IT resources comprise various computational services for shipping status and tax calculation. Offering these computational services to all customers, from single users to huge enterprises that contract out their whole IT infrastructure to external data centers, is the aim of cloud computing.

Actually, the cloud has provided a fresh method for turning IT into a comprehensive, worldwide service for end consumers.
Services can be reused and programmed. Online access to them is available from any location. Applications developed with this paradigm will operate on several sites, gathering data from each, merging it, and distributing it to any device on the universe in a personalized manner.
The current work suggests utilizing A* to control Cloud Computing. This work highlights the result of A* to Cloud Computing architecture, since A* uses algorithms to achieve higher performance (in terms of time).

II.PROBLEM DEFINITION:
Numerous studies were conducted in an effort to identify the optimal route. However, they have a lot of trouble figuring out the quickest and most efficient route to get there. Finding the cheapest way to travel is also a difficult issue. Therefore, the cost and shortest path was the main problem in reaching the destination. Previously, the Ant Colony approach was used by the researcher to solve problems in determining the optimal path. This algorithm processed based on how ants navigate to get from their colony to a food source, to find the best path through a network. Ants (at first) travel aimlessly before returning to their colony while leaving pheromone trails in search of food. But the pheromone trail begins to fade, lessening its potency of attraction. The pheromones have a longer time to disappear the longer it takes an ant to go down the trail and back. In contrast, a short path is traveled over more often than a longer one, which causes the pheromone density to rise on shorter paths.
Then, there would be limitations on the solution space research. As a result, when an ant identifies a good—that is, a short—route from the colony to a food source, other ants are more inclined to follow it. Eventually, positive feedback causes all the ants to follow the same way. It becomes some difficult to choose the best path of decision always.
The cost varies based on the structure as well. Different decisions, though, can lead to various perspectives on and approaches to the same issue. Although there are various algorithms utilized for calculation, the heuristic method known as the A* algorithm was used to find the Shortest Path and Cloud Computing to reduce the Cost.

III.OBJECTIVE:
When determining the best course of action using the A* algorithm, numerous objectives must be taken into account. They are listed below:
- Drawing a traversable path between nodes that is efficient.
- To provide a passage free of traffic.
- Improving performance in relation to time.
- Locates the least-cost route.
- To free up memory.
- To offer a quicker and more precise way to choose the best path of decision.

IV.SCOPE OF THE RESEARCH:
The main focus of this research was to determine the optimal path with least cost in Cloud applications. It is very difficult to estimate the cost from starting node to current, as well as from current node to destination node. But the best way to reach this research is using one of the heuristic technique known as A*. A* algorithm is to solve the optimal path problem under a variety of user-defined constraints. It is an optimal heuristic search. It guaranteed to find an optimal solution.

V.LITERATURE REVIEW:
The research by the author in [5] compares A* to various search algorithms and looks at how A-Star is being used in pathfinding. It also examines prospective advancements for the growth of A-Star in the future. The demands of the current pathfinding issues are too much for A* to handle. The issue only gets worse as grid size increases because other algorithms can achieve the same performance with less overhead. But when working with huge maps, A* can attain very quick timings with good

accuracy by utilizing creative changes like different types of heuristics or adding auxiliary components to the algorithm, all for a modest increase in overhead costs. Even if they are starting to show their age, enhanced algorithms built upon the traditional A* algorithm can easily keep up with the needs of contemporary pathfinding.

In order to reduce search redundancy in the majority of iterative search algorithms [9] introduced the Boundary Iterative-Deepening Depth-First Search (BIDDFS) algorithm, which repeats its search from the saved boundary location. The trial findings don't give as much specific information as the memory usage; they merely display the time required and the threshold, at which Dijkstra beats BIDDFS's time.

Two algorithms namely, Dijkstra and A* is compared in [8] to find the optimal path. Road networks and routing commonly use these two techniques. Comparing those two algorithms in order to solve the shortest path problem is the goal. It is be given that A* is more quickly than Dijkstra due to the heuristic value that was taken into account during the calculation, A* only scans the area in the direction of the destination, whereas Dijkstra searches by expanding out equally in all directions and typically ends up exploring a much larger area before the target is found, making it slower than A*. In this study, Dijkstra's and A* perform nearly equally when used to local or neighborhood maps; however, A* performs better when applied to a massive map.

The researcher in [3] combines the k-step look-ahead heuristic function with the A* search algorithm, then run a computational experiment to assess and contrast the outcomes on different-sized road networks. This novel heuristic significantly reduces runtime, especially for bigger networks, and that, as network size increases, a greater value of k is required to attain optimal efficiency.

Other earlier studies [6] directly compare Dijkstra's algorithm and A* on detailed road networks, but they don't alter any of the method's properties.

The article [4] implemented A* algorithm on Cloud Computing for resource allocation based on Mobile Internet-of-Things. The genetic algorithm's global search capability is added to the initial information allocation process in order to combine it with the Ant Colony algorithm and use it in the process of allocating cloud computing resources. The simulation findings demonstrate that the suggested approach can reduce the overall task execution time while significantly increasing user satisfaction and resource utilization. Because of the restricted amount of effective investigation time and energy, the suggested strategy still has limitations.

The QL_HEFT algorithm was improved in [7]. This study divides work among the processors more effectively and speeds up application completion. Computational findings show that the proposed technique outperforms the existing QL_HEFT algorithm in terms of speed. However, it still has an issue with communication and storage costs.

The author in [1] discussed about the difficulty of load balancing in a cloud environment is examined, and the shortcomings of the various load balancing techniques that have been employed in the past are methodically surveyed and examined. After a thorough examination, it was discovered that meta-heuristic approaches are the most appropriate. The meta-heuristic technique constantly aims to achieve the best possible outcome and is most suitable for determining the node's capacity. The design of the suggested work, ILOA_LB, uses a Bio-Inspired Lion Optimization Algorithm as a result of the analysis. Comparing the results to the metrics found, it is clear that it has higher Throughput, reduced Latency, quicker Response times, and enhanced Fault tolerance.

In [2] the researcher discussed a two different genetic and ant colony approaches to solve the classic computer science problem of shortest path. The researcher discussed the fundamentals of Ant Colony algorithms, Genetic algorithms, and the Shortest Path problem. The researcher also offers two potential strategies. The first technique measures its performance based on Packet Delay, Throughput, and Bandwidth consumption. The second method assist in concentrating research on areas of the search space that show promise.

Therefore, Pheromone builds up more quickly along the shorter channel around the obstruction. Ants go toward pathways with higher concentrations of pheromone, so finally they all follow the shorter path that avoids the obstruction.

Even though Ant-Colony gives Positive Feedback accounts for rapid discovery of good solutions, Efficient for Traveling Salesman Problem and similar problems and Can be used in dynamic applications (adapts to changes such as new distances, etc.), it has some drawbacks such as,

1. Theoretical analysis is difficult
2. Probability distribution changes by iteration
3. Research is experimental rather than theoretical
4. Time to convergence uncertain.

On the other hand, the Heuristic Algorithm A* will solve the Ant-Colony problem that was previously discussed. Furthermore, it is the responsibility of Cloud service providers to guarantee that data is transferred to their server at the lowest possible Cost and the shortest possible Distance. The most widely utilized option for path finding is A* since it is somewhat adaptable and applicable in a variety of situations.

VI.PROPOSED METHODOLOGY:

A heuristic in 1968 by Nils Nilsson to help it navigate an obstacle-filled area. This path-finding algorithm, known as A1, was a quicker variation of Dijkstra's algorithm, the most well-known formal method at the time for locating the shortest paths in graphs. Bertram Raphael dubbed the updated version A2 after he made several noteworthy enhancements to this method.

Subsequently, Peter E. Hart presented a case that proved A2, albeit with some modifications, to be the optimal algorithm for shortest paths. Next, Hart, Nilsson, and Raphael worked together to create a proof that, in certain scenarios, the updated A2 algorithm was the best for locating shortest pathways.

Hence, they gave the new algorithm in Kleene star syntax the name A*, The algorithm was originally described in 1968 by Peter Hart, Nils Nilsson, and Bertram Raphael of the Stanford Research Institute (now SRI International). It is an expansion of the 1959 algorithm created by Edsger Dijkstra. A* uses heuristics to obtain greater performance (in terms of time) [7].

- First, the propose algorithm estimates the cost of the path from the current node to the goal node and the path's cost from the start node to the present node.
- Once the user's location has been determined, we require the destination point, which we obtain from the dial pad with the user's selection. We can determine the shortest path between these two points if we have the location and destination information. We are able to quickly compute the path using any path finding algorithm since we save the locations and cells that are both available and unavailable in memory.

A* PATH FINDING ALGORITHM

1. Create a search graph G, consisting solely of the start node, n o . Put n o on a list called OPEN.
2. Create a list called CLOSED that is initially empty
3. If OPEN is empty, exit with failure.
4. Select the first node on OPEN, remove it from OPEN, and put it on CLOSED. Called this node n.
5. If n is a goal node, exit successfully with the solution obtained by tracing a path along the pointers from n to no in G. (The pointers define a search tree and are established in Step 7.)
6. Expand node n, generating the set M, of its successors that are not already ancestors of n in G. Install these members of M as successors of n in G.
7. Establish a pointer to n from each of those members of M that Ire not already in G (i.e., not already on either OPEN or CLOSED). Add these members of M to OPEN. For each member, m, of M that was already on OPEN or CLOSED, redirect its pointer to n if the best path to m found so far is through n. For each member of M already on CLOSED, redirect the pointers of each of its descendants in G so that they point backward along the best paths found so far to these descendants.

8. Reorder the list OPEN in order of increasing f values. (Ties among minimal f values are resolved in favor of the deepest node in the search tree.)
9. Go to Step 3.

LOCAL MINIMA PROBLEM:Lo

An algorithm is said to be in a local minima when it finds itself halted behind a barrier where it has no other place to go. The surrounding positions all have larger f-costs values than the current position, hence the algorithm cannot possibly escape from this situation because it has no other place to go. In the figure below, a local minima is depicted.

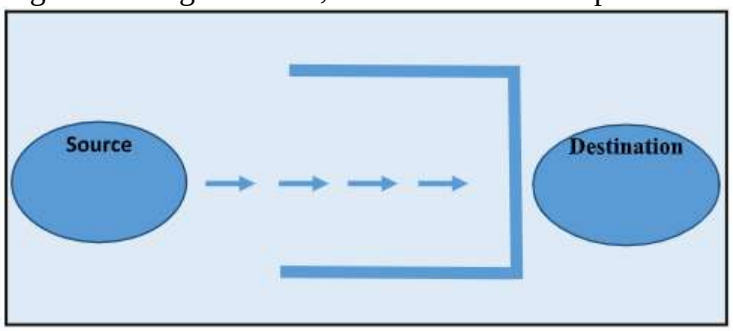

Figure :1 (Local Minima)

VII. RESULTS AND INFERENCES

Sample Number	Time(ms)	Path Length(Cells)
1	10.744	12
2	6.176	20

Table 1 Sample Simulation Data

SAMPLE 1

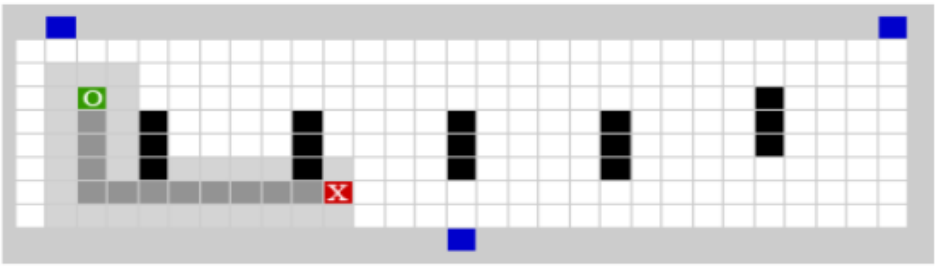

Figure 2 (Sample 1), 10.744 ms, 12 cells to go

SAMPLE 2

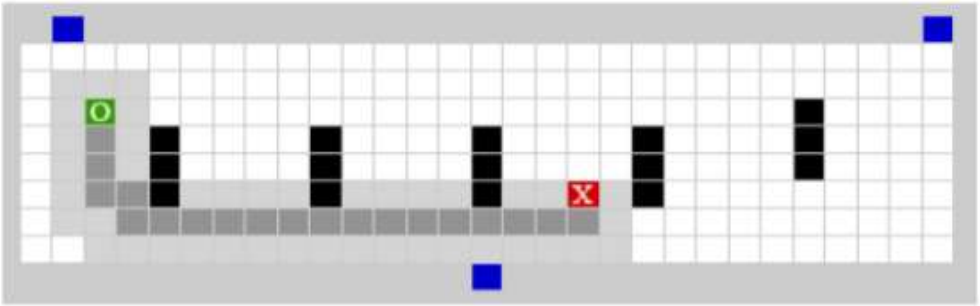

Figure 3 (Sample2) 6.176 ms, 20 cells to go

From these results the Cost Calculation Method of the first sample (Figure 2) gives the best result for finding the path in terms of time passed and the path length which will guide the user to walk on. By shortening the path, I can prevent to make unnecessary calculations and also can forward the user to the destination faster. In Table 1, sample data is summarized.

VIII. CONCLUSION:

In this paper problem of finding optimal paths in cloud computing has been addressed efficiently using A* algorithm. A penalty criterion has been used to find the best optimal paths for a desired source destination pair. It is evident from the results that the proposed solution is efficient enough to find the optimal paths in the given input. While searching for the best path it is desirable to find the most distinctly different paths.

IX. SCOPE OF FUTURE ENHANCEMENT:

Many ideas related to the efficient A* implementation be tried in future. Some of these ideas are discussed below.

1. To reduce the memory size by combining A* with some combinational algorithm.
2. Some heuristic method can be explored that is more suitable to calculate the abstract lighted graph
3. To overcome bottleneck of the binary heap while using multiple threads, the Lock-free or wait-free implementations of the binary heap could be done.

This work proposes 1) Finding optimal path and 2) Minimum cost. This work can also be extended to make the results for better quality in return for turn-around time. They are also supposed to be flexible in time and resources.

REFERENCES:

[1] Kaviarasan R , Balamurugan G ,Kalaiyarasan R , Venkata Ravindra Reddy Y ,"Effective load balancing approach in cloud computing using Inspired Lion Optimization Algorithm", Volume 6, December 2023, 100326

[2] Eman Drwish and Mohammed Wahed ,"Using Optimization Algorithms For Solving Shortest Path Problems", January 2022, Alfarama Journal of Basic & Applied Sciences , DOI:10.21608/ajbas.2021.90703.1062

[3] Kevin Y. Chen, An Improved A* Search Algorithm for Road Networks Using New Heuristic Estimation, 30 July, 2022, https://doi.org/10.48550/arXiv.2208.00312.

[4] Qiao Zhou, "Research on Optimization Algorithm of Cloud Computing Resource Allocation for Internet of Things Engineering Based on Improved Ant Colony Algorithm", Volume 2022 , https://doi.org/10.1155/2022/5632117

[5] Daniel Foead [a], Alifio Ghifari [a], Marchel Budi Kusuma [a], Novita Hanafiah [b], Eric Gunawan [b,] 19 February 2021, https://doi.org/10.1016/j.procs.2021.01.034, Volume 179, 2021, Pages 507-514,Elesvier

[6] J. Jain, "Shortest Path for Emergency Services: AComparative Analysis,"International Journal for Re-search in Applied Science and Engineering Technology,vol. 9, pp. 426–430, June 2021

[7] Zainab KashalanYeeser and Prof. Khaldun I. Arif, "An Efficient tasks scheduling using DAG for Cloud Computing", Turkish Journal of Computer and Mathematics Education, Vol.12 No.13 (2021), 6841 – 6857

[8] Dian Rachmawati1* and Lysander Gustin2*, "Analysis of Dijkstra's Algorithm and A* Algorithm in Shortest Path Problem" ,2020, Journal of Physics: Conference Series,1566 012061, doi:10.1088/1742-6596/1566/1/012061.

[9] Kai Li LIm, Kah Phooi Seng, Lee Seng Yeong. "Uninformed pathfinding: A new approach." Expert systems with applications 42(5):2722-2730,DOI:10.1016/j.eswa.2014.10.046.