1. **Abstract**

This poster presents AntFinder, a novel approach to the problem of discovering and ranking the characteristics of the data sources and query processing sites in a wide-area mobile distributed database system. We model the system as a graph with nodes representing data sources and query processing sites, some of which might be replicated. We introduce a heuristic technique inspired in Ant Colony Theory to dynamically discover, rank and catalog each data source or query-processing site. Our goal is to find possible paths to access the computational resources or data provided by the highest quality sites. We define quality in terms of performance, freshness and others. We present simulation results of the system using Java CSM and also a preliminary performance study designed to analyze the quality of paths found by the Ant Colony algorithm using only performance.

2. **Motivation**

Data integration scenarios, where we want to find which is the best data repository or processing site to use.

3. **Problem & Approach**

- Catalog is centralized and static.
- System is dynamic.
- Catalog changes continuously.
- Needs to be:
  - Manage in geocentralized fashion.
  - Changed dynamically as updates arrive.
  - Up-to-date as many updates as possible.

4. **Proposed Solution**

Our wide-area mobile distributed database system can be represented by a graph. Let G = (V, E) be a directed graph where V is a collection of sites and E is a collection of edges that represent connectivity between sites v, w belong to V. Nodes in V can be classified as data provider or query processing provider.

5. **ACO Overview**

- Ant Algorithms were inspired by the observation of real ant colonies.
- Ants are social insects.
- Ants have the ability to find shortest paths between food sources and their nest.
- Ants use indirect communication.

6. **System Architecture**

We choose NETTRAVELER as a database middleware system to use as the use-case environment. The Figure 5 depict the main components of our system.

7. **Quality Metrics**

We construct a framework to estimate the cost in the path over the graph. The cost might represent response time, resource source usage, last update time, or other metrics. Here we present one metric:

<table>
<thead>
<tr>
<th>Performance</th>
<th>Characteristics of a site:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Based on the computational capacity.</td>
</tr>
<tr>
<td></td>
<td>It is also called Quality of Service (QoS).</td>
</tr>
<tr>
<td></td>
<td>Provide a notion of speed or efficiency.</td>
</tr>
<tr>
<td></td>
<td>Can be measured base on:</td>
</tr>
<tr>
<td></td>
<td>Local resources: CPU time, disk bandwidth, waiting time for attention.</td>
</tr>
<tr>
<td></td>
<td>Network resources: data transfer costs.</td>
</tr>
<tr>
<td></td>
<td>We explore different models, and we implement the more complex, exponential weighting moving average.</td>
</tr>
</tbody>
</table>

8. **Algorithm**

The Ants visit the nodes and collect data information. Then they use it to rank the sites based on the selected metrics:

- Ant does the travel forward saving information about the situation, searching for a good path.
- After that, the ants do travel backward, placing the update in the pheromone trail (like real ants) and statistics. We illustrate the process in the following Fig 6.

9. **Experiments**

- Usability: Can our approach find optimal or near optimal solutions to the problem of finding a shortest path (v, w)? We implemented Java CSM programs to simulate the ACO framework.

10. **Conclusions**

- The ranking of data sources and query processing sites allow the system to improve its performance, including:
  - Quality of results.
  - Query optimization process.
  - Use of alternative techniques as ACO in path solution searching.
- We will add another features to NetTraveler System.

11. **References**