Abstract—ConvergeCast is one of group communication tasks in distributed systems: convergecast, broadcast, multicast and unicast. ConvergeCast algorithms allow concentrating network information for calculating of global parameters. When the ConvergeCast is required, information about the network itself such as routing table, network topology is not available. Distributed recursive wave is a way to construct algorithms that can be customized only by local parameters and functions. Distributed recursive wave then is suitable for distributed systems having network information dynamically changing without global information.

This article presents a new converge cast algorithm. This algorithm is experimentally implemented using Shell Script.

Index Terms—Distributed Converge Cast, Distributed Recursive Waves

1. INTRODUCTION
Converge cast is one of group communication tasks used when the communication network of distributed systems is installed or reconfigured. In this case, the global information of the system is not ready, so the protocol can only rely on the local information in each process. This feature is becoming more important for the continuously changing systems, depending on the objective and natural factors such as sensor networks and mobile networks.

Distributed recursive wave is a model enabling the construction of distributed algorithms with simplicity of the recursive algorithm, and allowing to adjust the implementation of recursive algorithms with local parameters. Recursive wave is based on the principle of echo, being made in two phases: forward phase and echo phase. In [1] a distributed recursive wave based broadcast algorithm uses only forward phase because of the nature of the broadcast algorithm. In [4,5] the authors have proved that distributed recursive wave is consistent in distributed systems with no global information.

This article presents the analysis and design of an algorithm to collect information (forward phase and echo phase) in a distributed system with the assumption that global information (network topology) is not available. The distributed system is assumed to be asynchronous.

The paper is structured as follows: Section 1 gives the general introduction. Section 2 presents the model of distributed recursive wave and the way of analysing and designing distributed algorithms using distributed recursive wave. Section 3 presents the algorithm to collect new information. Section 4 describes the implementation of the algorithm. Finally, the conclusions are found in section 5.
considered as an algorithmic model of the distributed recursive wave.

**Analysing and designing distributed algorithms using distributed recursive wave.**

The construction of a distributed algorithm using recursive waves can be divided into three steps:
1. Define variables and relationships between them.
2. Build the forwarding phase.
3. Build the echo (backwarding) phase.

First, the local data are represented by variables, including the pure data of the application and the variables related to the nature of the network. The operations determining the values of local variables can be included when they are needed for the algorithm.

Forwarding phase relates to all calculations before the processes are activated to perform an actual computation. In this phase there are two basic tasks: calculating going path of the recursive wave and preparing arguments for recursive wave call.

**3. CONVERGECAST ALGORITHM USING DISTRIBUTED RECURSIVE WAVE**

With the requirements stated in section 1 and the ways to design the algorithm as in section 2, the convergecast algorithm will be developed. First, the local variables of each process will be determined including:

- A variable NB: presenting neighborhood nodes of the node being watched. This variable can also be expanded to contain more information about the link between the current node and neighbor nodes. The value of variables can be determined locally.

**Function** `ConvergeCast(Data, ListOfProcessID) Of List`

```plaintext
if not Visited then
    R=Converge(Select(Neighborhood));
else
    Return CollectLocal();
end if;
end Function;
```

**Figure 2: Convergecast function**

- A visited variable is used to determine the status of recursive waves (finished / not finished).
- A variable R is used to store the results returned from the process of gathering information. This variable is only valid when the forward phase ends.

Forward phase is designed according to the strategies of application. On the basis of recursive waves, there can be the following possibilities:

- One selection: send information to only one process.
- All selection: send a message to all neighbor processes.
- Random selection.
- Selection according to network data of the local process.

The selection is generally expressed by a function Select. With the arguments above, information collecting algorithm is described as shown in Figure 2: Convergecast function.

The complexity of the algorithm depends on the function Select. In case of all selection, the complexity is \( O(d) \) in which \( d \) is the diameter of the network. The function Select can be adjusted to get the implementation process more efficient, but if the ratio of the number of selected processes to all of the neighboring processes is lowerly limited by none-zero value, the time complexity is still \( O(d) \).

4. IMPLEMENTATION

In order to implement the algorithm we can use a simulated or real environment. However, in both cases, distributed recursive calls need to be supported by the system. We need to find a distributed system that allows:
- Sending a script to a remote computer
- Activating and executing the script.

For these reasons, the author have selected Shell Script and Linux operating system as execution environment. With the assumption that above actions are supported, function ConvergeCast() in Figure 2 becomes ConvergeCastImpl() with shell language as in Figure 3: Implementing ConvergeCast by Shell.

```
#!/bin/bash
if [ "$VISITED" = "YES" ]; then
    echo "Collected Information"
    uname -a >return.dat
else
    for i in $(cat neighbors); do
done
    scp convergecast root@$i:~/run/convergecast
    ssh root@$i 'chmod +x /run/convergecast'
    scp root@$i:~/run/return.dat temp
    cat temp>>return.dat
    scp convergecast root@$i:~/run/convergecast
    export VISITED = "YES"
fi
```

Figure 3: Implementing ConvergeCast by Shell

The program is installed on computers running Linux operating system, which is configured to connect together using OpenSSL without password. An overlay network is built by creating on each computer a neighbor file containing a list of addresses of neighboring nodes. This is the only information about the configuration of the system. A return.dat file used to store returned results. The program runs correctly, as all the machines in the overlay network configurations are collected on the root machine.

The program is tested on various types of network. To test that the program is correct, a single machine is used with multiple IP addresses. Another scenario is to use 5 virtual machines on a physical machine. The program is executed with 5 topologies. The running time and the correctness of the result are metrics. The results are shown on the table

<table>
<thead>
<tr>
<th>Topology</th>
<th>Diameter</th>
<th>Number of executions</th>
<th>Correct result?</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>20</td>
<td>Yes</td>
<td>120</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>20</td>
<td>Yes</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>20</td>
<td>Yes</td>
<td>110</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>20</td>
<td>Yes</td>
<td>220</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>20</td>
<td>Yes</td>
<td>250</td>
</tr>
</tbody>
</table>

Unfortunately, we are not able to run the algorithm on a larger system, so the result can show only that the algorithm is correct. In addition, timing of the OpenSSL activities infuses strongly on We hope that in the future we’ll be able to experiment this algorithm on a simulation environment such as Planet-Lab or G-Lab, so that the results will be more persuasive.

5. CONCLUSIONS

The article has presented the model of distributed recursive waves and how to design and analyze a distributed algorithm based on distributed recursive wave. A new ConvergeCast algorithm based on distributed recursive wave has been presented. The complexity of the algorithm does not change in comparison with other algorithms, but the distributed recursive wave allows customizing the broadcast process to get effect suitable with the communication infrastructure.

The algorithm has been installed on Linux using Shell Script and satisfies the requirements of ConvergeCast algorithm.

The program can be applied to solve many problems, such as network routing, network management, and network monitoring. The required condition is to have the equipment to support Shell in overlay networks.

Based on distributed recursive wave, algorithms can be developed for environments more complex than distributed system, with error conditions, more severe operating conditions of the devices.

Acknowledgement

This work was supported by grant B2012-01-30 from Ministry of Training and Education of Vietnam

References


