International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 04 Issue: 07 | July -2017 www.irjet.net p-ISSN: 2395-0072

COMPUTATION OF REPLICATION RATE AND MEMORY USED FOR STATIC AND DYNAMIC DATA REPLICATION STRATEGIES

DEEPAK.V

Student, Department Of Telecommunication Engineering, Digital Communication (M.Tech), M.S.Ramaiah Institute Of Technology

---

Abstract: *Replication means taking a complete copy of user data to another system, usually at a separate geographical recovery location. This means you have a backup of your data you can use this stored file if the primary copy of file fails. This replicated copy can also be used for other things like off-site backup, load testing. The replication can happen in all sorts of different places and the replication is configured and managed by the controllers of the storage providers. It can work at the application level, where replication is configured and set up between your host and storage system and virtualizes your storage. That brings benefits in that the make and model of storage array at either end can be different so we are not confined to using the same model at both ends. The last place it can happen is at the host server level, and that gives you greater flexibility in allowing replication between different types of storage at both ends. In that model, the software or the operating system controls the replication between two different servers at two different sites and the replication is over IP links.*

***Key Words*: Replication, Static and Dynamic Replication Strategies, Replication Rate, Memory Used.**

1. INTRODUCTION

The communicational backbone of cloud computing is the Data Center Network (DCN) [3]. We use three DCN architectures namely: Three tier, Fat tree and D-cell. However, to meet the growing demands of the cloud computing, the Fat tree and D cell architectures were proposed. Therefore, we use the aforementioned three architectures to evaluate the performance of our scheme on legacy as well as state of the art architectures. The Fat tree and three tier architectures are switch-centric networks. The nodes are connected with the access layer switches. Multiple access layer switches are connected using aggregate layer switches. Core layers' switches interconnect the aggregate layer switches. The D cell is a server centric network architecture that uses servers in addition to switches to perform the communication process within the network. A server in the D cell architecture is connected to other servers and a switch. The lower level d cells recursively build the higher level d cells. The d cells at the same level are fully connected.

We compared the results of the DROPS methodology with fine-grained replication strategies, namely:(a) DRPA- star, (b) WA-star, (c) Aϵ-star, (d) SA1, (e)SA2, (f) SA3, (g) Local Min-Min, (h) Global Min-Min, (i) Greedy algorithm, and (j) Genetic Replication Algorithm (GRA).

2. METHODOLOGY

2.1 Impact of increase in the number of nodes

Step 1: Set the value of the number of fragments to 50.
Step 2: Initialize the value of the number of nodes to 100.
Step 3: Consider the Average Replication Cost(RC) for each method(strategy).
Step 4: Compute the network storage capacity for the assigned value of the nodes.
Step 5: Compute the running time and memory utilization for the present criteria.
Step 6: Repeat the process by changing the value of the number of nodes to 500,1200,2400 and 30000, for the same fragmental value ,i.e,50
Step 7: Compute the running time and memory utilization for each case…
Step 8: This procedure is employed to compute the constraint values of the methods involved for the purpose of comparison and approval of the scenarios.

2.2 Impact of increase in the number of fragments

Step 1: Set the value of the number of nodes(n) to 30000.
Step 2: Initialize the fragmental value to 50(frag)
Step 3: Consider the Average Replication Cost(RC) for each method(strategy).
Step 4: Compute the network storage capacity for the assigned value of the nodes.(nc)
Step 5: Compute the replication rate and memory utilization for the present criteria.
Step 6: Repeat the process by changing the value of the number of fragments to 100,200,400 and 500, for the same nodal value,i.e,30000
Step 7: Compute the replication rate(r) and memory utilization(u) for each case…

Step 8: This procedure is employed to compute the constraint values of the methods involved for the purpose of comparison and approval of the scenarios

2.3 Impact of increase in the network storage(from 20% to 40%)

Step 1: Initialize the fragmental value to 50(frag).
Step 2: Initialize the value of the number of nodes(n) to 100.
Step 3: Consider the Average Replication Cost(RC) for each method(strategy).
Step 4: Compute the network storage capacity for the assigned value of the nodes(nc).
Step 5: Compute the replication rate(r) and memory utilization(u) for the present criteria.
Step 6: Repeat the process by varying the value of the number of nodes to 500,1200,2400 and 30000 in an iterative manner, with the same consistent modification by varying the value of number of fragments to 100,200,400 and 500.
Step 7: Compute the replication rate and memory utilization for each case…
Step 8: This procedure is employed to compute the constraint values of the methods involved for the purpose of comparison and approval of the scenarios.

3. SYSTEM DESIGN

3.1 Impact of increase in the number of nodes

```
import java.util.Scanner;
import java.util.*;
public class case2
{
public static void main(String[] args)          //CALLING FUNCTION
{
//CALLED FUNCTIONS(BASED ON NUMBER OF ITERATIONS)
gra(); //Average RC Value=  69.66
greedy(); //Average RC Value=71.28
sa1();      //Average RC Value= 60.93
sa2();     //Average RC Value=52.22
sa3();     //Average RC Value=68.08
lmm();    //Average RC Value=39.8
drpa();   //Average RC Value=77.02
drops() //Average RC Value=24.78
wastar();  //Average RC Value=74.76
gmm();   //Average RC Value=45.91
ae-star();   //Average RC Value=73.86
}
//replication strategy
public static void replication strategy()
//Initialize the value of i,n and fragments
 //Enter the number of fragments
//Enter the Average RC value for each of the cases
for(i=1;i<=frag;i++)
//The network capacity is computed(bits)
nc=frag+(n-i)……..//DRPA Strategy
nc=frag+((n-i)/2)…….//LMM
nc=frag+(n-i)……..//WA-STAR Strategy
nc= frag+((n+i)/2)…….//GMM
nc=frag+(n-i)……..//AE-STAR Method…
nc=frag+(n-i)……..//SA1
nc=frag+(n+i)…….//SA2
nc=frag+(n-i)…….//SA-3
nc=frag+n……..//Greedy algorithm
nc=frag+n……..//GRA
nc=n+i……..//DROPS
for(i=1;i<=frag;i++)
r=((nc*(avg_rc))/((n+frag)*3600));
u=(r*8);
//The Replication rate is computed(bits/sec)
//The memory utilization is being calculated..
```

3.2 Impact of increase in the fragmental value

```
import java.util.Scanner;
import java.util.*;
public class case2
{
public static void main(String[] args)          //CALLING FUNCTION
{
//CALLED FUNCTIONS(BASED ON NUMBER OF ITERATIONS)
gra(); //Average RC Value=  66.95
greedy(); //Average RC Value=71.3
sa1();      //Average RC Value= 61.17
sa2();     //Average RC Value=50.25
sa3();     //Average RC Value=63.83
lmm();    //Average RC Value=43.43
drpa();   //Average RC Value=75.38
drops() //Average RC Value=24.33
wastar();  //Average RC Value=71.02
gmm();   //Average RC Value=51.37
ae-star();   //Average RC Value=70.58
}
//replication strategy
public static void replication strategy()
//Initialize the value of i,n and fragments
 //Enter the number of fragments
//Enter the Average RC value for each of the cases
for(i=1;i<=frag;i++)
//The network capacity is computed(bits)
nc=frag+(n-i)……..//DRPA Strategy
nc=frag+((n-i)/2)…….//LMM
nc=frag+(n-i)……..//WA-STAR Strategy
nc= frag+((n+i)/2)…….//GMM
nc=frag+(n-i)……..//AE-STAR Method…
nc=frag+(n-i)……..//SA1
nc=frag+(n+i)…….//SA2
```

```
nc=frag+(n-i)……..//SA-3
nc=frag+n……..//Greedy algorithm
nc=frag+n……..//GRA
nc=n+i……..//DROPS
for(i=1;i<=frag;i++)
r=((nc*(avg_rc))/((n+frag)*3600));
u=(r*8);
//The Replication rate is computed(bits/sec)
//The memory utilization is being calculated..
```

3.3 Impact of increase in the network storage.

```
 import java.util.Scanner;
import java.util.*;
public class case2
{
public static void main(String[] args) //CALLING FUNCTION
{
//CALLED FUNCTIONS(BASED ON NUMBER OF ITERATIONS)
gra(); //Average RC Value= 63.24
greedy(); //Average RC Value=70.29
sa1(); //Average RC Value= 60.12
sa2(); //Average RC Value=50.12
sa3(); //Average RC Value=62.48
lmm(); //Average RC Value=29.21
drpa(); //Average RC Value=71.71
drops() //Average RC Value=21.7
wastar(); //Average RC Value=71.36
gmm(); //Average RC Value=41.67
ae-star(); //Average RC Value=68.36
}
//replication strategy
public static void replication strategy()
//Initialize the value of i,n and fragments
 //Enter the number of fragments
//Enter the Average RC value for each of the cases
for(i=1;i<=frag;i++)
//The network capacity is computed(bits)
nc=0.2*(frag+(n-i))……//DRPA Strategy
nc=0.2*(frag+((n-i)/2))……..//LMM
nc=0.2*(frag+(n-i))……..//WA-STAR Strategy
nc=0.2*(frag+((n+i)/2))…….//GMM
nc=0.2*(frag+(n-i))……..//AE-STAR Method…
nc=0.2*(frag+(n-i))……..//SA1
nc=0.2*(frag+(n+i))……..//SA2
nc=0.2*(frag+(n-i))……..//SA-3
nc=0.2*(frag+n)……..//Greedy algorithm
nc=0.2*(frag+n)……..//GRA
nc=0.2*(n+i)……..//DROPS
for(i=1;i<=frag;i++)
r=((nc*(avg_rc))/((n+frag)*3600));
u=(r*8);
//The Replication rate is computed(bits/sec)
//The memory utilization is being calculated..
```

4. RESULTS.

4.1 Impact of increase in the number of nodes (constant fragment value=50,nodal values=100,500,1200,2400,30000)

4.1.1. To determine the replication rate

```
//Create the package.
//Create  the username variable.
//Declare the password as a String.
//Enter the Login Credentials
if(UserName.equals("Valid Username" && Password.equals("Valid Password"))
//The Entered Username and Password are correct

// Secret Key is generated....

//"File is being fragmented based on the data replication strategies...."//

//The fragments report to a primary node on the server accessed by the user....//

//The file is merged and decrypted for the shared secret key to provide the Replication rate(bits/sec)//

 launch(args);

else
//The Entered UserName and Password are incorrect
```

4.1.2.To determine the memory utilization

```
//Create the package.
//Create  the username variable.
//Declare the password as a String.
//Enter the Login Credentials
if(UserName.equals("Valid Username" && Password.equals("Valid Password"))
//The Entered Username and Password are correct

// Secret Key is generated....

//"File is being fragmented based on the data replication strategies...."//

//The fragments report to a primary node on the server accessed by the user....//

//The file is merged and decrypted for the shared secret key to provide the Memory Utilization//

 launch(args);

else
```

//The Entered UserName and Password are incorrect.

4.2 Impact of increase in the number of fragments(constant nodal value=30000; fragment values=50,100,200,400,500)

4.2.1.To determine the replication rate

```
//Create the package.
//Create  the username variable.
//Declare the password as a String.
//Enter the Login Credentials
if(UserName.equals("Valid Username" && Password.equals("Valid Password"))
//The Entered Username and Password are correct

// Secret Key is generated....

//"File is being fragmented based on the data replication strategies...."//

//The fragments report to a primary node on the server accessed by the user....//

//The file is merged and decrypted for the shared secret key to provide the Replication rate(bits/sec)//

 launch(args);

else
//The Entered UserName and Password are incorrect
```

4.2.2.To determine the memory utilization

```
//Create the package.
//Create  the username variable.
//Declare the password as a String.
//Enter the Login Credentials
if(UserName.equals("Valid Username" && Password.equals("Valid Password"))
//The Entered Username and Password are correct

// Secret Key is generated....

//"File is being fragmented based on the data replication strategies...."//

//The fragments report to a primary node on the server accessed by the user....//

//The file is merged and decrypted for the shared secret key to provide the Memory Utilization//

 launch(args);

else
//The Entered UserName and Password are incorrect
```

4.3 Impact of increase in the network storage capacity (fragment values=50,100,200,400,500 nodal values=100,500,1200,2400,30000)

4.3.1.To determine the replication rate

```
//Create the package.
//Create  the username variable.
//Declare the password as a String.
//Enter the Login Credentials
if(UserName.equals("Valid Username" && Password.equals("Valid Password"))
//The Entered Username and Password are correct

// Secret Key is generated....

//"File is being fragmented based on the data replication strategies...."//

//The fragments report to a primary node on the server accessed by the user....//

//The file is merged and decrypted for the shared secret key to provide the Replication rate(bits/sec)//

 launch(args);

else
//The Entered UserName and Password are incorrect
```

4.3.2.To determine the memory utilization

```
//Create the package.
//Create  the username variable.
//Declare the password as a String.
//Enter the Login Credentials
if(UserName.equals("Valid Username" && Password.equals("Valid Password"))
//The Entered Username and Password are correct

// Secret Key is generated for ....

//"File is being fragmented based on the data replication strategies...."//

//The fragments report to a primary node on the server accessed by the user....//

//The file is merged and decrypted for the shared secret key to provide the Memory Utilization//

 launch(args);

else
//The Entered UserName and Password are incorrect.
```

5. CONCLUSION AND FUTURE WORK

The user has to register in cloud, for each registered user, a unique secret key is generated. The user when wants to upload the file, it gets splits into small chunks and for every upload of file a secret file key is also generated when user wants to download a file, they should enter a secret file key of their file, then splits chunks get merged and can download the file. This provides security at client level as well as in network level; will save the time and resources utilized in downloading, updating, and uploading the file again .All the data from various servers are submitted in a centralized global scheduler and then this data is distributed among various datacenters for processing. The global scheduler maintains a first in first out technique for all submitted data. Various data applications produce and store data locally on nearest server. On the other hand, these data may be replicated on other datacenters in the network.

For the same model, we assume that each data has given the initial data location and the replication policy. We are considering various data replication methods to replicate and store data at various distributed servers. A local task of data is maintained by each of the datacenter in the network for the tasks that are programmed to a particular datacenter.

Due to the costly data transfer delay in geographically distributed datacenters, a task can only be planned on a datacenter which has that requirement of data. Temporarily, the completion of a particular task is scheduled by the latest completion of all its tasks in the system. As in the network if any one of the task is completed and stored in the datacenters then that task will remain in datacenter until it will be called. The global scheduler must be having all the tasks from various local schedulers in each datacenter. The above described model is related to various cloud platforms.

6. REFERENCES.

1)S. Sakr, A. Liu, D. Batista, and M. Alomari, “A Survey of Large-scale Data Management Approaches in Cloud Environments,” IEEE Comm. Surveys & Tutorials, vol. 13, no. 3, pp. 311-336, Third Quarter 2011.

2)R. Kienzler, R. Bruggmann, A. Ranganathan, and N. Tatbul, “Stream As You Go: The Case for Incremental Data Access and Processing in the Cloud,” Proc. IEEE ICDE Workshop Data Management in the Cloud (DMC’12), 2012.

3)X. Zhang, C. Liu, S. Nepal, and J. Chen, “An Efficient Quasi-Identifier Index Based Approach for Privacy Preservation over Incremental Data Sets on Cloud,” J. Computer and System Sciences, vol. 79, pp. 542-555, 2013.

4) ”Improving Cloud Based Online Social Network for Data Replication and Placement”, IEEE 9TH International Conference on Cloud Computing,pp.678-685, 2016.