Group Project - Distributed Content Searching Report

Problem Description

The given problem is to design, develop and debug a simple file search engine to find contents in a distributed system using RPCs, web services or REST APIs and measure and analyse its performance. The system should consist of more than 10 nodes sharing 20 files among them with each node contributing 3-5 files and some files may be present in multiple nodes.

Phase 1

Generate the network topology and the contents of each node

Phase 2

Design and develop a socket-based solution to find the files requested by different nodes

Phase 3

Extend the system to support file transfer through web services or REST API

Phase 4

Analyze the performance

Proposed Topology

As per the given problem description every new node needs to be connected to two randomly selected nodes in the current system and the peers are not specified to have significantly different capabilities. Moreover, due to network failures peers can be disconnected which might also cause partitions in the system. It cannot be guaranteed for a resource to be found in the system and also the number of peers connected to a single node would be random for each node. Thus, the proposed topology of the network is an Unstructured P2P topology.

Communication among nodes

Every node is connected to a random number of known peers on which it keeps track of. Thus, a single peer gives a node, a random set of peers to contact. In order to guarantee the discovery of a resource, flooding based communication is proposed for the system where every node that receives a message processes it and redirects it to its other peers.

In order to limit number of unnecessary messages being communicated among nodes of the topology, various measures has been taken such as controlled flooding where we make sure that no node re-floods/retransfers a message that has been already been once passed on by that

particular node. Through this methodology we ensure that the flooding definitely stops at a certain point given that number of nodes are finite.

Routing

Every node maintains a routing table on his own with the details of the nodes that are directly connected to them and also keeps track of the messages being passed through itself so that by any chance if it receives the same message again from any node, it can discard it instead of retransmitting. Inside the message body, message orderly stores the identities of the nodes that received and retransmitted that particular instance of message. So that when the search message finally reaches a suitable node with the required file, the destination node can interpret the message and understand which node exactly it has to transmit the required file to, in order to eventually received by the desired node. And each node on the path of destination node to desired node can do the same and find out the identity of the immediate next node that it should deliver the particular file.

File Transfer Protocol

The files are transmitted between peers over TCP/IP connections. As a node allows its peers to access files without username and password the used protocol can be categorized into Anonymous File Transfer Protocols.

Performance

Performance Analysis

Following is a comparison with number of hops counts and latency required to resolve each query with network having 12 nodes and 10 nodes.

Query	Hops (10)	Latency (10)	Hops (12)	Latency (12)
Twilight	2	11	2	129
Jack	5	25	9	3492
American Idol	1	7	2	121
Happy Feet	1	5	2	1
Twilight saga				
Happy Feet	2	37	3	16
Happy Feet	1	18		
Feet	2	32		
Happy Feet	2	32	1	31
Twilight	2	10	1	4
Windows	1	63	3	16
Happy Feet	2	17	2	8
Mission Impossible	1	361	3	20
Twilight	1	31	3	39
Windows 8	2	289	7	154
The	2	430	2	100

Happy	**2**	**23**	**1**	**17**
Windows 8	**2**	**12**	**3**	**977**
Happy Feet	**1**	**31**	**1**	**74**
Super Mario	**1**	**186**	**3**	**185**
Jack and Jill	**3**	**14**	**1**	**15**
Happy Feet	**1**	**59**	**7**	**287**
Impossible	**2**	**25**	**5**	**121**
Happy Feet	**4**	**917**	**1**	**6**
Turn up the Music	**2**	**558**	**6**	**242**
Adventures of Tintin	**5**	**14**	**5**	**12**
Twilight saga				
Happy Feet	**1**	**8**	**2**	**35**
Super Mario	**2**	**138**	**2**	**320**
American Pickers	**2**	**14**	**3**	**62**
Microsoft Office 2010	**1**	**10**	**1**	**137**
Twilight	**1**	**15**	**1**	**14**
Modern Family	**1**	**125**	**2**	**32**
Jack and Jill	**1**	**22**	**4**	**41**
Jill	**3**	**316**	**4**	**2327**
Glee	**1**	**64**	**1**	**203**
The Vampire Diaries				
King Arthur	**1**	**160**	**6**	**16**
Jack and Jill	**3**	**71**	**4**	**7**
King Arthur	**1**	**177**	**2**	**1**
Windows XP	**3**	**541**	**4**	**9**
Harry Potter				
Feet	**1**	**38**	**2**	**1**
Kung Fu Panda	**2**	**8**	**2**	**5**
Lady Gaga	**3**	**447**	**9**	**325**
Gaga	**2**	**92**	**7**	**278**
Happy Feet	**2**	**16**	**2**	**5**
Twilight	**1**	**13**	**2**	**79**
Hacking	**2**	**15**	**2**	**14**
King	**1**	**170**	**3**	**6**

Following graphs clearly show the reduction of latency and hop counts with reduction of nodes in the peer to peer network implemented.

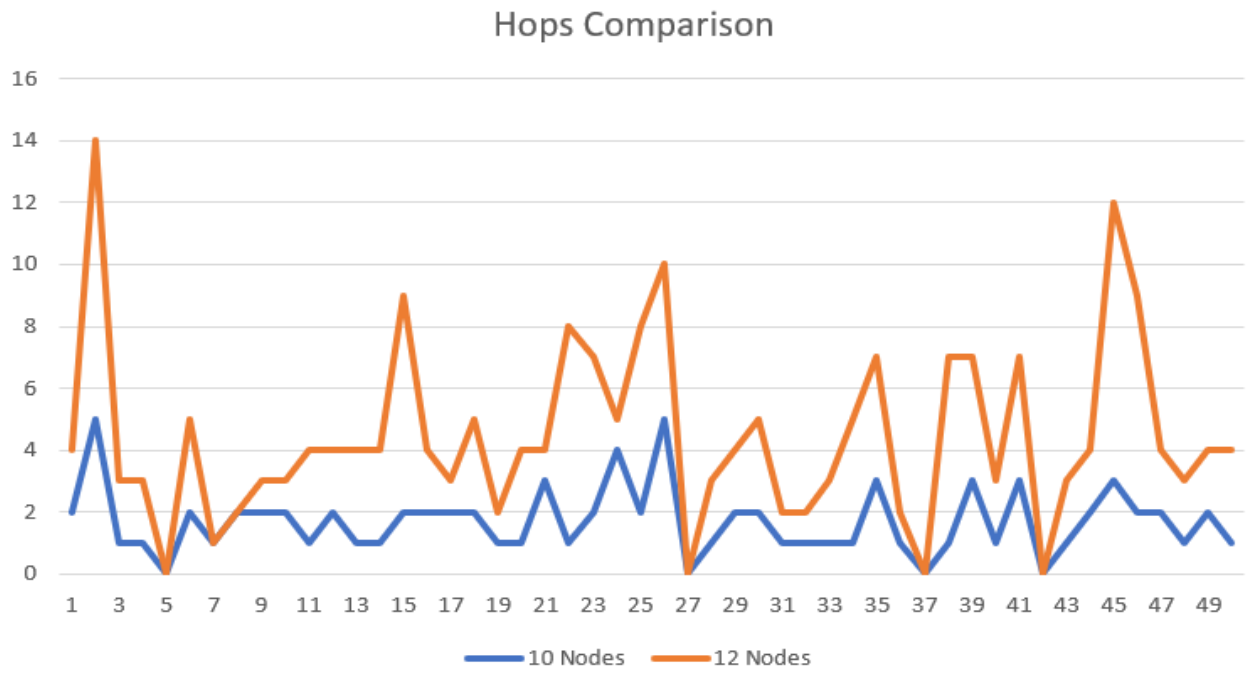

Figure 1 - Comparison of count of hops

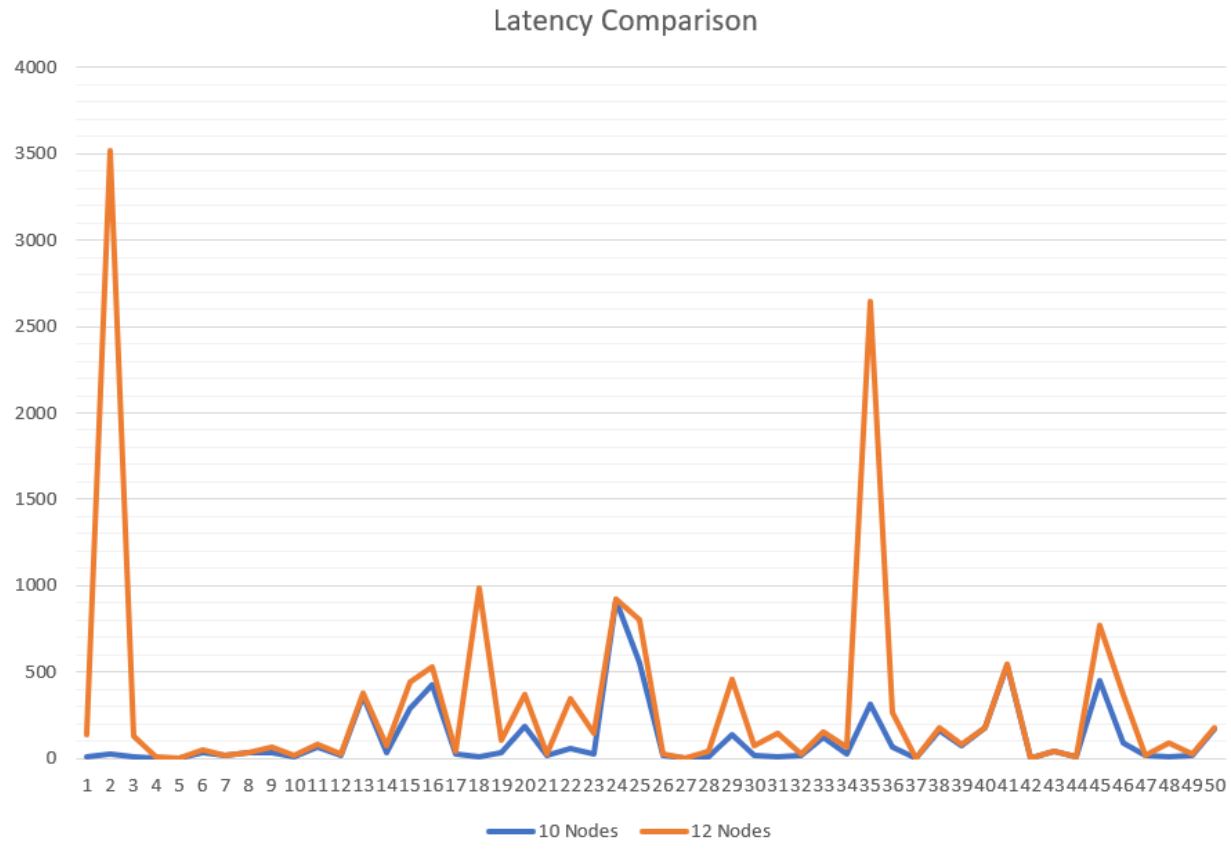

Figure 2 - Comparison of Latency

Following are some measurements we used to analyze performance of the distributed network.

Prop	Min		Max		Average		Std	
	10	12	10	12	10	12	10	12
Hops	1	1	5	9	1.85	3.14	0.99	2.12
Latency (ms)	1	1	917	3492	123.20	226.91	188.65	620.16

Further we programmatically calculated following properties for each node. Following images show results for two nodes for all the messages forwarded and received by the node.

```
Performance of Node9 of 192.168.8.147:54545

+----------------------+----------------+
| Received messages    |   24406        |
| Forwarded messages   |   10277        |
| Answered messages    |   4515         |
| Node degree          |   2            |
+----------------------+----------------+

+-----------------+----------+----------+----------+-----------+
| Property        |  Min     |  Max     |  Avg     |    SD     |
+-----------------+----------+----------+----------+-----------+
| Hops            |     1    |   11     |  9.80    |   1.59    |
| Latency         |   0.008  |  3.164   | 0.450    |  0.903    |
+-----------------+----------+----------+----------+-----------+
```

Figure 1 - Performance of Node 9

```
Performance of Node6 of 192.168.8.147:54544

+----------------------+----------------+
| Received messages    |   20057        |
| Forwarded messages   |   6796         |
| Answered messages    |   6680         |
| Node degree          |   2            |
+----------------------+----------------+

+-----------------+----------+----------+----------+-----------+
| Property        |  Min     |  Max     |  Avg     |    SD     |
+-----------------+----------+----------+----------+-----------+
| Hops            |     1    |   11     |  9.69    |   1.68    |
| Latency         |   0.007  |  3.368   | 0.720    |  0.808    |
+-----------------+----------+----------+----------+-----------+
```

Figure 2 - Performance of Node 6

Also following screenshot shows the implementation of the file download process using REST API. Every node is initialized with a REST API end point to download files. When one node has a file searched by another node, it sends the file download endpoint to the requested node. Then the request node uses this endpoint to download the file randomly generated by the source.

```
Welcome Client . Please Select whichever You Want

 1 - Register to BS
 2 - Search Files
 3 - Configure BS IP
 4 - Check Performance
 5 - Check Routing Table
 6 - Check FileList
 7 - Reset Performance Stats
 8 - Unregister node from the network
 9 - Download File

9
Choose the file you want to download
1 Lady_Gaga - http://192.168.8.146:45455/download/Lady_Gaga
2 Modern_Family - http://192.168.8.146:45453/download/Modern_Family
3 Glee - http://192.168.8.132:45454/download/Glee
4 American_Pickers - http://192.168.8.146:45456/download/American_Pickers
5 Happy_Feet - http://192.168.8.147:45454/download/Happy_Feet
6 Jack_and_Jill - http://192.168.8.132:45456/download/Jack_and_Jill
7 2 - http://192.168.8.147:45455/download/2
8 American_Idol - http://192.168.8.146:45453/download/American_Idol
9 Windows_8 - http://192.168.8.146:45456/download/Windows_8
10 Twilight - http://192.168.8.146:45456/download/Twilight
5
2020-02-07 13:01:09.119  INFO 14996 --- [io-45454-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring DispatcherServlet 'dispatcherServlet'
2020-02-07 13:01:09.125  INFO 14996 --- [io-45454-exec-1] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
2020-02-07 13:01:09.171  INFO 14996 --- [io-45454-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 45 ms

File Name : Happy_Feet
Hash value for the file : d41d8cd98f00b204e9800998ecf8427e
File Size : 2 MB

---- File downloaded Successfully ---------


Downloaded File Name : Happy_Feet downloaded
Hash value for the file : d41d8cd98f00b204e9800998ecf8427e
File Size : 2.0 MB
File Location : C:\Users\Supun\Downloads\Download\target\Happy_Feet downloaded
```

Figure 3 - File Download Process