1. Is your system: (not quite sure im thinking structured P2P)
   1. a publish/subscribe system?
   2. ... a message Queue?
   3. ... a structured P2P system?
   4. ... an unstructured P2P system?
   5. ... a distributed set implementation?
   6. ... a distributed hash table implementation?
2. What is the average-case, best-case, and worst-case space consumed at each Node?

Every node will have ever resource in the system so all cases are O(n) where n is the number of resources

1. What is the average-case, best-case and worst-case number and size of messages being sent as a result of
   1. A PUT message from a client, and

A put message will initiate a new message with the length of the resource for every node in the system.

* 1. A successful GET message from a client (that is, a value is found and sent back.)

Will have 1 very small request message and 1 response message that increases in length depending on the resource size

* 1. An unsuccessful GET message from a client (that is, no value is found.)

A single request message, and no response.

1. Based on 2 and 3, write a paragraph or two on the current scalability of your system.
2. Based on 2, 3 and 4, give suggestions for improving the scalability of your system.
3. Briefly explain your solution and why it works.

This solution will handle node failure fairly well, since every node will have all the resources that enter the system. However get client and put client will never know whether or not the node still exists.