Efficient Server-Mediated Peer-to-peer (P2P) Network

Zifan He[1]

[1]UCLA Connection Lab
Internet Research Initiative

May 23, 2022

Quad Chart

Topic: Efficient Server-Mediated P2P Network

- The increase of world data size indicates a elevating requirement of data storage.
- Peer-to-peer technology is popular in the field of file sharing.

Idea

- How could the P2P system search the target file efficiently given the architecture?
- What configuration should we set to minimize the cost of searching?
- How should we break the file into chunks to maximize the storage utilization?

Research

- Simulation: Use Python NetworkX to imitate P2P network dynamics.
- Analysis: Based on simulation result, generate hypothesis and analyze using probability and network theory.
- Compare the analysis with simulation for verification.

Result and Next Step

- We computed the optimal number of trusted nodes for given a certain P2P network configuration using different methods.
- Results of analysis and simulation are formulated in a paper (submitted to GlobeCom '22)
- Next step: File fragmentation.

Introduction and Motivation

- Conventional backup systems have some drawbacks:

Introduction and Motivation

- Conventional backup systems have some drawbacks:
  - Physical storage like hard drive or SSD is less flexible and may be broken accidentally.

Introduction and Motivation

- Conventional backup systems have some drawbacks:
  - Physical storage like hard drive or SSD is less flexible and may be broken accidentally.
  - Cloud storage will be charged monthly and it lacks control of privacy.

Introduction and Motivation

- Conventional backup systems have some drawbacks:
  - Physical storage like hard drive or SSD is less flexible and may be broken accidentally.
  - Cloud storage will be charged monthly and it lacks control of privacy.
- Though P2P network is often used for file sharing, it is also a good candidate for file backup: we can store our files in others' computers

Introduction and Motivation

- Conventional backup systems have some drawbacks:
  - Physical storage like hard drive or SSD is less flexible and may be broken accidentally.
  - Cloud storage will be charged monthly and it lacks control of privacy.
- Though P2P network is often used for file sharing, it is also a good candidate for file backup: we can store our files in others' computers
- Since the network structure for P2P system is more unorganized than server-client system used by Cloud storage, extra questions need to be considered to achieve optimal efficiency and utilization:

Introduction and Motivation

- Conventional backup systems have some drawbacks:
  - Physical storage like hard drive or SSD is less flexible and may be broken accidentally.
  - Cloud storage will be charged monthly and it lacks control of privacy.
- Though P2P network is often used for file sharing, it is also a good candidate for file backup: we can store our files in others' computers
- Since the network structure for P2P system is more unorganized than server-client system used by Cloud storage, extra questions need to be considered to achieve optimal efficiency and utilization:
  - How can we search and upload files quickly.

Introduction and Motivation

- Conventional backup systems have some drawbacks:
  - Physical storage like hard drive or SSD is less flexible and may be broken accidentally.
  - Cloud storage will be charged monthly and it lacks control of privacy.
- Though P2P network is often used for file sharing, it is also a good candidate for file backup: we can store our files in others' computers
- Since the network structure for P2P system is more unorganized than server-client system used by Cloud storage, extra questions need to be considered to achieve optimal efficiency and utilization:
  - How can we search and upload files quickly.
  - How should we cut files into chunks so that more space can be used.

Introduction and Motivation

- Conventional backup systems have some drawbacks:
  - Physical storage like hard drive or SSD is less flexible and may be broken accidentally.
  - Cloud storage will be charged monthly and it lacks control of privacy.
- Though P2P network is often used for file sharing, it is also a good candidate for file backup: we can store our files in others' computers
- Since the network structure for P2P system is more unorganized than server-client system used by Cloud storage, extra questions need to be considered to achieve optimal efficiency and utilization:
  - How can we search and upload files quickly.
  - How should we cut files into chunks so that more space can be used.
  - How can we do version control across multiple machines to update the files.

Introduction and Motivation

- Conventional backup systems have some drawbacks:
  - Physical storage like hard drive or SSD is less flexible and may be broken accidentally.
  - Cloud storage will be charged monthly and it lacks control of privacy.
- Though P2P network is often used for file sharing, it is also a good candidate for file backup: we can store our files in others' computers
- Since the network structure for P2P system is more unorganized than server-client system used by Cloud storage, extra questions need to be considered to achieve optimal efficiency and utilization:
  - How can we search and upload files quickly.
  - How should we cut files into chunks so that more space can be used.
  - How can we do version control across multiple machines to update the files.
- This year: mainly focused on performance analysis of file searching

Introduction and Motivation

- Conventional backup systems have some drawbacks:
  - Physical storage like hard drive or SSD is less flexible and may be broken accidentally.
  - Cloud storage will be charged monthly and it lacks control of privacy.
- Though P2P network is often used for file sharing, it is also a good candidate for file backup: we can store our files in others' computers
- Since the network structure for P2P system is more unorganized than server-client system used by Cloud storage, extra questions need to be considered to achieve optimal efficiency and utilization:
  - How can we search and upload files quickly.
  - How should we cut files into chunks so that more space can be used.
  - How can we do version control across multiple machines to update the files.
- This year: mainly focused on performance analysis of file searching

System Design

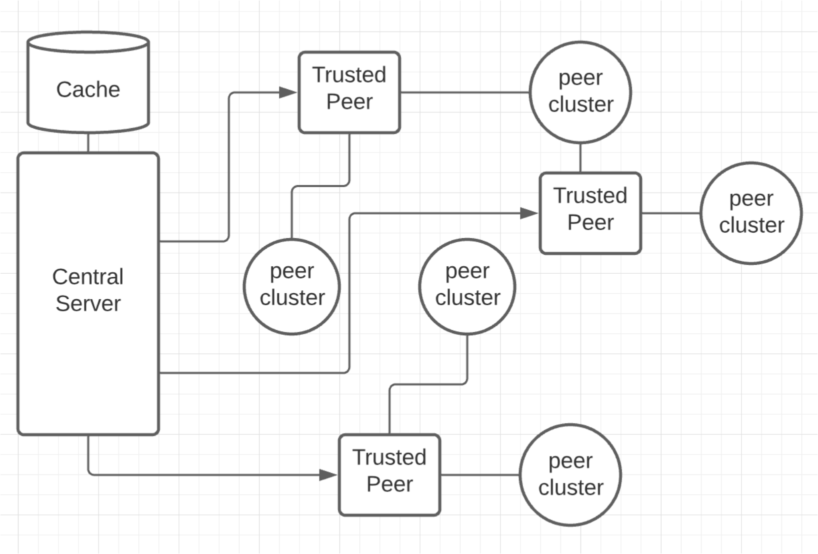


Figure: Server-Mediated Peer-to-Peer Network

Modeling

When it is hard to obtain the real data for analysis, a good way to model P2P network is using **random graph**.

Modeling

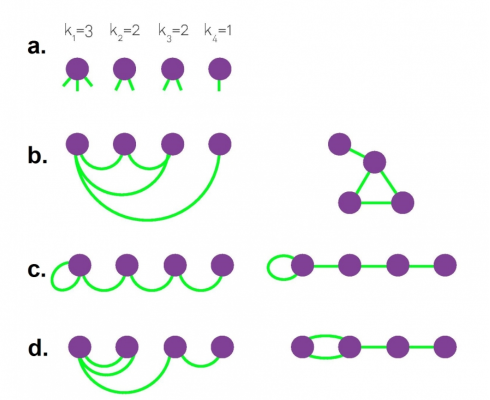
When it is hard to obtain the real data for analysis, a good way to model P2P network is using **random graph**.

We used one subset of random graph: **configuration model** with power-law degree distribution.

Modeling

When it is hard to obtain the real data for analysis, a good way to model P2P network is using **random graph**.

We used one subset of random graph: **configuration model** with power-law degree distribution.



Mathematical Analysis

Goals:

1. Find the average number of hops required to reach the file we want from the trusted peers.
2. Construct a cost function and find the optimal number of trusted peers to minimize the number of hops needed.

Average Number of Hops Needed

$$a_n = (1 - \frac{a_{n-1}}{N})L + a_{n-1}, a_1 = L$$

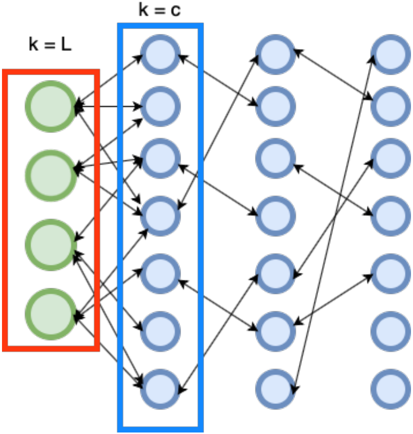


Figure: Flooding from Trusted Nodes

Average Number of Hops Needed

$$c_n = d_{n-1}(1-(1-\frac{D}{N})^{c_{n-1}+1}), c_1 = a_T$$

$$d_n = d_{n-1} - c_n$$

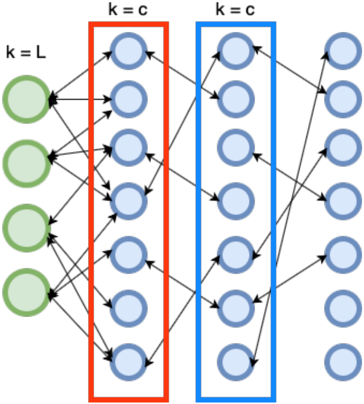


Figure: Flooding from Normal Nodes

Stopping Criteria

When should we stop flooding?

Stopping Criteria

When should we stop flooding?

Node Coverage (NC): check the fraction of nodes visited in the network. If it is higher of average possibility of hit a file replica, we will stop.

Stopping Criteria

When should we stop flooding?

Node Coverage (NC): check the fraction of nodes visited in the network. If it is higher of average possibility of hit a file replica, we will stop.

Minimum Average Shortest Path (MASP): Find the minimum of average shortest path to each node across each trusted nodes.

Stopping Criteria

When should we stop flooding?

Node Coverage (NC): check the fraction of nodes visited in the network. If it is higher of average possibility of hit a file replica, we will stop.

Minimum Average Shortest Path (MASP): Find the minimum of average shortest path to each node across each trusted nodes.

MASP is easier to compute analytically, while NC is easier to compute numerically.

Stopping Criteria

When should we stop flooding?

Node Coverage (NC): check the fraction of nodes visited in the network. If it is higher of average possibility of hit a file replica, we will stop.

Minimum Average Shortest Path (MASP): Find the minimum of average shortest path to each node across each trusted nodes.

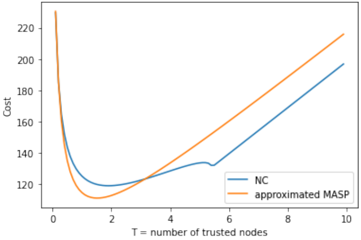
MASP is easier to compute analytically, while NC is easier to compute numerically.

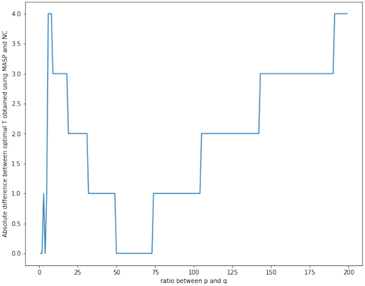
Optimization

$$C(p,q) = pT + q(W(r,T,N)+1)$$

$C(p,q)$: Cost function
p: Cost per instance of trusted peers
q: Cost per request of file per second
W: Minimum number of hops needed
r: Number of file replica in the network
T: Number of trusted nodes
N: Total number of nodes

Optimization





Simulation

```
for t in T:
  # create BA graph
  init_graph, init_n = init_gen(t)
  BA_graph = nx.barabasi_albert_graph(N, t, initial_graph=init_graph)
  nodes = []
  for n, nbrsdict in BA_graph.adjacency():
    if n not in init_n:
      nodes.append(n)
    else:
      trust_degree.append(len(nbrsdict))

  #run trials
  acc = 0
  for i in range(num_trial_trust_nodes_exp):
    disable = np.random.choice(nodes, size=int(proportion_disable * (N-t)), replace=False)
    Copy = copy.deepcopy(BA_graph)
    for n in disable:
      Copy.remove_node(n)

    file_list = np.random.choice(nodes, size=4, replace=False)
    min_level = N-1
    ans, file_key = bfs(Copy, copy.deepcopy(init_n), file_list)
    if ans != -1:
      min_level = ans
    acc += min_level

  res_list.append(acc/num_trial_trust_nodes_exp)

L = np.median(trust_degree)
```

Simulation

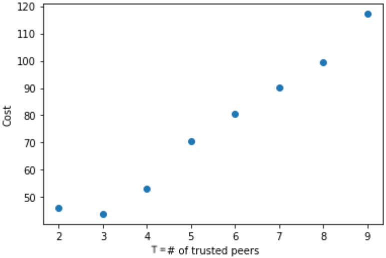


Figure: p=10, q=10

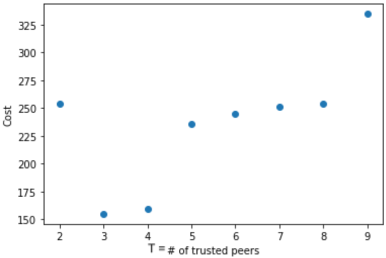


Figure: p=10, q=90

Extensions

We can also take cache resource into consideration.

Extensions

We can also take cache resource into consideration.

$$C(p, q, c) = pT + cS + (1 - H(S))(W(r, T, N) + 1)$$

Extensions

We can also take cache resource into consideration.

$$C(p, q, c) = pT + cS + (1 - H(S))(W(r, T, N) + 1)$$

c is the cost per cache resource, S is the size of cache resource, and $H(S)$ is the hit ratio of the cache.

Other Works and Future Goals

- The paper with title "Optimization of Assisted Search Over Server-Mediated Peer-to-peer Networks" has been submitted to GlobeCOM 2022.
- Next research goal: file segmentation to improve storage utilization:
  - Literature review: BitTorrent
  - Modeling with Python NetworkX with similar code.