Maintaining File Consistency in Your Gnutella-Style P2P System

Performance Evaluation

CS-550 Advanced Operating Systems

Name: Suraj Kumar Didwania (A20334147)

Name: Lawrence Amadi (A20382063)

CS550 PA3 Performance Evaluation

Following is the statistics of experiments of all the peers

Performance Evaluation is done based on 2 categories:

Push Approach: Responsibility of peer as server to broadcast information of the modification to all the neighbour peers.

Pull Approach: It's responsibility of Peer as client to fetch the information from server to know if there is any modification done once we have our TTR expired.

System Configuration:

- Total no of peers connects: 10
- All running in same system with different JVM (Different physical folder for each peer)
- RAM: 8GB
- Modification can be done only to the files in original folder.
- All the peers are up and running .
- Network is connected(we used configuration file to stablish connection)

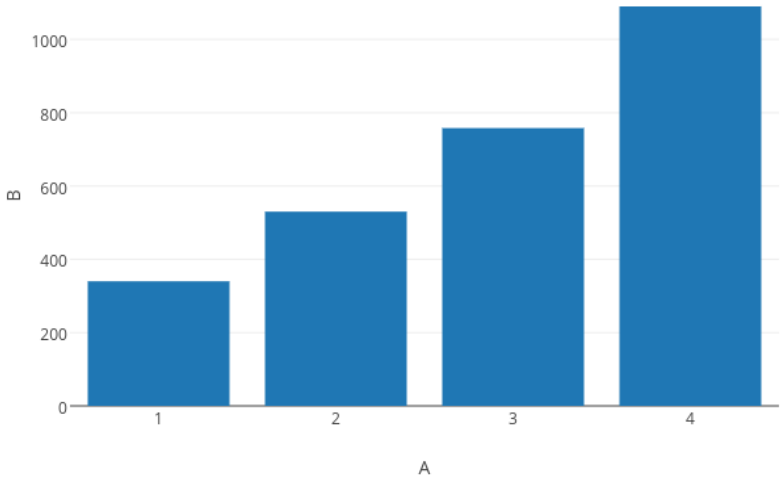
Push Approach:

Testing the effectiveness of PUSH. Let one peer do queries and downloads (and refreshes) randomly and collect the query results for each query, statistically compute the percentage of invalid query results that comes back (we define a query result to be invalid if the attached last-mod-time is less than that of the query result from the origin server). The remaining peers simulate modifications of their own files and broadcast invalidations.

Changing system load:

Average response time per client query request in ms

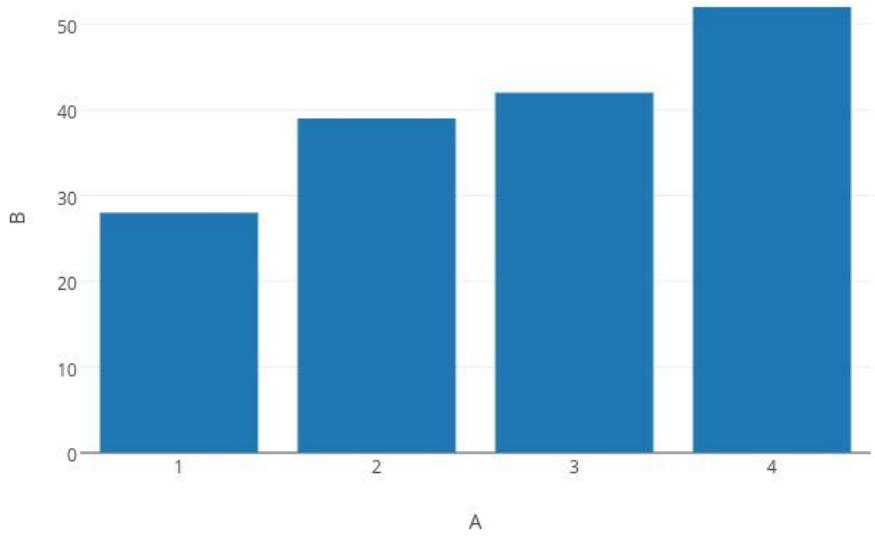
Peer	Average response time per client query request
One	340
Two	530
Three	758
Four	1090



Time to Refersh:

Average response time per client query request in ms to refresh document.
TTR set to 1 minute.

Peer	
One	28
Two	39
Three	42
Four	52



Percentage of Invalid Query: Invalid query request is negligible.

Conclusion: As the number of peers requests increases concurrently, the percentage of invalid query results that comes back is anyway less because of extra network occupied.

Pull Approach:

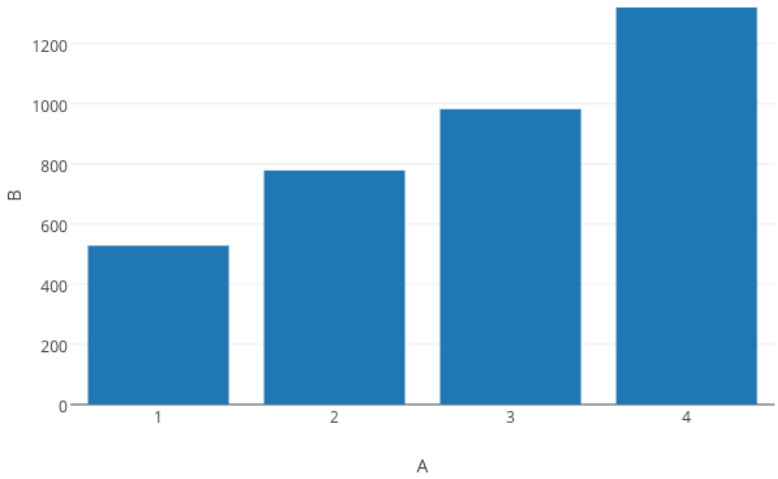
Testing the effectiveness of PULL. Let 2 to 3 peers do queries, downloads, and refreshes. The remaining peers simulate modifications using an exponential distribution. Collect Statistics of the percentage of invalid query results.

Time to download:-

Changing system load:

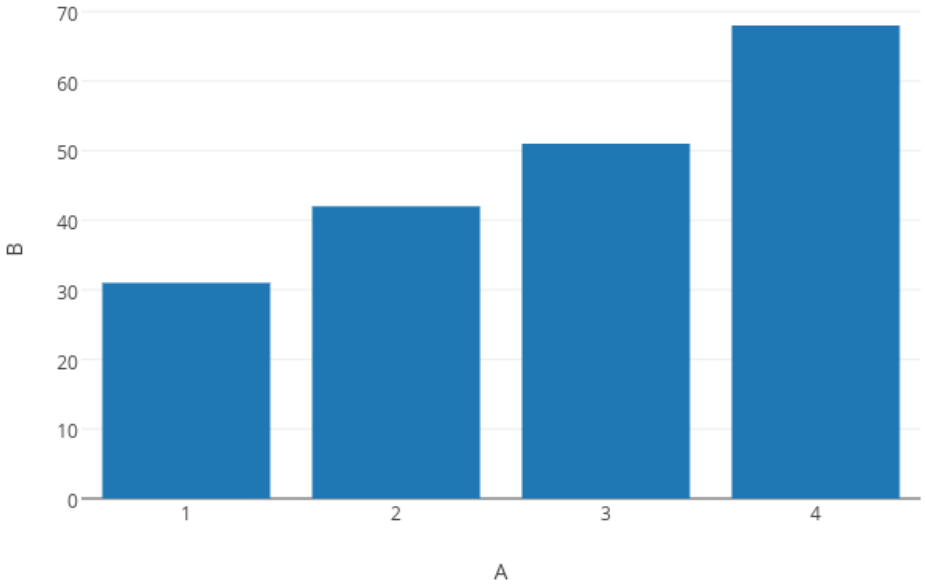
Average response time per client query request in ms

Peer	Average response time per client query request
One	528
Two	778
Three	982
Four	1320



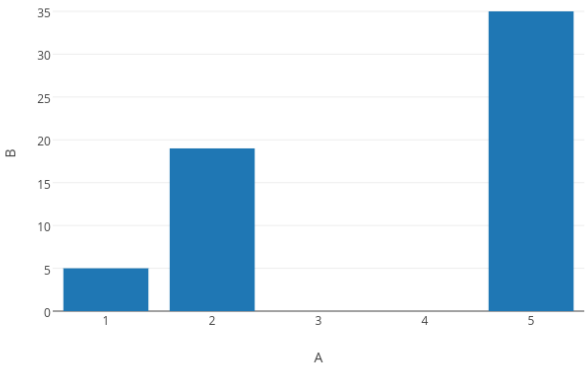
Time to refresh:-

Peer	Average response time per client query request
One	31
Two	42
Three	51
Four	68



Percentage of invalid query: With respect to TTR, percentage of invalid request

TTR	Percentage
1 min	5
2 min	19
5 min	35



Conclusion: As the number of peers requests increases concurrently, the Percentage of invalid query results that comes back is directly proportional to TTR kept and time taken for request is more than the push configuration.

Differences:

Parameter	Push	Pull
State of Server	List of Client caches	None
Response time at client	Immediate	Fetch Update time

	Push Approach	**Pull Approach**
Advantages	It's simple, stateless	Efficient in terms of network bandwidth usage as poll the server in Lazy fashion at a later time.
	Good consistency guarantees for peers that are online and reachable from the owner.	Is better suited for dynamic networks but provides weaker guarantees
Disadvangtages	Inefficient in terms of network bandwidth usage	Complex as the TTR and polling is involved.
	Not resilient to server crashes	High message overhead
Applicability	Proxy(Web page modified, notify each proxy)	HTTP