CS550 Programming Assignment 3 (PA#3)

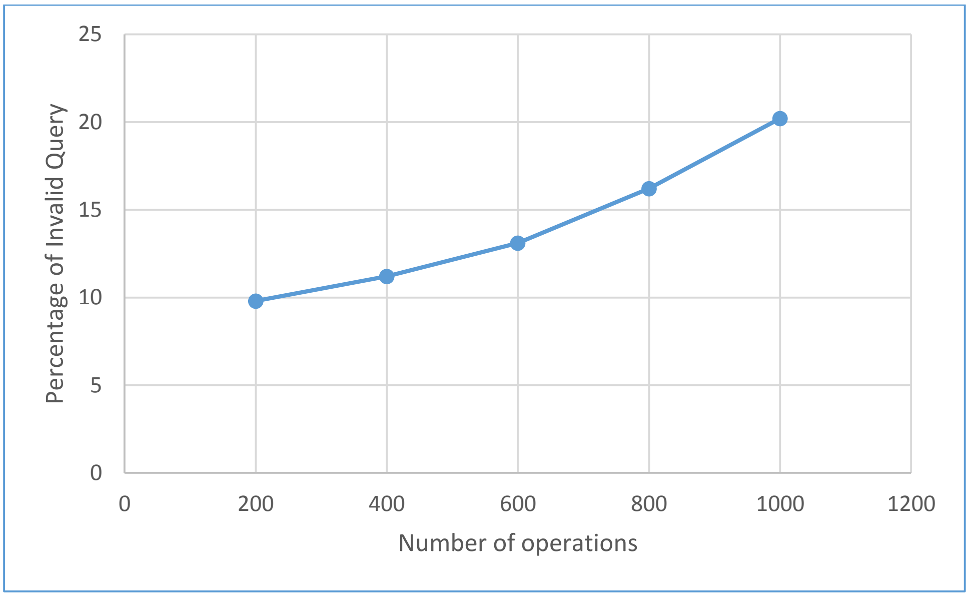
Maintaining File Consistency in the Hierarchical Gnutella-style P2P File Sharing System

Performance Evaluation

Scenario: A Single Peer searching and downloading and remaining peers modifying files:

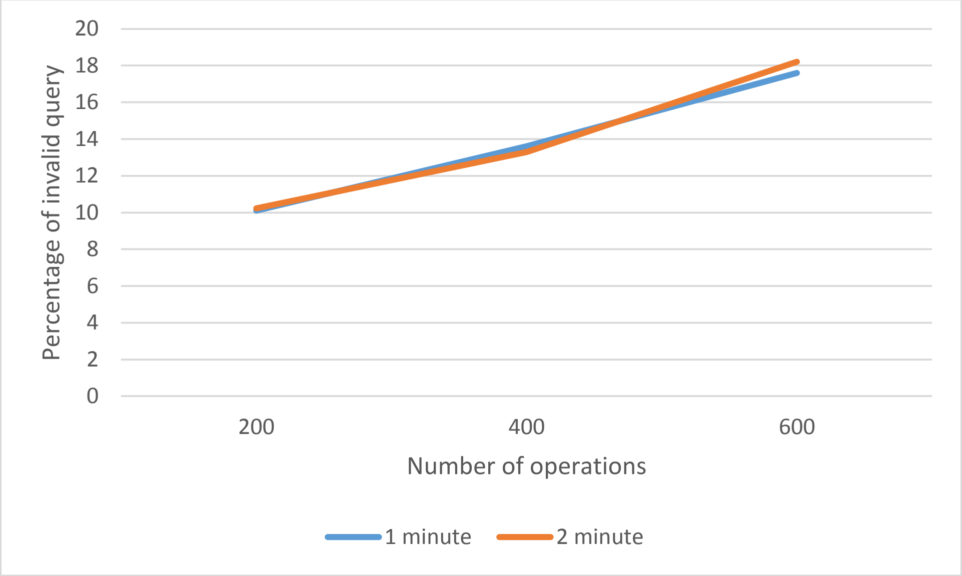
Push:

Number of Operations	Percentage of invalid query
200	9.8
400	11.2
600	13.1
800	16.2
1000	20.2



Pull:

	Percentage of invalid query	
Number of operations	1 minute	2 minute
200	10.11	10.23
400	13.6	13.3
600	17.6	18.2

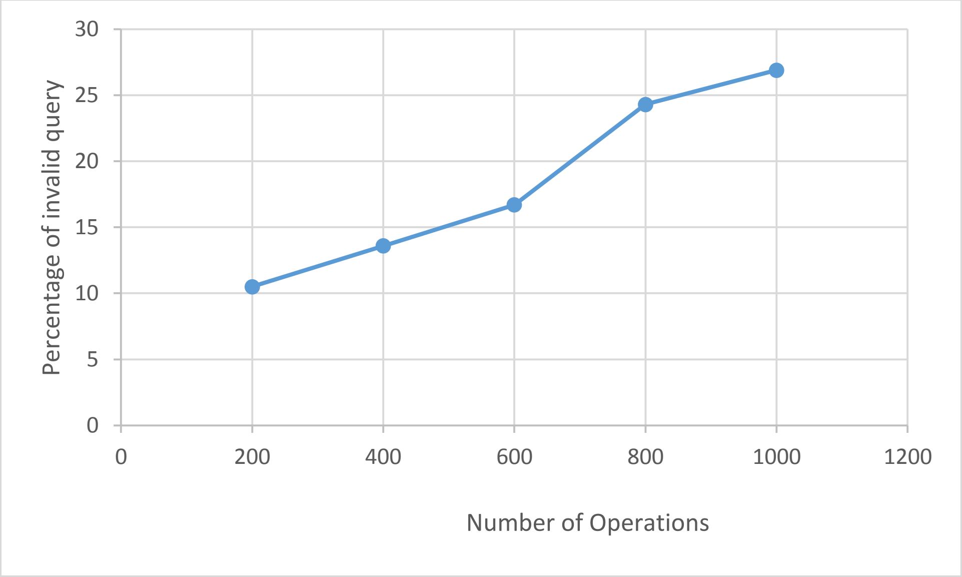


We observe that the as the number of requests increases, the percentage of invalid queries also increases.

Scenario: Two peers searching and downloading and the remaining peers modifying files:

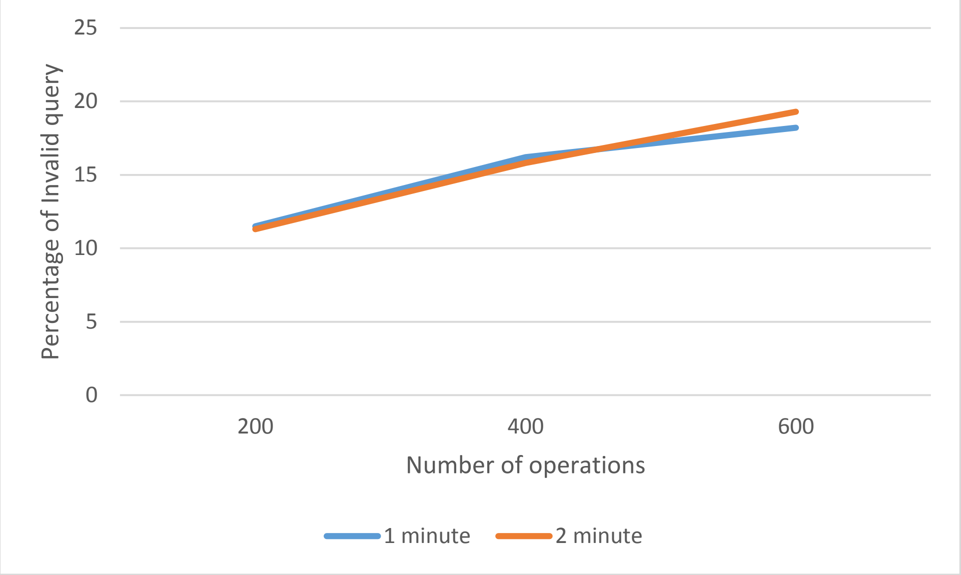
Push:

Number of Operations	Percentage of invalid query
200	10.5
400	13.6
600	16.7
800	24.3
1000	26.9



Pull:

Number of operations	Percentage of invalid query	
	1 minute	2 minute
200	11.5	11.3
400	16.2	15.8
600	18.2	19.3

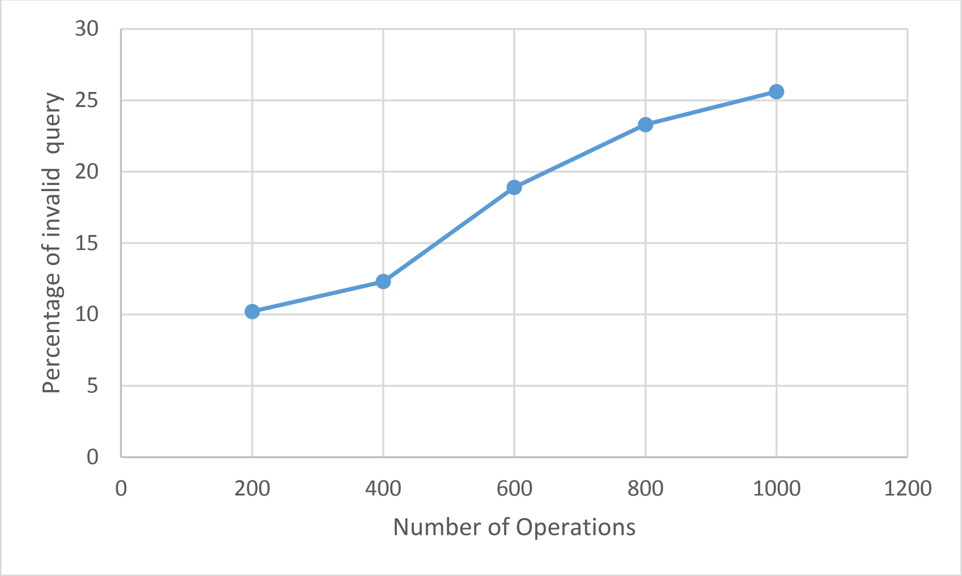


We observe that the as the number of requests increases, the percentage of invalid queries also increases.

Scenario: Three peers searching and downloading and the remaining peers modifying files:

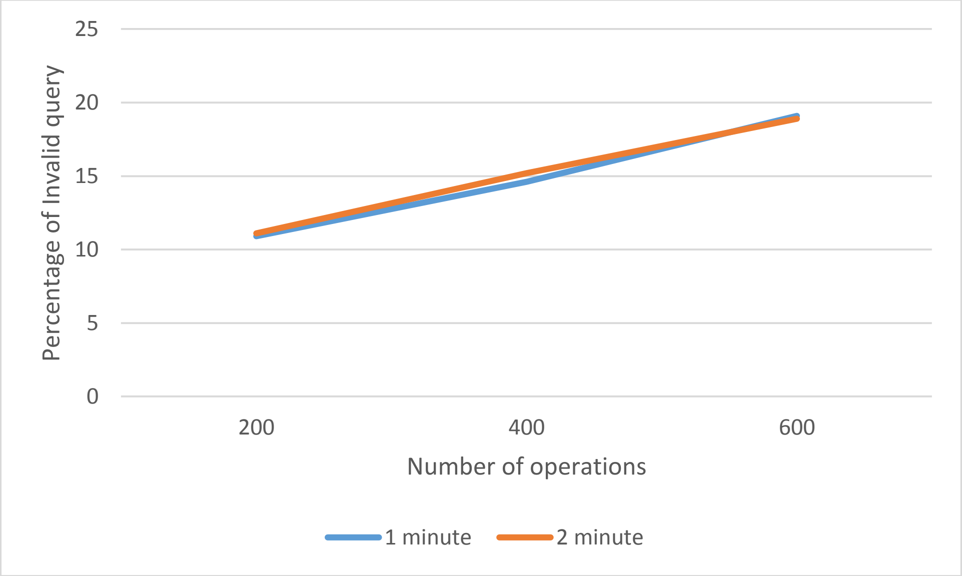
Push:

Number of Operations	Percentage of invalid query
200	10.2
400	12.3
600	18.9
800	23.3
1000	25.6



Pull:

Number of operations	Percentage of invalid query	
	1 minute	2 minute
200	10.9	11.1
400	14.6	15.2
600	19.1	18.9

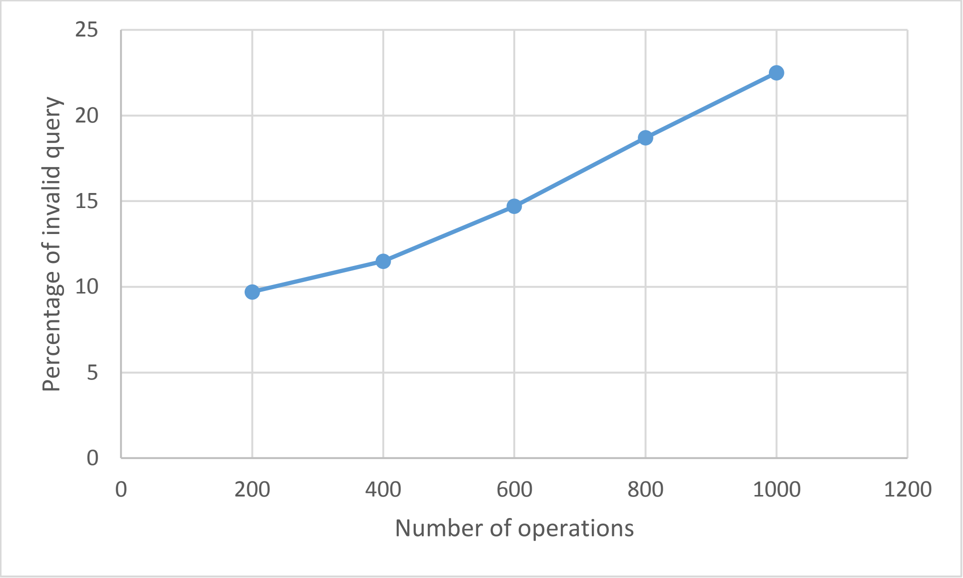


We observe that the as the number of requests increases, the percentage of invalid queries also increases.

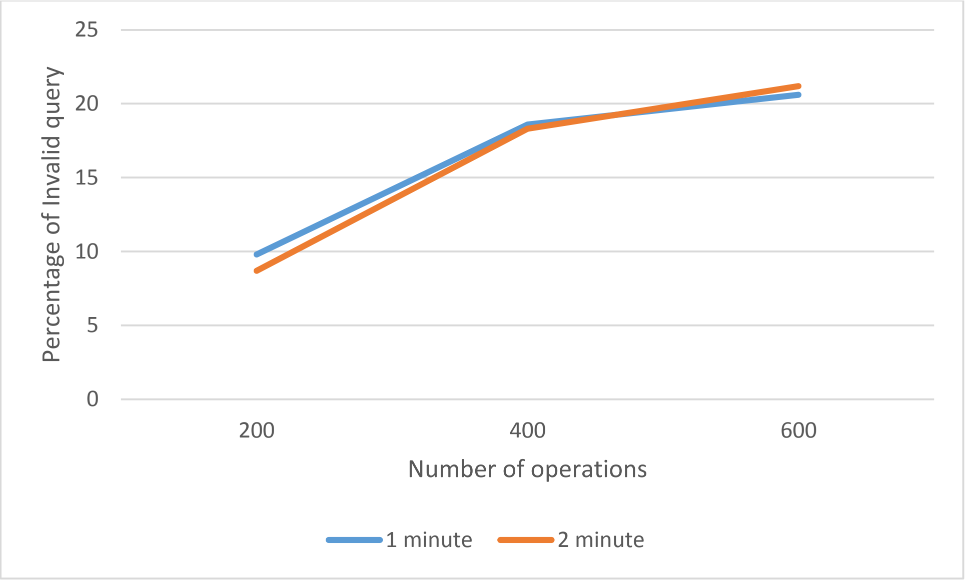
Scenario: Four peers searching and downloading and the remaining peers modifying files:

Push:

Number of Operations	Percentage of invalid query
200	9.7
400	11.5
600	14.7
800	18.7
1000	22.5



Number of operations	Percentage of invalid query	
	1 minute	2 minute
200	9.8	8.7
400	18.6	18.3
600	20.6	21.2



We observe that the as the number of requests increases, the percentage of invalid queries also increases.

Comparison between Push and Pull approaches:

Push Approach:

- Invalidate messages are passed down from the origin server to the leaf nodes.
- The origin server do not ask the other leaf nodes which object to push, so in presence of multiple origin servers and non-structured systems, the push approach leads to inefficiencies.
- In the implemented assignment the push approach is inefficient in terms of bandwidth usages, i.e. the origin server does not keeps the track of the peers which have the cached copy of the file, it broadcasts the invalidate message to every peer in the network.

  Applications: Push is normally associated with structured, tree based topologies.
  Example: Rolling out of software upgrades.

  Advantages:
- It is simple and stateless.
- It uses the underlying message routing framework.
- Provides strong consistency in the absence of failures.
- Pushing new modified versions of files improves the reliability.

Pull Approach:

- Pull is the opposite of push approach, where in leaf node polls the origin server without knowing the origin server has the object or not.
- Starvation may occur if the leaf node cannot find the origin server.

  Applications: Pull is normally associated with swarming, unstructured systems where leaf nodes seek for multiple parents.
  Example: Bitmap for a video playout buffer; 0 (missing chunk) 1 (available chunk).

  Advantages:
- The leaf node asks for new object from the origin server which is relatively efficient as compared to broadcasting invalidate messages to all the leaf nodes as seen in push approach.
- No multiple messages reach the same node.
- New object is requested only when it is needed.

Conclusion:

The percentage of invalid query results are increasing with the increase in number of operations (file requests).

By observing the plotted charts, the push based approach is giving slightly less number of invalid queries as compared to pull based.