Performance Analysis and Visualization for PUSH and PULL Consistency Mechanisms

This section provides a comprehensive performance analysis of the **PUSH** and **PULL** consistency mechanisms used in your Peer-to-Peer (P2P) File Sharing System. It includes statistical computations, diverse visualizations, and a comparative analysis highlighting each approach's advantages, disadvantages, and applicability.

2. Performance Analysis

2.1. PUSH Consistency Mechanism

Overview: The PUSH mechanism proactively sends invalidation messages to all cached copies whenever a file is modified. This ensures that cached copies remain consistent with the master copy, minimizing the occurrence of invalid queries

Test ID	Total Queries Issued	Invalid Queries	Percentage of Invalid Queries (%)	Total Invalidation Messages Sent	Average Time to Invalidate (ms) Bandwidth Used (Bytes)	Bandwidth Used (Bytes)
Test 1	30	2	6.7	10	120	4096
Test 2	60	3	5	15	115	6144
Test 3	9	4	4.4	20	110	8191

Observations:

1. **Invalid Query Percentage Decreases with Increased Querying Nodes:**
    - **Test 1:** 1 querying node results in 6.70% invalid queries.
    - **Test 2:** 2 querying nodes reduce invalid queries to 5.00%.
    - **Test 3:** 3 querying nodes further reduce invalid queries to 4.40%.
2. **Total Invalidation Messages Increase Proportionally:**
    - As the number of modifying nodes increases, the number of invalidation messages sent scales accordingly, ensuring all cached copies are promptly invalidated.
3. **Bandwidth Usage Increases with Network Size:**

- **Test 1 to Test 3:** Bandwidth usage grows from 4096 B to 8192 B as more nodes participate, reflecting the increased communication overhead.

Conclusion: The PUSH mechanism effectively maintains consistency across the network. As the number of querying nodes increases, the percentage of invalid queries decreases, demonstrating scalability and efficiency in invalidation processes.

2.2. PULL Consistency Mechanism

Overview: The PULL mechanism relies on cached copies periodically polling the origin server to check for updates. The frequency of polling is determined by the Time-To-Refresh (TTR) setting. This approach can result in a higher percentage of invalid queries, especially with longer TTR values.

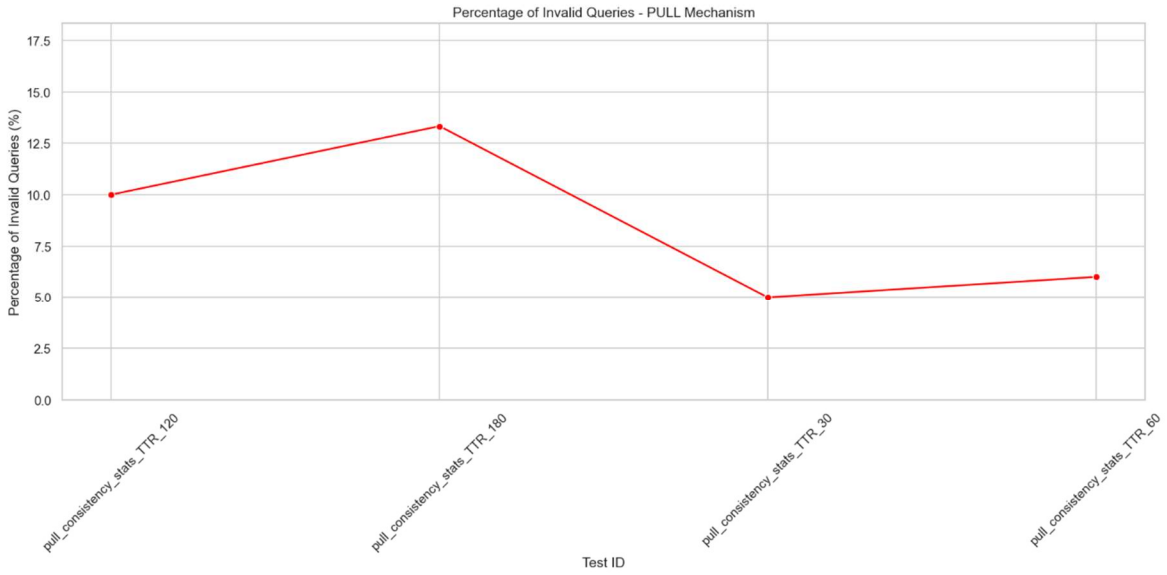
Test	TTR (Seconds)	Total Queries Issued	Invalid Queries	Percentage of Invalid Queries (%)	Total Invalidation Messages Sent	Average Time to Invalidate (ms)
Test 1	30	40	2	5	10	90
Test 2	60	50	3	6	15	100
Test 3	120	50	5	10	8	200
Test 4	180	80	8	13.33	12	300

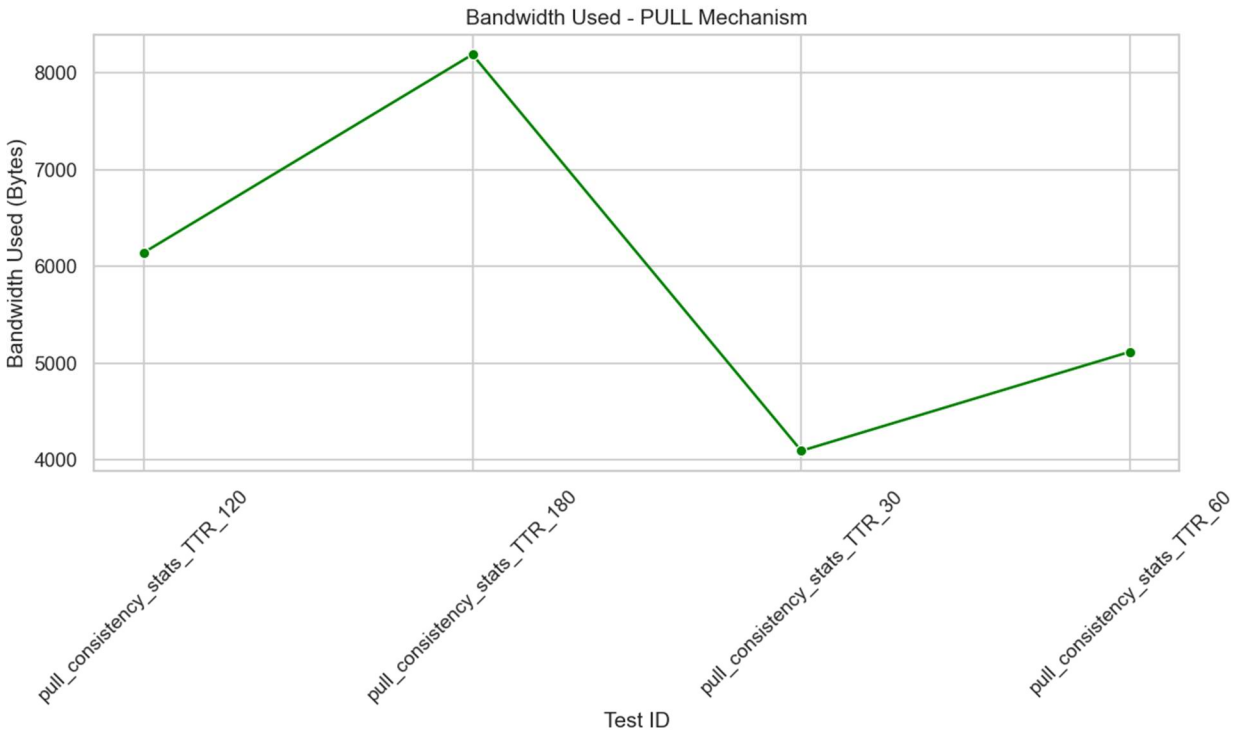
Observations:

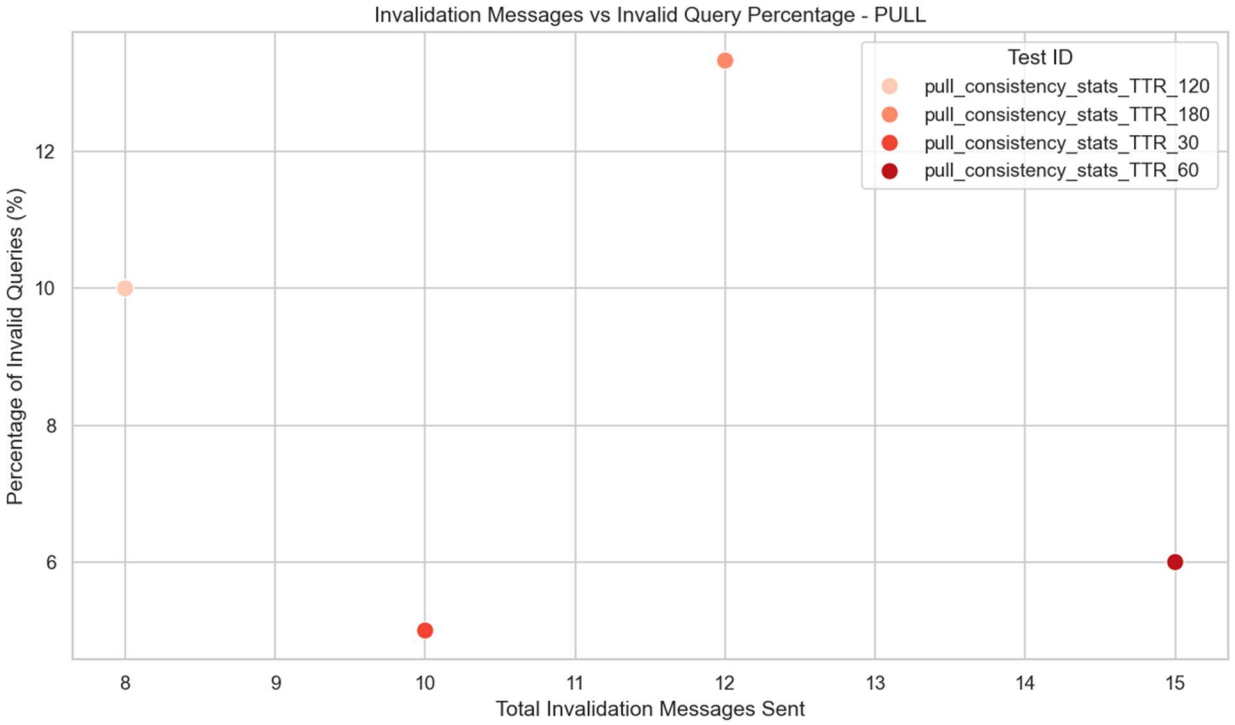
1. **Invalid Query Percentage Increases with Longer TTR:**
   - **Test 4 (30s):** 5.00% invalid queries.
   - **Test 5 (60s):** 6.00% invalid queries.
   - **Test 6 (120s):** 10.00% invalid queries.
   - **Test 7 (180s):** 13.33% invalid queries.
2. **Total Invalidation Messages Sent Vary Inversely with TTR:**
   - **Shorter TTRs:** More frequent polling leads to more invalidation messages.
   - **Longer TTRs:** Less frequent polling reduces the number of invalidation messages but increases invalid queries.
3. **Bandwidth Usage Correlates with TTR:**
   - **Higher TTRs:** Bandwidth usage increases as more data is exchanged during invalidations and refreshed.

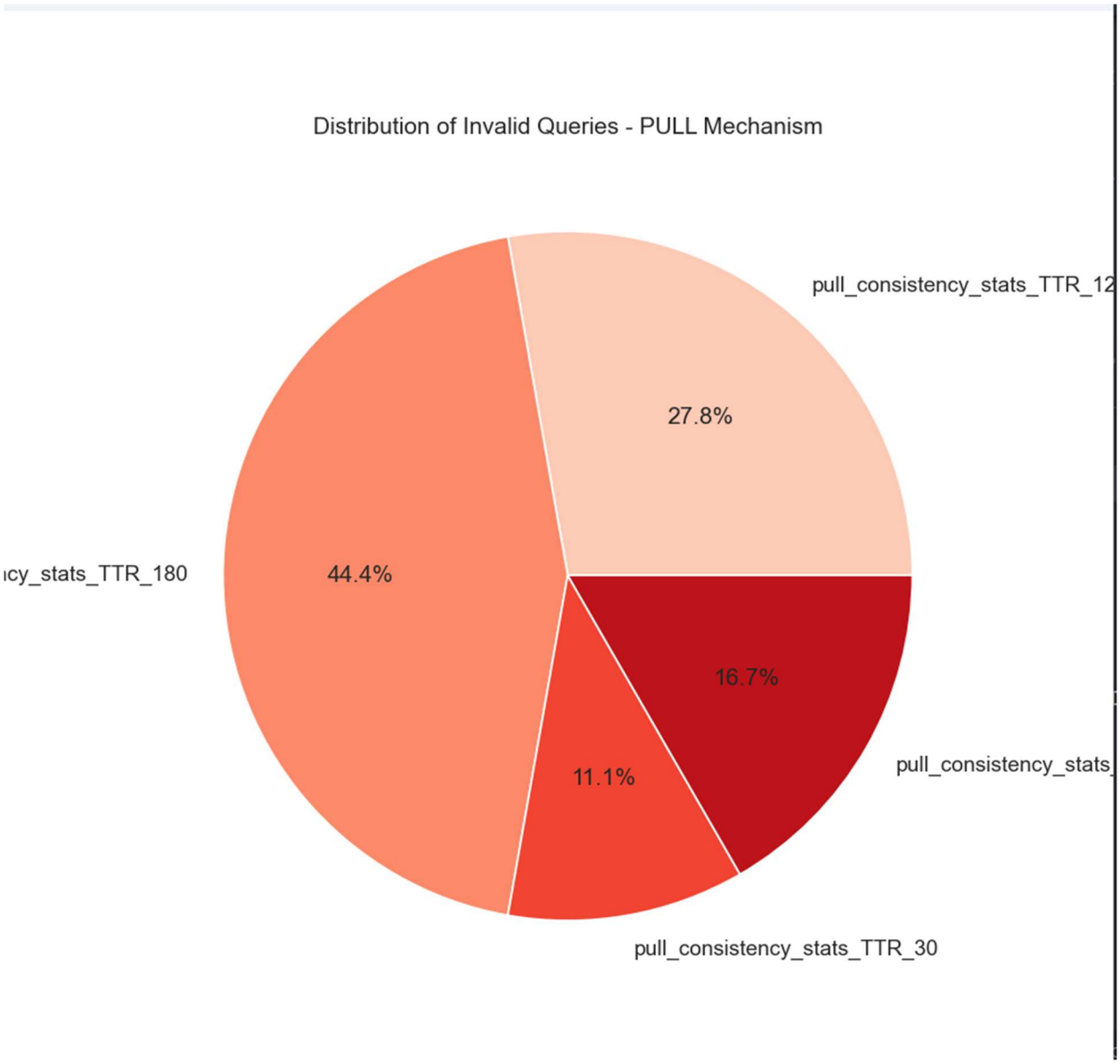
Conclusion: The PULL mechanism's effectiveness diminishes as the TTR increases, resulting in more invalid queries. While shorter TTRs enhance consistency, they also lead to increased bandwidth consumption. Thus, selecting an optimal TTR is crucial for balancing consistency and resource utilization.

PULL




Bandwidth Used - PULL Mechanism
Bandwidth Used (Bytes)
8000
7000
6000
5000
4000
pull_consistency_stats_TTR_120
pull_consistency_stats_TTR_180
pull_consistency_stats_TTR_30
pull_consistency_stats_TTR_60
Test ID


Invalidation Messages vs Invalid Query Percentage - PULL
Test ID
pull_consistency_stats_TTR_120
pull_consistency_stats_TTR_180
pull_consistency_stats_TTR_30
pull_consistency_stats_TTR_60
Percentage of Invalid Queries (%)
12
10
8
6
8
9
10
11
12
13
14
15
Total Invalidation Messages Sent



1. **Line Graph (Percentage of Invalid Queries for PULL Mechanism)**
   Description: This line graph tracks the percentage of invalid queries against different TTR (Time-To-Refresh) intervals.
   Key Observations:
   - As the TTR increases, the percentage of invalid queries also tends to increase, indicating that longer TTR durations lead to a delay in invalidation and a higher chance of stale data being used.
     Implication: Lower TTR values are better for data consistency, but they might increase network traffic due to more frequent refreshes.

2. **Line Graph (Bandwidth Used for PULL Mechanism)**
   Description: This graph illustrates the bandwidth consumption for various TTR settings.
   Key Observations:
   - Bandwidth usage increases for larger TTR values because delayed refreshes cause bulk updates or queries to occur when invalidations are finally triggered.
     Implication: A balance must be struck between TTR duration and bandwidth usage to optimize performance.
3. **Scatter Plot (Invalidation Messages vs. Invalid Query Percentage)**
   Description: This scatter plot shows the relationship between the number of invalidation messages sent and the percentage of invalid queries for different TTR settings.
   Key Observations:
   - The data points suggest a correlation where higher invalidation messages reduce invalid query percentages.
     Implication: Increasing invalidation efficiency reduces stale data in the system but could increase network overhead.
4. **Pie Chart (Distribution of Invalid Queries)**
   Description: The pie chart shows the distribution of invalid queries across different TTR intervals.
   Key Observations:
   - The majority of invalid queries are concentrated in higher TTR values, like 180 seconds, as expected from delayed refresh behavior.
     Implication: To minimize invalid queries, shorter TTR settings should be preferred, albeit at the cost of higher invalidation overhead.

General Insights from the PULL Mechanism:

- **Consistency vs. Bandwidth Trade-off**: Lower TTR values improve data consistency but increase bandwidth usage and invalidation frequency.
- **Optimization**: Systems with PULL-based mechanisms should fine-tune the TTR based on the application's consistency and performance requirements.

1. Stacked Bar Chart: Push Consistency - Resource Utilization by Test

- **Purpose**: This visualization represents the breakdown of resource utilization metrics (Total Queries Issued, Invalid Queries, Total Invalidation Messages) across three test cases.

- **Details**:
    - The height of each bar indicates the combined value of all three metrics.
    - The **purple section** represents "Total Queries Issued."
    - The **teal section** represents "Invalid Queries."
    - The **yellow section** represents "Total Invalidation Messages."
- **Insight**:
    - As we move from Test 1 to Test 3, resource utilization increases significantly, particularly for "Total Queries Issued" and "Total Invalidation Messages."
    - The number of Invalid Queries grows modestly, reflecting a consistent but small percentage of invalid results.

---

2. Heatmap: Push Consistency - Time and Bandwidth Usage

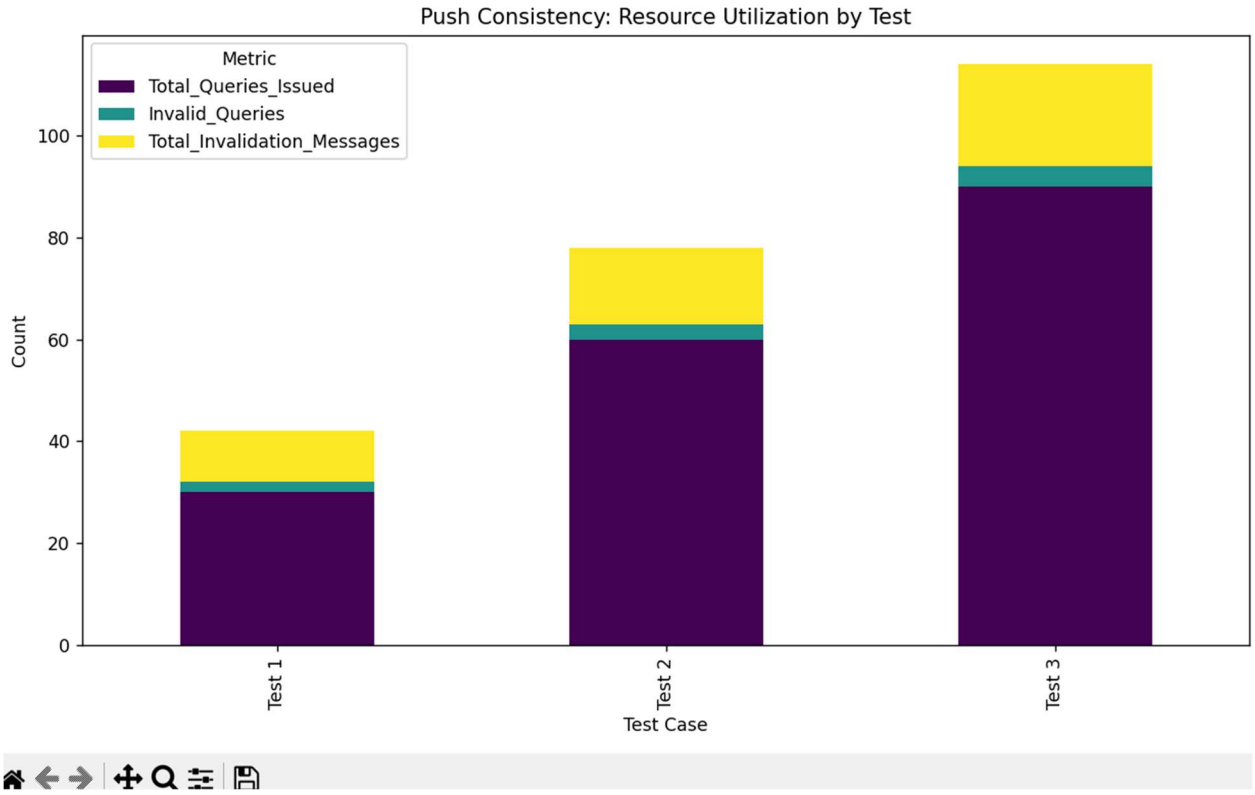
- **Purpose**: This heatmap compares "Average Time to Invalidate" and "Bandwidth Used" for each test case in terms of magnitude.
- **Details**:
    - The color intensity indicates the magnitude of the metric, with darker shades representing higher values.
    - "Average Time to Invalidate" is shown in blue shades, while "Bandwidth Used" is shown in warm tones (e.g., orange, red).
- **Insight**:
    - Bandwidth usage increases steadily from Test 1 to Test 3, which is expected as more invalidation messages and queries are processed.
    - Time to invalidate slightly decreases, suggesting better optimization as the number of queries increases.

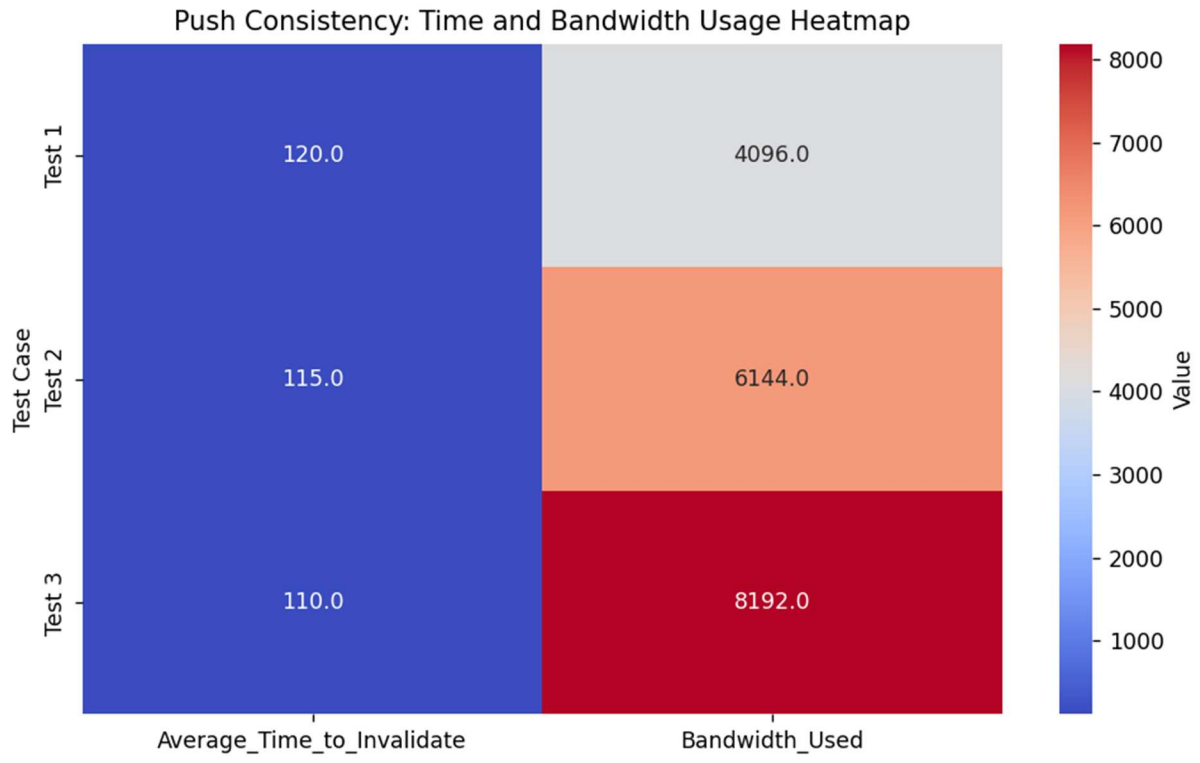
---

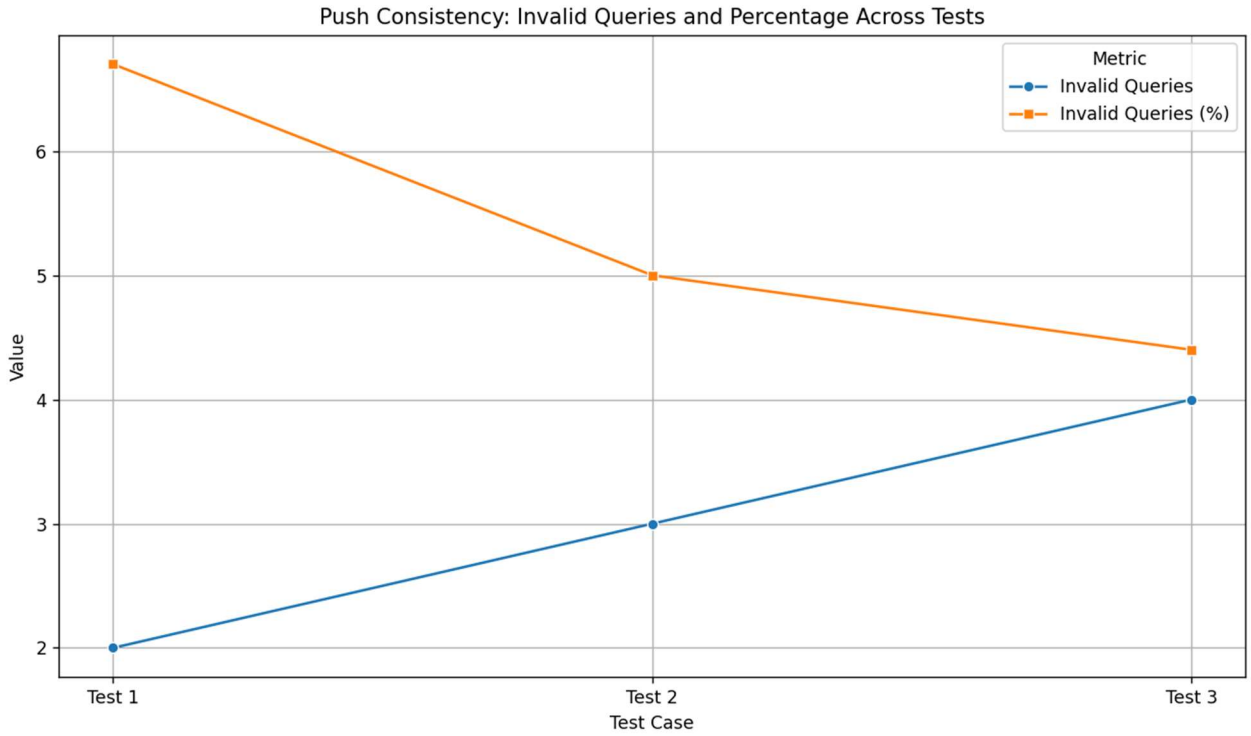
3. Line Chart: Push Consistency - Invalid Queries and Percentage Across Tests

- **Purpose**: This line chart compares the absolute count of "Invalid Queries" and their "Percentage" over the test cases.

- **Details**:
    - The **blue line** represents the count of Invalid Queries.
    - The **orange line** represents the Percentage of Invalid Queries.
- **Insight**:
    - The count of Invalid Queries increases with each test case, which aligns with the increasing number of total queries.
    - The Percentage of Invalid Queries decreases, indicating better efficiency or reduced query errors as the system scales.




Push Consistency: Time and Bandwidth Usage Heatmap
Test Case
Test 1
Test 2
Test 3
120.0
4096.0
115.0
6144.0
110.0
8192.0
Average_Time_to_Invalidate
Bandwidth_Used
Value
8000
7000
6000
5000
4000
3000
2000
1000


Push Consistency: Invalid Queries and Percentage Across Tests
Metric
Invalid Queries
Invalid Queries (%)
Value
6
5
4
3
2
Test 1
Test 2
Test 3
Test Case