# Test Driven Development (TDD) Works

The Engineering interviewing process, when exploring white-board algorithm implementations, often reveals aspects of engineering not considered by either the interviewer or candidate prior to the actual interaction. The two most common ignored issues in my experience are: a lack of testing approach when white boarding, and sometimes performance considerations.

I would like to use a recent interview experience to highlight both topics, and consider TDD in more detail. We will first review the design session interactions.

## Initial Problem

We must count unique client IP references to our web-site over some time period (a day let’s say). Note that we are only considering 32 bit IP address. We are not required to persist the IP access counts outside of the session. We will need to support various query types (e.g., counts for a specific subnet of the client URL.)

## Initial Proposed Solution

Use an In-Memory data grid to record client references ([Client-URL, Access Count] pairs.) A Data Grid platform offers a mechanism to store counts as necessary, and handles both distributing counting across monitored servers, and an aggregated view of the counts. The platforms support complex queries as well. Please review reference #1 for a background on Data Grid platforms.

## The Modified Problem

Imagine we have a single server and we must provide our own solution. What would be an appropriate data structure to store client URL and access counts? How might we query for subnet usage accounts? A subnet is a group of 32 bit client IPs beginning with the same bit pattern. Please see reference #2 for an explanation of subnets.

## Resources

1. An in-memory data grid overview and examples: <https://www.predictiveanalyticstoday.com/top-memory-data-grid-applications/>.
2. Subnet explanation: <https://www.pcwdld.com/subnet-mask-cheat-sheet-guide>.
3. GitHub code repository for this article: xxxxxxxx.