Reach Calculation on Eskimi DSP

Our DSP (Demand-Side Platform) includes a reporting system designed to track various metrics such as bids, impressions, and clicks. One important feature is the ability to calculate **unique reach**—the number of **distinct users** exposed to an ad—over time or throughout a campaign.

To achieve a balance between cost efficiency and performance, our platform uses **Redis's HyperLogLog (HLL)** data structure for estimating unique reach. While this provides fast and memory-efficient approximate cardinality calculations, it introduces a small margin of error, which is a known behavior described in Redis's documentation.

How Redis HyperLogLog (HLL) Works

HyperLogLog is a probabilistic data structure used to estimate the number of unique elements in a dataset. Unlike traditional methods that store each element explicitly (which could require large amounts of memory), HLL works by hashing elements and using a clever algorithm to approximate the count of unique items. The benefit is that HLL can store and process large datasets with a memory footprint of only 12 KB, regardless of the number of elements.

For reach calculations on our platform, Redis HLL is used to count the number of unique users that have seen an ad. This allows us to quickly and efficiently calculate reach without the need for expensive, memory-intensive processes.

Minor Discrepancy and Its Impact on Reporting

Because HLL is an approximation algorithm, there can be a slight margin of error in its calculations, typically around 0.81% in Redis. This means that while HLL provides a close estimate of unique users, it may not always be perfectly accurate.

Example: Impression Capping and Reach Discrepancy

Consider a scenario where a campaign has strict impression capping, such as 1 impression per user. In this case, you might expect the reach to be exactly equal to the number of impressions (since each user should see only one ad). However, due to the nature of HLL's probabilistic

counting, there could be a slight discrepancy between the reported reach and impressions.

For example:

- Expected reach = impressions: 1,000 unique users are shown 1 impression each.
- Reported reach using HLL: Could be slightly less (e.g., 995) or slightly more (e.g., 1,005) than the exact number of impressions due to the algorithm’s margin of error.

Why This Small Discrepancy Exists

This discrepancy is inherent in how the HyperLogLog algorithm estimates counts. The algorithm trades off perfect accuracy for speed and efficiency, which is why the memory footprint remains small even for large datasets. The small error is generally negligible but becomes noticeable in edge cases, such as when the number of impressions per user is strictly controlled.

Key Takeaways

- Efficient Reach Estimation: Our DSP platform leverages Redis HLL to efficiently calculate unique reach with minimal memory usage.
- Small Margin of Error: Redis HLL introduces a small error (~0.81%), meaning that reach estimates may not always be exact.
- Impression Capping Consideration: In scenarios with strict impression capping (e.g., 1 impression per user), small discrepancies between reported reach and impressions may occur due to HLL’s approximate nature.

While this method offers significant performance and cost benefits, it’s important to be aware of the potential for minor discrepancies in unique reach estimates. This is a natural outcome of using Redis HLL and is within the expected error range.

For cases where absolute accuracy is critical, alternative counting methods may be considered, but they typically involve higher memory and processing costs.

Revision #3
Created 20 September 2024 12:44:05 by Arūnas Butėnas
Updated 25 September 2024 06:28:52 by Arūnas Butėnas