**Subject:** Heads-up: Why reports feel “cold” after refresh (Fabric memory behavior) + our mitigation

Hi team,

Since last week we’ve been seeing **sporadic slow queries and “cold starts,” even on Import (InMemory) tables**. I’ve investigated the issue and found the **root cause**. This note explains what’s happening, why it’s new, and what we’re doing about it.

## **Context**

As you know, our company recently **migrated our Power BI Premium capacity (“classic Power BI nodes”) into Microsoft Fabric**. We were told the product experience would remain the same — reports, datasets, and the service all continue to look and feel familiar. However, behind the scenes the **semantic model engine has changed**. Fabric uses a **new unified engine** that introduces some subtle but important differences in how memory is managed.

## **What changed**

In classic Power BI, Import tables stayed fully resident in memory after refresh until they were explicitly unloaded. In Fabric, the engine uses **on-demand (lazy) loading** and **fine-grained eviction** at the column level — even for Import tables.

This means:

* After a refresh (or when Fabric needs RAM for other workloads), columns that haven’t been queried recently may be **unloaded from memory**.
* The first user query against those columns has to **reload them into memory**, which is why it feels slower (“cold start”).
* Eviction can also happen **even without a refresh** if the capacity is under pressure, because memory is shared across all workloads in Fabric (semantic models, Lakehouse/Warehouse, Spark, dataflows, etc.).
* Importantly, **an entire semantic model can be evicted or “go to sleep” after a period of inactivity**, requiring it to be reloaded when queried again. This also causes the first query after idle time to run slower until the model is back in memory.

Microsoft documents this behavior clearly for **Direct Lake** models (refresh can evict all resident columns; first queries are slower while data is paged back in), and we are now observing the same principle applied more broadly in Fabric.

You may also see this warning in VertiPaq Analyzer / DAX Studio:

*“Column not resident in memory, statistics may not be accurate.”*  
 This is the engine telling us the column has been evicted and will be reloaded on demand.

## **Why this impacts Marvel’s experience**

* **After refresh**: commonly used columns start “cold,” so the first few queries run slower while memory warms up.
* **During the day**: if capacity pressure rises, **columns or entire models** can be evicted and need to be reloaded.
* **After inactivity**: if reports aren’t used for a while, the model can “go to sleep” and will need to be brought back into memory before queries perform normally.

This explains the sporadic slowness Marvel has reported since the migration.

## **What we’ve shipped (mitigations)**

Given the criticality, we’ve added two capabilities to our refresh service:

1. **Targeted warm-up after each refresh**  
    Immediately after refresh, we run lightweight DAX queries to **pre-load a subset of high-value attributes** and our main measure (**Report Value**). This ensures the most common report paths are already “hot” when users interact.
2. **Keep-alive operation**  
    On a schedule and/or on demand, we run a minimal query to **keep the model active** and quickly re-warm key columns if they’ve been evicted due to inactivity or capacity churn. This reduces “first-click” delays for users returning after idle periods.

We’ll tune the warm-up set based on usage telemetry so we maximize user responsiveness while staying memory-efficient.

## **What you can do / What we’ll monitor**

* If you notice a slow first interaction right after refresh or after the model has been idle, try again — performance should normalize once memory is warmed.
* Please continue flagging slow time windows; we’ll correlate with **Fabric Capacity Metrics app** data and adjust warm-up coverage as needed.
* We’ll also explore capacity scheduling/isolation to reduce competition from other workloads if necessary.

## **Further reading**

* Large semantic models: On-demand load & evictions (Microsoft Learn)
* Direct Lake behavior (Microsoft Learn)
* Power BI on-demand load announcement (Microsoft blog)
* Capacity monitoring (Microsoft Learn)

**Bottom line:** Although we were told nothing would change when moving to Fabric, the **engine behind our Import models has changed**, and that’s why we’re seeing these new cold-start behaviors. We’ve already implemented **post-refresh warm-up** and **keep-alive** functionality in our refresh service to mitigate the impact and keep the user experience smooth. We’ll keep refining this as we monitor Marvel’s feedback and capacity telemetry.

Thanks,  
 Julio  
 Aggregation Platform / Market Risk