data virtuality

Reengineering Data Virtualization -
Why the question isn't "Caching vs Replication" BUT "Caching and Replication"

When it comes to data virtualization[1] the following three concerns are often raised:

1. Data virtualization is putting an additional load on the data sources.
2. Data virtualization performs badly and isn't scalable.
3. Data virtualization lacks capabilities for data historization/data cleansing/master data management/batch data import.

Most data virtualization providers try to address these concerns with ***caching***. No doubt that caching is an essential component that brings data virtualization to life. It turns the theory into a practically usable technology. And for the first two aspects, caching surely helps - to a certain degree.

However, to fully address all three aspects, ***data replication*** is needed. Below is an overview of how the concerns are addressed by the two approaches, caching and replication:

	Caching	Replication
Load on the data sources	TTL (time-to-live) logic of the cache can be problematic from the perspective of the source systems: the refresh of data puts load on the source systems at the times when they may be busy with other high priority work.[2]	Schedules allow full control of when and how the updates are happening, and thereby, of the load put on the data sources. To further reduce the load on the data source, see Change Data Capture (CDC) capability below.

[1] Definition of **data virtualization** and **data federation**:
From the ***technical standpoint***, data federation refers to a subset of data virtualization which concentrates on the performant querying of data across disparate databases. Data federation is typically limited to integrating databases, whereas data virtualization expands the connectivity agnostically to include any flavor of RDBMS, data appliance, NoSQL, Web Services, SaaS, and enterprise applications and also includes moving and transforming the data.

Having said that, vendors often use the term data virtualization to refer to their offerings which mostly concentrate on data federation. This leads to the wide perception that data virtualization is a "newer" ***marketing term*** for data federation. In this paper, we will use the marketing led definition of data virtualization. But we will also demonstrate how the concept of data virtualization actually goes far beyond the data federation part.

[2] Many data virtualization vendors have understood the shortcoming of classical caching and therefore, use replication-like techniques instead, for example to refresh data storage on schedule or keep the complete large data sets in the storage. However, they still use the word caching to describe the process for marketing purposes.The topic of historization and cleansing is not addressed by this.

Performance and scalability	Performance challenges of processing small data sets can be solved but not for large data sets. Large data sets need special performance optimizations for loading as well as processing which are not provided by caching. Also, normally cache is supposed to be smaller in size than the original data. This limits the usefulness of the cache when it comes to analytical operations across the entire data set.[3]	By replicating the data into the storage and/or by optimization through precalculating the results (of common calculations), the performance can be increased in the most efficient way - especially for large data sets.
Historization and transformation/cleansing possibilities	Caching lacks any historization and transformation possibilities.	Replication facilitates data transformation, data historization, slowly changing dimensions, as well as data cleansing capabilities and enables further possibilities such as CDC load.

Let's take a closer look at the shortcomings of caching in the data virtualization set up:

Additional Load on the Data Sources

As the word caching suggests, the times when caches are updated depend on how stale, i.e. outdated the data in the cache are. However, to reduce the load on the data sources, it needs to be the other way around: the refreshes need to be controllable and should be determined by the requirements of the source systems - not by the cache.

Conclusion: Caching is only a temporary solution as in many cases scheduled updates are better. Some data virtualization vendors acknowledge this problem and provide external solutions that enable users to schedule the cache refreshes. Effectively, their caches thereby use replication logic rather than cache logic. This work-around isn't fully embedded in the actual data virtualization solution which makes the processes complex. This is one of the pitfalls that should be avoided during the vendor selection process.

Bad Performance and Scalability

The performance challenges resulting from processing small data sets can certainly be solved with caching. However, for larger data sets, caching is not adequate for three key reasons:

a) it leaves very little control and flexibility to how the data is loaded and stored.
b) it lacks the means that are provided by analytical databases to speedup the calculations of aggregations
c) it also lacks the capabilities of indexing and pre-calculating results of common data operations such as JOINs

Conclusion: Again, caching is better than nothing. But to ensure good performance when processing large data sets, caching is insufficient. To achieve the best performance and scalability, you need more control over the structure of the data by creating indexes, distribution keys, transparent aggregations, etc. which requires a solution beyond caching.

[3] Footnote 2 also applies to this part of caching.

Lack of Storage and Transformation Capabilities (for historization, slowly changing dimensions, cleansing etc.)

This concern which includes batching data import, data historization, and complex multi-step data transformations is totally disregarded by data virtualization, with or without caching. But for every data integration project, especially the modern ones, this issue needs to be addressed. Importing flat files from FTP, matching and cleansing customer address sets from two different source systems, tracking changes in employee responsibilities over time etc. are only a few operational business requests that cannot be solved with pure data virtualization without the data storage and transformation capabilities.

Conclusion: Due to the lack of historization and cleansing capabilities, data virtualization with caching alone cannot fulfill all the business needs. A solution to be carefully considered is the combination of data virtualization and data replication with its different shades (ETL, ELT, CDC,...).

This is why, when it comes to selecting a data virtualization solution, the question isn't caching vs replication but rather caching AND replication in a single tool.

Gartner®

> A mix of data integration approaches thus becomes crucial, spanning physical delivery to virtualized delivery, and bulk/batch movements to event-driven granular data propagation. In particular, when data is being constantly produced in massive quantities and is always in motion and constantly changing (e.g., IoT platforms and data lakes), attempts to collect all of this data are potentially neither practical nor viable. This is driving an increase in demand for connection to data, not just the collection of it.
>
> Gartner, Magic Quadrant for Data Integration Tools 2019

How caching in combination with advanced data materialization[4] and data replication[5] capabilities solve the three concerns:

[4] **Advanced data materialization** means producing a shadow copy of a source table or virtual view in the central storage. This copy is managed and updated in a fully transparent and automated way.

[5] **Data replication** is the creation of new tables in central storage which represent the data from the source systems in a transformed, historized or cleansed form.

Full Control over the Load on the Data Sources

With advanced data materialization and replication, you have full control over the update schedules and can adjust them according to the requirements of the source systems. For some data sources, an even better approach is to replicate the data out of the sources using Change Data Capture (CDC) techniques.

Increased Performance and Scalability

The ability to replicate the data, in addition to caching, provides the most effective way to increase the performance of the platform. Data that uses a lot of performance during a query are either cached, materialized in a database or replicated in a database, depending on the use case. Furthermore, indexes, distribution keys, transparent aggregations, etc. can be created.

Facilitated Storage and Transformation (historization, slowly changing dimensions, cleansing etc.) Possibilities

No doubt that data transformation, data historization, slowly changing dimensions, data cleansing etc. can be facilitated by ETL/ELT[6]. Additionally, a lot of Master Data Management (MDM) challenges such as data cleansing and data field normalization can be solved by having a central storage and by applying Procedural SQL[7] to harmonize the master data directly within the data integration platform.

[6] **Extract, transform, load (ETL) and Extract, load, transform (ELT)** are general procedures of copying data from one or more sources into a target storage which represents the data differently from the source(s) or in a different context than the source(s). ETL/ELT are a type of data replication.

[7] Data virtualization is often thought of as an approach to just query data across different data sources with SQL, so primarily thinking about the DQL subset of SQL (SELECT statement). However, we see that a lot of data management challenges can be solved by also applying the DML (UPDATE and INSERT) part of SQL, and even more challenges like MDM and data cleansing can be solved by applying Procedural SQL, this is why we support all these capabilities in our data virtualization solution.

Conclusion

The shortcomings of caching described above, especially when talking about analytical use cases might give the impression that caching is bad and should not be totally avoided. But that's certainly not the case.

The key is how to reengineer data virtualization and ask how to best implement caching AND replication. This includes knowing when to use cache and when not. Some of the use cases realized by our customers when caching is the best way to go include:

- Operational/transactional use cases
  The amount of data involved in each query is rather small but the number of queries per second is very high. Therefore, it is an ideal case for caching. However, when the number of queries is high BUT the amount of data involved is high as well, we recommend our customers to materialize data in a high-performance in-memory database like Exasol, SingleStore (formerly MemSQL) etc.

- Data sources have fine-grained permission, data visibility controls or fine-grained permission enforcement is delegated by DV to the data source (for example row-level permissions enforced on the data source side).

  For these use cases, user-scoped in-memory caching of the data can provide a good performance boost, given that each user may see their own variant of the source data. On the other side, replication is needed to ensure high performance, to facilitate data historization, and enable activities like data cleansing, master data management, etc.

That's why the Data Virtuality Data Platform combines the two approaches: caching and replication - to make a real difference and enable you to quickly respond to ever changing business needs and competitive threats.

About Data Virtuality

- Founded:
  2012 by Nick Golovin (PhD) in Leipzig, Germany after 8 years of research
- Offices:
  Munich, San Francisco, Leipzig
- Solutions:
  Data Virtuality Platform SaaS
  Data Virtuality Platform On-Premises
  Data Virtuality Pipes Professional
  Data Virtuality Pipes
- Acknowledgements:
  Honorable Mention in 2022 Gartner Magic Quadrant for Data Integration Tools
- Awards:
  Most Innovative Data Management Provider 2022, 2021 and 2019 (A-Team Insights)
  2020 and 2019 Deloitte Technology Fast 50

Message: info@datavirtuality.com
Visit: datavirtuality.com
Request Demo: demo@datavirtuality.com
Data Virtuality Platform SaaS Free Trial: https://eu.platform.datavirtuality.com/#/start-trial

data virtuality