MCG GLOBAL SERVICES

Modernize Data Warehousing

Beyond Performance

The Importance of Other Key Attributes

Prepared by:
William McKnight
McKnight Consulting Group
www.mcknightcg.com
January 2021

Contents

Introduction

Performance is a critical point of interest when it comes to selecting an analytics platform because it ultimately impacts total cost of ownership, value, and user satisfaction.

To measure data warehouse performance, we use similarly priced specifications across data warehouse competitors. In the emerging world where the specifications are obfuscated, or at least unpublished, comparison can be made on price.

Usually when people say they care about performance, it is the ultimate metric of price/performance that they mean. Or they should. Price/performance equalizes competitors. We calculate price-per-performance using the formula Elapsed Time (seconds) x Cost of Platform ($/hour) / 3,600 (seconds/hour) where the elapsed time is actually the duration of the thread. To calculate this price/performance, you need system cost. System cost can be difficult to compare because vendor platforms vary in their pricing and licensing models, but it can, and must, be done in an evaluation of a mission-critical data warehouse workload with a platform decision.

The realities of creating fair tests can be overwhelming to many shops, and is a task usually underestimated. So, while very important, if you start here (at price/performance), you can frequently end here in your evaluation. That would be a big mistake.

The Perils of Performance Alone

The resources and cost necessary to run queries are important when you are at the point of running a query. A modern workload is less frequently a set number of queries, but more of an interactive variable number of queries and query complexity based on conditions such as the time the analyst has to devote to the matter, its urgency, and other attributes of the platform that support the business.

The more of these key data warehouse attributes a platform has, the more valuable the database will be to any organization.

A lack of certain key features and functions in the chosen platform leads to increased time spent on tasks such as configuration, management, and tuning and a decrease in time spent on business analysis. Missing attributes in a selected platform can lead to coding workarounds, or non-intuitive implementations, that create obstacles to understanding. This also contributes to long-term resource usage and cost. A platform lacking labor or time-saving capabilities can impose tremendous costs on staffing a project. A platform limited in its analytics capabilities can lead to an incomplete picture, or the need to separately buy missing functionalities. Having these capabilities available can be an essential upside to your data warehouse modernization efforts. The more of these key data warehouse attributes that are available, the more valuable the database will be to any organization.

On the other hand, some data warehouse platforms have features that appear beneficial and desirable. However, there can be some hidden downsides to those features that many organizations do not realize until they deploy these platforms into production and experience them firsthand.

> **Example –** When a cloud database charges by compute, automatically increasing compute for the sake of performance or concurrency with autoscaling seems valuable, but using this feature means costs are also automatically increased.

The list of important data warehouse attributes that will need to be implemented one way or another in an enterprise data platform, and therefore should be evaluated along with query price/performance, is long, and will be the focus of this paper.

Cost Predictability and Transparency

The cost profile options for cloud databases are straightforward if you accept the defaults for simple workload or proof-of-concept (POC) environments. However, it can be enormously expensive and confusing if you seek the best price/performance for more robust, enterprise workloads and configurations.

Initial entry costs and inadequately scoped environments can artificially lower expectations of the true costs of jumping into a cloud data warehouse environment, just as failing to consider hardware costs can artificially lower expectations of true costs on-premises. Cost predictability can certainty only happen when the entire picture of a production analytic environment is considered. This consists of all workloads, a true concurrency profile, an accurate assessment of users, and a consideration of the durations of process execution.

Some vendors have reserved instance pricing, which can be substantially cheaper than on-demand pricing. However, reserved instance pricing can only be procured with year-plus commitments, which give the vendor some revenue assurance, and the pricing is cheapest when paid in full up front. Long commitments go counter to the agile nature of today's projects.

For some, you pay for compute resources as a function of time, but you also choose the hourly rate based on certain enterprise features you need. For the lowest level of support, there is a rate per node per hour. If you want, for example, "multi-cluster" (automatic scale out of additional clusters), enterprise-level security, and more support, there is a different rate per node per hour for that. It goes up from there.

> **Example** – If you need more Vertica clusters for concurrency support, up to 3 production clusters are included in the license. You may need more than one production cluster for other reasons. The Trade Desk has two 420-node production clusters on Amazon in two different regions to make sure latency is kept low since they have sub-second SLA's on over 10 PB of data. Some vendors would charge the same price again for the second cluster, some would charge separately for machine learning, or security.

Data professionals who used to be valued for tuning queries are now valued as much for tuning costs.

With some platforms, you pay for bytes processed and the underlying architecture is unknown. The environment is scaled automatically without affecting price. There is also a cost-per-hour flat rate where you would need to calculate how long it would take to run your queries to completion to predict costs.

Some vendors have alternative names for nodes. One “unit” on Microsoft Azure, for example, is currently 64 GB of memory, 8vCPU, and 1 TB of disk. A “slot” on Google Cloud is a logical unit of compute measurement (CPU, etc.). This could be an evolving unit of measure where hardware is incremented in the cloud.

Customers need to analyze current workloads, performance, and concurrency and project those into realistic pricing in alternative platforms. Data professionals who used to be valued for tuning queries are now valued as much for tuning costs.

Machine Learning

Today, data warehouse query languages need to be extended to include machine learning, or firms may find the programming required will be too challenging to keep pace. Machine learning completely changes the approach to data analysis. Classification, recommendation, and predictive algorithms have begun to replace much of the interaction with data, reaching even deeper insights. Where there is only business intelligence today, there will also be artificial intelligence tomorrow and the data warehouse should be a key asset on that journey – not a hinderance or a looming replacement project.

Data warehouses today need to weave machine learning into their data processing workflows to gather non-obvious insights and predictive statements about the business and customers. Such findings are key to improving product and service offerings, and maintaining efficiency and competitiveness on the market.

SQL is the long-standing language of the relational database, supported by thousands of tools and known by millions of users. Vendors must accommodate this knowledge and extend SQL to include machine learning functions and algorithms to expand the capabilities of those tools and users. The whole data science workflow needs to be supported, from data exploration and preparation to model evaluation and management.

If your database does not include machine learning, there are many extra things to be concerned with. Other components will be needed to complete the toolbox and get the job done.

Ideally, security for machine learning will be the same as database security. A user may need to be able to provision workspaces with security controls in place using machine learning templates for automation. All the associated services (Storage, Key Vault and Container Registry) that many machine learning tools work with should also support various security controls. Machine learning models should have the same role-based security and management methods as tables.

The data warehouse also needs to be able to operate at scale and into the complete depth of the data, going well beyond the sampling approach that some tools employ and into deep levels of specificity. Only complete depth will accurately reveal the patterns buried in the ever-increasing data.

Resource Elasticity

A data warehouse needs to be able to scale up and down and take advantage of the elastic compute and storage capabilities in the cloud, public or private, without disruption or delay. If there is delay in scaling, or disruption for migration or repartitioning, one of the key benefits of the cloud is lost.

The more the customer needs to be involved in resource determination and provisioning, the less elastic, and less modern, the solution is. The more granular the growth of the clusters, and the less of a stairstep approach to resources, the more elastic the solution is.

One thing to watch for in elasticity scaling is keeping the amount of money spent by the customer under the customer's control. For example, depending on how concurrency is handled in the licensing structure, automatically doubling the compute power to support twice as many concurrent workloads may double the software cost for the time period the additional cluster is up and running. If concurrency is included in the software license, then the software cost may stay constant. The customer would only owe additional cloud infrastructure fees, which is more manageable.

The more granular the growth of the clusters, and the less of a stairstep approach to resources, the more elastic the solution is.

Cost Consciousness and Licensing Structure

Look for databases with built-in, cost-conscious features. Be on the lookout for cost optimizations like not paying when the system is idle, compression to save storage costs, and moving or isolating workloads to avoid contention. Look for the ability to directly operate on compact open file formats, such as Parquet and ORC, at a lower cost than internally optimized data formats. Since multiple analytical processes can work with these formats, this ability not only helps avoid data duplication into proprietary storage formats but also provides highly cost-efficient long-term storage.

Also, costs can spin out of control if you have to pay a separate license for each deployment option or each machine learning algorithm. This is standard with some cloud machine learning options.

Finally, also consider if you will be paying per user, per node, per terabyte, per CPU, per hour, etc.. This changes the cost model, as does getting charged for “extras” like development, test and disaster recovery clusters.

Data Storage Format Alternatives

Cloud object stores, such as Amazon S3, Google Cloud Storage, etc., provide the ability to store many types of data. Cloud object storage is relatively inexpensive making data storage at high scale affordable. On-premises, specialized private cloud storage options such as Pure Flashblade, Dell EMC, or HPE Scality tend to offer similar data type storage flexibility, equivalent to Amazon's S3, and HDFS offers data format storage flexibility on commodity hardware.

Analysis of these various data formats provides timely and valuable insights to businesses. But analysis doesn't just require storage, it requires compute. On the cloud, compute resources (compute nodes/CPU's) are relatively expensive.

To take full advantage of the elasticity of the cloud without driving up costs, data warehouse compute and storage need to be scaled separately. To do this, the data warehouse software must be able to store its own data in object stores like S3, and analyze that data with independently and dynamically provisioned compute. It must have the capability to take down unneeded compute nodes when workloads decrease and scale up when they increase.

To take full advantage of the many types of data available, such as Apache ORC, Apache Parquet, JSON, Apache Avro, etc., modern data warehouses need to be able to analyze that data without moving or altering it. Obviously, any data warehouse should analyze data in its own optimized internal format, but it also needs to efficiently query as many of the other available formats as possible.

… The ability to join data between the various internal and external data formats provides the highest level of analytic flexibility.

A unified analytics warehouse that supports these various data formats means you have the benefit of querying them directly, without greatly expanding the hierarchical complex data types to a standard tabular data structure for analysis. Having to transform the data causes delay and structure bloat causes query response slowdown.

You should also be able to import data directly from these formats into the data warehouse storage format for even faster querying. Or, you should be able to export database data directly to efficient long-term storage formats like ORC and Parquet for older data that doesn't require fast querying.

The ability to join data for analysis between the various internal and external data formats provides the highest level of analytic flexibility.

Unstructured and Semi-Structured Data Support

Unstructured data has always been a valuable asset to organizations, but it can be difficult to manage. Emails, documents, medical records, contracts, design specifications, legal agreements, advertisements, delivery instructions, and other text-based sources of information historically have not fit neatly into tabular relational databases. Even many NoSQL databases and Hadoop solutions do not adequately address the specialized pre-processing, query, and organization requirements of pure unstructured text data, and instead rely on techniques that essentially structure the data first.

A modern data warehouse should be able to add metadata to unstructured text for easy search, and do text analytics on unstructured data, without first enforcing structure.

In addition, a lot of data which is of extreme value to certain use cases and organizations arrives in a constant stream of message formats.

> **Examples –** Streaming sensor data for use cases like predictive maintenance or smart buildings. Streaming call detail record data for telecom use cases like churn reduction, customer service improvement, and network optimization.

A modern data warehouse should be able to constantly ingest semi-structured message formats such as JSON, XML, and Avro in parallel with queries. Streaming data requires constant ingestion, so a data warehouse can't wait to query when ingestion is done, or schedule ingestion for slow query times. It should be able to interpret the data as columns, joining seamlessly with other columns in the database, without forcing a transformation step into the mix that adds complexity, fragility, and delay.

The presence of a robust semi-structured and unstructured text data analytics program is a telltale sign of whether or not an organization has achieved a high level of modernization. When the speed of change, complexity, and unpredictability are high, companies are turning to the insights buried in a haystack of unstructured information to try to keep up with the pace of change without drowning in data. For many companies, this data is largely unexplored territory. Thus, the need for fast, easy-to-use, semi-structured and unstructured data capabilities continues to grow.

Having semi-structured and unstructured text data manipulation capabilities in the data warehouse that are seamless with structured data manipulation capabilities allows you to express analysis more naturally, ...

Modern challenges represent a shift in how people find critical insights. Conventional analysis focuses on fast answers to predictable questions. For example, what is this month's sales by region? For years, business intelligence has driven a familiar, well-worn path of finding quantitative answers to recurring questions. However, in the face of unpredictable change and complexity, the operational strength of conventional business intelligence is less helpful. This is coming up with only part of the answer that is needed. To complete the picture, analysis of unstructured data with machine learning and advanced analytics is needed.

Having semi-structured and unstructured text data manipulation capabilities in the data warehouse that are seamless with structured data manipulation capabilities allows you to express analysis more naturally, reduce costly data transformations, and ultimately incorporate unstructured data into the organization.

Also, it's important to be able to manipulate the data wherever it is. Connectivity with other software is important. A data warehouse database that works great but doesn't connect well with standard ETL or data visualization software, for instance, would be largely useless.

Concurrency Scaling

Concurrency is affected by a number of factors and takes on many forms. The most common operation that impacts data warehouse operations is when queueing occurs due to a number of users performing analyses at the same time. Concurrency issues can manifest as slow performance, uneven performance or even query timeout. If the database has concurrency limitations, designing around them is difficult at best, and limiting to effective data usage. This reduces the value of data. If you can't run as many queries on your data as your company requires, that impacts the business value that can be derived from it. Should compute clusters be provisioned larger to begin with, or is it a matter of allowing the clusters to automatically scale up?

If the data warehouse automatically scales up to overcome concurrency limitations, this may be costly if the data warehouse charges by compute node. It can also be unpredictable and hard to forecast or budget for.

If the data warehouse charges per user, costs will also increase as the data warehouse is put to more use in the company. Becoming broadly data-driven can be a costly operation in that case, albeit far more predictably so than when charges scale automatically without customer control.

Look for a data warehouse to provide linear scaling in overall query workload performance as concurrent users are added. Some databases offer up to 3 times the licensed production capacity to support extremely high concurrency without increasing software costs.

Easy Administration

Overall costs, time, as well as storage and compute resources are affected by the simplicity of configurability and overall use. A database that is both easy to administer and flexible in handling almost any usage pattern saves on overall costs as well as storage and compute resources.

The platform should have embraced a self-sufficiency model for its customers and be well into the process of automating repetitive tasks. By leveraging automation, an enterprise relinquishes the need for granular control and enables its people to focus on usage.

Easy administration starts with setup and getting to the first prompt. This should be a simple process of asking basic information and providing helpful information for selecting the storage and node configurations.

The platform should have embraced a self-sufficiency model for its customers and be well into the process of automating repetitive tasks.

The data warehouse should support mission-critical business applications with minimal downtime. Check on “hot pluggable” components, understand system downtime requirements and any issues that might deny or degrade service to end users. These can include batch load times, node failure, software/hardware upgrades, severe system performance issues, and system maintenance outages.

The data warehouse should provide a single point of control to simplify system administration tasks. This includes making disaster recovery a simple process. The faster and more straightforward a DR process is, the less impact disasters will have.

Even though we look to modernize the data warehouse with easy administration, the platform should provide tuning capabilities, should you need them. The only alternative to having tuning capabilities is living with the performance you receive without them or increasing the specification of or number of nodes – and cost.

Optimizer Robustness

The data warehouse should be designed for complex decision support and machine learning activity in a multi-user, mixed workload, highly concurrent environment. Check on the maturity of the optimizer for supporting every type of query with good performance and to determine the best execution plan based on changing data demographics. Check on conditional parallelism and what the causes are of variations in the parallelism deployed. Check on dynamic and controllable prioritization of resources for queries. Check on time and requirements for optimal queries, such as compiling indexes or updating statistics.

Check on workload isolation capabilities, too. One of the advantages of workload isolation is that data loads and ELT workloads can be separated from analytical workloads so they don't interfere with each other. You don't have to wait till 3 AM to do batch loads. This is required for supporting streaming data loads, since loading never stops with streaming data. It also allows greater freedom to do ad-hoc queries and resource intensive acts like training a machine learning model, without concern that it will adversely impact other workloads.

Conclusion

These days, the best answer to your data warehouse selection may not be as evident as it was in the past, and clearly not as simple as measuring performance, or even price/performance, on selected queries. Given the high relevance of data analysis and information, investment and innovation continue in this area and there are a surprisingly large number of options. Many platforms may work, but at high short- and long-term prices. Making a selection consistent with your goals as an organization and with a broad and detailed view of the evaluation attributes is paramount to data warehouse modernization success.

Attributes to consider:

- ☐ Cost Predictability and Transparency
- ☐ Multi-Cluster Costs
- ☐ In-Database Machine Learning
- ☐ SQL Compatibility
- ☐ Provisioning Workloads with Security Controls
- ☐ ML Security same as Database Security
- ☐ Resource Elasticity
- ☐ Automated Resource Elasticity
- ☐ Granular Resource Elasticity
- ☐ Licensing Structure
- ☐ Cost Conscious Features
- ☐ Data Storage Alternatives

- Unstructured and Semi-Structured Data Support
- Streaming Data Support
- Connectivity with standard ETL and Data Visualization software
- Concurrency Scaling
- Seamless Upgrades
- Hot Pluggable Components
- Single Point of Entry for System Administration
- Easy Administration
- Optimizer Robustness
- Disaster Recovery
- Workload Isolation

About the Author: William McKnight

William McKnight has advised many of the world's best-known organizations. His strategies form the information management plan for leading companies in various industries. He is a prolific author and a popular keynote speaker and trainer. He has performed dozens of benchmarks on leading database, data lake, streaming and data integration products. William is the #1 global influencer in data warehousing and master data management and he leads McKnight Consulting Group (MCG), which has twice placed on the Inc. 5000 list. MCG can be reached at www.mcknightcg.com.