White paper *Technical*

Exasol

A Peek under the hood

Contents

A peek under the hood

01

Introduction

Exasol was founded in Nuremberg, Germany, in the year 2000 – a time when two trends in hardware were starting to emerge:

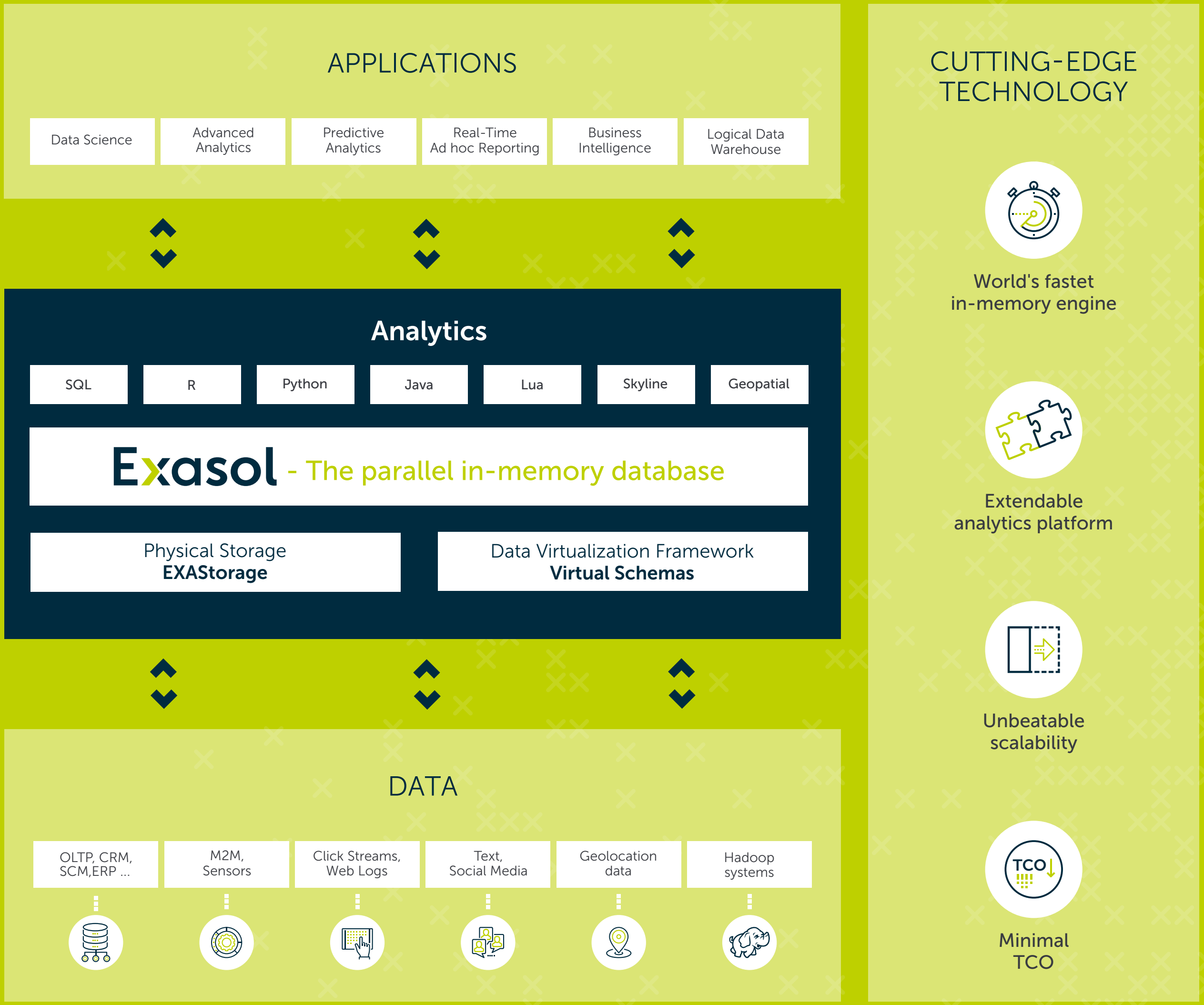
Major improvements in processing power were no longer coming from ever increasing clock speeds of central processing units (CPUs), but instead from parallel and distributed systems.

After a long period of slow improvement, random-access memory (RAM) started becoming much larger and cheaper with each successive generation.

With backgrounds in high-performance computing and scientific computing, Exasol's founders recognized that new opportunities were made possible by these trends. With RAM falling in cost and rising in capacity and cluster computing being merely a commodity, it was now conceivable to apply the principles and architectures of high-performance computing to database design. In the years that followed, the company exclusively focused on delivering ultra-fast, massively scalable analytic performance.

Today, successful companies around the world rely upon Exasol's innovative in-memory analytic database solution to run their businesses faster and smarter. The company's flagship product, Exasol, is a high-performance in-memory database designed specifically for analytics. Exasol holds performance records in the TPC-H online transaction processing benchmark from the Transaction Processing Performance Council (TPC) for decision-support databases, outperforming competitors by orders of magnitudes and scaling up to hundreds of terabytes of data.
The goal of this paper is to offer a deeper understanding of a selection of design principles, core features, and technologies that underpin Exasol's analytical performance and, at the same time, take a glimpse at what Exasol has to offer beyond raw performance.

01 Introduction



For Exasol, great performance is not an end in itself, but - as indicated by the diagram - one important characteristic of a complete analytic system.

02

Being really fast

In their quest for speed over the last two decades, Exasol's engineers came up with a number of techniques and solutions in order to achieve specific tasks with maximum performance and over the course of time a number of general principles emerged.

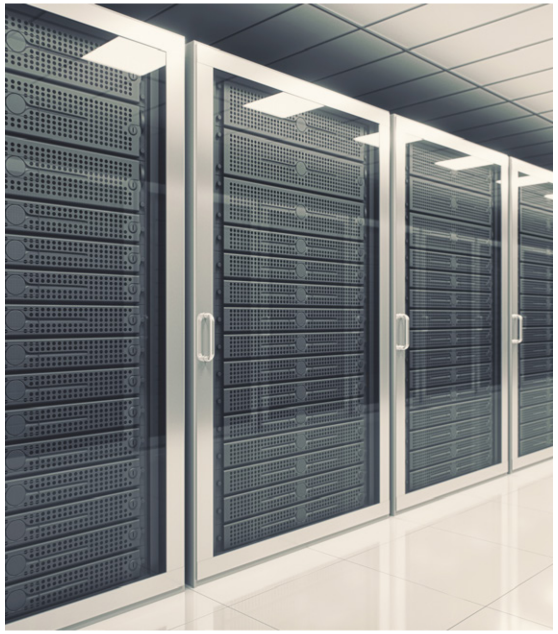
Massively Parallel Processing - MPP

The design of Exasol is inspired by system architectures from the field of high-performance computing. In order to maximize hardware usage, parallel processing and high-performance communication techniques are used at different levels of the architecture. These techniques have been designed and built from scratch exactly for the requirements of Exasol.

At the cluster level, Exasol essentially follows the SPMD (single program, multiple data) paradigm. Different machines in the cluster execute the same program, while their internal state is independent most of the time. Machines communicate via asynchronous streaming/message passing. Exasol is designed to operate without any dedicated "master node" which would constitute a single point of failure and a performance bottleneck. Instead, Exasol's cluster technology can be installed across hundreds of machines, all working in parallel to answer a query without a significant performance overhead.

Still, global synchronization is an expensive operation as all participants have to wait for the others to be in some well-defined state. Therefore, Exasol is designed to avoid global synchronizations during query processing as much as possible.

In turn, every machine is programmed according to the SMP (symmetric multiprocessing) paradigm. Processes and threads exploit the high performance features offered by modern multi-core shared-memory architectures, which results in the maximum utilization of existing standard server hardware. Per CPU, all the SIMD (single instruction, multiple data) features of modern standard processors are utilized, which allows Exasol to deliver the highest performance rate at every level.



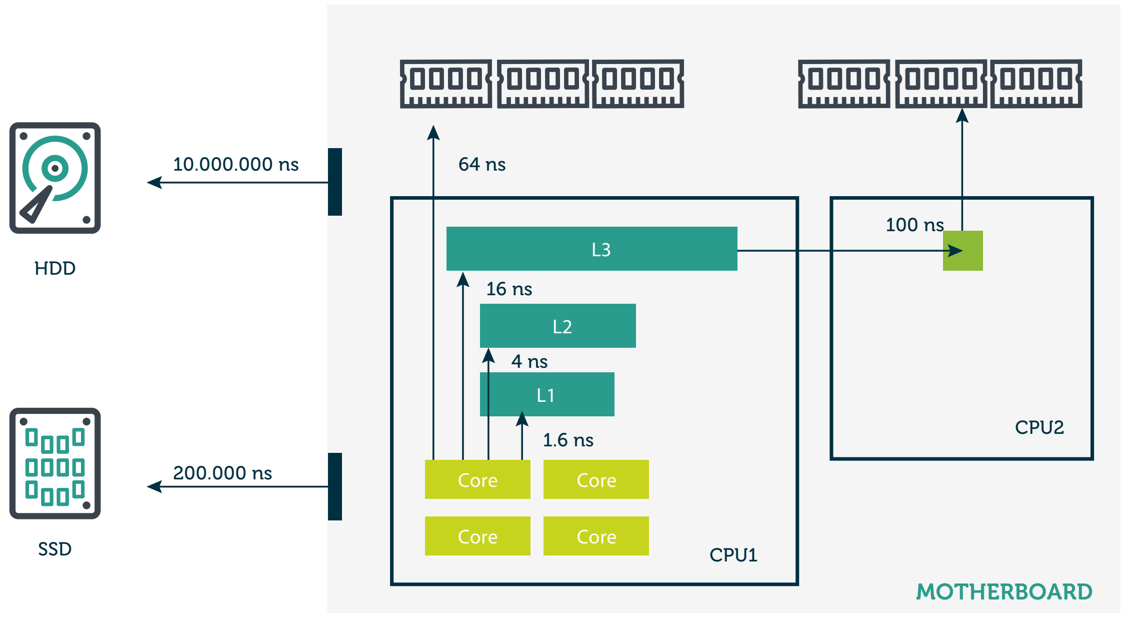
02

Being really fast

Large-Scale In-Memory Architecture

One popular misconception about in-memory analytic databases is the idea that they have to hold all permanent and temporary data in RAM all the time, effectively putting very tight constraints on the size of the data that can be stored and processed in such systems. For Exasol, such constraints are no issue at all.

After more than two decades of research and investment in in-memory technology, Exasol has reached a high level of maturity and versatility and Exasol has gained a deeper understanding of what in-memory is really about: In-memory is not a single technical feature of a system but instead an overall design approach for storage and processing algorithms, an approach that is at work throughout the whole system. The basic assumption that in-memory rests upon is the following:



02

Being really fast

Large-Scale In-Memory Architecture

By the time data needs being accessed, it resides in RAM or even CPU cache. Exasol's approach to in-memory therefore involves:

Algorithms that work under the assumption that data access is very cheap (in terms of time)

Machinery that – in the background – works very hard to make the in-memory assumption a reality most of the time

To facilitate such a design, the following basic approaches are used across Exasol's architecture:

1. Data compression:

By utilizing different levels of compression, the trade-off between low space requirements and high performance is adjusted for different operations.

In general, compression is a key enabling factor for in-memory databases as it reduces the need for hard disk drive accesses and also reduces the pressure on CPU caches.

2. Transparent replication:

It is cheap to store small data several times, which can reduce query execution times dramatically (see section 2.5). Users of Exasol are never affected by replication as the system handles all the details and even stops replicating data when it becomes too large.

3. Pre-fetching:

Exasol is very good at predicting the future access of data. Pre-fetching is performed on all levels of granularity:

a. RAM - L2 cache
b. Block device - RAM
c. Data from remote machines - temporary local buffers (see section 2.4)

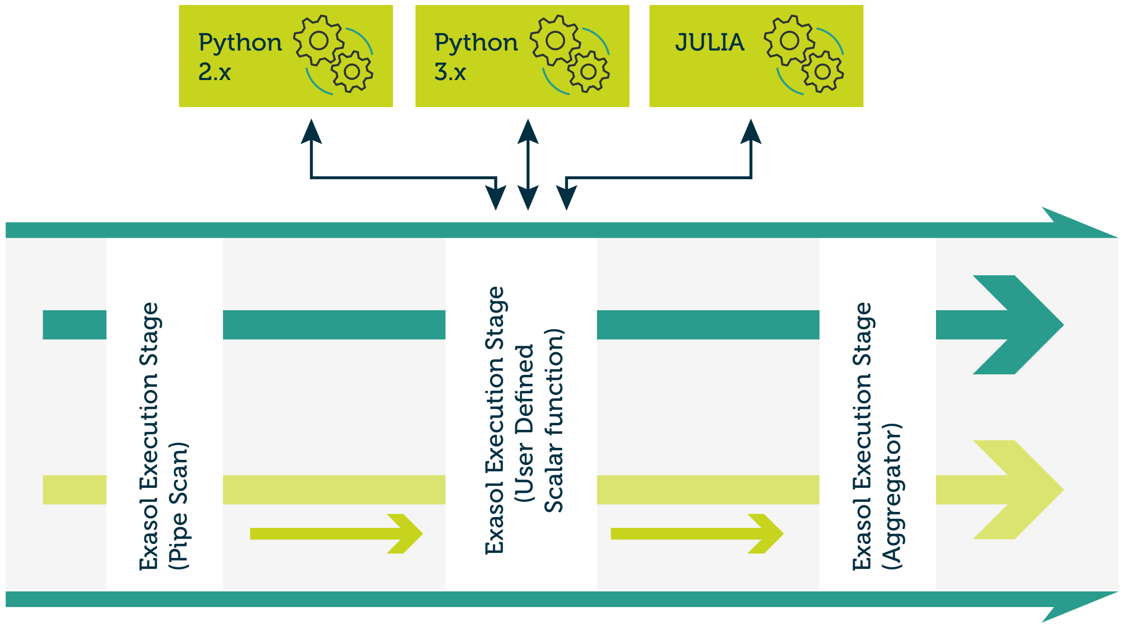
02 Being really fast

Large-Scale In-Memory Architecture

Pipelining

Given today's multi-core architectures, it is mandatory to avoid cases of synchronous waiting whenever possible in order to best utilize the available processing power. Exasol's execution pipeline manages a number of worker threads and buffers for intermediate results, thereby eliminating the need for direct thread communication and increasing the amount of parallel processing.

Additionally, the execution pipeline comes with built-in support for distributed computing. Intermediate results can be sent off to any other machine in the cluster for further processing. Such pipelines are very handy for global computations like global JOINs or the Skyline operator (see section x.x).


Python 2.x
Python 3.x
JULIA
Exasol Execution Stage (Pipe Scan)
Exasol Execution Stage (User Defined Scalar function)
Exasol Execution Stage (Aggregator)

02

Being really fast

Large-Scale In-Memory Architecture

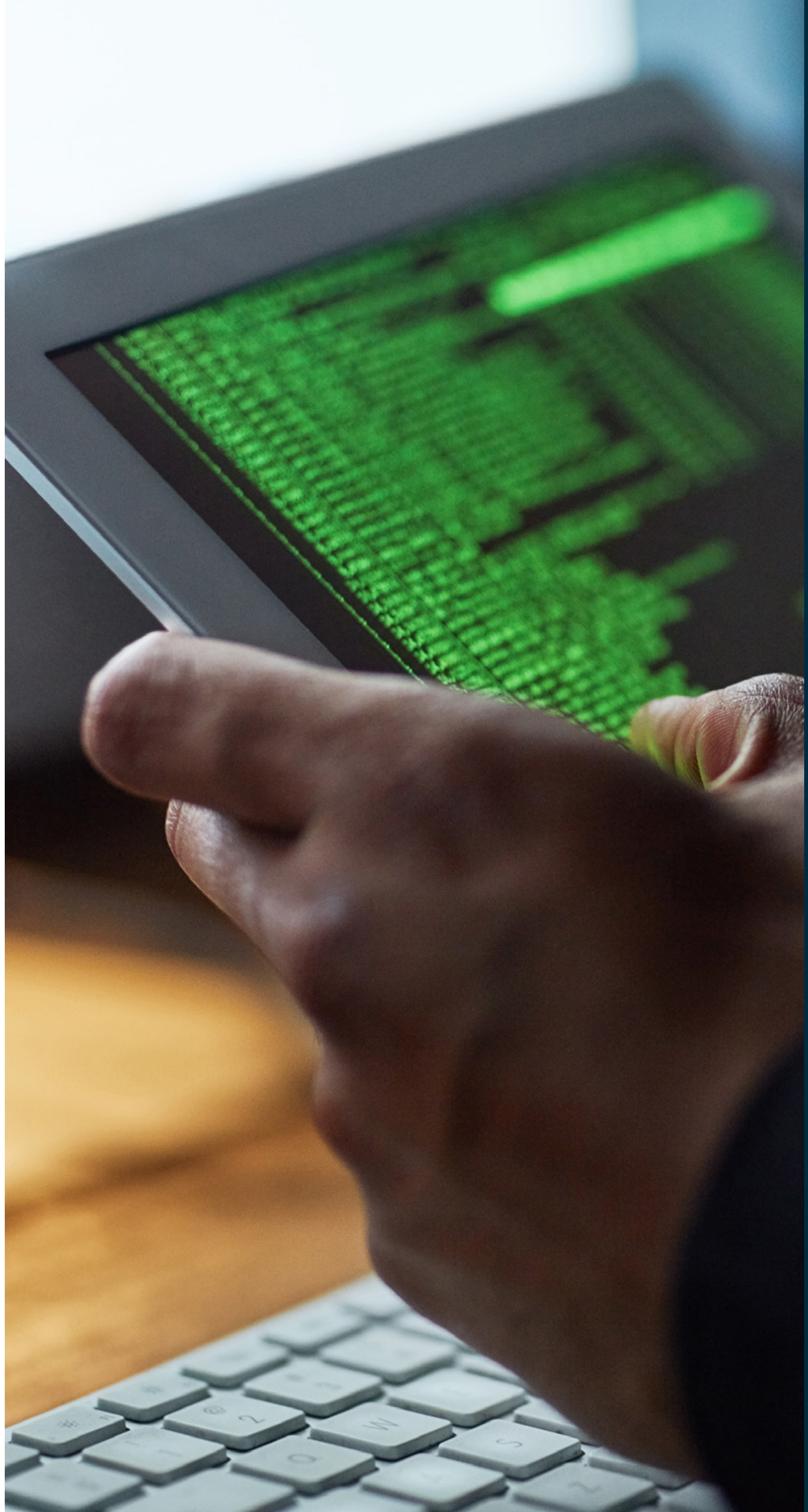
Data Locality

Careless design and implementation of algorithms may lead to situations in which processing units have to wait for certain data to become available. Therefore a strategy to avoid cache misses is an important topic for in-memory databases.

Exasol's operators are carefully designed and coded to anticipate when certain pieces of data are needed and go to great lengths to ensure that the right data is available in the proper layer of the memory hierarchy at the right moment. The problem of data locality emerges at several levels in the system. For instance, there are special CPU instructions that direct the processor cache to pre-fetch certain memory locations so that data is already in the processor cache when needed.

Similarly, Exasol contains elements that ensure that certain pieces of data are available on a machine when accessed, effectively providing a cluster-wide pre-fetch for remote data. Similar behaviors are supported by the internal memory management components.

By predicting the immediate future state of a complex computation at different levels of granularity, Exasol achieves a high level of data locality, which is one of the key factors for its performance



02

Being really fast

Filters, Joins and Sorting

Filters and joins are two of the core operations of relational databases. Exasol is equipped with a number of different algorithms for these operations. The actual choice of the algorithm is made by Exasol's cost-based query optimizer, which employs a number of statistics and meta-level knowledge in order to predict the cost of an operation in a given scenario.

Filters, even when they involve full scans of tables, are usually rather cheap operations, as they can be performed fully in parallel on all the CPU cores in the cluster. Internally, Exasol may produce (and maintain and delete) indexes as needed. If a suitable index already exists for a given operation like for a filter, this index will be used, yielding even faster filter operations. Another alternative is to employ global knowledge about the data contained inside a column (for example min/max values) in order to avoid filtering when it is known that it will not yield a result anyway.

Regarding the JOIN operation, Exasol contains several implementations that are optimized for different scenarios concerning the size and distribution of the involved tables. All tables in Exasol are horizontally distributed among all machines in order to achieve maximum parallelization and to balance the load of operations.

A general strategy in distributed databases is to avoid global JOIN operations, hence JOINs where the matching rows may reside on any two machines in the cluster. For instance, in typical business intelligence (BI) scenarios, small tables are often joined with a large table. For this important use case, Exasol has the ability to replicate tables (usually the smaller one) on every machine, thereby guaranteeing that all the results of the JOIN operator are on the same machine, effectively turning a costly global operation into a cheap local one.

Another approach would be to distribute the table according to the predicate to be joined. Actually Exasol allows users to influence how data is distributed in the cluster via DDL (data definition language) statements (the DISTRIBUTE BY clause for columns), but such actions should never be mandatory in Exasol.

Also, while avoiding global JOINs is a reasonable strategy, global JOINs in Exasol are nothing to be afraid of at all. Exasol features fully distributed indexes that can be used for joining columns and that do not need to be replicated on every machine but instead utilize the complete cluster for index lookups. In addition, Exasol supports the caching of JOIN results to further increase performance and reduce the inter-machine communication.

Sorting is one of the key operations in any database and a major building block for a number of features beyond the classical ORDER BY queries. For instance, for analytical functions, Exasol supports window functions that are run on the result sets of SQL operations. These functions may contain new data partitions and orderings which are implemented using sorting. For distributed global sorting, Exasol uses a combination of radix sort and merge sort.

02

Being really fast

Query Optimizer and Query Cache

Like all modern relational databases, Exasol contains sophisticated rule- and cost-based query optimizers. But in addition to the standard metrics like the number of data read operations, Exasol's cost model also quantifies characteristics that are specific to its cluster architecture.

One such characteristic is the number of machines in the cluster – for some operations there are even implementations that are specifically designed for systems running on a single machine. Other characteristics are the replication status and the distribution key of tables. If a table is replicated, it may be used as root table in JOIN operations, yielding globally independent local JOINs. Equally, if two large tables are joined on some attributes and the tables are distributed with respect to these attributes, again, global JOIN operations can be replaced by local JOIN operations.

As a rule of thumb, Exasol's optimizer tries to create queries that can run independently on each machine in the cluster, minimizing the need for coordination. On the other hand, if such an option is not available for a given query, the use of smart global indexes significantly reduces the cost of global operations. Exasol's optimizer is rather aggressive concerning the availability of indexes or replications. If it is convinced that such structures would improve query performance, it automatically creates indexes and replications. This is even the case for temporary results (like sub-selects with ORDER BY and LIMIT clauses), thus pursuing optimal and parallel execution of queries.

Exasol's query cache stores the results of previous queries along with metadata that allows it to decide when the recorded results are still valid. In general, not all data can be cached (for example "SELECT random() FROM DUAL"), but when it is, the query cache can speed up repetitive queries tremendously, especially in environments in which data changes slowly relative to query frequency.

Typical use cases include monitoring applications like BI dashboards, which constantly refresh their display even if the underlying data is changing slowly, or even classical search-result lists where a certain number of results is displayed together on one page. If the user selects the next page, basically the same query is executed again, taking great advantage of Exasol's query cache.

03

Providing a Great User Experience

A great database user experience does not mean an arcane set of features but rather a database system that silently provides all the necessary facilities but otherwise steps out of the way of users. To this end, Exasol contains a number of features working in concert to provide a strong user experience.

The general goal for the Exasol user experience is to provide maximum flexibility and expressiveness. For instance, in Exasol it is not necessary to predefine aggregations or sorted projections on data or to manually distribute data in order to achieve reasonable performance. Such requirements, while probably giving some performance improvement, would be a tremendous handicap for analysts using the product.

03

Providing a Great User Experience

Self-Optimization

Exasol performs a lot of global optimization work without any human interaction – for instance:

Indexes that are used in many operations, like JOIN, FILTER, and so on, are created, maintained, and disposed of without bothering the user with such details.

The query optimizer – which is part of the SQL compiler – incorporates statistics about the system that are collected all the time. There is no need for UPDATE STATISTICS or similar manual interventions. Additionally, the optimizer is rather robust in the sense that different similar versions of the same query are usually compiled in the same way; there is no need to tinker with queries in order to achieve excellent performance.

Small tables are replicated on all machines if there are enough idle resources available.

In order to decide which indexes to create, which tables to replicate, and many more tasks, the system continuously monitors its own behavior in order to provide the most accurate information for the query optimizer.

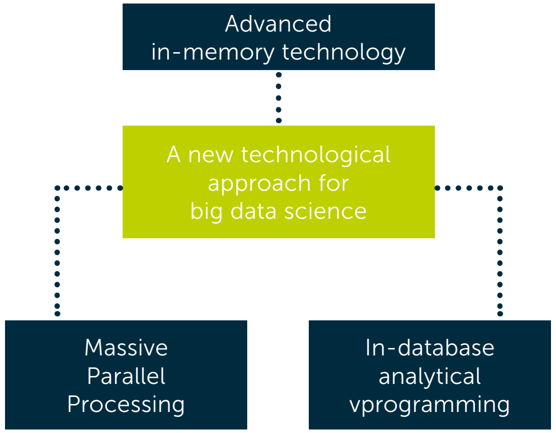
Exasol is a tuning-free database in the sense that it should never be necessary to manually interact with database internals in order to get an answer in a reasonable amount of time. Nonetheless, it is of benefit to understand where the system is spending its time. To this end, Exasol offers detailed query profiles that indicate the amount of time spent, memory consumed, and number of rows produced for each step in query execution.

03

Providing a Great User Experience

Advanced Analytics and Data Science

Exasol contains a number of features that considerably extend the basic high-performance SQL capabilities towards advanced analytical and Data Science applications.


Advanced in-memory technology
A new technological approach for big data science
Massive Parallel Processing
In-database analytical vprogramming

Exasol's SQL engine may be extended with user-defined functions (UDFs). These are tightly integrated with SQL processing and can be run fully in parallel and distributed in the cluster.

Alongside out-of-the-box support for R, Python, Java and Lua and the capability to upload any additional packages into the system, Exasol offers an open framework to integrate nearly any analytical programming language of your choice.

Exasol already provides Open Source language clients such as C++, and the Exasol community is able to develop and share additional ones quite easily by implementing a very thin and well-documented API. Afterwards, the language "client" just needs to be uploaded into Exasol's system to be made available for sophisticated in-database analytics, combined with standard SQL queries. The flexibility of Exasol's UDF framework allows you to create scalar, aggregate, analytic functions and even MapReduce procedures.

With UDFs integrating existing solutions right into Exasol's SQL processing is a straightforward process, which in turn means that the complete power of the parallel distributed in-memory architecture is utilized. Such an approach is especially suited for Big Data scenarios, in which it is utterly impossible to export all the data before the actual processing. With Exasol's UDFs, the analysis is pushed down to the data.

03

Providing a Great User Experience

Advanced Analytics and Data Science

The Skyline Operator

For many decision-making applications, selecting even second-best options might involve a large cost and the burden of missed opportunities. To address such scenarios, the Exasol Skyline operator allows you to compute exactly all "the best" answers to decision problems that are defined by partial orderings over items. Technically speaking, Skyline performs multi-criteria optimization, also known as Pareto optimization, right inside the database, taking full advantage of Exasol's distributed and parallel processing.

Skyline is an extension to standard SQL. A new clause is added to the SELECT statement which enables users to specify partial orderings for result sets involving multiple criteria very easily. Skyline then computes the global maxima of these orderings. It eliminates all tuples for which better alternatives exist in the result set, effectively constituting a global filter in which the presence of tuples in the result set is influenced by all the other tuples under consideration – an operation which goes way beyond the capabilities of all other database systems.

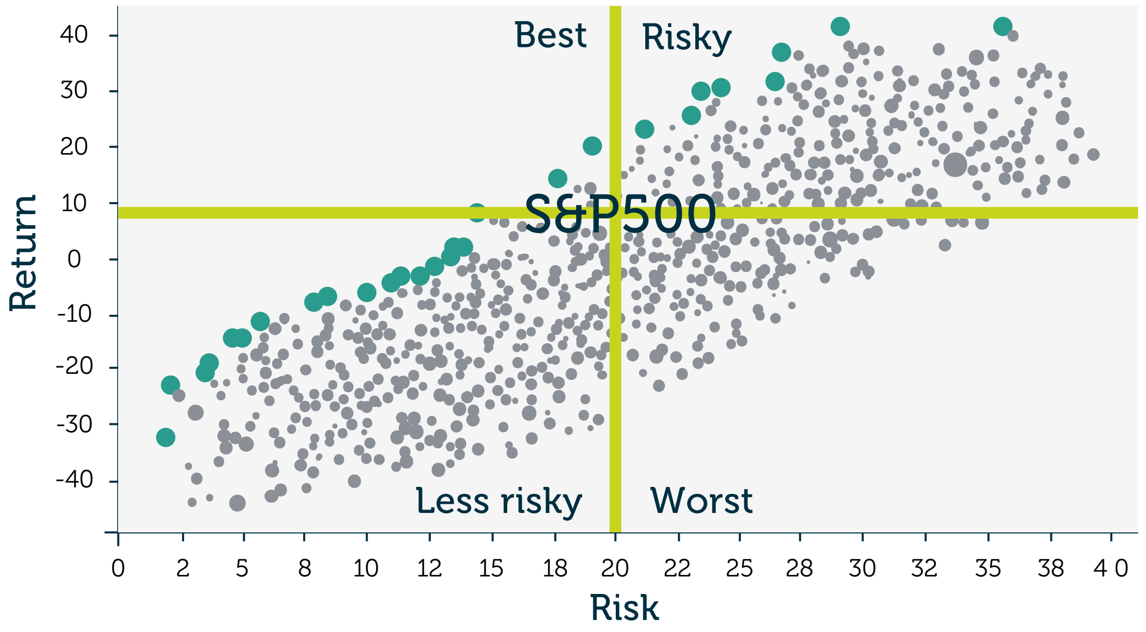


Figure 1: Example of the Fund Universe Comparison graph (http://www.fundreveal.com) and the best-matches only set (bold points).

If you are interested in more details about Data Science on Big Data, we recommend our whitepaper "Big Data Science – the future of analytics with Exasol."

04

Supporting Business Integration and Day-to-Day Operation

Being a high performance database is surely an important capability. But if large corporations want to place a new database system into the core of their infrastructure and connect them to dozens of BI tools, or if dynamic data-driven companies want to build their entire business model on data analytics, then other features become also very important: operational ease-of-use, broad interfaces and tool support, robustness to different workload scenarios and hardware failures and the scalability to grow with new use cases and data volumes.

In this section we briefly discuss a selection of Exasol's features in this area which make working with the database an enjoyable experience.

04

Supporting Business Integration and Day-to-Day Operation

Interfaces and Tool Integration

Exasol's maturity is borne out of its ability to interact easily with standard technologies and BI tools. All the important connection standards such as JDBC (Java Database Connectivity), ODBC (Open Database Connectivity), ADO.NET, MDX (MultiDimensional eXpressions) are supported.

Exasol is officially certified and successfully in production use for nearly every Business Intelligence tool in the market, such as MicroStrategy, Cognos, Tableau and SAP BusinessObjects.

Additionally, Exasol offers a JSON over WebSocket API which makes it feasible to integrate Exasol into nearly any programming language of your choice. This protocol is more native, faster and avoids any overhead of driver managers. Further, it supports client/server compression. A native Python driver is just one example implementation and available in Exasol's open source repository.

For administrators, there is a web-based administration console – EXAoperation – that supports all types of administration tasks, such as installation, backup management, monitoring, and cluster enlargement. It contains a user interface for installing and maintaining several Exasol databases within a few minutes. Administration tasks can also be automatized via an XML-RPC interface, which allows for integration into system administration and management tools.

EXAplus is Exasol's own graphical SQL client that is specifically tailored for interacting with Exasol databases. Alongside convenient SQL interaction, it also supports Exasol-specific features, such as very fast import and export of data, schema browsing, and auto-completion.

04

Supporting Business Integration and Day-to-Day Operation

Data Ingestion and Data Integration

In today's data-driven world, data ingestion and flexible data integration are crucial tasks for data management systems. Alongside maximizing the pure data throughput, data latency and the ability to access and analyze new data as quickly as possible and the same time as flexible as possible are crucial.

Exasol has created a wide variety of optimizations to make data loading and manipulation a straightforward and enjoyable experience.

Parallel bulk pull loading via SQL commands IMPORT/EXPORT:

Instead of providing a client bulk loading tool, Exasol has integrated a powerful loading capability directly into the parallel database engine. Users can simply execute an SQL statement which defines the data source and format, and Exasol will start to pull data into the database and convert the data types in parallel.

This architecture leads to higher performance and the ability to integrate the data transfer directly into a fast ELT process without any need to persistently store intermediate results. Further, it avoids that users have to install any client software to ingest data.

Exasol is able to load data at a rate of more than 10 TB per hour in our labs; in real-world customer installations, most often the data source proves to be the actual performance bottleneck.

Users can load from local or remote files, from standard input stream and from any JDBC-connectable data source. Exasol also supports loading CSV data right from Hadoop's Distributed File System (HDFS) via the WebHDFS REST API with support for parallel loading of compressed files.

Needless to say, Exasol is compatible with the standard ETL tools such as Talend or Informatica.

Advanced Integration Framework based on ETL-UDFs

For more sophisticated tasks (like Hadoop's native storage formats) or custom APIs (web-services etc.), Exasol's advanced open and extensible framework based on ETL-UDFs can be utilized to solve almost any integration requirement.

For instance, data in HDFS/Hadoop is stored in a variety of file formats such as csv, json, avro, thrift or in formats specialized for SQL-on-Hadoop solutions such as RCFile, ORC and Parquet. These data formats evolve as quickly as the whole Hadoop and Spark ecosystem itself.

Exasol delivers a flexible and extensible Hadoop adapter based on its ETL-UDF framework. It is published as open source and can be modified or extended by any customer or partner, completely independent of Exasol's software release cycle.

04

Supporting Business Integration and Day-to-Day Operation

Data Ingestion and Data Integration

Technically, the Hadoop ETL-UDF integration adapter is tightly coupled to HCatalog as it uses the serializers, deserializers and metadata maintained by HCatalog/HIVE in order to interpret and import the data stored in HDFS (Columns and types, partitions, buckets, HDFS file locations, Input/Output format and SerDe and SerDe properties.).

Therefore, Exasol instantly supports all HCatalog/Hive supported data formats. Whenever HCatalog supports a new data format, Exasol can interpret and therefore import that data format as well.

Automatic data distribution and internal index structures

Since Exasol is parallel, cluster-based MPP software, parallelism is a very important reason for its maximal performance and scalability. To ensure the cluster hardware is leveraged optimally, Exasol distributes data evenly and across the servers. Furthermore, internal indices are automatically created and maintained to optimize query performance. All this is done under the hood and does not need any user interaction. Only some power-users decide to define the data distribution by specifying certain table columns. This is actually the only tuning parameter the database provides. Exasol's mantra has always been that users don't need to care about system settings so that they can concentrate on using the software instead of constantly analyzing and tuning it.

Hybrid row/column storage

Exasol combines its columnar data storage which is optimal for analytical queries and data compression with row-based storage which reduces the latency of incremental, small data inserts. So-called tail blocks store the recent data and make single row inserts very fast. Intelligent background algorithms automatically merge the content of these tail blocks into the main column-based data structures. So, from outside, the system looks like a columnar database, but inside and transparently, Exasol facilitates minimal data latency by storing its data in a hybrid way.

Updates and schema modifications:

Many database systems struggle with update processes and have a large internal overhead when changing data content and structures, e.g. if you add columns or delete data. Exasol has implemented a broad range of features (INSERT, UPDATE, MERGE, and IMPORT) and optimizations (in-place updates, incremental index changes, and column [re-] compression, data distribution) to minimize the system impact of data manipulation operations. As a user, you simply do not have to care about the internal workings.

04 Supporting Business Integration and Day-to-Day Operation

The Virtual Schema Framework for Data Virtualization & Hybrid Clouds

In another section of this document we advocated ELT over ETL by loading the not-yet processed data from the external source into Exasol, instead of preprocessing the data externally first.
There might be reasons and use cases where this approach is still suboptimal:

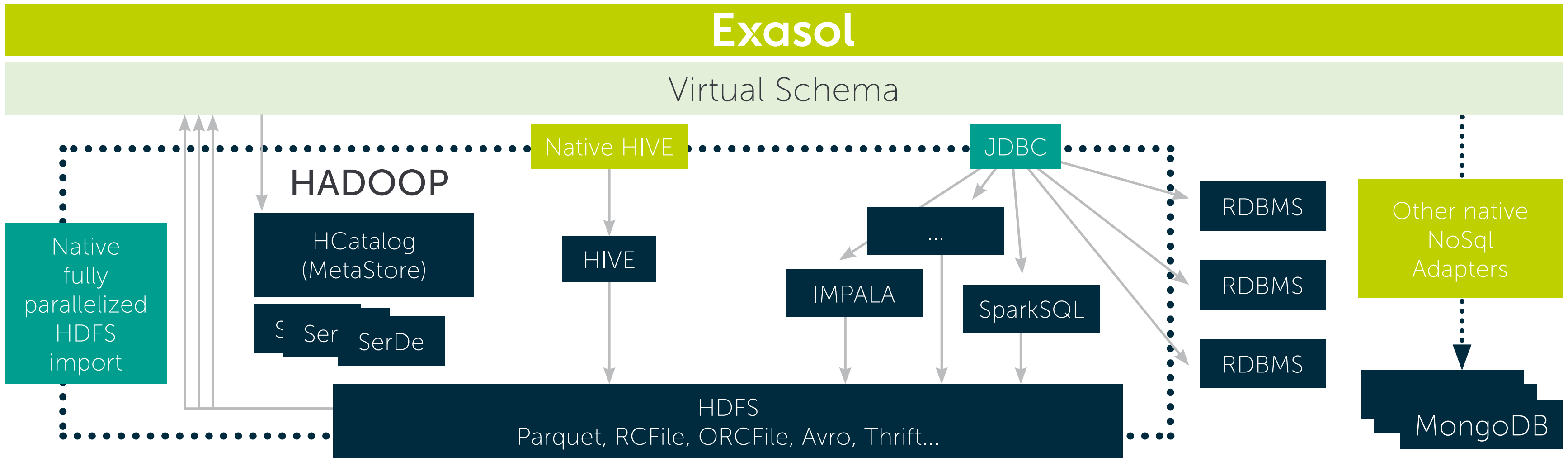
Data volumes stored in the external system are too large to import them into Exasol

Different uses cases/users need to access different parts of these large data volumes making it impossible to identify and import just the relevant part of the data

Data in the external data source changes continuously, but users have high requirements in terms of up-to-dateness and want to access the latest "live" data (e.g. data stored in operational cloud systems like Salesforce or the ERP system),

Trying to solve these problems using ETL/ELT would cause high maintenance efforts for ETL/ELT jobs/processes as these processes would have to be continuously updated.

Furthermore, typical ETL integration and modification processes would take too long as typically maintained by IT-departments and managed using formal processes.
To solve this goal conflict and to reduce data redundancy Exasol provides the Virtual Schema data integration framework.



04 Supporting Business Integration and Day-to-Day Operation

The Virtual Schema Framework for Data Virtualization & Hybrid Clouds

The concept of virtual schemas provides a powerful abstraction to access arbitrary data sources. Virtual schemas are a kind of read-only link to an external source and contain virtual tables which look like regular tables except that the actual data is not stored locally. Whether virtual or physical, this is fully transparent from the application perspective.

When connecting a data source using virtual schemas, only the metadata of virtually connected data sources is transferred.
Access to these virtual schemas is dynamically forwarded to the connected data sources. The data is transferred "on demand." Predicates and subqueries are intelligently pushed down to the remote data source based on its capabilities.

Joins between virtually-connected and physical tables or between multiple virtually connected tables are possible.

A unique aspect of Exasol's virtual schemas is its extensibility. It is based on a well-documented, easy-to-use framework that is available as open source GitHub project. Customers and partners can easily add new or customize existing adapters and share it within the Exasol community.

Capabilities of virtual schemas :

Integrate multiple data sources into one transparent high performance layer

Agile live access to most recent data of remote data sources while reducing redundancy

Ability to implement two-layer concept of hot and warm/cold data

Extensible open source framework for integrating additional systems and services with the programming language of your choice

04 Supporting Business Integration and Day-to-Day Operation

The Virtual Schema Framework for Data Virtualization & Hybrid Clouds

Hybrid Cloud Deployments

In general, the hybrid cloud concept combines on-premise (or private cloud) systems with public cloud deployments within one seamless IT landscape.
The most relevant reasons for hybrid cloud in the context of data management systems and database in particular are:

Privacy and data protection: Manage sensitive or critical workloads on-premise while storing and using public cloud offerings to manage less-critical information.

Data locality: If you generally operate systems in the cloud as well as on-premise it might be not feasible to transfer all the data located in the cloud to on-premise systems and vice-versa to perform analytics that comprises all the data.

Performance and availability: Manage data that has to be delivered within sub-seconds using your on-premise high performance hardware while leveraging cost-efficient cloud servers for data with lower requirements in terms of response-time and availability.

Cloud Migration: When migrating your IT landscape from on-premise to the cloud or vice-versa, a hybrid cloud approach permits a step-wise migration process: use-case by use case or database table by database table without any service interruption. Users always see a consistent and complete view. For them it's fully transparent what is in the cloud and what is not.

With its new concept of virtual schemas, Exasol provides a powerful abstraction functionality to integrate and transparently access arbitrary data sources in such hybrid environments.

Just set up one Exasol database cluster in the cloud, another one on-premise and connect both systems using our virtual schemas technology. Or connect any other data management source in your data center or cloud environments to create one transparent, virtual access layer.

A user who connects to the local Exasol database will then be able to access the data of the connected database systems, locally and remotely. Therefore, it's completely transparent where the data is stored and managed. But the actual processing is automatically split across the different platforms.

04

Supporting Business Integration and Day-to-Day Operation

The Virtual Schema Framework for Data Virtualization & Hybrid Clouds

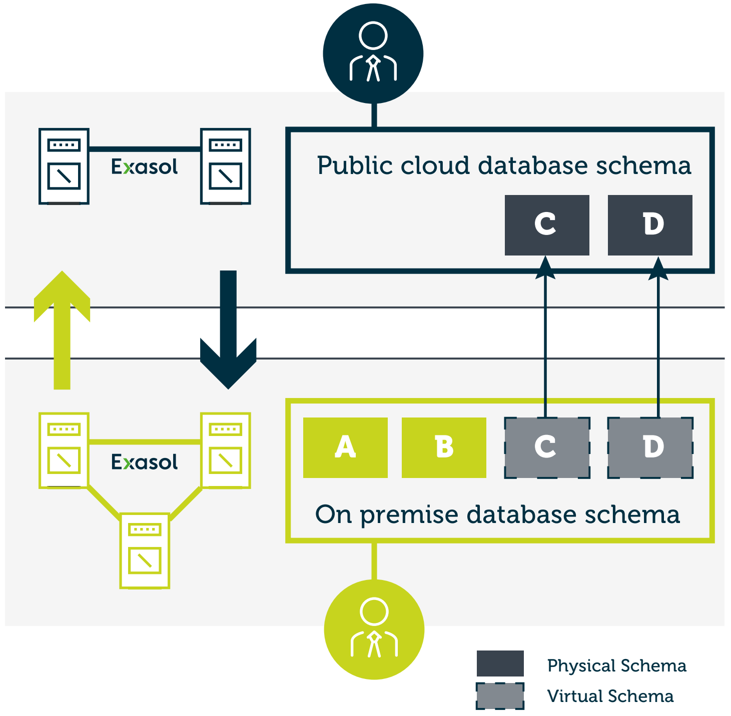
Think of virtual schemas as a read-only link to an external source that contains virtual tables that look like regular tables except that the data is not stored locally.

When a user executes a query that references a virtually connected schema and a physical schema, Exasol automatically sends parts of the query to the virtually connected data source. Intelligent pushdown algorithms ensure that as many functions and predicates as possible are executed on the remote system to minimize the amount of data that needs to be transferred between the systems located on-premise and in the cloud.

It looks like a normal schema with data stored in the database the user is interacting with. But the actual processing is automatically split across the different platforms.

In general, a virtual schema based hybrid cloud deployment can be implemented using any data management system supported by virtual schemas (e.g. Oracle, SQL Server, Hadoop or any other ODBC/JDBC-compliant database) in combination with an Exasol database instance.

Exasol has decided to publish the virtual schemas extension as open source project on GitHub (Link) so that additional adapters to further data sources can easily be shared and extended by the growing Exasol community.



04

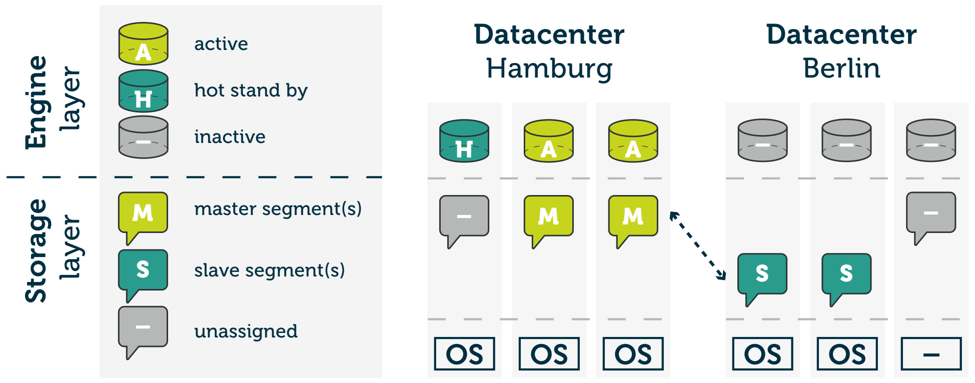
Supporting Business Integration and Day-to-Day Operation

Fail Safety, Dual Data Center Operation and Backup/Restore

Exasol supports fail safety via redundancy. Clusters can be equipped with spare machines. If a live machine in the cluster goes down due to hardware failure, one of the spare machines is up and ready to replace the failed machine in seconds. In practice it is quite likely that users would not even notice that a machine had gone down.

Even more serious situations like complete power failures are handled gracefully and – most importantly – without losing any data.

For disaster recovery, Exasol supports synchronous dual data center setups where one cluster can be “stretched” across two data centers. If the active data center goes down, the second data center can quickly take over operations. Both parts of the cluster are always in-sync, thus making data loss impossible and the backup system up and running instantly.



Exasol supports incremental online backups during normal operation and also explicitly scheduled backups. All data (and backup data) is stored redundantly so that the failure of machines in the cluster does not affect data integrity. Furthermore, for maximum recovery performance in non-critical scenarios, backups can be created inside the cluster or externally or even be stored at remote locations for maximum safety.

Backups can be stored on standard file systems, FTP servers or even on HDFS (via WebHDFS).

In addition to full restores from backups, Exasol also features selective restores of single elements from previous backups.

04

Supporting Business Integration and Day-to-Day Operation

SQL Preprocessor

SQL Preprocessor is a feature specifically designed for consultants and solution engineers. It allows them to apply any kind of textual transformation to queries before they are fed into the SQL compiler. A typical use case is the adaption of SQL statements that were written with other database systems in mind. Such statements sometimes occur in queries that are automatically generated by other tools. In these cases, it is rather straightforward to map unknown function names or data types to their equivalents in Exasol. Other use cases are user-defined short cuts or the ad hoc implementation of missing or experimental functionality.

SQL Preprocessor is integrated into all Exasol SQL processes. If users define a preprocessor script, this script is called for all statements and its output is given to the SQL compiler for further processing. In order to ease the tasks of defining transformations to SQL, the preprocessor provides a module for tokenizing strings into SQL-related tokens.

SQL Preprocessor itself may issue any number of SQL queries during the transformation, which effectively allows you to use SQL and the full power of Exasol in order to generate the actual SQL query to be executed.

05 Summary

Now, as you have taken a peek under the hood, here you can find a summary of the top reasons why you should consider using Exasol:

In-memory technology

Innovative in-memory algorithms enable large amounts of data to be processed in main memory for dramatically faster access times.

Column-based storage and compression

Columnar storage and compression reduce the number of I/O operations and amount of data needed for processing in main memory and accelerate analytical performance.

Massively Parallel Processing (MPP)

Exasol was developed as a parallel system based on a shared-nothing architecture. Queries are distributed across all nodes in a cluster using optimized, parallel algorithms that process data locally in each node's main memory.

High user concurrency

Thousands of users can simultaneously access and analyze large amounts of data without compromising query performance.

Scalability

Linear scalability lets you to extend your system and increase performance by adding additional nodes.

Tuning-free database

Intelligent algorithms monitor usage and perform self-tuning tasks, which optimize performance and minimize any data administration overhead.

05 Summary

Faster access to more data sources

Through a data virtualization framework called “virtual schemas” as well as a high performance data integration framework, you can connect to and analyze data from more sources than ever before.

Comprehensive support for Hadoop

Hadoop integration has never been so easy. Exasol supports all native HDFS formats enabling you to perform high-speed analytics against structured and unstructured data faster and easier.

Advanced in-database analytics

Alongside out-of-the-box support for R, Python, Java and Lua, Exasol allows you to integrate the analytics programming language of your choice and use it for in-database analytics.

Unrivalled connectivity

Easily connect to your existing SQL-based BI, reporting and data integration tools via ODBC, JDBC, .NET as well as a JSON-based web socket API.

Ultimate flexibility

Exasol fits in with your business model and can be deployed in a number of different ways. Choose the option the best meets your needs: software-only, appliance or in the cloud (EXACloud, Bigstep, Microsoft Azure or Amazon Web Services).

If you want to learn more about Exasol, get in contact with us today (see Exasol.com for how to do that).

If you want to get your hands on Exasol sooner, check out one of the free trial options:
1. Download our free single-node edition for commercial and private/academic use.
2. Sign up for a personal demo system hosted in the cloud
3. Try Exasol on Amazon Web Services Marketplace or Microsoft Azure Marketplace

Just visit www.exasol.com/download to learn more.

Exasol AG
Neumeyerstr. 22 – 26
90411 Nuremberg
Germany
www.exasol.com

Tel: +49 911 23991-0
Email: info@exasol.com

Follow us for the latest content: