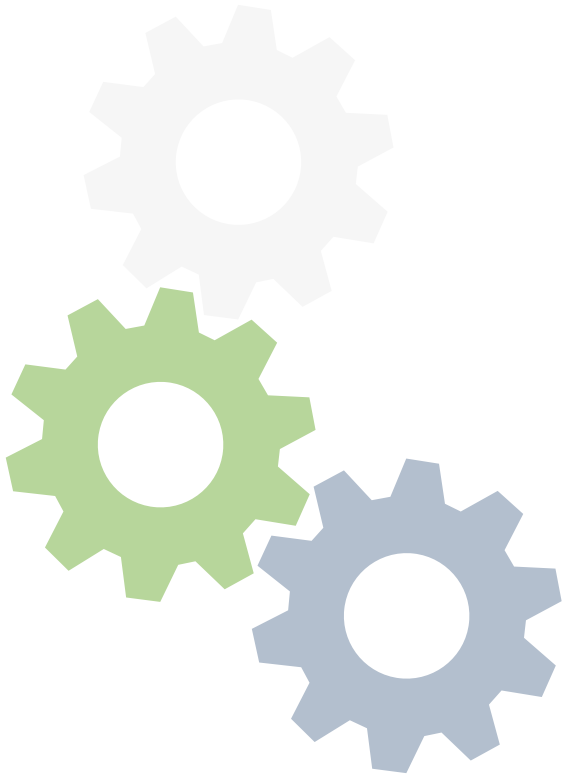
EXASOL

EXASolution

Technical Whitepaper

A Peek under the Hood

Table of Contents



1 Introduction

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

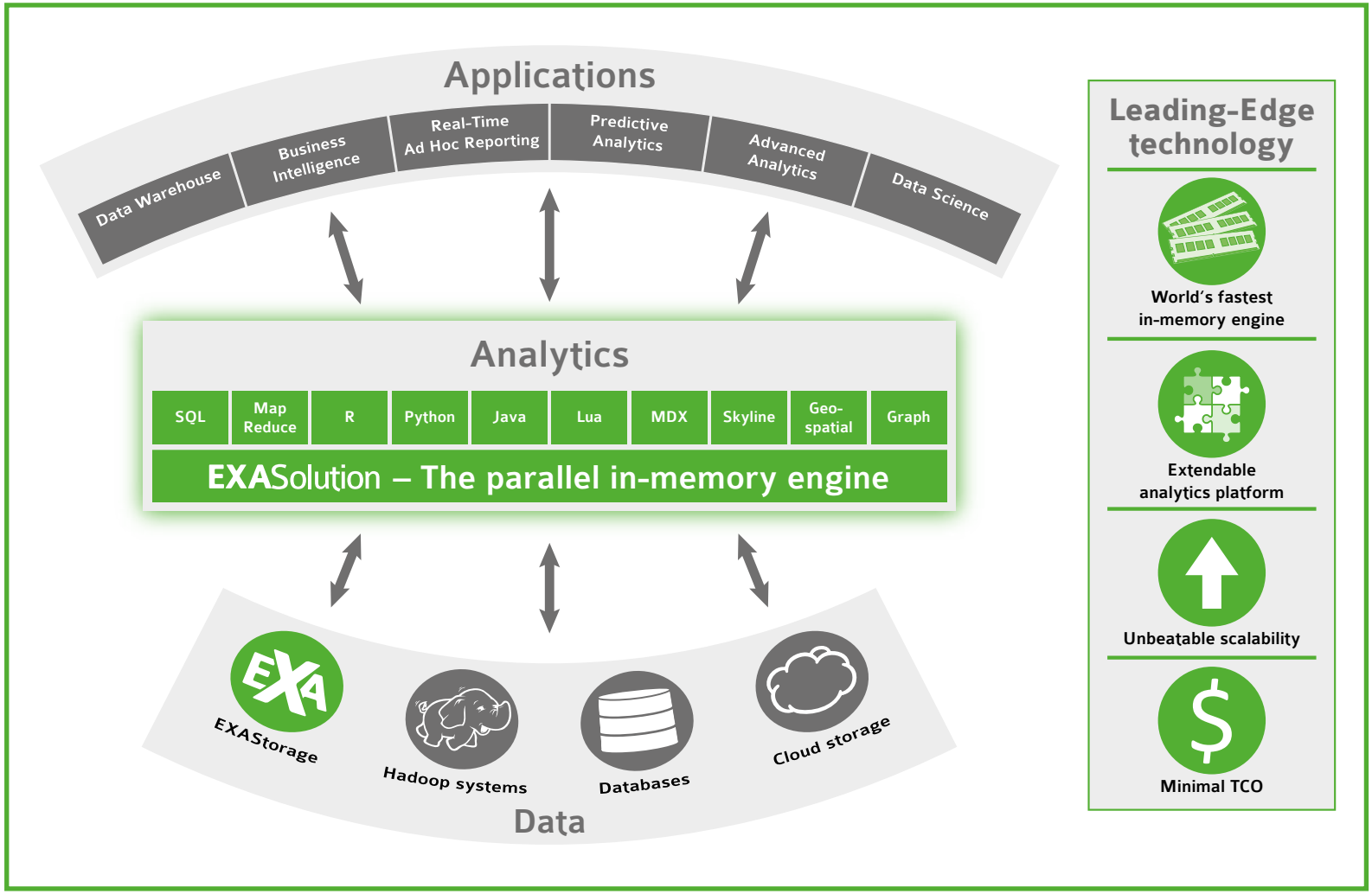
1. The major improvements in processing power were not coming from ever increasing clock speeds of central processing units (CPUs) anymore, but instead from parallel and distributed systems.
2. After a long period of slow improvement, random-access memory (RAM) started becoming much larger and cheaper with each succeeding generation.

EXASOL's founders, coming from backgrounds in high-performance computing and scientific computing, 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, EXASolution, is a high-performance in-memory database designed specifically for analytics. In 2014, EXASolution set new performance records in the TPC-H online transaction processing benchmark from the Transaction Processing Performance Council (TPC) for clustered decision-support databases, outperforming competitors by orders of magnitudes.

The goal of this paper is to offer a deeper understanding of a selection of design principles, core features, and technologies that underpin EXASolution's analytical performance and, at the same time, take a glimpse at what EXASolution has to offer beyond raw performance. It is organized along the following topics:

1. Being really fast
2. Providing a great user experience for analysts
3. Supporting business integration and day-to-day operation



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

2 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 some general principles emerged. Sections 2.1 to 2.4 discuss such general approaches to high speed while sections 2.5 to 2.9 present more specific solutions for typical challenges in relational database management system design.

2.1 Massively Parallel Processing – MPP

The design of EXASolution 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 on different levels of the architecture. These techniques have been designed and built from scratch exactly for the requirements of EXASolution.

At the cluster level, EXASolution 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. EXASolution 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 allows installations with hundreds of machines, all working in parallel to answer a query, without imposing 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, EXASolution is designed to avoid global synchronizations during query processing as much as possible.

Every machine, in turn, 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 maximum utilization of existing standard server hardware. Per CPU all the SIMD (single instruction, multiple data) features of modern standard processors are utilized completing the picture of EXASolution reaching for the highest performance on every level.

2.2 Large-Scale In-Memory Architecture

One popular misconception about in-memory analytic databases is the idea that they would have to hold all permanent and temporary data in the available RAM all the time, effectively putting very tight constraints on the size of the data that could be stored and processed in such systems. For EXASolution such constraints are no issue at all. After more than two decades of research and investment in in-memory technology, EXASolution 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 is 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:

By the time at which 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 three basic approaches are used throughout EXASolution'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 EXASolution 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:
   EXASolution is quite 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)

2.3 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. EXASolution's execution pipeline manages a number of worker threads and buffers for intermediate results, thereby largely 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 feature (see section 3.6).

2.4 Data Locality

Careless design and implementation of algorithms may easily lead to situations in which processing units have to wait for necessary data to become available. Therefore a strategy to avoid cache misses is an important topic for in-memory databases.
EXASolution'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, EXASolution 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 behaviours are supported by the internal memory management components.
By predicting the immediate future state of a complex computation on different levels of granularity, EXASolution achieves a high level of data locality, which is one of the key factors for its performance.

2.5 Filters and JOINS

Filters and JOINs are two of the core operations of relational databases. EXASolution is equipped with a number of different algorithms for these operations. The actual choice of the algorithm is made by EXASolution'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 CPUs in the cluster. Internally EXASolution 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, EXASolution contains several implementations that are optimized for different scenarios concerning the size and distribution of the involved tables. All tables in EXASolution 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 EXASolution 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 EXASolution 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 EXASolution.

Also, while avoiding global JOINs is a reasonable strategy, global JOINs in EXASolution are nothing to be afraid of at all. EXASolution 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, EXASolution supports the caching of JOIN results to further increase performance and reduce the inter-machine communication.

2.6 GROUP BY

GROUP BY is one of the major SQL constructs used for analytical queries. It is one of the operations that are rather hard to implement properly for distributed databases. In EXASolution there are three main strategies:

1. If the data is already distributed according to the GROUP BY keys, then GROUP BY is actually a local operation that can be performed in parallel on each machine.
2. If there is an index for the GROUP BY keys, the index can be used to compute the members of a group in a single step per group most of the time. This property allows for efficent parallel GROUP BY computations.
3. Otherwise, groups have to be identified on the fly and data has to be distributed accordingly. EXASolution's implementation for this scenario is based on iterative in-memory grouping and off-memory merge operations.

In general, each group is assigned one machine in the cluster that will perform its final aggregation step. Before rows aresent to the machine that performs the final aggregation – depending on the actual operation being performed – group members may be partially pre-aggregated.

This strategy ensures maximal scalability of aggregations.

2.7 Sorting

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, EXASolution 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, EXASolution uses a combination of radix sort and merge sort.

2.8 Query Optimizer

Like all modern relational databases, EXASolution contains sophisticated rule- and cost-based query optimizers. But in addition to the standard metrics like the number of data read operations, EXASolution'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, EXASolution'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. EXASolution'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.

2.9 Query Cache

EXASolution'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 (Web-) page. If the user selects the next page, basically the same query is executed again, taking great advantage of EXASolution's query cache.

3 Providing a Great User Experience for Analysts

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, EXASolution contains a number of features working in concert to provide a strong user experience.
The general goal for the EXASolution user experience is to provide maximum flexibility and expressiveness. For instance, in EXASolution it is not necessary to predefine 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 encumbrance for analysts using the product.

3.1 Self-Optimization

EXASolution performs a lot of global optimization without any human interference – 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.

3.2 Profiling

EXASolution 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 educational to understand where the system is spending its time. To this end, EXASolution offers detailed query profiles that indicate the amount of time spent, memory consumed, and number of rows produced for each step in query execution. In addition to profiles of past queries, it is possible to create on-the-fly profiles of currently executing queries. This latter feature is specifically designed to support certain monitoring scenarios. Furthermore, histories of profiles are stored in the database and can be used to detect performance trends ("the query runs fastest on Friday evenings").

3.3 Data Manipulation

Data manipulation in a distributed transactional ACID-compliant database like EXASolution is a non-trivial operation. In order to support fast data manipulation operations in such an environment, EXASolution incorporates the following strategies:

- Incremental operations:
  Transparent indexes and other derived data is not re-calculated from scratch after data manipulation; instead, increments are created that are merged when convenient.

For instance, DELETE operations work by merely marking deleted rows in a bit vector, making that operation very fast. If a certain percentage of a table's rows are marked as deleted, the table is automatically reorganized and the deleted rows are physically removed.

Small INSERT operations on columns with an index are very fast by using index-increment structures in order to maintain the correctness of the index. The additional data for such INSERTs is stored in special table tail structures that serve as an insert cache for the main body of the tables.

- Real updates:
  UPDATE operations physically change the value that is stored at a certain position. This approach is much faster than the alternative of simulating updates by using INSERT+DELETE sequences.

3.4 Multi-User Resource Management

Resource management takes care of assigning CPUs and other resources to active sessions and in general to maximizing throughput. EXASolution allows a huge number of sessions to execute queries at the same time. If the load gets too high, in order to achieve optimal throughput, sessions may be queued and continued in round-robin fashion at the start and end of transactions, essentially adapting the well-known time-slice model from operating systems to a distributed database.

User priorities enable assigning priorities to sessions and users, which easily allows for running several automated (background) queries in parallel while still achieving good performance for interactive sessions.

EXASolution supports query timeouts and employs a number of self-maintenance features like high-memory watchdogs that are able to terminate misbehaving queries when necessary. Without such features, simple bugs in user-defined functions (like non-terminating loops, an so on) would unnecessarily waste resources.

3.5 Advanced Analytics

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

EXASolution'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. UDFs can be defined either by choosing one of the supported UDF programming languages or by specifying a connection to an external service implementing the UDF⇔EXASolution interaction protocol (see eUDFs below).

At the time of this writing, EXASolution directly supports UDFs written in the languages Java, R, Python, and Lua. Typical use cases for UDFs are functions that behave like Mapper or Reducer functions in Hadoop. Using standard SQL, these can be combined into MapReduce computations right inside of EXASolution, even when nested deep down in a hierarchy of sub-queries. Other use cases are custom scalar, aggregate, and analytical functions.

With UDFs it is straightforward to integrate existing solutions right into EXASolution's SQL processing, thereby utilizing the complete power of the parallel distributed architecture. 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 EXASolution's UDFs, the analysis is moved to the data.

In addition to UDFs written in Java, R, Python, or Lua, EXASolution supports a generic interface for external user-defined functions (eUDFs). This is a public protocol,

implemented by EXASolution SQL processes in such a way that all external processes that support this protocol can be embedded right in the middle of the computation of SQL query results. On the technical level, eUDFs services are integrated via the very popular ZeroMQ and Google Protocol Buffers libraries, making writing eUDFs a quite pleasant and familiar endeavour. Section 4.5 describes a typical use case for eUDFs: native Hadoop integration.
Finally, EXASolution comes with a wide range of analytical functions already built in. These are functions that are computed over the result sets generated by standard SQL queries. Analytical functions partition, order, and slice the result sets in order to model rather complex analytical scenarios.

3.6 Skyline

For many decision-making applications, ranging from the selection of stocks for funds or other investments to choosing candidates for job interviews, selecting even second-best options might involve large cost and the burden of missed opportunities. Due to information overload, even for small numbers of options, finding the best ones is quite cumbersome and practically intractable already for medium numbers of options. To address such scenarios, EXASolution's "Skyline" feature 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 EXASolution's distributed and parallel processing. Depending on the specified ordering, Skyline is often able to reduce very large sets of options to small selections which often could be inspected by domain experts – even manually – afterwards, effectively overcoming the problem of information overload.
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.
Skyline is based on the results of a two-year research project with academia, funded by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology.

4 Supporting Business Integration and Day-to-Day Operation

EXASolution is a full-blown, standard-compliant, distributed, transactional SQL database. In addition to executing analytical queries very quickly, it provides all the important features needed for day-to-day operation in businesses around the globe. In this section we briefly present a selection of these features.

4.1 Fail Safety, Backup and Restore

EXASolution 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.

EXASolution supports state-of-the-art 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 the 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.

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

4.2 Interfaces

One of the indicators of EXASolution's maturity is its ability to interact nicely with standard technologies and BI tools. All the important connection standards like JDBC (Java Database Connectivity), ODBC (Open Database Connectivity), ADO.NET, MDX (MultiDimensional eXpressions) are supported. There is also support for integration with LDAP (Lightweight Directory Access Protocol) and Active Directory installations.

EXASolution is certified for use with MicroStrategy, Cognos, and Tableau. Furthermore, EXASolution is in productive use with SAP BusinessObjects and other Business Intelligence tools (via generic ODBC or JDBC support) at our customers.

For administrators, there is a Web-based administration console – EXAoperation – that supports all kinds of administrative tasks, such as installation, backup management, monitoring, and cluster enlargement. It contains a user interface for installing and maintaining several EXASolution 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 EXASolution databases. Besides convenient SQL interaction, it also supports EXASolution-specific features, such as very fast import and export of data, schema browsing, and auto-completion.

4.3 ETL/ELT

In classical data warehousing, ETL (extract-transform-load) describes the process of loading data into the data warehouse in the right format. With the data volumes that are common for EXASolution there usually is a better approach: ELT (extract-load-transform). This means:

1. Load the not-yet properly processed data from the external source into EXASolution.
2. Perform all necessary transformation steps inside EXASolution to make the best use of the processing power of the cluster.

Given the recommendation of ELT over ETL, the main focus of EXASolution regarding ETL are high-performance import and export of large data sets – a task that is non-trivial, given the distributed nature of EXASolution.

To address this challenge, EXASolution features EXAloader, a component of EXASolution specifically designed for parallel and distributed import and export of various kinds of data from different sources and targets. Import statements can even be used right inside SQL where sub-selects are supported. This allows – conceptually – treating different data sources, like local or remote files or query results from different databases, in the same way as local relational expressions. Another feature of EXAloader is the ability to load data fault-tolerantly. In this mode, data that does not match the import definition is written to an error table or error file, where it can be inspected after the loading process.

On a technical level, EXAloader uses a pull approach for loading rather than requiring client software tools for bulk loading. Instead of sequentially reading rows from one large file, it initiates loading requests from all the machines in the cluster in parallel, each machine loading independently from the other. Using this technology, EXASolution 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 employed data source is the actual performance bottleneck.

Currently, EXAloader supports loading from local or remote files, loading from any JDBC-connectable data source or Oracle connections. EXAloader also supports loading data right from the Hadoop's Distributed File System (HDFS) via the WebHDFS REST API with support for parallel loading of compressed files. A deeper integration of Hadoop and EXASolution that requires no import/export operations can be achieved by using the Hadoop Integration Service (see section 4.5).

4.4 Time Zone Support

In addition to the standard TIMESTAMP type, EXASolution supports the data type TIMESTAMP WITH LOCAL TIME ZONE – a feature that is useful for global businesses and organizations, as it allows entities in different locations around the globe to interact seamlessly via shared data. Internally, all such timestamps are stored at the time zone Coordinated Universal Time and all operations on such timestamps are performed using Coordinated Universal Time, guaranteeing that no information is lost due to conversion between different time zones. When it comes to interfacing with the outside world, user session variables control which time zone is used to interpret or display the data.

4.5 Hadoop Integration Service via eUDF

Section 3.5 describes EXASolution's support for eUDFs. One popular example of an eUDFs is EXASOL's Hadoop Integration Service, which allows you to transparently integrate Hadoop queries with SQL queries and which is

implemented via the eUDF framework. While EXAloader supports the import and export of files and tables from HDFS to EXASolution and vice versa (see section 4.4), with Hadoop Integration Service, SQL queries can process the results of Hadoop jobs and Hadoop jobs can work with the data provided by EXASolution SQL processes without import/export operations. Given EXASolution's versatile handling of complicated sub-selects, rather challenging SQL-Hadoop integration scenarios can be modelled easily. For instance, with Hadoop Integration Service, it is straightforward to integrate a Map-Job executed in Hadoop with an in-memory Reduce-Job executed in EXASolution.
Technically, the Hadoop Integration Service consists of Hadoop input/output formats that encapsulate the communication with EXASolution via the eUDF protocol.

4.6 Python and R Packages

EXASolution comes with Python and R packages for easy and idiomatic interaction with EXASolution directly from those languages. This feature makes EXASolution available right inside of standard tools like RStudio or IPython and allows for easy integration inside of data science workflows.
On a basic level, the packages' functionality allows you to load data from EXASolution into native Python or R data structures via the "readData" function and to write data structures from Python or R into EXASolution using the "writeData" function. These commands work similar to ODBC connections but operate more than an order of magnitude faster.
Using the "createScript" function, it is possible to define functions right from Python or R that are automatically mirrored into the connected EXASolution database. On the Python and R side, wrapper functions are created that transparently communicate with the database. When such a function is called, one can specify a table that should be used as input for the function. When it is executed, it runs massively parallel in the EXASolution cluster and finally returns its results into the Python or R environment.
Using the Python or R package, data scientists are able to create large-scale solutions using their familiar everyday development tools.

4.7 SQL Preprocessor

SQL Preprocessor is a feature specifically designed for consultants and solutions 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 often 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 EXASolution. Other use cases are user-defined short cuts or the ad hoc implementation of missing or experimental functionality.
SQL Preprocessor is integrated into all EXASolution 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 EXASolution in order to generate the actual SQL query to be executed.

5 Summary

EXASolution is best known for its extraordinary in-memory performance, underlined by its dominance in the TPC-H benchmark for analytical databases. But this is only half the story. With EXASolution, system performance is complemented by a deep concern for usability and business integration and operation. This paper gives just a number of hints about what EXASolution has to offer. It is the strong line-up of successful users – operating in a wide range of different fields – that is the practical proof that EXASolution is well suited as central component in great businesses.

If you want to learn more about EXASolution, we invite you to get in contact with us today (see EXASOL.com for how to do that). If instead you want to get your hands on EXASolution sooner, we ask you to check out one of the free trial options on http://www.exasol.com/en/test-drive/.