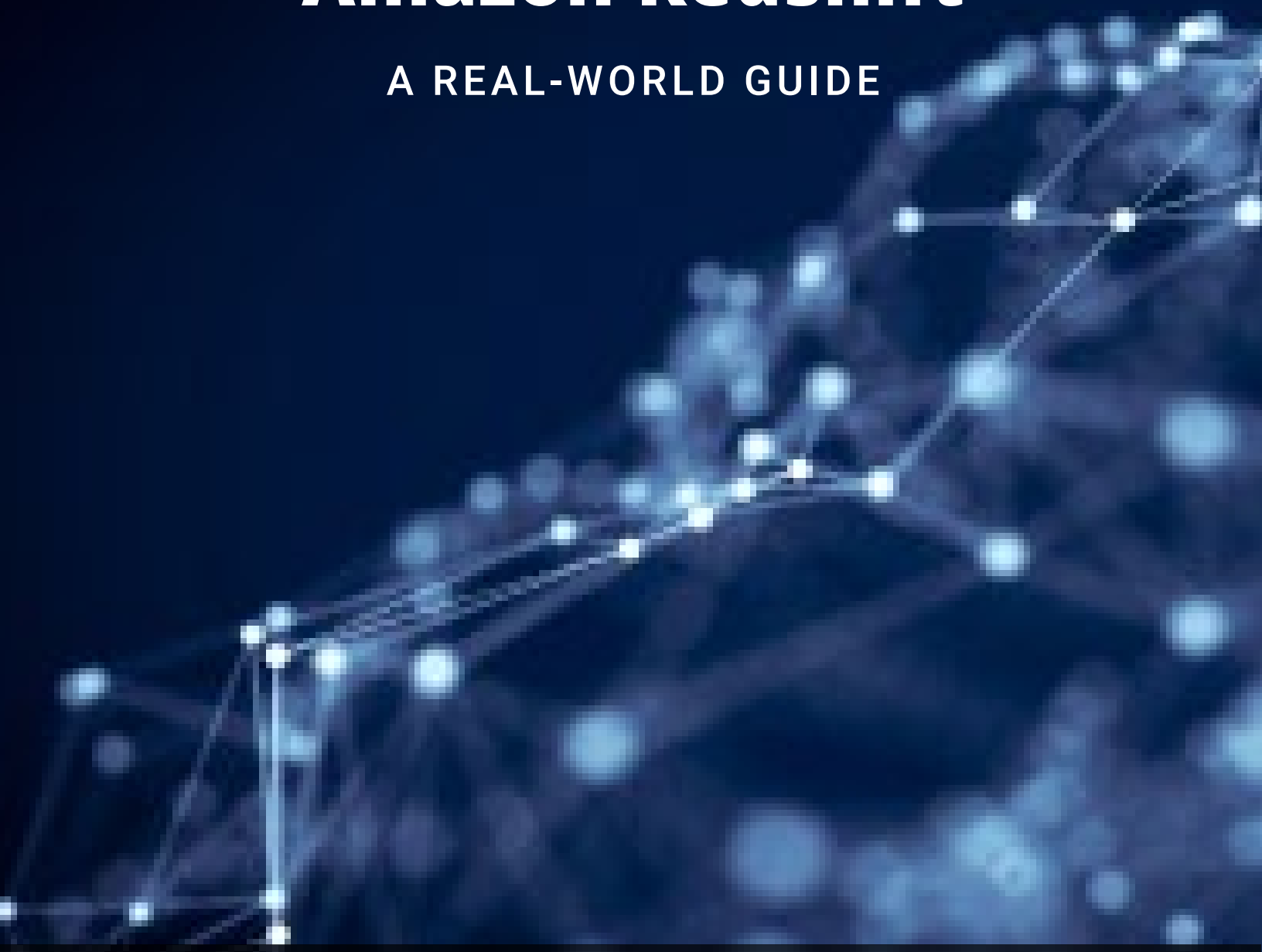
Optimizing Amazon Redshift

A REAL-WORLD GUIDE



MATILLION

aws

TABLE OF CONTENTS

About this book

Chapter 1: Amazon Redshift Architecture. Look under the hood" to understand Amazon Redshift's technical workings. Gain insight into how Amazon Redshift Spectrum provides superior scalability when compared to other cloud offerings.

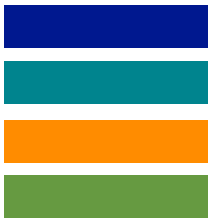
Chapter 2: Data Loading Optimizations. Learn best practices for how to efficiently load data into Amazon Redshift to improve business performance and to save costs.

Chapter 3: Data Querying Optimization. Make the most of Amazon Redshift's power to query and transform data.See how you can best scale your compute power to load and query large datasets by using Amazon Redshift Spectrum.

Chapter 4: Manaing Costs on Amazon Redshift. Save costs by leveraging Amazon Redshift's pay-as-you-go pricing model, making Amazon Redshift more cost efficient than on-premise alternatives. Learn how you can better forecast, limit, and control your costs.

Key

When reading the eBook, look out for colored boxes for additional resources, how to's, best practices, and hints and tips.

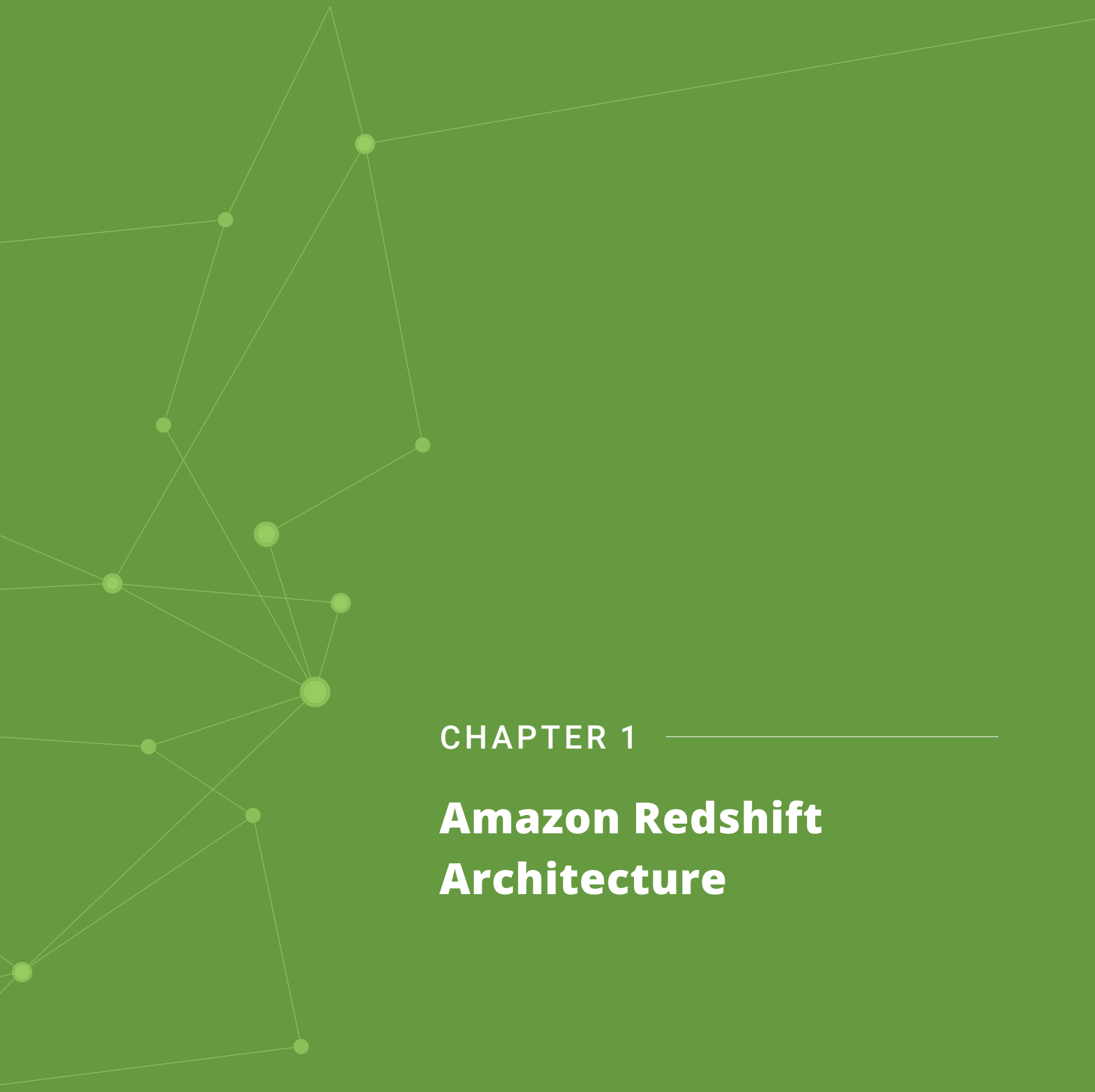


Further details and resources

'How to' information and examples

Best Practices and optimizations

Hints and tips from Matillion



CHAPTER 1

Amazon Redshift Architecture

CHAPTER 1

Modern businesses possess and continually generate greater volumes of data than ever before. To gain value from this data, businesses are seeking methods to leverage it as a means to make better, more informed decisions. The challenge they face is how to get that data across its various locations and shapes into a cloud data warehouse and in an analytics ready form - not a trivial task.

Many choose cloud data warehouses as a centralized repository to empower analytics-based decision making from this variety of data - but you likely have already taken that step and are either evaluating or are currently making use of Amazon Redshift as your cloud data warehouse. Selecting Amazon Redshift puts your business ahead of the curve (and your competitors, with their outdated on-premises databases) and gets you further down the path towards turning data-driven decision making into a normal, every day event for your business.

Whether you're a data warehouse developer, a data architect or manager, a business intelligence specialist, an analytics professional, or a tech-savvy marketer, you now need to make the most of the Amazon Redshift's platform to get the most out of your data. That's where this eBook comes in handy. It introduces the Amazon Redshift warehouse and helps you understand the various options available for managing your data and transforming your business' data into decisions.

This eBook is brought to you by Matillion. Matillion is an industry-leading data transformation solution that was originally introduced on Amazon Web Services for the Amazon Redshift cloud data warehouse. Delivering a true end-to-end data transformation solution (not just data migration or data preparation), Matillion provides an instant-on experience to get you up and running in just a few clicks, a pay-as-you-go billing model to cut out lengthy procurement processes, as well as an intuitive user interface to minimize technical pain and speed up time to results. Matillion's data transformation solution is available for Amazon Redshift on the AWS Marketplace. More information is available at **www.matillion.com**.

What is Cloud Data Warehousing?

Cloud Data Warehousing applies the principles of cloud-based economics, scalability, and performance to modern data warehouse implementations. Given that today's businesses generate staggering amounts of data, learning to get the most value from that data is paramount to success.

Just as Amazon Web Services (AWS) transformed general IT infrastructure by delivering on-demand, scalable, fast, and cost-effective computing resources, Amazon Redshift does the same for data warehousing and big data analytics.

Organizations choose Amazon Redshift for its affordability, flexibility, and powerful feature set:

- Massively Parallel Processing (MPP) columnar data store spins up quickly and can process billions of rows of data at the cost of just a few cents an hour
- Enterprise-class relational database query and management system
- Supports client connections with many types of applications, including business intelligence (BI), reporting, data, and analytics tools
- Execute analytic queries to retrieve, compare, and evaluate large amounts of data in multiple-stage operations

Amazon Redshift

Amazon Redshift is designed for performance and ease of use, reducing development time and saving money on data storage and compute for your business.

1.1 How is Amazon Redshift different?

Businesses have been using proprietary, on-premises, RDBMS platforms as Data Warehouse back-ends for decades. Even when clustered, such platforms fail to deliver cost effectiveness, flexibility, scalability, and performance compared to Amazon Redshift.

Traditional RDBMS (Oracle, SQL Server) and MPP (Netezza, Teradata, Vertica) platforms represent a large capital expenditure for your data warehousing organization, particularly when proprietary license fees are required. These licenses vary from expensive to extraordinarily expensive, with open source platforms also requiring the purchase of costly hardware. Owning hardware requires high-priced configuration and maintenance, and often hardware must be purchased ahead of the demand for it. This means you're paying for excess capacity that is ultimately limited and won't be able to grow without further investment.

If you are migrating from an on-premise databases, such as MySQL, Oracle, DB2, PostgreSQL, Teredata or any other JDBC compliant RDMS, to Amazon Redshift, **Matillion** has an easy to configure Database Query Component, getting your business data to the cloud with speed and simplicity.

Amazon Redshift, on the other hand, offers a pay-as-you-go model which allows you to smoothly scale your capacity and costs in line with your organization's demand for data warehousing capabilities.[1] Amazon Redshift compute nodes can be sized up or added with a few clicks - without the burden of expensive licenses. Plus, Amazon Redshift is primarily a managed solution allowing you to focus on the data and developing your warehouse rather than maintaining it. On top of that, Amazon Redshift scales almost limitlessly with the ability to grow, process, and query datasets up to petabyte scales.

1 Amazon also offers a Reserved Node payment model, which offers a large discount in exchange for a 1- or 3-year commitment. See the section below on **Using Reserved Nodes.**

1.2 Under the Hood of Amazon Redshift

Amazon Redshift instances are delivered as a cluster, containing both a leader node as well as a number of compute nodes. You decide at the cluster level whether to optimize your Amazon Redshift implementation for computing power or high storage capacity. Specifying dense storage nodes creates a cluster optimized for huge data volumes with hard disk drives (HDDs), while the dense compute option uses solid state disks (SSDs) for faster performance. Clusters can be resized at any time as long as the cluster itself remains large enough to house the amount of data it already contains. In the past, cluster resizes needed to be planned carefully as they caused the database to go into "read only" mode during the resize operation; manual cluster resizes still incur such outages. Recently, however, Amazon added Elastic Resize, which minimizes downtime.

> Please see **Resizing Clusters documentation** from Amazon for more details.

As noted above, each Amazon Redshift cluster is made up of a leader node and at least one compute node. The leader node serves as the point of communication for your application, which presents itself as single, cohesive data warehouse even though, behind the scenes, the leader node is actually coordinating activity across as many as 128 compute nodes. From the outside, the leader node looks like a Postgres database, on which Amazon Redshift's core technology was originally based.

The actual work the user requests of the cluster is done by the its compute nodes, each one a virtual machine. Each node contains multiple slices, which are roughly analogous to a processor or core allocated to work on the data contained within that particular node. The leader node allocates work to the compute node, which further distributes the work among its available slices. The correct use of slices allows the cluster make use of its massive parallel processing (MPP) capabilities. When the cluster allocates work to a node, the node can further divide this work down across its available slices as long as the data is distributed appropriately.

> More on this in our **Massively Parallel Processing** section below.

The more nodes you have, the more work your Amazon Redshift cluster is capable of. However, in the absence of best practices, additional nodes can yield negligible (or in fact negative) real world performance and scalability gains. It's also worth bearing in mind that the more nodes your Amazon Redshift cluster has, the more expensive it is to run.

Optimum performance depends on following best practices for schema design to ensure the right data is distributed to the right nodes. Let's consider some of Amazon Redshift's more advanced features and then review Amazon Redshift optimization and best practices.

1.3 Advanced Features

COLUMN ORIENTATION

Traditional RDBMS platforms are row oriented: their queries must fetch entire records from disk. In many cases, this approach incurs large amounts of unnecessary work, especially when the end user is only interested in a small subset of columns from the tables in question.

If you were implementing a data warehouse on one of these traditional RDBMS platforms you would likely design a star-schema, populate it with data, and then index the fields that users want to filter and group by. But because knowing all the queries your user base will ask in advance is very difficult, you end up indexing everything. At which point, you have got two complete copies of your data: one in the main tables, and one in the indices.

Columnar data stores, like Amazon Redshift, solve this problem by conceding that every column of interest is going to need an index. It thereby does away with the main data store. In this way, Amazon Redshift's architecture combats unnecessary, row-level work by organizing itself with a column orientation, and only performing input and output (I/O) operations for the columns required by a given query. Additionally, and because you can't know in advance what fields your users want to filter on, columnar data stores create a data structure similar to an index for every column of data, thus eliminating the tabular data store. As such, each column only exists in Amazon Redshift as an indexed structure. All values of the same type and those with similar values are organized next to each other in the indices, which makes compression far more efficient than in traditional relational database management systems.

- Terrific compression, since values in a single column are likely to have similar values which compression can take advantage of.
- Columns not included in a query are never read from disk, which can improve performance.
- On-technical users can be allowed to explored without hitting limits, because views, keys and indexes to be don't need to be maintained in order to optimize query performance.

In most database platforms, "column encoding" refers to the character set used in the column (e.g. Latin-1 or Unicode). Not so with Amazon Redshift, which always stores data using UTF-8 and Unicode. Here, "column encoding" refers instead to the different compression strategies you can use for the data distributed within each column. You can allow Amazon Redshift to choose one for you, or you can select one manually based on your storage and performance preferences. Compression preserves your I/O channels' capabilities and increases throughput.

MASSIVELY PARALLEL PROCESSING (MPP)

Amazon Redshift is an MPP platform. Designing your tables to take advantage of this parallelism is critical to good performance across large datasets, which leverages Amazon Redshift's various distribution styles. The discussion below highlights the performance benefits of these different styles.

ALL Distribution

ALL is the simplest distribution style. Setting a table to use this distribution style instructs Amazon Redshift to make a complete copy of the table on every node in the cluster. The benefit here is that when you ask the cluster to execute a query that includes a join, each node executing that join has a local copy of the table. As such, Amazon Redshift isn't required to copy the data across the network from node to node to complete the query. Otherwise, if a particular node is tasked with completing part of a joined query but didn't have the required data locally, it would have to reach into other nodes in the cluster, negatively affecting query performance by incurring additional network chatter. The downside of using the ALL distribution style is that the table is duplicated on every node, which takes up additional space. This distribution style is, therefore, best suited to small reference or dimension tables that are frequently joined to larger, fact tables.

⊕ Simple distribution style that doesn't requiring copying the data, which improves performance.

⊖ This might not be ideal for large tables since ALL distribution duplicates tables on every node, using additional space.

EVEN Distribution

Specifying EVEN distribution balances table rows on each node in the cluster. Queries involving that table are then distributed over the cluster with each slice on each node working to provide the answer in parallel. This distribution style is an especially optimal choice when tables aren't involved in joins. Conversely, joins involving rows matched by tables on different nodes increases network traffic and diminishes query performance. The exception is when joining from tables using EVEN to tables using an ALL distribution style since the second table's records are already available on each node in the cluster.

⊕ Use EVEN distribution when you are not using joins.

Automatic Distrubtion

Absent another distribution style specification, Amazon Redshift will use an automatic distribution. Tables start out with an ALL distribution style and then are switched to EVEN distribution as the table's volume grows.

NOTE: USING AUTOMATIC DISTRIBUTION

Note that when Automatic Distribution changes from ALL to EVEN, is one directional; Amazon Redshift will never reverse the distribution style from EVEN back to ALL for tables with low volumes.

KEY Distribution

With a KEY distribution style, you specify a column to distribute on and then, cleverly, Amazon Redshift ensures that all the rows with the same value of that key are placed on the same node. An exemple use case of a KEY distribution would be to optimize a join between two large tables, each using the same column or columns for its distribution key. Here, you don't want to set either table to ALL, because duplicated storage would be too expensive. EVEN distribution, throughout the course of the join, would likely force each table segment to interact with every segment in the other table, creating large amounts of network congestion and poor performance. Instead, distributing both tables on their common key or keys and then joining on those fields will perform well, because the common keys in both tables are guaranteed to be co-located on the same nodes.

⊕ Using KEY distribution ensures that common keys are handled by the same node, which is ideal for joining large tables as it will cost less than other distribution options.

1.4 Amazon Spectrum

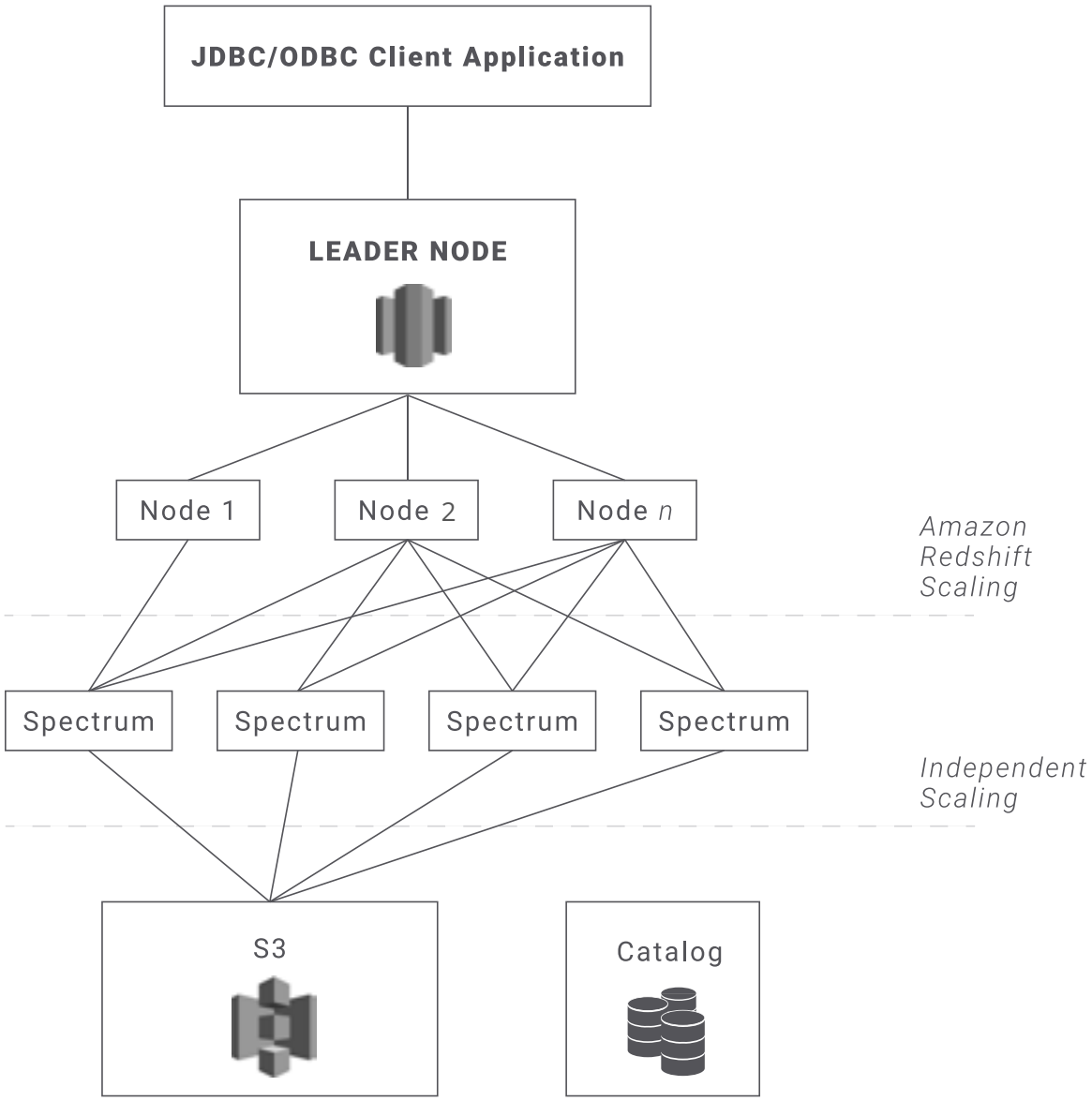
Historically, when a data warehouse reached capacity, users would have to go through a time-consuming approval, procurement, and implementation process before a they could finally add and utilize improved warehouse capacities. Then, users would spend the next year or so consuming the capacity they added, only to run out again at a later time.

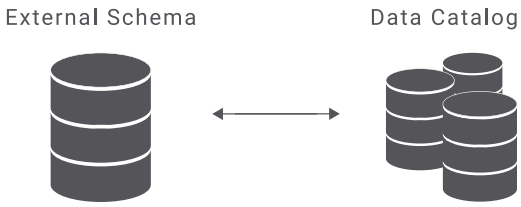
Amazon Redshift offers a huge improvement over this process, allowing a cluster to be resized with a few clicks, or even scaled automatically. Still, manual resize operations (now known as "classic resizes") terminate all executing queries both when the resize command is issued and then again after it completes. The Amazon Redshift cluster also remains in read-only mode during classic resizes when the operation is done. Elastic resizes offer improvements here, but some downtime is still required in elastic mode.

For more information on resizing check out AWS' documentation on **Elastic Resize**.

Amazon Spectrum allows users to write SQL against very large datasets stored in Amazon Simple Storage Service (S3), without having to load them into Amazon Redshift. Amazon Spectrum utilizes thousands of nodes in order to scale independently and run rapid queries on 'External' tables. It also allows for tables to be seamlessly joined across Amazon Redshift and Amazon S3.

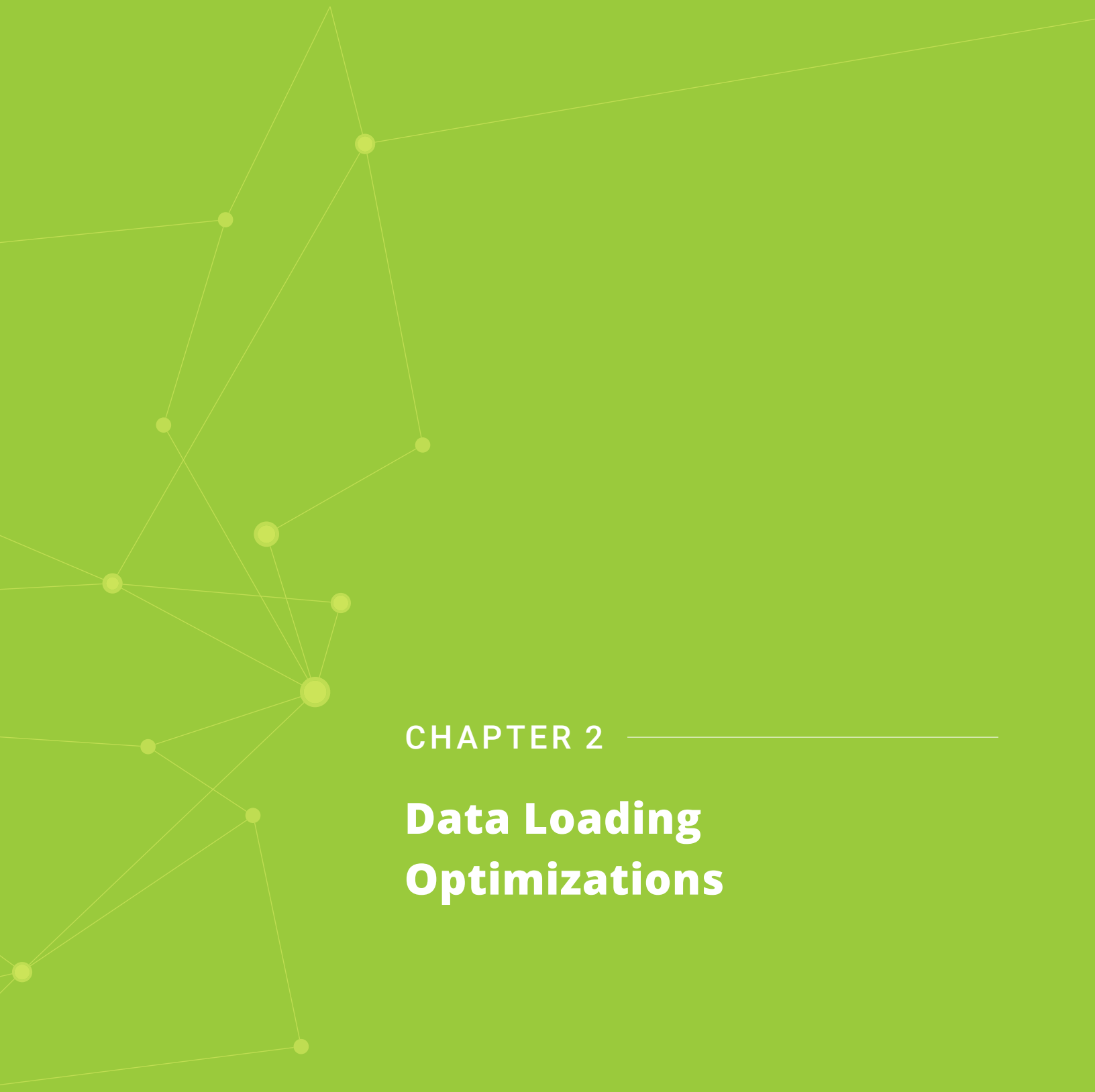
With Amazon Spectrum, data volumes can keep growing on Amazon S3 without interrupting data processing to scale up capacity since it scales independently and without any downtime. With this service, users can now scale to accommodate larger and larger amounts of data than the Amazon Redshift cluster would have otherwise been capable of processing with its own resources.





NOTE: USING AMAZON ATHENA AND SPECTRUM

Amazon Spectrum also shares a catalogue with Amazon Athena and Amazon Glue, so they can be used together. This is useful if you are pursuing a Data Lake strategy in Amazon S3 as it makes the same datasets available to the other Amazon services you are using.



CHAPTER 2

Data Loading Optimizations

CHAPTER 2

Turning now back to Amazon Redshift (as opposed to Amazon Spectrum), let's consider how to get the most out of AWS' warehousing platform.

Designing for Performance

Following the best practices below as you create a schema saves you from 1) time-consuming schema reorganizations as well as 2) having to compose complicated queries later to get the answers you need.

Amazon Redshift makes it simple to start gaining actionable insights from your data. Here are some techniques we've identified that can help streamline the load and query processes, saving you time and leveraging your resources in the most effective way. Once your Amazon Redshift schema contains data, changing it will be time-consuming.

2.1 Best Practices

DATA LOADING OPTIMIZATION

There are several ways to use Amazon Redshift to move data. Based on our experience, we've created a list of best practices to help make your usage as productive and smooth as possible.

COPY Command

Use the COPY command to load data into Amazon Redshift. The first time you use the command to populate an empty table, use a significant and representative dataset so Amazon Redshift can evaluate it properly and optimize compression for your particular data distribution.

A single COPY command can be used to load multiple files simultaneously. The COPY command treats multiple files as one, large logical file, so the load will happen in parallel and at much higher speeds. Ideally, the number of files should map to a multiple of the number of slices in the Amazon Redshift cluster - each slice loads a single file. As such, the files should be of roughly equal size so the work is distributed approximately evenly.

⊖ Avoiding both multi-row and single inserts in favor of preparing an input file on Amazon S3. Nothing beats the COPY command for performance!

Numerous examples of how to employ the COPY command are available here: **Amazon Redshift COPY Examples**.

Table Distribution and Sort Keys

Define table distribution styles prior to loading in anticipation of their access paths.

Use the ALL distribution style for smaller dimension and reference tables

- Use EVEN for large, staging tables that are either 1) not joined to other tables or 2) are only joined to tables having an ALL distribution style.
- Use the automatic distribution style (by specifying no distribution style at all), which will start tables off using the ALL distribution style and switch them to EVEN as they grow.
- Use the KEY distribution style to distribute rows according to the values in a given column. This places matching values on the same node slice and facilitates joins between tables distributed on the same key.

⊕ Specifying the same column as both the sort and distribution key allows Amazon Redshift's optimizer to use a sort merge join, which is faster than a hash join operation.

As with distribution keys, it's best to specify sort keys prior to data loads by anticipating typical access paths for the table in question. Choose the best sort key, optimizing first for joins and then for filtering.

⊕ Use Spectrum to help with filtering and aggregating very large data sets. Filter and aggregate with Amazon Redshift Spectrum to take the load off the data warehouse. Then you can start joining data, which can be handled within Amazon Redshift.

Compression

Compress data files prior to (or during) your load operation. The CPU time spent is regained in reduced bandwidth requirements and faster data transfer times.

HELPFUL HINT: COMPRESSION ON SPECTRUM

Compressing your files before they're loaded into S3 decreases Amazon Spectrum query times and consequently reduces costs for both storage and query execution. Although AWS supports several compression schemes, certain compression types and file formats enhance performance more than others.

⊕ We recommend Parquet, which is both column-oriented and compressed. In testing, these files processed twice as fast as CSV and compressed CSV files. However, converting existing data files to the Parquet format can be time-consuming; doing so is only worthwhile if the data fits squarely into a write-once-read-often access pattern.

Bulk Loading Data

Use bulk insert operations to create copies of data already inside your Amazon Redshift cluster. These operations work quickly and can leverage Amazon Redshift's built-in functions to transform data into a different format on the fly. Bulk insert operations include inserting SELECT statements into existing tables as well as the CTAS (Create Table As Select) statement.

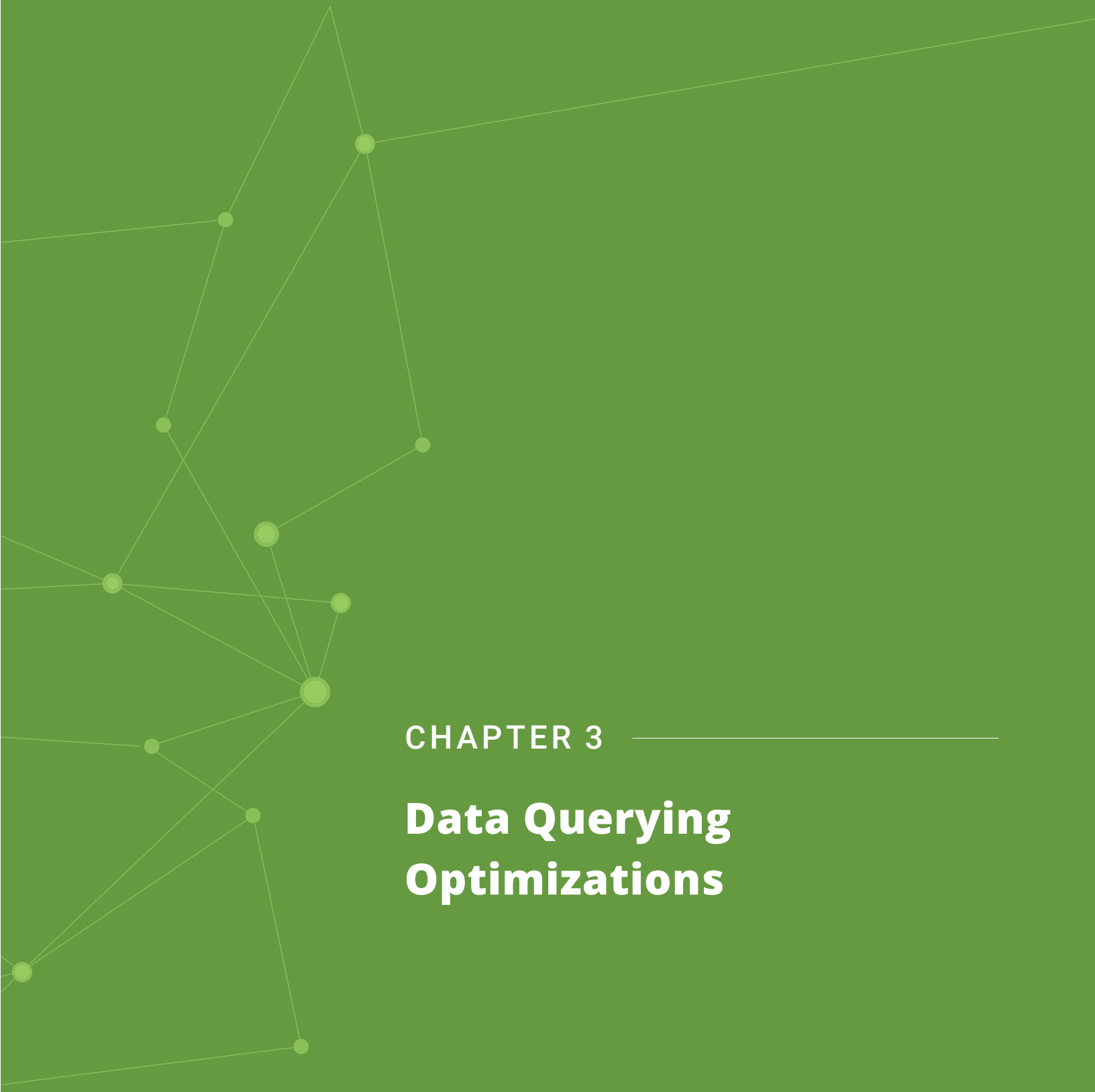
- ⊕ Efficiently move data from staging tables into persistent table structure.
- ⊕ Manipulate data using SQL functions built natively into Amazon Redshift.
- ⊖ Relies on user to clean up staging data after bulk load completes.

Organizing your Data

Organizing your data at load time means you can access this data more efficiently at a later time. For example, time series data with a limited retention period should be loaded into separate tables according to the periodicity you desire. Such tables should have the same table structure and only be differentiated by the time element specified within the table name. Doing so allows you to drop whole tables to prune the oldest data, which avoids the need to execute VACUUM commands after deletes.

- ⊕ Use date/time data types to store date values rather than storing string-based versions of the same. Doing so allows you to use Amazon Redshift's rich set of date functions.

Matillion ETL for Amazon Redshift has over 60 pre-built data connectors. Check our all our **integrations here**. Don't see what you need? We have a universal API connector and JDBC driver for enabled data sources.



CHAPTER 3

Data Querying Optimizations

CHAPTER 3

Designing for Efficiency

In addition to loading optimizations, there are some steps you can take to improve performance and reduce costs when querying your data.

3.1 Best Practices

DATA QUERYING OPTIMIZATION

To start fully benefiting from the power of Amazon Redshift, you'll want to start querying and transforming your data. Here are some best practices for querying data in a way that is efficient for Amazon Redshift, saving you critical resource.

Vacuum

Amazon Redshift recently added an auto-vacuum feature that removes fragmentation caused by delete operations. This automated VACUUM DELETE ONLY job runs at times when system use is low and pauses itself if user activity picks up while the job is running. However, this job only reclaims disk space for deletes and does not address any sort key maintenance needed as a result of updates. If you frequently update sort key columns we suggest scheduling VACUUM SORT ONLY operations during off hours to avoid users experiencing performance degradation due to resource scarcity.

⊕ Restores any sort key efficiency lost due to fragmentation

⊖ Time consuming operation best executed during off hours.

Workload Management Capabilities

Configure Workload Management (WLM) to ensure higher-priority queries always have access to the resources they need to complete in a timely manner. We suggest configuring separate queues for reporting, data transformation processes, superusers, and any other high-priority operations you run on Amazon Redshift. This ensures that no one class of users will overwhelm the cluster with queries that prevent others from getting their work done.

⊕ Prioritize your queries and manage your workloads, to ensure you have the resources needed, when you need them.

ANALYZE Commands

Maintain up-to-date table statistics by running ANALYZE commands automatically on load, especially during the COPY processes. Doing so gives Amazon Redshift's query optimizer the statistics it needs to determine how to run queries with the most efficiency.

If, you are making use of Amazon Redshift Spectrum features, there is no equivalent ANALYZE command to use. However, you can specify a numRows property on Spectrum objects and setting these values accurately helps the optimizer do a better job when developing execution plans.

- ⊕ Use Analyze commands to make sure Amazon Redshift can build the most optimal query execution plans.
- ⊕ Spectrum tables don't have an ANALYZE equivalent. Instead, the numRows property can be used.
- ⊖ Like VACUUM, ANALYZE commands can consume significant resources; be sure to run them when users won't be affected or use WLM settings to minimise their impact.

Review Execution Plans

Use the EXPLAIN command to review query execution plans. Note that MERGE joins are the best and most performant join type in Amazon Redshift (unlike traditional RDBMS systems). Wherever possible, eliminate the HASH joins in your execution plans in favor of MERGE joins by joining on columns that are included in both tables sort and distribution keys.

- ⊕ EXPLAIN commands allow you to review your execution to identify and address performance bottlenecks.
- ⊖ Avoid HASH joins as these will be less performant and may make queries cost more. Instead use MERGE commands to reduce costs and improve performance.

Scale Up

It may be that your query is running optimally and simply doesn't have the resources it needs to provide an adequate response time. The beauty of cloud-base data warehouses is the ability to scale them up to bring more computing power to bear. Consider scaling your cluster up to leverage a more powerful cluster against your response time requirements. As volumes grow, you can scale to adapt to more and more data, which lets you leverage pay-as-you-go cloud economics.

- ⊕ Scale your cluster to ensure you have the resources to handle growing data volumes.
- ⊖ Scaling up will costs more, so make sure you're adding capacity that's needed.

Partitioning Data

If you are using Amazon Redshift Spectrum, we advise partitioning your data. Partitioning allows you to use one table definition with multiple files, each a subpart of that same table. This can increase your query efficiency and potentially reduce the cost of executing your queries. You can partition tables so that you have different data files in different places on Amazon S3, which can then be queried as if they were a one large table. Date partitions can also be used to quickly and easily prune data to filter out unneeded records and focus on a specific subset. This is especially useful for time series data with which we suggest creating a table that defines the partition according to some time element, i.e. year, month, or day. This reduces costs by allowing Spectrum to process only partitions that are relevant to the query. Check your execution plans to confirm that the partition filter is taking place in Spectrum.

⊕ Follow **Amazon's best practice for partitioning data**:

- Use columns that you commonly filter by as partitions.
- Conduct partition pruning.
- Consider "file placement, query pattern, file size distribution, number of files in a partition, number of qualfied partitions" which also affect performance.

Amazon Redshift's flexibility, and native ability to interact with standard SQL and your existing Business Intelligence (BI) tools make it an affordable solution for running complex analytic queries against massive volumes of structured data. For organizations that want to discover the value in their data and use it to drive innovation and sales, Amazon Redshift is the ultimate solution.

3.1 Integration with Other AWS Services

As noted in the **Amazon Spectrum section above**, Amazon Redshift can use Amazon Spectrum to take advantage of both Amazon Glue and Amazon Athena. Amazon Redshift also natively integrates with other AWS services.

Amazon Redshift uses the COPY command to bulk load data into the cluster and a number of other AWS services can serve as the source data for this command, including Amazon S3 and Amazon DynamoDB. Third party solutions that help with data ingestion and transformation can help automate such data loads. Matillion, for example, offers integration with AWS CloudWatch for publishing custom metrics such as the number of successful data load and transformation job executions, their durations, and the record counts such jobs have transferred.

Matillion works with AWS to help you automate and monitor your data transformation efforts. Send notifications using the AWS SNS and SQS services to give your business confidence that data ingestion and transformation jobs will run as expected. In the event a job doesn't run, you will be notified immediately so any underlying issues can be resolved, keeping data analytics and reporting on track! You can also take advantage of a built-in scheduler to ensure your Amazon Redshift tables are all updated consistently based on time or event.

CHAPTER 4

Managing Costs on Amazon Redshift

CHAPTER 4

While Amazon Redshift is by nature a cost efficient, fully managed data warehousing solution with a pay-per-use model, there are some steps you can take to further control your data warehousing costs. Such techniques include sizing down your node size and count during low-traffic periods, pre-purchasing reserved nodes, and enabling cost monitoring.

4.1 Sizing Down

Sizing down refers to your ability to shrink an existing cluster so that it consumes less resources and is therefore less expensive. Many customers do so in the evening or on the weekend, when traffic is low and less stringent SLAs may be in place.

Amazon Redshift allows you to control both the type and number of nodes that comprise your cluster. Nodes are available as a small 2 to 4 vCPU virtual machine (VM) or as much larger 32 to 36 vCPU VMs. Storage capacities vary proportionally to the node size VM you've selected, depending on the broad category of node you've selected, either compute optimized (dense compute) or storage optimized (dense storage).

You are free to downsize the number and type of nodes in your cluster, but note that to do so the new cluster capacity must be able to accommodate the data volume you've already loaded into the cluster you're resizing. Also note that, as mentioned above, clusters undergoing a resize are put into read-only mode. The resize operation also terminates all currently executing queries when the resize command begins as well as when it completes.

4.2 Using Reserved Nodes

Regarding Reserved Nodes, this pricing model makes the most sense for Data Warehouses that are well established with predictable resource consumption. As such, a significant discount on the Amazon Redshift service can be obtained by committing to use Amazon Redshift for a one-year or three-year term.

The discount varies by how much your organization is willing to pay upfront. If that amount is zero, you can still obtain a discount for a one year commitment. Partial upfront payments allow for an even greater discount, and full payments confer the largest discount of all.

> See the AWS documentation on **Purchasing Amazon Redshift Reserved Nodes** for the most up to date details.

4.3 Cost Monitoring

AWS offers cost monitoring through its CloudWatch service. Once enabled, you can specify notification thresholds that, when breached, will email subscribers with an alert. Using CloudWatch as a cost monitor can head off billing surprises by alerting you when you reach budget milestones, or well before that depending how you choose to configure your thresholds.

4.4 Total Cost of Data Analytics

Finally, note that Amazon Redshift's pay-as-you-go model and near limitless scalability gives you the ability to apply the same principles to other areas of your business intelligence technology stack. Traditional ETL (Extract, Transformation, and Load) software requires a dedicated server with hefty compute, memory, as well as storage resources. Such software packages need these resources to land large extract files from source systems and manipulate those extracts before loading them into the data warehouse. On-premises ETL servers run into the same scalability issues as older data warehouses: they essentially waste their expensive and unused capacity until they run out of the same. Plus, ETL software is extremely expensive to license. These licenses often require extra fees for higher data volumes and charge even more for connector add-ons that give you access to the source systems you need to load into your warehouse.

Modern data transformation solutions that use an ELT (Extract, Load, and Transformation) architecture offer major improvements over this model. Data transformation solutions, such as **Matillion**, instruct Amazon Redshift to use its scalable and massively parallel processing capabilities to load your data into the cluster then transforming it into the form you specify once that load completes. Modest computing capabilities suffice to generate these instructions for Amazon Redshift and allow you to leverage the investment you've already made in your cloud data warehouse platform, which handles the heavy lifting required during the ELT process. Additionally, Matillion offers an hourly, pay-as-you-go model that includes both free support and access to all of its connectors for that same hourly fee. Matillion's approach to data transformation wouldn't be possible without Amazon Redshift's innate scalability, which provides the resources needed for extract, load, transform operations and allows business intelligence organizations to forgo the more costly and less flexible ETL pattern.

With consolidated billing and a pay-as-go model, Amazon Redshift and third party solutions like Matillion can help you get more value and reduce the total cost of your data analytics needs.

Conclusion

CONCLUSION

We hope you enjoyed this eBook and that you have found some helpful hints and tips on how to make the most of your Amazon Redshift cloud data warehouse. Implementing the best practices and optimizations described in this eBook should greatly enhance big data analytics performance and reduce your Amazon Redshift costs. This way you can spend less resource on overhead and focus on what's really important - answering your organization's most pressing business questions.

About Matillion

Matillion is an industry-leading data transformation solution that was originally introduced on Amazon Web Services for the Amazon Redshift cloud data warehouse. Delivering a true end-to-end data transformation solution (not just data migration or data preparation), Matillion provides an instant-on experience to get you up and running in just a few clicks, a pay-as-you-go billing model to cut out lengthy procurement processes, and an intuitive user interface to minimize technical pain and speed up time to results. Matillion's data transformation solution is available for Amazon Redshift on **AWS Marketplace**.

More information is available at **www.matillion.com**.

David Lipowitz, Solution Architect
Dayna Shoemaker, Product Marketing Manager

Speed. Savings. Simplicity. Scalability.

Amazon Redshift is the cloud data warehouse leader. We are the leader in data transformation. Use both for a best-in-breed data analytics solution.

- **An instant-on purchase experience via the AWS Marketplace, to get you up and running in just a few click**
- **A pay-as-you-go billing model, to eliminate lengthy procurement processes**
- **An intuitive user interface, to minimize technical pain and speed up time-to-results**

Get a Demo of Matillion

See first-hand how you can consolidate and aggregate ALL your business data on Amazon Redshift.

aws

MATILLION