

# AWS 上的数据仓库

2016 年 3 月



© 2016, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

## 版权声明

本文档仅用于参考。自本文件发行之日起，它代表了 AWS 当前的产品服务和实践，如有变更，恕不另行通知。客户负责对此文件的信息以及对 AWS 的产品或服务的任何使用进行自我独立的评估，每项产品或服务均按“原样”提供，无任何类型的保证，不管是明示还是暗示。本文档不形成 AWS、其附属公司、供应商或许可方的任何保证、表示、合同承诺、条件或担保。AWS 对客户承担的责任和义务受 AWS 协议制约，本文档不是 AWS 与客户直接的协议的一部分，也不构成对该协议的修改。

# 目录

摘要	4
简介	4
现代分析和数据仓库体系结构	5
分析体系结构	6
数据仓库技术选项	11
行式数据库	11
列式数据库	12
大规模并行处理体系结构	14
<b>Amazon Redshift 深入分析</b>	<b>14</b>
性能	14
持久性和可用性	15
可扩展性和弹性	15
接口	16
安全性	16
成本模型	17
适合的使用模式	17
不适用的模式	18
迁移到 Amazon Redshift	18
一步式迁移	19
两步式迁移	19
数据库迁移工具	19
设计数据仓库工作流程	20
总结	22
贡献者	22
阅读更多内容	23
备注	24

## 摘要

全球各大企业的数据工程师、数据分析师和开发人员都期待通过将数据仓库迁移到云来提高性能和降低成本。本白皮书讨论了实现分析和数据仓库体系结构的现代方法，简要介绍了 Amazon Web Services (AWS) 上可用于实施此体系结构的服务，并提供了利用这些服务来构建数据仓库解决方案的常见设计模式。

## 简介

如今，数据和分析对于企业来说是不可或缺的。几乎所有大型企业都已构建了数据仓库，以便使用从各种来源（包括企业自己的事务处理系统和其他数据库）获取的数据进行报告和分析。

不过，构建并运行数据仓库（用来保存从一个或多个数据源获取的信息的中央存储库）始终是一件复杂而费钱的事情。大多数数据仓库系统的设置都很复杂，前期投入的软件和硬件费用达到数百万美元，并且规划、采购、实施和部署过程需要花费几个月的时间。在您进行初始投资并建立数据仓库以后，您必须聘用一个数据库管理员团队，来确保您的查询能够快速运行并防止数据丢失。

此外，传统的数据仓库难以扩展。当数据量增加或者您需要向更多用户提供分析和报告时，您要么选择接受较低的查询性能，要么选择在成本高昂的升级过程中投入更多的人力和物力。实际上，一些 IT 团队出于维护现有服务水平协议的考虑而反对增加数据或添加查询。许多企业在艰难地维护着与传统数据库供应商之间的良好关系。这些企业常常被迫升级托管系统的硬件，或者开展旷日持久的协商来延长已过期的许可证。在达到一个数据仓库引擎的扩展限制时，他们将被迫迁移到同一家供应商提供的另一个使用不同 SQL 语义的引擎。

**Amazon Redshift** 极大地减少了与部署数据仓库系统相关的成本和工作量，并且不会牺牲功能和性能，从而彻底改变了企业对数据仓库的看法。**Amazon Redshift** 是一个完全托管的 **PB** 级高速数据仓库解决方案，让您能够使用现有的商业智能 (BI) 工具来分析海量数据，这个过程非常简单，而且经济高效。借助 **Amazon Redshift**，您只需花费十分之一的成本即可获得可执行大规模并行处理 (MPP) 的列式数据仓库引擎的性能。您可以从按使用量付费的小规模数据仓库（每小时 0.25 美元）开始，然后再扩展到 **PB** 级（每年每 TB 1,000 美元）。

自 2013 年 2 月发布以来，Amazon Redshift 一直是增长最快的 AWS 服务之一，拥有遍布各个行业的数千家不同规模的企业客户。NTT DOCOMO、FINRA、Johnson & Johnson、Hearst、Amgen 和 NASDAQ 等企业都已迁移到 Amazon Redshift。根据 [Forrester Wave: 企业数据仓库 \(2015 年第 4 季度\)](#) 报告，Amazon Redshift 在行业中处于领先地位。<sup>1</sup>

在本白皮书中，我们提供了您在实施数据仓库从本地到云的战略转移时需要利用的各种信息：

- 现代分析体系结构
- 适用于该体系结构的数据仓库技术选项
- 有关 Amazon Redshift 及其差异化功能的深入分析
- 使用 Amazon Redshift 及其他服务在 AWS 上构建完整数据仓库系统的蓝图
- 有关从其他数据仓库解决方案迁移到我们的合作伙伴生态系统的实用提示

## 现代分析和数据仓库体系结构

再次说明，*数据仓库*是用于保存从一个或多个数据源获取的信息的中央存储库。数据通常从事务系统和其他关系数据库传输到数据仓库中，而且一般包括结构化、半结构化和非结构化的数据。这些数据将会定期处理、转换和提取。用户包括数据科学家、业务分析师和决策者，他们通过 BI 工具、SQL 客户端和电子表格来访问数据。

为什么要彻底重新构建数据仓库，而不是直接在记录事务的联机事务处理 (OLTP) 数据库上运行分析查询？为了回答这个问题，我们来看一看数据仓库与 OLTP 数据库之间的差别。数据仓库是针对批量写入操作以及读取大量数据的操作进行优化的，而 OLTP 数据库是针对持续写入操作以及大量的小规模读取操作进行优化的。通常，数据仓库会因较高的数据吞吐量要求而使用非规范化架构（如星型架构和雪花型架构），而 OLTP 数据库则使用高度规范化的架构，这些架构更适合高事务吞吐量的要求。星型架构包含多个引用大量维度表的大型事实数据表。雪花型架构是星型架构的扩展，包含更加规范化的维度表。

在将数据仓库作为与源 OLTP 或其他源系统分开的单独数据存储来管理时，为了充分利用这种使用方式带来的优势，我们建议您构建高效的数据管道。此类管道会从源系统中提取数据，将数据转换为适用于数据仓库的架构，然后将该架构加载到数据仓库中。在下一部分中，我们将讨论分析管道的构建块以及可供您用来构建管道的各种 AWS 服务。

## 分析体系结构

分析管道旨在处理来自各种异构源（如数据库、应用程序和设备）的大量传入数据流。

典型的分析管道分为以下阶段：

1. 收集数据。
2. 存储数据。
3. 处理数据。
4. 分析数据和可视化数据。

有关说明，请见下面的图 1。



图 1：分析管道

## 数据收集

在数据收集阶段，需要考虑到您可能具有不同类型的数据，例如事务数据、日志数据、流数据和物联网 (IoT) 数据。AWS 针对上述每种数据提供了数据存储解决方案。

### 事务数据

事务数据（例如，电子商务购买交易和金融交易）通常存储在关系数据库管理系统 (RDBMS) 或 NoSQL 数据库系统中。所选择的数据库解决方案取决于具体的使用案例和应用程序特征。如果数据的结构化不太理想，不符合已定义架构的要求；或者架构经常发生变化，那么适合使用 NoSQL 数据库。另一方面，当事务分布在多个表行上并且查询需要复杂联接时，适合使用 RDBMS 解决方案。Amazon DynamoDB 是一种完全托管的 NoSQL 数据库服务，可用作您的应用程序的 OLTP 存储。利用 Amazon RDS，您可以为您的应用程序实施基于 SQL 的关系数据库解决方案。

### 日志数据

通过可靠地捕获系统生成的日志，您可以利用日志中存储的信息来解决问题、实施审核和执行分析。Amazon Simple Storage Service (Amazon S3) 是一个面向用于执行分析的非事务数据（例如日志数据）的常用存储解决方案。由于 Amazon S3 提供 11 个 9 的持久性（即 99.99999999% 的持久性），因此它还是一个非常不错的存档解决方案。

### 流式处理数据

Web 应用程序、移动设备以及许多软件应用程序和服务都会生成数量惊人的[流式处理数据](#)，有时每小时会生成数 TB 的数据。这些数据需要持续地进行收集、存储和处理。<sup>2</sup>利用 Amazon Kinesis 服务，您可以轻松自如地应对，且成本较低。

### IoT 数据

全球的设备 and 传感器在不断地发送着各种消息。企业发现，现在捕获这些数据并从中获得情报的需求在不断增长。借助 AWS IoT，各种互连设备能够轻松安全地与 AWS 云进行交互。AWS IoT 让您能够轻松地使用各种 AWS 服务（如 AWS Lambda、Amazon Kinesis、Amazon S3、Amazon Machine Learning 和 Amazon DynamoDB）来构建可用于收集、处理和分析 IoT 数据并根据这些数据采取操作的应用程序，而无需管理任何基础设施。

## 数据处理

通过收集过程，您可以获得可能包含有价值信息的数据。您可以分析已提取的信息来挖掘可帮助您拓展业务的情报。例如，您可以从这些情报中获悉用户的行为方式以及您产品的相对受欢迎程度。收集此类情报的最佳实践是，将您的原始数据加载到数据仓库中，然后执行进一步的分析。

为此，可以使用两类处理 workflows：批处理 workflow 和实时处理 workflow。最常见的处理形式包括联机分析处理 (OLAP) 和 OLTP，二者分别使用其中一类处理 workflow。联机分析处理 (OLAP) 通常采用批处理形式。相反，OLTP 系统是面向实时处理的，通常不太适合于批次形式的处理方式。您可以将数据处理与 OLTP 系统分离，使数据处理不会影响到 OLTP 工作负载。

首先，我们来了解一下批处理。

### *提取、转换和加载 (ETL)*

ETL 是从多个源中提取数据并将数据加载到数据仓库系统的过程。ETL 通常是一个具有精确定义的工作流的持续性过程。在这个过程中，最初将从一个或多个源中提取数据。随后，对提取的数据进行整理、扩充和转换，然后加载到数据仓库中。在 ETL 管道中经常会使用 Hadoop 框架工具（例如，Apache Pig 和 Apache Hive）来转换大量数据。

### *提取、加载和转换 (ELT)*

ELT 在 ETL 的基础上做了修改，在这个过程中，提取的数据将首先加载到目标系统中。在将数据加载到数据仓库中后，将执行转换。通常，当您的目标系统强大到足以处理转换时，非常适合使用 ELT 过程。Amazon Redshift 通常在 ELT 管道中使用，因为它可以非常高效地执行转换。

### *联机分析处理 (OLAP)*

OLAP 系统将聚合的历史数据存储于多维架构中。OLAP 系统广泛地用于数据挖掘，让您能够提取数据并在多个维度上敏锐地观察趋势。由于 Amazon Redshift 已针对快速联接进行优化，因此它通常用于构建 OLAP 系统。

现在，我们来了解一下实时数据处理。

### 实时处理

我们在前面介绍了流数据，并提到 **Amazon Kinesis** 是一个用于捕获和存储流数据的解决方案。您可以逐个记录或者在一段时间范围内，按顺序以递增方式处理此类数据，并将处理后的数据用于各种各样的分析（包括关联、聚合、筛选和采样）。此类处理称作实时处理。利用从实时处理中得到的信息，公司能够了解其业务和客户活动的许多方面（例如，服务使用情况（用于计量或计费）、服务器活动、网站点击率以及设备、人员和物品的地理位置）并及时应对新出现的情况。实时处理需要高度并发的可扩展处理层。

您可以使用 **AWS Lambda** 来实时处理流数据。**Lambda** 可以直接处理来自 **AWS IoT** 或 **Amazon Kinesis Streams** 的数据。利用 **Lambda**，您可以运行代码，而无需配置或管理服务器。

此外，也可以使用 **Amazon Kinesis Client Library (KCL)** 处理来自 **Amazon Kinesis Streams** 的数据。与 **AWS Lambda** 相比，在对传入数据进行批处理以便执行进一步的处理方面，**KCL** 更加灵活。您也可以使用 **KCL** 在处理逻辑中应用大量转换和自定义设置。

将流数据加载到 **AWS** 中的最简单的方法是使用 **Amazon Kinesis Firehose**。它可以捕获流数据并自动将流数据加载到 **Amazon Redshift** 中，从而使您能够利用现成的 **BI** 工具和仪表板执行近实时分析。您可以使用 **Firehose** 定义批处理规则，然后 **Firehose** 将负责对数据进行可靠的批处理并将数据传输到 **Amazon Redshift**。

## 数据存储

您可以将数据存储于数据仓库或数据集中，下文将对此进行讨论。

### 数据仓库

就如前面提到的，**数据仓库** 是用于保存从一个或多个数据源获取的信息的中央存储库。通过数据仓库，您可以使用 **BI** 工具对大量数据运行快速分析，并发现数据中隐藏的模式。数据科学家可以查询数据仓库来执行脱机分析并发现趋势。整个组织中的用户可以借助临时 **SQL** 查询、定期报告和仪表板的方式，使用数据来制定重要业务决策。

### 数据集市

**数据集市**是一种简单形式的数据库，侧重于某个特定的职能领域或主题。例如，您可以为组织中的每个部门创建特定的数据集市，也可以根据区域划分数据集市。您可以从大型数据库和/或操作型存储来构建数据集市。数据集市易于设计、构建和管理。不过，由于数据集市侧重于特定的职能领域，因此跨职能领域的查询会因分布情况而变得复杂。

除了数据库之外，您还可以使用 **Amazon Redshift** 构建数据集市。

### 分析和可视化

在处理数据并使其可用于进一步的分析后，您需要使用正确的工具来分析处理后的数据并实现数据可视化。

在许多情况下，您可以使用用于处理数据的相同工具来执行数据分析。可以通过 **SQL Workbench** 等工具，使用 **ANSI SQL** 来分析您在 **Amazon Redshift** 中保存的数据。**Amazon Redshift** 还能够与市场提供的常用第三方 **BI** 解决方案很好地配合使用。

**Amazon QuickSight** 是一项快速 **BI** 服务，以云为强大后盾，可让您轻松地实现数据可视化、执行临时分析以及快速地从数据中获得业务见解。**Amazon QuickSight** 与 **Amazon Redshift** 相集成，目前提供的是预览版，预计将于 2016 年晚些时候正式发布。

如果您将 **Amazon S3** 用作主存储，则执行分析和实现数据可视化的常见方式是在 **Amazon Elastic MapReduce (Amazon EMR)** 上运行 **Apache Spark** 笔记本。通过使用此过程，您可以灵活地运行 **SQL** 或执行用 **Python** 和 **Scala** 等语言编写的自定义代码。

**Apache Zeppelin** 是一个开源 **BI** 解决方案，提供另一种实现可视化的方法。您可以在 **Amazon EMR** 上运行 **Apache Zeppelin**，使用 **Spark SQL** 对 **Amazon S3** 中的数据进行可视化处理。您也可以使用 **Apache Zeppelin** 对 **Amazon Redshift** 中的数据进行可视化处理。

### 使用 AWS 服务的分析管道

**AWS** 提供一系列广泛的服务来实施端到端分析平台。图 2 说明了上文介绍的服务以及它们适用于分析管道中的哪个部分。

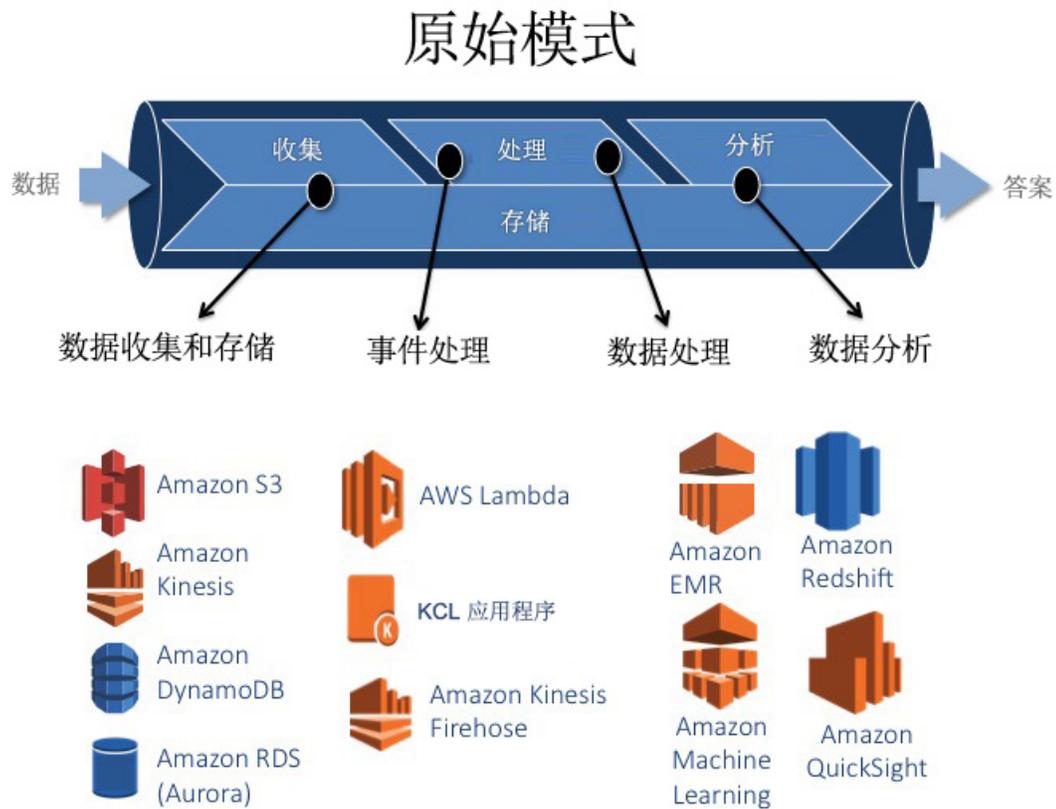


图 2：使用 AWS 服务的分析管道

## 数据仓库技术选项

在本部分中，我们将介绍可用于构建数据仓库的各种选项：行式数据库、列式数据库和大规模并行处理体系结构。

### 行式数据库

行式数据库通常在一个物理块中存储整行。通过二级索引可以实现读操作的高性能。Oracle Database Server、Microsoft SQL Server、MySQL 和 PostgreSQL 等数据库是行式数据库系统。传统上，这些系统用于数据仓库，但它们更适合于执行事务处理 (OLTP) 而不是分析。

为了优化用作数据仓库的行式系统的性能，开发人员使用了各种各样的方法，包括构建具体化视图、创建预先聚合的汇总表、基于每种可能的谓词组合构建索引、实施数据分区以通过查询优化程序利用分区修剪以及执行基于索引的联接。

传统的行式数据存储受单个设备上可用的资源的限制。数据集市通过使用功能分片，在一定程度上缓解了此问题。您可以将数据仓库拆分为多个数据集市，每个数据集市满足一个特定职能领域的要求。不过，当数据集市随着时间的推移而增大时，数据处理的速度会变慢。

在行式数据仓库中，对于块中的所有行，每个查询都必须通读满足查询谓词的所有列，包括您未选择的列。使用此方法会使数据仓库中出现明显的性能瓶颈，因为您的表具有许多列，而查询只使用其中几个列。

## 列式数据库

列式数据库将每个列组织到单独的一组物理块中，而不是将整行包装到一个块中。利用此功能，列式数据库能够对只读查询实现更高的 I/O 效率，因为它们只需从磁盘（或者从内存）中读取查询所访问的那些列。这种方式使得列式数据库比行式数据库更适用于数据仓库。

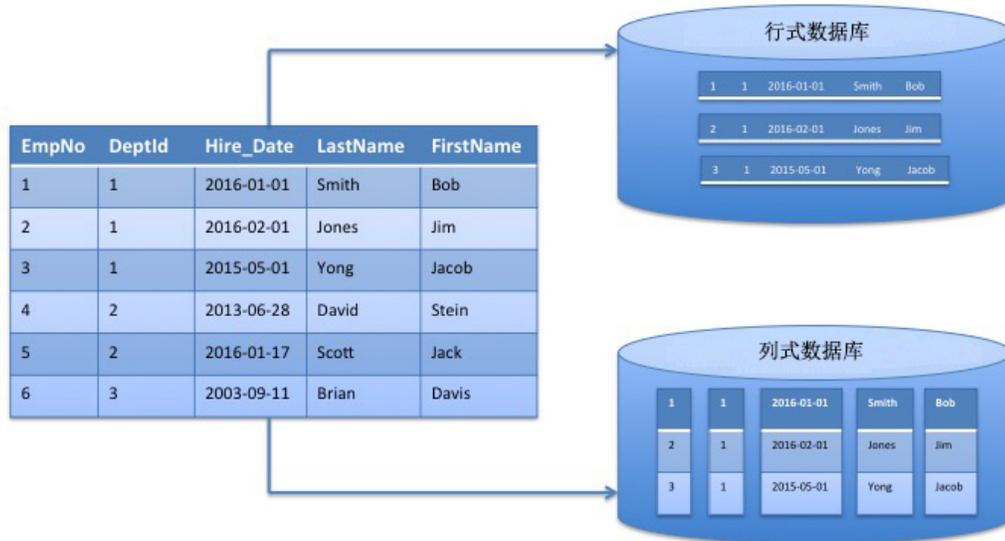


图 3: 行式数据库与列式数据库

上面的图 3 说明了行式数据库与列式数据库之间的主要差异。在行式数据库中，行将包装到各自的块中；而在列式数据库中，列将包装到各自的块中。

除了实现更快的 I/O 之外，使用列式数据库的又一个最大好处是改进了压缩。由于每个列都包装到各自的一组块中，因此每个物理块将包含相同的数据类型。当所有数据的数据类型都相同时，数据库可以使用极为高效的压缩算法。因此，与行式数据库相比，您需要的存储空间更少。这个方法还可以极大地减少 I/O，因为存储相同数据所使用的块更少。

用于数据仓库的一些列式数据库包括 Amazon Redshift、Vertica、Teradata Aster 和 Druid。

## 大规模并行处理体系结构

利用 MPP 体系结构，您可以使用集群中可用的所有资源来处理数据，从而大大提高了 PB 级数据仓库的性能。使用 MPP 数据仓库时，您只需向集群中添加更多节点，即可提高性能。Amazon Redshift、Druid、Vertica、GreenPlum 和 Teradata Aster 是一些基于 MPP 体系结构构建的数据仓库。诸如 Hadoop 和 Spark 这样的开源框架也支持 MPP。

## Amazon Redshift 深入分析

Amazon Redshift 采用了列式 MPP 技术，提供了高性能、经济实用的数据仓库所具备的主要好处，包括高效压缩、减少了 I/O 以及较低的存储要求。它基于 ANSI SQL，因此您只需对现有查询进行少量更改，甚至不需要进行任何更改，即可运行这些查询。因此，它已成为当今的企业数据仓库和数据集市的理想选择。在本部分中，我们将会深入分析 Amazon Redshift，并且更详细地介绍其功能。

通过使用列式存储并且在各个节点之间并行执行分布式查询，Amazon Redshift 能够针对几乎任何数据大小实现极快的查询和 I/O 性能。它会自动执行与设置、配置、监控、备份及保护数据仓库相关的大多数常见管理任务，因此管理起来简单且管理成本较低。利用这种自动化功能，您可以在几分钟内构建 PB 级数据仓库，而传统的本地实施需花费几周或几个月的时间才能完成此任务。

## 性能

Amazon Redshift 使用列式存储、数据压缩及区域映射，可降低执行查询所需的 I/O 数量。交错排序可实现高速性能，而不会造成维护索引或投影方面的开销。

Amazon Redshift 采用 MPP 体系结构，能够以并行的分布式处理方式来执行 SQL 操作，从而利用所有可用资源。基础硬件是针对高性能数据处理而设计的，使用本地附加存储来最大程度地提高 CPU 与驱动器之间的吞吐量，同时使用 10 GigE 网状网络在各个节点之间提供最高吞吐量。可以根据您的数据仓库需求来调整性能：AWS 提供了配备固态驱动器的密集计算 (DC)，还提供了密集存储 (DS) 选项。持续部署的软件升级可以不断地改进性能，而不需要用户进行干预。

## 持久性和可用性

为了提供可能最佳的数据持久性和最高可用性，Amazon Redshift 将自动检测并替换您的数据仓库集群中的任何故障节点。替换节点立即可供使用，并首先加载您最频繁访问的数据，以便您尽快地恢复数据查询。由于 Amazon Redshift 会镜像您整个集群中的数据，因此它使用来自另一个节点的数据来重新构建故障节点。在配置替换节点并将其添加到集群（通常只需几分钟时间）之前，集群处于只读模式。

Amazon Redshift 集群位于一个[可用区](#)中。<sup>3</sup>不过，如果您需要为 Amazon Redshift 设置多可用区，可以创建镜像并自行管理复制和故障转移。

只需在 Amazon Redshift 管理控制台中点击几次，您即可使用 Amazon Redshift 设置可靠的灾难恢复 (DR) 环境。可以在多个 AWS 区域内保留备份的副本。如果一个 AWS 区域出现服务中断，则可从其他 AWS 区域中的备份还原您的集群。只需花几分钟时间来启动还原操作，即可对集群进行读/写访问。

## 可扩展性和弹性

只需在控制台中点击几次或者使用一个[API 调用](#)，您就能在性能或容量需求发生变化时轻松地更改数据仓库中的节点的数量和类型。<sup>4</sup>使用 Amazon Redshift，您能够从最低 160 GB 的单个节点开始，然后通过添加多个节点进行扩展，一直到 1 PB 甚至更高容量的压缩用户数据。有关更多信息，请参阅 Amazon Redshift 集群管理指南中的[关于集群和节点](#)。<sup>5</sup>

在调整大小时，Amazon Redshift 会将您现有的集群置于只读模式，并配置一个选定大小的新集群，然后将数据从您的旧集群并行复制到您的新集群。在这个过程中，您只需为活动的 Amazon Redshift 集群付费。在配置新集群的同时，您可继续对您的旧集群进行查询。在将您的数据复制到您的新集群后，Amazon Redshift 会自动将查询重定向至您的新集群，并移除旧集群。

可以使用 Amazon Redshift API 操作以编程方式启动集群、扩展集群、创建备份、还原备份等。利用此方法，您可以将这些 API 操作集成到您现有的自动化堆栈中或构建可满足您需求的自定义自动化功能。

## 接口

Amazon Redshift 提供了自定义的 Java 数据库连接 (JDBC) 和开放式数据库连接 (ODBC) 驱动程序，您可以从控制台的 **Connect Client** 选项卡中下载这些驱动程序，这意味着您可以使用各种各样熟悉的 SQL 客户端。您也可以使用标准 PostgreSQL JDBC 和 ODBC 驱动程序。有关 Amazon Redshift 驱动程序的更多信息，请参阅 Amazon Redshift 数据库开发人员指南中的 [Amazon Redshift 和 PostgreSQL](#)。<sup>6</sup>

此外，您可以找到大量已经过验证的与多个[常见 BI 和 ETL 供应商](#)之间的集成的示例。<sup>7</sup>在这些集成中，将在每个计算机节点上并行执行加载和卸载，以最大程度地提高在多种资源（包括 Amazon S3、Amazon EMR 和 Amazon DynamoDB）中提取或导入数据的速率。可使用 Amazon Kinesis Firehose 轻松地将流数据加载到 Amazon Redshift 中，从而利用现有的 BI 工具和仪表板来执行近实时分析。可以使用该控制台或 Amazon CloudWatch API 操作来找到 Amazon Redshift 数据仓库集群的计算使用率、内存使用率、存储使用率和读/写流量之类的度量。

## 安全性

为了保障数据安全性，您可以在基于 [Amazon Virtual Private Cloud \(Amazon VPC\) 服务](#) 的 Virtual Private Cloud 中运行 Amazon Redshift。您可以使用 VPC 的软件定义的联网模型来定义防火墙规则，这些防火墙规则将根据您配置的规则来限制流量。<sup>8</sup>Amazon Redshift 支持客户端应用程序与 Amazon Redshift 数据仓库集群之间的 SSL 连接，这可以在传输期间对数据进行加密。

Amazon Redshift 计算节点可以存储您的数据，但数据只能从集群的领导节点进行访问。这种隔离进一步增强了安全性。Amazon Redshift 与 [AWS CloudTrail](#) 集成，使您能够审核所有 Amazon Redshift API 调用。<sup>9</sup>为了确保您的静态数据安全，Amazon Redshift 会在每个数据块写入硬盘时使用硬件加速的 AES-256 加密功能对数据块进行加密。此加密操作在 I/O 子系统中的较低级别进行；I/O 子系统会对写入到硬盘中的所有数据进行加密，包括中间查询结果。数据块按原样进行备份，这意味着备份也进行了加密。默认情况下，Amazon Redshift 负责密钥管理，不过您可以选择[使用自己的硬件安全模块 \(HSM\) 来管理您的密钥](#)，或者通过 [AWS Key Management Service](#) 管理您的密钥。<sup>10,11</sup>

## 成本模型

Amazon Redshift 不要求长期订阅，也不需要提前支付费用。利用此定价方法，在出现相应需求之前，您不必为规划和购买数据仓库容量而产生的资本费用以及由此带来的复杂性而头疼。根据集群中的节点的大小和数目收费。

您可以使用最高与您配置的存储同等大小的备份存储，而不需要额外支付费用。例如，如果您有一个包含两个 XL 节点（共 4 TB 存储）的活动集群，AWS 将在 Amazon S3 上提供最多 4 TB 的备份存储，不收取额外费用。如果您的备份存储超出了预配置存储大小，或者在您的集群终止之后存储备份，则将按标准 [Amazon S3 费率](#) 计费。<sup>12</sup>对于 Amazon S3 与 Amazon Redshift 之间的通信，不收取数据传输费。有关更多信息，请参阅 [Amazon Redshift 定价](#)。<sup>13</sup>

## 适合的使用模式

Amazon Redshift 非常适合使用现有 BI 工具执行的联机分析处理 (OLAP)。各个组织可以使用 Amazon Redshift 来完成以下工作：

- 运行企业 BI 和报告
- 分析多种产品的全球销售数据
- 存储股票交易历史数据
- 分析广告浏览量和点击量
- 汇总游戏数据
- 分析社会趋势
- 评估医疗领域的临床质量、运营效率和财务绩效

## 不适用的模式

Amazon Redshift 不太适合以下使用模式：

- **小型数据集** — Amazon Redshift 专为集群中的并行处理而构建。如果您的数据集小于 100 GB，则您将不会获得 Amazon Redshift 提供的所有好处，而 Amazon RDS 可能是更合适的解决方案。
- **OLTP** — Amazon Redshift 设计用于数据仓库工作负载，可提供极快的分析功能且成本较低。如果您需要的是高性能事务系统，则可能需要选择基于 Amazon RDS 或 NoSQL 数据库（如 Amazon DynamoDB）构建的传统关系数据库系统。
- **非结构化数据** — Amazon Redshift 中的数据必须通过已定义的架构进行结构化。Amazon Redshift 不支持对每个行采用任意架构结构。如果您的数据是非结构化数据，则可在 Amazon EMR 上执行提取、转换和加载 (ETL) 过程来准备好数据，然后再将数据加载到 Amazon Redshift 中。对于 JSON 数据，您可以存储密钥值对，并在查询中使用[本机 JSON 函数](#)。<sup>14</sup>
- **BLOB 数据** — 如果您计划存储二进制大对象 (BLOB) 文件（例如，数字视频、图像或音乐），则可能需要考虑将数据存储存储在 Amazon S3 中，然后在 Amazon Redshift 中引用其位置。在此方案中，Amazon Redshift 会记录有关二进制对象的元数据（例如，项目名称、大小、创建日期、所有人、位置等），而大对象本身存储在 Amazon S3 中。

## 迁移到 Amazon Redshift

如果您决定从现有数据仓库迁移到 Amazon Redshift，则应根据以下几个因素来选择迁移策略：

- 数据库及其表的大小
- 源服务器与 AWS 之间的网络带宽
- 是通过一个步骤还是一系列步骤来完成到 AWS 的迁移和转换
- 源系统中的数据更改率
- 迁移期间的转换
- 您打算用于迁移和 ETL 的合作伙伴工具

## 一步式迁移

一步式迁移适用于不需要持续操作的小型数据库。客户可以使用逗号分隔的值 (CSV) 文件的形式提取现有数据库，然后使用诸如 AWS Import/Export Snowball 之类的服务将数据集传输到 Amazon S3，以便将数据加载到 Amazon Redshift 中。接下来，客户可以测试目标 Amazon Redshift 数据库是否与源的数据保持一致。在通过所有验证后，数据库将转换到 AWS。

## 两步式迁移

两步式迁移通常用于任意大小的数据库：

1. **迁移初始数据：**从源数据库中提取数据，最好是在非使用高峰期间进行，以最大程度地降低影响。接下来，使用上面的一步式迁移方法将数据迁移到 Amazon Redshift。
2. **迁移更改数据：**在切换之前，需要将迁移初始数据后在源数据库中发生更改的数据传播到目标数据库。此步骤对源数据库与目标数据库进行同步。在迁移所有更改的数据后，您可以验证目标数据库中的数据，执行必要的测试；如果通过了所有测试，则切换到 Amazon Redshift 数据仓库。

## 数据库迁移工具

有多种工具和技术可用于数据迁移。其中的一些工具可以互换使用，您也可以使用市场上提供的其他第三方工具或开源工具。

1. [AWS 数据库迁移服务](#)支持上面的一步式迁移过程和两步式迁移过程。<sup>15</sup>要采用两步式迁移过程，您需要启用补充日志记录以捕获对源系统所做的更改。可以在表级别或数据库级别启用补充日志记录。
2. 下面列出了其他一些数据集成合作伙伴工具：
  - Attunity
  - Informatica
  - SnapLogic
  - Talend
  - Bryte

有关数据集成和咨询合作伙伴的更多信息，请参阅 [Amazon Redshift 合作伙伴](#)。<sup>16</sup>

## 设计数据仓库工作流程

在前面几个部分中，我们介绍了 Amazon Redshift 的一些功能，这些功能使 Amazon Redshift 成为数据仓库的理想之选。为了说明如何使用 Amazon Redshift 设计数据仓库工作流程，下面我们来看一看最常见的设计模式与示例使用案例。

假设一家跨国服装制造企业拥有 1000 多家零售店，该公司在百货商店和折扣店中销售特定系列的服装，并且开设了网店来经营业务。从技术角度来说，这三种渠道目前都是单独经营的。它们具有不同的管理层、销售点系统和财务部。单靠一种系统无法将所有相关的数据集整合在一起让 CEO 全方位地了解整个企业。

此外，假设 CEO 希望从公司的全局出发了解这些渠道并能够执行临时分析，例如：

- 这些渠道中存在哪些趋势？
- 这些渠道在哪些地区的业绩更加出色？
- 公司的广告和促销活动的效果如何？
- 每个服务系列的趋势是什么？
- 哪些外在因素（例如，失业率和天气情况）影响了公司的销售业绩？
- 商店属性（例如，员工的任期和管理、零售商场和封闭式购物中心、店内商品的位置、促销、推荐商品、销售广告单和店内展示）对销售业绩有何影响？

企业数据仓库可以解决这些问题。它不仅收集来自所有这三个渠道的各种系统的数据，还收集来自公共数据源（如天气报告和经济报告）的数据。每个数据源每天都会发送数据以供数据仓库使用。由于每个数据源的结构可能不同，因此需要执行提取、转换和加载 (ETL) 过程，将数据格式重新设置为常用结构。然后，可以同时来自所有源的数据进行分析。为此，我们使用下面的数据流体系结构：

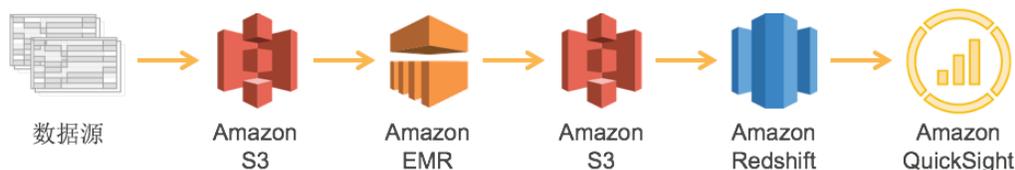


图 4：企业数据仓库工作流程

1. 在这个过程中，第一步是将来自各种源的数据迁移到 **Amazon S3** 中。**Amazon S3** 提供了一个高持久性、低成本且可扩展的存储平台，可以同时从许多不同的来源将数据写入到该平台，而且成本非常低。
2. **Amazon EMR** 用于将数据从源格式转换为目标格式并整理数据。**Amazon EMR** 已实现与 **Amazon S3** 的内置集成，这允许使用并行线程在 **Amazon EMR** 集群中的每个节点与 **Amazon S3** 之间来回传输数据。

通常，数据仓库在夜间获取新数据。由于午夜不需要执行分析，因此唯一要求是，此转换过程必须在 CEO 和其他业务用户上午需要访问报告和仪表板的时间之前完成。因此，您可以利用 [Amazon EC2 竞价市场](#) 来进一步降低 ETL 的成本。<sup>17</sup> 一个好的竞价策略是，从午夜开始以非常低的价格竞价，并随着时间的推移逐步提高价格，直到获得容量。在接近截止时间时，如果竞价未成功，您可以恢复按需使用价格，确保仍能满足完成时间要求。虽然每个源在 **Amazon EMR** 上的转换过程可能各不相同，但借助 AWS 即付即用模式，您可以为每个转换创建一个单独的 **Amazon EMR** 集群，并将其调整到适当的容量来完成所有数据转换作业，而不需要与其他作业争用资源。

3. 每个转换作业均会将经过整理的格式化数据加载到 **Amazon S3** 中。由于 **Amazon Redshift** 可以在每个集群节点上使用多个线程从 **Amazon S3** 并行加载数据，因此我们在此处再次使用 **Amazon S3**。**Amazon S3** 还提供历史记录并用作系统之间的可信的格式化数据来源。如果随时间的推移出现了其他要求，则可在其他工具中使用 **Amazon S3** 上的数据来执行分析。
4. **Amazon Redshift** 会在自己的表中对数据执行加载、排序、分布和压缩操作，以便分析查询能够高效地并行执行。当数据大小随时间的推移和业务的拓展而增大时，您可以通过添加更多节点来轻松地增加容量。
5. 要实现分析可视化，您可以使用 **Amazon QuickSight** 或许多合作伙伴可视化平台之一。这些平台使用 ODBC 或 JDBC 连接到 **Amazon Redshift**。CEO 及其员工可在此处查看报告、仪表板和图表。如今，高管可以利用数据，围绕公司资源制定更好的决策，这最终将为股东创造更高的收益和价值。

当您扩大业务、建立新的渠道、发布其他特定于客户的移动应用程序以及加入更多数据源时，您可以轻松地扩展这个灵活的体系结构。只需在 **Amazon Redshift** 管理控制台中单击几次，或者使用一些 API 调用，即可轻松完成。

## 总结

我们发现数据仓库正在发生战略性的转移，企业正在将其分析数据库和解决方案从本地解决方案迁移到云，从而充分利用云的简单性、高性能和高成本效益。本白皮书全面地介绍了 AWS 上的数据仓库的当前形势。AWS 提供了一系列广泛的服务和一个强大的合作伙伴生态系统，让您能够在云中轻松地构建和运行企业数据仓库。最终，您可以获得一个高性能、高成本效益的分析体系结构，这个体系结构能够随着您在 AWS 全球基础设施中的业务一起扩展。

## 贡献者

以下为对此文档有贡献的个人和组织：

- Babu Elumalai, Amazon Web Services 的解决方案架构师
- Greg Khairallah, Amazon Web Services 的首席 BDM
- Pavan Pothukuchi, Amazon Web Services 的首席产品经理
- Jim Gutenkauf, Amazon Web Services 的高级技术作者
- Melanie Henry, Amazon Web Services 的高级技术编辑
- Chander Matrubhutam, Amazon Web Services 的产品营销人员

## 阅读更多内容

有关更多帮助，请查阅以下资源：

- [Apache Hadoop 软件库](#)<sup>18</sup>
- [Amazon Redshift 最佳实践](#)<sup>19</sup>
- [Lambda 体系结构](#)<sup>20</sup>

## 备注

- 1 <https://www.forrester.com/report/The+Forrester+Wave+Enterprise+Data+Warehouse+Q4+2015/-/E-RES124041>
- 2 <http://aws.amazon.com/streaming-data/>
- 3 <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html>
- 4 <http://docs.aws.amazon.com/redshift/latest/APIReference/Welcome.html>
- 5 <http://docs.aws.amazon.com/redshift/latest/mgmt/working-with-clusters.html#rs-about-clusters-and-nodes>
- 6 [http://docs.aws.amazon.com/redshift/latest/dg/c\\_redshift-and-postgresql.html](http://docs.aws.amazon.com/redshift/latest/dg/c_redshift-and-postgresql.html)
- 7 <http://aws.amazon.com/redshift/partners/>
- 8 <https://aws.amazon.com/vpc/>
- 9 <https://aws.amazon.com/cloudtrail/>
- 10 <http://docs.aws.amazon.com/redshift/latest/mgmt/working-with-HSM.html>
- 11 <https://aws.amazon.com/kms/>
- 12 <http://aws.amazon.com/s3/pricing/>
- 13 <http://aws.amazon.com/redshift/pricing/>
- 14 <http://docs.aws.amazon.com/redshift/latest/dg/json-functions.html>
- 15 <https://aws.amazon.com/dms/>
- 16 <https://aws.amazon.com/redshift/partners/>
- 17 <http://aws.amazon.com/ec2/spot/>
- 18 <https://hadoop.apache.org/>
- 19 <http://docs.aws.amazon.com/redshift/latest/dg/best-practices.html>
- 20 [https://en.wikipedia.org/wiki/Lambda\\_architecture](https://en.wikipedia.org/wiki/Lambda_architecture)