

云享家 | AWS云数据仓库Redshift让数据飞起来

原创：杜柳松 BespinGlobal 1周前



云数据仓库让企业梦想成长，数据仓库通过与相关服务的更多抽象和集成来继续简化数据挖掘，这在某些情况下，无需调出实例。最终目标是针对各种规模的公司，效仿那些强调自动化和从数据收集中挖掘出更多信息的网络规模企业的成功。

Bespin Global China资深云架构师杜柳松将为您**详细解析AWS Redshift的结果缓存机制**，亦将**手把手的带诸位一起完成实操**。



本文涵盖内容：

- 一键部署Redshift集群
- 导入测试样本数据
- 对比不同设置下的查询性能
- 释放测试环境资源

本文不涵盖内容：

- SQL客户端软件的选择与使用
- SQL、JSON和CloudFormation模板的语法
- AWS CLI命令
- AWS的常用术语、VPC、IAM、EC2、S3、CloudFormation的基本知识

由于篇幅所限，大量文档引用、表格、SQL语句和CloudFormation的大段模板，点击“阅读”中。

ut further ado, let' s go!

Redshift简介

Amazon Redshift (后简称Redshift) 数据仓库是一个企业级的、可以用于关系数据库查询的、托管的AWS数据仓库产品。Redshift通过大规模并行处理、列式数据存储和非常高效且具有针对性的数据压缩编码方案的组合, 实现高效存储和最优查询性能。

Redshift支持与多种类型的应用程序(包括商业智能(BI)、报告、数据和分析工具)建立客户端连接。本文使用SQL客户端软件连接Redshift。

Redshift性能优化概述

数据仓库性能优化是一个及其宽泛的命题, Redshift也不例外。优化的思路包括但不限于:**合理选键(sort key和distribution key)、Result Caching、数据压缩、数据加载时优化、优化查询语句等。**

本文此次仅针对Result Caching一项进行介绍。如有兴趣, 读者可以自行参阅Redshift最佳实践等文档, 以获取其他方面的信息。

Redshift性能优化之Result Caching

为了缩短查询执行时间并改进系统性能, Redshift 在leader node的内存中缓存特定查询类型的结果。当用户提交查询时, Redshift 会在结果缓存中检查是否有查询结果的有效缓存副本。如果在结果缓存中找到匹配项, Redshift会使用缓存的结果而不执行查询。

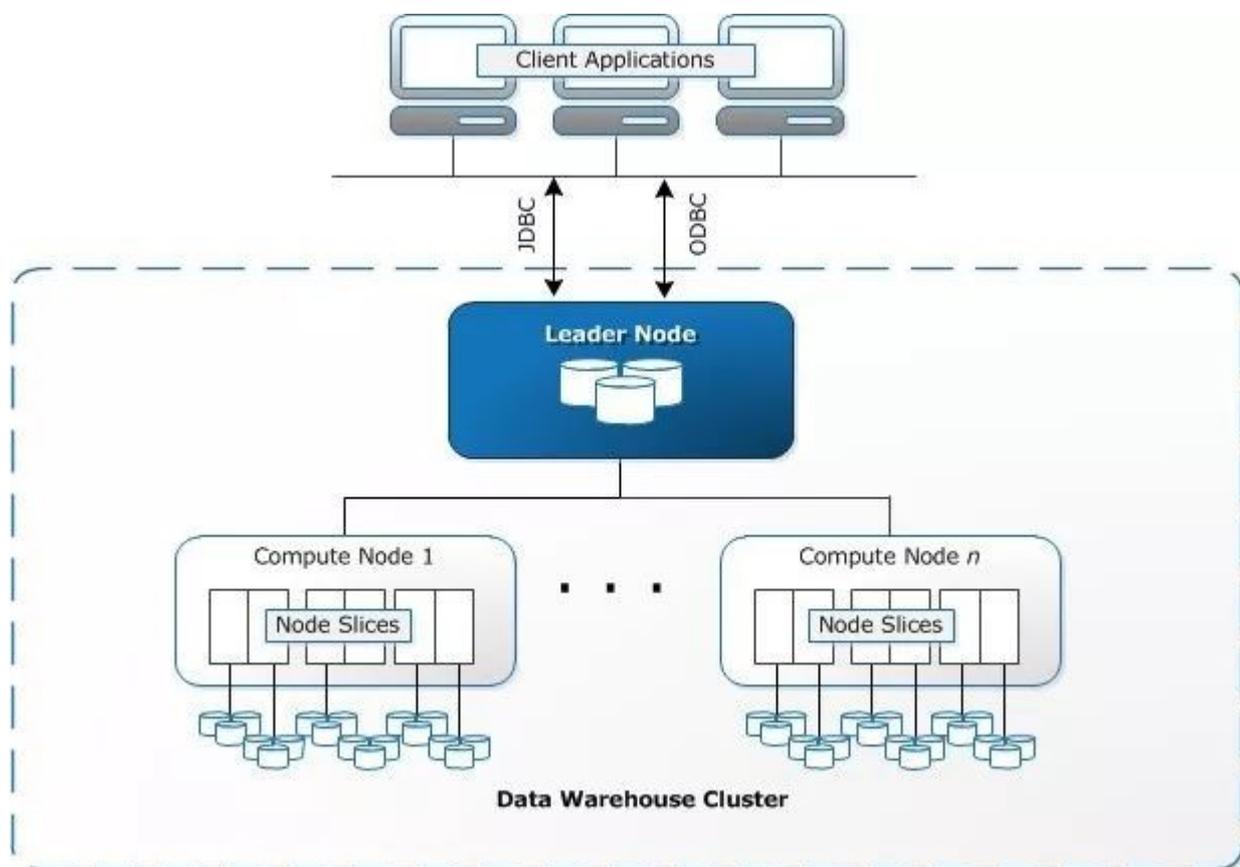


图1 Redshift内部架构示意图

默认情况下，结果缓存处于启用状态。要为当前会话禁用结果缓存，需要明确地将 `enable_result_cache_for_session` 参数设置为 `off`（后文会介绍相关命令）。

实验准备

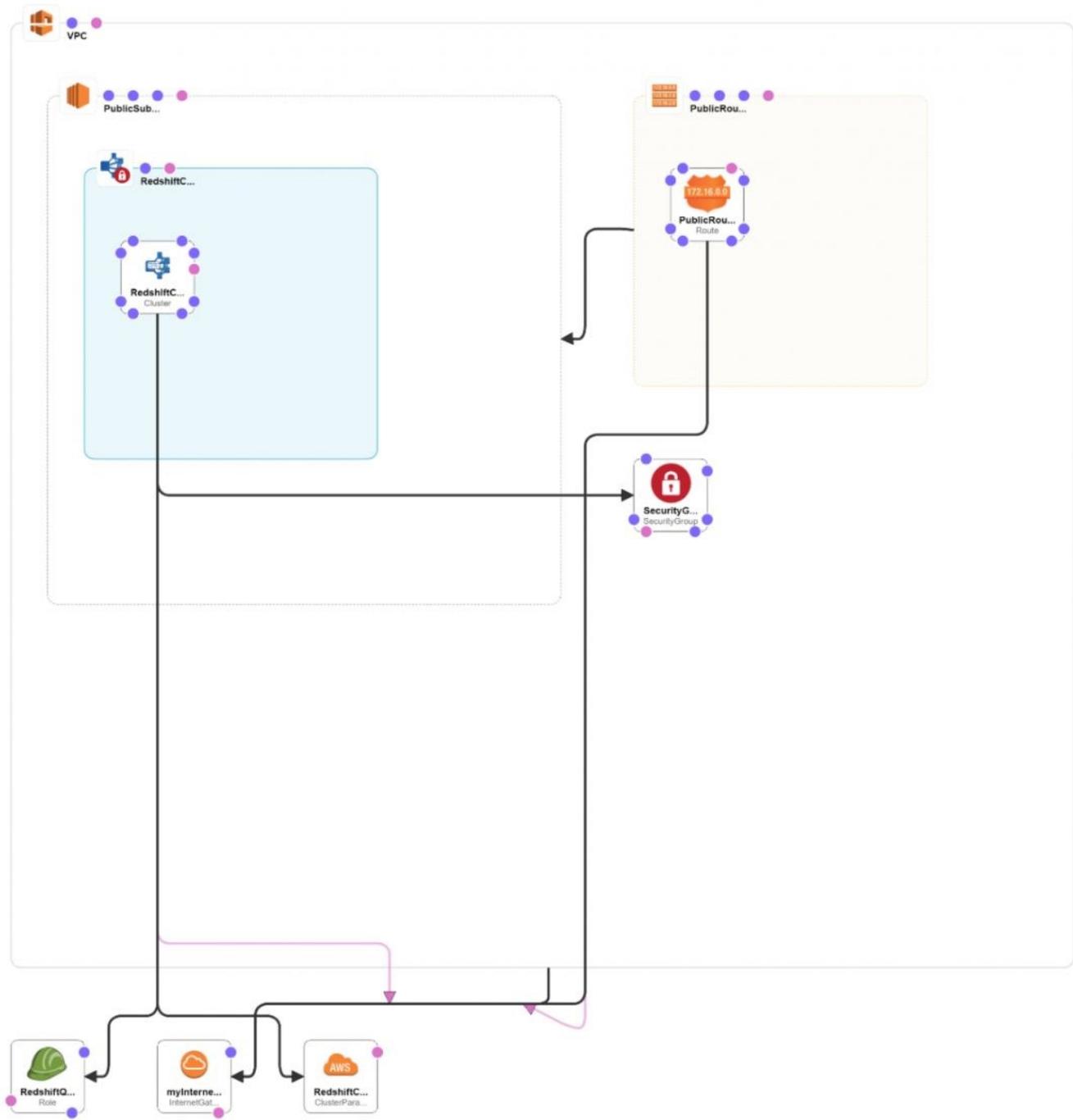
如需按照本文完成实验，需准备好以下条件：

- AWS global账号。
- AWS China账号（可选）。如严格按照本文一步步进行操作（直接COPY代码），需要AWS China账号。如使用AWS global进行实验，则可能需要对代码自行进行微调。
- 可用的SQL连接客户端，需要支持JDBC连接。

Infrastructure环境部署

使用本文文末“[阅读原文](#)”中所附的CloudFormation代码一键部署本次实验环境。本次实验使用的是AWS宁夏region。

实验环境如下图：



使用文末链接中的CloudFormation模板生成资源，并参考下图进行最终配置。

Specify Details

Specify a stack name and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. [Learn more.](#)

Stack name

Parameters

MasterUserName

MasterUserPassword

The password that is associated with the master user account for the cluster that is being created.

RedshiftAz

The AZ of the Redshift. Be careful to choose, as the current version does not recognize which AZ supports Redshift service. Some AZs do not support Redshift!

RedshiftDb

The DB name of the Redshift. DbName parameter must be lowercase, begin with a letter, contain only alphanumeric characters, underscore ("_"), plus sign ("+"), dot ("."), at ("@"), or hyphen ("-"), and be less than 64 characters.

RedshiftRoleName

The IAM role name that will be attached to the Redshift.

其中：

- Redshift的Master用户名需要记录下来，后面需要用到。此处使用“master”。
- Redshift的可用区需额外注意。例如宁夏有3个AZ，cn-northwest-1a，cn-northwest-1b，cn-northwest-1c，但其中只有cn-northwest-1a和cn-northwest-1b支持Redshift服务。此处使用“cn-northwest-1a”。
- Redshift的IAM role名称需要记录下来，后面需要用到。使用其替换本文中的<role-name>。此处使用“redshiftquerycacheRole”。
- 此外，CloudFormation模板中已指定Redshift cluster包含两个节点，节点类型为dc1.large。

大约7分钟后，整个环境的资源完全生成。在Outputs tab处获取Redshift database连接信息。记录下来，后面会用到。

Key	Value	Description	Export Name
RedshiftClusterInfo	redshift-querycache- cn-northwest-1.redshift.amazonaws.com.cn	The connection endpoint for the Amazon Redshift cluster.	

实验源数据准备

STEP 1: 数据复制

本次实验所用到的数据借用AWS提供的用于Redshift测试的样本数据。

为了保证后面的步骤中，数据可以顺利load到Redshift，此处先将文件复制到本地环境。

由AWS提供的原始数据位于AWS global的名为“awssampledbuswest2”的S3 bucket中，“ssbgz” prefix下。所有有效的AWS global的用户，都可以使用具有访问S3权限的AWS access key和AWS secret access key（下文简称AK和SK）得以访问上述位置中的资源。

为了执行数据拷贝，我们需要首先启动一台AWS本地（此处为宁夏region）的EC2。SSH到其上后，运行以下命令，将源数据拷贝到本地EBS卷（需要具有相应AWS global的S3的权限才能成功访问，可通过“aws configure”命令设置AK、SK、region）。

```
aws s3 cp s3://awssampledbuswest2/ssbgz/ . --recursive
```

完成上述操作后，运行以下命令将本地数据拷贝到国内S3 bucket（需要具有相应AWS China的S3的权限才能成功访问，可通过“aws configure”命令设置AK、SK、region）。记录S3 bucket的名字，后面需要用到（如果需要，新建一个S3 bucket）。使用读者自己的环境中的bucket名称替换本文中的<aws-china-s3-bucket>。

```
aws s3 cp . s3://<aws-china-s3-bucket>/ --recursive
```

STEP2: 导入数据

以下步骤均配有SQL语句，由于篇幅所限，请详见文末“[阅读原文](#)”。

1. 查看当前的table(s)，防止有同名的table(s)存在干扰实验的正常进行。返回结果显示当前'public'的schema下没有table
2. 创建本次实验所需的tables。
3. 查看上一步骤tables的创建结果。返回结果说明本次实验所需的tables已经创建成功。
4. 将数据分别load到对应table表中。
 - 使用之前保存测试数据文件的AWS S3 bucket名称替换下面的<aws-china-s3-bucket>字段。
 - 使用读者的AWS account ID替换下面的<aws-account-id>字段。
 - 使用之前创建并attach到Redshift的IAM role名字替换下面的 <role-name> 字段。

实验环境准备

创建测试用户

因为后面测试会验证结果缓存对来自不同用户的查询的影响，所以此处需创建database user accounts用于后面的测试。SQL语句详见文末“[阅读原文](#)”。

用户授权

用户创建完成后，还需要给用户赋予足够的权限以使其能够成功执行SQL查询语句。SQL语句详见文末“[阅读原文](#)”。

测试计划

设计SQL查询命令

设计SQL查询命令。后面的部分将基于不同的条件执行如下所示的相同的SQL查询命令。以

对比不同条件下，执行查询所花费的时间。SQL语句详见文末“阅读原文”。

设计Benchmarks表格

设计benchmarks表格，以直观展现各种情况下缓存的效果，并对如何有效利用结果缓存优化数据仓库查询获得深入的理解。

在接下来的测试中，我们会在这个表格中记录每次执行查询所花费的时间。

	第1次运行	第2次运行	第3次运行	第4次运行	第5次运行
是否开启结果缓存	N	N	N	Y	Y
以哪个database user account运行SQL查询	master	master	master	master	leo
1 st query					
2 nd query					
3 rd query					
Total elapsed time					

性能测试

- 为当前会话禁用结果缓存

执行以下SQL命令，对当前session明确地禁用result caching。SQL语句详见文末“阅读原文”。

- 第1次SQL查询

执行第1次SQL查询，查询语句为上面列出的3个语句。

根据返回结果填写benchmarks表格中的第2列。由于篇幅所限，填表结果详见文末“阅读原文”。

- 第2次SQL查询

再次运行上述的3个SQL查询语句，以除去在leader节点上compile SQL语句所耗费的时间。根据返回结果填写benchmarks表格中的第3列。由于篇幅所限，填表结果详见文末“阅读原文”。可见，在leader节点上执行compile作业也会耗费一定的时间。

- 为当前会话启用结果缓存

执行以下SQL命令，对当前session明确地启用result caching。SQL语句详见文末“阅读原文”。

- 第3次SQL查询

再次运行3个SQL查询语句，根据返回结果填写benchmarks表格中的第4列。由于篇幅所限，填表结果详见文末“阅读原文”。

- 第4次SQL查询

再次运行3个SQL查询语句，根据返回结果填写benchmarks表格中的第5列。由于篇幅所限，填表结果详见文末“阅读原文”。

- 第5次SQL查询

以leo用户执行同样的3个SQL查询语句，根据返回结果填写benchmarks表格中的第6列。

	第1次运行	第2次运行	第3次运行	第4次运行	第5次运行
是否开启结果缓存	N	N	N	Y	Y
以哪个 database user account运行SQL查询	master	master	master	master	leo
1 st query	20.75s	14.59s	15.89s	0.05s	0.05s
2 nd query	30.69s	16.94s	16.27s	0.08s	0.12s
3 rd query	23.9s	13.38s	13.42s	0.08s	0.09s
Total elapsed time	1m 15s	44.9s	45.58s	0.21s	0.26s

查证结果缓存的使用情况

要确定查询是否使用了结果缓存，需要查询svl_qlog系统视图。

- 如果查询使用了结果缓存，source_query列会返回源查询的查询ID。
- 如果未使用结果缓存，则source_query列值为NULL。

注意：SQL查询需要以master用户执行。**SQL语句详见文末“阅读原文”。**

以下返回结果说明了：用户master和leo提交的查询使用了来自master之前运行的查询的结果缓存（查询ID为566, 574和576）。

usename	query	elapsed	source_query	substring
leo	620	46	576	-- Query 3 -- Drill down in time to just one month select c_
leo	619	357	574	-- Query 2 -- Restrictions on two dimensions select sum(lo_r
leo	618	69	566	-- Query 1 -- Restrictions on only one dimension. select sum
master	588	54	576	-- Query 3 -- Drill down in time to just one month select c_

master	587	363	574	-- Query 2 -- Restrictions on two dimensions select sum(lo_r
master	586	37	566	-- Query 1 -- Restrictions on only one dimension. select sum
master	576	13312008		-- Query 3 -- Drill down in time to just one month select c_
master	574	16168575		-- Query 2 -- Restrictions on two dimensions select sum(lo_r
master	566	15841640		-- Query 1 -- Restrictions on only one dimension. select sum
master	539	13244945		-- Query 3 -- Drill down in time to just one month select c_
master	536	16839301		-- Query 2 -- Restrictions on two dimensions select sum(lo_r
master	531	14533302		-- Query 1 -- Restrictions on only one dimension. select sum
master	497	23811613		-- Query 3 -- Drill down in time to just one month select c_
master	494	30601234		-- Query 2 -- Restrictions on two dimensions select sum(lo_r
master	489	20695577		-- Query 1 -- Restrictions on only one dimension. select sum

测试收尾

- 撤销用户权限

本步骤意图撤销为了完成本次实验所创建的Redshift数据库用户的访问权限。在删除用户前必需经由此步骤。生产环境下建议将权限转移给其他用户。SQL语句详见文末“阅读原文”。

- 删除用户

删除本次实验创建的Redshift数据库用户。SQL语句详见文末“阅读原文”。

- 查看用户

查看当前剩余用户（除rdsdb和master外），以确认上一步执行结果。SQL语句详见文末“阅读原文”。返回结果显示，当前没有除rdsdb和master以外的database user account。

- 删除CloudFormation Stack

删除本次实验所创建的CloudFormation Stack，以释放AWS Redshift资源。

- 删除S3数据

(可选) 删除S3 bucket中的原始测试数据，以及S3 bucket。

最后，欢迎持续关注我们的公众号，以及时获得我们发布的有关大数据、AI、Serverless、DevOps等方向的最新信息。

附录

CloudFormation模板详见文末“阅读原文”。

END

相 关 阅 读

云享家 | 没有绝对的安全，只有在阿里云的最佳实践。

云享家 | 医疗和金融行业在AWS平台的合规配置

云享家 | 基于AWS云平台的IDS联合监控分析

云享家 | 多面手完善AWS CloudWatch监控图表

点击下方
阅读原文



阅读原文