

分析型数据库AnalyticDB

应用开发及示例程序

访问 AnalyticDB

负载均衡与读写分离

VPC网络

账号与权限

1. AnalyticDB兼容MySQL协议，支持JDBC, ODBC, 和RESTful接口；
2. AnalyticDB支持MySQL自带的客户端连接；
 - 通过mysql命令行连接AnalyticDB进行查询时，为了识别查询中的HINT，命令行参数需要指定 -c 参数：
`mysql -h<host> -P<port> -u<user_name> -p<password> -D<db_name> -c -A`
3. AnalyticDB支持通过JDBC连接，支持大部分版本的mysql-jdbc驱动；
4. AnalyticDB支持通过ODBC连接，通过MySQL官网ODBC驱动；
5. AnalyticDB支持RESTful接口访问，RESTful API由FN Proxy Service支持；

最简单直接的连接并访问AnalyticDB的方式是通过JDBC，AnalyticDB支持MySQL自带的客户端以及大部分版本的mysql-jdbc驱动。

支持的MySQL JDBC驱动版本号

- 5.0系列：5.0.2, 5.0.3, 5.0.4, 5.0.5, 5.0.7, 5.0.8
- 5.1系列：5.1.1, 5.1.2, 5.1.3, 5.1.4, 5.1.5, 5.1.6, 5.1.7, 5.1.8, 5.1.11, 5.1.12, 5.1.13, 5.1.14, 5.1.15, 5.1.16, 5.1.17, 5.1.18, 5.1.19, 5.1.20, 5.1.21, 5.1.22, 5.1.23, 5.1.24, 5.1.25, 5.1.26, 5.1.27, 5.1.28, 5.1.29, 5.1.31, 5.1.32, 5.1.33, **5.1.34（推荐）**, 5.1.35, 5.1.36, 5.1.37

MySQL命令行参数

- 通过mysql命令行连接AnalyticDB进行查询时，为了识别查询中的HINT，命令行参数需要指定 **-c** 参数：
`mysql -h<host> -P<port> -u<user_name> -p<password> -D<db_name> -c -A`

JDBC连接代码片段

不带重试的JDBC样例程序片段

带重试的JDBC样例程序片段

注意：
useUnicode=true
&characterEncoding=UTF-8

```
try {  
    Class.forName("com.mysql.jdbc.Driver");  
    String url = "jdbc:mysql://[idbname].[ads].[aliyuncs.com]:5544/my_ads_db?useUnicode=true&  
characterEncoding=UTF-8";  
  
    Properties connectionProps = new Properties();  
    connectionProps.put("user", "my_access_key_id");  
    connectionProps.put("password", "my_access_key_secret");  
  
    connection = DriverManager.getConnection(url, connectionProps);  
    statement = connection.createStatement();  
  
    String query = "select count(*) from information_schema.tables";  
    rs = statement.executeQuery(query);  
  
    while (rs.next()) {  
        System.out.println(rs.getObject(1));  
    }  
}
```

代码片段参考链接：[带重试/不带不重试的JDBC样例程序片段](https://help.aliyun.com/document_detail/26423.html?spm=a2c4g.11186623.6.624.zZyMym)

https://help.aliyun.com/document_detail/26423.html?spm=a2c4g.11186623.6.624.zZyMym

当使用JDBC连接池连接AnalyticDB时，推荐使用Druid连接池

Druid连接池配置

- maxActive: 100 --- 根据最大值和业务并发量来定，建议该值大于等于业务并发数
- initialSize: 5
- maxWait: 60000
- minIdle: 10
- maxIdle: 20
- timeBetweenEvictionRunsMillis: 2000
- minEvictableIdleTimeMillis: 600000
- maxEvictableIdleTimeMillis: 900000
- validationQuery: `show status like '%Service_Status%'`; 推荐的AnalyticDB服务状态检测语句
- testWhileIdle: true
- testOnBorrow: false
- testOnReturn: false
- removeAbandoned: true
- removeAbandonedTimeout: 180

JDBC 连接池使用示例

```
import com.alibaba.druid.pool.*;
DruidDataSource dataSource = new DruidDataSource();
dataSource.setDriverClassName("com.mysql.jdbc.Driver");
dataSource.setUsername("my_access_key_id");
dataSource.setPassword("my_access_key_secret");
dataSource.setUrl("jdbc:mysql://mydbname-xxxx.ads-hz.aliyuncs.com:5544/my_ads_db");
// 连接数配置
dataSource.setInitialSize(5);
dataSource.setMinIdle(1);
dataSource.setMaxActive(10);
// 启用监控统计功能
dataSource.setFilters("stat");
// for mysql
dataSource.setPoolPreparedStatements(false);
// 使用心跳语句检测空闲连接
dataSource.setValidationQuery("show status like '%Service_Status%'");
dataSource.setTestWhileIdle(true);
```

通过MySQLdb模块来访问AnalyticDB

[MySQLdb模块介绍](#)

<https://pypi.python.org/pypi/MySQL-python/1.2.5>

Python访问ADS示例代码

```
import MySQLdb
```

```
# user为目标AnalyticDB数据库实例的用户账号（通常为阿里云AK的access id）
```

```
# passwd为目标AnalyticDB数据库实例的用户密码（通常为阿里云AK的access key）
```

```
db = MySQLdb.connect(host='cn-xxx.ads.aliyun.com', port=9999, user='user', passwd='123456', db='testdb')
```

```
cursor = db.cursor() # 使用cursor()方法获取操作游标
```

```
cursor.execute("SELECT VERSION()") # 使用execute方法执行SQL语句
```

```
data = cursor.fetchone() # 使用 fetchone() 方法获取一条数据
```

```
db.close() #关闭连接
```


在PHP环境中，通过php-mysql 5.1.x模块（win,php_MySQL.dll）来访问AnalyticDB

PHP访问ADS示例代码

```
//数据库的连接url，请在控制台中的连接信息中获取
//数据库的端口号，请在控制台中的连接信息中获取
$ads_server_name="mydbname-xxxx.ads-cn-hangzhou-1.aliyuncs.com ";
$ads_username="my_access_key_id"; // 连接数据库用户名
$ads_password="my_access_key_secret"; // 连接数据库密码
$ads_database="my_ads_db"; // 数据库的名字
$ads_port=3003;
$ads_conn=mysqli_connect($ads_server_name, $ads_username, $ads_password, $ads_database, $ads_port);

$strsql="SELECT user_id FROM my_ads_db.my_table limit 20;";
$result=mysqli_query($ads_conn, $strsql);
while($row = mysqli_fetch_array($result)) {
echo $row["user_id"] ; //user_id为列名
}
```

重试与异常处理

- 网络环境下使用AnalyticDB，为了保证业务高可用性，建议采用适当的重试和异常处理机制
- 无论采用JDBC直连、JDBC连接池、PHP/Python还是ODBC等方式连接AnalyticDB，应采用适当的重试机制来保证业务的连接可用性
- 在海量数据并发处理任务量大的情况下，在查询过程中偶尔出现异常时，建议采用适当的重试机制来重发查询
- 代码片段参考链接：[带重试/不带重试的JDBC样例程序片段](#)

应用连接池建议

- Java应用推荐使用DruidDataSource，但是不建议开启sql过滤等辅助功能
- 整个数据库的连接数（读）参考： $\text{min} = \text{Frontnode的数量} * 5$, $\text{max} = \text{FRONTNODE的数量} * 30$
- 心跳包：必须，建议20-40s一次，建议使用show status like '%Service_status%' 语句。
注意：无心跳包，2min无查询自动断开连接，会出现如下访问错误：
Communication links failure ... The last packet successfully received from the server was xxx milliseconds ago ...
- 注意不要出现多个线程同时使用一个连接的情况，即不建议多线程之间共享使用Connection对象

访问 AnalyticDB

负载均衡与读写分离

VPC网络

账号与权限

关于负载均衡SLB介绍 <http://www.aliyun.com/product/slb>

- 创建DB时，根据指定的服务参数（ecuCount），在AnalyticDB集群中分配特定数量的前端服务机（FrontNode），这些节点对用户透明，用户得到的是访问该数据库的URL：
如：mydbname-xxxx.ads-hz.aliyuncs.com:5544，可通过控制台获取
- AnalyticDB使用阿里云负载均衡产品SLB来对访问请求进行负载均衡，用户无需关心访问AnalyticDB时的负载均衡策略
- 每个DB共享SLB实例对应的VIP，通过不同的端口实现DB的负载均衡。
注：v2.3以上版本支持开关控制DB独占SLB实例

关于读写分离

- AnalyticDB支持DB的读写分离来解决因读写流量较大导致相互影响的问题
- 读写分离原理：一部分FrontNode节点挂到读域名和端口；另外一部分FrontNode节点挂到写域名和端口
- 写的FrontNode禁用了计算节点数据版本汇报管理和其他系统服务，故写的域名不可进行数据查询

访问 AnalyticDB

负载均衡与读写分离

VPC网络

账号与权限

VPC功能介绍

- AnalyticDB从2.3版本开始支持VPC功能，默认使用single tunnel方式，也可以配置切换到any tunnel
- 公共云端暂时没有针对VPC特性进行改造，若想要使用VPC功能，可提工单申请单独开启VPC
- 专有云V3版本中已经开启VPC，也可通过DTCenter进行VPC改造；
注：目前暂不支持反向访问

Single tunnel

- AnalyticDB默认的VPC方式，从用户VPC网段内获取一个IP用作当前库的访问VIP，在用户VPC网段内访问AnalyticDB
- 该模式可以实现不同VPC内实现网络隔离的目标，同时支持无缝切换VPC网络的命令

Any tunnel

- 通过配置切换，修改配置后下一次创建库时生效或者通过修改元数据及重启FrontNode生效
- 返回的VIP在所有VPC内网络都通畅，理论上满足不了不同VPC之间网络隔离的需求

访问 AnalyticDB

负载均衡与读写分离

VPC网络

账号与权限

Access Key ID/Secert

AK ID/Secert介绍

- Access Key ID 和 Access Key Secret 是访问AnalyticDB（包括阿里云API）的唯一凭证
- Access Key ID 是类似身份的标识，而 Access Key Secret 的作用是签名您的访问参数，以防被篡改。
注意：Access Key Secret 类似于登录密码，不要向任何人泄漏

获取AK ID/Secert

创建AK ID/Secert

管理控制台

产品与服务

搜索

费用

工单

备案

企业

支持

简体中文

Access Key管理 (2)

刷新

创建Access Key

⚠ Access Key ID和Access Key Secret是您访问阿里云API的密钥，具有该账户完全的权限，请您妥善保管。

Access Key ID	Access Key Secret	状态	创建时间	操作
LT...pNn...9FWv...3U	显示	启用	2017-10-10 14:39:55	禁用 删除
LT...OWN...4FW...xST	显示	启用	2017-08-21 18:22:44	禁用 删除

账号类型

- AnalyticDB用户是基于阿里云帐号及其子帐号进行认证、权限管理等操作
- 可通过访问控制服务(RAM)在云帐号下创建子帐号并对子帐号添加授权策略，用该子帐号访问AnalyticDB
- **注意：**无论是云帐号还是其子帐号，AnalyticDB都需要使用帐号对应的Access Key ID和Access Key Secret来访问

权限管理

- AnalyticDB支持基于数据库表的层级权限管理模型，提供类似MySQL的ACL授权模式
- 一个ACL授权由被授权的用户、授权对象和授予的对象权限组成。
- 和MySQL略有不同的是，分析型数据库目前不支持针对用户在host上授权

权限对象

- Database(库), 即 db_name.* 或 * （默认数据库）, 指定数据库或数据库上所有表/表组；
- TableGroup(表组), 即 db_name.table_group_name 或 table_group_name , 指特定表组；
- Table(表), 即 db_name.table_name 或 table_name , 指特定表；

权限类型	库级别	表组级别	表级别	说明
SELECT	有	有	有	查询数据
LOAD DATA	有	有	有	导入表（分区）数据
DUMP DATA	有	有	有	导出表（分区）数据
DESCRIBE	有	有	有	查看数据库、表/表组信息
SHOW	有	有	有	列出数据库、表/表组内部对象
ALTER	有	有	有	修改表/表组定义
DROP	有	有	有	删除数据库、表/表组或分区
CREATE	有	有	有	创建表/表组
INSERT	有	有	有	执行Insert的权限
DELETE	有	有	有	执行Delete的权限
ALL	有	有	有	以上所有权限



Thanks!

咨询邮箱：ADB_SUPPORT@service.alibaba.com