

exBase

安装说明文档

(Version 3.11)

改版记录

版本	发布日期	描述	作者	复审	批准
v1.0	14/9/2022	新建用户操作说明	张泽涛		

变更记录

变更 编号	日期	变更项	描述	基线版本	变更请求编号

【版权声明】

©2007-2021 北京海量数据技术股份有限公司 版权所有

本文档著作权归 **北京海量数据技术股份有限公司**（简称“海量数据”）所有，未经海量数据事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

北京海量数据技术股份有限公司保留所有的权利。

【商标声明】



及其它海量数据产品和服务相关的商标均为 **北京海量数据技术股份有限公司** 及其关联公司所有。

本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍海量数据全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的产品、服务的种类、服务标准等应由您与海量数据之间的商业合同约定，除非双方另有约定，否则，海量数据对本文档内容不做任何明示或模式的承诺或保证。

目录

1. 安装环境要求.....	4
2. 安装过程.....	4
3. 插件安装.....	7
3.1. 前提条件.....	7
3.2. 配置 decoderbufs.....	9
3.3. 集群配置.....	11
4. exBase 的启动、停止.....	12
5. 补丁包安装说明.....	13
6. 使用前的补充操作建议.....	14

1. 安装环境要求

- ❖ 操作系统: CentOS/Redhat 7.6 及以上
- ❖ CPU: 8 核
- ❖ 内存: 16GB 以上
- ❖ 磁盘空间: 磁盘空间用于安装程序和数据校验, 建议不少于 500GB。如果迁移数据较大, 建议预留迁移数据量 2.5 倍以上的磁盘空间。
- ❖ 文件系统: xfs
- ❖ 系统安装时不需要进行磁盘划分
- ❖ 挂载 yum 源

2. 安装过程

(1) 安装依赖包;

```
yum install -y libicu
yum install -y libicu-devel
yum install -y openssl-devel
yum install -y gcc
yum install -y unzip
yum install -y perl-ExtUtils-Embed
yum install -y libxslt
yum install -y tcl
yum install -y java-1.8.0-openjdk
```

注意: 依赖包一定要安装完成, 如漏装依赖包 openssl-devel 会导致 exbase 启动失败。

(2) exBase 安装包大小约 400MB, 将安装包上传, 并解压到根目录。此处以 exbase-3.11 安装包为例;

```
unzip exbase-3.11_centos-7.x86_64.zip -d /exbase_environment
```

(3) 上传并安装;

若指定安装目录为**根目录**, 需检查目录/deployment 和/data 是否存在, 若存在则删除, 这个目录中若有文件, 请自行备份; 若指定安装目录为非根目录, 只需检查/deployment 即可。

```
chmod +x /exbase_environment/exbase_install.sh
cd /exbase_environment/
./exbase_install.sh
```

运行安装脚本后，用户需指定安装目录的绝对路径，如下图所示：

```
-----  
当前为root用户  
-----  
自定义exBase安装路径: /exbase  
exBase will be installed to '/exbase'  
创建安装目录
```

指定路径后会进行依赖检查和端口检测：

```
-----依赖检查-----  
gcc           :已安装 gcc-7.3.0-20210605.41.oe1.aarch64  
libc_u        :已安装 libc_u-62.1-5.oe1.aarch64  
libc_u-devel  :已安装 libc_u-devel-62.1-5.oe1.aarch64  
openssl-devel :已安装 openssl-devel-1.1.1f-19.oe1.aarch64  
unzip         :已安装 unzip-6.0-46.oe1.aarch64  
perl          :已安装 perl-5.28.3-6.oe1.aarch64  
libxslt       :已安装 libxslt-1.1.34-2.oe1.aarch64  
tcl           :已安装 tcl-8.6.10-4.oe1.aarch64  
java-1.8.0-openjdk :已安装 java-1.8.0-openjdk-1.8.0.342.b07-0.oe1.aarch64  
-----端口检查-----  
31030:端口可用  
31003:端口可用  
31004:端口可用  
2181 :端口可用  
9092 :端口被占用  
重新设置端口(y/n)█
```

若是默认端口被占用，则需指定相应的端口：

```
9092 :端口被占用  
重新设置端口(y/n)y  
kafkaPort端口: 31005
```

指定安装的用户，可以使用现有的非超级管理员用户(非 root)，也可以创建用户和所属用户组：

```
-----  
请输入安装用户名(非root用户): appusr  
安装的用户: appusr,所属分组: appusr  
-----
```

然后等待安装脚本运行结束，安装日志位于/exbase_environment/exBase_install.log

安装日志上报错： ERROR StatusLogger No log4j2 configuration file found. Using default configuration: logging only errors to the console 无需理会；成功安装部分日志如下：

```
[root@localhost exbase_environment]# cat exBase_install.log
create install directory
install AtlasDB start
init AtlasDB ...
The files belonging to this database system will be owned by user "appusr".
This user must also own the server process.

The database cluster will be initialized with locale "en_US.UTF-8".
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".

Data page checksums are disabled.

fixing permissions on existing directory /exbase/data/confdb ... ok
creating subdirectories ... ok
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting default timezone ... Asia/Shanghai
selecting dynamic shared memory implementation ... posix
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

WARNING: enabling "trust" authentication for local connections
You can change this by editing pg_hba.conf or using the option -A, or
--auth-local and --auth-host, the next time you run initdb.

Success. You can now start the database server using:

    /exbase/deployment/atlasdb/bin/pg_ctl -D /exbase/data/confdb/ -l logfile start

set confdb parameters
enable and start atlasdb.service
Created symlink from /etc/systemd/system/multi-user.target.wants/atlasdb.service to /etc/systemd/system/atlasdb.service.
● atlasdb.service - AtlasDB server
   Loaded: loaded (/etc/systemd/system/atlasdb.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2021-08-04 17:39:54 CST; 1s ago
 Process: 31573 ExecStart=/exbase/deployment/atlasdb/bin/pg_ctl -D ${ADATA} start (code=exited, status=0/SUCCESS)
 Main PID: 31575 (atlasdb)
   Tasks: 8
  CGroup: /system.slice/atlasdb.service
          └─31575 /exbase/deployment/atlasdb/bin/atlasdb -D /exbase/data/confdb
            └─31576 atlasdb: logger
              └─31578 atlasdb: checkpointer
                └─31579 atlasdb: background writer
                  └─31580 atlasdb: walwriter
                    └─31581 atlasdb: autovacuum launcher
                      └─31582 atlasdb: stats collector
                        └─31583 atlasdb: logical replication launcher
```

```
exbase-1.jar kafka-kafka /exbase/kafka/config/server.properties
Aug 04 17:40:06 localhost.localdomain systemd[1]: Started kafka.
install incremental start
set incremental.service
Created symlink from /etc/systemd/system/multi-user.target.wants/incremental.service to /etc/systemd/system/incremental.service.
unzip oracle client
Archive: instantclient-basic-linux.x64-12.2.0.1.0.zip
  inflating: instantclient_12_2/adrci
  inflating: instantclient_12_2/BASIC_README
  inflating: instantclient_12_2/genezi
  inflating: instantclient_12_2/libclntshcore.so.12.1
  inflating: instantclient_12_2/libclntsh.so.12.1
  inflating: instantclient_12_2/liboci1.so
  inflating: instantclient_12_2/libmq11.so
  inflating: instantclient_12_2/libnm12.so
  inflating: instantclient_12_2/libocci.so.12.1
  inflating: instantclient_12_2/liboci11.so
  inflating: instantclient_12_2/libocidbc12.so
  inflating: instantclient_12_2/libons.so
  inflating: instantclient_12_2/liboracm12.so
  inflating: instantclient_12_2/ojdbc8.jar
  inflating: instantclient_12_2/udrvci
  inflating: instantclient_12_2/xstreams.jar
init confdb data
psql:/exbase/data/object.sql:6: NOTICE: table "table_fullrefresh_result" does not exist, skipping
psql:/exbase/data/object.sql:31: NOTICE: table "incremental_offset" does not exist, skipping
psql:/exbase/data/object.sql:61: NOTICE: table "runtask" does not exist, skipping
psql:/exbase/data/object.sql:79: NOTICE: table "incremental_serviceinfo" does not exist, skipping
psql:/exbase/data/object.sql:91: NOTICE: table "tb_incremental_task_state" does not exist, skipping
psql:/exbase/data/object.sql:107: NOTICE: table "tb_consolelog" does not exist, skipping
Last login: Wed Aug 4 17:39:56 CST 2021
delete /exbase/data/object.sql
start incremental.service
● incremental.service - incremental
   Loaded: loaded (/etc/systemd/system/incremental.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2021-08-04 17:40:11 CST; 1s ago
  Main PID: 32729 (java)
     Tasks: 32
  CGroup: /system.slice/incremental.service
          └─32729 /usr/bin/java -Duser.timezone=GMT+8 -Dfile.encoding=utf-8 -Ddb2.jcc.charsetDecoderEncoder=3 -Doracle.jdbc.timezoneAsRegion=false -Dinstall-path=/exbase -jar -server -Xm
s256m -Xmx4096m -XX:PermSize=32m -XX:MaxPermSize=512m -XX:SurvivorRatio=1 -XX:NewRatio=1 -XX:UnlockExperimentalVMOptions -XX:+UseG1GC -XX:G1NewSizePercent=20 -XX:G1MaxNewSizePer
cent=50 -XX:ParallelGCThreads=8 -XX:ConcGCThreads=2 -XX:MaxGCPauseMillis=400 -XX:GCPauseIntervalMillis=2000 -XX:ReduceInitialCardMarks -XX:+UseAdaptiveSizePolicy -XX:+AggressiveOpti
mizations -XX:+UseBiasedLocking -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/exbase/data/coredump -XX:+OptimizeStringConcat -XX:+UseFastAccessorMethods -Djava.ext.dirs=/exbase/deployment/lib/incre
mental/ /exbase/deployment/incremental/incremental.jar

Aug 04 17:40:11 localhost.localdomain systemd[1]: Started incremental.
Aug 04 17:40:11 localhost.localdomain java[32729]: OpenJDK 64-Bit Server VM warning: ignoring option PermSize=32m; support was removed in 8.0
Aug 04 17:40:11 localhost.localdomain java[32729]: OpenJDK 64-Bit Server VM warning: ignoring option MaxPermSize=512m; support was removed in 8.0
Aug 04 17:40:11 localhost.localdomain java[32729]: ERROR StatusLogger No log4j2 configuration file found. Using default configuration: logging only errors to the console.
install incremental finish
set auto backup confdb
no crontab for root
no crontab for root
set firewall
FirewallD is not running
FirewallD is not running
install finish
[root@localhost exbase_environment]#
```

(4) 安装许可: 提供内存大小、线程数与网卡信息。把这些信息反馈给支持人员, 制作 licence。

安装结束后, 若是已经申请使用证书, 可以继续安装证书:


```
-----安装证书-----
是否安装证书(y/n)y
请选择证书的形式(1:压缩包/2: 源文件)和存放路径: 1
请输入压缩包的路径和压缩包名: /exbase/deployment/lic exbase3.920220914.tar.gz
cp: '/exbase/deployment/lic/exbase3.920220914.tar.gz' and '/exbase/deployment/li
激活exbase :=====
[root@vb84 exbase_environment]#
```

安装证书后即可使用 exbase。

也可在安装包所在目录下通过如下命令安装证书。

```
./add_license.sh
```

```
[root@vb84 exbase_environment]# ./add_license.sh
-----安装证书-----
分别输入安装用户 用户组 安装路径: /exbase appusr appusr
请选择证书的形式(1:压缩包/2: 源文件)和存放路径: 1
请分别输入压缩包的路径 压缩包名: /exbase/deployment/lic exbase3.920220914.tar.gz
```

(5) 启动进程

```
systemctl start incremental
systemctl start exbase
```

(6) 登录系统，网址为：<http://<服务器 ip>:31030>

(7) 若正常访问，将安装文件删除

```
rm -rf /exbase_environment*
```

3. 插件安装

注意：若需要使用反向增量迁移功能，需要在 PG/Vastbase G100/Vastbase E100 配置 decoderbufs。

3.1. 前提条件

- ❖ PG 数据库为 10 及以上版本, exBase 为 2.10 及以上版本, Vastbase [G100](#) 为 V2.2.3 及以上版本
- ❖ PG/[Vasbase E100](#)/[Vastbase G100](#) 在 exBase 配置数据源时使用的用户需要有 replication 的权限
 - 授权语句: alter user 用户名 replication;
- ❖ 若 Vastbase 作采集库 (如 Vastbase 正向增量、Oracle to Vastbase 的反向增量) , 则需要把 PUBLIC 这一 schema 的权限赋予连接用户:
 - grant all on schema public to 用户名;
- ❖ 需要在源库 Oracle/Mysql 及目标库 PG/[Vasbase E100](#)/[Vastbase G100](#) 创建增量相关表。

源库为 Oracle 的参考脚本 “oracle_pg_反向 oracle.sql” , “oracle_pg_反向 pg.sql”
源库为 Mysql 的参考脚本 “mysql_pg_反向 mysql.sql” 、 “mysql_pg_反向 pg.sql”

oracle_pg_反向 oracle.sql 脚本:

```
-- oracle_pg_反向 oracle.sql 脚本:  
drop table ${username}.kafkaoffset;  
create table ${username}.kafkaoffset (  
  jobid varchar2 (128) primary key,  
  topic varchar2 (64),  
  lastoffset number,  
  lastsuboffset number,  
  last_scn_number number,  
  scnnumber number,  
  transaction_id blob
```

```
-- oracle_pg_反向 pg.sql 脚本:  
drop table if exists ${username}.incremental_offset;  
create table ${username}.incremental_offset (  
  jobid varchar(128) primary key,  
  topic varchar(64) not null,  
  startlsn bigint,  
  collectcommitlsn bigint,  
  collectoffset bigint,  
  slotname varchar(128),  
  sourcetime timestamp,  
  updatetime timestamp default current_timestamp,  
  collecttime timestamp  
);
```

mysql_pg_反向 mysql.sql 脚本:

```
-- mysql_pg_反向 mysql.sql 脚本  
CREATE TABLE ${username}.\`kafkaoffset` (  
  \`jobid` varchar(128) NOT NULL,  
  \`topic` varchar(64) DEFAULT NULL,  
  \`lastoffset` mediumtext,  
  \`lastsuboffset` mediumtext,  
  \`last_scn_number` mediumtext,  
  \`scnnumber` mediumtext,  
  \`transaction_id` varchar(128) DEFAULT NULL,  
  \`applytime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON  
UPDATE CURRENT_TIMESTAMP,  
  \`updatetime` timestamp NULL DEFAULT NULL,  
  PRIMARY KEY (`jobid`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

mysql_pg_反向 pg.sql 脚本:

```
-- mysql_pg_反向 pg.sql 脚本  
create table ${username}.incremental_offset (  
  jobid varchar(128) primary key,  
  topic varchar(64) not null,  
  startlsn bigint,  
  collectcommitlsn bigint,  
  collectoffset bigint,  
  slotname varchar(128),  
  sourcetime timestamp,  
  updatetime timestamp default current_timestamp,  
  collecttime timestamp
```

注：在 PG/[Vastbase E100](#)/Vastbase [G100](#) 创建表时需要用 exBase 配置数据源连接时使用的用户

Vastbase [G100](#) V2.2.3 及以后版本自带 decoderbufs，可以不用编译 decoderbufs，但需要配置应用端 IP 的复制权限，在 pg_hba.conf 文件末尾新增一行“host replication user ip/掩码 md5”，其中，user 为 exBase 配置数据源连接时使用的用户，ip 为 exBase 所在服务器 ip，掩码可以为 0；在 postgresql.conf 配置文件中
进行相关配置，详见配置 decoderbufs 中的步骤 4。

3.2. 配置 decoderbufs

所需安装包 decoderbufs install.zip 存放在网盘目录：[安装包->exBase->V3.10->操作手册](#)。解压即得下述的 3 个包。

安装 decoderbufs 插件所用到的包：

protobuf-master.zip、protobuf-c.zip、decoderbufs.zip

安装前可以查下是否已有 1.2+ 版本的 protoc：

protoc --version

```
[root@localhost yum.repos.d]# protoc --version  
libprotoc 3.12.0
```

如果有可以直接看步骤 2 的环境变量设置，然后从步骤 3 开始。

1、步骤一：安装 protobuf

(1) 安装 protobuf 依赖：

```
yum install autoconf  
yum install automake  
yum install libtool  
yum install make  
yum install gcc gcc-c++  
yum install unzip
```

(2) 编译安装 protobuf:

```
unzip protobuf-master.zip
cd protobuf-master
./autogen.sh
./configure --prefix=/usr
make
make install
ldconfig
```

2、步骤二: 安装 protobuf-c

(1)设置环境变量 PKG_CONFIG_PATH

确认 PKG_CONFIG_PATH 包含 protobuf.pc 文件父目录, 对应上面的 protobuf 的安装目录下的 pkgconfig:

```
[root@localhost protobuf-master]# find / -name protobuf.pc
/root/protobuf/protobuf-master/protobuf.pc
/usr/lib/pkgconfig/protobuf.pc
```

```
export PKG_CONFIG_PATH=/usr/lib/pkgconfig:$PKG_CONFIG_PATH
```

(2)编译安装 protobuf-c:

```
unzip protobuf-c.zip
cd protobuf-c
chmod u+x autogen.sh
./autogen.sh
./configure --prefix=/usr
make
make install
```

(3)设置环境变量 LD_LIBRARY_PATH

确认 LD_LIBRARY_PATH 包含 libprotobuf-c.so 的父目录, 即对应 protobuf-c 的安装目录:

```
export LD_LIBRARY_PATH=/usr/lib:$LD_LIBRARY_PATH
```

(1)(3) 设置的环境变量也可以写入 ~/.bashrc 或者 /etc/profile 等文件持久化。

3、编译安装 decoderbufs

\$PGHOME 指数据库安装目录

```
unzip postgres-decoderbufs.zip
cd postgres-decoderbufs
export PATH=$PGHOME/bin:$PATH
make
```

```
make install
```

如需要为其它的同一版本的数据库安装，可以复制使用同版本数据库编译生成的插件，在当前目录分别复制编译生成的 decoderbufs.so 和 decoderbufs.control 文件至对应目录，其中\$PGHOME 是 PG 的安装目录：

```
mv decoderbufs.so $PGHOME/lib/postgresql
mv decoderbufs.control $PGHOME/share/postgresql/extension
```

若没有 postgresql 目录，统一去掉 postgresql 子目录：

```
mv decoderbufs.so $PGHOME/lib/
mv decoderbufs.control $PGHOME/share/extension
```

将两个文件的用户权限修改为 PG 安装用户权限

4、修改配置文件 postgresql.conf

(1) wal_level 改为 logical

```
# - Settings -
wal_level = logical                # minimal, replica, or logical
                                   # (change requires restart)
```

(2) shared_preload_libraries 选项添加 decoderbufs :

```
                                   # (change requires restart)
shared_preload_libraries = 'decoderbufs'# (change requires restart)
```

(3) 根据需要调整 max_wal_senders、max_replication_slots 和 wal_sender_timeout 的数量。

```
#max_wal_senders = 10
```

```
#max_replication_slots = 10
```

默认值是 10 ，最多只能建立 10 个复制槽。

```
#wal_sender_timeout = 50s
```

重启 PG 数据库。启动前注意启动用户的环境变量 LD_LIBRARY_PATH 需要包含 libprotobuf-c.so 的父目录。

3.3. 集群配置

把 Vastbase G100 作为采集库时，如为集群架构，需要进行如下配置。

(1) 修改 has 配置文件，其为 has/etc 下的 yaml 文件：

```
has/bin/hasctl -c has/etc/patroni conf.yaml edit-config
```

(2) 执行上述命令后进入修改界面，修改 wal_level=logical

(3) 保存退出

(4) 重启集群

4. exBase 的启动、停止

(1) 服务启动，安装时已将服务加入自启项，手工启动命令如下

➤ 启动配置库

```
systemctl start atlasdb
```

➤ 启动 incremental 进程

```
systemctl start incremental
```

➤ 启动 exbase 进程

```
systemctl start exbase
```

(2) 服务停止

➤ 停止 exbase 进程

```
systemctl stop exbase
```

➤ 停止 incremental 进程

```
systemctl stop incremental
```

➤ 停止配置库

```
systemctl stop atlasdb
```

(3) 服务重启

➤ 重启配置库

```
systemctl restart atlasdb
```

➤ 重启 incremental 进程

```
systemctl restart incremental
```

➤ 重启 exbase 进程

```
systemctl restart exbase
```

(4) 查看进程状态

➤ 查看 exbase 进程状态

```
systemctl status exbase
```

➤ 查看 incremental 进程状态

```
systemctl status incremental
```

➤ 查看配置库状态

```
systemctl status atlasdb
```

5. 补丁包安装说明

前置条件：补丁包的安装是基于现有系统上进行的，因此，安装补丁包前需要确保目标机器已安装了 exbase。

补丁包大小约 150MB，将安装包上传，并解压到任意目录。此处以 Upgrade_exbase-3.7_centos-7.x86_64.2021092720.zip 为例：

```
unzip Upgrade_exbase-3.7_centos-7.x86_64.2021092720.zip
```

解压后会生成一个 upgrade 目录，进入目录执行安装命令：

```
cd upgrade/  
./setup.sh
```

执行安装脚本时需要输入原有 exbase 的安装路径，脚本会根据输入的路径校验目录是否为 exbase 的安装路径。若是自定义用户和用户组，执行 setup.sh 的时候需要输入安装的用户和用户组：

```
# ./setup.sh appusr appusr
```

执行 upgrade_sql.sh 的时候需要输入安装路径，安装用户，配置库端口：

```
./upgrade_sql.sh /exbase appusr 31003
```

检查是否重启成功，重启后 exbase.log 会有如下打印信息

```
11:32:12:540 [INFO] - cn.com.atlasdata.exbase.schedule.IncrementalServiceHeartbeat.heartbeat(fa:78) - 安装包重启成功...  
11:32:12:542 [INFO] - cn.com.atlasdata.exbase.schedule.IncrementalServiceHeartbeat.heartbeat(fa:78) - 心跳检测启动...  
11:31:00:640 [INFO] - cn.com.atlasdata.exbase.license.LicenseChecker.run(qb:7) - -----执行定时检查license任务-----  
11:32:11:545 [INFO] - org.springframework.boot.StartupInfoLogger.logStarting(StartupInfoLogger.java:50) - Starting ExbaseApplication v3.2-SNAPSHOT on localhost.localdomain with  
PID 30566 (/exbase/deployment/exbase/exbase.jar started by appusr in /)  
11:32:11:545 [INFO] - org.springframework.boot.SpringApplication.logStartupProfileInfo(SpringApplication.java:675) - No active profile set, falling back to default profiles: de  
fault  
11:32:14:790 [INFO] - org.apache.juli.logging.DirectJDKLog.log(DirectJDKLog.java:173) - Initializing ProtocolHandler ["https-jsse-nio-31030"]  
11:32:14:806 [INFO] - org.apache.juli.logging.DirectJDKLog.log(DirectJDKLog.java:173) - Starting service [Tomcat]  
11:32:14:808 [INFO] - org.apache.juli.logging.DirectJDKLog.log(DirectJDKLog.java:173) - Starting Servlet Engine: Apache Tomcat/9.0.13  
11:32:14:829 [INFO] - org.apache.juli.logging.DirectJDKLog.log(DirectJDKLog.java:173) - The APR based Apache Tomcat Native library which allows optimal performance in productio  
n environments was not found on the java.library.path: [/exbase/deployment/lib/so:/usr/lib64:/usr/lib:/usr/java/packages/lib/amd64:/usr/lib64:/lib64:/lib:/usr/lib]  
11:32:15:192 [INFO] - org.apache.juli.logging.DirectJDKLog.log(DirectJDKLog.java:173) - Initializing Spring embedded WebApplicationContext  
11:32:15:544 [INFO] - cn.com.atlasdata.helper.debuglog.DebugLog.info(DebugLog.java:156) - [UnitPool] cn.com.atlasdata.businessHelper.jdbc.ConnectionPoolManager (ConnectionPoolM  
anager.java:100): dbtype ->postgresql  
11:32:15:545 [INFO] - cn.com.atlasdata.helper.debuglog.DebugLog.info(DebugLog.java:156) - [UnitPool] cn.com.atlasdata.businessHelper.jdbc.ConnectionPoolManager (ConnectionPoolM  
anager.java:105): Init connection succeeded  
11:32:15:589 [INFO] - cn.com.alibaba.druid.pool.DruidDataSource.init(DruidDataSource.java:889) - {dataSource-1} inited  
11:32:16:250 [INFO] - cn.com.atlasdata.exbase.ExbaseApplication.setJdbcInfo(ExbaseApplication.java:250) - init jdbc driver:oracle  
11:32:16:260 [INFO] - cn.com.atlasdata.exbase.ExbaseApplication.setJdbcInfo(ExbaseApplication.java:250) - init jdbc driver:postgresql  
11:32:16:260 [INFO] - cn.com.atlasdata.exbase.ExbaseApplication.setJdbcInfo(ExbaseApplication.java:250) - init jdbc driver:atlasdb  
11:32:16:260 [INFO] - cn.com.atlasdata.exbase.ExbaseApplication.setJdbcInfo(ExbaseApplication.java:250) - init jdbc driver:thbase  
11:32:16:260 [INFO] - cn.com.atlasdata.exbase.ExbaseApplication.setJdbcInfo(ExbaseApplication.java:250) - init jdbc driver:dm  
11:32:16:261 [INFO] - cn.com.atlasdata.exbase.ExbaseApplication.setJdbcInfo(ExbaseApplication.java:250) - init jdbc driver:vastbase  
11:32:16:261 [INFO] - cn.com.atlasdata.exbase.ExbaseApplication.setJdbcInfo(ExbaseApplication.java:250) - init jdbc driver:sysql  
11:32:16:261 [INFO] - cn.com.atlasdata.exbase.ExbaseApplication.setJdbcInfo(ExbaseApplication.java:250) - init jdbc driver:tdsql  
11:32:16:735 [INFO] - cn.com.atlasdata.exbase.schedule.AutoCleanHistoryData.cleanHistoryData(fa:176) - 自动清理历史数据启动...  
11:32:16:739 [INFO] - org.apache.juli.logging.DirectJDKLog.log(DirectJDKLog.java:173) - Starting ProtocolHandler ["https-jsse-nio-31030"]  
11:32:16:763 [INFO] - cn.com.atlasdata.exbase.schedule.AutoCleanHistoryData.cleanHistoryData(fa:102) - 开始回收空间...  
11:32:16:870 [INFO] - cn.com.atlasdata.exbase.schedule.AutoCleanHistoryData.cleanHistoryData(fa:58) - 回收空间完毕  
11:32:16:870 [INFO] - cn.com.atlasdata.exbase.schedule.IncrementalServiceHeartbeat.heartbeat(ua:21) - 心跳检测启动...  
11:32:16:158 [INFO] - org.apache.juli.logging.DirectJDKLog.log(DirectJDKLog.java:173) - Using a shared selector for servlet write/read  
11:32:16:197 [INFO] - org.springframework.boot.StartupInfoLogger.logStarted(StartupInfoLogger.java:59) - Started ExbaseApplication in 0.669 seconds (JVM running for 10.413)  
11:33:10:348 [INFO] - cn.com.atlasdata.exbase.license.LicenseChecker.run(qb:216) - -----执行定时检查license任务-----  
[root@localhost upgrade]#
```

6. 使用前的补充操作建议

1、迁移目标库为 VastBase G100 时，检查目标库文件描述符上限是否满足迁移需求，推荐设置为 10240000。

以下是修改系统资源限制的方法：

#数据库操作系统用户 Vastbase limits 配置

```
echo "vastbase soft core unlimited">>/etc/security/limits.conf
```

```
echo "vastbase hard core unlimited">>/etc/security/limits.conf
```

```
echo "vastbase hard nproc unlimited">>/etc/security/limits.conf
```

```
echo "vastbase soft nproc unlimited">>/etc/security/limits.conf
```

```
echo "vastbase hard memlock unlimited">>/etc/security/limits.conf
```

```
echo "vastbase soft memlock unlimited">>/etc/security/limits.conf
```

```
echo "vastbase hard nofile 1024000">>/etc/security/limits.conf
```

```
echo "vastbase soft nofile 1024000">>/etc/security/limits.conf
```

2、支持的浏览器及版本：

Chrome79 及以上版本，64bit

Firefox53 及以上 3 版本，64bit

QQ 浏览器及 10 以上版本

以及其他主流浏览器的较新版本



电话: 010-82838118

地址: 北京市海淀区学院路 30 号科大天工大厦 B 座 6 层

官网: www.vastdata.com.cn