**图表

描述已自动生成**

**【客户名称】**

**Vastbase G100数据库巡检报告**

图片包含 游戏机, 盘子, 食物, 画

描述已自动生成

北京海量数据技术股份有限公司

20XX年XX月

**修改历史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **日期** | **版本号** | **作者** | **修改说明** | **更改请求号** |
| 2021/11/16 | 1.0 | 何俊庆 | 新建 |  |
|  |  |  |  |  |
|  |  |  |  |  |

注释：“变更请求号”为文档正式发布后需要变更时的编号。

**审阅**

|  |  |  |
| --- | --- | --- |
| **姓名** | **日期** | **备注** |
|  |  |  |
|  |  |  |
|  |  |  |

**分发**

|  |  |  |  |
| --- | --- | --- | --- |
| **公司** | **姓名** | **日期** | **备注** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**目 录**

[1. 概述及总结 4](#_Toc466421306)

[1.1 概要 4](#_Toc827578392)

[1.2 巡检范围 4](#_Toc2005936372)

[1.3 总结建议 4](#_Toc426829951)

[2. 【Vastbase】数据库巡检 5](#_Toc1135605477)

[2.1 操作系统检查与分析结果 5](#_Toc1434081050)

[2.1.1 操作系统配置文件、静态配置信息 5](#_Toc1391237069)

[2.1.2 selinux 动态配置信息 5](#_Toc719470147)

[2.1.3 Transparent Huge Pages 5](#_Toc1801828019)

[2.2 数据库运行结果 7](#_Toc1656608986)

[2.2.1 配置文件检查 7](#_Toc501744347)

[2.2.2 用户或数据库级别定制参数 8](#_Toc1796441907)

[2.2.3 数据库错误日志分析 8](#_Toc1326537776)

[2.2.4 数据库空间使用分析 8](#_Toc2092661725)

[2.2.5 数据库连接分析 9](#_Toc2025925156)

[2.2.6 数据库性能分析 9](#_Toc1370873707)

[2.2.7 数据库垃圾分析 10](#_Toc2069828533)

[2.2.8 数据库年龄分析 10](#_Toc520556378)

[2.2.9 数据库归档和流复制状态分析 11](#_Toc142667168)

[2.2.10 数据库风险分析 11](#_Toc1215342524)

[2.2.11 重置统计信息 12](#_Toc1544834251)

1. 概述及总结
   1. 概要

数据库在整个业务系统中处于非常核心的地位，数据库的性能好坏将直接影响到整个应用系统的性能。但往往在应用系统实际运行过程中，由于系统数据量的增加、业务模块应用逻辑的修改、应用版本变更、用户量增长等各种原因都可能使数据库性能下降，从而导致应用系统性能下降，影响用户感知。因此需要定期对数据库进行全面的性能分析，分析性能变化趋势及产生这些变化的原因，以便做出优化调整，保持应用系统良好的性能。

* 1. 巡检范围

|  |  |  |
| --- | --- | --- |
| **序号** | **数据库** | **数据库版本** |
| 1 | VastbaseG100 |  |

* 1. 总结建议

|  |  |  |
| --- | --- | --- |
| **业务系统** | **分析项目** | **调整建议** |
| **XXX管理系统** | 操作系统配置文件、静态配置信息 |  |
| Transparent Huge Pages |  |
| 数据库配置文件检查 |  |
| 数据库性能分析 |  |
| 数据库垃圾分析 |  |

1. 【Vastbase】数据库巡检
   1. 操作系统检查与分析结果

|  |  |  |
| --- | --- | --- |
| **栏目** | **状态** | **调整建议** |
| **操作系统配置文件、静态配置信息** |  |  |
| **selinux 动态配置信息** |  |  |
| **Transparent Huge Pages** |  |  |

* + 1. 操作系统配置文件、静态配置信息

----->>>---->>> 主机名:

{{hostnametxt}}

----->>>---->>> IP地址信息:

{{ipaddresstxt}}

----->>>---->>> 操作系统内核:

{{oskerneltxt}}

----->>>---->>> 内存(MB):

{{memorytxt}}

----->>>---->>> CPU:

{{cputxt}}

----->>>---->>> 操作系统配置文件、静态配置信息:

----->>>---->>> /etc/sysctl.conf :

{{ossysctltxt }}

----->>>---->>> /etc/security/limits.conf :

{{userlimittxt }}

----->>>---->>> 磁盘空间：

{{dirutiltxt}}

建议:

* + 1. selinux 动态配置信息

----->>>---->>> selinux 动态配置信息:

{{selinuxtxt }}

建议:

* + 1. Transparent Huge Pages

----->>>---->>> 建议禁用Transparent Huge Pages (THP):

{{hugepagetxt}}

建议:

* 1. 数据库运行结果

|  |  |  |
| --- | --- | --- |
| **栏目** | **状态** | **调整建议** |
| **配置文件检查** |  |  |
| **用户或数据库级别定制参数** |  |  |
| **数据库错误日志** |  |  |
| **数据库空间使用分析** |  |  |
| **数据库连接分析** |  |  |
| **数据库性能分析** |  |  |
| **数据库垃圾分析** |  |  |
| **数据库年龄分析** |  |  |
| **数据库XLOG, 流复制状态分析** |  |  |
| **数据库安全或潜在风险分析** |  |  |
| **重置统计信息** |  |  |

* + 1. **数据库信息**

----->>>---->>> 数据库版本:

{{versiontxt}}

----->>>---->>> 用户已安装的插件版本:

{{dbextensiontxt}}

----->>>---->>> 用户使用了多少种数据类型:

{{typetxt}}

----->>>---->>> 用户创建了多少对象:

{{objectstxt}}

----->>>---->>> 用户对象占用空间的柱状图:

{{dbspacetxt}}

建议:

* + 1. 配置文件检查

----->>>---->>> 获取pg\_hba.conf md5值:

{{hbamdtxt}}

建议: 主备md5值一致(判断主备配置文件是否内容一致的一种手段, 或者使用diff).

----->>>---->>> 获取pg\_hba.conf配置:

{{pghbatxt}}

建议:

主备配置尽量保持一致, 注意trust和password认证方法的危害(password方法 验证时网络传输密码明文, 建议改为md5), 建议除了unix socket可以使用trust以外, 其他都使用md5或者LDAP认证方法.

建议先设置白名单(超级用户允许的来源IP, 可以访问的数据库), 再设置黑名单(不允许超级用户登陆, reject), 再设置白名单(普通应用), 参考pg\_hba.conf中的描述.

----->>>---->>> 获取postgresql.conf md5值:

{{mdtxt}}

建议:

主备md5值一致(判断主备配置文件是否内容一致的一种手段, 或者使用diff).

----->>>---->>> 获取postgresql.conf配置:

{{postgresqlconftxt}}

建议:

建议修改的参数列表如下:

* + 1. 用户或数据库级别定制参数

----->>>---->>> 用户或数据库级别定制参数:

{{dbuserconftxt}}

建议:

* + 1. 数据库错误日志分析

----->>>---->>> 获取错误日志信息:

{{errorlogtxt}}

----->>>---->>> 获取连接请求情况:

{{dbconnectiontxt}}

----->>>---->>> 获取认证失败情况:

{{authenticationfailedtxt}}

建议：

当前系统运行正常，建议继续保持

* + 1. 数据库空间使用分析

----->>>---->>> 输出表空间对应目录:

{{echotablespacetxt}}

建议:

注意表空间如果不是软链接, 注意是否刻意所为, 正常情况下应该是软链接.

----->>>---->>> 输出表空间使用情况:

{{tablespaceusgtxt}}

建议:

注意检查表空间所在文件系统的剩余空间。

----->>>---->>> 输出数据库使用情况:

{{databaseusgtxt}}

建议:

注意检查数据库的大小, 是否需要清理历史数据.

----->>>---->>> TOP 10 size对象:

{{topobjecttxt}}

建议:

经验值: 单表超过8GB, 并且这个表需要频繁更新 或 删除+插入的话, 建议对表根据业务逻辑进行合理拆分后获得更好的性能, 以及便于对膨胀索引进行维护; 如果是只读的表, 建议适当结合SQL语句进行优化.

* + 1. 数据库连接分析

----->>>---->>> 当前活跃度:

{{activitytxt}}

建议:

如果active状态很多, 说明数据库比较繁忙. 如果idle in transaction很多, 说明业务逻辑设计可能有问题. 如果idle很多, 可能使用了连接池, 并且可能没有自动回收连接到连接池的最小连接数.

----->>>---->>> 总剩余连接数:

{{countconnectiontxt}}

建议:

给超级用户和普通用户设置足够的连接, 以免不能登录数据库.

----->>>---->>> 用户连接数限制:

{{userlimittxt}}

建议:

给用户设置足够的连接数, alter role ... CONNECTION LIMIT .

----->>>---->>> 数据库连接限制:

{{databaselimittxt}}

建议:

给数据库设置足够的连接数, alter database ... CONNECTION LIMIT .

* + 1. 数据库性能分析

----->>>---->>> TOP 5 SQL : total\_cpu\_time

{{totalcputimetxt}}

建议:

检查SQL是否有优化空间, 配合auto\_explain插件在csvlog中观察LONG SQL的执行计划是否正确.

----->>>---->>> 索引数超过4并且SIZE大于10MB的表:

{{indexmbtxt}}

建议:

索引数量太多, 影响表的增删改性能, 建议检查是否有不需要的索引.

----->>>---->>> 上次巡检以来未使用或使用较少的索引:

{{littleusedindextxt}}

----->>>---->>> 数据库统计信息, 回滚比例, 命中比例, 数据块读写时间, 死锁, 复制冲突:

{{countinformationtxt}}

建议:

回滚比例大说明业务逻辑可能有问题, 命中率小说明shared\_buffer要加大, 数据块读写时间长说明块设备的IO性能要提升, 死锁次数多说明业务逻辑有问题, 复制冲突次数多说明备库可能在跑LONG SQL.

----->>>---->>> 检查点, bgwriter 统计信息:

{{bgwritertxt}}

建议:

checkpoint\_write\_time多说明检查点持续时间长, 检查点过程中产生了较多的脏页.

checkpoint\_sync\_time代表检查点开始时的shared buffer中的脏页被同步到磁盘的时间, 如果时间过长, 并且数据库在检查点时性能较差, 考虑一下提升块设备的IOPS能力.

buffers\_backend\_fsync太多说明需要加大shared buffer 或者 减小bgwriter\_delay参数.

* + 1. 数据库垃圾分析

----->>>---->>> 表引膨胀检查:

{{tableflationchecktxt}}

----->>>---->>> 索引膨胀检查:

{{indexinflationchecktxt}}

建议:

如果索引膨胀太大, 会影响性能, 建议重建索引, create index CONCURRENTLY ... .

----->>>---->>> 垃圾数据:

{{junkdatatxt}}

建议:

通常垃圾过多, 可能是因为无法回收垃圾, 或者回收垃圾的进程繁忙或没有及时唤醒, 或者没有开启autovacuum, 或在短时间内产生了大量的垃圾 .

可以等待autovacuum进行处理, 或者手工执行vacuum table .

* + 1. 数据库年龄分析

----->>>---->>> 数据库年龄:

{{databaseagetxt}}

建议:

数据库的年龄正常情况下应该小于vacuum\_freeze\_table\_age, 如果剩余年龄小于5亿, 建议人为干预, 将LONG SQL或事务杀掉后, 执行vacuum freeze .

----->>>---->>> 表年龄:

{{tableagetxt}}

建议:

表的年龄正常情况下应该小于vacuum\_freeze\_table\_age, 如果剩余年龄小于5亿, 建议人为干预, 将LONG SQL或事务杀掉后, 执行vacuum freeze .

----->>>---->>> 长事务, 2PC:

{{longtransactiontxt}}

建议:

长事务过程中产生的垃圾, 无法回收, 建议不要在数据库中运行LONG SQL, 或者错开DML高峰时间去运行LONG SQL. 2PC事务一定要记得尽快结束掉, 否则可能会导致数据库膨胀.

参考: http://blog.163.com/digoal@126/blog/static/1638770402015329115636287/

* + 1. 数据库归档和流复制状态分析

---->>>---->>> 是否开启归档, 自动垃圾回收:

{{ontxt}}

建议:

建议开启自动垃圾回收, 开启归档.

----->>>---->>> 归档统计信息:

{{archivecounttxt}}

----->>>---->>> 流复制统计信息:

{{streamtxt}}

建议:

关注流复制的延迟, 如果延迟非常大, 建议排查网络带宽, 以及本地读xlog的性能, 远程写xlog的性能.

----->>>---->>> 流复制插槽:

{{replicationsolttxt}}

* + 1. 数据库风险分析

----->>>---->>> 密码泄露检查

{{checklogtxt}}

{{checkstatementstxt}}

{{pgauthidtxt}}

{{pgviewstxt}}

建议:

暂未出现密码泄露问题

---->密码到期检查

{{passwdauthidtxt}}

建议:

到期后, 用户将无法登陆, 记得修改密码, 同时将密码到期时间延长到某个时间或无限时间，暂不需要修改。

----->>>---->>> 普通用户对象上的规则安全检查

{{usersafetxt}}

建议:

防止普通用户在规则中设陷阱, 注意有危险的security invoker的函数调用, 超级用户可能因为规则触发后误调用这些危险函数(以invoker角色).

----->>>---->>> 普通用户自定义函数安全检查

{{userfunctiontxt}}

建议:

防止普通用户在函数中设陷阱, 注意有危险的security invoker的函数调用, 超级用户可能因为触发器触发后误调用这些危险函数(以invoker角色).

----->>>---->>> unlogged table 和 哈希索引

{{unloggedtabletxt}}

建议

unlogged table和hash index不记录XLOG, 无法使用流复制或者log shipping的方式复制到standby节点, 如果在standby节点执行某些SQL, 可能导致报错或查不到数据

在数据库CRASH后无法修复unlogged table和hash index, 不建议使用.

PITR对unlogged table和hash index也不起作用.

----->>>---->>> 触发器, 事件触发器：

{{ddltxt}}

建议

请管理员注意触发器和事件触发器的必要性.

----->>>---->>> 检查是否使用了a-z 0-9 \_ 以外的字母作为对象名：

{{specialtxt}}

建议：

建议任何identify都只使用 a-z, 0-9, \_ (例如表名, 列名, 视图名, 函数名, 类型名, 数据库名, schema名, 物化视图名等等).

----->>>---->>> 锁等待:

{{lockwaittxt}}

建议:

锁等待状态, 反映业务逻辑的问题或者SQL性能有问题, 建议深入排查持锁的SQL.

----->>>---->>> 继承关系检查:

{{inheritancecheckingtxt}}

建议:

如果使用继承来实现分区表, 注意分区表的触发器中逻辑是否正常, 对于时间模式的分区表是否需要及时加分区, 修改触发器函数 . "

建议继承表的权限统一, 如果权限不一致, 可能导致某些用户查询时权限不足.

* + 1. 重置统计信息

----->>>---->>> 重置pg\_stat\_statements统计信息:

{{resettxt}}

建议: