



# 数据库配置

## Enterprise applications

NetApp  
May 09, 2024

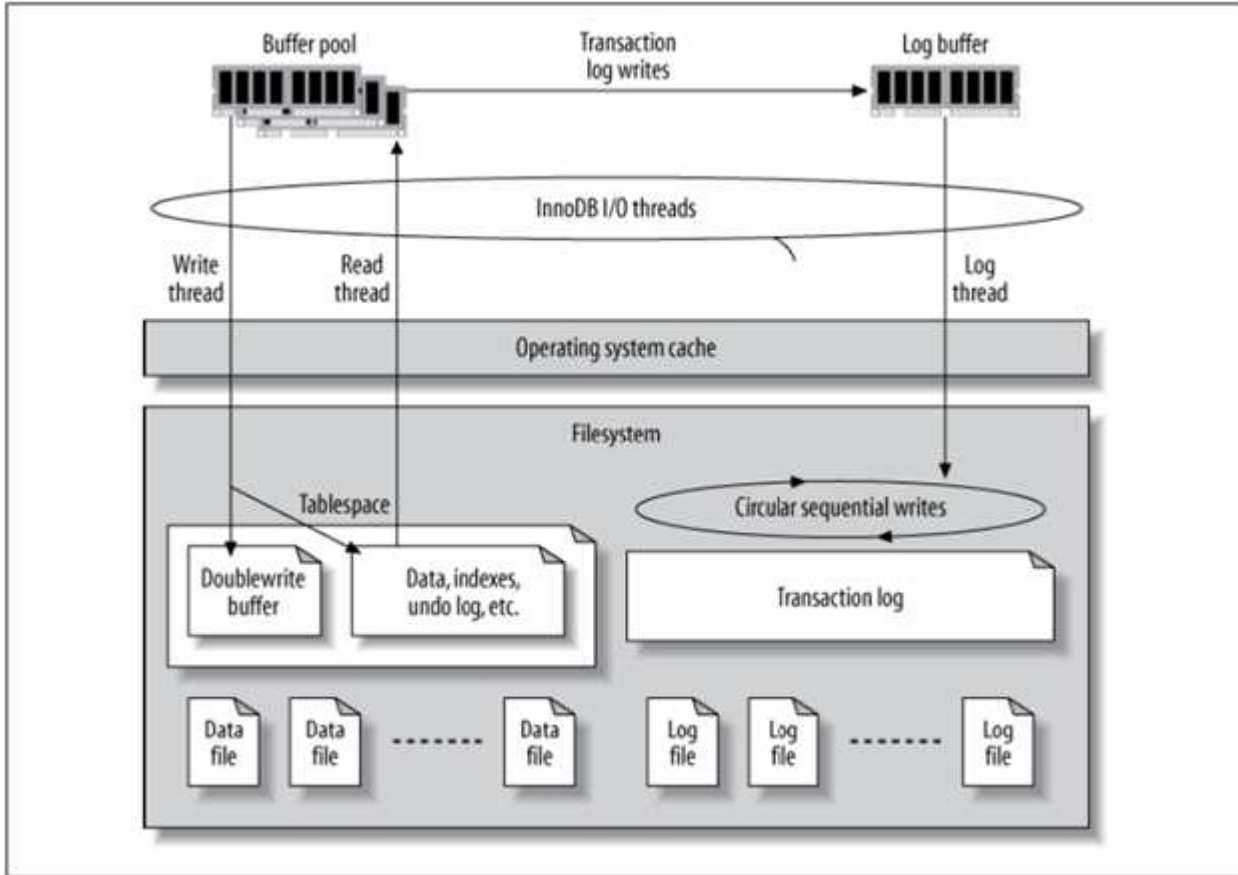
# 目录

数据库配置 .....	1
MySQL和InnoDB .....	1
MySQL配置参数 .....	3
InnoDB_log_file_size .....	3
INNODB_FLOG_AT_TRx_Commit .....	4
InnoDB_doublewrite .....	4
InnoDB_buffer_pool_size .....	5
INNODB_FLUG_METHOD .....	5
InnoDB_IO_Capacity .....	5
InnoDB_LRU_SCAN_深度 .....	6
open_file_极限值 .....	6

# 数据库配置

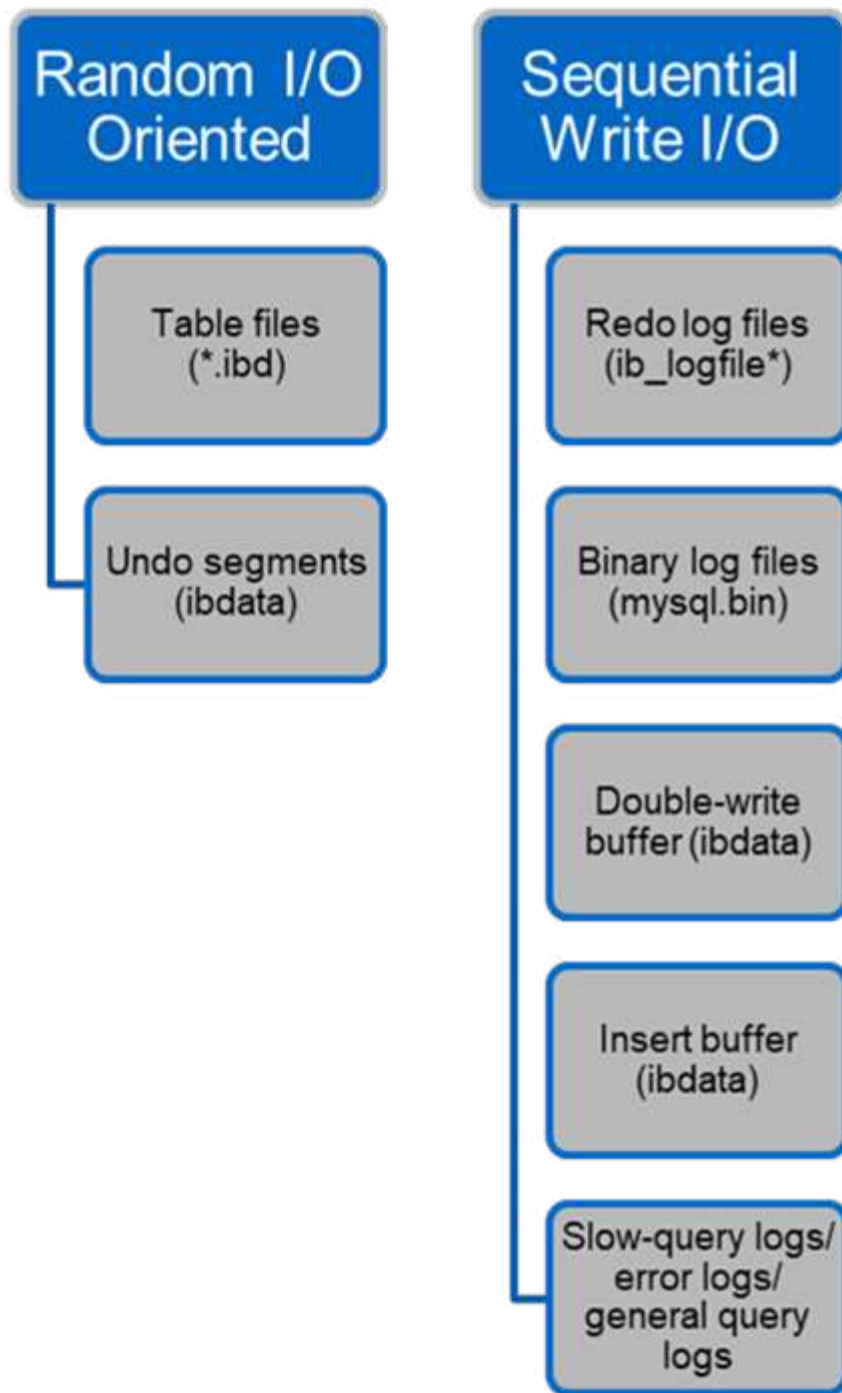
## MySQL和InnoDB

InnoDB充当存储和MySQL服务器之间的中间层、它将数据存储到驱动器。



MySQL I/O分为两类：

- 随机文件I/O
- 顺序文件I/O



数据文件会随机读取和覆盖、从而导致IOPS较高。因此、建议使用SSD存储。

重做日志文件和二进制日志文件是事务日志。它们会按顺序写入、因此、您可以通过写入缓存在HDD上获得良好的性能。恢复时会发生顺序读取、但很少会导致性能问题、因为日志文件大小通常小于数据文件、顺序读取比随机读取(发生在数据文件上)速度快。

双写缓冲区是InnoDB的一项特殊功能。InnoDB首先将已刷新的页面写入双写入缓冲区、然后将这些页面写入数据文件上的正确位置。此过程可防止页面损坏。如果没有双写入缓冲区、则在向驱动器写入的过程中发生电源故障时、页面可能会损坏。由于写入双写入缓冲区是顺序的、因此针对HDD进行了高度优化。恢复时进行顺序读取。

由于ONTAP NVRAM已提供写入保护、因此不需要双写缓冲。MySQL有一个参数、`skip_innodb_doublewrite`，禁用双写缓冲区。此功能可以显著提高性能。

插入缓冲区也是InnoDB的一项特殊功能。如果内存中不存在非唯一的二级索引块、则InnoDB会将条目插入到插入缓冲区中、以避免随机I/O操作。插入缓冲区会定期合并到数据库中的二级索引树中。插入缓冲区通过将I/O请求合并到同一个块来减少I/O操作的数量；随机I/O操作可以是顺序的。插入缓冲区也针对HDD进行了高度优化。正常操作期间会同时进行顺序写入和读取。

撤消段是随机I/O导向的。为了保证多版本并发(MVCC)，InnoDB必须在撤消段中注册旧图像。从撤消段读取以前的图像需要随机读取。如果您运行具有可重复读取的长事务(例如、`myq_dump`—单个事务)或运行长查询、则可能会发生随机读取。因此、在这种情况下、最好在SSD上存储撤消区块。如果只运行短事务或查询、则随机读取不是问题描述。

\*由于InnoDB I/O的特性、NetApp建议\*采用以下存储设计布局。



- 一个卷、用于存储面向MySQL的随机和顺序I/O文件
- 另一个卷、用于存储MySQL的纯顺序I/O导向文件

此布局还有助于您设计数据保护策略和策略。

## MySQL配置参数

NetApp建议使用一些重要的MySQL配置参数来获得最佳性能。

Parameters	值
<code>InnoDB_log_file_size</code>	256 M
<code>INNODB_FLOG_AT_TRx_Commit</code>	2.
<code>InnoDB_doublewrite</code>	0
<code>INNODB_FLUG_METHOD</code>	同步
<code>InnoDB_buffer_pool_size</code>	11g
<code>InnoDB_IO_Capacity</code>	8192.
<code>InnoDB_buffer_pool_instances</code>	8.
<code>InnoDB_LRU_SCAN_深度</code>	8192.
<code>open_file_Limit</code>	65535

要设置本节中所述的参数、必须在MySQL配置文件(`my.cnf`)中进行更改。NetApp最佳实践是内部测试的结果。

## InnoDB\_log\_file\_size

为InnoDB日志文件大小选择合适的大小对于写入操作以及在服务器崩溃后有适当的恢复时间非常重要。

由于登录到该文件的事务太多、因此日志文件大小对于写入操作非常重要。修改记录后、所做的更改不会立即写入表空间。而是将更改记录在日志文件的末尾、并将页面标记为脏。InnoDB使用其日志将随机I/O转换为顺序I/O

日志已满时、脏页会按顺序写出到表空间中、以释放日志文件中的空间。例如、假设服务器在事务处理期间崩溃、而写入操作仅记录在日志文件中。在服务器恢复运行之前、它必须经过恢复阶段、在此阶段中、系统将重写记录在日志文件中的更改。日志文件中的条目越多、服务器恢复所需的时间就越长。

在此示例中、日志文件大小会影响恢复时间和写入性能。在为日志文件大小选择正确的数字时、请平衡恢复时间与写入性能。通常、128 M到512 M之间的值都很好。

## INNODB\_FLUSH\_LOG\_AT\_TRX\_COMMIT

数据发生更改时、更改不会立即写入存储。

相反、数据会记录在日志缓冲区中、日志缓冲区是InnoDB分配给缓冲区的内存的一部分、用于缓冲日志文件中记录的更改。在提交事务时、缓冲区已满时或每秒一次(以先发生的事件为准)、InnoDB会将缓冲区转至日志文件。用于控制此过程的配置变量为InnoDB\_FLUSH\_LOG\_AT\_TRX\_COMMIT。值选项包括：

- 设置时 `innodb_flush_log_trx_at_commit=0`，InnoDB会将修改后的数据(位于InnoDB缓冲池中)写入日志文件(`ib_logfile`)，并每秒刷新日志文件(写入存储)。但是、提交事务时、它不会执行任何操作。如果发生电源故障或系统崩溃、则未转储的数据都不可恢复、因为它不会写入日志文件或驱动器。
- 设置时 `innodb_flush_log_trx_commit=1`，则InnoDB会将日志缓冲区写入事务日志，并将每个事务转储到持久存储。例如、对于所有事务提交、InnoDB先写入日志、然后再写入存储。存储速度较慢会对性能产生负面影响；例如、每秒InnoDB事务数会减少。
- 设置时 `innodb_flush_log_trx_commit=2`，InnoDB会在每次提交时将日志缓冲区写入日志文件；但是，它不会将数据写入存储。InnoDB每秒刷新一次数据。即使出现电源故障或系统崩溃、选项2数据也会显示在日志文件中、并且可以恢复。

如果性能是主要目标、请将该值设置为2。由于InnoDB每秒向驱动器写入一次、而不是每次提交事务、因此性能会显著提高。如果发生电源故障或崩溃、可以从事务日志中恢复数据。

如果数据安全是主要目标、请将该值设置为1、以便每次提交事务时、InnoDB都会转至驱动器。但是、性能可能会受到影响。



\* NetApp建议\*将InnoDB\_FLUSH\_LOG\_AT\_TRX\_COMMIT值设置为2以提高性能。

## InnoDB\_doublewrite

时间 `innodb_doublewrite` 处于启用状态(默认值)时、InnoDB会将所有数据存储两次：首先存储到双写入缓冲区、然后存储到实际数据文件。

您可以使用关闭此参数 `--skip-innodb_doublewrite` 适用于基准测试、或者您更关心最高性能而非数据完整性或可能出现的故障。InnoDB使用一种称为双写的文件转储技术。在将页面写入数据文件之前、InnoDB会将其写入一个称为双写入缓冲区的连续区域。在完成对双写缓冲区的写入和刷新后、InnoDB会将页面写入到其在数据文件中的正确位置。如果操作系统或myqld进程在页面写入期间崩溃、则InnoDB稍后可以在崩溃恢复期间从双写入缓冲区中找到一个良好的页面副本。



\* NetApp建议\*禁用双写缓冲区。ONTAP NVRAM具有相同的功能。双重缓冲会不必要地损害性能。

## InnoDB\_buffer\_pool\_size

InnoDB缓冲池是任何调整活动最重要的部分。

InnoDB在很大程度上依赖于缓冲区池来缓存索引并对数据、自适应哈希索引、插入缓冲区以及内部使用的许多其他数据结构进行分段。缓冲池还会缓冲对数据的更改、以便不必立即对存储执行写入操作、从而提高性能。缓冲池是InnoDB不可或缺的一部分、必须相应地调整其大小。设置缓冲池大小时、请考虑以下因素：

- 对于仅限InnoDB的专用计算机、请将缓冲池大小设置为可用RAM的80%或更多。
- 如果它不是MySQL专用服务器、请将大小设置为RAM的50%。

## INNODB\_FLUSH\_METHOD

InnoDB\_flush\_方法 参数用于指定InnoDB如何打开和刷新日志和数据文件。

### 优化

在InnoDB优化中、如果适用、设置此参数会调整数据库性能。

以下选项用于通过InnoDB转储文件：

- `fsync`。InnoDB使用 `fsync()` 系统调用以刷新数据和日志文件。此选项为默认设置。
- `O_DSYNC`。InnoDB使用 `O_DSYNC` 选项打开并刷新日志文件，并使用`fsync()`刷新数据文件。InnoDB不使用 `O_DSYNC` 这是因为在许多UNIX版本上都存在问题。
- `O_DIRECT`。InnoDB使用 `O_DIRECT` 选项(或 `directio()` 在Solaris上)以打开数据文件并使用 `fsync()` 刷新数据和日志文件。此选项在某些版本的GNU或Linux、FreeBSD,和Solaris上可用。
- `O_DIRECT_NO_FSYNC`。InnoDB使用 `O_DIRECT` 选项；但是、它会跳过 `fsync()` 系统调用。此选项不适用于某些类型的文件系统(例如XFS)。如果不确定文件系统是否需要 `fsync()` 系统调用(例如、要保留所有文件元数据)、请使用 `O_DIRECT` 选项。

### 观察结果

在NetApp实验室测试中、`fsync` NFS和SAN上使用了默认选项、与相比、这是一个非常好的性能提升 `O_DIRECT`。将刷新方法用作时 `O_DIRECT` 通过使用ONTAP、我们发现客户端会以串行方式在4096块的边界处写入大量单字节写入。这些写入会增加网络延迟并降低性能。

## InnoDB\_IO\_Capacity

在InnoDB插件中、从MySQL 5.7中添加了一个名为InnoDB\_io\_Capacity的新参数。

它控制InnoDB执行的最大IOPS数(包括异常页面的转储速率以及插入缓冲区[ibuf]批大小)。InnoDB\_IO\_Capacity 参数可通过InnoDB后台任务(例如从缓冲区池转储页面以及合并更改缓冲区中的数据)设置IOPS上限。

将InnoDB\_IO\_Capacity参数设置为系统每秒可执行的大约I/O操作数。理想情况下、应尽可能降低设置值、但不要太低、以免后台活动减慢。如果设置值过高、则会从缓冲池中删除数据、并且插入缓冲区的速度过快、无法缓存、因此无法提供显著优势。



\* NetApp建议\*如果通过NFS使用此设置、则分析IOPS (Sysbench/FIO)的测试结果并相应地设置参数。除非您在InnoDB缓冲池中看到的已修改或脏页面多于所需数量、否则请使用可能的最小值进行转储和清除以保持最新。



请勿使用极值(如20、000或更多)、除非您已证明较低的值不足以满足工作负载要求。

InnoDB\_IO\_Capacity参数用于调节转储速率和相关I/O



如果将此参数或InnoDB\_IO\_capacity\_max参数设置得过高、并因过早转储而浪费I/O操作、可能会严重影响性能。

## InnoDB\_LRU\_SCAN\_深度

。 `innodb_lru_scan_depth` 参数会影响InnoDB缓冲池的刷新操作的算法和启发式。

调整I/O密集型工作负载的性能专家主要关注此参数。对于每个缓冲池实例、此参数用于指定页面清理程序线程应继续扫描的最近最少使用(Least Recently Used、LRU)页面列表中的下限、以查找要刷新的脏页面。此后台操作每秒执行一次。

您可以向上或向下调整该值、以最大程度地减少可用页数。不要将该值设置得比所需值高得多、因为扫描会产生显著的性能成本。此外、请考虑在更改缓冲池实例数时调整此参数、因为 `innodb_lru_scan_depth * innodb_buffer_pool_instances` 定义页面清理程序线程每秒执行的工作量。

小于默认值的设置适用于大多数工作负载。只有在典型工作负载下具有备用I/O容量时、才应考虑增加此值。相反、如果写入密集型工作负载使I/O容量饱和、请减小该值、尤其是在缓冲池较大的情况下。

## open\_file\_极限值

。 `open_file_limits` 参数用于确定操作系统允许myqld打开的文件数。

运行时此参数的值是系统允许的实际值、可能与服务器启动时指定的值不同。在MySQL无法更改打开文件数量的系统上、此值为0。有效 `open_files_limit` 值基于系统启动时指定的值(如果有)和的值 `max_connections` 和 `table_open_cache` 使用以下公式:

- $10 + \text{max\_connections} + (\text{table\_open\_cache} \times 2)$
- $\text{max\_connections} \times 5$ .
- 操作系统限制(如果为正数)
- 如果操作系统限制为无限大: `open_files_limit` 值在启动时指定; 如果无、则为5、000

服务器将尝试使用这四个值中的最大值来获取文件描述符数量。如果无法获取多个描述符、则服务器会尝试获取系统允许的多个描述符。



## 版权信息

版权所有 © 2024 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

## 商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。