



详解 MySQL 5.5 新特性

性能与扩展

MySQL® 白皮书

2010 年 12 月

目录

| | |
|---------------------------------|-----------|
| 介绍 | 3 |
| InnoDB 是默认的存储引擎..... | 3 |
| 提高性能和可扩展性 | 3 |
| 性能和可扩展性基准 | 6 |
| 提高实用性 | 8 |
| 提高易管理性和效率 | 12 |
| 提高可用性 | 12 |
| 改善检测和诊断..... | 15 |
| MySQL 5.5:生产准备软件与支持..... | 16 |
| 结论 | 17 |
| 更多信息..... | 17 |

介绍

MySQL 是当今在用的最值得信赖的开源数据库平台。基于其卓越的跨平台、应用程序栈以及瞩目的性能、可靠性和易用性，现今，许多比较流行的，访问流量大的网站都选择使用 MySQL 数据库。现在 MySQL Oracle 提供了一个完整的 LAMP（Linux、Apache、MySQL、PHP、Perl、Python 等）框架，基于 LAMP，任何规模和志向的用户或客户均可利用现有的最佳技术解决方案和支持构建自己的应用程序和产品。MySQL 5.5 提供了一组专用功能集，在当今现代化、多功能处理硬件和软件以及中间件构架涌现的环境中，极大地提高了 MySQL 的性能、可扩展性、可用性。

MySQL5.5 融合了 MySQL 数据库和 InnoDB 存储引擎的优点，能够提供高性能的数据管理解决方案，包括：

- InnoDB 作为默认的数据库存储引擎
- 提升了 Windows 系统下的系统性能和可扩展性
- 改善性能和可扩展性，全面利用各平台现代多核构架的计算能力
- 提高实用性
- 提高易管理性和效率
- 提高可用性
- 改善检测与诊断性能

本白皮书介绍性的讲解了 MySQL 5.5 和 InnoDB 的一些增强性能，这些增强性能极大地提高了系统和 MySQL 的性能。下面，我们将详细介绍每一个关键的增强性能及其实现过程。

InnoDB 是默认的存储引擎

Oracle 公司的 InnoDB 存储引擎广泛应用于基于 MySQL 的 Web、电子商务、金融系统、健康护理以及零售应用。因为 InnoDB 可提供高效的 ACID 独立性(Atomicity)、一致性 (Consistency)、隔离性 (Isolation)、持久性 (Durability) 兼容事务处理能力，以及独特的高性能和具有可扩展性的构架要素。另外，InnoDB 设计用于事务处理应用，这些应用需要处理崩溃恢复、参照完整性、高级别的用户并发数，以及响应时间超时服务水平合同。

其中，MySQL 5.5 最显著的增强性能是将 InnoDB 作为默认的存储引擎。在 MyISAM 以及其它表类型依然可用的情况下，用户无需更改配置，就可构建基于 InnoDB 的应用程序。我们已经对 InnoDB 进行了优化，能够充分利用现代化的硬件和操作系统资源。MySQL5.5 以及 InnoDB 新的优化、设置以及默认行为说明如下：

提高性能和可扩展性

MySQL5.5 引入了一种可重构 InnoDB，这种结构包含许多性能和扩展性特征。在 InnoDB 中增加以下增强性能，可延伸 MySQL5.5 的性能和可扩展性：

- **提高了 Windows 系统下的系统性能和可扩展性**——一直以来，MySQL 数据库一直能在基于 UNIX 的平台中良好地运行。现在，随着越来越多的开发者开始在 Windows 系统中构建和开发应用程序，MySQL 数据库开始逐渐从 Windows 平台下的桌面开发向产品数据中心转变。实际上，Windows 是目前应用最普遍的 MySQL 应用平台，MySQL 5.5 包含了诸多针对 Windows 平台的专用改进措施，这些措施增强了系统和应用的性能和可扩展性，可应对高并发数和用户负载。

MySQL5.5 针对 Windows 系统所作的改进主要包括：

- 现在，MySQL 使用 Windows 系统原生的同步基元实现互斥和加锁算法。
- 现在，MySQL 使用 Windows 原生的独立操作而不是代理线程，实现和释放读/写专用锁定。
- 现在，MySQL 默认使用 Windows 原生的操作系统的内存分配程序。
- 现在，Windows 系统中的 MySQL 的输入/输出线程最大处理能力已经与其它平台的最大处理能力相同。
- 其它平台上所作的既有优化措施现在已经移植到 Windows 平台中的 MySQL。
- 清理了许多 Windows 平台下特有的漏洞。
- **提高了默认线程并发数**——现在，通过利用多线程和多核系统的处理能力，InnoDB 默认可执行无限个并发执行线程。用户可以通过设置 `innodb_thread_concurrency` 的值，将默认并发数设置成针对具体应用的最佳值。
- **后台输入/输出线程控制**——用户现在可以使用两种新的针对所有平台的配置参数，即 `innodb_read_io_threads` 和 `innodb_write_io_threads`。这两种参数可以设置用于读和写请求的后台线程数，这有助于用户在高端、多核系统中调整和扩展他们的 MySQL 应用程序。
- **主线程输入/输出速率控制**——用户现在可以通过设置 `my.cnf` 或 `my.ini` 文件中的 `innodb_io_capacity` 参数，配置 InnoDB 运行后台任务的输入/输入容量和带宽。
- **操作系统内存分配程序使用控制**——用户现在可以控制，InnoDB 是使用自带的内存分配程序，还是使用当前部署的操作系统中现有的更高效的内存分配程序。通过在 MySQL 5.5 选项文件（`my.cnf` 或 `my.ini`）中设置新的系统配置参数 `innodb_use_sys_malloc`，可方便地进行控制。默认设置值为 1，表示 InnoDB 使用操作系统的内存分配程序。
- **适应性散列索引(Hash Index)控制**——用户现在可以关闭适应性散列索引功能，该功能内置在 InnoDB 中，默认启动。这个功能有助于我们调整应用程序或系统。潜在的适应性散列索引进程（如“SHOW ENGINE INNODB STATUS”输出表格所示）会造成 RW-latch 争用，进而堵塞很多进程，因此应该关闭该功能，提升系统性能。
- **插入缓冲 (Insert Buffering) 控制**——用户现在可以关闭 InnoDB 的插入缓冲功能（该功能可执行与工作数据集，或插入适合或几乎适合 InnoDB 缓冲池数据等操作有关的功能）。这样就能够防止 InnoDB 合并已经缓冲插入到 InnoDB 缓冲池中的数据，保留更多的缓冲池空间，保存大多数活动访问数据页面。
- **通过快速加锁算法提高可扩展性**——对于大多数平台来说（UNIX、Linux、Windows），InnoDB 现在不再使用代理(POSIX)线程，而是使用原生的独立操作来实现互斥和读/写锁定。这提高了 InnoDB 的性能和可扩展性，特别是在多核系统中。
- **恢复组提交 (Restored Group Commit)**——这种优化措施能够让 InnoDB 於一次性低階 I/O 作业执行一组提交（写入日志），而不是针对每一个提交分写进行刷新 (Flushing) 和同步操作，这可以极大地提高吞吐量。
- **提高恢复性能**——InnoDB 以其能够可靠地恢复崩溃的数据而著称。在之前的几个版本中，如果数据量较大，恢复时间（在下一次启动前需要应用进行扫描并应用重做日志）可能非常长，并且服务器重启的时间也很长。MySQL5.5 加入了许多默认的优化措施，能够加速下次启动前的扫描和重做日志应用，这样能够快速重启。之前因为恢复时间很长而人工将重做日志文件设置的很小的用户，现在可以增加日志文件的大小，无需担心。
- **多缓冲池实例**——现在，缓冲池的大小都是几个 GB 大小，各种不同的数据库线程可以保存以及不断地读取和更新数据页面。MySQL5.5 数据库系统消除了因为某个线程正在更新缓冲池而造成

其它线程必须等待的瓶颈。现在所有与缓冲池有关的结构都可以成倍增加，例如保护互斥结构、最新使用（LRU）信息以及刷新（Flush）列表等。用户现在可以控制和调整多少个缓冲池实例，但为了向后兼容，默认值仍然设置为 1。将几个存储容量为 GB 大小的缓冲池（每个缓冲池的存储容量都是 GB 大小或更大）合并起来使用能发挥该功能的最佳特性。

- **多个回滚段（Multiple Rollback Segments）**—InnoDB 现在可以使用多个回滚段来提升性能和可扩展性，并且能够极大地增加并发事务处理的数量。之前的几个 InnoDB 版本最多只能处理 1023 个并发事务处理操作，现在 MySQL 5.5 可以支持高达 128K 的并发事务处理操作，创建回滚数据（undo data）（来自插入、更新、和删除操作）。这种改进措施减少了在单个回滚段上的互斥争用，增加了吞吐量。

- **Linux 系统固有的异步输入/输出**—MySQL 5.5 数据库系统也提高了 Linux 系统的输入/输出请求并发数量。之前的几个版本中，InnoDB 已经通过内置输入/输出调用的方式为 Linux 系统提供了“模拟异步输入/输出”，但是事实是查询线程将会堵塞所有的竞争线程，直到输入/输出请求处理完成。现在 MySQL 5.5 为基于 Linux 和 Windows 的数据库系统提供了真实的异步输入/输出支持。如果新的默认设置与主机输入/输出子系统不兼容，要使用该功能，需要在 Linux 系统中安装 libaio 用户空间库，并配置 innodb_use_native_aio 选项参数（该参数可以关闭）。

- **扩展变化缓冲：添加了删除缓冲和清除缓冲**—与所有的数据库系统一样，InnoDB 通过使用索引列使查询和主键查找效率更高，性能更强。当对主键列进行插入、删除或更新时，二级索引或删除主键外其它列上的数据需要写入磁盘并始终保持最新。之前的几个 InnoDB 版本中引入了一种优化措施，当数据库因为插入（INSERT）操作而发生变化时，这种措施可以延迟磁盘写入，以便维护二级索引。这意味着 InnoDB 可以延迟或缓冲二级索引发生的变化，直到索引的内容在正常操作期间读入缓冲池，例如某个请求查询。然后在内存中快速更改，并使用正常的写脏数据块方法闪存到磁盘中。当缓冲池中的变化影响到一组连续的磁盘数据块组时，这些数据将会被闪存（Flush），这比按照数据变化的时间一条一条的写入要高效得多。同样，MySQL 5.5 还为删除操作扩展了同样的功能（首先是删除标记操作，然后使用收集/清除所有已删除记录的清除操作）。现在可以使用 innodb_change_buffering 配置参数来控制删除缓冲和既有插入缓冲功能，默认在所有的系统中都使用。

- **改善了日志系统互斥和单独刷新（Flush）列表互斥**—包括前面提到的缓冲池和刷新（Flush）列表在内的各种操作都受某个缓冲池互斥的保护，该互斥会形成争用和不必要的延迟。在 MySQL 5.5 中，刷新（Flush）列表拥有自己的互斥信号，从而减少了与其它缓冲池操作的争用。这是一种新增加的默认行为，无需配置即可使用。如果对多个缓冲池进行改进，效果会更好，因为每一个缓冲池实例都有一个单独的刷新（Flush）列表互斥，从而进一步减少了争用。

- **改善清除程序进度**—InnoDB 中的清除操作是一类定期回收无用数据的操作。在之前的几个版本中，清除操作是主线程的一部分，这意味着运行时它可能会堵塞其它的数据库操作。在 MySQL 5.5 中，该操作运行于独立的线程中，并支持更多的并发数。用户可通过设置 innodb_purge_threads 配置参数来选择清除操作是否使用单独线程，默认情况下参数设置为 0（不使用单独线程），设置为 1 时表示使用单独的清除线程。

- **改善事务处理中的元数据锁定**—在之前的几个版本中，当某个事务处理需要为语句中的表格使用元数据锁定时，会在语句的末尾释放该锁定信号。这种方法有一个缺点，如果一个处于活动事务处理中的其它会话正在使用的表格，产生一个数据定义语言（DDL）语句，该语句可能会按照错误的顺序写进二进制日志。MySQL 5.5 通过禁止会话执行另一个会话中，未完成事务处理正在使用的表格中的 DDL 语句，能够保证事务处理的串行化。通过在事务处理正在使用的表格中使用元数据锁定和延迟释放这些锁定，直到事务处理结束，就可以实现这个目的。这里，元数据锁定方法意味着某个会话中的某个事务处理正在使用的表格，在事务处理结束前，不能在另一个会话的 DDL 语句中使用。例如，表格 t1 被某个事务处理使用，如果另一个会话试图执行 DROP TABLE t1 操作，那么这个操作将会被阻塞，直到这个事务处理结束。这些改进以及各种优化措施（MySQL

如何管理表锁定）提高了 OLTP（联机事务处理）应用程序的性能，特别是那些需要频繁使用 DLL 语句的应用程序。

关于 MySQL 5.5 和 InnoDB 性能和扩展性增强的所有措施，包括如何启动和实现，请参见：

<http://dev.mysql.com/doc/refman/5.5/en/index.html>

<http://dev.mysql.com/doc/innodb-plugin/1.1/en/index.html>

性能和可扩展性基准

在 MySQL5.5 中，我们做了大量的工程性工作，并且重构了 InnoDB，旨在在各种异构平台，特别是在多处理器/多核、超线程构架中，为用户提供“透明”性能和可扩展性效益。为便于快速参考，下面是在 Oracle 的内置系统平台基准测试下，对 Linux 平台和 Windows 平台下的 MySQL5.5 版本和 5.1 版本所作的性能比较。这些基准包括在 5.1 版本中配置了可选的 InnoDB1.0 插件和默认内置的 InnoDB 引擎的实例。

MySQL 5.5 在 Linux 系统平台下的 SysBench 测试基准

根据情况，Linux 系统平台的 SysBench 基准根据以下要求配置服务器：

Intel Xeon x7460, x86_64

4 颗 CPU，每颗 CPU 有 6 个核

主频 2,86 GHz、32 GB 内存

操作系统：Fedora 10

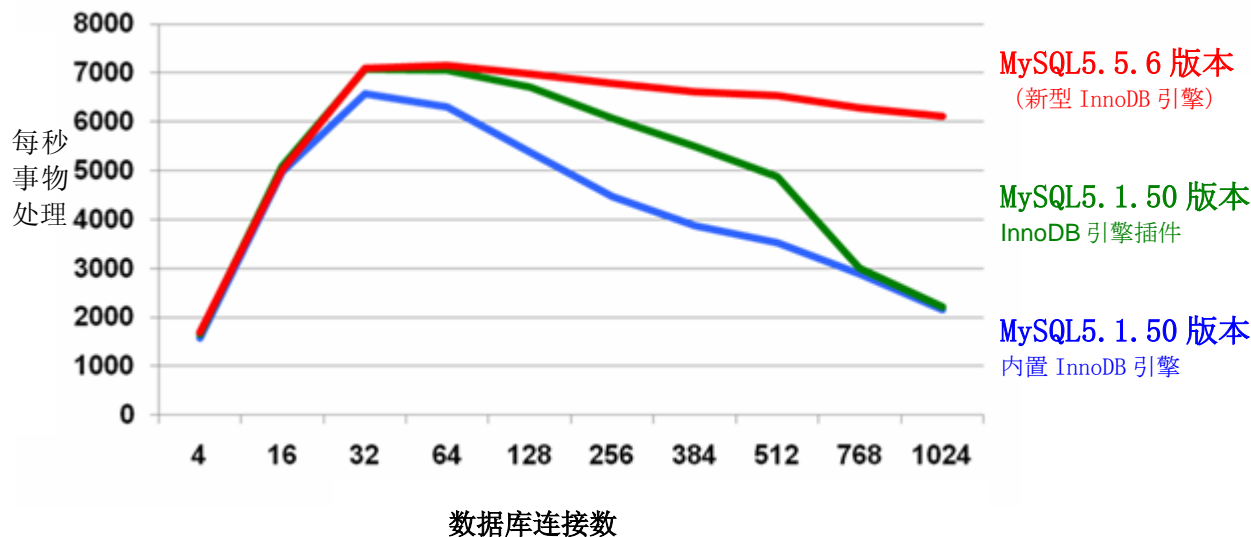


图 1 Linux 系统下 SysBench 基准 – 只读 (Read Only)

图 1 显示了 Linux 系统中，在高事务处理频率和连接负载的情况下，MySQL5.5 在处理只读 (Read Only) 操作时的性能和可扩展性是 MySQL5.1 的 200%。新的并发数改进措施使 MySQL5.5 在高事务处理频率和用户连接负载下，仍然能够保持很高的性能，在物理服务器资源已经达到饱和的情况下，应用程序仍然能够响应各种事件。

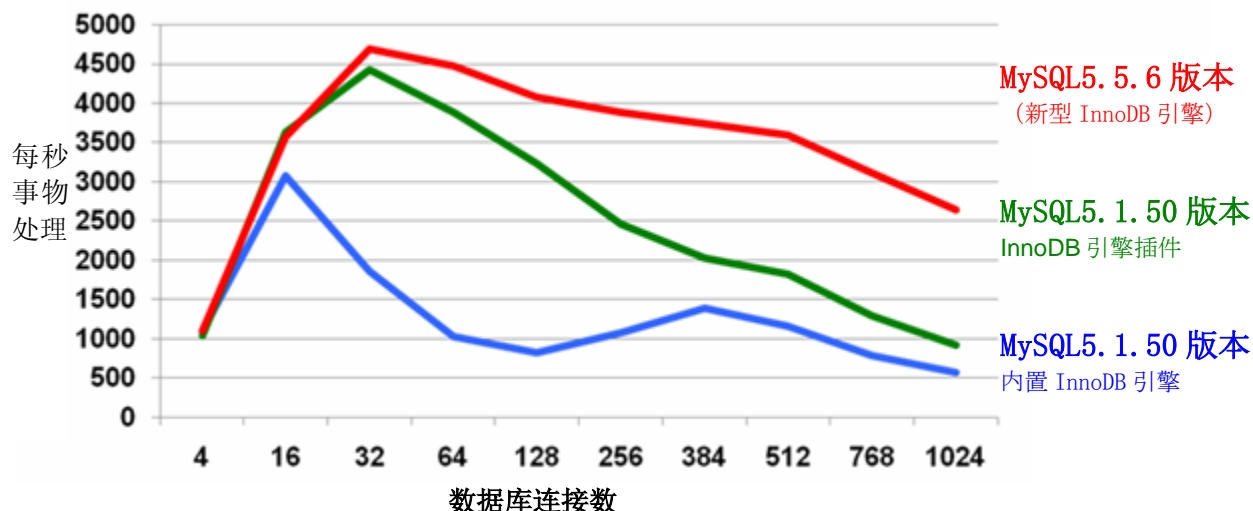


图2 Linux 系统下 SysBench 基准-读/写 (Read/Write)

图 2 显示了 Linux 系统中，在高事务处理频率和连接负载的情况下，MySQL5.5 在处理读/写操作时的性能和可扩展性是 MySQL5.1 的 370%。虽然在高事务处理频率和连接负载的情况下，MySQL 5.5 的性能有所下降，但新的并发数改进措施使 MySQL5.5 在物理服务器资源已经达到饱和的情况下，仍然保持高于 MySQL5.1 之前版本的性能。

MySQL 5.5 在 Windows 系统平台下的 SysBench 测试基准

根据情况，Windows 系统平台的 SysBench 基准根据以下要求配置服务器：

Intel x86_64

4 颗 CPU，每颗 CPU 有 2 个核

主频 3.166 GHz、8 GB 内存

操作系统：Windows Server 2008

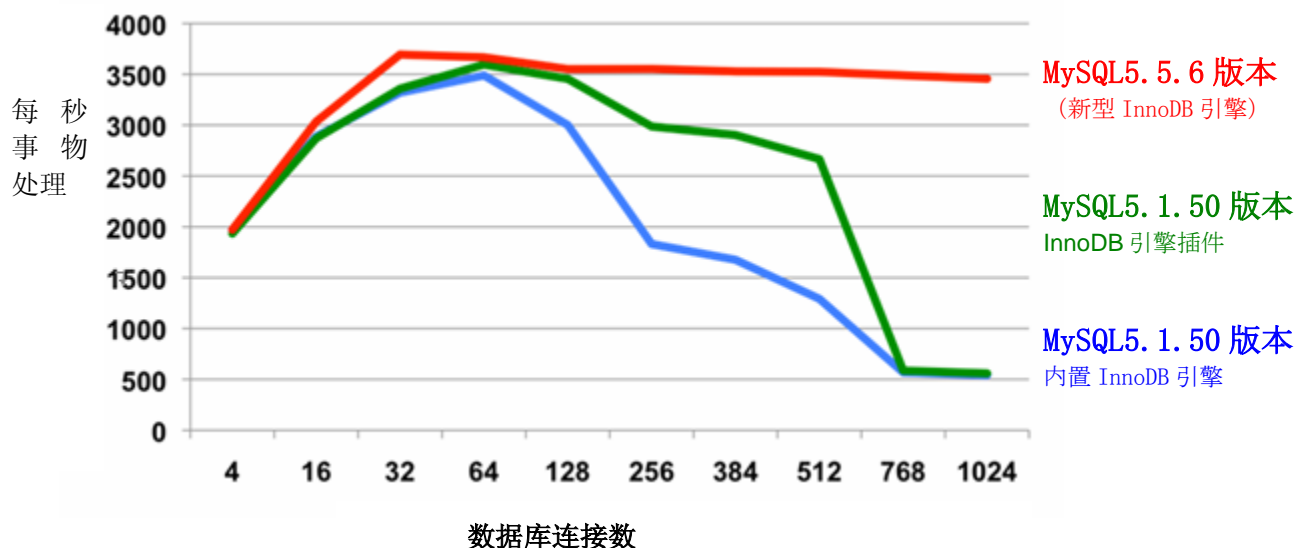


图3 Windows 系统下 SysBench 基准-读/写 (Read/Write)

图 3 显示了 Windows 系统中，在高事务处理频率和连接负载的情况下，MySQL5.5 在处理只读 (Read Only) 操作时的性能和可扩展性是 MySQL5.1 的 540%。新的 Windows 专用并发数改进措施使 MySQL5.5 在高事务处理频率和用户连接负载的情况下，仍然能够保持很高的性能，在物理服务器资源已经达到饱和的情况下，应用程序仍然能够响应各种事件。

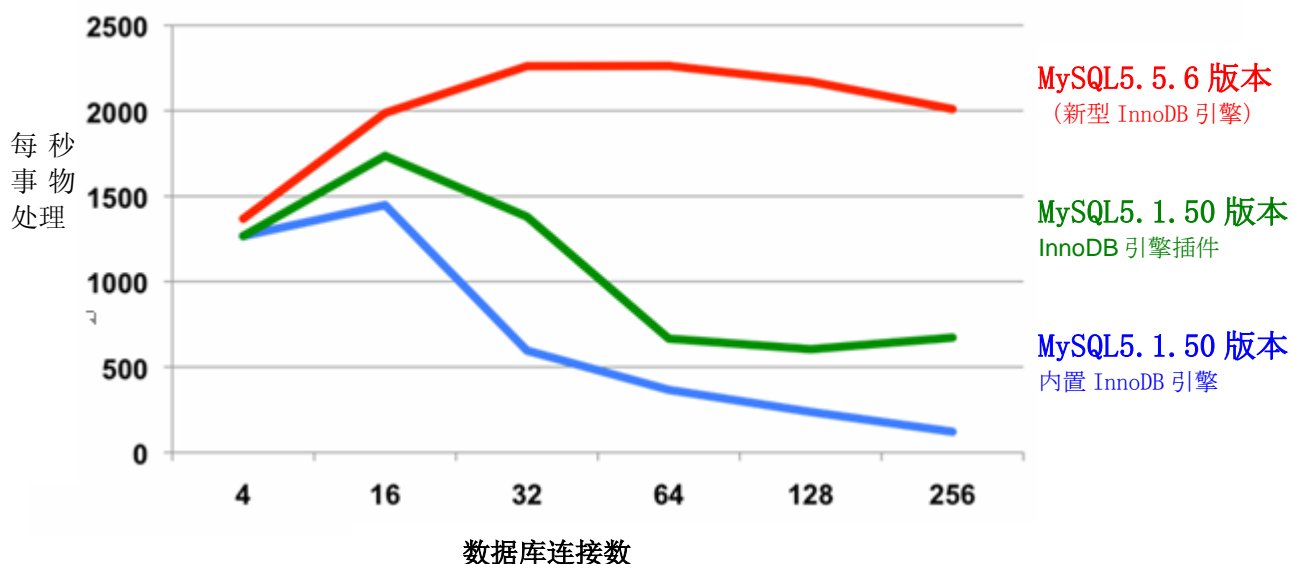


图4 Windows 系统下 SysBench 基准-读/写 (Read/Write)

图 4 显示了 Windows 系统中，在高事务处理频率和连接负载的情况下，MySQL5.5 在处理读/写操作时的性能和可扩展性是 MySQL5.1 的 1500%。新的 Windows 专用并发数改进措施使 MySQL5.5 在高事务处理频率和用户连接负载的情况下，仍然能够保持很高的性能，在物理服务器资源已经达到饱和的情况下，应用程序仍然能够响应各种事件。应该注意到 Windows 系统下的 MySQL5.5 数据库系统已经针对大多数基于 Windows 的应用程序部署使用案例进行了优化。过去，这些使用案例使用的都是 Microsoft SQL Server 数据库系统。

提高实用性

MySQL 数据库系统的复制 (Replication) 功能是使用广泛且最受欢迎的功能，因为它具有可扩展性，并为数据冗余和实用性提供了便捷、有力的解决方案。在 MySQL5.5 中，我们根据用户的要求，对复制功能进行了增强，从而提供更好的实用性：

- ☐ 确保主服务器和从服务器之间数据的一致性
- ☐ 如果复制不能正常工作，立即检测
- ☐ 允许崩溃的从服务器依据主服务器的中继日志 (Relay Log) 自动恢复
- ☐ 允许用户为特定的服务器过滤各种事件
- ☐ 正确转换主服务器和从服务器之间的数据类型

MySQL5.5 复制功能包括以下几个增强措施，可在关键操作给予用户支持：

半同步复制 (Semi-synchronous Replication)

默认情况下，MySQL5.5 的复制功能是异步的，这意味着当谈到数据一致性时，主服务器及其从服务器是独立的。异步复制可以提供最佳的性能，因为主服务器在将更新的数据写入它的二进制日志 (Binlog) 文件中后，无需等待验证更新数据是否已经复制到至少一台拓扑从服务器中，就可以自由处理其它进入的事务处理请求。虽然快，但这也同时带来了很高的风险，如果在主服务器或从服务器端发生故障，会造成主服务器/从服务器数据的不一致，甚至在恢复时造成数据丢失。

MySQL5.5 引入了一种半同步复制功能，该功能可以确保主服务器和访问链中至少一台从服务器之间的数据一致性和冗余。在这种配置结构中，一台主服务器和其许多从服务器都进行了配置，这样在复制拓扑中，至少有一台从服务器在父主服务器进行事务处理前，必须确认更新已经收到并写入了其中继日志 (Relay Log)。当出现超时，源主服务器必须暂时切换到异步复制模式重新复制，直到至少有一台设置为半同步复制模式的从服务器及时收到信息。

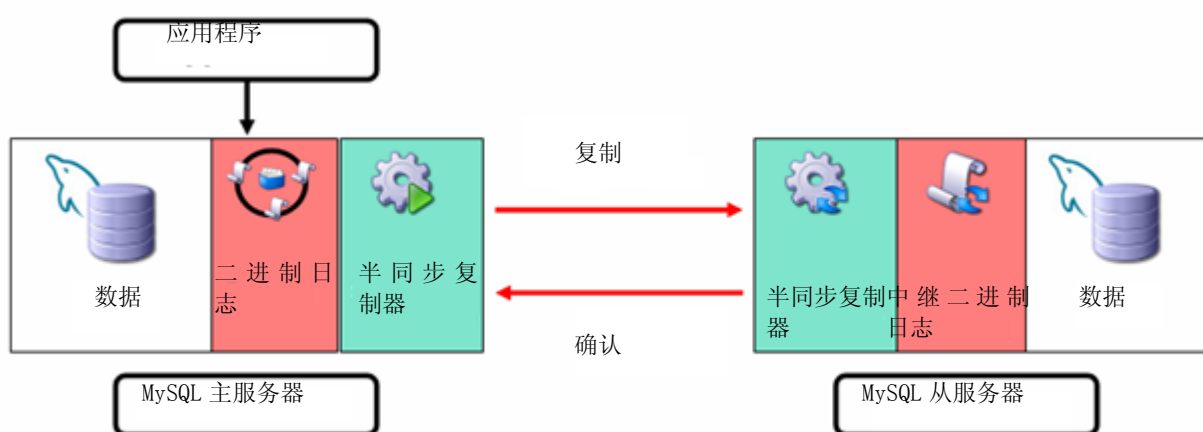


图5 半同步复制模式

半同步复制模式必须在主服务器和从服务器端同时启用，否则主服务器默认使用异步复制模式。MySQL5.5 使用一种全新的插件构架实现了半同步复制模式。为此，请使用下面的命令和变量设置启动 MySQL5.5 主服务器和从服务器。静态设置信息也可添加到“my.*”配置文件中：

启动主服务器的半同步复制功能：

```
INSTALL PLUGIN 'rpl_semi_sync_master' SONAME 'semisync_master.so';
SET rpl_semi_sync_master_enabled=1;
SET rpl_semi_sync_master_timeout=1000; (1 秒, 默认为 10 秒)
```

启动从服务器或多个从服务器的半同步复制功能：

```
INSTALL PLUGIN 'rpl_semi_sync_slave' SONAME 'semisync_slave.so';
SET rpl_semi_sync_slave_enabled=1;
START SLAVE;
```

一旦启动半同步复制模式，将会显示出新系统及其状态变量，这些变量可用于检查配置和运行状态。每个值使用 **SHOW VARIABLES** 和 **SHOW STATUS** 显示，包括：

在主服务器中：

- **Rpl_semi_sync_master_status** –用于指示主服务器使用的是异步复制模式，还是半同步复制模式

- Rpl_semi_sync_master_clients –用于显示有多少个从服务器配置成了半同步复制模式
- Rpl_semi_sync_master_yes_tx – 显示从服务器确认的成功提交数量
- Rpl_semi_sync_master_no_tx – 显示从服务器确认的不成功提交数量

在从服务器中：

- Rpl_semi_sync_slave_status –用于指示从服务器是否启动半同步复制模式

复制 Heartbeat

MySQL5.5 提供了一种全新的复制 **Heartbeat** 选项，当复制功能停止工作时，该选项能够帮助用户立刻察觉问题。**Heartbeat** 是一种定期从主服务器节点发送到从服务器节点的消息。可将从服务器配置成自动检查连接和消息的状态；如果从服务器没有接收到该消息，那么从服务器应该知道，与主服务器之间的节点连接出现了故障。

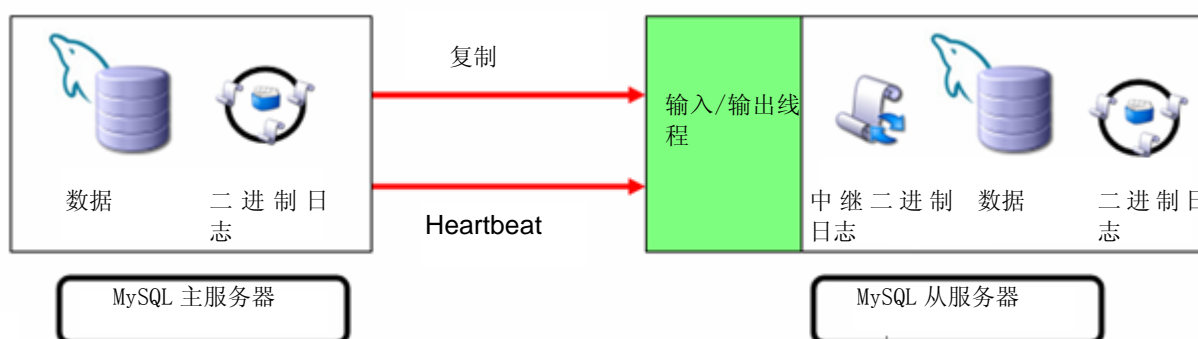


图6 复制 Heartbeat

复制 **Heartbeat** 是一种可选配置，使用下列命令在 MySQL5.5 从服务器上启用：

```
STOP SLAVE;
CHANGE MASTER TO master_heartbeat_period= milliseconds;
START SLAVE;
```

监控下面的状态变量，可以很容易检测到主服务器是否空闲，并瞬间获得从服务器的细粒度（Finer-Grained）估计，以用于恢复：

```
SHOW STATUS like 'slave_heartbeat period'
SHOW STATUS like 'slave_received_heartbeats'
```

中继日志自动恢复（Automatic Relay Log Recovery）

在启动时，MySQL5.5 允许用户随意配置从服务器，自动丢弃自己未处理的中继日志（Relay Log），然后从源主机服务器恢复挂起的事务处理，从而确保了主服务器/从服务器的数据一致性。这种方法可以确保当从服务器崩溃后，潜在可能崩溃的备份日志不被处理。出于兼容性考虑，默认情况下该功能是关闭的，将 relay_log_recovery 的值设置为 1 时，可在从服务器上开启该功能。

根据服务器过滤项复制（Replication Per Server Filtering）

循环或多主服务器复制，提供了一种可用度非常高的部署方案，该方案可在拓扑环中的任何服务器发生故障或移走的情况下，确保数据的冗余。在这种配置方案中，需要对拓扑环中所有主服务器进

行配置，使每个主服务器同时也是一个从服务器。写入任何主服务器的更新信息会在拓扑环中的主机中依次进行复制，直到重新回到源服务器（这个服务器是自己事件消息的终止器）。当某个节点发生故障时，将受影响的服务器从拓扑环中移除，然后简单地将拓扑环中的另外一台服务器指定为它的从服务器，并继续处理。在这里我们说明一下：

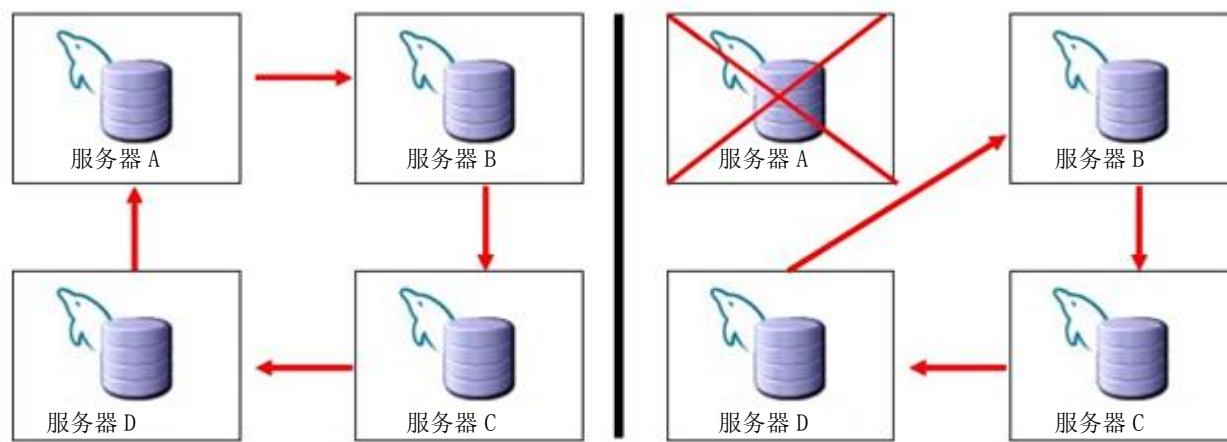


图7 多主服务器复制和故障恢复

在之前的几个版本中，当某个服务器由于故障、维护等原因从拓扑环中移除时，用户需要手动确保在新的呼叫链中终止所有的更新信息。MySQL5.5 提供了一组新的省时的命令，用户使用这组命令很容易就可以过滤出与已移除服务器相关的事件。在上述例子中，当服务器 A 从拓扑环中移除后，用户现在可以在呼叫链中的下一台服务器中输入下面的命令，即可过滤出任何与服务器 A 相关的事件：

```
Server B> CHANGE MASTER TO MASTER_HOST=D ...  
IGNORE_SERVER_IDS=(A)
```

从服务器复制支持的数据类型转换（Replication Slave Side Data Type Conversions）

在 MySQL5.1 中，主服务器和从服务器间精确的数据类型转换仅被基于声明的复制支持。在这种配置中，只要底层数据有较好的兼容性（例如，从 INT 型转换到 TINYIN 类型），主服务器和从服务器中表类型可以不同。现在 MySQL5.5 在主服务器和从服务器之间提供了一种精确的数据类型转换机制，受基于声明的操作和基于行的操作支持。支持整数、小数、字符串、二进制、BIT、ENUM（枚举类型）和 SET（集合）域之间的转换。

在 MySQL5.5 中，新的 SET 类型变量可以启动变换操作，但需要重新启动从服务器后才可以生效。设置以及可以启动的变换如下：

SET SLAVE_TYPE_CONVERSIONS="ALL_LOSSY" -向较小域类型的转换（例如，INT 转换为 TINY）

SET SLAVE_TYPE_CONVERSION="ALL_NON_LOSSY" -向较大域类型的转换（例如 TINY 转换为 INT）

要想了解更多关于 MySQL5.5 复制功能的信息，包括如何启动、监控和调整，请参见：

<http://dev.mysql.com/doc/refman/5.5/en/replication.html>.

提高易管理性和效率

MySQL5.5 提供一整套更加完善的内部 InnoDB 设置控制方法，所有诸如性能、扩展、存储都可以针对具体的使用案例和应用负载轻松地监控、调整和优化。这些先进的特性和功能主要包括：

- **建立快速索引（Faster Index Creation）** – 现在，MySQL5.5 可以添加（Add）或删除（Drop）索引，而无需复制整个目标表中的底层数据。该功能提高了创建和删除 InnoDB 表中索引的效率和速度。
- **高效的数据压缩（Efficient Data Compression）** – 新的 InnoDB 表压缩选项极大地减少了 MySQL 数据库的存储空间要求，并通过减少输入/输出工作量、各种操作费用，提高了应用程序的吞吐量。用户可使用 CREATE TABLE 和 ALTER TABLE 设置新的 ROW_FORMAT 和 KEY_BLOCK_SIZE 参数，按照 1K、2K、4K、8K 或标准 16K 字节压缩等级存储数据页。另外，InnoDB 还根据压缩要求提供了新的 INFORMATION_SCHEMA 表格，用户可以进行监控和调整，从而实现最佳效率。
- **为大物件和可变长度列提供高效存储** – MySQL 5.5 允许用户使用 ROW_FORMAT=DYNAMIC 或 ROW_FORMAT=COMPRESSED 创建表，这提高了大对象的存储效率。使用这些选项参数，长列值完全可以跨页存储，并且与簇索引相关的记录仅包含 20 个字节长度的指标，指向溢出页。
- **新的 INFORMATION_SCHEMA 表** – 7 个新的表提供了与 InnoDB 压缩和事务处理锁定有关的具体信息：
 - INNODB_CMP – 包含与压缩表相关的操作和效率的状态信息
 - INNODB_CMP_RESET – 同上，但是对该表的读取操作会重置报告的统计信息
 - INNODB_CMPMEM – 包含驻留在缓冲池中压缩页的状态信息
 - INNODB_CMPMEM_RESET – 同上，但是对该表的读取操作会重置报告的统计信息
 - INNODB_TRX – 包含 InnoDB 内当前执行的每个事务处理的信息，包括事务处理是否在等待某个锁定，事务处理何时开始，以及事务处理正在执行的具体 SQL 语句
 - INNODB_LOCKS – 包含一行关于每一个被堵塞的事务处理的信息，该列描述了事务处理请求的每个锁定，以及事务处理正在等待什么信号
 - INNODB_LOCK_WAITS – 包含某个正在等待具体锁定的事务处理的相关信息。

要想了解更多关于 MySQL5.5 和 InnoDB 性能和可扩展性增强的措施，包括如何启动和实现，请参见：

<http://dev.mysql.com/doc/refman/5.5/en/index.html>

<http://dev.mysql.com/doc/innodb-plugin/1.1/en/index.html>

提高可用性

MySQL5.5 扩展了在早前的版本中已经引入的存储对象、表/索引分区功能的可用性，这些先进特性和功能包括：

针对 SIGNAL/RESIGNAL 的新 SQL 语法

MySQL5.0 数据库系统引入了存储过程、触发器和视图，因此开发者可以封装和重用通用的应用程序代码组件。MySQL5.5 数据库对这些流行的功能进行了扩展，包括支持 ANSI/ISO 标准 SIGNAL/RESIGNAL 语法。开发者现在可以在存储对象（包括存储过程、函数、触发器）内部使用 SIGNAL/RESIGNAL 语法，用于项目的前期研发和调试，然后产生错误条件，并使用该条件调用主机应用程序中的专用错误处理程序。例如，SIGNAL 语法是如何在存储过程内部起作用的：

```
CREATE PROCEDURE p (pval INT)
```

```
BEGIN
DECLARE specialty CONDITION FOR SQLSTATE '45000';
IF pval = 0 THEN
    SIGNAL SQLSTATE '01000';
ELSEIF pval = 1 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'An error occurred';
ELSEIF pval = 2 THEN
    SIGNAL specialty
    SET MESSAGE_TEXT = 'An error occurred';
ELSE
    SIGNAL SQLSTATE '01000'
    SET MESSAGE_TEXT = 'A warning occurred', MYSQL_ERRNO = 1000;
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'An error occurred', MYSQL_ERRNO = 1001;
END IF;
END;
```

在这个简单的例子中，如果 `pval` 为 0，`p()` 会发出一个警告信息，因为 `SQLSTATE` 值以 '01' 开头，而该值正好处于警告级别。警告信息并不会终止存储过程，但在存储过程返回后，主机应用程序可使用 `SHOW WARNINGS` 查看该信息。

如果 `pval` 的值为 1，`p()` 会发出一个错误信号，并设置 `MESSAGE_TEXT` 条件信息项。错误信息会终止存储过程，并返回带有错误信息的文本。

如果 `pval` 的值为 2，尽管 `SQLSTATE` 的值已经使用本例中命名的条件进行了指定，但同样还会发送错误信号。

如果 `pval` 为其它值，`p()` 会先发送一个警告信号，并设置消息文本以及错误编号条件信息项。该警告信息不会终止存储过程，所以存储过程会继续执行；然后 `p()` 发送一个错误信号，错误信号会终止存储过程。由警告信息设置的消息文本和错误编号被错误信号设置的值替换，并返回错误信息。

新的 `RESIGNAL` 语法会传递错误条件信息，在存储过程或函数、触发器或事件的内部的复合语句中执行条件处理器（Handler）时可以使用该信息。在传递之前，`RESIGNAL` 可能会改变某些或全部信息，从而使其既能处理错误，又能返回错误信息。另外，执行处理器中的 `SQL` 语句，会破坏激活处理器（Handler）的信息。如果给定的处理器（Handler）能够处理一部分情况，`RESIGNAL` 也可使某些存储程序缩短，然后将“上线（up the line）”条件传递给另外一个存储过程。使用 `RESIGNAL` 和 `SIGNAL` 语法产生错误条件的简单示例，如下：

```
DROP TABLE IF EXISTS xx;
delimiter //
CREATE PROCEDURE p ()
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        SET @error_count = @error_count + 1;
        IF @a = 0 THEN RESIGNAL SET MYSQL_ERRNO = 5; END IF;
    END;
    DROP TABLE xx;
END//
delimiter ;
SET @error_count = 0;
SET @a = 0;
CALL p();
```

要想了解更多关于 MySQL 5.5 中 `SIGNAL/RESIGNAL` 语法的信息，请参见：

<http://dev.mysql.com/doc/refman/5.5/en/signal-resignal.html>.

新的表/索引分区选项

MySQL5.5 将表和索引 RANG 和 LIST 分区范围扩展到了非整数列和日期，并增加了在多个列上分区的能力。COLUMNS 关键字支持这些选项。如何将该功能应用到非整数列的简单示例，如下：

```
CREATE TABLE expenses (
  expense_date DATE NOT NULL,
  category VARCHAR(30),
  amount DECIMAL (10,3)
);
```

之前，该表只能使用整数值进行分区，所以当因为快速查找而需要按类别进行分区时，需要将类别列转换成一个交叉引用的整数，并另外创建一个查询表。在 MySQL5.5 中，只需使用新的分区选项即可轻松完成该过程：

```
ALTER TABLE expenses
PARTITION BY LIST COLUMNS (category)
(
  PARTITION p01 VALUES IN ( 'lodging', 'food'),
  PARTITION p02 VALUES IN ( 'flights', 'ground transportation'),
  PARTITION p03 VALUES IN ( 'leisure', 'customer entertainment'),
  PARTITION p04 VALUES IN ( 'communications'),
  PARTITION p05 VALUES IN ( 'fees')
);
```

该功能如何应用于日期分区的示例，如下：

```
CREATE TABLE t2
(
  dt DATE
)
PARTITION BY RANGE COLUMNS (dt)
(
  PARTITION p01 VALUES LESS THAN ('2007-01-01'),
  PARTITION p02 VALUES LESS THAN ('2008-01-01'),
  PARTITION p03 VALUES LESS THAN ('2009-01-01'),
  PARTITION p04 VALUES LESS THAN (MAXVALUE));
```

最后，一个关于多列分区的例子，如下：

```
CREATE TABLE employees (
  first_name varchar(14) NOT NULL,
  last_name varchar(16) NOT NULL,
  gender char(1) DEFAULT NULL,
  hire_date date NOT NULL
)
PARTITION BY RANGE COLUMNS(gender,hire_date)
(PARTITION p01 VALUES LESS THAN ('F','1990-01-01'),
PARTITION p02 VALUES LESS THAN ('F','2000-01-01'),
PARTITION p03 VALUES LESS THAN ('F',MAXVALUE),
PARTITION p04 VALUES LESS THAN ('M','1990-01-01'),
PARTITION p05 VALUES LESS THAN ('M','2000-01-01'),
PARTITION p06 VALUES LESS THAN ('M',MAXVALUE),
PARTITION p07 VALUES LESS THAN (MAXVALUE,MAXVALUE))
```


在之前的各种版本中，可通过 **INFORMATION_SCHEMA** 按如下方式检查分区分布：

```
SELECT
    partition_name part,
    partition_expression expr,
    partition_description descr,
    table_rows
FROM
    INFORMATION_SCHEMA.partitions
WHERE
    TABLE_SCHEMA = schema()
    AND TABLE_NAME='employees';
```

| part | expr | descr | table_rows |
|------|------------------|-------------------|------------|
| p01 | gender,hire_date | 'F','1990-01-01' | 66212 |
| p02 | gender,hire_date | 'F','2000-01-01' | 53832 |
| p03 | gender,hire_date | 'F',MAXVALUE | 7 |
| p04 | gender,hire_date | 'M','1990-01-01' | 98585 |
| p05 | gender,hire_date | 'M','2000-01-01' | 81382 |
| p06 | gender,hire_date | 'M',MAXVALUE | 6 |
| p07 | gender,hire_date | MAXVALUE,MAXVALUE | 0 |

MySQL5.5 也通过下面的新可用性功能提高了分区的易用性：

TRUNCATE PARTITION – 与以前的 **DROP PARTITION** 工作方式类似，但是它会在适当的位置放置一个现存的分区，因此可使用新的或重定义的分区值对其快速填充。它扩展了现有的切割选项，这样可以方便地删除（Drop）分区 或简单清空后再重新使用。

TO_SECONDS – 允许用户将 **DATE** 和 **DATETIME** 列转换成秒，这样分区可以在小于 1 天的时间间隔内建立。

更多关于 MySQL5.5 分区和现有新选项的信息，请参见：

<http://dev.mysql.com/doc/refman/5.5/en/partitioning.html>.

改善检测和诊断

新性能构架

MySQL5.5 引入了一种新的性能构架（**PERFORMANCE_SCHEMA**, **P_S**），用于监控 MySQL 服务器运行时的性能。与既有的 **INFORMATION_SCHEMA**（信息构架，该构架用于记录和分析 MySQL 数据库元数据）不同，新的 **P_S** 构架是一种新的存储引擎和基本数据库，用于对与数据和度量相关的目录性能进行当前和历史分析。

可以监控底层服务器事件或任何耗费时间或资源的服务器操作。MySQL5.5 的事件采集和监控具有以下特征：

- **P_S** 存储引擎使用用户在服务器源代码中设置的“检测点”采集事件数据
- 采集的数据存储在新 **performance_schema** 数据库的表中，可以使用 **SELECT** 语句向查询其它表一样查询这些表。通过使用 **SQL** 语句更新 **performance_schema** 数据库中的表可以动态修改 **P_S** 的配置。配置的变化会立刻影响数据采集。另外，**performance_schema** 数据库中的表是各种视图或一

些不会持久存储在磁盘中的临时表。

- **P_S 事件**与写入服务器二进制日志文件（描述数据修改）中的事件以及事件调度事件有本质的不同。
- 当前 **P_S 事件**、事件历史以及摘要都可用。因此用户能够确定检测活动执行了多少次，以及这些活动花费了多少时间。事件信息可以显示特殊线程、锁定/互斥的活动，以及与特殊文件和对象有关的活动。
- **P_S 检测**在所有 MySQL 支持的平台上都可以使用，但是定时器和检测设备的类型可能不被所有的平台和所有的存储引擎都支持或实现。

激活 **P_S** 不会对服务器的行为、性能或吞吐量造成任何变化。不像其它的服务器功能，与 **P_S** 相关的线程调度或执行计划没有单独的线程，也不会发生变化。另外，除了服务器启动时需要分配内存外，其它情况下不需要分配内存。基于可靠性的考虑，即使 **P_S** 采集发生故障，服务器代码仍然会正常继续执行，并且这时可以选择是在事件采集期间继续执行进程或事件返回后继续执行进程，一旦选定，将会给予优先权并快速分配内存。这是因为采集操作正在进行，同时还需要返回事件的情况永远不会发生。最后，我们对检测（Instrumentation）进行了改写以便向后兼容，所以检测的实现方法改变了，但是之前的检测代码仍然继续有效。

更多关于 MySQL 性能架构，包括如何启动以及如何检测 MySQL 服务器代码的信息，请参见：

<http://dev.mysql.com/doc/refman/5.5/en/performance-schema.html>.

MySQL 5.5:生产准备软件与支持

MySQL 数据库系统是当今在用的最流行的开源数据库，并且在通用公共许可证（GPL）下仍然可用。为了 MySQL 的新用户和高级用户能够构建关键业务应用程序，MySQL5.5 完全可用于产品研发、质量保证（QA）和生产，并且在 Oracle 世界级的软件和服务的支持下，您完全可以满怀信心地部署该数据库系统。MySQL5.5 提供了多种商业授权模式，允许用户开发、部署、监控和备份最安全、最新版本的 MySQL 数据库系统，所有的版本都有 Oracle 公司全天候技术支持。MySQL 商业授权软件和服务包括：

- **MySQL 数据库**—最安全、最新的 MySQL 数据库版本，能够应用于要求最苛刻的在线服务、Web 服务以及 OLTP 应用程序。MySQL 商用服务器全面集成了安全事务处理、支持完全提交的 ACID 兼容数据库、回滚、崩溃恢复以及行级锁定功能。MySQL 数据库全面支持 Linux、UNIX、MacOSX 和 Windows 系统平台，以其卓越的性能、可靠性和易用性享誉世界。
- **MySQL 企业级备份**—可以实现 MySQL 数据库非阻塞在线“Hot”备份。同时，MySQL 企业级备份还全面支持压缩增量备份、部分备份以及及时点恢复（Point-in-time Recovery）
- **MySQL 企业级监控器和查询分析器**—能够对 MySQL 数据库系统进行持续的监控，当出现故障时，在故障影响到关键系统或应用程序之前，能够主动通知数据库管理员可能出现的潜在问题，并提出查询和调整的方法。监控器提供 MySQL 专家咨询，能够对设置和调整 MySQL 配置和变量信息提供深入、详细的指导，以优化安全性、性能和可用性。开发者可以使用内置的高级查询分析器直观地查找和调整耗费较大的代码，而无需再使用 Slow Query Log、SHOW PROCESS LIST 或其它工作量较大的方法。
- **MySQL 工作台**—MySQL 工作台为数据库构架师、开发者、数据库管理员提供了一种基于图形用户界面的数据模型、SQL 开发、部署和全面的管理工具（服务器配置、用户管理、对象管理）。
- **Oracle 公司为 MySQL 提供全生命周期服务**—通过 MySQL 技术支持服务，可直接与专家级 MySQL 支持工程师沟通，工程师时刻准备帮助用户开发、部署和管理 MySQL 应用程序。MySQL 的支持团队由经验丰富的 MySQL 开发人员和数据专家组成，他们能够解决用户所面临的各种问

题和挑战，因为他们曾经克服过这些挑战。

要想了解更多关于商业 MySQL 软件和支持服务的信息，请参见：

<http://mysql.com/products>

<http://mysql.com/support/>

<http://mysql.com/training/>

<http://mysql.com/consulting/>

结论

MySQL 性能出众、可靠性高、方便使用，是当今世界最流行、使用最广泛的开源数据库。在此基础上，MySQL5.5 引入了“透明”性能和可扩展性效益，可全面利用各系统平台下的现代化、多处理构架计算能力。尤其是在 Windows 平台下，MySQL5.5 的性能和可扩展性的提升最为显著，可以与其它支持的平台相比肩（或更好）。MySQL5.5 最关键的改进在于重构了 InnoDB 引擎，将其作为 MySQL 数据库系统的默认存储引擎。另外，MySQL5.5 还提供了新的复制监控和易管理功能，这些功能允许用户立刻察觉复制操作何时停止、何时出现问题，从而便捷地恢复，并使主服务器和从服务器同步。新的 InnoDB 表存储和压缩选项有助于减少 MySQL 数据足迹（Data Footprint），并使数据检索效率更高。对于开发者而言，MySQL5.5 提供了新的 SQL 语法和分区选项，使应用程序的开发、调试和调整更加容易。最后，新的性能构架让数据库管理员（DBAs）和开发者对 MySQL 数据库的性能有了一个基本的了解，所以他们能够在 MySQL 源代码级别上清楚地了解在什么地方他们的应用程序花费了宝贵的时间和资源。这些功能，以及 Oracle 在硬件、软件和服务上的付出，使得 MySQL5.5 是以往版本中最好的一个发行版本。

更多信息

更多关于 Oracle 的 MySQL 产品信息和服务，请参见：

<http://www.mysql.com/products/>.

MySQL 客户和案例研究

<http://www.mysql.com/customers>

MySQL 企业版

<http://mysql.com/products/enterprise/>

MySQL 企业级备份

<http://www.mysql.com/products/enterprise/backup.html>

MySQL 企业级监控和顾问

<http://www.mysql.com/products/enterprise/monitor.html>

MySQL 查询分析器

<http://www.mysql.com/products/enterprise/query.html>

MySQL 高实用性

http://www.mysql.com/products/enterprise/high_availability.html

MySQL 专业服务和咨询

<http://mysql.com/consulting/>

请直接联系甲骨文（免费电话）

中国：800 810 0161

香港：800 901 039

台湾：0800 672 253