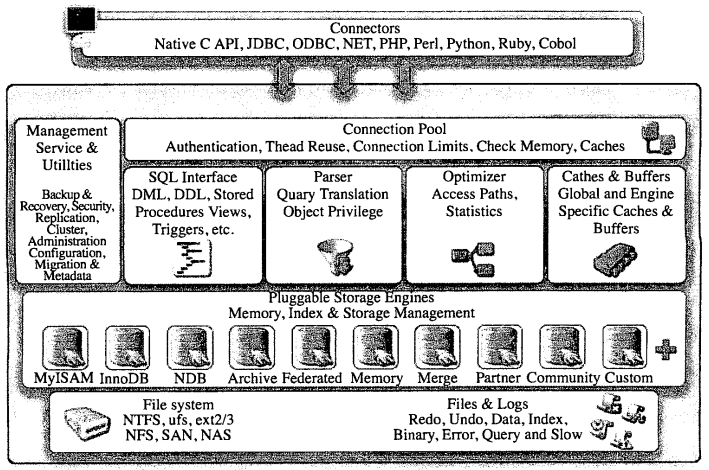
mysql 体系结构：

单进程多线程



从上往下，从左到右，依次是：开放给各编程语言的connectors，管理服务和工具，连接池，SQL接口，查询分析器，优化器，缓冲组件，插件式引擎，物理文件。

记住：引擎是基于表而不是基于数据库的，即同一个数据库中的表可以使用不同的引擎。

InnoDB存储引擎的特点：

主要面向OLTP应用，行锁设计，支持外键，提供非锁定读(和oracle类似)。

使用多版本控制(MVCC)来获得高并发性，实现sql标准的四种隔离级别，默认为repeatable。使用next-key locking来避免幻读(phantom)。此外，还提供插入缓冲(insert buffer)，二次写（double write），自适应哈希索引（adaptive hash index），预读（read ahead）等高性能功能。

使用聚簇存储方式（类似于oracle的聚簇索引表）。

InnoDB体系结构：

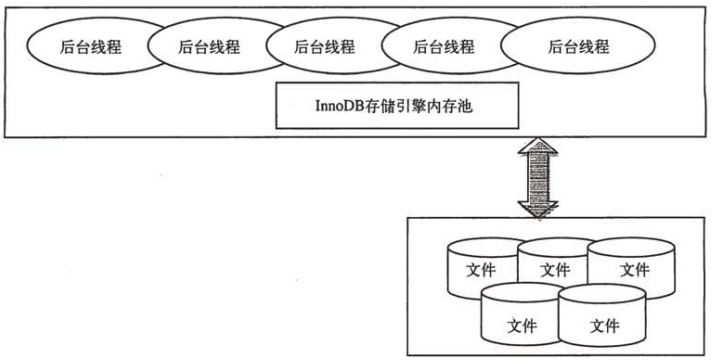
默认情况下，InnoDB的后台线程有：4个IO thread（分别是insert buffer thread，log thread，read thread和write thread，），1个master thread，1个锁监控线程，1个错误监控线程。

InnoDB内存块分为：缓冲池（buffer pool），重做日志缓冲池（redo log buffer）以及额外的内存池（additional memory pool）。

缓冲池存放各种数据的缓存，按页读取（每页16k大小），然后使用LRU算法来保留存放在该池中的数据。按照一定的频率将脏页（修改过的页）刷新（flush）到文件。

缓冲池存放的数据页类型有：索引页，数据页，undo页，插入缓冲，自适应哈希索引，InnoDB存储的锁信息，数据字典信息等。

额外的内存池用于对某些数据结构本身分配内存。



插入缓冲：对于B\*树索引，插入和修改操作的性能必然会被限制。使用了插入缓冲后，假如该索引页已在缓冲区，则立即插入，如果不在，则先放入一个插入缓冲区，再以一定的频率将多个操作合并刷出到文件。注意，索引不能是唯一索引，也不能是主键索引。

两次写：保证数据可靠性。

自适应哈希索引：并不是对表建立（表上没有这种索引），而是通过监测查询频率对某些缓冲页建立。注意该索引只能用于等值查询，对范围查询无效。

文件：

参数文件：和oracle的参数文件类似，但只提供了文本模式。

日志文件：

错误日志：以err结尾，不仅记录错误信息，也记录警告信息和正确信息。

慢查询日志：通过设置参数，记录运行时间超过(大于)阀值的查询语句，此阀值可以精确到微秒。也可以记录那些没有使用索引的查询。

查询日志：同慢查询日志，不过记录得更全。

二进制日志：记录对数据库进行更改的所有操作。

表结构定义文件：mysql对数据的存储是以表为单位的（不同于oracle以表空间为单位）。不论使用那种存储引擎，都以frm为后缀来记录表结构定义。

InnoDB使用的文件：

表空间文件：

重做日志文件：和oracle的重做日志文件类似，文件名为ib\_logfile0，ib\_logfile1…

表：

InnoDB存储引擎表比较像oracle中的索引组织表。每个表都有一个主键，如果用户没有显式地指定主键，则（1）：如果有唯一索引，则使用该索引列为主键（2）引擎自动创建一个6字节大小的指针。

存储结构：按表空间存放，表空间划分为段（segment）、区（extent）、页（page），其中页也被称为块（block）。和oracle比较类似。

page 130页