

Nhibernate 性能问题性能问题 02—跨线程访问、无限结果集,无显示事务提交/回滚

这几个问题也是在开发中存在的，跨线程访问，暂时还没有碰到过实际使用场景。无限结果集相信都很熟悉。为了避免 NH 在一次查询中返回一个无线庞大的结果集。那么。我们通常的解决手段是分页。第三个问题也是比较常见的，正如第一篇中说道的一样，显示的使用事务，但是往往容易遗忘显示的使用事务提交或者回滚。

为什么不能访问无线结果集。庞大的数据一下子驻进内存会直接导致服务器响应缓慢甚至直接发生内存溢出宕机。

好，分别说下把。

针对 NH 的跨线程使用，Nhibernate 的 ISession 并不是线程安全的。所以，如果我们需要跨线程来对 ISession 访问。那么要尤其注意线程同步(关于线程同步的知识。请自行查阅线程安全的相关知识和 dem0)。

可能还是有些抽象。假设一下是使用场景。使用多线程异步请求加载数据，也就是不同的线程使用同一个 Session 来获取数据，比如，主线程请求线路数据，然后在主线程执行的代码中创建新线程，并且让他用同个 Session 去请求站点数据。这种使用情景，从节省资源的角度来讲其实是合理的。但是，要特别注意线程同步。上面的那个场景仅仅是我能想到的一个比较简单的场景。而且那里面跨线程，也未必会涉及到公共资源。所以脏数据写入或者读取，只能说可能性很小。给一点代码。下面这小段代码就是模拟跨线程访问 Session 的，看下。

```
using(var session = factory.OpenSession())
{
    var wait = new ManualResetEvent(false); // 1
    ThreadPool.QueueUserWorkItem(state => // 2
    {
        using(var tx = session.BeginTransaction())
        {
            session.Get<Person>(1);

            tx.Commit();
        }

        wait.Set();
    });
    wait.WaitOne(); // 3
}
```

- 1 初始化一个线程，消息通知，详细使用见 ms 官方，电梯=>[ManualResetEvent 的使用](#)
2. 线程池，开启一个线程，lamda 表达式内部为新开启线程要执行的方法，很好理解。新开个线程，异步去请求 id 为 1 的 Person
3. 3 处主线程阻塞等待一个通知，在通知未到之前，处于阻塞状态。Wait.set()，另一个新启动的线程中发出通知消息。Wait.set()代码执行完毕后。3 处代码所在线程(主线程)立马释放。嗯。差不多这样把。这个就是跨线程访问相同 Session 的一个小场景。不多说。

但是，如果是一些很复杂的使用场景，尤其是跨线程访问了相同资源。比如跨线程对同一个或者同一批相同的数据进行了操作。那么，在这种情况下，完全有极大可能数据出现差错。

所以，如果确实碰到了一些需要跨线程访问 `ISession` 的场景。那么，一定要确保线程的同步。嗯，这种场景确实没碰到过。

我个人不建议跨线程访问 `ISession`，保证访问 `ISession` 和创建 `ISession` 的线程为同一线程。不会有啥问题的。Ok。再简单讲下无限数据集返回的情况。

在使用 `Nh` 过程中，如果我们不确定到底数据会有多少的时候，那么必须使用 `SetMaxResults()` 限定最大数据行，如果你用的 `NH` 的命名查询。一般是调用存储过程返回数据。做报表是常见使用场景。直接使用 `sql` 语句做命名查询，太少见。。。。。

Ok，甭管是存储过程还是，显示的 `sql` 查询，要限定最大数据行。用什么 `top`, `limit` 等等等。。。

通常，开发和测试阶段，数据都是很少的。但是，如果上线运行了。不限定返回数据行是一个不定时的炸弹，网络流量大，`server` 挂掉或者 `explorer` 死掉。

看点代码，`NH` 分页，很好懂，不多说。

```
session.CreateQuery("from tk_id,Price,TicketNo,OrderNo tb_ordertickets")
    .SetFirstResult(0)
    .SetMaxResults(25)
    .List();
```

mySql 使用 `limit` 限定数据行。

```
IsRMB) tb12
ON tb11.Vipid = tb12.VipId
AND tb11.IsRMB = tb12.IsRMB) tb10
ON tb9.VipId = tb10.VipId
AND tb9.IsRMB = tb10.IsRMB )tb13
WHERE (puserName IS NULL OR tb13.UserName LIKE CONCAT('%',puserName,'%'))
AND (pcardNo IS NULL OR tb13.CardNo=pcardNo) LIMIT pageIndex , pageSize
UNION ALL
```

Mssql，使用 `row_number` 函数进行分页。

```

select * from products as p left join suppliers as s
on p.supplierid=s.supplierid

select * from (select p.productid,p.productname ,row_number() over
(order by p.productid asc) as rowNO from products as p inner join
suppliers as s
on p.supplierid=s.supplierid ) as c where c.rowNo>10 and c.rowNo<=20

```

也不多说。Oracle 也是有这个的。不介绍。

嗯，如果你需要分页显示。当然，限定数据行一定是必须的。如果不需要分页。需要返回更多数据，无非是传入页数据行的大小问题。

看第三个问题。无显示提交/回滚事务

```

using(var transaction = session.BeginTransaction())
{
    var person= session.Get<Person>(5);
    person.AddComment(comment);
}

```

显示的开启一个事务，从 Person 表取 id 为 5 的对象，有问题么？

有问题，一旦开启事务，必须显示的进行事务提交和回滚操作。

就像这样

```

using(var transaction = session.BeginTransaction())
{
    var person = session.Get<Person>(5);
    person.AddComment(comment);
    transaction.Commit()
}

```

当然事务块内如果是设计到多表，或者一些复杂存在潜在异常的操作，那么还需要加上 try-catch..在 exception 内添加 rollback 的使用。

记得事务和 commit/rollback 配对使用。

如果是配置了二级缓存，在第一篇我已经提到过。二级缓存与 db 进行数据同步的标志是事务提交操作(显示的调用 commit)。

如果未使用 commit，二级 cache 不会缓存任何对象，即使再进行相同对象的获取，Nh 的处理一定是重新向 db 发出 sql 进行数据请求，而不是直接从缓存中读取。如果无事务提交操

作，配二级 cache 又有何意义呢。

2013-10-20