**CCFlow5执行效率**

**测**

**试**

**报**

**告**

**2013年1月26号**

目录

[1流程运行与外界那素有关系? 3](#_Toc347406472)

[1.1外部因素 3](#_Toc347406473)

[1.2引擎内部因素 3](#_Toc347406474)

[2影响ccflow运行效率的4大重要表 3](#_Toc347406475)

[2.1流程信息注册表WF\_GenerWorkFlow 4](#_Toc347406476)

[2.2流程工作人员表WF\_GenerWorkerlist 4](#_Toc347406477)

[2.3流程实例表 NDxxRpt 4](#_Toc347406478)

[2.4流程轨迹表 NDxxTrack 4](#_Toc347406479)

[3为提高效率采取的优化措施 5](#_Toc347406480)

[4.对ccflow极限测试报告 5](#_Toc347406481)

[4.1极限测试流程条件 5](#_Toc347406482)

[4.2测试环境 5](#_Toc347406483)

[4.3测试结果 6](#_Toc347406484)

[5 对ccflow5的压力测试报告 6](#_Toc347406485)

[5.1压力测试步骤 6](#_Toc347406486)

[5.2数据为100万时执行的结果 6](#_Toc347406487)

[6并发测试报告 7](#_Toc347406488)

[6.1测试环境 7](#_Toc347406489)

[6.2极速测试结果 7](#_Toc347406490)

[6.3单条流程100万数据测试结果 7](#_Toc347406491)

# 1流程运行与哪些因素有关系

**总述：**

一条流程就类于一辆汽车，它的运行速度与它的负载(载重货物以及使用的附加功能(比如启用空调)有关系。

我们研究一下调用ccflow的API过程，如下图：  


AB点之间的时间段是ccflow执行一个接口所需要的时间。A点与B点时间之外的点是开发者根据自己的业务需要执行的业务逻辑。

我们所研究的就是AB这段时间内执行的外部因素与内部因素，分析这些因素，以尽可能的提高ccflow的执行效率。

## 1.1引擎外部因素

1. 硬件情况。
2. 网络环境。
3. 用户并发数。

## 1.2引擎内部因素

1. 节点表单字段数据的大小。
2. 方向条件的多少。
3. 是否启用消息机制。
4. 节点设置的内容。
5. 配置在流程引擎上的事件多少与事件的执行效率。
6. 是否有单据生成。
7. 流程引擎中的４大表内的数据量、流程注册表、流程工作人员表、数据实例数据表、流程轨迹表（也叫流程日志表）。

# 2影响ccflow运行效率的4大表

## 2.1流程信息注册表WF\_GenerWorkFlow

**此表用途：**

流程启动时，要向此表中写入一条数据，这个表用来控制流程运行的状态、运行的位置以及发起人的基础信息。

流程实例完成后，就会清除这条记录。

**此表的主键与索引:**

以int 类型的WorkID 做为主键。经常用到FK\_Flow和 Starter 两个字段做查询，需要对它们建立索引。

## 2.2流程工作人员表WF\_GenerWorkerlist

**此表用途：**

流程启动时，要向此表中写入一条数据，以标明谁是开始节点的参与人员，以后在每个节点的运动过程中，节点的每个参与人员都会有一条数据。

它是WF\_GenerWorkFlow的从表，用来记录流程实例的参与者，如果一个流程实例完成了，ccflow便会清除它。

**此表的主键与索引:**

以int 类型的WorkID 、 string类型的FK\_Emp、int类型的FK\_Node三个键做为主键。经常用到FK\_Node和WorkID 两个字段做查询、删除或者更新。需要对它们建立索引。

## 2.3流程实例表 NDxxRpt

**此表用途：**

一个流程有一个实例表，它默认的命名规则是以ND+ int.Prase(FlowNo)+Rpt组成的，您可以在设计流程前指定这个表名。

此表存储流程运行的信息，比如发起人、发起时间、当前节点、结束人、流程状态等，有一些字段与WF\_GenerWorkFlow重复，但是流程运行完成后，此表的数据不删除。此表方便用户对它进行统计分析、查询。

**此表的主键与索引:**

以int类型的WorkID做为主键，经常以WFState与Rec 两个字段做查询，所以要把它们设置成索引。

## 2.4流程轨迹表 NDxxTrack

**此表用途：**

一个流程有一个轨迹表，我们也称为日志表，它默认的命名规则是以ND+ int.Prase(FlowNo)+ **Track**组成的。

此表存储对流程执行的所有动作，比如：发送、退回、转发、移交、删除…… 它忠实地记录用户对一个流程实例操作的过程，没有删除的操作，所以此表包含的数据最多。

**此表的主键与索引:**

以int类型的MyPK做为主键，经常以WorkID字段做查询，所以要把它们设置成索引。

# 3．为提高效率采取的优化措施

1. 尽可能事先计算好节点与流程设置的信息，比如：在设计时就计算好节点的方向、一个节点下面有几个节点与它连接、此节点是否绑定了单据打印、是否设置了消息订阅、是否有事件。事先计算有助于减轻在流程运行时的SQL访问。
2. 流程引擎表与流程实例数据表的主键与索引尽可能用int类型，根据经常用的查询来设置索引。
3. 大量采用实体缓存，避免数据库的多次交互操作。
4. 对数据库的访问以参数的方式执行:  
   多达95%以上的sql都是用参数执行的，您可以通过监控工具捕获到它们。CCFlow运行在bp框架上，所有的实体类都放在SQLCash这个类中，所有的update,delete,insert,select操作都是采用参数执行，生成的sql放入缓存里。

# 4.对ccflow极限测试报告

此处的ccflow极限测试是指在最低负载、最低启用外部功能的情况下所能跑出的最大速度，也就是引擎极限功能。

## 4.1极限测试流程条件

1．最简单的线性流程只有三个节点：开始、中间、结束。

2、每个节点表单有最少的字段数。

3、流程模式为合并节点表单模式。

4、流程引擎表的数据量与流程实例数据量都为0。

5、测试方式是循环调用ccflow的API接口，这三个节点的应用场景为开始节点发送、处理中间点、结束流程。

**可以预测的结果：**开始节点消耗会大，因为它启动流程时访问的表插入的数据较多，中间点相对较小，结束点没有寻找下一个节点的任务，但是处理结束时逻辑消耗的资源较多。

## 4.2测试环境

软件环境：win7，services pack1. 32位, IIS7+sqlserver2008.

硬件环境: 笔记本 内存: 2.00GB CPU: 2.53 GHZ

## 4.3测试结果

CPU使用率在18-25之间, 物理内存在81%左右。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **节点名称** | **运行次数** | **时长/秒** | **每秒完成数** | **分析** |
| 开始节点 | 1000 | 38.8656988 | 25.73条/秒 | 说明：一秒可以发起25.73个流程。  开始节点在启动时所做的事情比较多，消耗资源相对较大。 |
| 中间点 | 1000 | 28.9224508 | 34.57条/秒 | 说明：一秒可以处理34.57个中间点的工作发送。 |
| 结束点 | 1000 | 32.8636713 | 30.43条/秒 | 说明: 每秒处理30.43个流程结束点。结束点不向下执行发送，但是结束工作需要做的事情与向下发送消耗的资源大致相抵. |

Ccflow已经为您准备好了测试代码，安装好ccflow的DEMO环境后就即可执行。

http://localhost/wf/Testing/Test3Node024.aspx

# 5 对ccflow5的压力测试报告

## 5.1压力测试步骤

为流程引擎表压入100万个待办工作、为工作者列表压入200万条数据、为一个待测试的流程实例表与流程日志表各压入100万条数据，正常情况这下，这对一个省局单位来说应该足够大。

其它：压入数据的要求必须是随即产生的。

## 5.2数据为100万时执行的结果

CPU使用率在18-25之间

物理内存在81%左右

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **节点名称** | **运行次数** | **时长/秒** | **每秒完成数** | **分析** |
| 开始节点 | 1000 | 58.6873031 | 17.64条/秒 | 向流程信息注册表插入数据,故导致数据比中间点变慢. |
| 中间点 | 1000 | 23.4468412 | 42.66条/秒 | 解释不清楚为什么出现这样的结果。 |
| 结束点 | 1000 | 307.567998 | 3.25条/秒 | 最后要清除流程信息注册表与工作人员表，由此导致执行删除的sql变慢。 |

# 6对ccflow5的并发测试报告

## 6.1测试环境

软件环境：win7，services pack1. 32位, IIS7+sqlserver2008.

硬件环境: 笔记本 内存: 2.00GB CPU: 2.53 GHZ

## 6.2极速测试结果

（正在改造程序适应测试工具，尚未完成）

## 6.3单条流程100万数据测试结果

（正在改造程序适应测试工具，尚未完成）

**==================== 对ccflow5 有效 ====================**