第32卷 第20期 计 算 机 工 程 2006年10月
Vol.32 № 20 Computer Engineering October 2006
·软件技术与数据库· 文章编号：1000—3428(2006)20—0103—03 文献标识码：A 中图分类号：TP393.06

实时协同编辑系统的容错流量控制研究

王书斌，朱晓旭，吕 强
(苏州大学计算机科学与技术学院，苏州 215006)

摘 要：为了有效利用网络带宽、降低网络流量，并在保持逻辑向量时钟的检查点全局一致基础上，使得系统在恢复复制进程时，做到网络流量小、等待时间短、用户费用少，容错系统将依据编辑文档的编辑距离、编辑相似度，以最小的代价为恢复进程提供最新检查点和最新文档，实现广域网环境下优化的实时协同编辑系统。提出流量控制的算法，并通过 OPNET 建模和实验测试，对整个系统容错时的流量进行了分析。

关键词：容错；计算机支持的协同工作；流量控制；编辑距离；编辑相似度

Flow Control Research on Fault-tolerant in Thereal-time Cooperative Editing System

WANG Shubing, ZHU Xiaoxu, LV Qiang
(School of Computer Science and Technology, Soochow University, Suzhou 215006)

【Abstract】 On the base of maintaining the consistence of global snapshot, the flow control needs to satisfy the low thought-out, short waiting time and low cost in WAN. The fault tolerant system provides the low cost of ov or editing document on the foundation of editing distance and editing similarity during the process of editing, optimizing the real time editing system in WAN. The algorithms are put forward about flow control. Then its performance in the system is analyzed by tool of OPNET.

【Key words】 Fault-tolerant; CSCW; Flow control; Editing distance; Editing similarity

计算机支持的协同工作(CSCW)可以在广域网环境下将不同的站点联系起来，允许多个协作者同时阅读和编辑同一篇文档。而容错技术为系统提供容错能力，提高整体可靠性，保持系统始终处在安全状态。即使错误发生，系统也能够继续执行，一旦错误消失，新创建的复制进程将取代原来崩溃进程继续执行。在实时协同系统中，无论是在集中式、复制式还是混合式模式[7]下，地理上分布的协作者需要将实时的编辑文档或者编辑的操作在各个编辑者之间传输，保持同步，多人协作完成文档的编辑。而带宽是系统的有限、有偿资源。作为用户希望在不影响效率的情况下，尽可能地减少系统带宽占用。

在进程失效之前，每个进程都会将自己的全局检查点和相关的临时文件记录到可靠的存储器上。当某个站点进程失效后，系统需要为这个站点重新创建一个新的复制进程，此时，系统需要其它的进程为新创建的进程提供最新的文档和最近的操作。在这个过程中，许多 CSCW 系统，如 REDUCE 只是通过重新登录，重传文档，重新加入编辑系统。用户不得不花长时间等待文件在互联网上的传输。

本文针对以上的问题，具体从解决协同写作过程中的流量问题入手，解决网络数据流量大而导致数据延时问题。我们在对传输的流量进行分析后，剔除和优化冗余数据，降低传输的数据流量，实时传送最新的编辑信息。在分析和比较多种策略的基础上，针对 CSCW 的网络流量，提出了自己的一个解决方案。

1 常用容错流量控制

容错系统的目标是能透明地进行故障恢复，尽可能地减少由于回滚产生的计算量及尽可能减少由协议产生的额外计算量[1]。而流量控制就是为了实现尽可能降低回滚计算量和协议额外计算量。一般的容错系统基于消息日志的恢复方法记录进程所有的接收和发送消息。按照特定的算法为每个协作进程作全局检查点。当某个复制进程发生失效时，新创建的复制进程借助周边已知站点进行站点恢复。常用的方法主要分为 2 种：一种是基于最新文档的恢复策略，就是将其它站点(如服务器)的最新文档重新复制一份，如REDUCE、COWORD；另一种方法是基于将两个全局检查点之间的操作向量差进行传递，如RECIPE[5]。这些实时编辑系统提供的恢复流量方法只是局限于功能上可用，并没有考虑容错系统目标的实现。

2 容错流量控制的形式化描述

本节形式化地引入一个简单的容错流量控制系统。

定义 1 每个站点都有一个历史记录缓冲HB，对于每个站点$S_i(0<i<n)$，记录的每次操作O_j，则有

$$HB = \sum_{i=0}^{n} \sum_{j=0}^{m} O_{ij}$$

定义 2 逻辑时钟 v 在分布式计算系统中，逻辑时钟可以对事件唯一排序。

定义 3 检查点用来维护全局快照的一致性的时间点。主要目的是实现以下 2 个一致性，从而维护数据收敛。

基金项目：“211”工程重点建设学科基金资助项目(x2118004)
作者简介：王书斌(1978－)，男，硕士生，主研方向：计算机支持的协同工作(CSCW)，分布式计算；朱晓旭，讲师；吕 强，教授
收稿日期：2005-11-09 **E-mail：**wangsoobin@sina.com

定义 4 σ_p是一致的(consistent)，当且仅当满足因果关系[2]。即

$$\forall e_1, e_2 : \wedge \begin{pmatrix} e_2 \in \sigma \\ e_1 \to e_2 \end{pmatrix} \Rightarrow e_2 \in \sigma$$

如果进程对象执行中没有失败，则其对象满足一致性。

定义 5 σ_p是*l*- consistent。进一步定义如下：

$$\forall e_1 : executedbyp \wedge q \in \forall_{e_2} : \wedge \begin{pmatrix} e_2 \in \sigma_P \\ e_1 \to e_2 \end{pmatrix} \Rightarrow$$
$$\forall_r : e_1 \in P \wedge (Service(r) = Service(q) : e_1 \in \sigma_P)$$

如果集合 *P* 没有发生崩溃，则定义 1 和定义 2 等价。

定义 6 操作向量 *ov*：将编辑过程的基本操作——插入、删除、替换，加上逻辑时钟以及相关的进程信息构成操作向量 *ov*。

3 编辑动态变更的评估标准

3.1 LCS (Longest Common Subsequences)

最长的相同子串序列：在字符集∑中，*X*含有字符串序列 $x_1, x_2, \ldots, x_n$，而*Y*含有字符串序列 $y_1, y_2, \ldots, y_m (n \le m)$。*X*和*Y*的最长的相同匹配子串序列就是LCS，当然其中存在相同长度的多对子字符串序列[4]。数学递归定义如下：

$$L[i,j]=\begin{cases} 0 & i=0 \text{或者} j=0, \\ L[i-1,j-1]+1 & i,j>0 \text{并且 } a_i=b_j, \\ \max\{L[i-1,j], L[i,j-1]\} & \text{其它情况} \end{cases}$$

3.2 编辑距离

编辑距离[3]：在字符集∑中，两个字符串x和y的编辑距离d(x,y)是通过字符符号的插入、删除、替换这一系列的操作，将x转化成y所需要的最小花费：

$$d(x,y) = \min\{\ c(\omega) : h(\omega) = (x,y)\ \}$$

其中h(ω)是x向y转化的一个序偶队列：h(ω) =(x,y)；如果各元素的转化代价的权值相等，则当a≠b时，c((a,b))=1；否则c((a,b))=0。c(ω)是转化代价，ω=$\omega_0\omega_1\cdots\omega_n$，c(ω) 是各元素的转化代价之和，即

$$c(\omega) = \sum_{i=0}^{n} c(\omega_i)$$

3.3 编辑相似度

编辑相似度计算方法 1：词串*X*含有单词$x_1, x_2, \ldots, x_n$，而词串*Y*含有单词 $y_1, y_2, \ldots, y_m$。其中对于每一个x_i, $1 \le i \le n$，而对于每一个y_j，$1 \le j \le m$，令

DoS_{xi} $(1 \le i \le n)$表示x_i单词和*Y*的匹配数目；

DoS_{yj} $(1 \le j \le m)$表示y_j单词和*X*的匹配数目。

词串 X 和 Y 的相似度为

$$SIM_F^{RS}(X,Y) = \frac{\sum_{i=1}^{n} DoS_{xi} + \sum_{j=1}^{m} DoS_{yj}}{n+m}$$

编辑相似度[6]计算方法 2：假设SSNC 代表词串X 和词串Y 相同连续字符数平方和，它由MCWPA 滑动窗口算法实现：

$$SIM_F(X,Y) = \sqrt{\frac{SSNC}{(n+m)^2}}$$

3.4 动态变更的评估标准综合比较

LCS的计算复杂度是*O(mn)*。在确定了最长的相同子串序列后，还需要再次找出差异，更新字符串，在这其中又会涉及到字符串的匹配问题。计算复杂度大。计算编辑距离*d(x,y)*以动态规划算法为代表。而根据实时协同编辑系统的特点，可以利用用户实际操作来计算两个检查点的编辑距离。这样计算的编辑距离达到同样的效果，在编辑的同时，计算编辑*d(x,y)*。编辑相似度计算方法 2 中，词串F_x和词串F_y的相似度计算和编辑相似度计算方法 1 的不同在于计算SSNC值的计算。而针对这个公式提出的MCWPA窗口滑动机制可以快速地进行SSNC的计算，这种窗口模式可以有效回避重复对字符的访问，同时提高算法的时间复杂度。通过对 3 种不同的动态变更的评估标准比较，下面的算法中选用编辑距离和编辑相似度计算方法 2 进行容错流量的控制。

4 容错流量控制方法

4.1 容错算法(FC)

首先查找是否存在与p相连，并且状态正常的复制进程。∃q∧(Q, V_q)广播消息给所有相连进程，完成向量时钟v的最小需要的消息服务Q。如果一个复制进程收到消息，它就会将当前时钟与v比较。如果v的状态是当前状态的前序，它将广播自己的标志。进程p等待从复制进程的消息反馈。如果过了一段时间后，没有反馈，p就认定没有合适的复制进程存在。反之，如果存在，p任意地选择一个。

如果不存在复制进程，CollectRecoveryInfo(Q,V_q)被广播给所有的相连进程，这些进程给进程p返回完成服务Q，所有恢复消息p与V_q保持一致。用函数Recover(Q,*e*,V_q)将消息排序，利用检查点文件，创建一个新的进程，重行执行需要恢复的消息。下一步，恢复协议，重新设置Original(Service(q))的值。如果需要，返回新进程的标识。算法描述如下：

```
procedure FC(process q) {
 q_ex←∃q∧ (Service(q),V_q);
 if(q_ex=ø) then
   e←CollectRecoveryInfo(Service(q), V_q);
q_new←Recover(Service(q), e, V_q);
if(Crashed(q)∧Original(Service(q))=q) then
   reset Original(Service(q));
else
   q_new←q_ex;
return q_new ; }
```

4.2 系统容错流量控制算法

系统容错流量控制算法描述如下：

```
procedure CollectRecoveryInfo(Service(q), Vq) {
ov←chck (Vp)－chck (Vq);
λ← ov /L(F (Vq));
if (λ>>1)    return send(F,opnew);
S←SIMF(Service(σp),Service(σq));
D←d(HBp,HBq);
if ((L(F (Vq)×S>D) then
   V←HBq－{Oundo∧Oredo};
   send(V, opnew);
else
   send(F, opnew); }
```

在CollectRecoveryInfo(Service(q), V_q)流量控制算法中，系统首先比较当前进程q和需要新创建复制进程q相隔的检查点。通过逻辑时钟比较，将两个检查点之间的操作，组织成操作向量*ov*。如果σ和最新文件的长度比值远远大于 1，则重传最新的文档和相关最新的文档操作。否则计算编辑相似度和编辑距离(为了提高效率可以将它们的计算合并到检查点的记录中，此时只需要取这两个参数即可)。然后根据编辑相似度和编辑距离，比较恢复传输的时间复杂度和空间复杂度，决定具体的传输量。同时，对*ov*作进一步的优化。

5 容错流量控制实验分析

5.1 建立模型

针对以上的算法，下面给出在OPNET上的建模。如图 1

所示，每个站点S_i都维持着一个安全状态。当状态开始时，进程自动进入初始状态init。在这个状态，进程会检查上次运行是正常退出，还是从故障中恢复。init状态发送join或者recover消息给它的邻接站点并且等待回应。当消息到达站点时，该站点会检查来自不同站点的消息是否满足一致性。如果发生非一致性，则系统要求重做。如果是进行进程恢复，则提供创建新进程的相关上下文。

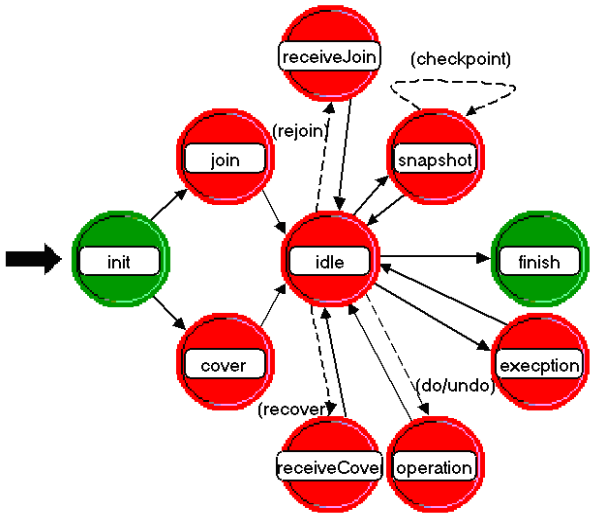

图 1 进程状态转换

针对传统模式下的容错系统的模型。对于 WAN 普通用户上传最高速率为 1 024bps，不考虑时延，每隔 20s 做一次全局检查点，文档大小约为 1MB，文档的修改服从正态分布。

5.2 实验结果分析

通过图 2 可以看到，该模型单纯以文档方式编辑，而操作向量在两个相隔检查点之间作线性单调递增。传统的方法不采取流量控制时，文件传输随着文件大小的递增而增加。而只传输操作向量时，取决于前后两个检查点文档版本的编辑距离。在图 3 中可以更加清楚地看到这一点。

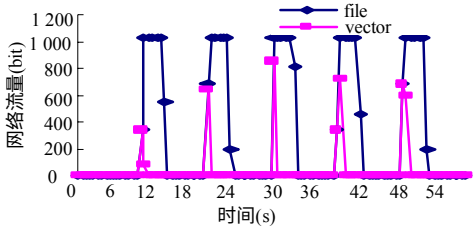

图 2 速率为 1 024bps 下的网络流量

如果文件很大，或者文档短时间内进行了大量操作，那么用这种传统的方法使传输出现了明显地增加。而对恢复的复制进程，二者都可以进行有效的恢复。如果单纯地考虑网络的流量，可以取传输流量最少者完成传输要求。但此时忽视了恢复对恢复进程的影响。只是单纯地考虑网络流量，没有考虑在各个复制进程的执行期间空间/时间的复杂度。对于一个实时系统而言，如果恢复时使用操作向量需要从父亲全局一致点或者祖先全局一致点将文档计算到当前的全局一致的文档，然后再更新从这个一致点开始的其它操作。而对于文档的操作则是将此时的文档计算好或者取出临时保存的版本，交付给复制进程。将最后一个检查点之后的操作向量更新。整个计算的时间复杂度直接与文档的编辑相似度和文档的编辑距离相关。通过算法，再综合比较时间/空间复杂度，决定传输的最优方法。虽然某个时刻会牺牲少量的带宽，但换得的是用户更快地恢复响应时间和整体性能的优化。

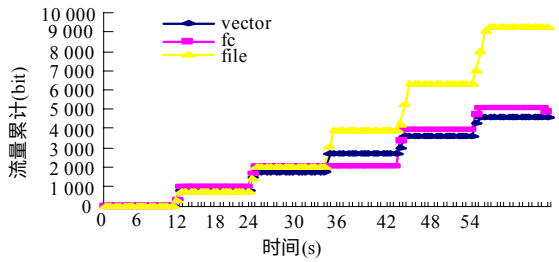

图 3 速率为 1 024bps 下的网络流量

6 结束语

文中引入编辑动态变更的评估标准，设计了一种容错流量控制方法，有效地支持集中式和复制式结构的协同编辑系统，减少了网络流量，提高了系统整体性能。随着实时协同系统在互联网上的发展，对整个编辑系统的网络性能会提出更高的要求。下一步工作将针对大型文件，进行上下文切分或者使用压缩算法，进一步减少网络的流量，提高实时编辑系统的响应速度。

参考文献

1 Shim H S, Tolerating A P. Client and Communication Failures in Distributed Groupware Systems[C]. Proceedings of the 17th IEEE Symposium on Reliable Distributed Systems, 1998: 221-228.

2 Bhatia K. Dynamic Replication in Wide Area Environments Using Message Logging[D]. University of California, 2001.

3 Mohri M. Edit-distance of Weighted Automata, General Definitions and Algorithms[J]. General Definitions and Algorithms, 2003, 14(6): 957-982.

4 Wangner R A. The String-to-String Correction Problem[J]. Journal of the ACM, 1974, 21(1): 168-173.

5 Shen H, Recipe S C. A Prototype for Internet-based Real-time Collaborative Programming[C]. Proceedings of the 2nd Annual International Workshop on Collaborative Editing Systems in Conjunction with ACM CSCW Conference, Philadelphia, Pennsylvania, 2000-12.

6 Yuan S S, Yang Q X. Faster Algorithm of String Comparison[M]. Springer-Verlag, 2003.

7 Rodden T, Sawyer P, Sommerville I. Architectural Support for Cooperative Multi-user Interfaces[J]. IEEE Computer, 1994, 27(5): 37-46.