

一种可靠的消息传输机制

夏盛明,尹宝林

(北京航空航天大学 计算机学院,北京 100083)

摘 要:分布式系统中节点之间的异步消息传递常通过消息传输代理进行。消息代理节点如果不可靠,将直接影响到消息的传递。文中提出了组代理机制,利用一组节点作为消息传输代理共同传递消息,通过增加消息传递的途径解决了由单个转发节点失效而阻碍消息传递的问题,提高了消息代理传递消息的可靠性。组代理机制实现了各节点对重复消息的正确处理,使接收节点实现了对重复消息的过滤,保证了最终接收数据的唯一性。最后通过模拟实验证明了组代理机制提高了消息代理的可靠性。

关键词:异步消息;组代理机制;可靠;重复消息

中图分类号:TP316

文献标识码:A

文章编号:1673-629X(2009)10-0030-03

A Reliable Message Transfer Mechanism

XIA Sheng-ming, YIN Bao-lin

(School of Computer Science and Engineering, Beihang University, Beijing 100083, China)

Abstract: Nodes in the distributed system often transfer asynchronous message through the message transfer agent. The message transfer agent was usually a single node. If the node was unreliable and easy to fail, it would affect the delivery of the messages. Proposed a new way of using a group of nodes as a message transfer agent to deliver messages, called group message transfer mechanism. It increased the reliability and prevented the delay of message transfer caused by the failure of a single node as the message transfer agent. Using a group of nodes to transfer messages would cause duplicate messages to arrive at the target node. Group message transfer mechanism provided ways to filter all the duplicate messages to ensure that no duplicate messages would be accepted by the target node. Last in this paper, the idea was proved by the experiment.

Key words: asynchronous message; group message transfer mechanism; reliable; duplicate message

0 引言

分布式系统需要使不同的处理单元(PE)上并行运行的各部分协同工作,而消息传递作为一种进程间松耦合的通信手段,是分布式系统节点之间通信的重要手段^[1,2]。分布式系统中消息的传递主要采取了异步方式^[3]。基于代理的消息传输方式,消息发送节点将消息存储在消息传输代理上,由消息传输代理负责将消息传递给接收节点。在由单个节点作为消息传输代理的情况下,如果传输代理节点产生故障将导致消息无法传递^[4,5]。

组代理机制通过使用一组节点作为消息传输代理转发消息,增加了网络中消息传递的途径来提高消息传输的可靠性。通过组代理机制传递消息,只有在所

有组代理节点都无法访问发送节点和接收节点时,消息才无法得到传递。从消息成功传输的概率上看,假设使用单个节点作为消息传输代理转发消息成功的概率是 p ,则使用 n 个节点组成消息传输代理,成功传递消息的概率是 $1 - (1 - p)^n$,因为 $p < 1$,所以 $1 - p < 1$,可得 $(1 - p)^n < 1 - p$,进而推出 $1 - (1 - p)^n > 1 - (1 - p)$,即 $1 - (1 - p)^n > p$ 。

可见,一组节点共同转发消息比单个节点转发消息更可靠。但是,组代理机制由于使用了多个节点传递消息,需要解决如下问题:如何确保消息由多个节点实现可靠传递;组代理节点上可能产生的消息堆积问题;各节点如何处理重复消息,接收节点怎样保证接收数据的唯一性等^[6]。

文中假定只要任意两个节点在网络上的有效时间范围存在重叠,则这节点之间就可成功实现消息的传递。将介绍组代理机制内容的相关内容,并进行了分析,最后进行了消息传输模拟实验,实验结果证明了组代理机制提高了消息传输代理的可靠性。

收稿日期:2009-02-06;修回日期:2009-05-17

作者简介:夏盛明(1982-),男,硕士研究生,研究方向为分布式应用技术;尹宝林,教授,博士生导师,研究方向为汉语语音识别、分布式应用技术等。

1 组代理机制模型

组代理机制模型中只考虑对一条数据消息的传输情况。多条消息的传递可依此情况类推,多条消息传递的顺序性要求可由上层应用消息队列维护^[7-9],这里将不予考虑。消息组代理模型如图 1 所示。

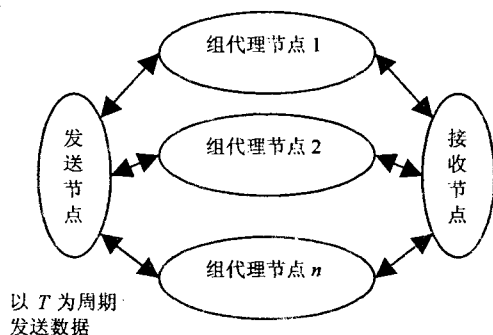


图 1 组代理机制模型

1.1 组代理模型组成元素

发送节点是消息的源节点;接收节点是消息的目的节点;组代理节点是组成消息传输代理的节点,用于共同传递发送节点和接收节点之间的消息;节点之间的有向连线则代表节点之间进行的消息传递。 T 是发送消息的时间周期。

1.2 组代理机制中传递的消息类型

消息组代理机制中传递的消息包括了三种类型:DATA 消息,ACK 消息和 FIN 消息。DATA 消息是由发送节点向接收节点传递的数据消息,包含了所要传递的数据信息;ACK 消息是由接收节点向发送节点传递的确认消息;FIN 消息是由发送节点向接收节点传递的完成消息。

2 组代理机制的数据可靠传输协议

数据可靠传输协议用于发送节点和接收节点之间的协商,具体内容如下:发送节点将从消息发送队列取出 DATA 消息向接收节点传递;接收节点接收到 DATA 消息后,将把 DATA 消息保存在消息接收队列,然后将 ACK 消息传递给发送节点;发送节点在接收到 ACK 消息后,删除在消息发送队列中保存的 DATA 消息,然后将 FIN 消息传递给接收节点,通知接收节点完成了消息的传递。接收节点在接收到 FIN 消息后,可将 DATA 消息提交给上层应用接收,并在接收队列中继续保留 DATA 消息,经过延迟时间 T 用于过滤可能接收到的重复 DATA 消息,然后将 DATA 消息删除。如图 2 所示。消息组代理节点负责在发送节点和接收节点之间传递消息,将删除未能在时间 T 内传递的 DATA 消息,并持久存储 ACK 消息和 FIN 消息直到消息成功得到传递。

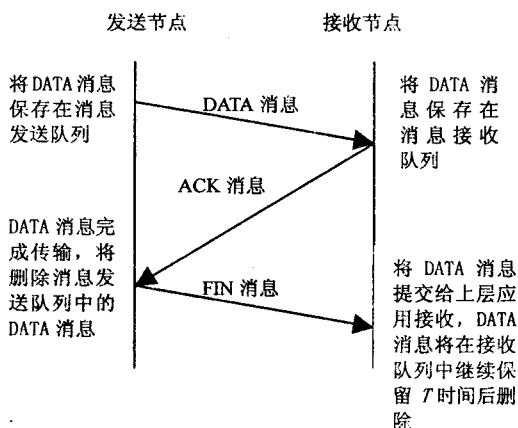


图 2 数据可靠传输协议

3 节点对消息的存储方式

发送节点上不存储 FIN 消息。只将 DATA 消息存储在消息发送队列中并向接收节点传递。在接收了 ACK 消息后,发送节点将先保存 ACK 消息,然后把 DATA 消息从消息发送队列中删除,然后再删除 ACK 消息。

组代理节点对 DATA 消息的存储由时间 T 限制。自接收时间算起超过时间 T 仍未能传递给接收节点的 DATA 消息将被删除。组代理节点将持久存储 ACK 消息和 FIN 消息,对 ACK 消息和 FIN 消息的存储没有时间上限制。ACK 消息和 FIN 消息只有在成功转交后,才会被删除。

接收节点不存储 ACK 消息。接收节点把接收到的 DATA 消息存储在消息接收队列中。在接收到 FIN 消息后,接收节点先保存 FIN 消息,提交 DATA 消息,再经过时间 T 延迟后,才将 DATA 消息从接收队列删除,最后删除 FIN 消息。

4 对组代理机制的分析

4.1 组代理机制实现了消息可靠传输

组代理机制通过数据可靠传输协议实现了发送节点和接收节点对传输数据的同步。发送节点能够确定数据成功到达了目的地,接收节点也能够过滤到达的重复消息,保证了接收数据的唯一性。

4.2 消息不会在组代理节点上堆积

堆积指的是节点不断接收到消息却无法将消息转发出去,导致了消息在节点上的积累。组代理节点在接收到 DATA 消息后,在时间 T 内尝试将 DATA 消息传递给接收节点,超过时间 T 仍未能成功传递给接收节点的消息将被组代理节点删除,防止了无法得到传递的 DATA 消息在组代理节点上堆积。

ACK 消息不会在组代理节点上产生堆积。只有

在组代理节点不断将来自发送节点的 DATA 消息传递给接收节点,而却无法将来自接收节点的 ACK 消息交付给发送节点,才会产生堆积现象,而这是不可能产生的。同理,由 ACK 消息在组代理节点上不会堆积可知,FIN 消息同样不会在组代理节点上堆积。

4.3 节点实现了对重复消息的正确处理

由于使用了多个组代理节点传递消息,发送节点可能接收重复的 ACK 消息,接收节点可能接收到重复的 DATA 消息和 FIN 消息。组代理机制对重复消息做了有效的处理,保证消息传递的正确性。

发送节点收到 ACK 消息后,不论 ACK 消息是否重复都将发送 FIN 消息。

接收节点接收到 FIN 消息后,可确定 DATA 消息完成可靠传输。然后将先查看消息接收队列中是否已保存有此 FIN 消息。如果消息接收队列中已有此 FIN 消息,那么接收消息是重复的,接收节点将丢弃接收的 FIN 消息。如果消息接收队列中没有此 FIN 消息,接收节点将把 FIN 消息保存在消息接收队列中。接收节点将接着查询消息接收队列中是否存在 FIN 消息关联的 DATA 消息,如果不存在 DATA 消息,那么 FIN 消息也是重复的,接收节点将把 FIN 消息删除。如果存在 DATA 消息,接收节点提交 DATA 消息内容,在经过时间 T 延迟后,接收节点将把 DATA 消息移出消息接收队列,最后将把 FIN 消息删除。

接收节点接收到 DATA 消息后,将判断消息接收队列中是否已经存在此 DATA 消息,如果存在此 DATA 消息,说明接收到的 DATA 消息是重复消息,接收节点将丢弃接收到重复的 DATA 消息。接收节点查看在消息接收队列中是否有此 DATA 消息关联的 FIN 消息,如果没有,接收节点将向传递此 DATA 消息的组代理节点发送 ACK 消息,否则不发送。接收节点采用以上方式判定 DATA 消息是否重复,是因为保存消息接收队列中的 DATA 消息在未接收到 FIN 消息前不会被移出消息接收队列,只有在接收到了 FIN 消息后,对应的 DATA 消息在经过延迟时间 T 后才会被移出消息接收队列。

接收节点接收到 FIN 消息后,说明发送节点收到了接收节点发送的 ACK 消息,发送节点确认了 DATA 消息已被接收节点成功接收,发送节点将不会再发送 DATA 消息。而接收节点在接收到 FIN 消息后,经过延迟时间 T 才将 DATA 消息移出消息接收队列,是因为在发送节点发出 FIN 消息后的时间 T 范围内,组代理节点仍可能向接收节点传递此重复的 DATA 消息。如果在接收到 FIN 消息后马上将 DATA 消息移出消息接收队列,那么组代理节点发送的重复 DATA 消息将

被当作新消息而被接收节点接收。在延迟时间 T 内,组代理节点向接收节点传递的重复 DATA 消息将被接收节点识别并被丢弃,而未能在时间 T 内成功传递的重复 DATA 消息将被组代理节点删除。在延迟时间 T 后,发送节点和组代理节点都不可能再传递重复的 DATA 消息了,于是接收节点便可将此 DATA 消息移出消息接收队列。

5 消息传输模拟实验

实验模型中的节点包含发送节点、接收节点和多个组代理节点。消息传输模型中有如下设定:消息在周期时间 T 内,由发送节点随机发送一次;在周期时间 T 内,节点的随机最长有效时间范围与周期 T 的比率是 t ;进行消息传递的代理节点的数目是 n 。

当 $n = 1$ 时,即采用单个节点作为消息传输代理转发消息,只要代理节点和发送节点有效时间上存在重叠,即认为消息得到了成功传递(假设传输代理总是能够将接收来自发送节点的消息传递给接收节点)。

当 $n > 1$ 时,即采用组代理机制转发消息,在周期时间 T 内,如果组代理节点同发送节点和接收节点在有效时间范围上都存在重叠,即认为此组代理节点成功地完成了消息的传递(不考虑时间范围重叠的顺序性因素);如果组代理节点未与接收节点存在有效时间重叠,即使此组代理与发送节点在有效时间范围上存在重叠,仍认为此组代理节点传递消息失败。

通过改变 t 和 n ,得到消息成功传递的概率变化如图 3 所示。

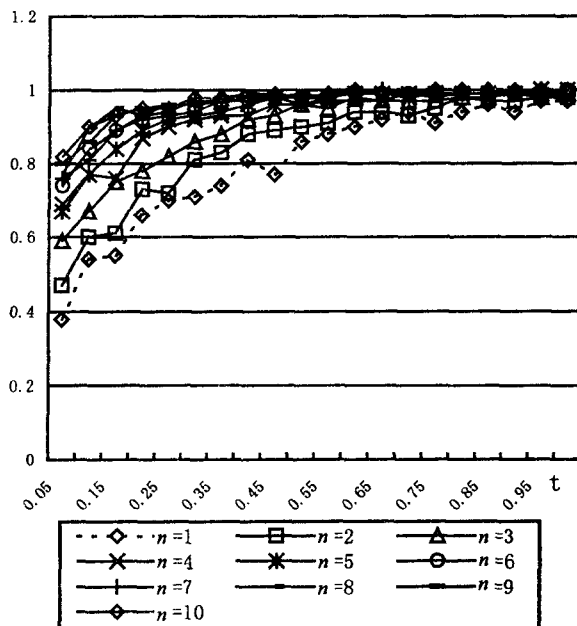


图 3 消息成功传输概率图

(下转第 36 页)

规则。

第三,由实验可见,支持度、置信度框架中如果一个项目大量出现,那么它非常容易成为候选项目集,包括它的 2—项集,3—项集也会大量,并容易被当作关联规则而被挖掘出来,这是否完全正确呢?值得考虑。

总之,兴趣度的衡量框架优于支持度、置信度框架的原因在于计算了各个项目之间相互条件下出现的概率,对单一大量出现的项目不敏感(当然如果它引起其他项目大量出现就会敏感了)。所以,笔者认为,关联规则挖掘时更加应当考虑兴趣度。

5 结束语

文中利用比较直观的方法分析了支持度置信度框架下的两种算法(Apriori 算法和 FUP 算法)和兴趣度框架下兴趣度算法之间的不同点,并作出了结论。事实证明,兴趣度能够更好地体现数据挖掘的初衷。文中还在第 2 章证明了一个推论,能够说明为什么在增量数据库中,支持度和兴趣度都变化的情况下,兴趣度的变化要小于支持度的变化,从而使挖掘出的新挖掘出的关联规则小于支持度挖掘出的关联规则。

由于兴趣度框架还在发展阶段,有很多体系还不完善,比如,虽然兴趣度挖掘出的关联规则少于支持度置信度挖掘出的关联规则,但是否这些规则更能说明问题呢,虽然针对该问题举了一个例子,但要从理论上证明起来,还有待进一步研究。

(上接第 32 页)

从图中可看出,节点有效时间比率 t 较小时,随着 n 的增加,消息成功传输的概率也逐渐增加。因此,使用组代理机制能够使得在节点不可靠的情况下,通过增加组成消息传输代理的节点的数目,提高代理传递消息的可靠性。

6 结束语

文中提出的组代理机制利用多个节点共同转发消息,在提高了代理传递消息的可靠性的同时,也保证了消息传递的正确性。组代理机制克服了采用单个节点传递消息时由于单个转发节点失效而影响消息传递的缺点,在部分组代理节点失效的情况下,消息仍能得到传输。

参考文献:

- [1] Wu Jie. 分布式系统设计[M]. 高传善,等译. 北京:机械工业出版社,2001.
- [2] Sun Microsystems. Java Message Service(JMS)Specification

参考文献:

- [1] Tansel A U, Ayan N F. Discovery of Association Rules in Temporal Databases[C]//Proceedings of the International Conference on Information Technology. [s. l.]: Institute of Electrical and Electronics Engineers Computer Society, 2007: 371-376.
- [2] 黄德才, 张良燕. 一种改进的关联规则增量式更新算法[J]. 计算机工程, 2008, 34(10): 38-42.
- [3] 李志刚. 基于项集支持度的关联规则增量更新算法—BISUA[J]. 计算机工程与设计, 2007(9): 4072-4075.
- [4] Tomas S, Bodagala S, Alsabti K, et al. An Efficient Algorithm for the Incremental Updation of Association rules in Large Databases[C]//In Proceedings of the International Conference on Knowledge Discovery and Data Mining. [s. l.]: Institute of Electrical and Electronics Engineers Computer Society, 1997: 263-266.
- [5] 徐 勇, 周森鑫. 一种改进的关联规则挖掘方法研究[J]. 计算机技术与发展, 2006, 16(3): 77-79.
- [6] 杨泽民. 一种改进的关联规则增量式更新算法[J]. 大同大学学报: 自然科学版, 2007(4): 112-124.
- [7] Lee W J, Lee S J. An Efficient Mining Method for Incremental Updation in Large Databases[C]//4th International Conference on Intelligent Data Engineering and Automated learning. [s. l.]: Institute of Electrical and Electronics Engineers, 2003.
- [8] 程玉胜, 邓小光, 江效尧. Apriori 算法中频繁项集挖掘实现研究[J]. 计算机技术与发展, 2006, 16(3): 58-60.
- [9] V1.0.2[EB/OL]. 1999-11. <http://java.sun.com/jms>.
- [3] Tai S, Rouvellou I. Strategies for integrating messaging and distributed object transactions[C]//IFIP/ACM. International conference on Distributed Systems Platforms and Open Distributed Processing. New York: Springer-Verlag, 2000: 308-330.
- [4] Jalote P. Fault Tolerance in Distributed Systems[M]. USA: Prentice Hall, Inc, 1994.
- [5] Shevat A. Distributed Enterprise Messaging with MantaRay[EB/OL]. 2004-12. <http://www.onjava.com/Pub/a/onjava/2004/12/08/mantaRay.html>.
- [6] 陈建巍, 李 京, 丁 柯. 可靠消息传输服务的设计和实现[C]//第七届计算机科学与技术研究生学术讨论会论文集. 中国计算机学会. 广元:[出版者不详], 2002.
- [7] 王小霞, 陈 亮. 一种消息队列中间件的设计与实现[J]. 计算机工程, 2005, 31(21): 81-83.
- [8] 赵 伟, 周 兵. 基于代理的消息中间件的设计与实现[J]. 计算机工程, 2004, 30(22): 91-92.
- [9] 徐 晶, 许 玮. 消息中间件综述[J]. 计算机工程, 2005, 31(16): 73-76.